LIBR…
2
2002

This is to certify that the

thesis entitled

Simulation of IC Engine In-Cylinder Flows
Using KIVA-3V

presented by

Gurutejas Rao

has been accepted towards fulfillment
of the requirements for

MASTERS degree in MECHNICAL ENGINEERING

[illegible]
Major professor

Date April 23, 2002

SIMULATION OF IC ENGINE IN-CYLINDER FLOWS USING KIVA-3V

By

Gurutejas Rao

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

SIMULATION OF IC ENGINE IN-CYLINDER FLOWS USING KIVA-3V

By

Gurutejas Rao

KIVA-3V is a widely used CFD code for the analysis of transient, 2-D or 3-D, chemically reactive flows with spray. KIVA-3V also has deforming mesh capability for the analysis of geometries with moving boundaries. The most popular application of the KIVA-3V code is the analysis of IC engine in-cylinder flows. Though KIVA-3V is a very useful tool for the analysis of in-cylinder flows, the code possesses a few characteristics that limit its application in this field. In particular, K3VPREP, the pre-processor used to generate the computational mesh for KIVA-3V, can generate only simple, pre-determined shapes that are hard coded in the pre-processor program. Complicated geometries found in modern IC engines cannot be created using the K3VPREP pre-processor. Even if a grid can be generated by K3VPREP, control of vertex distribution in the mesh is very limited. In addition, the H-H structured grid requirement in KIVA-3V causes the y^+ values near grid "corners" to fall below the acceptable range for the standard k-ε turbulence model with wall functions to be valid. This is a potential source of error in the computed solution. This thesis describes and documents a procedure developed at Michigan State University's Engine Research Lab that overcomes the limitations in mesh generation in KIVA-3V. Modifications made to this procedure that allow the creation of canted valve engine geometries are also described in this thesis. Finally, the effect of the H-H grid on y^+ and subsequently on the computed solution is investigated and possible solutions to this problem are provided.

For Amma, Bappa and Muniya

and

Sarah, for waiting…

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Dr. Tom Shih, my advisor, for his guidance, patience and kindness. Dr. Shih helped me take that vital first step towards a better understanding of CFD and its place in the grand scheme of things. His kindness and support continue to this day.

I would also like to thank my fellow desis Uday, Manish, Raka, Lambu and Dilip for being such good friends.

My thanks also to my fellow graduate students Anthony, Xubin, Shih and Leo for their support and help.

And to all those people who made my years at Michigan State so wonderful, thank you.

TABLE OF CONTENTS

LIST OF FIGURES vii

CHAPTER 1
INTRODUCTION 1
1.1 – Overview 1
1.2 – Limitations of KIVA-3V 2
1.3 – Objective 3

CHAPTER 2
TRADITIONAL KIVA-3V MESH GENERATION METHOD 5
2.1 - Overview 5
2.2 - The traditional KIVA-3V mesh generation method 5
2.3 - Limitations of the traditional mesh generation method 11

CHAPTER 3
AN IMPROVED KIVA-3V MESH GENERATION PROCEDURE 13
3.1 – Overview 13
3.2 – The MSU-ERL method 13
3.3 – Limitations of the MSU-ERL method 17
3.4 – Modifications to allow the generation of canted valve geometries 19

CHAPTER 4
PROCEDURE TO GENERATE A CANTED VALVE IC ENGINE MESH 23
4.1 – Overview 23
4.2 – The Template Diagram 24
4.3 – The Gridgen mesh 30
4.4 – Creating the IPREP file 32
4.5 – K3V1PREP, GRIDV and K3V2PREP_PENT 42

CHAPTER 5
EFFECT OF THE GRID ON Y^+ AND THE COMPUTED SOLUTION 44
5.1 – Overview 44
5.2 – Cases studied for this investigation 46
5.3 – Comparison of Y^+ values 47
5.4 – Comparison of the computed solution 53

5.5 – Comparison of the solution with Star-CD 81
5.6 – Conclusions and recommendations 86

REFERENCES 88

LIST OF FIGURES

Note: images are in color.

Figure 2.1
Schematic of the traditional mesh generation procedure 6

Figure 2.2
IC engine geometry and its corresponding Template Diagram 7

Figure 3.1
Schematic of the MSU-ERL method 15

Figure 3.2
Canted valve engine geometry 18

Figure 3.3
Cross section of Gridgen mesh when it is imported into K3V2PREP_PENT 20

Figure 3.4
Intake region rotated to a vertical position 20

Figure 3.5
Valve top surface shape generated by NPROVTOP parameter in IPREP file 21

Figure 3.6
Valve bottom surface shape generated by the
NPROVFCE parameter in the IPREP file 21

Figure 3.7
Cross section of final mesh, with intake region rotated back to its original angle 22

Figure 4.1a
3-D view of the canted valve engine 24

Figure 4.1b
Front view of the canted valve engine 25

Figure 4.1c
Top view of the canted valve engine 25

Figure 4.2a
Front view of the Template Diagram 26

Figure 4.2b
Top view of the Template Diagram 26

Figure 4.3
Template with interior block lines extended 27

Figure 4.4
Template with the lines of blocks above the cylinder extended 28

Figure 4.5a
Final Template diagram with blocks numbered, front view 29

Figure 4.5b
Final Template diagram with blocks numbered, top view without runner blocks 29

Figure 4.6
Gridgen mesh split into sections corresponding to the external blocks in the Template ..31

Figure 4.7
NTYPE 34

Figure 4.8
NREGK3 35

Figure 4.9
Block corner numbering 35

Figure 4.10
Moving surfaces 37

Figure 4.11
Reshape indices 39

Figure 5.1
y^+ in boundary layer 44

Figure 5.2a
cross section of vertical valve IC engine mesh with 50K grid points 45

Figure 5.2b
Cross section of vertical valve IC engine mesh with 400k grid points 45

Figure 5.3a
Grid distribution for 50k 46

Figure 5.3b
Grid distribution for Grid0 46

Figure 5.3c
Grid distribution for Grid1 46

Figure 5.3d
Grid distribution for Grid2 46

Figure 5.4a
y^+ values at 30 degrees after TDC for 50k 47

Figure 5.4b
y^+ values at 30 degrees after TDC for Grid0 47

Figure 5.4c
y^+ values at 30 degrees after TDC for Grid1 47

Figure 5.4d
y^+ values at 30 degrees after TDC for Grid2 47

Figure 5.5a
y^+ values at 60 degrees after TDC for 50k 48

Figure 5.5b
y^+ values at 60 degrees after TDC for Grid0 48

Figure 5.5c
y^+ values at 60 degrees after TDC for Grid1 48

Figure 5.5d
y^+ values at 60 degrees after TDC for Grid2 48

Figure 5.6a
y^+ values at 90 degrees after TDC for 50k 49

Figure 5.6b
y^+ values at 90 degrees after TDC for Grid0 49

Figure 5.6c
y^+ values at 90 degrees after TDC for Grid1 49

Figure 5.6d
y^+ values at 90 degrees after TDC for Grid2 49

Figure 5.7a
y^+ values at 120 degrees after TDC for 50k 50

Figure 5.7b
y^+ values at 120 degrees after TDC for Grid0 50

Figure 5.7c
y^+ values at 120 degrees after TDC for Grid1 50

Figure 5.7d
y^+ values at 120 degrees after TDC for Grid2 50

Figure 5.8a
y^+ values at 150 degrees after TDC for 50k 51

Figure 5.8b
y^+ values at 150 degrees after TDC for Grid0 51

Figure 5.8c
y^+ values at 150 degrees after TDC for Grid1 51

Figure 5.8d
y^+ values at 150 degrees after TDC for Grid2 51

Figure 5.9a
y^+ values at 180 degrees after TDC for 50k 52

Figure 5.9b
y^+ values at 180 degrees after TDC for Grid0 52

Figure 5.9c
y^+ values at 180 degrees after TDC for Grid1 52

Figure 5.9d
y^+ values at 180 degrees after TDC for Grid2 52

Figure 5.10a
Grid at 60 degrees after TDC 53

Figure 5.10b
Relative difference in velocity between 50k and Grid2
at 60 degrees after TDC at section A-A 54

Figure 5.10c
Relative difference in velocity between 50k and Grid2
at 60 degrees after TDC at section B-B 54

Figure 5.11a
Grid at 120 degrees after TDC 54

Figure 5.11b
Relative difference in velocity between 50k and Grid2
at 120 degrees after TDC at section A-A 55

Figure 5.11c
Relative difference in velocity between 50k and Grid2
at 120 degrees after TDC at section B-B 55

Figure 5.11d
Relative difference in velocity between 50k and Grid2
at 120 degrees after TDC at section C-C 55

Figure 5.11e
Relative difference in velocity between 50k and Grid2
at 120 degrees after TDC at section D-D 55

Figure 5.11f
Relative difference in velocity between 50k and Grid2
at 120 degrees after TDC at section E-E 55

Figure 5.12a
Grid at 180 degrees after TDC 56

Figure 5.12b
Relative difference in velocity between 50k and Grid2
at 180 degrees after TDC at section A-A 56

Figure 5.12c
Relative difference in velocity between 50k and Grid2
at 180 degrees after TDC at section B-B 56

Figure 5.12d
Relative difference in velocity between 50k and Grid2
at 180 degrees after TDC at section C-C 57

Figure 5.12e
Relative difference in velocity between 50k and Grid2
at 180 degrees after TDC at section D-D 57

Figure 5.12f
Relative difference in velocity between 50k and Grid2
at 180 degrees after TDC at section E-E 57

Figure 5.12g
Relative difference in velocity between 50k and Grid2
at 180 degrees after TDC at section F-F 57

Figure 5.13a
Grid at 60 degrees after TDC 58

Figure 5.13b
Relative difference in velocity at 60 degrees after TDC
between grid0 and Grid2 at section A-A 58

Figure 5.13c
Relative difference in velocity at 60 degrees after TDC
between Grid0 and Grid2 at section B-B 58

Figure 5.14a
Grid at 120 degrees after TDC 59

Figure 5.14b
Relative difference in velocity at 120 degrees after TDC
between Grid0 and Grid2 at section A-A 59

Figure 5.14c
Relative difference in velocity at 120 degrees after TDC
between Grid0 and Grid2 at section B-B 59

Figure 5.14d
Relative difference in velocity at 120 degrees after TDC
between Grid0 and Grid2 at section C-C 60

Figure 5.14e
Relative difference in velocity at 120 degrees after TDC
between Grid0 and Grid2 at section D-D 60

Figure 5.14f
Relative difference in velocity at 120 degrees after TDC
between Grid0 and Grid2 at section E-E 60

Figure 5.15a
Grid at 180 degrees after TDC 61

Figure 5.15b
Relative difference in velocity at 180 degrees after TDC
between Grid0 and Grid2 at section A-A 61

Figure 5.15c
Relative difference in velocity at 180 degrees after TDC
between Grid0 and Grid2 at section B-B 61

Figure 5.15d
Relative difference in velocity at 180 degrees after TDC
between Grid0 and Grid2 at section C-C 62

Figure 5.15e
Relative difference in velocity at 180 degrees after TDC
between Grid0 and Grid2 at section D-D 62

Figure 5.15f
Relative difference in velocity at 180 degrees after TDC
between Grid0 and Grid2 at section E-E 62

Figure 5.15g
Relative difference in velocity at 180 degrees after TDC
between Grid0 and Grid2 at section F-F 62

Figure 5.16a
Grid at 60 degrees after TDC 63

Figure 5.16b
Relative difference in velocity at 60 degrees after TDC
between Grid1 and Grid2 at section A-A 63

Figure 5.16c
Relative difference in velocity at 60 degrees after TDC
between Grid1 and Grid2 at section B-B 63

Figure 5.17a
Grid at 120 degrees after TDC 64

Figure 5.17b
Relative difference in velocity at 120 degrees after TDC
between Grid1 and Grid2 at section A-A 64

Figure 5.17c
Relative difference in velocity at 120 degrees after TDC
between Grid1 and Grid2 at section B-B 64

Figure 5.17d
Relative difference in velocity at 120 degrees after TDC
between Grid1 and Grid2 at section C-C 65

Figure 5.17e
Relative difference in velocity at 120 degrees after TDC
between Grid1 and Grid2 at section D-D 65

Figure 5.17f
Relative difference in velocity at 120 degrees after TDC
between Grid1 and Grid2 at section E-E 65

Figure 5.18a
Grid at 180 degrees after TDC 66

Figure 5.18b
Relative difference in velocity at 180 degrees after TDC
between Grid1 and Grid2 at section A-A 66

Figure 5.18c
Relative difference in velocity at 180 degrees after TDC
between Grid1 and Grid2 at section B-B 66

Figure 5.18d
Relative difference in velocity at 180 degrees after TDC
between Grid1 and Grid2 at section C-C 67

Figure 5.18e
Relative difference in velocity at 180 degrees after TDC
between Grid1 and Grid2 at section D-D 67

Figure 5.18f
Relative difference in velocity at 180 degrees after TDC
between Grid1 and Grid2 at section E-E 67

Figure 5.18g
Relative difference in velocity at 180 degrees after TDC
between Grid1 and Grid2 at section F-F 68

Figure 5.19a
Velocity profile for 50k and Grid2 along centerline at 60 degrees
after TDC at section A-A 69

Figure 5.19b
Velocity profile for 50k and Grid2 along centerline at 60 degrees
after TDC at section B-B 69

Figure 5.20a
Velocity profile for 50k and Grid2 along centerline at 120 degrees
after TDC at section A-A 70

Figure 5.20b
Velocity profile for 50k and Grid2 along centerline at 120 degrees
after TDC at section B-B 70

Figure 5.20c
Velocity profile for 50k and Grid2 along centerline at 120 degrees
after TDC at section C-C 71

Figure 5.20d
Velocity profile for 50k and Grid2 along centerline at 120 degrees
after TDC at section D-D 71

Figure 5.21a
Velocity profile for 50k and Grid2 along centerline at 180 degrees
after TDC at section A-A 72

Figure 5.21b
Velocity profile for 50k and Grid2 along centerline at 180 degrees
after TDC at section B-B 72

Figure 5.21c
Velocity profile for 50k and Grid2 along centerline at 180 degrees
after TDC at section C-C 73

Figure 5.21d
Velocity profile for 50k and Grid2 along centerline at 180 degrees
after TDC at section D-D 73

Figure 5.21e
Velocity profile for 50k and Grid2 along centerline at 180 degrees
after TDC at section E-E 74

Figure 5.21f
Velocity profile for 50k and Grid2 along centerline at 180 degrees
after TDC at section F-F 74

Figure 5.22a
Velocity profile for Grid0, Grid1 and Grid2 along centerline at
60 degrees after TDC at section A-A 75

Figure 5.22b
Velocity profile for Grid0, Grid1 and Grid2 along centerline at
60 degrees after TDC section B-B 75

Figure 5.23a
Velocity profile for Grid0, Grid1 and Grid2 along centerline at 1
20 degrees after TDC at section A-A 76

Figure 5.23b
Velocity profile for Grid0, Grid1 and Grid2 along centerline at 1
20 degrees after TDC at section B-B 76

Figure 5.23c
Velocity profile for Grid0, Grid1 and Grid2 along centerline at 1
20 degrees after TDC at section C-C 77

Figure 5.23d
Velocity profile for Grid0, Grid1 and Grid2 along centerline at 1
20 degrees after TDC at section D-D 77

Figure 5.23e
Velocity profile for Grid0, Grid1 and Grid2 along centerline at 1
20 degrees after TDC at section E-E 78

Figure 5.24a
Velocity profile for Grid0, Grid1 and Grid2 along centerline at
180 degrees after TDC at section A-A 78

Figure 5.24b
Velocity profile for Grid0, Grid1 and Grid2 along centerline at
180 degrees after TDC at section B-B 79

Figure 5.24c
Velocity profile for Grid0, Grid1 and Grid2 along centerline at
180 degrees after TDC at section C-C 79

Figure 5.24d
Velocity profile for Grid0, Grid1 and Grid2 along centerline at
180 degrees after TDC at section D-D 80

Figure 5.24e
Velocity profile for Grid0, Grid1 and Grid2 along centerline at
180 degrees after TDC at section E-E 80

Figure 5.24f
Velocity profile for Grid0, Grid1 and Grid2 along centerline at
180 degrees after TDC at section F-F 81

Figure 5.25
Cross sections through the engine 82

Figure 5.26a
Flow field at 60 degrees after TDC, Grid2 section X-X 83

Figure 5.26b
Flow field at 60 degrees after TDC, Star-CD section X-X 83

Figure 5.26c
Flow field at 60 degrees after TDC, Grid2 section Y-Y 83

Figure 5.26d
Flow field at 60 degrees after TDC, Star-CD section Y-Y 83

Figure 5.27a
Flow field at 120 degrees after TDC, Grid2 section X-X 84

Figure 5.27b
Flow field at 120 degrees after TDC, Star-CD section X-X 84

Figure 5.27c
Flow field at 120 degrees after TDC, Grid2 section Y-Y 84

Figure 5.27d
Flow field at 120 degrees after TDC, Star-CD section Y-Y 84

Figure 5.28a
Flow field at 180 degrees after TDC, Grid2 section X-X 85

Figure 5.28b
Flow field at 180 degrees after TDC, Star-CD section X-X 85

Figure 5.28c
Flow field at 180 degrees after TDC, Grid2 section Y-Y 85

Figure 5.28d
Flow field at 180 degrees after TDC, Star-CD section Y-Y 85

Chapter 1

Introduction

1.1 Overview

Computational Fluid Dynamics (CFD) has found widespread use in industry and academia for the analysis of complex fluid flow and heat transfer problems. CFD greatly reduces the need for costly and time-consuming physical experiments and allows the engineer to investigate many different solutions to a particular problem in a shorter period of time. The automotive industry is an area where the use of CFD to solve engineering problems has increased rapidly in the past several years.

While there are many areas in the automotive industry where CFD can be effectively applied, the analysis of Internal Combustion (IC) engine in-cylinder flows has been an area of particular interest. Understanding flow phenomena in IC engines is vital to improving their efficiency and performance. The use of CFD for IC engine in-cylinder flow analysis is still in a state of development and shows tremendous promise for the future.

While there are several CFD codes that can be used to analyze flow phenomena in IC engines, one of the more popular codes for this purpose is KIVA-3V. The KIVA-3V code is designed to analyze transient, 2-D or 3-D, chemically reactive flows with sprays in the IC engine combustion chamber. Some salient features of the KIVA-3V code are as follows [1]:

- Has deforming mesh capability to simulate the motion of engine valves and pistons.

- Has several turbulence models: standard k-ε with wall functions, sub-grid scale (SGS) model and the RNG k-e model.
- Accounts for chemistry through the Arrhenius model, equilibrium model, mixing controlled turbulent combustion model and a soot model.
- Spray physics modeled include droplet breakup, coalescence and particle-based wall-film.
- Contains a library of 37 types of fuels.
- Has an open source code, allowing the user to modify the computer program or add new functionalities and physical models.

KIVA-3V and the codes preceding it were all developed at Los Alamos National Laboratory by Dr. A. A. Amsden and his colleagues.

1.2 Limitations of KIVA-3V

While KIVA-3V contains many features that make it useful for the simulation of flows in simple IC engines, it possesses certain limitations that prevent it from being readily applied to the analysis of modern IC engines. Some of these limitations are described below:

- It is very difficult, or even impossible, to generate the complex geometries found in modern IC engines using the KIVA-3V pre-processor. The KIVA-3V pre-processor can only generate pre-determined shapes hard coded into the computer program. The user can change the dimensions of these shapes, but their basic topology cannot be altered, limiting the types of IC engine geometries that can be modeled.

- Mesh control in the KIVA-3V pre-processor is poor. The distribution of interior vertices in the mesh is largely determined by linear interpolation schemes in the code. These schemes do not always provide the best possible grid distribution. The user cannot manipulate individual or groups of vertices to improve or alter the mesh distribution and quality.
- The KIVA-3V solver accepts only multi-block, continuous H-H grids. When an H-H grid is used to generate a computational mesh for the cylindrical geometries typically found in IC engines, the vertices at the "corners" of the mesh are invariably located very close to the mesh boundary or wall. When using the standard k-ε turbulence model with wall functions, the value of y^+ in these regions falls below the acceptable limit. This is a potential source of error in the computed solution.

1.3 Objective

The objective of this work is to address the inherent limitations of KIVA-3V and attempt to provide a means to overcome them. The work presented in this thesis is as follows:

- Provide a brief overview of the traditional KIVA-3V mesh generation method.
- Document and describe a method, developed at the Michigan State University Engine Research Lab (MSU-ERL), in which the KIVA-3V mesh can be generated using Gridgen, a popular GUI based pre-processor software with many advanced mesh generation tools.

- The method developed at the MSU-ERL cannot be used to generate canted valve engine geometries. Describe and document modifications to the MSU-ERL method that allow the generation of canted valve engine geometries.
- Describe the generation of a canted valve engine mesh using the modified MSU-ERL method. The procedure described can be used as a guide for generating IC engine meshes using the MSU-ERL method.
- Investigate the effect of the H-H grid requirement on y^+ and subsequently on the computed solution. Suggest guidelines to reduce the impact of the H-H grid requirement on the computed solution.

Chapter 2

The Traditional KIVA-3V Mesh Generation Method

2.1 Overview

KIVA-3V has a three-part structure: pre-processor, solver and post-processor. Each component of KIVA-3V is an independent, stand-alone computer program. KIVA-3V's pre-processor is called K3VPREP. K3VPREP is specifically designed for the generation of IC engine geometries, though it can also be used to generate other types of geometries. Using K3VPREP to generate simple IC engine geometries is relatively easy, but it has certain limitations that inhibit the creation of the complex geometries found in modern IC engines. This chapter describes the traditional K3VPREP mesh generation procedure and highlights its limitations. Note that this chapter provides a basic outline of the traditional mesh generation procedure. More detail can be obtained by referring to the KIVA-3V user's manual [1].

2.2 The Traditional KIVA-3V Mesh Generation Method

K3VPREP is a template based pre-processor. A template comprised of rectangular blocks is constructed first, with each block or combination of blocks representing a particular geometric feature or region in the IC engine. The blocks are meshed using a continuous H-H structured grid consisting of only hexahedral elements. Each block is then shaped using built-in parametric block-shaping tools to produce the IC engine geometry and the final computational mesh. The traditional mesh generation procedure contains four main steps, as illustrated in Figure 2.1. Each step is described in the paragraphs that follow.

$$\begin{tikzcd}
\textbf{Template Diagram} \arrow[d] \\
\textbf{IPREP} \arrow[d] \\
\textbf{K3VPREP} \arrow[d] \\
\textbf{TAPE17}
\end{tikzcd}$$

Figure 2.1: Schematic of the traditional mesh generation procedure

<u>Template Diagram</u>

The first step in creating a computational mesh using K3VPREP is constructing the Template Diagram. The Template Diagram is a sketch of the block-structured template that represents all geometric features of the IC engine with rectangular blocks. Figure 2.2 shows a canted valve IC engine with intake port only, along with the corresponding Template Diagram. Note that the internal blocks representing the valves are not shown in Figure 2.2.

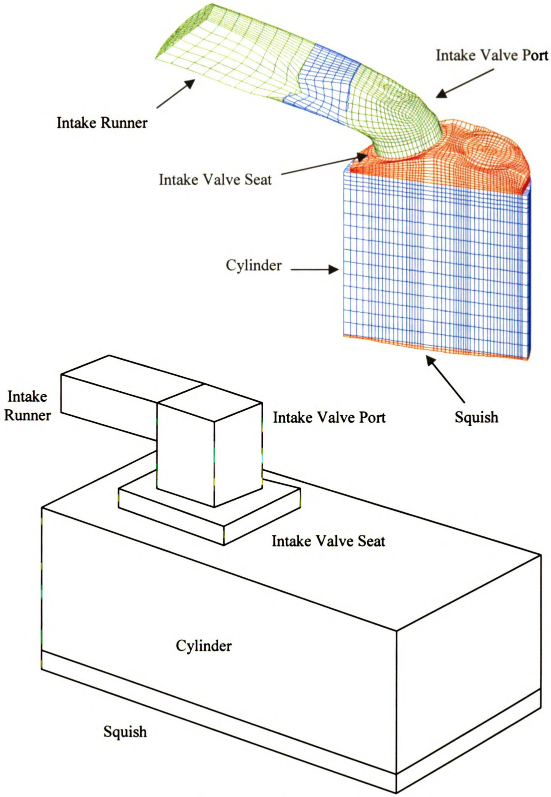


Figure 2.2: IC engine geometry and its corresponding Template Diagram

The Template diagram is used as a reference when creating the IPREP input file.

<u>IPREP file</u>

K3VPREP is not a GUI based code. All data describing the IC engine geometry and the mesh must be provided in the form of an alphanumeric input file called the IPREP file. This is the second step in the traditional mesh generation process. The IPREP file contains the following information:

- Dimensions, location and mesh density (number of vertices in the x, y and z direction) of each block in the template.
- Block connectivity information.
- Boundary face information, i.e. wall, inlet, outlet, symmetry plane etc.
- Geometric data for the block-shaping tools in K3VPREP.

The block-shaping tools in K3VPREP read the geometric data provided in the IPREP file and transform the block-structured mesh into the correct IC engine geometry. The block-shaping tools in K3VPREP are parametric in nature and can be used to create only a certain number of pre-determined shapes. The block-shaping tools and the shapes they generate are listed below. More detail on the block-shaping tools can be found in the KIVA-3V user's manual [1].

- NBO
    - Creates an axisymmetric or asymmetric cup or dome in the piston face or cylinder head.
    - Creates a curved squish region on the piston face and the cylinder head.
- NLOCXY

    - Used for localized geometry refinement of blocks above the cylinder head, such as combustion chambers. Specifies the outline of the blocks by defining the x-y coordinates of the vertices that are part of the block edges.
- NVALVEPORT
    - Creates simple "bent tube" valve port geometries.
- NRUNNER
    - Rounds the edges of the blocks intake/exhaust runner blocks.
- NSIAMESE
    - Smoothly joins pairs of intake/exhaust runners.
- NDISH
    - Creates a shallow dish in the piston face.
- NSCALLOP
    - Creates scallops in the piston face.
- NPROVTOP, NPROVFCE
    - Create the valve profiles.
- NTILT
    - Tilts regions of the mesh to create canted valve geometries.

No other shapes, besides those described above, can be generated using K3VPREP.

<u>K3VPREP</u>

K3VPREP reads the data in the IPREP file and executes a sequence of operations to generate the computational mesh. The main steps in this sequence are described below:

1. Read the IPREP file.
2. Create all blocks in the template.

3. Mesh each block.
4. Merge internal blocks with the external blocks they are contained in.
5. Shape the engine cylinder.
6. Shape the valve seats.
7. Shape the valve ports.
8. Shape the runners.
9. Connect remaining blocks to form a single, continuos mesh.
10. Shape the intake and exhaust valves
11. Tilt the intake and exhaust regions if it is a canted valve engine.
12. Create a dish in piston face if needed.
13. Create scallops in piston face if needed.
14. Write final TAPE17 mesh file.

<u>TAPE17 File</u>

The final mesh file that is the output from K3VPREP is called the TAPE17 file. The TAPE17 file contains the following general information:

- X, Y, Z coordinates of each vertex.
- Vertex type, i.e. fluid or solid.
- Cell face boundary type, i.e. wall, inlet, outlet, moving surface etc.
- Connectivity information.

The X, Y, Z coordinates of the vertices completely describe the IC engine geometry and the vertex distribution in the mesh. The rest of the data in the TAPE17 file, such as vertex type and cell face information, is used by the KIVA-3V to run the CFD simulation.

The creation of the TAPE17 file is the last step of the traditional mesh generation method.

2.3 Limitations of the Traditional Mesh Generation Method

Most modern CFD pre-processors can import the IC engine geometry directly from CAD software in IGES, STL or other formats. The computational mesh is generated directly from the CAD data, ensuring that all geometric features of the engine are captured in the mesh. K3VPREP does not have this facility. The user must create the engine geometry from scratch using the geometry creation commands available in K3VPREP. The geometry creation commands in K3VPREP can generate only a pre-determined set of shapes, as described earlier in this chapter. These shapes are parametric in nature. The user can only modify the parameters that alter the dimensions of these shapes. Shapes not defined by these parameters cannot be created by using K3VPREP. Thus, more complex engine geometries cannot be created, thereby limiting the usefulness of the KIVA-3V code.

Even for geometries that can be created using K3VPREP, the user has very little control over the distribution of grid points in the mesh. The distribution of grid points is largely controlled by linear interpolation schemes in the K3VPREP code. The user cannot control the location of individual or groups of grid points using the tools available in the IPREP file. While the mesh distribution schemes in K3VPREP might be sufficient for simple engine geometries, a mesh generated using K3VPREP often contains areas of poor mesh quality. This includes cells with very high skewness and aspect ratio. In these circumstances it is desirable for the user to be able to fine-tune the mesh by manipulating individual or groups of vertices, but this is not possible in K3VPREP.

Finally, the lack of a GUI makes K3VPREP hard to use. The user must provide vast amounts of numerical data in form of the IPREP file in order to generate even a simple engine geometry. This makes the IPREP file very complicated and tedious to create, with plenty of opportunity for error. Furthermore, the user must wait until the entire K3VPREP program is executed to see the results of the mesh generation. The lack of real-time visualization of the mesh generation process is an added hindrance to using K3VPREP.

Chapter 3

An Improved KIVA-3V Mesh Generation Method

3.1 Overview

As described in the previous chapter, generating a computational mesh for complex IC engine geometries is very difficult using K3VPREP. An improved mesh generation method has been developed at the Michigan State University Engine Research Lab (MSU-ERL) to overcome the limitations of K3VPREP. The only limitation of the MSU-ERL method is that it cannot be used to generate canted valve engine geometries. Modifications were made to the MSU-ERL method to overcome this limitation. This chapter first describes the MSU-ERL method and then describes the modifications made to allow for the generation of canted valve IC engine geometries.

3.2 The MSU-ERL Method

The primary drawback to K3VPREP is its limited geometry creation and meshing tools. The popular commercial pre-processors available in the market today are all GUI based and have powerful geometry creation tools or can import geometries from CAD packages such as Pro-Engineer or Unigraphics. The main motivation behind the MSU-ERL method is to allow the IC engine mesh to be created using one of these commercially available pre-processors. The pre-processor chosen for MSU-ERL method was Gridgen, a software especially well suited for generating the type of structured grids that KIVA-3V requires.

The KIVA-3V solver requires the computational mesh to be in TAPE17 format. In addition to vertex coordinates and connectivity, the TAPE17 file provides vertex and cell information that is required by the KIVA-3V solver in order to perform the CFD simulation. If this vertex and cell information is missing, the KIVA-3V solver program will not be able to perform the CFD simulation. Gridgen provides only the vertex coordinates and connectivity information. For this reason, it is not possible to use a mesh from Gridgen directly in the KIVA-3V solver. The MSU-ERL method overcomes this limitation by replacing the vertex coordinates of the mesh created in K3VPREP with the vertex coordinates of the mesh created in Gridgen. Since the geometry and mesh distribution in TAPE17 file is completely described by the vertex coordinates, replacing them with the vertex coordinates from the Gridgen mesh transfers the IC engine mesh from Gridgen to the TAPE17 file. All other vertex and cell information required by the solver is created by K3VPREP based on the information in the IPREP file. The MSU-ERL method contains 4 additional steps, as illustrated in Figure 3.1

$$\begin{tikzcd}
\text{Template Diagram} \arrow[d] \\
\text{Gridgen mesh} \arrow[d] \\
\text{IPREP file} \arrow[d] \\
{}
\end{tikzcd}$$

$$\begin{tikzcd}
\text{K3V1PREP} \arrow[d] \\
\text{GRIDV} \arrow[d] \\
\text{K3V2PREP} \arrow[d] \\
\text{TAPE17}
\end{tikzcd}$$

Figure 3.1: Schematic of the MSU-ERL method

Template Diagram

The Template Diagram is created in exactly the same way as the traditional mesh generation method. There are no changes whatsoever.

Gridgen

The second step in the MSU-ERL method is to create the desired mesh in Gridgen. The Gridgen mesh is created based on IC engine geometry imported from a CAD software. The Gridgen mesh does not contain the valve geometry, as this is done in the K3V2PREP code. More detail on this step is provided in the next chapter.

IPREP file

The IPREP file is generated based on the Template Diagram and the Gridgen mesh. Each block in the template is defined as before. Care is taken to assign a mesh density to

each block that matches exactly with the corresponding geometric region in the Gridgen mesh. This is done to ensure that the mesh generated in K3VPREP has the same number of vertices as the Gridgen mesh, thereby allowing its vertex coordinates to be replaced correctly. The IPREP file in the MSU-ERL method does not contain any block shaping commands. This is because the Gridgen mesh that will replace the K3VPREP mesh already has the correct geometry. The only block shaping information in the IPPEP file is the valve shaping data. The valve shaping data is used by K3VPREP to create the valve shapes in the Gridgen mesh. More detail on creating the IPREP file is provided in the next chapter.

<u>K3V1PREP</u>

The K3V1PREP code is a modified version of K3VPREP that outputs a mesh file just before the valve shaping algorithm is invoked. This mesh file, known as the TAPE16 file, is in the same format as the TAPE17 file. The only difference is that the TAPE17 file is the final mesh file with valves while the TAPE16 file does not contain the valve shapes. The sequence of events that occur in the K3V1PREP code are as follows:

1. Read the IPREP file.
2. Create all blocks in the template.
3. Mesh each block.
4. Merge internal blocks with the external blocks they are contained in.
5. Shape the engine cylinder.
6. Shape the valve seats.
7. Shape the valve ports.
8. Shape the runners.

9. Connect remaining blocks to form a single, continuos mesh.
10. Write the TAPE16 file.

GRIDV

The GRIDV code takes the TAPE16 file and replaces its vertex coordinates with the vertex coordinates of the Gridgen mesh. The output of the GRIDV code is used in the K3V2PREP code and is also known as TAPE16.

K3V2RPEP

K3V2PREP is also a modified version of the K3VPREP code. K3V2PREP takes the TAPE16 mesh file from GRIDV, creates the valve shapes, and writes out the final TAPE17 mesh file for use in the KIVA-3V solver. The sequence of events that take place in the K3V2PREP are as follows:

1. Read the IPREP file.
2. Read in TAPE16 file from GRIDV.
3. Shape the intake and exhaust valves
4. Create a dish in piston face if needed.
5. Create scallops in piston face if needed.
6. Write final TAPE17 mesh file.

The creation of the TAPE17 file is the final step of the MSU-ERL method. This is the mesh file that is used in the KIVA-3V solver.

3.3 Limitations of the MSU-ERL Mesh Generation Method

In the MSU-ERL method, the Gridgen mesh is imported into K3V2PREP before the valve shaping subroutines are invoked. The valve shaping algorithms require the that the valve regions be vertical, i.e. the valve axes be parallel to the axis of the engine cylinder.

If the valve regions are not vertical, the logic in the valve shaping algorithms fails and the valve shapes cannot be generated. Consider the canted valve engine geometry generated in Gridgen shown below.

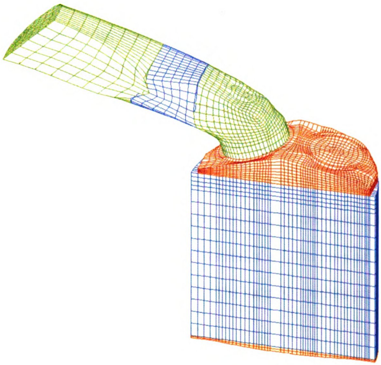


Figure 3.2: canted valve engine geometry

It can be seen that the intake valve region is already tilted in the mesh, i.e. the valve regions are not vertical. If this mesh is imported into K3V2PREP, the valve shaping subroutines will be unable to create the proper valve shapes since the valve regions are already tilted.

3.4 Modifications to the MSU-ERL Method to Allow for Canted Valve Geometries

In order to solve this problem, a few modifications were made to the K3V2PREP code. The modified K3V2PREP code is called K3V2PREP_PENT. These modifications perform the following additional tasks during the mesh generation process.

- Immediately before the valve shaping subroutines are invoked, the grid points belonging to the intake and exhaust regions are rotated in the x-z plane until they are "vertical", i.e. the valve axis is parallel to the cylinder axis.
- The grid points belonging to the intake and exhaust regions are identified by the value of their IDREG parameter in the mesh file. A value of IDREG greater than unity indicates that that particular grid point is in the intake or exhaust region.
- Once the intake and exhaust regions are vertical, the valve shaping subroutines are invoked and the valve shapes created. The valve shape data provided by the user in the IPREP file remains unchanged.
- After the valve shapes have been created, the grid points in the intake and exhaust regions are rotated back to their initial orientation, resulting in the proper canted valve geometry with valves.

The angle through which the vertices are rotated is equal to the angle of the canted regions in the engine. The following figures illustrate the rotation of the intake and exhaust regions, the generation of the valve mesh and the final canted valve engine mesh generated by the K3V2PREP_PENT code.

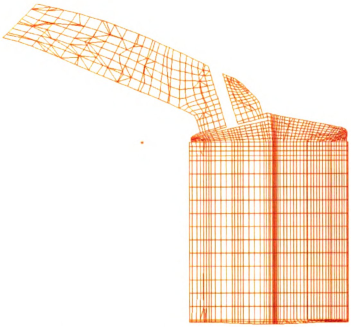


Figure 3.3: Cross section of Gridgen mesh when it is imported into K3V2PREP_PENT

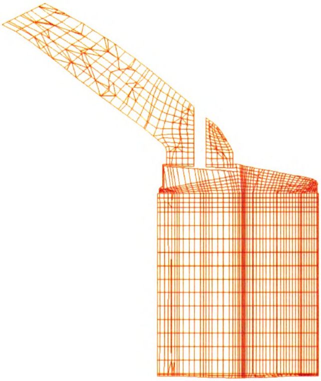


Figure 3.4: Intake region rotated to a vertical position

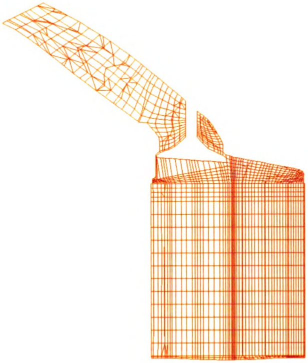


Figure 3.5: Valve top surface shape generated by NPROVTOP parameter in IPREP file

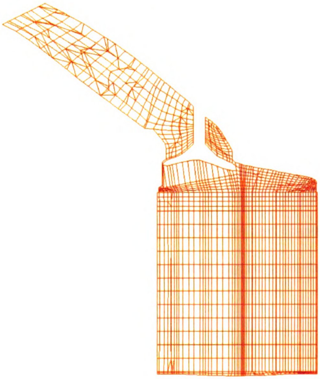


Figure 3.6: Valve bottom surface shape generated by the NPROVFCE parameter in the IPREP file

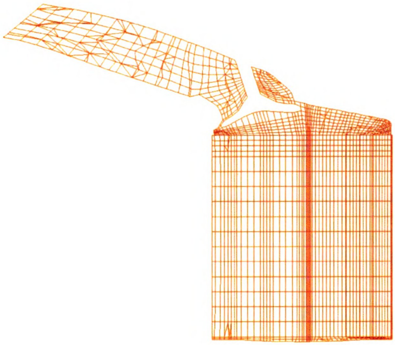


Figure 3.7: Cross section of final mesh, with intake region rotated back to its original angle

The only modifications to the MSU-ERL method are in the K3V2PREP code, and are transparent to the user. The steps for creating the Template and the parameters invoked in the IPREP file remain the same for vertical valve and canted valve engines. The only difference lies in the use of the K3V2PREP or K3V2PREP_PENT program. K3V2PREP is used for vertical valve engines and the K3V2PREP_PENT for canted valve engines. The modified MSU-ERL method is described in detail in the next chapter where the step-by-step procedure for the generation of the canted valve engine geometry shown in Fig. 3.5 is described.

Chapter 4

Procedure to Generate a Canted Valve IC Engine Mesh

4.1 Overview

This chapter describes the step-by-step procedure required to generate a canted valve engine geometry using the modified MSU-ERL method described in Chapter 3. Any canted valve engine geometry can be generated using the process described in this chapter. The steps described to generate the Template Diagram and the IPREP file are the same for any IC engine geometry.

The basic process to create the canted valve engine geometry using the modified MSU-ERL method is described below.

1. Create the Template Diagram.
2. Create the Gridgen mesh based on the Template Diagram.
3. Create the IPREP file based on the Gridgen mesh and the Template Diagram.
4. Run the K3V1PREP code with the IPREP file as input.
5. The output of K3V1PREP is the TAPE16 file.
6. Run GRIDV with the TAPE16 file and the Gridgen mesh as input.
7. The output of GRIDV is the FORT.20 file.
8. Rename the FORT.20 file as TAPE16.
9. Run K3V2PREP_PENT with the IPREP file and the new TAPE16 file as input.
10. The output of K3V2PREP_PENT is the TAPE17 file, which is the final mesh file.

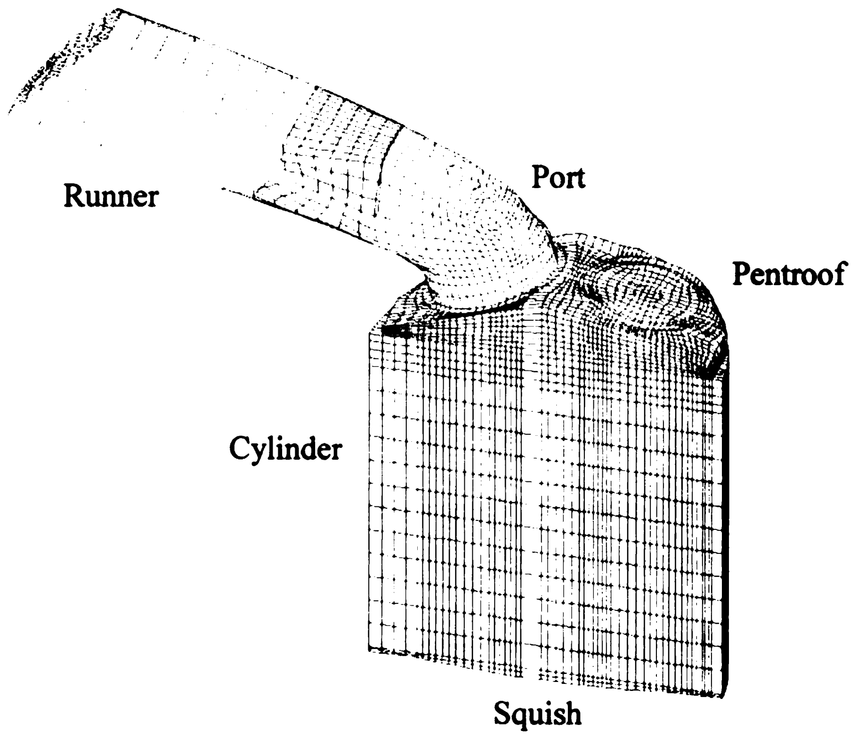
The following sections explain each step of the process in detail and provide guidelines to facilitate the easy generation of the mesh.

4.2 The Template Diagram

The Template Diagram is a schematic illustration that represents the IC engine geometry using rectangular blocks. It is used as the basis for generating the IPREP file as well as a reference for generating the Gridgen mesh. The following steps explain how to create the Template Diagram. These steps are the same for any IC engine geometry.

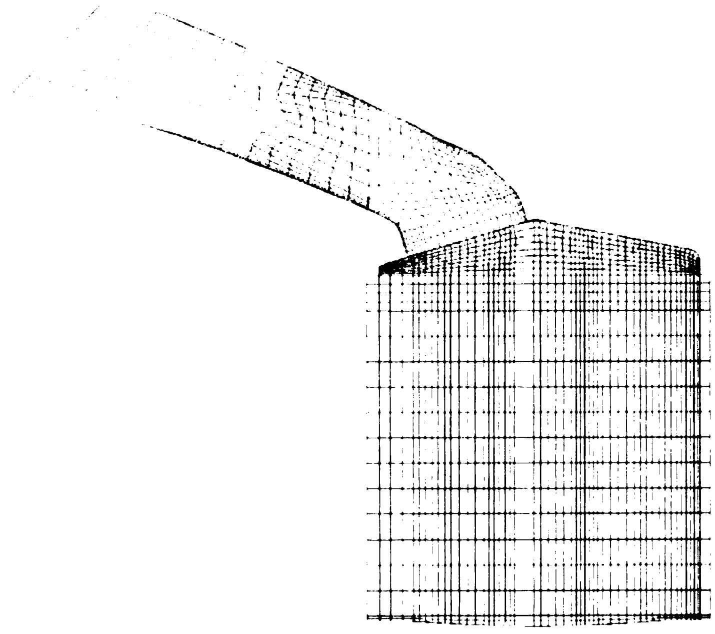
Step 1

Create a diagram representing the engine geometry using rectangular blocks. Figure 4.1 shows the canted valve engine geometry and Figure 4.2 shows the corresponding Template diagram.



(a)

Figure 4.1: a) 3-D view of the canted valve engine


(b)

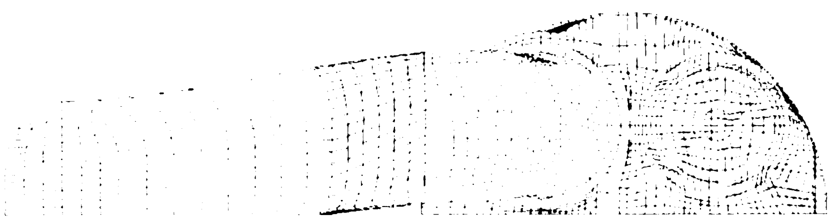
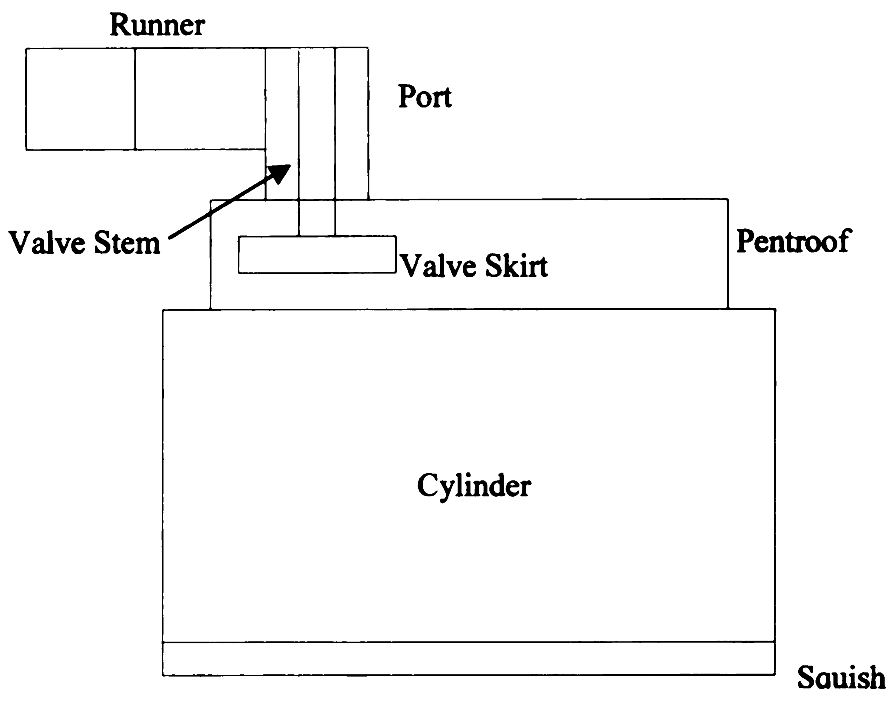
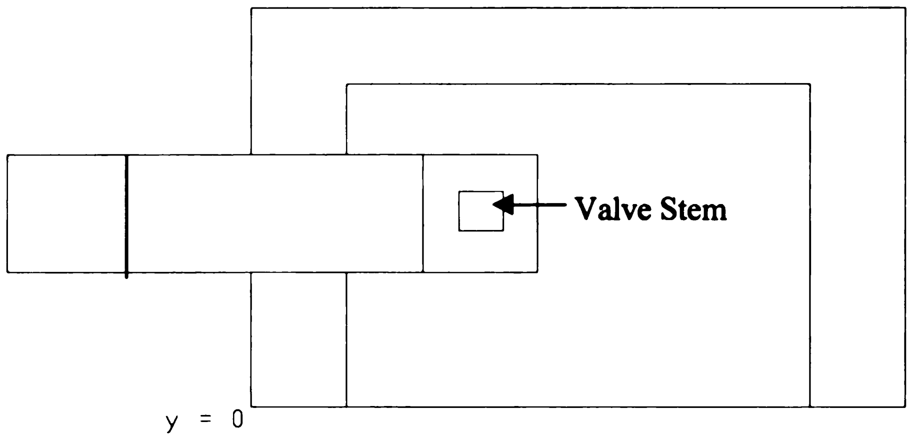

(c)

Figure 4.1: b) Front view of the canted valve engine, c) Top view of the canted valve engine

Note that for this case, only the intake geometry is modeled. The exhaust valve, port and runner are excluded. Figure 4.1 shows only the exterior of the geometry and though they are present, does not show the valves inside the engine. This engine has a symmetry plane at y = 0, therefore only half the engine is modeled.



(a)



(b)

Figure 4.2: a) Front view of the Template Diagram, b) Top view of the Template Diagram

For canted valve geometries, the valve is always generated at its fully closed position. KIVA-3V requires that even fully closed valves have one layer of cells separating the valve top surface and the valve seat. For vertical valve IC engines, the valve can be generated at the fully closed or fully open position.

For a half engine, the symmetry plan is always at y = 0. Notice that part of the runner has a boundary face at y = 0. This is where the intake runners join together in the full engine geometry. The axis of the cylinder is co-linear with the z-axis, with the origin at the intersection of the cylinder axis and the piston face. Since this geometry has a shallow dish or squish region in the piston face, the origin is at the intersection of the axis and the plane at which the piston face would be in the absence of a squish region. The Template diagram shows the interior blocks representing the valve and valve stem.

Step 2

Extend the lines of the interior blocks until they intersect with the boundary lines.

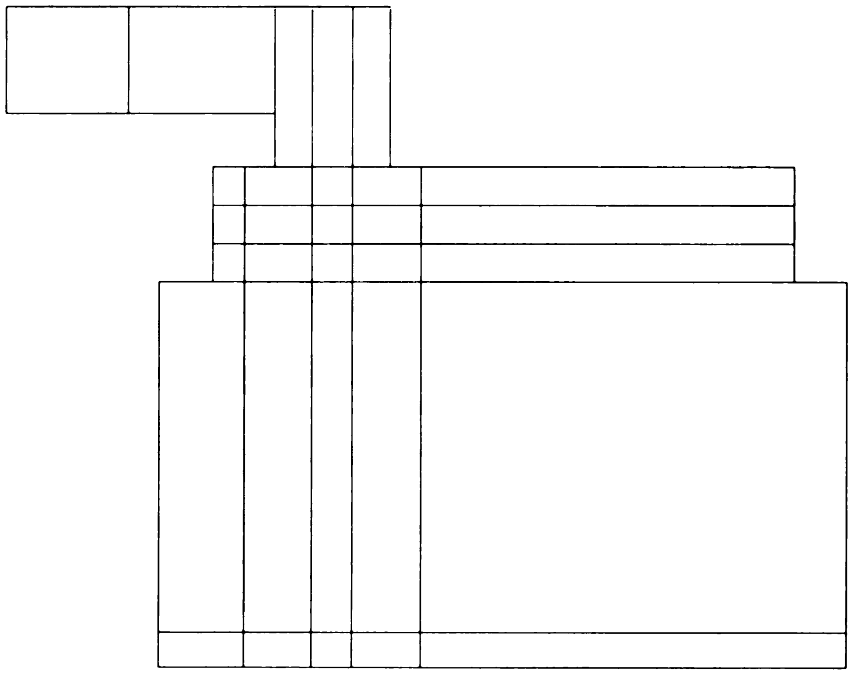


Figure 4.3: Template with interior block lines extended

Step 3

Extend the lines of blocks above the cylinder and pentroof blocks all the way through the bottom of the cylinder block.

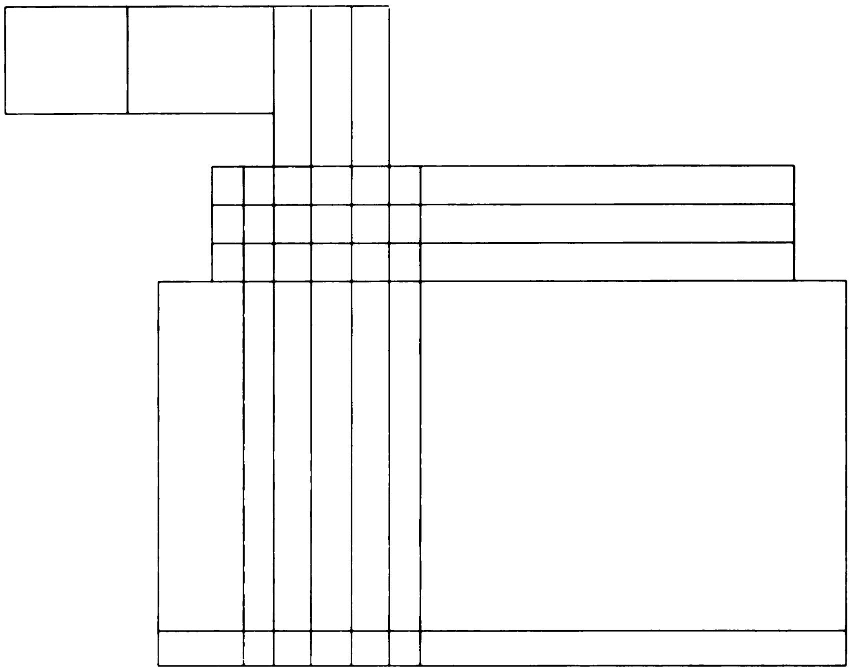
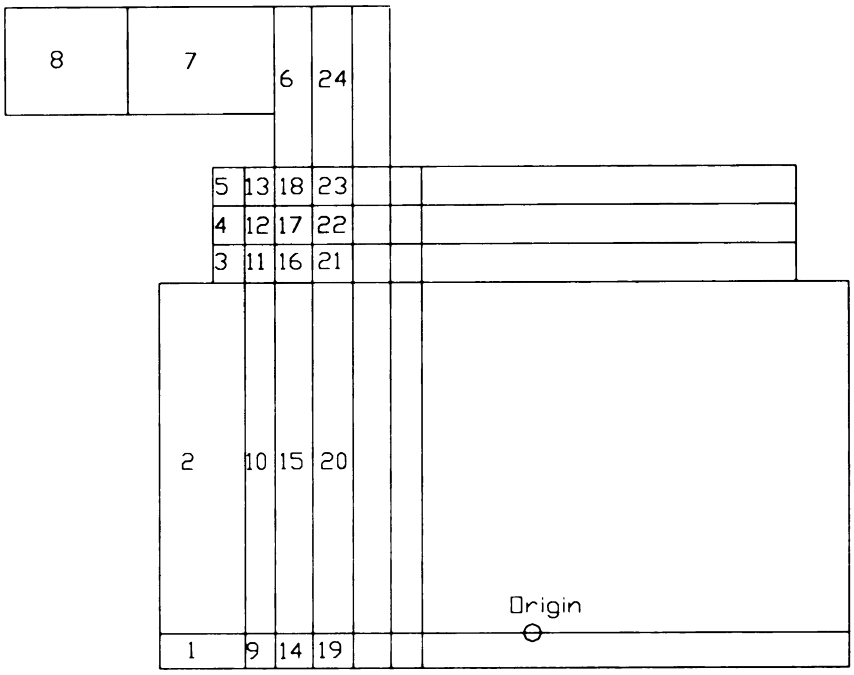


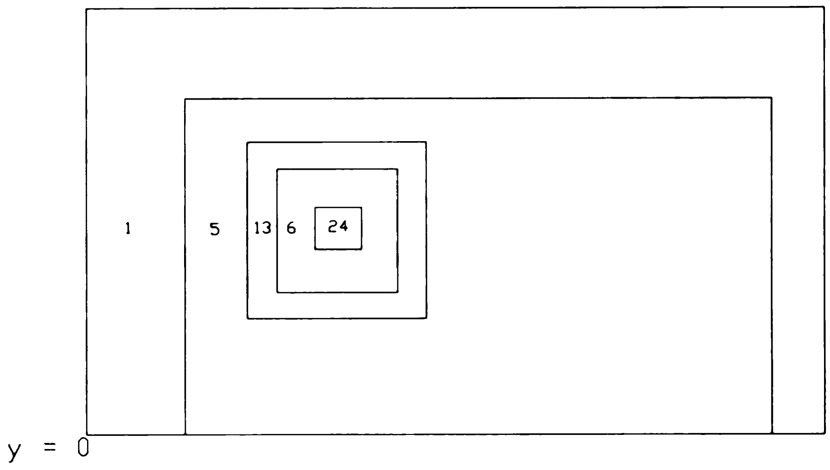
Figure 4.4: Template with the lines of blocks above the cylinder extended

Step 4

Number the blocks, starting with the external blocks and then the internal blocks. External blocks are defined as those blocks that are not contained within any other block. The numbering order is not critical but it is strongly recommended that the user follow the numbering sequence suggestion in Figure 4.5 in order to make the generation of the IPREP file easier. The Template is now complete. Dimensions of the blocks, based on the engine dimensions can be marked on the Template diagram to facilitate the generation for the IPREP file.



(a)



(b)

Figure 4.5: Final Template diagram with blocks numbered, a) front view, b) top view without runner blocks

4.3 The Gridgen Mesh

The following steps detail the guidelines to be followed when generating the Gridgen mesh that will be used in the MSU-ERL method. It is assumed that the user already knows how to operate the Gridgen software.

Step 1

Generate a mesh of the engine from imported CAD surface geometry. The engine mesh must be split into sections corresponding to the external blocks, numbered 1-8 in Fig. 4.5, as depicted in Fig. 4.6. The Gridgen mesh must comply with the following rules:

- Only hexahedral cells may be used.
- The mesh must be continuous.
- If a valve seat is present, it must be at least 2 cells thick. There is no valve seat in this engine geometry, but this rule must be adhered to whenever a valve seat is present.
- There must be at least 2 cells between the edge of the valve seat and the edge of the port where it meets the valve seat. Once again, this geometry does not have a valve set, but this rule must be adhered to whenever a valve seat is present.
- The grid lines in the valve travel region must be perpendicular to the direction of travel of the valve.
- The cells in the valve travel region must have as little skew or warp as possible.
- The Gridgen mesh must not contain the valve geometry, but must have grid lines representing the valve axis, valve skirt outline, etc.
- The mesh must be in Plot-3D format.

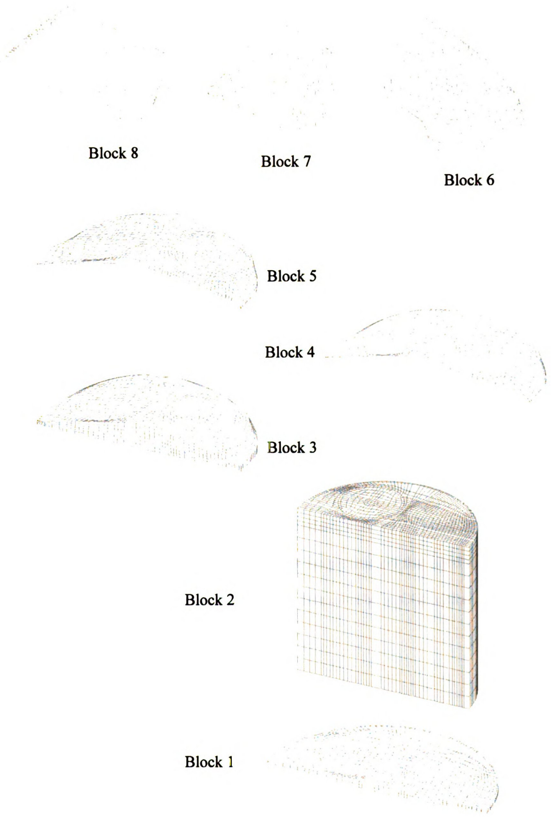


Figure 4.6: Gridgen mesh split into sections corresponding to the external blocks in the Template

Step 2

Once the Gridgen mesh is complete, each section depicted in Figure 3.6 must be converted to UNV format. The program CONV_UNV, shown in Appendix B, can be used for this purpose.

Step 3

After each Gridgen mesh section is converted to UNV format, they must be combined into a single file called gridgen.unv. The program COMBINEGRID, shown in Appendix C, can be used for this purpose. Now the Gridgen mesh is ready to be used in the MSU-ERL method.

4.4 Creating the IPREP File

The IPREP file contains several parameters that describe the blocks in the Template diagram. The information contained in the IPREP file includes the block size and spatial location, boundary type for the block faces, valve shape data etc. In the existing K3VPREP mesh generation method, the IPREP file is used to describe the complete geometry of the engine. But in the MSU-ERL method, none of the block-shaping parameters are invoked. The following steps list the parameters that need to be listed in the IPREP file and the data it must contain. All data must be in CGS units.

NAME

- Format 10A8.
- Description of engine or any other description..

BORE

- Format A8, F10.5
- Diameter of the engine cylinder.

STROKE

- Format A8, F10.5
- Piston travel distance from BDC to TDC.

SQUISH

- Format A8, F10.5
- Clearance between cylinder head and piston crown at TDC.

THSECT

- Format A8, F10.5
- Describes section of engine being generated.
- Full engine has THSECT=360
- Half engine with symmetry plane at y=0 has THSECT=180
- Any section of the engine less than half has THSECT equal to some value that evenly divides into 360.

NBLOCKS

- Format A8, I5
- Total number of blocks in the Template Diagram (Figure 4.5)

The following lines are repeated NBLOCK times

N, NX, NY, NZ, NBO, NTYPE, NREGK3, IGHOST

- Format 8I4
- N is the block number
- NX, NY, NZ are the number of cells in the x, y and z direction.
- NBO is a block-shaping command and is not used.

- NTYPE specifies the block type. This is based on the location of the block in the mesh and to which part of the engine geometry it belongs.
    - Piston cup=1
    - Squish region=2
    - Head dome=3
    - All others=4
    - Valve ports and valve stems in the ports=5
- NREGK3 identifies the region to which the block belongs.
    - Cylinder or squish region=1
    - Intake region=2
    - Exhaust region=3
    - If more than one intake or exhaust region existed, the intake regions would all be numbered first, followed by the exhaust regions.
- IGHOST specifies if the block is a fluid or solid block.
    - All valve and valve stem blocks are solid.
    - IGHOST=0 for fluid block.
    - IGHOST=1 for solid block

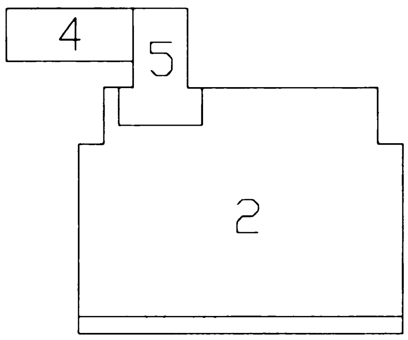


Figure 4.7: NTYPE

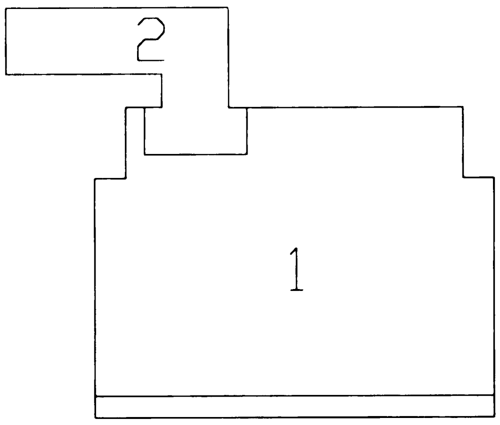


Figure 4.8: NREGK3

<u>XC1, XC2, XC3, XC4, XC5, XC6, XC7, XC8</u>

- Format 8F8.3
- X coordinates of the corners of the block. The block corners in KIVA-3V are numbered as shown in Figure 14. The front of the block is the face 1-2-5-8.

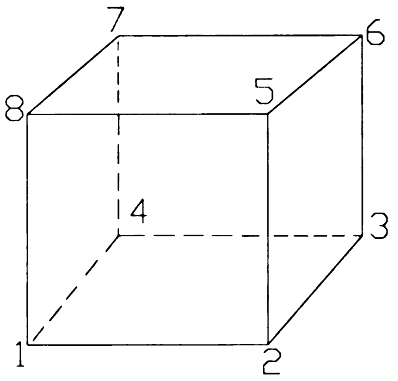


Figure 4.9: Block corner numbering

- XC1 refers to corner 1, XC2 to corner 2 and so on.

- All blocks that share an axis with the valves are assumed to initially be centered on the origin and the coordinates defined accordingly. These blocks will be translated into the proper position later in the IPREP file.

YC1, YC2, YC3, YC4, YC5, YC6, YC7, YC8

- Format 8F8.3
- Y coordinates of corners of the block.
- Same numbering system as above.

ZC1, ZC2, ZC3, ZC4, ZC5, ZC6, ZC7, ZC8

- Format 8F8.3
- Z coordinates of corners of the block.
- Same numbering system as above.

FACEL, FACER, FACEF, FACED, FACEB, FACET

- Format 6F8.3
- Assigns boundary conditions to the left, right, front, derriere (back), bottom and top faces of the block respectively.
- Moving face=1.0
- Solid=2.0
- Axis=3.0
- Fluid=4.0
- Front periodic=5.0 (FACEF only)
- Derriere periodic=6.0 (FACED only)
- Specified inflow=7.0
- Continuative outflow=8.0

- Pressure inflow=9.0
- Pressure outflow=10.0

<u>FACEIDL, FACEIDR, FACEIDF, FACEIDD, FACEIDB, FACEIDT</u>

- Format 6F8.3
- Indicates whether the block face is moving and if it is, which moving face it belongs to.
- The piston face is always assigned a value of 1.0
- The top surfaces of the valves are odd and the bottom surfaces of valves are even.
- The surfaces of the intake valves(s) are numbered first followed by the surfaces of the exhaust valve(s).

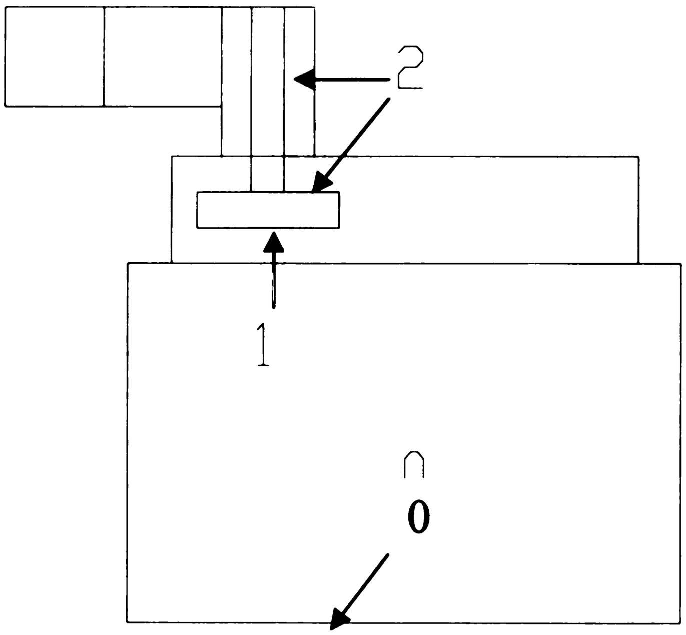


Figure 4.10: Moving surfaces

TRANSLATE

- Format A10, I5
- Number of blocks to be translated.
- All blocks that share a common axis with a valve are translated.

The following line is repeated TRANSLATE number of times.

NBLK1, TRANSX, TRANSY, TRANSZ

- Format I4, 1X, 3F10.3
- NBLK1 is the block number.
- TRANSX, TRANSY, TRANSZ are the x, y, z distance the block is to be translated from the origin.
- In this case, the

RESHAPE

- Format A10, I5
- Number of blocks to be reshaped.
- Reshape must be performed on each block that is contained within another block.

NBLK1, NBLK2, INDEX1, INDEX2, INTRP, IRELAX

- Format 6I4
- NBLK1 is the inner block.
- NBLK2 is the outer block.
- INDEX1, INDEX 2 are the indices of NBLK2 where reshaping is to start.
- INTRP=1 to interpolate all non-assigned points between the faces of the blocks.
- IRELAX=1 to relax the points interpolated when INTRP=1

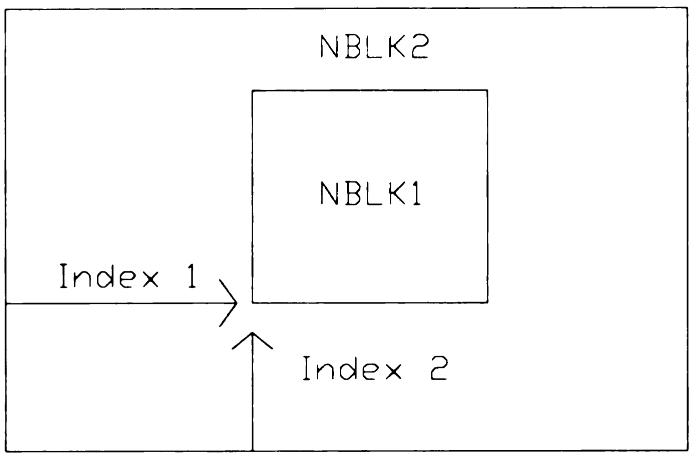


Figure 4.11: Reshape indices

<u>NPATCH</u>

- Format A8, I5
- Number of external blocks remaining after reshaping (see Figure 8).

<u>NBLK1, NFACE, NBLK2, INDEX1, INDEX2, ICOOR</u>

- Format 6I4
- This says patch NFACE of block NBLK1 onto the opposite face of block NBLK2 using the coordinates from block ICOOR.
- NFACE has the following values
    - Left face=1
    - Right face=2
    - Front face=3
    - Derriere face=4
    - Bottom face=5
    - Top face=6

- Refer to the KIVA-3V manual [1] for more detailed information on patching.

<u>NPROVTOP</u>

- Format A10, I5
- Number of valve top surfaces to be profiled

<u>IDSURF, NZSEAT, NZEDGE, NZPORT, NPTSPRO</u>

- Format 3F8.3
- This is repeated NPROVTOP times.
- IDSURF is the number of the valve face being profiled (see Fig. 3.10).
- NZSEAT is the number of cells between the valve surface and the valve seat.
- NZEDGE is the number of cells between the top of the valve seat and the edge of the runner.
- NZPORT is the number of layers of cells from the top of the valve seat to the top of the valve port.
- NPTSPRO is the number if lines of valve profile data that follows for that particular valve surface.
- NZEDGE and NZPORT are used by K3VPREP to smoothly interpolate the grid lines between the valve surface profiles to a flat line at the top of the valve port.
- The values of NZEDGE and NZPORT are not fixed and can be altered to obtain a better mesh quality in the valve port region.

<u>XSHIFT, YSHIFT, RADPORT</u>

- Format 3F8.3
- XSHIFT and YSHIFT are the coordinates of the valve axis in x-y space.
- RADPORT is the radius of the valve port of that valve.

RVALVE, ZVALVE

- Format 2F8.3
- The valve top surface profile data.
- Radius vs. z height, with Z=0 at the edge of the valve skirt.
- There are NPTSPRO lines of r vs. z data for each valve surface being profiled.

NPROVFCE

- Format A10, I5.
- Number of valve bottom surfaces to be profiled.

IDSURF, NPTSPRO

- Format 2I4.
- IDSURF is the number of the valve surface being profiled (see Fig. 3.10).
- NPTSPRO is the number if lines of valve profile data that follows for that particular valve surface.

XSHIFT, YSHIFT

- Format 3F8.3
- XSHIFT and YSHIFT are the coordinates of the valve axis in x-y space

RVALVE, ZVALVE

- Format 2F8.3
- The valve bottom surface profile data.
- Radius vs. z height with Z=0 at the valve edge.
- The first r value is at the valve axis and therefore must be zero

This is the final IPREP parameter than needs to be defined for the MSU-ERL method. All other parameters are assigned a value of zero

4.5 K3V1PREP, GRIDV and K3V2PREP_PENT

The MSU-ERL method involves running two separate versions of the K3VPREP code, K3V1PREP and K3V2PREP, along with a code called GRIDV that replaces the coordinates of the K3VPREP mesh with those of the Gridgen mesh. For the canted valve engine, K3V2PREP is replaced by K3V2PREP_PENT. This is the only difference between generating a vertical valve engine mesh and a canted valve engine mesh. The following steps are followed to run the code and obtain the final mesh for simulation in KIVA-3V.

Step 1

- Run the K3V1PREP code with the IPREP file as input.
- The K3VPREP code outputs a file called TAPE16, which is basically the mesh file in TAPE17 format after reshaping and before patching.

Step 2

- Run the GRIDV program, with the TAPE16 file and the final Gridgen mesh in UNV format as input.
- The output for this is the FORT.20 file. The FORT.20 file is essentially the TAPE16 file with all the x, y, z coordinates replaced by those from the Gridgen mesh.
- Rename the FORT.20 file as TAPE16
- Note that the mesh at this point does not contain the valve geometry.

Step 3

- Run the K3V2PREP_PENT code with the new TAPE16 file and the IPREP file as input.
- In the K3V2PREP_PENT code, the angle at which the intake and exhaust regions are tilted must be specified. This value is hard coded and must be modified by the user and the code recompiled for each engine geometry generated. If the engine geometry has vertical valves, the K3V2PREP code can be used, or the angle of tilt can be specified as zero. The final mesh with the correct geometry and valve shapes will be output as the TAPE17 file.

This is the end of the MSU-ERL mesh generation method. The final TAPE17 file is then fed into the KIVA-3V solver to obtain a numerical solution of the flow in the cylinder.

Chapter 5

Effect of the KIVA-3V H-H Grid on Y^+ and the Computed Solution

5.1 Overview

Since a mesh fine enough to resolve all the length scales in a turbulent flow will require immense computational resources to solve, turbulence models are used in most CFD analyses to simulate the behavior of turbulent flows. The KIVA-3V solver offers three types of turbulence models. They are: the standard k-ε model with wall functions, the sub-grid scale (SGS) model and the RNG k-ε model. The most frequently used turbulence model in KIVA-3V is the standard k-ε model with wall functions.

The k-ε turbulence model with wall functions is valid as long as the y^+ values of the grid points next to the wall are in the range of 30-200 [3]. One of the main factors affecting the y^+ values is the perpendicular distance of the grid point from the wall. If the grid point is too close to the wall, the y^+ values will fall below the acceptable range of 30-200, resulting in an inaccurate solution [3]. Figure 5.1 illustrates this concept.

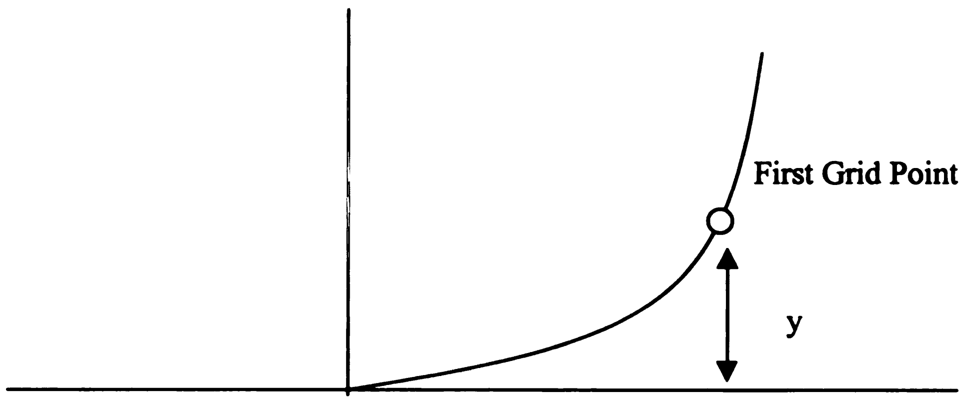


Figure 5.1: y^+ in boundary layer

When modeling cylindrical geometries using KIVA-3V, there are invariably many areas of the mesh where the grid points are located very close to the wall. This is especially true near the "corners' of cylindrical geometries. Figure 5.2 shows a cross section through the cylinder of a vertical valve IC engine mesh. Figure 5.2a is the cross section of a mesh with approximately fifty thousand grid points and Fig. 5.2b is a cross section of a mesh with approximately four hundred thousand grid points. The "corners" of the mesh are indicated.

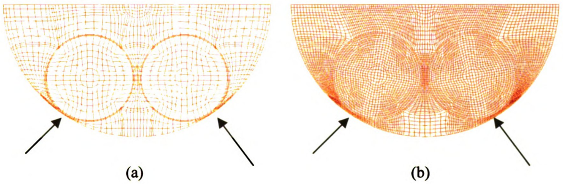


Figure 5.2: a) Cross section of vertical valve IC engine mesh with 50K grid points, b) Cross section of vertical valve IC engine mesh with 400k grid points

It can be seen in Fig 5.2 that the grid points near the corners of the mesh are clustered very close to the wall, especially for the 400k mesh. When using the k-ε turbulence model with wall functions, the y^+ values in these regions are expected to fall below the acceptable range of values. This chapter investigates the effect of the grid structure y^+ and subsequently on the computed solution.

5.2 Cases Studied for this Investigation

The following simulations were performed to investigate the effect of the grid structure on y-plus. The names by which these grids will be referred to for the reminder of the chapter are indicated in parenthesis.

- 50,000 grid point mesh (50k)
- 400,000 grid point mesh (Grid0)
- 400,000 grid point mesh with one plane of cells next to wall removed (Grid1).
- 400,000 grid point mesh with two planes of cells next to the wall removed (Grid2).

The initial conditions, boundary condition and operating conditions for all four grids were exactly the same. Figure 5.3 shows a cross section through the cylinder of each of the four grids described above.

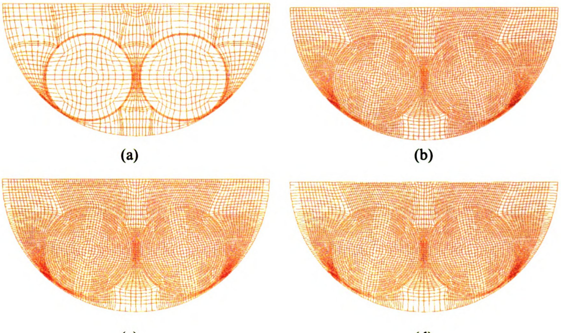


Figure 5.3: Grid distribution for a) 50k, b) Grid0, c) Grid1, d) Grid2

5.3 Comparison of Y^+ Values

The following figures show the computed y-plus contours on the surface of each grid at every 30 degrees after TDC of the piston (start of the intake stroke).

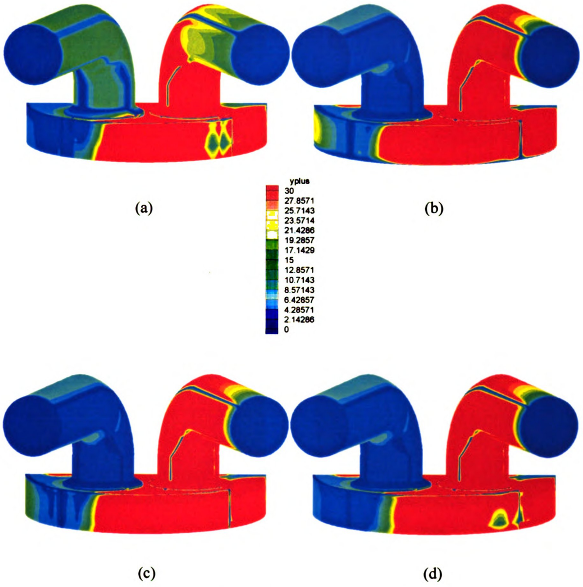


Figure 5.4: y^+ values at 30 degrees after TDC for a) 50k, b) Grid0, c), Grid1, d) Grid2

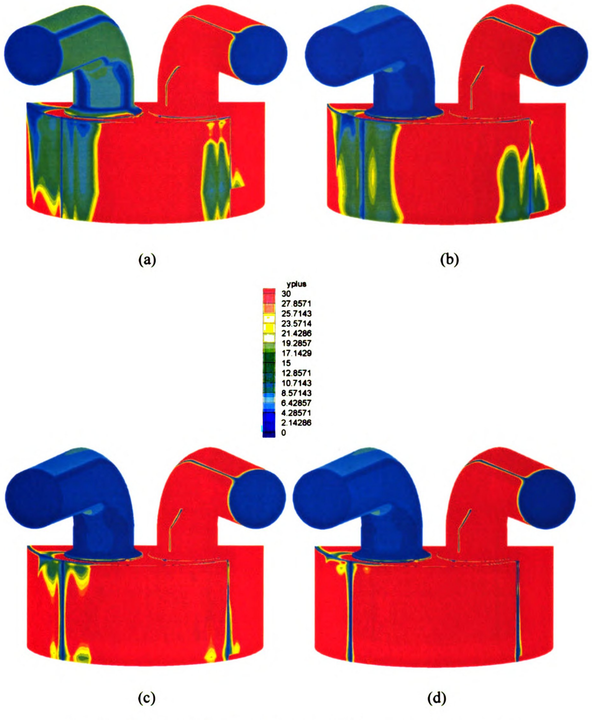


Figure 5.5: y^+ values at 60 degrees after TDC for a) 50k, b) Grid0, c), Grid1, d) Grid2

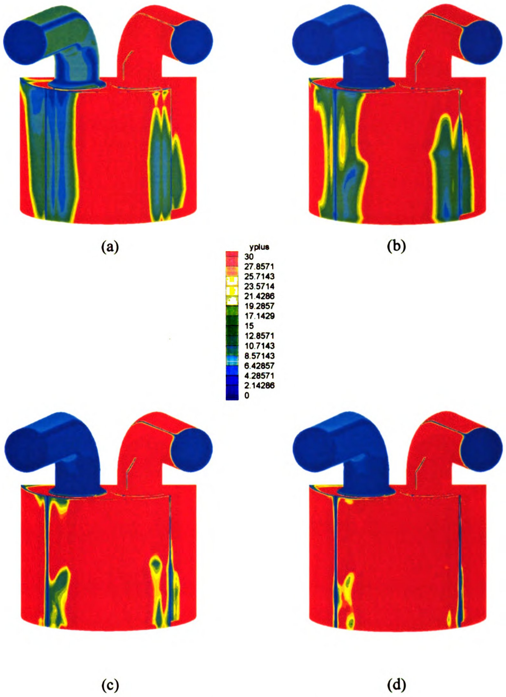


Figure 5.6: y^{+} values at 90 degrees after TDC for a) 50k, b) Grid0, c), Grid1, d) Grid2

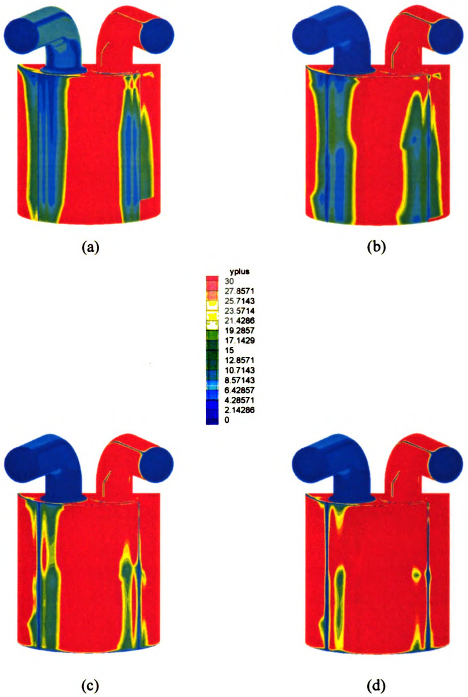


Figure 5.7: y^+ values at 120 degrees after TDC for a) 50k, b) Grid0, c), Grid1, d) Grid2

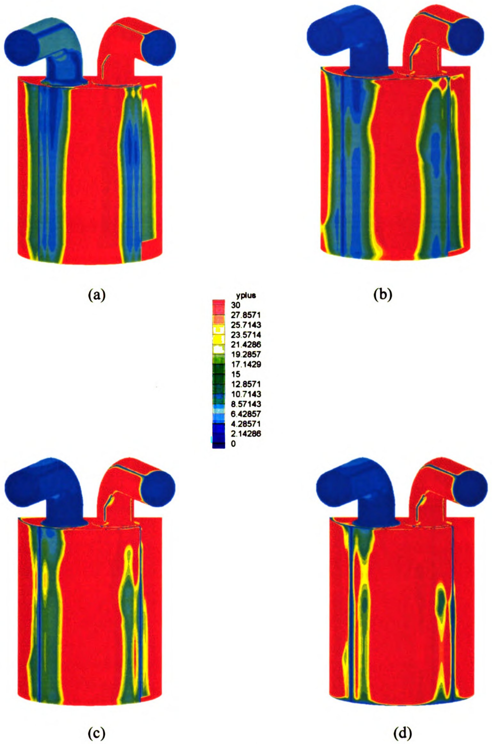


Figure 5.8: y^+ values at 150 degrees after TDC for a) 50k, b) Grid0, c), Grid1, d) Grid2

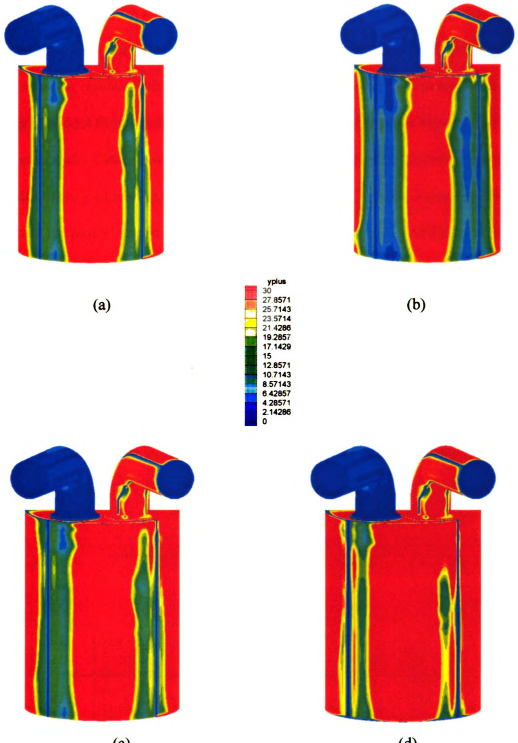


Figure 5.9: y^+ values at 180 degrees after TDC for a) 50k, b) Grid0, c), Grid1, d) Grid2

It can be seen from Fig 5.4 – 5.9 that the areas of bad y-plus correspond exactly with the corners of the mesh where the grid points are very close to the wall, as depicted in Fig. 5.3. Removing layers of cells next to the wall is shown to decrease the areas of bad y-plus, as is evident from the difference in the computed y-plus values between Grid0, Grid1 and Grid2. Grid2, with two layers of cells next to the wall removed, shows markedly better y-plus values than the 50k, Grid0 and Grid1 grids. Assuming that a finer mesh with the best y^+ values provides the most accurate solution, Grid2 is taken as a baseline for comparison with the solutions from the other three cases.

5.4 Comparison of the Computed Solution

In order to study the effect of the grid on y-plus and subsequently on the computed solution, the relative difference in velocity between the four cases was investigated. The following figures show contours of relative difference in velocity between 50k and Grid2 at different cross sections in the cylinder at 60, 120 and 180 degrees after TDC.

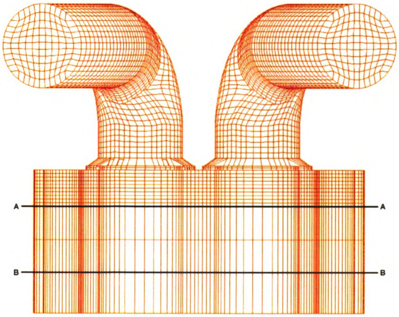


Figure 5.10: a) Grid at 60 degrees after TDC

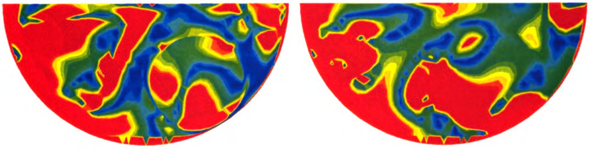
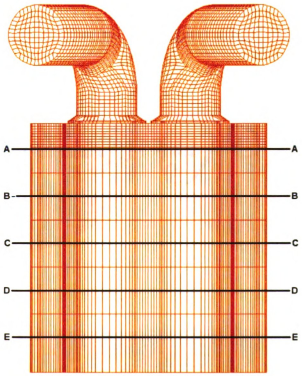

(b) (c)

Figure 5.10: Relative difference in velocity between 50k and Grid2 at 60 degrees after TDC at b) section A-A and c) section B-B



(a)

Figure 5.11: a) Grid at 120 degrees after TDC

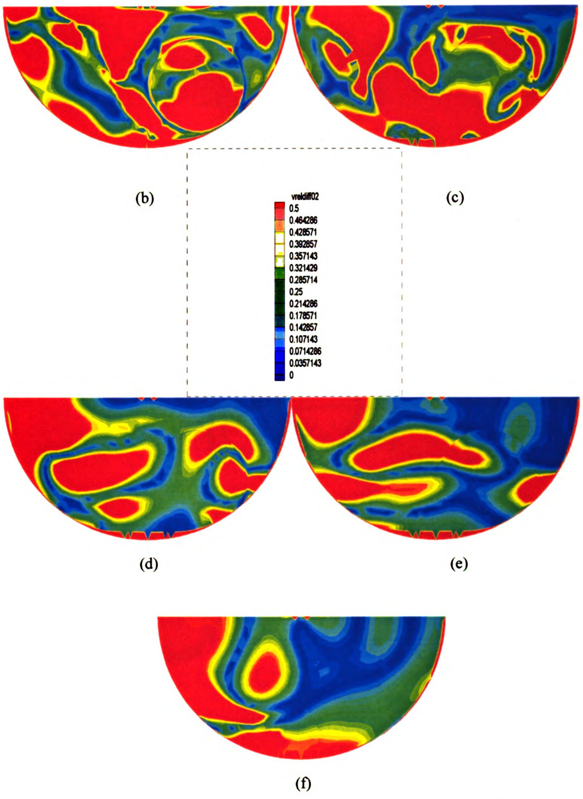
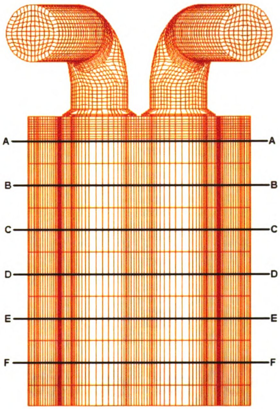
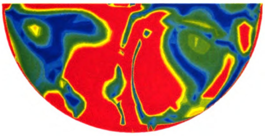


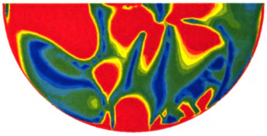
Figure 5.11: Relative difference in velocity between 50k and Grid2 at 120 degrees after TDC at b) section A-A, c) section B-B, d) section C-C, e) section D-D and f) section E-E



(a)



(b)



(c)

Figure 5.12: a) Grid at 180 degrees after TDC, relative difference in velocity at 180 degrees after TDC between 50k and Grid2 at b) section A-A and c) section B-B

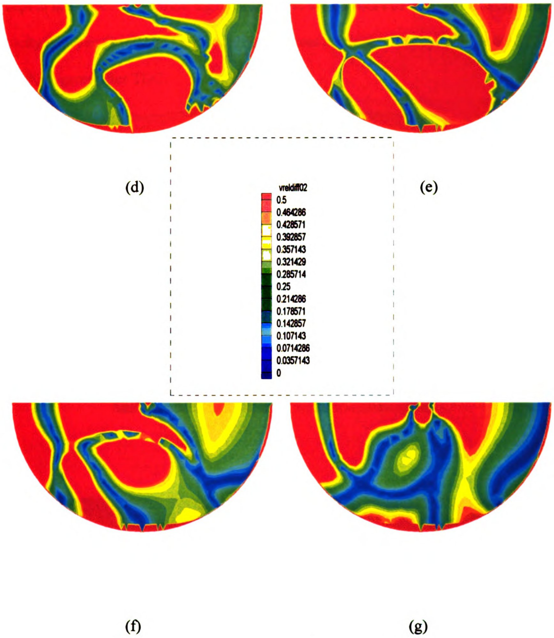
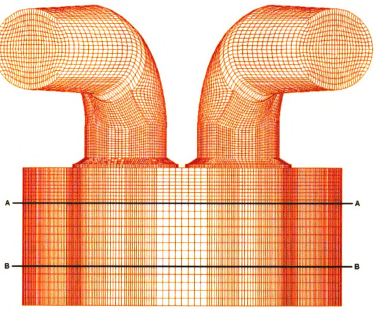
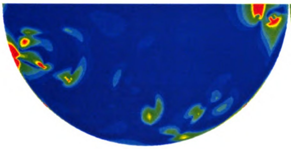


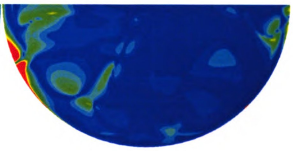
Figure 5.12: relative difference in velocity between 50k and Grid2 at 180 degrees after TDC at d) section C-C, e) section D-D, f) section E-E and g) section F-F

It can be seen from the figures above that the solutions for 50k and Grid2 are quite different. While this can be attributed to the difference in y-plus, it can also be attributed to the difference in grid density. More conclusive evidence of the effect of y-plus on the solution was sought by comparing grids with the same grid distribution everywhere except at the walls. The following figures shows the contours of relative difference in velocity between Grid0 and Grid2 at different cross sections in the cylinder at 60, 120 and 180 degrees after TDC.



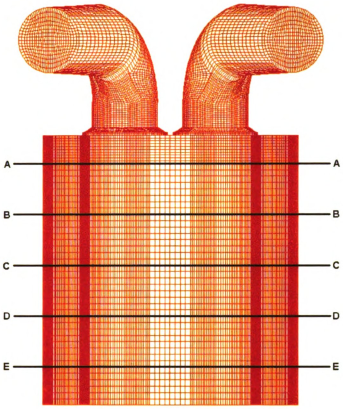
(a)



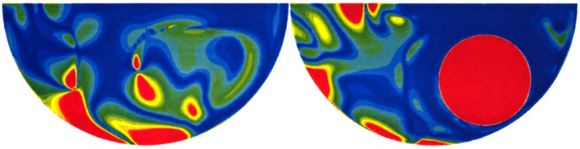


(b) (c)

Figure 5.13: a) Grid at 60 degrees after TDC. Relative velocity difference at 60 degrees after TDC between grid0 and Grid2 at a) section A-A and b) section B-B

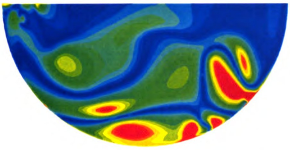
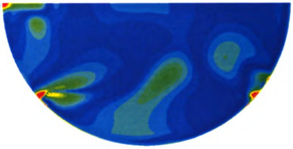


(a)

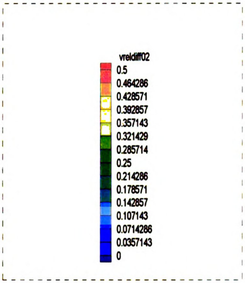


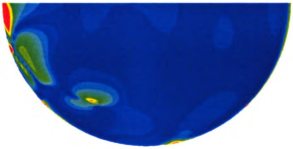
(b) (c)

Figure 5.14: a) Grid at 120 degrees after TDC. Relative difference in velocity at 120 degrees after TDC between Grid0 and Grid2 at b) section A-A and c) section B-B

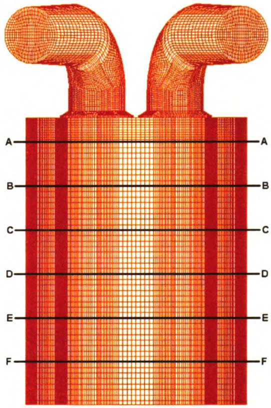
(d) (e)



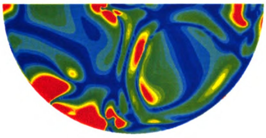


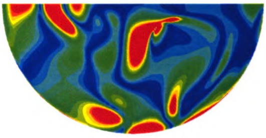
(f)

Figure 5.14: Relative difference in velocity at 120 degrees after TDC between Grid0 and Grid2 at d) section C-C, e) section D-D and f) section E-E



(a)





(b) (c)

Figure 5.15: a) Grid at 180 degrees after TDC. Relative difference in velocity at 180 degrees after TDC between Grid0 and Grid2 at b) section A-A and c) section B-B

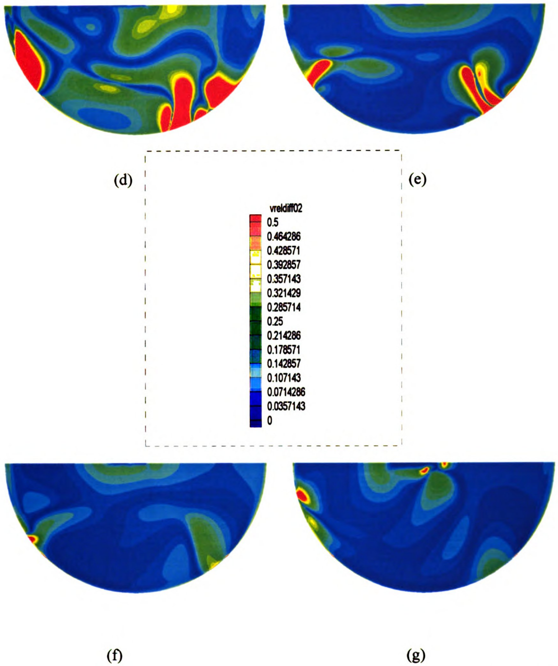
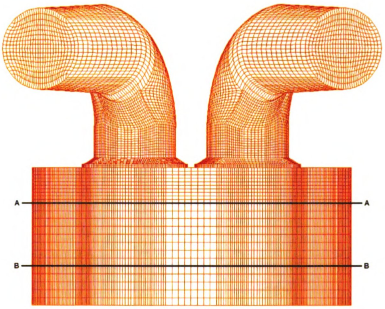
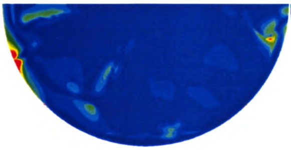


Figure 5.15: Relative difference in velocity at 180 degrees after TDC between Grid0 and grid2 at d) section C-C, e) section D-D, f) section E-E and g) section F-F

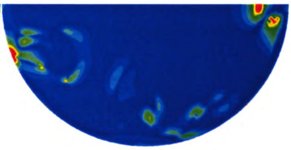
The following figures shows the contours of relative difference in velocity between Grid1 and Grid2 at different cross sections in the cylinder at 60, 120 and 180 degrees after TDC.



(a)

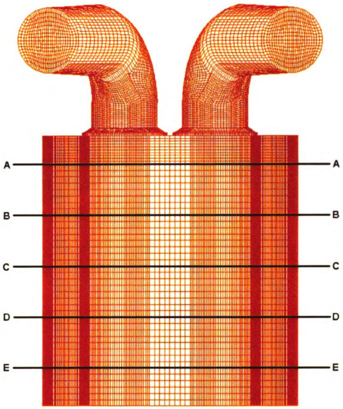


(b)

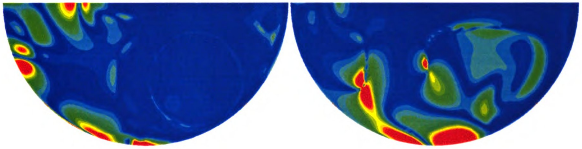


(c)

Figure 5.16: a) Grid at 60 degrees after TDC. Relative difference in velocity at 60 degrees after TDC between Grid1 and Grid2 at b) section A-A and c) section B-B



(a)



(b) (c)

Figure 5.17: a) Grid at 120 degrees after TDC. Relative difference in velocity at 120 degrees after TDC between Grid1 and Grid2 at b) section A-A and c) section B-B

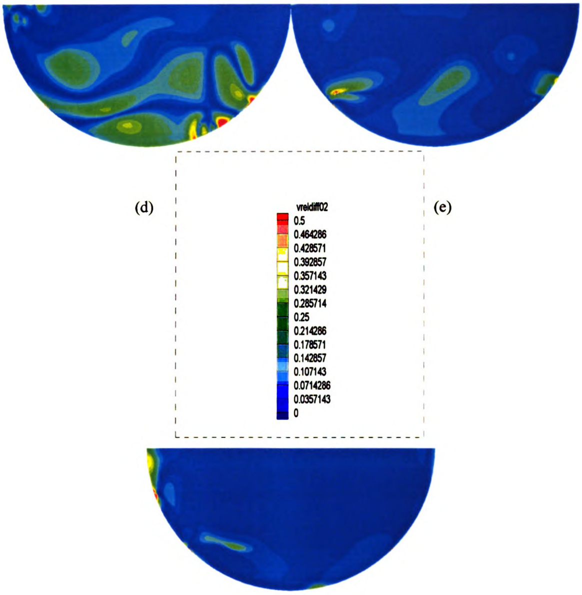
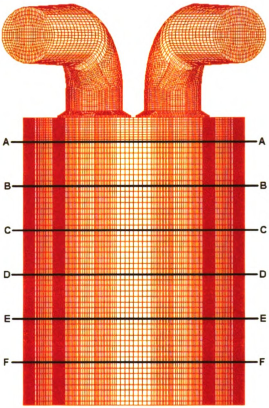
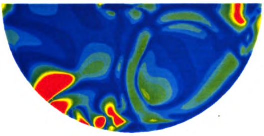


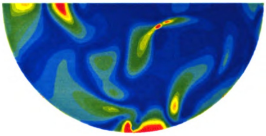
Figure 5.17: Relative difference in velocity at 120 degrees after TDC between Grid1 and Grid2 at d) section C-C, e) section D-D and f) section E-E



(a)



(b)



(c)

Figure 5.18: a) Grid at 180 degrees after TDC. Relative difference in velocity at 180 degrees after TDC between Grid 1 and Grid 2 at b) section A-A and c) section B-B

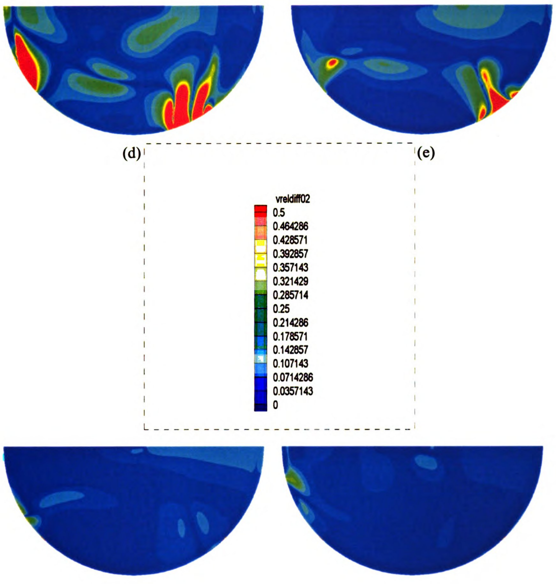
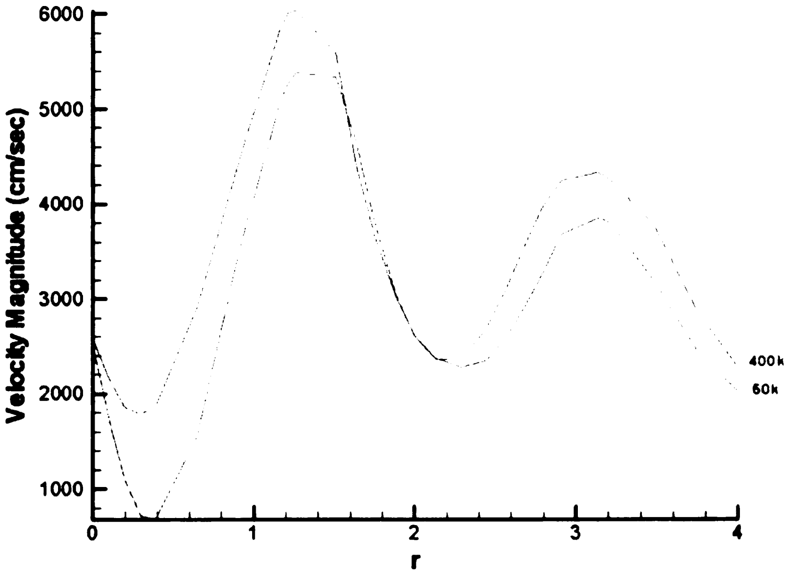
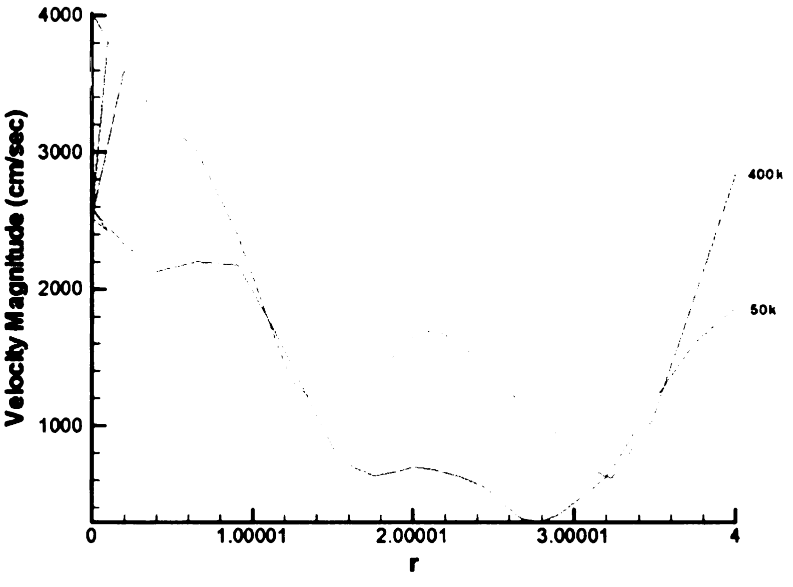


Figure 5.18: Relative difference in velocity at 180 degrees after TDC between Grid1 and Grid2 at d) section C-C, e) section D-D, f) section E-E and g) section F-F

It is clear from the plots of relative difference in velocity that the different 400k grids produced different solutions. The largest difference in the solution once again appears to occur near the mesh corners, where the y-plus values are below the acceptable range for the k-ε turbulence model to be valid. The following figures show the velocity profiles at the centerline of each cross section to further illustrate the difference in the solutions for the 50k, Grid0, Grid1 and Grid 2 cases. The cross sections are the same as those presented in Fig 5.10-5.18.

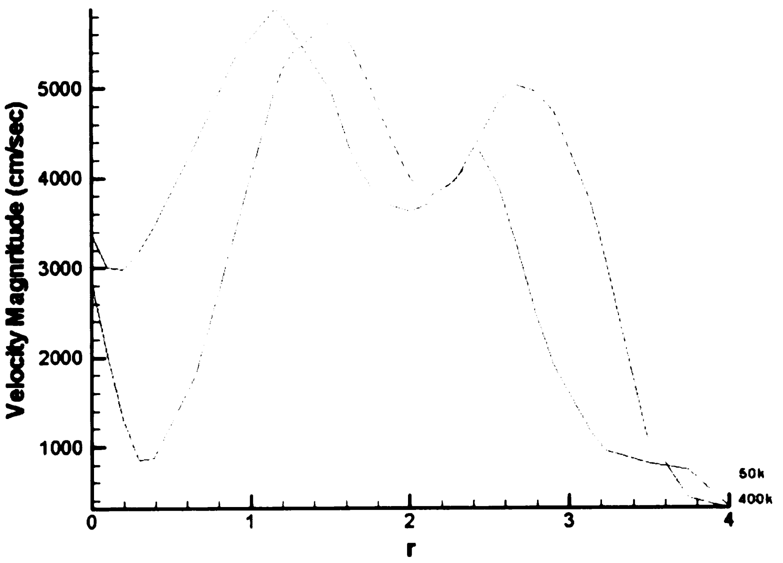


(a)



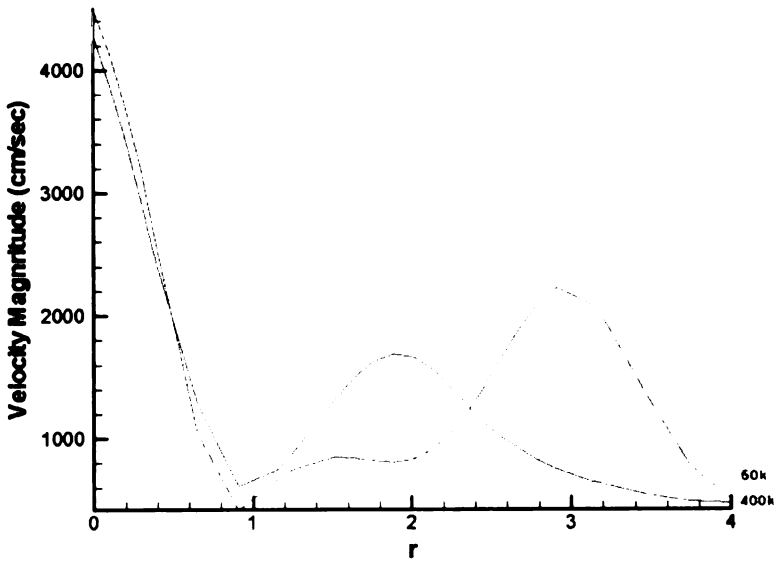
(b)

Figure 5.19: Velocity profile for 50k and Grid2 along centerline at 60 degrees after TDC at a) section A-A and b) section B-B



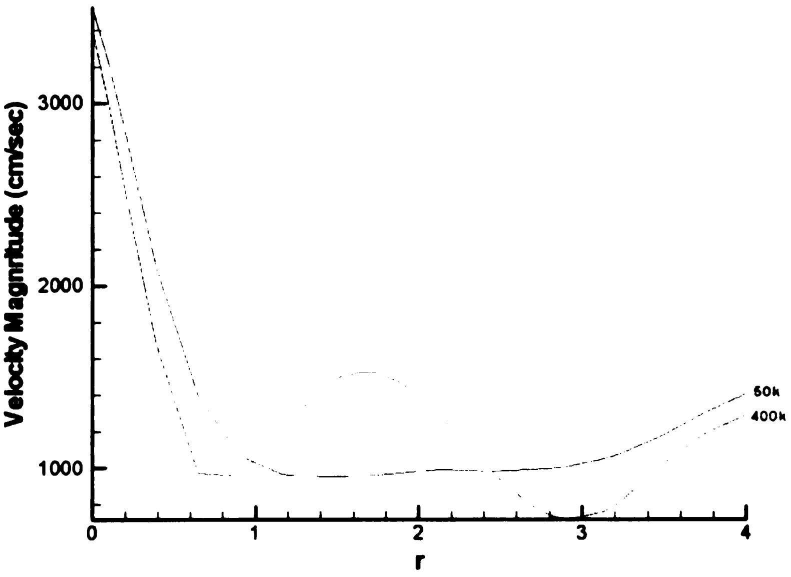
Velocity Magnitude (cm/sec)
5000
4000
3000
2000
1000
0
1
2
3
4
r
60k
400k

(a)

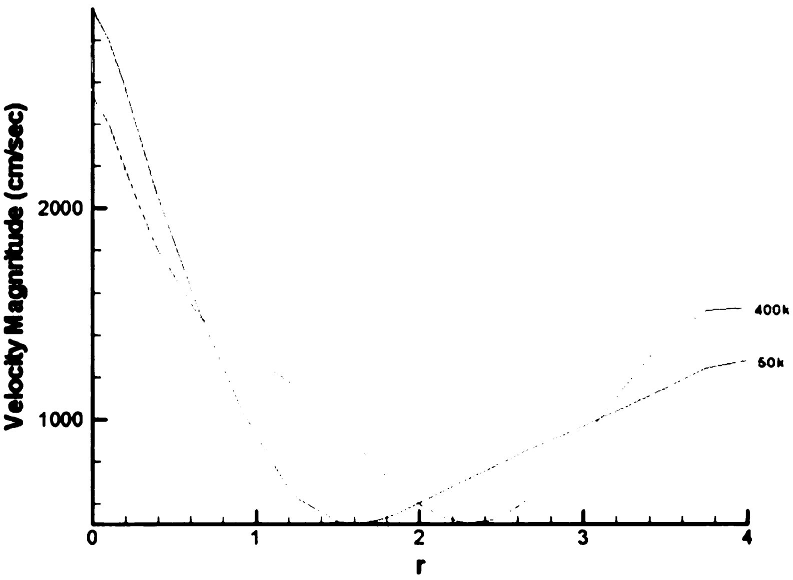


Velocity Magnitude (cm/sec)
4000
3000
2000
1000
0
1
2
3
4
r
60k
400k

(b)

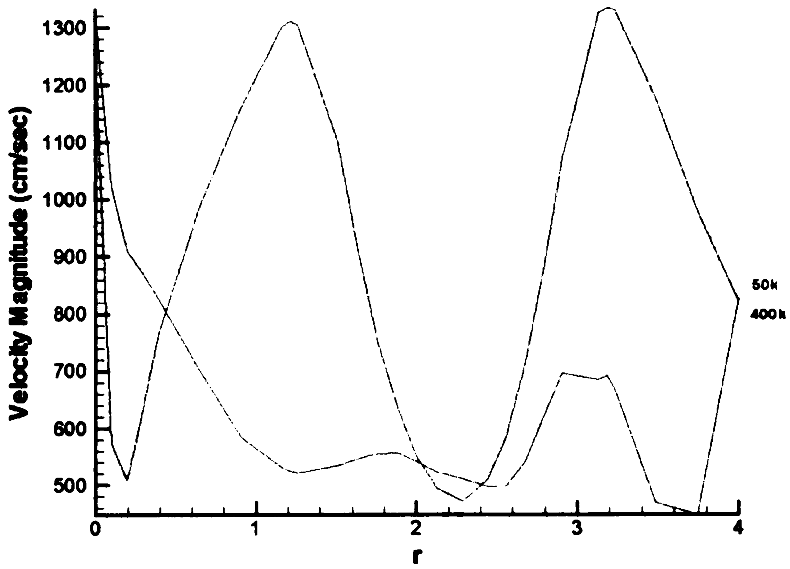


(c)



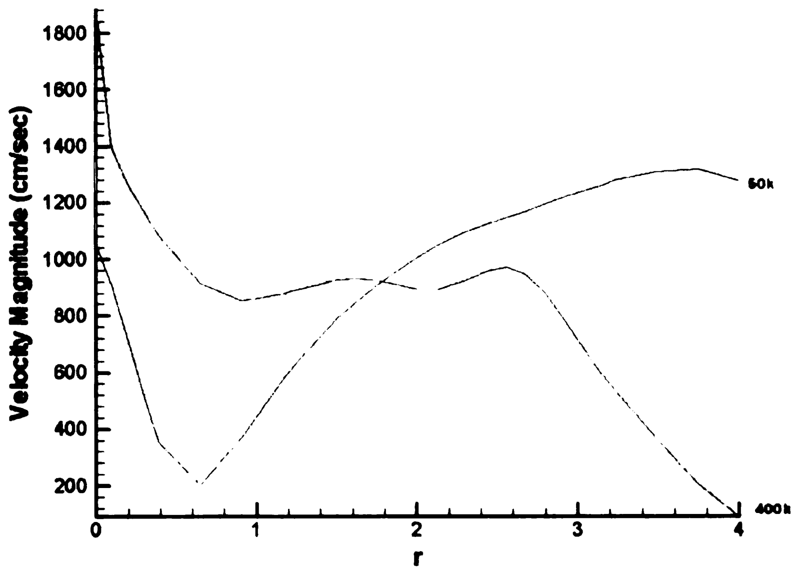
(d)

Figure 5.20: Velocity profile for 50k and Grid2 along centerline at 120 degrees after TDC at a) section A-A, b) section B-B, c) section C-C and d) section D-D



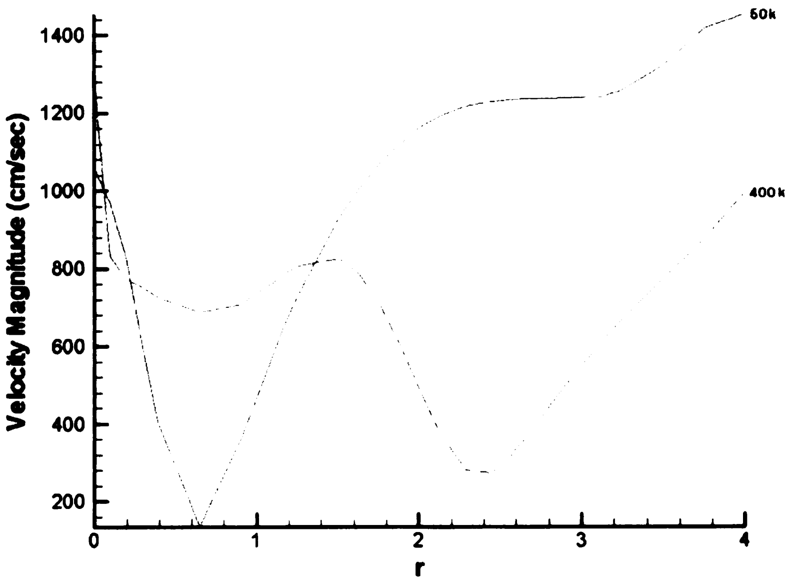
Velocity Magnitude (cm/sec)
1300
1200
1100
1000
900
800
700
600
500
0
1
2
3
4
r
50k
400k

(a)



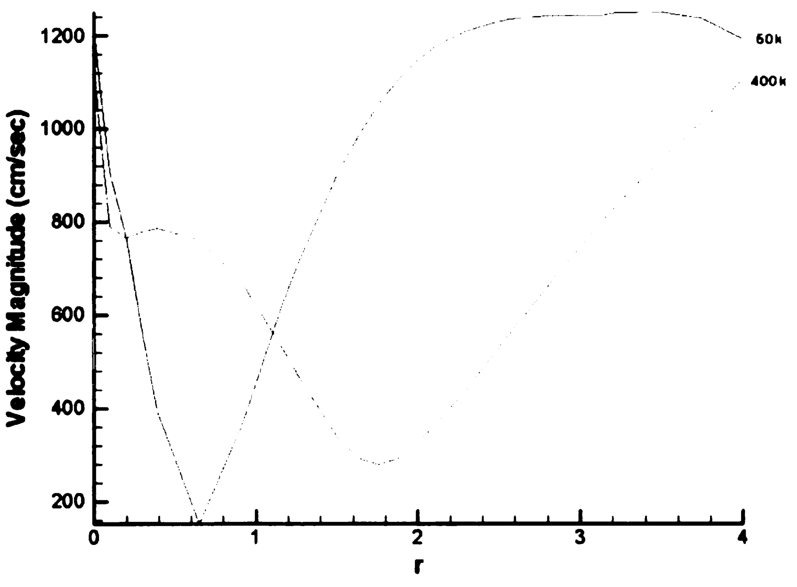
Velocity Magnitude (cm/sec)
1800
1600
1400
1200
1000
800
600
400
200
0
1
2
3
4
r
50k
400k

(b)



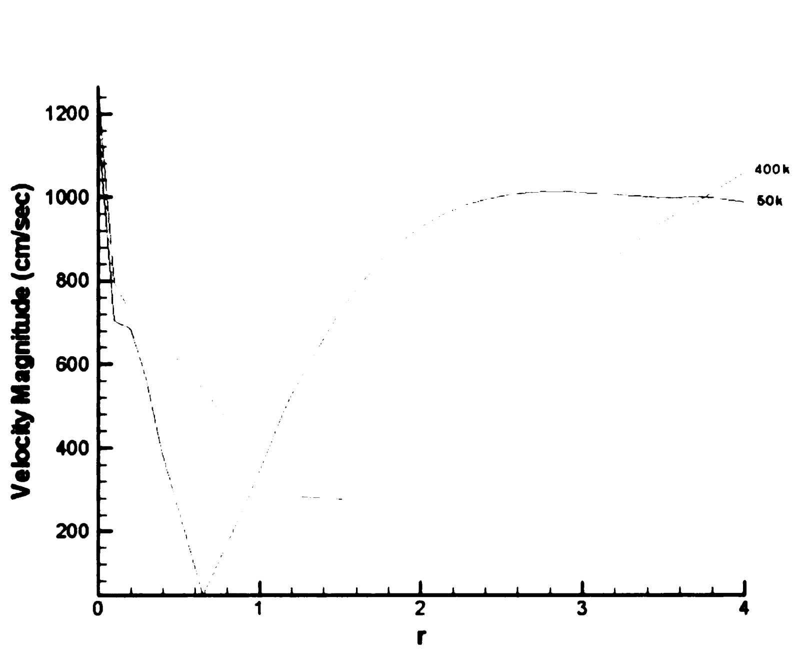
60k
400k
Velocity Magnitude (cm/sec)
r

(c)

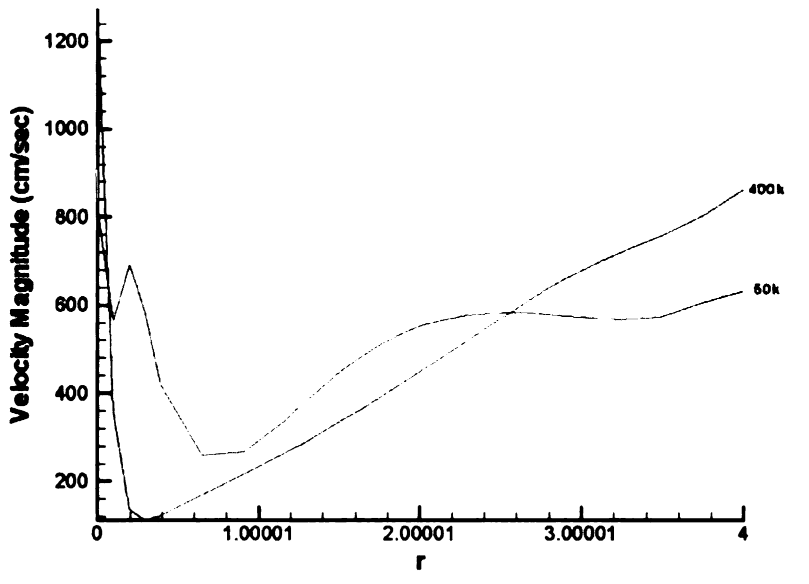


60k
400k
Velocity Magnitude (cm/sec)
r

(d)

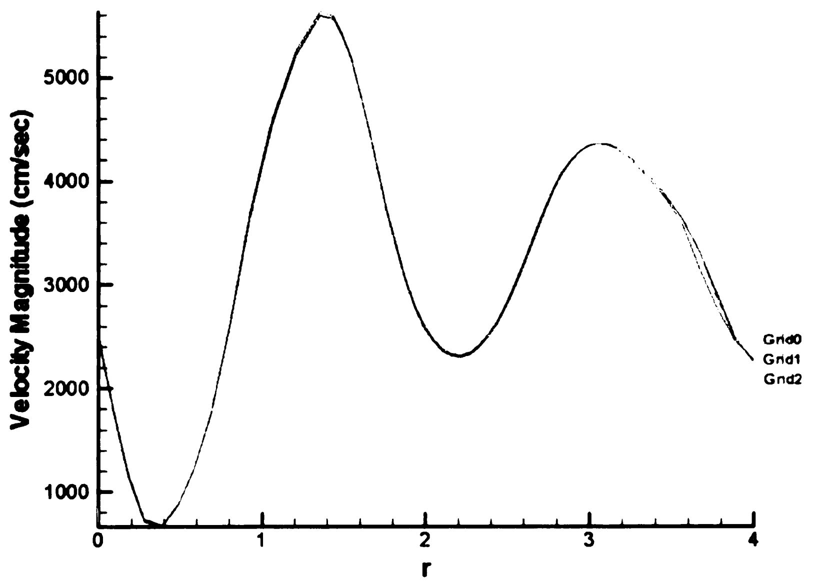


(e)

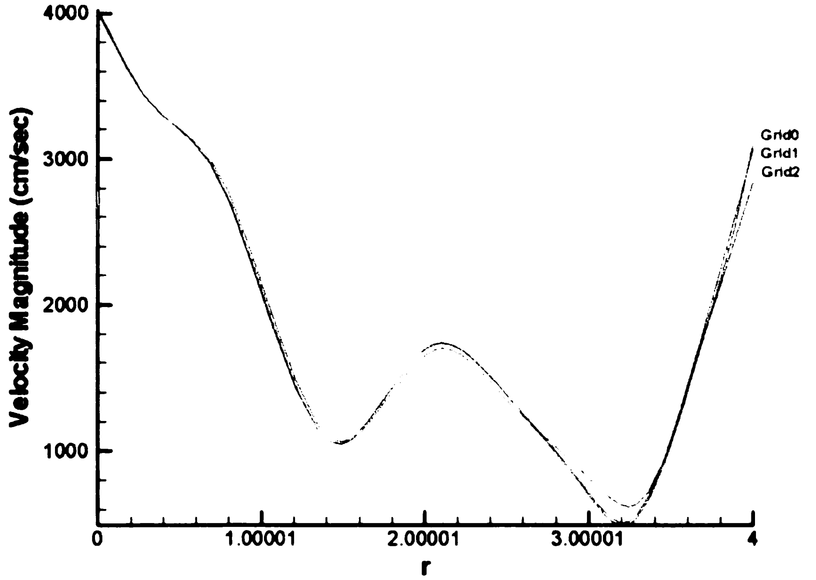


(f)

Figure 5.21: Velocity profile for 50k and Grid2 along centerline at 180 degrees after TDC at e) section E-E and f) section F-F

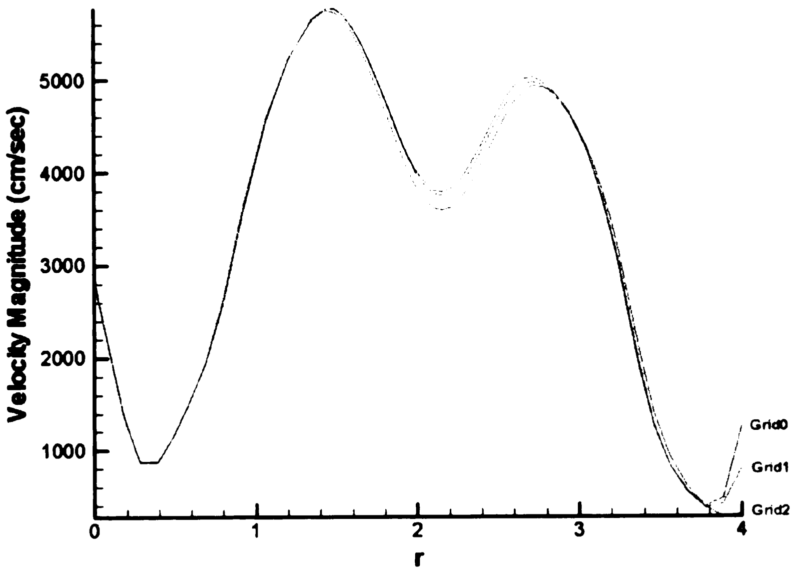


(a)

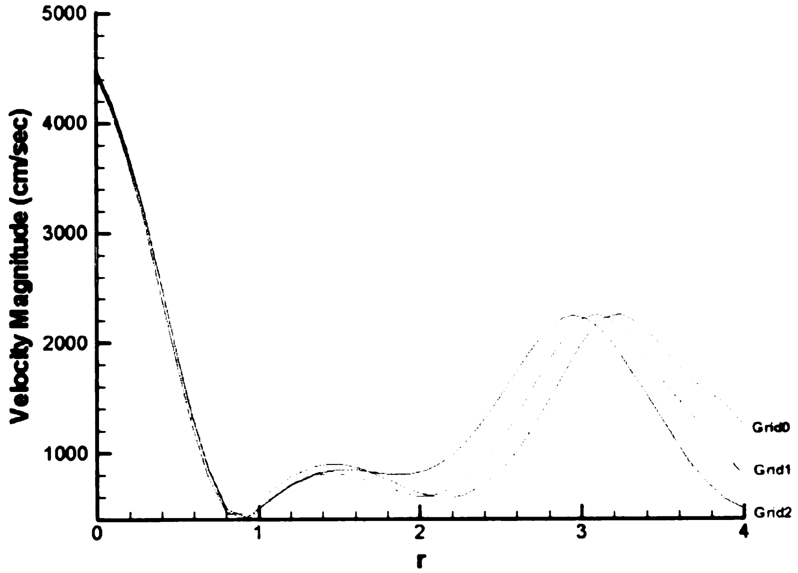


(b)

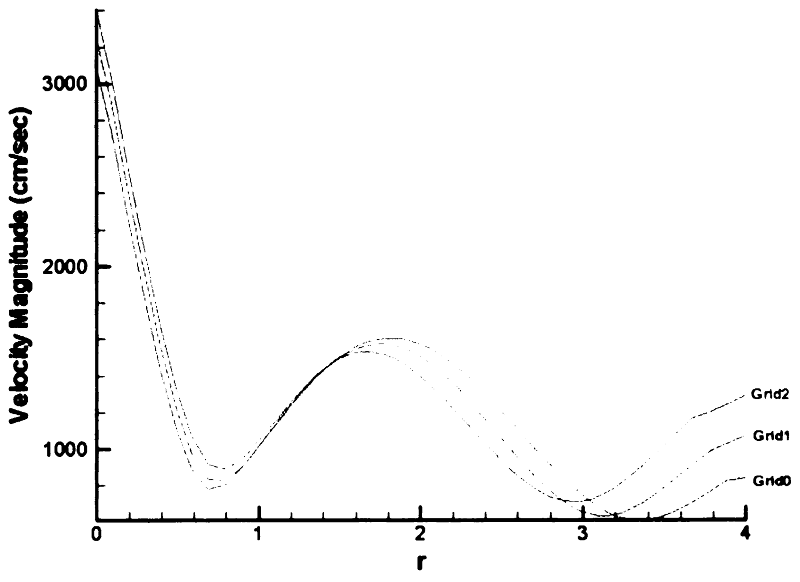
Figure 5.22: Velocity profile for Grid0, Grid1 and Grid2 along centerline at 60 degrees after TDC at a) section A-A and b) section B-B


Velocity Magnitude (cm/sec)
5000
4000
3000
2000
1000
0
1
2
3
4
r
Grid0
Grid1
Grid2

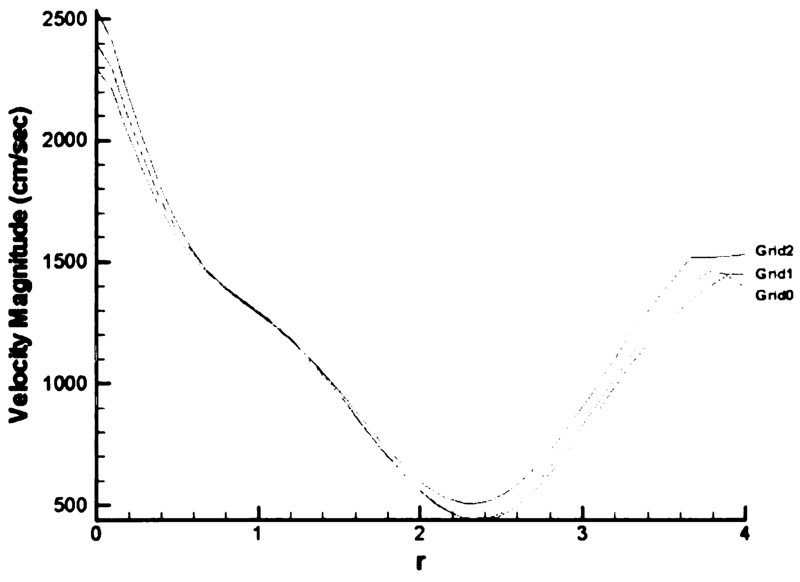
(a)


Velocity Magnitude (cm/sec)
5000
4000
3000
2000
1000
0
1
2
3
4
r
Grid0
Grid1
Grid2

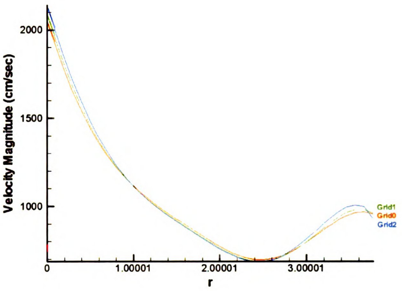
(b)


Velocity Magnitude (cm/sec)
3000
2000
1000
0
1
2
3
4
r
Grid2
Grid1
Grid0

(c)

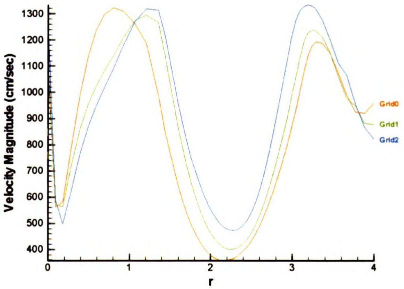

Velocity Magnitude (cm/sec)
2500
2000
1500
1000
500
0
1
2
3
4
r
Grid2
Grid1
Grid0

(d)

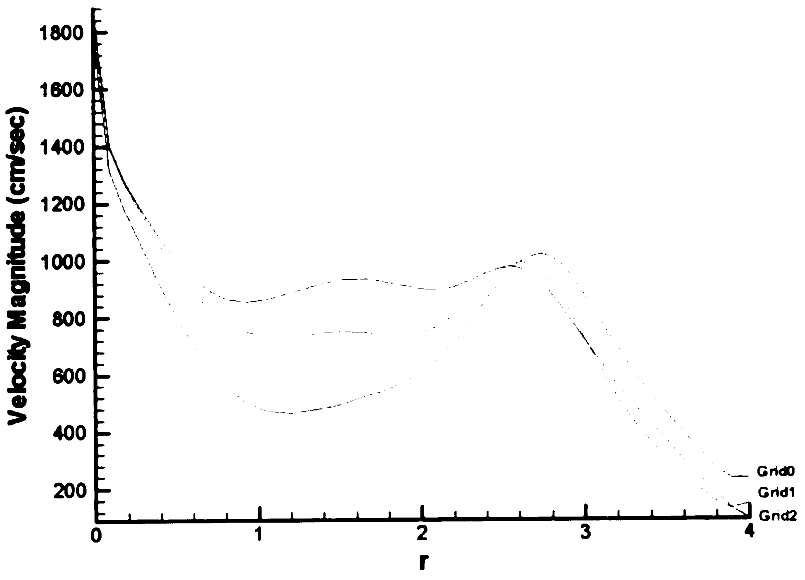


(e)

Figure 5.23: Velocity profile for Grid0, Grid1 and Grid2 along centerline at 120 degrees after TDC at a) section A-A and b) section B-B, c) section C-C, d) section D-D and e) section E-E

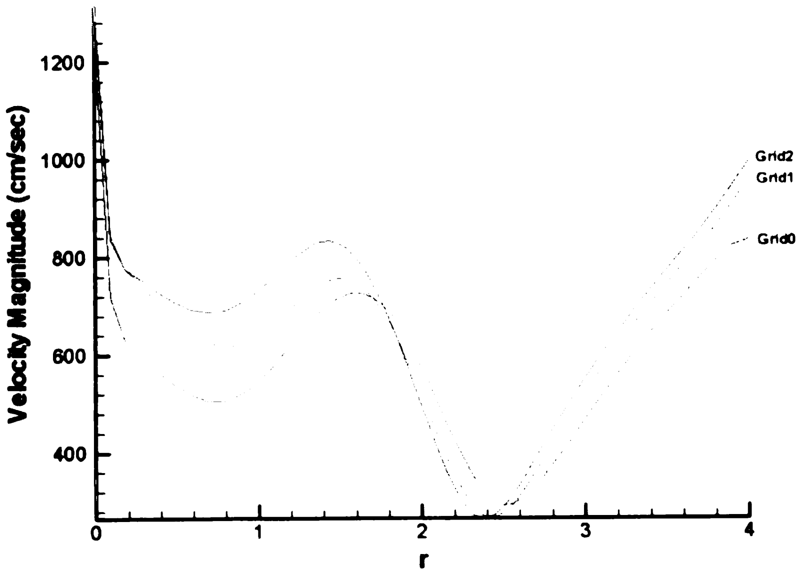


(a)



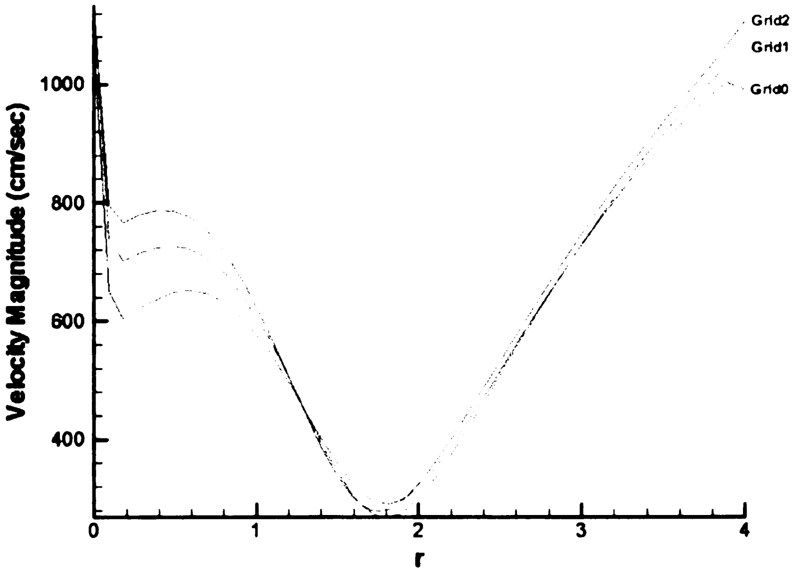
Velocity Magnitude (cm/sec)
1800
1600
1400
1200
1000
800
600
400
200
0
1
2
3
4
r
Grid0
Grid1
Grid2

(b)

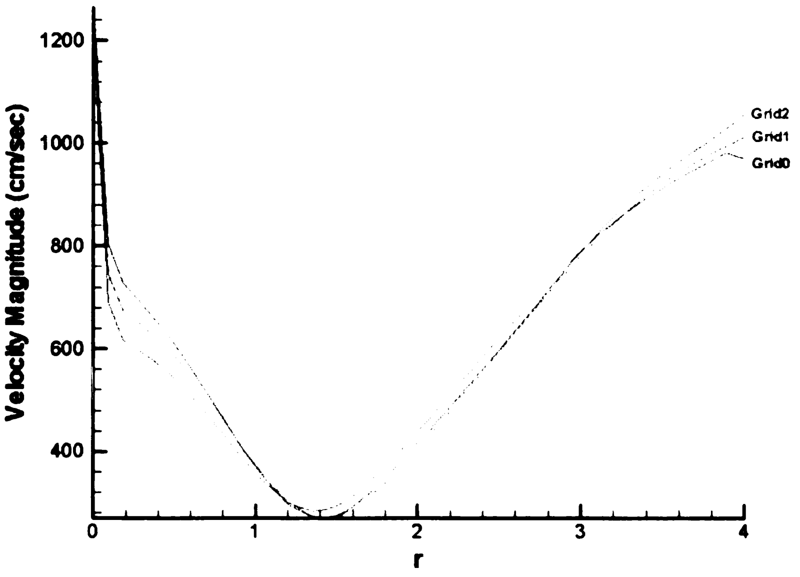


Velocity Magnitude (cm/sec)
1200
1000
800
600
400
0
1
2
3
4
r
Grid2
Grid1
Grid0

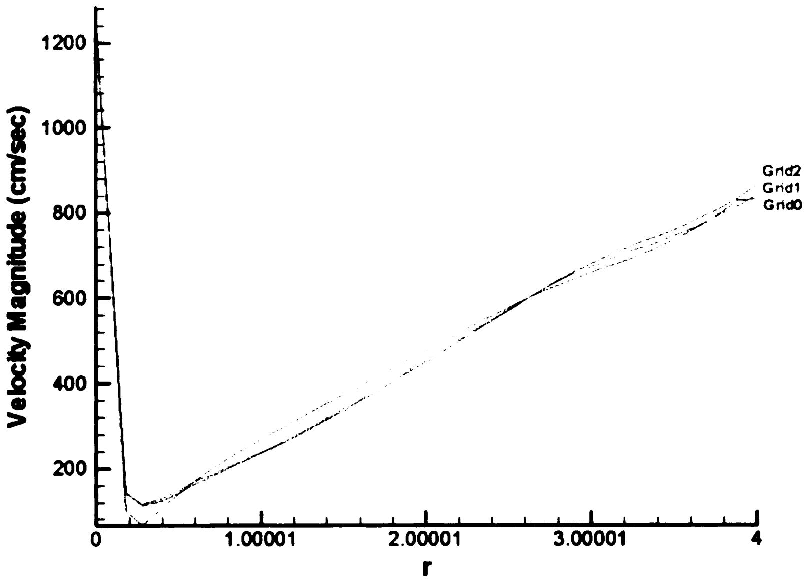
(c)


Grid2
Grid1
Grid0
Velocity Magnitude (cm/sec)
1000
800
600
400
0
1
2
3
4
r

(d)


Grid2
Grid1
Grid0
Velocity Magnitude (cm/sec)
1200
1000
800
600
400
0
1
2
3
4
r

(e)



(f)

Figure 5.24: Velocity profile for Grid0, Grid1 and Grid2 along centerline at 180 degrees after TDC at a) section A-A and b) section B-B, c) section C-C, d) section D-D, e) section E-E and f) section F-F

The plots of velocity profile for 50k, Grdi0, Grid1 and Grid2 along the centerline of the cross sections of the cylinder show further demonstrate the difference in the solutions obtained from the four grids.

5.5 Comparison of the Solution with Star-CD

The previous sections demonstrated the difference in the solution obtained form the 50k, Grid0, Grid1and Grid2 grids. Grid 2 was assumed to be the most accurate since the areas of bad y-plus were at a minimum among the four cased studied. In order to further study if the grid structure and its effect on y-plus had a significant effect on the accuracy of the solution, the solution from Grid2 was compared to a solution computed for the same engine using the Star-CD commercial CFD code. The Star-CD code does not have the same H-H grid requirement as KIVA-3V. Star-CD allows far greater flexibility in creating the mesh and produces a mesh with no "corners". The comparison between the Star-CD solution and the KIVA-3V solution was made by studying the general flow field computed by each code. The following figures show the flow field in the engine at 60, 120 and 180 degrees after TDC of the piston.

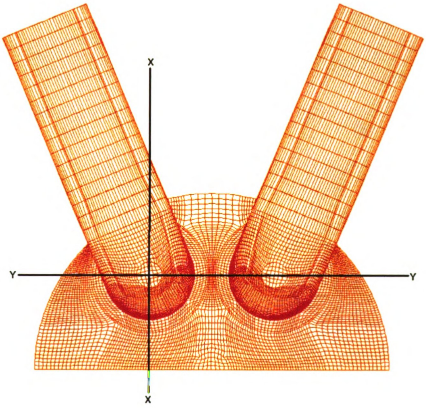
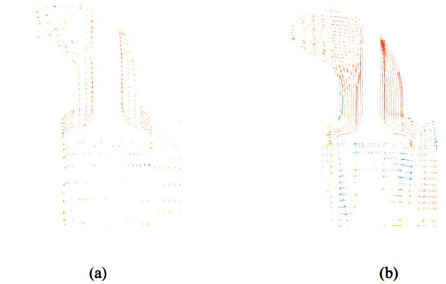


Figure 5.25: Cross sections through the engine



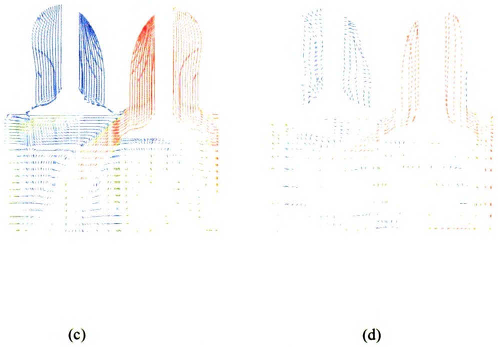


Figure 5.26: Flow field at 60 degrees after TDC, a) Grid2 section X-X, b) Star-CD section X-X, c) Grid2 section Y-Y, d) Star-CD section Y-Y

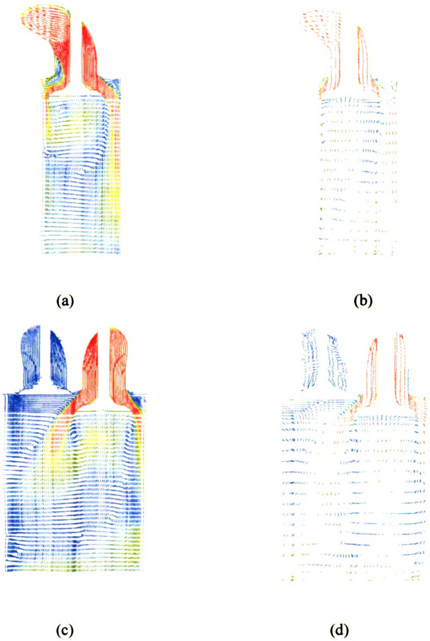
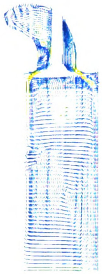
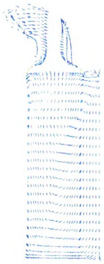
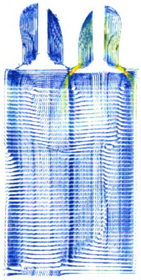


Figure 5.27: Flow field at 120 degrees after TDC, a) Grid2 section X-X, b) Star-CD section X-X, c) Grid2 section Y-Y, d) Star-CD section Y-Y


(a)


(b)


(c)

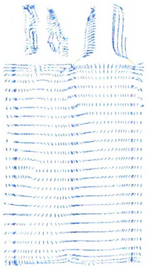

(d)

Figure 5.28: Flow field at 180 degrees after TDC, a) Grid2 section X-X, b) Star-CD section X-X, c) Grid2 section Y-Y, d) Star-CD section Y-Y

It can be seen from the figures above that the general flow structure in Grid2 solution and the Star-CD solution are approximately the same. There is more detail in the Grid2 solution owing to the greater number of grid points but the overall structure of the flow is the similar. The flow in areas where the y-plus in Grid2 is bad is similar to the flow in the corresponding Star-CD result. This suggests that the bad y-plus values in the Grid2 solution did not adversely affect the overall flow field in the engine. However, it is not clear if the Star-CD solution can itself be considered an accurate enough solution to use as a baseline for comparing with the KIVA-3V solutions.

5.6 Conclusions and Recommendations

The simulations described in this chapter and the results obtained from them clearly show that the grid structure in KIVA-3V affects the computed y^+ values at the grid points. Removing planes of cells next to the wall is shown to be an effective method to reduce the areas of bad y^+. But it was not possible to determine whether removing planes of cells improved the solution obtained in KIVA-3V. While the solutions from KIVA-3V and Star-CD showed very similar flow fields, it is not clear whether either of them can be considered the most accurate solution obtainable. Further study must be done to determine if areas of bad y^+ in KIVA-3V indeed affects the accuracy of the solution. While it is impossible to completely get rid of grid "corners" in KIVA-3V due to its H-H grid requirement, it is possible to reduce its severity by either removing planes of cells next to the wall or by generating a mesh in Gridgen that always places the first grid point far enough away from the wall to ensure a y^+ value greater than 30. Since KIVA-3V can handle moving grids, it may also be possible to modify the code such that the location of the first grid point away from the wall are dynamically adjusted during the simulation to ensure that the y-plus values are always in the acceptable range. This will be a more desirable solution to the problem because y-plus is a function of engine rpm, and crank angle and it will be impossible to generate a mesh where all regions satisfy the minimum y^+ value requirement.

References

[1] Amsden, A.A., KIVA-3V: A Block-structured KIVA Program for Engines with Vertical or Canted Valves, LA-13313-MS, (July 1997)

[2] Amsden, A.A, O'Rourke, P.J., Butler, T.D., KIVA-II: A computer Program for Chemically Reactive Flows with Sprays, LA-11560-MS (May 1989)

[3] Wilcox, D.C., Turbulence Modeling for CFD, 2nd Edition, DCW Industries (July 1998)