

THREE-DIMENSIONAL RETARDING WALLS AND
FLOW IN THEIR VICINITY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

KEMAL ATILGAN TOKER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
THE DEPARTMENT OF MECHANICAL ENGINEERING

DECEMBER 2004

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. S. Kemal İDER
Chair of Mechanical Engineering Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. M. Haluk AKSEL
Supervisor

Examining Committee Members:

Prof. Dr. Kahraman ALBAYRAK (METU, ME) _____

Prof. Dr. M. Haluk AKSEL (METU, ME) _____

Prof. Dr. Hüseyin VURAL (METU, ME) _____

Prof. Dr. İsmail Hakkı TUNCER (METU, AE) _____

Prof. Dr. Ünver KAYNAK (TOBB ETU, ME) _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name :

Signature :

ABSTRACT

THREE-DIMENSIONAL RETARDING WALLS AND FLOW IN THEIR VICINITY

TOKER, Kemal Atılgan
Ph.D., Department of Mechanical Engineering
Supervisor: Prof. Dr. M. Haluk AKSEL

December 2004, 164 pages

The performance prediction of solid propellant rocket motor depends on the calculation of internal aerodynamics of the motor through its operational life. In order to determine the control volume boundaries, in which the solutions will be carried out, a process called “grain burnback calculation” is required. During the operation of the motor, as the interface between the solid and gas phase moves towards the solid propellant in a direction normal to the surface, the combustion products are generated and the control volume expands. This phenomenon requires handling of moving boundaries as the solution proceeds.

In this thesis, Fast Marching Method is implemented to the problem of grain burnback. This method uses the upwind nature of the propellant interface motion and solves the Eikonal type equations on a fixed three-dimensional tetrahedron mesh. The control volume is coupled to a one-dimensional and a three-dimensional Euler aerodynamic solver in order to evaluate the performance of the engine. The speed by which the interface moves depends on the static pressure on the surface of the propellant which is provided by the flow solver. An iterative method has been proposed between the interface capturing algorithms and the flow solver. Computed solutions from zero-dimensional, one-dimensional and three-dimensional flow solvers are compared and validated with experimental data.

Key-words: Solid Propellant Rocket Motor, Grain Burnback, Fast Marching Methods, Finite Volume Method.

ÖZ

ÜÇ-BOYUTLU GERİLEYEN DUVARLAR VE ÇEVRELERİNDEKİ AKIŞ

TOKER, Kemal Atılgan

Doktora, Makina Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. M. Haluk AKSEL

Aralık 2004, 164 sayfa

Katı yakıtlı roket motorları için performans tahmini, motor içerisindeki aerodinamik hesaplamalar sayesinde yapılır. Hesaplamaların yapılacağı kontrol hacminin oluşturulması için, geriye doğru yanma hesaplamalarının yapılması gerekmektedir. Motor çalıştığı süre boyunca katı ve gaz fazları arasında kalan ara yüzey, yakıtı doğru yüzeye normal yönde hareket ederken, yanma ürünleri ters yönde kontrol hacme akarlar. Bu durumda hareket eden yüzeylerin, çözüm ilerlerken takip edilmesi gerekmektedir.

Bu tez içerisinde geriye yanma problemi “Hızlı İlerleme Metodu (Fast Marching Method)” ile çözümlenmiştir. Bu metod yakıt-gaz arayüzünün tek yöne hareket etme özelliğinden faydalanıp 3-boyutlu sabit tetrahedron ağ üzerinde Eikonal tipi denklemleri çözmektedir. Kontrol hacim tek-boyutlu ve üç-boyutlu Euler akış

özücüler ile performans tahmini yapmak için birleřtirilmiřtir. Yakıt yanma hızı bu özücülerden gelen yüzey statik basıncına baėlı kalarak güncellenir. Bu yüzden ara yüzü takip eden ve akıř özen programlar birbirini takip ederek alıřırlar. özücülerden elde edilen sonuçlar gerçek roket ateřleme verileri ile karşılařtırılmıř ve sonuçların doėruluėu ispatlanmıřtır.

Anahtar Kelimeler: Katı Yakıtlı Roket Motoru, Geriye Yanma, Hızlı İlerleme Metodu, Sonlu Hacim Metodu.

To İrem TOKER

ACKNOWLEDGEMENTS

I would like to express my deepest thanks and gratitude to Prof. Dr. Haluk AKSEL for his supervision, encouragement, understanding and constant guidance.

Also I would like to express my gratitude to Prof. Dr. Ahmet ÜÇER for initializing and supporting this thesis.

I would like to thank to Dr. Tuğrul TINAZTEPE for crucial advises on burnback and flow solvers, which help combining the mathematical work to real physics of the rocket propulsion.

I would like to express my sincere appreciation to my colleagues Görkem ERTÜRK, Uğur ARKUN, Başar SEÇKİN and Ali AKGÜL for their crucial advises and invaluable efforts during the preparation of this thesis.

I would like to thank to Erinç ERDEM, Prof. Dr. İsmail Hakkı TUNCER and Dr. Mehmet Ali AK for supplying the 2-dimensional unstructured flow solver and help me extend its capabilities. I also would like to thank to Dr. Abdullah ULAŞ for his advises on physics of solid propellant combustion.

I would like to thank to ROKETSAN for partially supporting this study.

My gratitude is endless for my wife and my family whom this thesis is devoted to.

TABLE OF CONTENTS

ABSTRACT.....	IV
ÖZ	VI
ACKNOWLEDGEMENTS	IX
TABLE OF CONTENTS	X
LIST OF TABLES.....	XIII
LIST OF FIGURES.....	XIV
LIST OF SYMBOLS	XIX
1 INTRODUCTION.....	1
1.1 LITERATURE SURVEY.....	3
1.1.1 <i>Interface Tracking Methods</i>	6
1.1.2 <i>Interface Capturing Methods</i>	8
1.1.3 <i>Rocket Performance Prediction</i>	11
1.2 PURPOSE OF THE STUDY	14
1.3 CONTENTS OF THE THESIS REPORT.....	14
2 RETARDING WALLS.....	15
2.1 PROPERTIES OF CURVES.....	15
2.1.1 <i>Velocity of Interface</i>	16
2.1.2 <i>Parametric Representation of Curve Evolution</i>	16
2.1.3 <i>Growth of Oscillations</i>	17
2.1.4 <i>Entropy Condition</i>	18
2.1.5 <i>Boundary Value Formulation</i>	20
2.1.6 <i>Initial Value Formulation</i>	21
2.1.7 <i>Advantages of Both Formulations</i>	22

2.1.8	<i>Comparison of Formulations</i>	22
2.2	NUMERICAL ALGORITHMS FOR BOUNDARY VALUE PROBLEM.....	23
2.3	FAST MARCHING METHOD.....	24
2.3.1	<i>The Update Procedure</i>	25
2.3.2	<i>Approximation on Triangulated Domains</i>	25
2.3.3	<i>Derivative Approximations on Unstructured Mesh</i>	26
2.3.4	<i>Upwind Criteria</i>	28
2.4	CUT-CELL METHODOLOGY	30
2.4.1	<i>Interface in an Element</i>	31
2.4.2	<i>Capturing Interface for Zero-dimensional Solvers</i>	34
2.4.3	<i>Capturing Interface for One-dimensional Solvers</i>	35
2.4.4	<i>Capturing Interface for Three-Dimensional Flow</i>	35
2.5	NUMERICAL TEST CASES FOR FAST MARCHING METHOD.....	39
2.5.1	<i>Error Definitions</i>	39
2.5.2	<i>Grid Size</i>	40
2.5.3	<i>Two-Dimensional Test Cases</i>	40
2.5.4	<i>Three-Dimensional Test Cases</i>	53
3	FLOW SOLVERS FOR ROCKET MOTOR PERFORMANCE PREDICTION	63
3.1	ZERO-DIMENSIONAL FLOW SOLVER.....	64
3.1.1	<i>Governing Equations</i>	65
3.2	ONE-DIMENSIONAL FLOW SOLVER	67
3.2.1	<i>Program Definitions</i>	67
3.2.2	<i>Flow Solver Formulation</i>	69
3.2.3	<i>Solution Technique</i>	70
3.2.4	<i>Nozzle Flow and Thrust Calculations</i>	71
3.2.5	<i>Time Dependence</i>	72
3.3	THREE-DIMENSIONAL FLOW SOLVER.....	73
3.3.1	<i>Conservative Form of Euler Equations</i>	74
3.3.2	<i>Jacobian Matrices</i>	75
3.3.3	<i>Characteristics of Euler Equations</i>	78
3.3.4	<i>Finite Volume Solution</i>	81
3.3.5	<i>Time Integration by Runge-Kutta Method</i>	82
3.3.6	<i>Van Leer Flux Splitting</i>	83
3.3.7	<i>Riemann Solver of Roe</i>	85
3.3.8	<i>Initial and Boundary Conditions</i>	90
3.3.9	<i>Non-Dimensional Parameters</i>	93
3.3.10	<i>Test Cases for Three-Dimensional Flow Solver</i>	94

4	COUPLED TEST CASES AND RESULTS.....	107
4.1	INCREASING INTERFACE CAPTURING ACCURACY	107
4.2	RELATED PARAMETERS.....	109
4.3	TEST CASE 1 – 6C4 TEST MOTOR.....	111
4.4	TEST CASE 2 – BOOSTER SRM.....	124
4.5	TEST CASE 3 – DEFECTS IN GRAIN	131
4.6	TEST CASE 4 – HUMP EFFECT	135
5	CONCLUSION AND DISCUSSION.....	139
5.1	SUGGESTIONS FOR FUTURE WORK	140
5.2	CONCLUSION	141
	REFERENCES	142
A	USER’S MANUAL.....	151
A.1	PRE-PROCESSING STEP.....	151
A.1.1	<i>I-Deas Universal File Format.....</i>	<i>151</i>
A.1.2	<i>Universal Data Set Number 2411</i>	<i>153</i>
A.1.3	<i>Universal Data Set Number 2412</i>	<i>153</i>
A.1.4	<i>Defining Boundary Conditions</i>	<i>154</i>
A.1.5	<i>Program “ MESH_3D ”</i>	<i>154</i>
A.2	PROCESSING STEP	156
B	THREE DIMENSIONAL SOLUTIONS.....	160
B.1	6C4 RESULTS	160
B.2	BOOSTER SRM RESULTS	161
B.3	BUBBLE IN PROPELLANT RESULTS	163
	CURRICULUM VITAE.....	164

LIST OF TABLES

Table 2.1 Scenarios for Different Element Types	31
Table 2.2 Hybrid Mesh Connectivity	38
Table 2.3 Two-dimensional Test Case 0: Error Estimates	42
Table 2.4 2-dimensional Test Case 1 Error Estimates.....	44
Table 2.5 2-dimensional Test Case 2 Error Estimates.....	47
Table 2.6 2-dimensional Test Case 3 Error Estimates.....	50
Table 2.7 2-dimensional Test Case 4 Error Estimates.....	52
Table 2.8 3-dimensional Test Case 1 Error Estimates.....	55
Table 2.9 3-dimensional Test Case 2 Error Estimates.....	58
Table 2.10 3-dimensional Test Case 3 Error Estimates.....	61
Table 2.11 Three-dimensional Test Case 4 Error Estimates	62
Table 3.1 Geometric Properties and Flow Parameters for Test Case 1	100
Table 3.2 Geometric and Flow Properties for Test Case 2.....	104
Table 4.1 Thermo-chemical Parameters	110
Table 4.2 Input Parameters for 6C4 Motor.....	112
Table 4.3 Input Parameters for Booster SRM.....	125
Table A.1 Format for Data Set 2411.....	153
Table A.2 Format for Data Set 2412.....	153
Table A.3 Colour Codes for Boundary Condition Definitions in I-DEAS.....	154
Table A.4 Parameters for Fast Marching Input File	156
Table A.5 Output files of Fast Marching Program	159

LIST OF FIGURES

Figure 1.1 Main Parts of a Solid Propellant Rocket Motor	2
Figure 1.2 Interface Modeling Techniques	4
Figure 2.1 Moving Interface with Speed ϕ	15
Figure 2.2 Parametric Representation of a Curve	17
Figure 2.3 Effect of Variation	18
Figure 2.4 Entropy Condition	19
Figure 2.5 Huygens' Solution to Propagation	20
Figure 2.6 Wave Motion in one-dimension	20
Figure 2.7 Fast Marching Method Update Procedure	24
Figure 2.8 Directional Derivative Definitions	26
Figure 2.9 (a)Acceptable and (b)Non-Acceptable Gradient Approximations	29
Figure 2.10 Scenarios of Interface in an Element	33
Figure 2.11 Area of Interface	34
Figure 2.12 Stations along the Axis of the Motor	35
Figure 2.13 Hybrid Mesh near Interface	36
Figure 2.14 Methods of Mesh Modification	37
Figure 2.15 Hybrid Mesh Formation	39
Figure 2.16 Test Case 1 Geometry	41
Figure 2.17 Test Case 0: Results	41
Figure 2.18 Test Case 1 Geometry	42
Figure 2.19 Test Case 1: Results for Grid Size of 100 units	43
Figure 2.20 Test Case 1: Results for Grid Size of 50 units	43
Figure 2.21 Test Case 1: Results for Grid Size of 10 units	43
Figure 2.22 (a) Outward Burning Circle (b)Inward Burning Circle	45

Figure 2.23	Test Case 2: Outward Burning Results with Grid Size of 100 units...	46
Figure 2.24	Test Case 2: Outward Burning Results with Grid Size of 10 units.....	46
Figure 2.25	Test Case 2: Inward Burning Results with Grid Size of 10 units	46
Figure 2.26	(a) Outward Burning Smooth (b) Inward Burning Smooth	48
Figure 2.27	Test Case 3: Outward Regression Results with Grid Size of 50 units .	49
Figure 2.28	Test Case 3: Outward Regression Results with Grid Size of 10 units	49
Figure 2.29	Test Case 3: Inward Regression Results with Grid Size of 10 units....	49
Figure 2.30	Colliding Circles.....	51
Figure 2.31	Test Case 4: Regression Results with Grid Size of 50 units	51
Figure 2.32	Test Case 4: Regression Results with Grid Size of 10 units	52
Figure 2.33	Test Case 1: Regression Results with Grid Size of 100 units	53
Figure 2.34	Test Case 1: Regression Results with Grid Size of 25 units	54
Figure 2.35	Burn area versus Web for Test Case 1	54
Figure 2.36	Test Case 2: Outward Regression Results with Grid Size of 25 units .	55
Figure 2.37	Test Case 2 Inward Regression Results with Grid Size of 100 units...	56
Figure 2.38	Test Case 2 Inward Regression Results with Grid Size of 25 units.....	56
Figure 2.39	Burn area vs. Web for Test Case 2 –Outward.....	57
Figure 2.40	Burn area vs. Web for Test Case 2 –Inward.....	57
Figure 2.41	Test Case 3: Outward Regression Results with Grid Size of 100 unit.	58
Figure 2.42	Test Case 3: Outward Regression Results with Grid Size of 25 units .	59
Figure 2.43	Test Case 3: Inward Regression Results with Grid Size of 100 units..	59
Figure 2.44	Test Case 3: Inward Regression Results with Grid Size of 25 units....	59
Figure 2.45	Burn area vs. Web for Test Case 3 –Outward.....	60
Figure 2.46	Burn area vs. Web for Test Case 3 –Inward.....	60
Figure 2.47	Test Case 4: Regression Results with Grid Size of 25 units	61
Figure 3.1	Zero-dimensional SRM Conservative Relations.....	65
Figure 3.2	Stations for One-dimensional Solver	70
Figure 3.3	Structure of Riemann Problem for x -split Euler Equations.....	85
Figure 3.4	Cube Test Case Geometry	95
Figure 3.5	Cube Test Case Grid.....	96
Figure 3.6	Cube Test Case (10x10x10)– Mach Contours	97
Figure 3.7	Cube Test Case (10x10x10)– Pressure at Symmetry Axis.....	97

Figure 3.8	Cube Test Case (20x20x20) – Mach Contours	98
Figure 3.9	Cube Test Case (20x20x20) – Pressure at Symmetry Axis.....	98
Figure 3.10	Residual for Cube Test Case (20x20x20) with Van Leer’s Method ...	99
Figure 3.11	Residual for Test Case (20x20x20) with Roe’s Method	99
Figure 3.12	Geometry of Test Case 1	100
Figure 3.13	Two-dimensional Grid of Test Case 1	101
Figure 3.14	Three-dimensional Grid of Test Case 1	101
Figure 3.15	T-108 Test Case 1A- Mach Contours.....	102
Figure 3.16	T-108 Test Case 1A - Static Pressure.....	102
Figure 3.17	Residual for T-108 Test Case 1A with Van Leer’s Method.....	103
Figure 3.18	Residual for T-108 Test Case 1A with Roe’s Method	103
Figure 3.19	T-108 Test Case 2 – Grid Representation	105
Figure 3.20	T-108 Test Case 2 – Mach Contours.....	105
Figure 3.21	T-108 Test Case 2 – Pressure along Symmetry Axis.....	106
Figure 3.22	Residual for T-108 Test Case 2 – Roe’s Method	106
Figure 4.1	Locating Node Position.....	108
Figure 4.2	Segregated Interface Capturing	109
Figure 4.3	Coupled Interface Solution on Flow Domain.....	109
Figure 4.4	6C4 Test Motor	111
Figure 4.5	6C4 Flow Mesh	112
Figure 4.6	6C4 Mach Contours at 0.2 s.	113
Figure 4.7	6C4 Pressure Along the x-axis at 0.2 s.....	113
Figure 4.8	6C4 Mass Flow Rate Convergence at 0.2 s.....	114
Figure 4.9	6C4 Convergence History at 0.2 s.....	114
Figure 4.10	6C4 Mach Contours at 0.6 s.	115
Figure 4.11	6C4 Pressure Along the x-axis at 0.6 s.....	115
Figure 4.12	6C4 Mass Flow Rate Convergence at 0.6 s.....	116
Figure 4.13	6C4 Convergence History at 0.6 s.....	116
Figure 4.14	6C4 Mach Contours at 1.0 s.	117
Figure 4.15	6C4 Pressure Along the x-axis at 1.0 s.....	117
Figure 4.16	6C4 Mass Flow Rate Convergence at 1.0 s.....	118
Figure 4.17	6C4 Convergence History at 1.0 s.....	118

Figure 4.18 6C4 Mach Contours at 1.4 s.....	119
Figure 4.19 6C4 Pressure Along the x-axis at 1.4 s.....	119
Figure 4.20 6C4 Mass Flow Rate Convergence at 1.4 s.....	120
Figure 4.21 6C4 Convergence History at 1.4 s.....	120
Figure 4.22 6C4 Head End Pressure versus Time Prediction.....	121
Figure 4.23 6C4 Thrust versus Time Prediction.....	121
Figure 4.24 Effect of Mesh Intensity at the Throat.....	122
Figure 4.25 Geometry of Space Launcher Rocket Motor.....	124
Figure 4.26 Booster SRM Flow Mesh.....	125
Figure 4.27 Mach Contours for Booster at 0.5 s.....	126
Figure 4.28 Booster Pressure along the Axis at 0.5 s.....	126
Figure 4.29 Mach Contours for Booster at 18.18 s.....	127
Figure 4.30 Booster Pressure along the Axis at 18.18 s.....	127
Figure 4.31 Mach Contours for Booster at 24.96 s.....	128
Figure 4.32 Booster Pressure along the Axis at 24.96 s.....	128
Figure 4.33 Mach Contours for Booster at 28.86 s.....	129
Figure 4.34 Booster Pressure along the Axis at 28.86 s.....	129
Figure 4.35 Booster Pressure – Time History Prediction.....	130
Figure 4.36 Booster Thrust – Time History Prediction.....	130
Figure 4.37 Location of Air Void for Test Case 3-A.....	132
Figure 4.38 Location of Air Void for Test Case 3-B.....	132
Figure 4.39 Grid for Test Case 3.....	132
Figure 4.40 Burn Area Comparison for Voids in Grain.....	133
Figure 4.41 Comparison of Pressure for Test Case 3.....	133
Figure 4.42 Comparison of Thrust for Test Case 3.....	134
Figure 4.43 Burn Back Results for Test Case 3-A.....	134
Figure 4.44 Burn Back Results for Test Case 3-B.....	134
Figure 4.45 Hump Parameter as a Function of the Radial Direction.....	136
Figure 4.46 Hump Parameter as a Function of the Axial Direction.....	136
Figure 4.47 6C4 Pressure Time History with Hump Effect.....	137
Figure 4.48 6C4 Thrust Time History with Hump Effect.....	138
Figure A.1 Data Set Example for UNV Format.....	152

Figure A.2 Example HA2 File.....	155
Figure A.3 Fast Marching Code Input File.....	158
Figure B.1 6C4 Mach Contours at 0.2 s	160
Figure B.2 6C4 Mach Contours at 0.6 s.	160
Figure B.3 6C4 Mach Contours at 1.0 s	161
Figure B.4 6C4 Mach Contours at 1.4 s	161
Figure B.5 Booster Mach Contours at 0.5 s.....	161
Figure B.6 Booster Mach Contours at 18.18 s.....	162
Figure B.7 Booster Mach Contours at 24.96 s.....	162
Figure B.8 Booster Mach Contours at 28.86 s.....	162
Figure B.9 Single Void in 6C4 Grain	163
Figure B.10 3 Voids in 6C4 Grain.....	163

LIST OF SYMBOLS

A	surface area
\mathbf{A}	Jacobian matrix
$\tilde{\mathbf{A}}$	Jacobian matrix with primitive variables
\mathbf{B}	Jacobian matrix
$\tilde{\mathbf{B}}$	Jacobian matrix with primitive variables
c	speed of sound
C^*	characteristic speed of propellant
\mathbf{C}	Jacobian matrix
$\tilde{\mathbf{C}}$	Jacobian matrix with primitive variables
C_p	specific heat at constant pressure
C_f	thrust coefficient
C_d	divergence loss coefficient
e	internal energy per unit volume
e_0	total internal energy
G	mass flux
\vec{H}	flux vector with components F, G, K
h_o	total enthalpy
\vec{i}	unit normal vector in x-direction
\vec{j}	unit normal vector in y-direction
\vec{k}	unit normal vector in z-direction
\dot{m}	mass flow rate
M	Mach number
N	number of nodes
\vec{n}	normal vector
n	pressure exponent
p	pressure
P	conservative matrix in primitive variables
R	gas constant
r	radius
r_b	propellant burn rate
t	time
T	interface capturing time parameter, temperature
u	x-component of velocity vector
\vec{U}	unit vector matrix

Q	column vector of conservative variables
v	y-component of velocity vector
V	velocity vector magnitude
w	z-component of velocity vector
x,y,z	Cartesian space coordinates

Other Symbols:

ρ	density
γ	specific heat ratio
∂	partial derivative operator
Ω	control volume
Γ	closed smooth curve
κ	curvature of curve
λ	eigenvalues
Λ	diagonal matrix of eigenvalues
ϕ	speed of interface
ψ	level set function
Θ	Hamiltonian function
∇	directional derivative matrix

Superscripts:

L	left component
n	quantity at discrete time level n
R	right component
*	non-dimensional quantity

Subscripts:

ch	chamber
cal	calculated value
g	generated
i	element number
j	node number
inj	injection property
n	normal component
0	stagnation state
per	perimeter
ref	reference
$real$	real value
th	throat
x	x component
y	y component
z	z component

CHAPTER 1

INTRODUCTION

Solid propellant rocket motor (SRM) is widely used in military and civil applications where high thrust is required in order to transfer payloads. The solid propellant is a combination of fuel and oxidizer, which enables the motor to operate in all environmental conditions. Due to its relatively simple design, when compared to other types of rocket motors, the operational reliability of a SRM is very high.

The SRM consists of five main parts, which are given as follows (Figure 1.1):

1. Grain: The solid propellant, which generates high temperature and high-pressure gas upon ignition.
2. Igniter: The combustion of the grain is initiated by the igniter.
3. Motor Case: The high pressure builds up in the motor case, which sustains the combustion, and eventually the thrust is created by this high pressure.
4. Nozzle: The high-pressure gas products are expanded to ambient pressure and thrust is efficiently obtained.
5. Insulation: The motor case is protected from hot gases by insulation which has a low thermal conductivity.

The performance characteristics of a SRM, which can be simplified to the thrust-time curve of the motor, depend on the geometry of the grain, propellant properties and flow variables. Any design or performance prediction effort, couple all of the above mentioned topics and the applied science devoted to these problems is called internal ballistics [1].

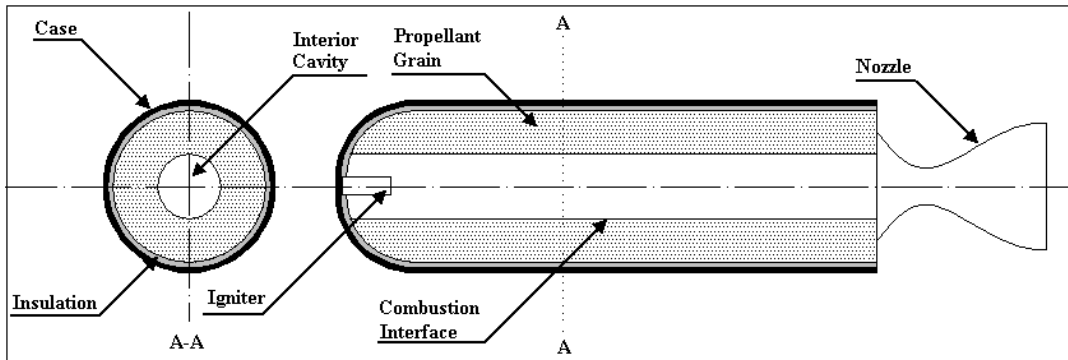


Figure 1.1 Main Parts of a Solid Propellant Rocket Motor

Provided that propellant properties and geometry are known, the performance characteristic is obtained by the solution of the flow internally. The flow solution should consider subsonic to supersonic compressible turbulent flow, combustion, reactive multi-phase flow, and acoustic properties in order to fully capture the nature of the operation of the SRM.

The objective of the solid propellant grain designer is to provide rocket motor with a propellant grain geometry that will evolve combustion products consistent with the thrust-time schedule required for the mission [2].

In a SRM during the operation, although nozzle and motor case insulation change their shape, all the parts can be considered stationary except for the grain. The grain changes its shape as the burning surface at each point recedes in a direction normal to the surface at that point. Thus the control volume of flow expands as the combustion process continues. Accurate prediction of internal flow properties depends on, determination of propellant burning surface or flow control volume

[3]. The flow control volume can be found by tracking or capturing the interface between the gas side and the propellant side.

The performance prediction requires a control volume in which the flow solutions will be carried out. The analysis can be performed in two- and three-dimensions. The two dimensional cases consist of grains with constant port area, axi-symmetric grains and end-burning grains. Analytical calculations give accurate results for these types of grain geometry. However, most of the high performance SRM requires much more complex grain configurations, which have burning surfaces in all three dimensions and numerical solution techniques are required.

1.1 LITERATURE SURVEY

The solution techniques applied to moving boundary problems are suitable for the investigation of SRM and a useful classification is given by Smolianski [4], which considers the solution process under three main topics:

1. Flow Modeling: is used for solving the physics of the problem to obtain flow properties. Methods fall into *Eulerian*, *Lagrangian* or *mixed Eulerian-Lagrangian* category. Eulerian methods use a fixed frame of reference and the fluid travels between computational cells. In Lagrangian methods, the cells carry the same fluid elements. Thus the coordinate system moves with the fluid. The mixed methods carry characteristics of both methods.
2. Interface Modeling: is used for finding the location and shape of interface. Methods fall into *interface tracking* and *interface capturing* category. In interface tracking methods, the moving front is explicitly tracked by following the nodes along the path of each fluid particle. In interface capturing methods the grid is kept stationary and the fluid particles on the front are captured. Both sides of the domain are computed. A representative comparison is given in Figure 1.2.

3. Flow-Interface Coupling: The flow and interface solutions can be coupled by segregated or integrated methods. In the former method as the calculations for the flow proceeds, the interface is kept frozen and a new position of interface is found while the flow properties are kept frozen. In integrated methods the flow and interface are solved simultaneously.

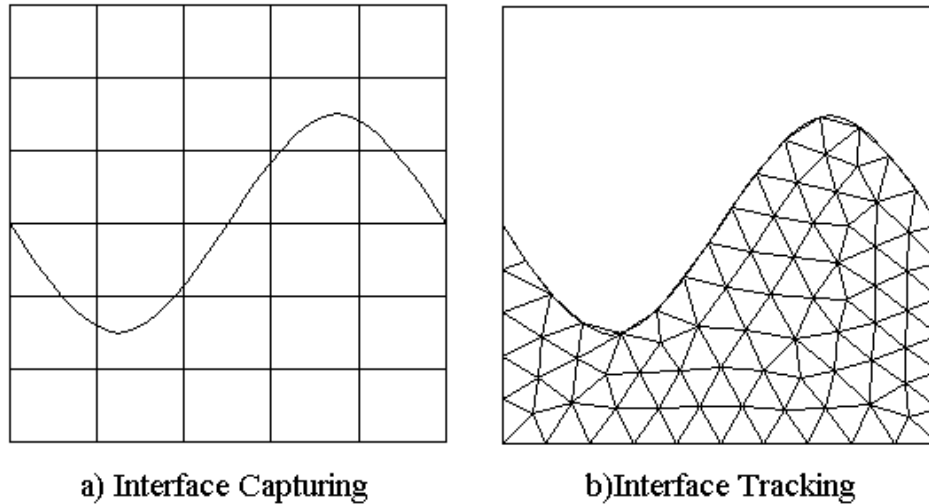


Figure 1.2 Interface Modeling Techniques

Considering the characteristics of the two approaches used for interface motion, the following comparison is done by Shyy et al [5].

- Interface Definition: The interface tracking methods define the interface as a discontinuity and explicitly track its evolution. All the information regarding the interface is already known, whereas in interface capturing methods even to find the interface location may need complicated procedures.
- Interfacial Boundary Conditions: In the interface tracking methods, boundary conditions can be applied at the exact location of the interface since the interface position is known at each instant. In the interface capturing methods, the boundary conditions are manipulated to appear in

the governing transport equations. This may lead to the smearing of the boundary information and again complicated procedures are required in order to minimize diffusion of the interface.

- Discretization of the Domain: In the interface tracking methods, the grid adapts to the interface and hence grid re-arrangement and motion terms have to be incorporated. When the interface begins to distort, the grid needs to be regenerated each time. The resulting grid may be skewed and unevenly distributed where several procedures are required in order to preserve the connectivity and intensity. The interface capturing methods have an advantage in this regard since the computations are performed on fixed grid; hence the need for grid rearrangement is eliminated. However when the interface is arbitrarily shaped, improved resolution in desired regions is difficult to obtain, unless complicated local refinements are adopted.
- Movement and Deformation of the Interface: Interface tracking methods have so far experienced difficulty in handling topological changes, mainly due to the breakdown of the grid arrangement and the need for redistribution of field information in the vicinity of the interface for unstructured grid methods. In interface capturing methods merged and separated grids are taken care of automatically.

The advantages of these methods are dependent on the application that they are utilized. For a general review of these methods one may refer to Shyy [5], Hou [6] and Sethian [7].

A clear classification of methods used in literature is difficult to obtain since each technique is a mixture of several other methods. In this report, the classification is made according to the type of method by which the interface motion is solved.

1.1.1 Interface Tracking Methods

The most known interface tracking methods are *Lagrangian Method*, *Smoothed Particle Hydrodynamics (SPH) Method*, *Arbitrary Lagrangian-Eulerian (ALE) Method* and *Front Tracking Method*. The related references are given below.

Okamoto [8] analyzed the sloshing problem in closed containers with a Lagrangian finite element method. The node elements follow the fluid particles therefore new node positions are calculated at each time interval by the flow solver for the whole domain. This method is suitable if the deformation in the domain is limited so that the connectivity of elements does not change. For large deformations, the methods using Lagrangian flow solvers require a re-meshing and re-zoning procedure proposed by Fyfe [9].

Christodoulou [10] uses boundary fitted finite element discretization for free surface flows. An integrated solution method is used, where structured elliptic mesh generation and flow equations are coupled and solved simultaneously. This method produces accurate results if the boundary deformation is small. Camacho [11] used unstructured boundary-conforming curvilinear grids for ductile penetration where large deformations occur. The mesh is adapted or generated automatically at each time step according to the level of deformation. By this way accuracy increases at the cost of mesh generation.

Hirt et al [12] proposed Arbitrary Lagrangian-Eulerian (ALE) method which reconstructs the mesh at locations near the interface with respect to the fluid in a prescribed manner, the mesh is kept fixed at all other locations. The interface is tracked by Lagrangian motion of vertices initially at the front.

Cristini et al. [13] presents an algorithm for adaptive restructuring of meshes on evolving surfaces. The resolution of the relevant local length scale is maintained

everywhere with prescribed accuracy through the minimization of an appropriate mesh energy function by a sequence of local restructuring operations.

Monaghan [14] applies Smoothed Particle Hydrodynamics (SPH) method to incompressible fluid for free surface motion. This method does not require a grid since the properties are obtained by interpolation from the set of particles, which are disorderly scattered in the domain. The flow solutions are carried in purely Lagrangian manner and several phases or species can be solved simultaneously. Morris [15] extended SPH method to incorporate surface tension effects by coupling continuum surface force method and removing the instabilities created at the interface particles.

In front tracking methods the flow solutions are carried on a fixed grid where the interface is tracked separately. Hyman [16] gives an overview of early front tracking methods on fixed mesh.

The boundary conditions may be applied to inner cells where all domains are solved simultaneously in *immersed boundary method*. Peskin [17] proposed momentum forcing where interface is conserved and Goldstein [18] proposed feedback forcing in order to represent solid bodies. Unverdi [19] and Juric [20] applied this technique to several multi-fluid problems. Kim [21] has increased the stability of immersed boundary technique by applying a second order interpolation scheme for evaluating the momentum forcing on the body surface and inside the body.

Udaykumar [22] [23] uses circular arcs in order to obtain shape information such as normal and curvature. The Eulerian grid aids the merger/ breakup procedures at the interface. Both sides of the domain require refining or coarsening of cells in orthogonal domain. Tryggvason et al [24] implemented front tracking method with solutions of Navier-Stokes equations on a fixed mesh for free surface flow solutions.

Glimm et al [25], [26] uses a front tracking algorithm where the reconstruction of the interface is performed by initially projecting the front on to a fixed mesh and then reconstructing the interface by using this projected information. By this way, the even distribution of surface mesh is guaranteed. Torres [27] uses a point-set method where the connectivity of points by using an indicator function is not required. Shin [28] et al. uses a *level-contour reconstruction* method, which allows one to redistribute or delete interfacial points, based on a fixed mesh behind the tracking particles.

Popinet [29] uses a front-tracking algorithm for the solution of the two-dimensional incompressible Navier Stokes equations with interface forces. The interface is represented using an ordered list of marker points. A parametric representation of the interface is obtained using polynomials to connect marker points, which require a re-distribution step due stretching and compression of the interface. The volume fraction can be calculated by using the parametric form of the interface and surface forces are calculated on the interface after the flow solutions are performed.

Cruchaga [30] proposes a Lagrangian interface technique defined in fixed mesh finite element formulation. Johnson [31] used overset meshes for tracking interface on a fixed Cartesian mesh. Yang [32] proposed a cut cell method in Cartesian grid where the cells are split by the interface and boundary conditions applied at each cut surface.

1.1.2 Interface Capturing Methods

The most known interface capturing methods are *Marker and Cell (MAC) Method*, *Volume of Fluid (VOF) Method*, *Level Set Method* and *Phase Field Method*. The related references are given below.

The *marker* is a virtual particle to represent liquid regions. They have no quantifiable properties such as mass or energy, and serve only as a tool emphasizing the fluid region. First the governing equations are solved on the fixed

grid containing the markers and the velocity field is calculated. Next each particle is moved according to the velocity field. The interface of the free surface is observed through marker distribution.

Marker-and-cell method is an extended version of marker particle method based on finite difference methods, which requires orthogonal grid domain. Nakayama [33] proposed a new technique by combining marker particle method with finite element methods, which generalize the treatment of curved wall boundaries and the viscous stresses.

York et al [34] proposes *material-point method*, which has evolved from particle-in-cell method developed by Brackbill [35], in which material points have mass and velocity and carry stress, strain. The interaction of these points and momentum equations are solved on a fixed mesh at the background. Brackbill [36] proposed a continuum method for modeling surface tension in marker and cell methods.

Dai [37] et al solves Navier-Stokes equations for the liquid phase on a deforming unstructured mesh. The technique tracks the boundary precisely, similar to marker and cell method. However the adaptive mesh follows the interface. Furthermore, this new method avoids the surface reconstruction required in volume of fluid methods. By locally fitting the free surface to a parabola when evaluating curvature, problems with numerical noise in the solution are avoided. A new time step criterion is introduced based on free surface numerical stability.

In free surface flows where coalescence and detachment of interface occurs, a deforming mesh may not be used. Hirt [38] proposed volume of fluids (VOF) method which uses a fixed grid where each cell is assigned a value based on the fraction of that cell containing the material inside. The cell fractions are used to characterize the interface as the flow solutions continue.

VOF is diffusive on the sharp interface locations therefore several methods are devised to reconstruct an approximation to the interface accurately. Ashgriz[39]

proposed flux line-segment (FLAIR) technique, Rudman [40] proposed flux-corrected transport (FCT) technique, Puckett [41] proposed least squares volume-of-fluid interface reconstruction (LVIRA) technique, Rider [42] proposed piecewise linear interface calculation (PLIC) technique, Aliabadi [43] proposed stabilized-finite-element/interface-capturing (SFE/IC) technique and Scardovelli [44] proposed Eulerian implicit- Lagrangian explicit (EI-LE) technique. Maronnier [45] proposed an implicit splitting algorithm in order to decouple advection and diffusion phenomena. Advection is solved using a fixed structured grid, where diffusion is solved on unstructured mesh. Smaller grids may be used for the diffusion solutions, which decrease smearing of the interface, and the algorithm becomes unconditionally stable.

Kelecy [46] proposed a surface capturing method, which uses artificial compressibility for phase definitions. Both sides of the domain are solved simultaneously where the interface is captured by the discontinuity in density.

In contrast to discontinuous interface definition, a continuous interface may be defined as a zero level set of a continuous function. Osher [47] [48] and Sethian [7] [49] described the interface as Hamilton-Jacobi type equations. The equations can be in the form of either boundary value problem, which are solved by fast marching methods, or initial value problem which are solved by narrow band level set method. Level set methods are suitable for moving boundaries which move under the speed with varying sign whereas the fast marching method can only deal with positive speed problems.

The level set formulation is extended to triangular unstructured mesh by Barth [50] and Sethian [51]. If the velocity of the interface is only defined on the front itself, in order to conserve the continuous nature of the solution an auxiliary step is required. Extended velocity technique, which is developed by Adalsteinsson [52], finds a velocity field dependent on the features of the front itself by extrapolation. The level set method has been applied to two-phase flows [53], diffusion [54], fatigue crack propagation [55], geodesic paths [56] and several other physical

phenomena. The accuracy required at the interface may be increased by mesh adaptation around the interface only. Sethian [7] proposed techniques for triangular mesh and Kohno [57] proposed techniques for orthogonal mesh adaptation.

Kobayashi [58] proposed *phase-field method*, which defines a phase function for the definition of interface. The phase function has a value of 0 at fluid regions and 1 at solid regions. The sharp change of the values in between 0 and 1 defines the interface thickness. The shape of the phase function is conserved through the solution of free energy equation. For recent developments in phase field technique one may refer to Kobayashi [59] and Shyy [5].

1.1.3 Rocket Performance Prediction

The motion of the interface depends on several factors. The geometry of the surface is not the only information required for the motion. It is required to obtain the speed by which the interface moves where the propellant and flow properties play a huge role in its definition. Chemical composition of the propellant, manufacturing processes, initial temperature of propellant and many more parameters should be taken into consideration [60].

Advanced models using both finite element and finite volume approaches have been developed by the solid rocket propulsion contractors. It is difficult for outside organizations to gain detailed information regarding these models due to their proprietary nature [61]. Therefore, information on application of moving interface to SRM and on coupled performance prediction methods is limited

Stone [62] describes mathematical equations that relate solid propellant grain geometry to the progressivity ratio and burn area in two dimensions. These equations are solved by computers of the time and burn area vs. web data is presented in tabulated form for the use of designers. Star and Wagon Wheel designs were considered in the investigation. Up until to this investigation,

empirical correlations were used by the designers. Therefore new geometric grain designs were not feasible.

Zarda [63] describes a solid modeling program, which is similar to solid modeling tools. This program generates the propellant grain by using primitives such as sphere, cylinder and rectangular blocks. The basic operations such as join, cut and intersection are used to form complicated geometry. By changing the dimension of primitives and basic operation constraints, each burning interval can be obtained. Each burn instant is another solid model, which can be used to find inertial and surface properties. This method is widely used in motor design. The procedure can be automated and results are most reliable. However this procedure can not be coupled with variable burn rate through the domain since the primitives will deform after a while.

Hejl [61] uses a marker and cell method in order to follow the interface. At sharp corners, the model refines the mesh and continues to operate with less error. This method can handle variations in burn rate however following each node in the domain is tedious and time consuming for 3-dimensions. These types of models best suited for two-dimensional cases.

Saintout [64] uses a body fitted coordinate system in order to solve the moving interface. Equations similar to Hamilton-Jacobi equations, which describe the surface growing normal to itself according to the local burning rate, are then solved on the mesh at each step of the regression. The update of the mesh is as follows: surface mesh is generated and curvilinear coordinate system is created. Each node is displaced along its normal vector proportional to its burning rate: the junctions between this updated mesh and the restricted surfaces, and the crossings between parts of the mesh are controlled. The mesh which is corrected becomes the new burn surface area. This method is suitable for variable burn rate. However the mesh correction procedure is tedious, mesh smoothing and mesh adaptation is required due to nature of the solution method.

The Solid Performance Program (SPP) has become the standard reference computer program in United States for predicting the delivered performance of solid propellant rocket motors. Dunn [65] describes the grain design module of this program. The grain geometry is described by the frustum of a right circular cone, a right circular cylinder, a right triangular prism, a sphere and a torus. Burning is assumed to be normal to the surface and the volume of propellant is calculated by integrating the propellant area along the grain axis. The propellant burning surface is obtained from the derivative of the volume with respect to the burn distance. The calculations are general in that any grain can be evaluated if the initial grain geometry is described by using any combination of the five basic figures. The ballistic module is supplied with burning surface area, cross sectional area and perimeter from the grain design module. The ballistic module solves the one-dimensional flow problem with energy and mass addition. The analysis employs a pseudo-transient treatment of the start-up problem and quasi-steady state solution thereafter. The SPP code is a preliminary design and analysis code, which is fast and easy to use.

Le Breton [60] describes a method to predict the performance of large-scale solid rocket motors by using the results of the 3-dimensional surface burnback results coupled with a simple ballistic model. The problem that they considered is the effect of filling of the motor chamber with batches of propellant with different burning rates. Their 3-D burnback code is the same one, which is developed by Saintout [64].

Alavilli et al. [66] adopted a finite volume, multi-block, structured mesh approach in obtaining the 3-dimensional flow simulation in SRM. The N-S equations are solved on dynamic meshes whose boundaries adapt to conform to the propellant surface that deforms due to the loads imposed on it and also regresses due to burning. Re-meshing is handled by the solver, which is performed on parallel computers.

1.2 PURPOSE OF THE STUDY

The purpose of this study is to describe and develop a methodology, which combines interface-capturing methods with flow solution techniques in order to analyze the Solid Propellant Rocket Motor performance.

Given the propellant properties and geometry of all the parts, the aim is to solve the flow internally with moving grain boundary. The motion of the propellant during the combustion is followed in three-dimensions by an interface capturing algorithm which is called “Fast Marching Method”. The domain in which the flow solutions will be carried out is obtained by a cut cell method on unstructured domain. A Finite Volume solver is used to obtain flow properties in the fluid domain in a segregated fashion.

The results obtained from the methodology, which is described in this thesis, are compared with actual rocket firing data and with results of previously validated performance prediction codes. The data and information supplied in this thesis report covers only the unclassified part of the work achieved.

1.3 CONTENTS OF THE THESIS REPORT

In Chapter 2, the theory of moving front and numerical solution methods are given. The Fast Marching Algorithm is described for the efficient solution of the interface on structured and unstructured mesh. The interface capturing methodology is described so that the control volumes in which the flow solvers will operate are obtained. In Chapter 3, the flow solvers for the performance prediction are described for zero-, one- and three-dimensional domains. Test cases for validation of the flow solvers are given. In Chapter 4, the coupled test case results are compared with experimental data and validated calculation results. In Chapter 5, a conclusion and suggestions for future applications are presented.

CHAPTER 2

RETARDING WALLS

The retarding walls are considered as the interfaces, which separate two different phases. The Fast Marching is used as the interface capturing method. The aim in interface capturing methods is to compute the location of this interface, which is moving with speed $\phi(x, t)$ in the local normal direction as shown in Figure 2.1.

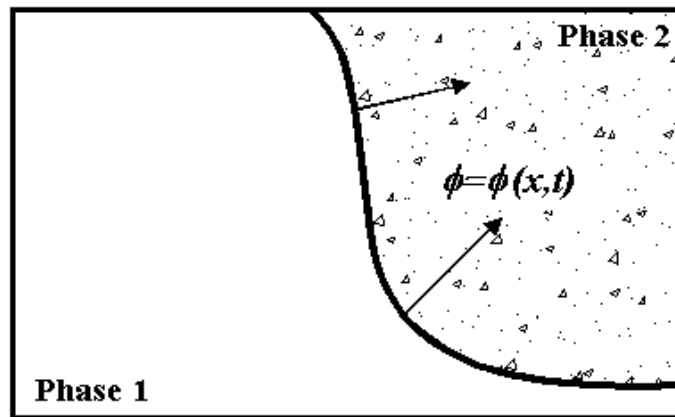


Figure 2.1 Moving Interface with Speed ϕ

2.1 PROPERTIES OF CURVES

In two-dimensions the interface is a curve and in three-dimensions the interface becomes a surface. In this section the general properties of curves are given for a complete discussion of the problem.

2.1.1 Velocity of Interface

The rate of interface regression depends on:

- i. Geometric Parameters: The surface normal and curvature of interface,
- ii. Physical Parameters: The physical and chemical properties of the solid region,
- iii. Flow Parameters: Pressure, temperature and acceleration effects.

Calculation of interface regression rate is described in Chapter 3, which is independent of the interface modeling due to the segregated nature of the solution technique.

2.1.2 Parametric Representation of Curve Evolution

Assume that Γ is a simple, smooth, closed curve. Let $x(s, t)$ is the parametric representation of Γ , $n(s, t)$ is the parameterization of outward normal, $\kappa(s, t)$ be the parameterization of curvature and $\phi(\kappa)$ is the speed of interface as a function of curvature. The equation of motion can be written as [7] :

$$\frac{dx}{dt} = \phi(\kappa) \frac{(y_s)}{(x_s^2 + y_s^2)^{1/2}} \quad (2.1)$$

$$\frac{dy}{dt} = \phi(\kappa) \frac{(-x_s)}{(x_s^2 + y_s^2)^{1/2}} \quad (2.2)$$

where $\kappa = \frac{y_{ss}x_s - x_{ss}y_s}{(x_s^2 + y_s^2)^{3/2}}$. This is the Lagrangian representation of motion, which is shown in Figure 2.2.

The front tracking methods use Equations (2.1) and (2.2) in order to calculate the position of marker points for the next time interval. By using these equations, some important parameters and characteristics of curves may be obtained.

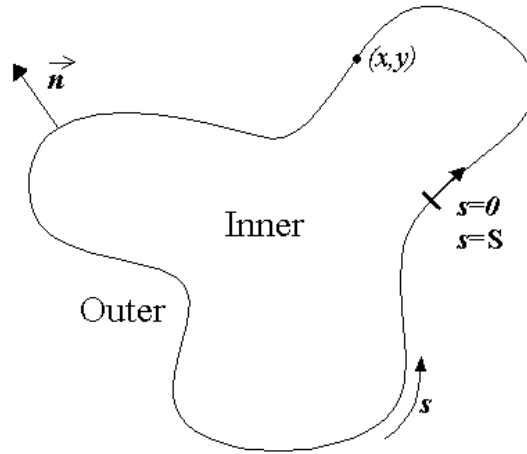


Figure 2.2 Parametric Representation of a Curve

2.1.3 Growth of Oscillations

Initial instabilities in a curve may increase thus avoiding the solution to converge. If the curve can be parametrically described, the *total variation* can be calculated as:

$$Var(t) = \int_0^S |\kappa(s, t)| g(s, t) ds \quad (2.3)$$

where $g(s, t) = (x_s^2 + y_s^2)^{1/2}$.

$Var(t)$ is a measure of the wrinkling of the curve. If the value of $Var(t)$ increases, the stability of the solution decreases. The wrinkling is shown in Figure 2.3. It has been shown by Sethian [7] that, if $\phi_\kappa \leq 0$ when $\kappa=0$, then $\frac{dVar(t)}{dt} < 0$, which suggests that the curve smoothens out and the energy of the front dissipates.

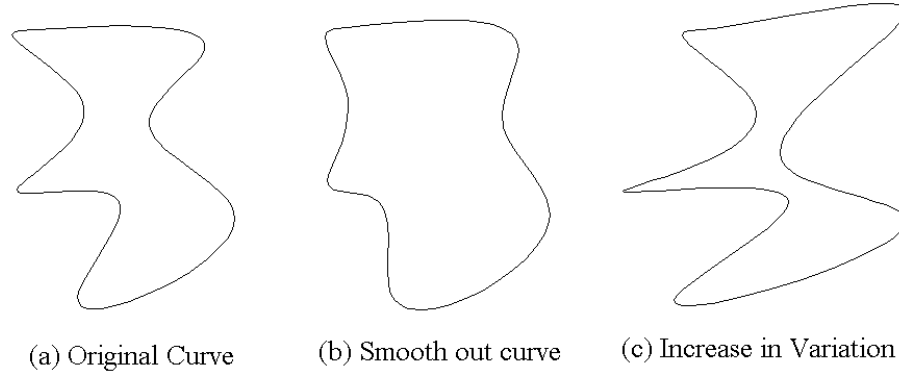


Figure 2.3 Effect of Variation

The motion of the front during the combustion process is independent of the curvature of the surface and the speed function has a positive sign at all times. Therefore the condition $\phi_\kappa \leq 0$ will be satisfied in our solutions. This suggests that our calculations will be stable at all times with decreasing variation.

2.1.4 Entropy Condition

During the motion of a curve, except for all-convex curves, the smoothness is soon lost. As can be seen in Figure 2.4, the cosine curve creates a sharp nose through its motion with constant speed. Once the corner develops, the normal can not be defined at the point of discontinuity. At this point, a weak solution is required in order to continue the solution.

One of the weak solutions is obtained by allowing a swallowtail to be generated, which permits the curve pieces coming from opposite directions to cross each other. However this solution is not applicable for our purposes.

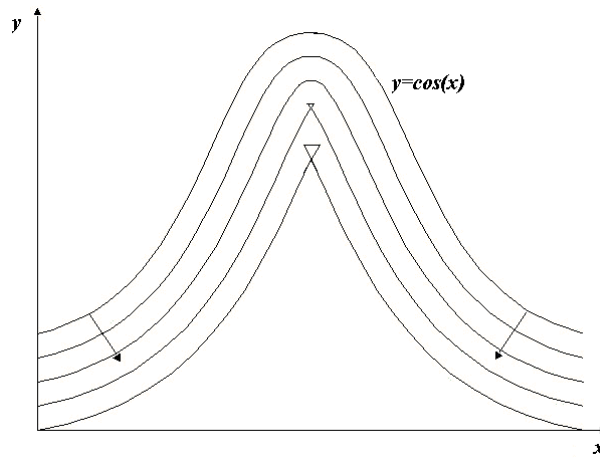


Figure 2.4 Entropy Condition

Another weak solution is obtained through Huygens' Principle, which suggests drawing of circular wave-fronts at each point on the curve with their radius proportional to the speed $\phi(x,t)$. The envelope of these wave-fronts is taken as the new position of the curve as shown in Figure 2.5. The entropy condition means that once a corner has developed, the solution is no longer reversible. The problem cannot be run backward in time without the loss of initial data. The sharp corner in Figure 2.4 can not be reversed because information is lost at the corner. A mathematical discussion can be found in reference [7].

Similar to Huygens' principle, Fast Marching Method satisfies the entropy condition. Since the grain burnback calculations does not require generation of the initial front, the method is satisfactory.

The Huygens' solution also satisfies the viscosity solution at initially sharp corners, which smoothens out at discontinuities. In the limit of this process the Eikonal solution is obtained. The numerical test cases given in Section 2.5, show both entropy and viscosity satisfying results.

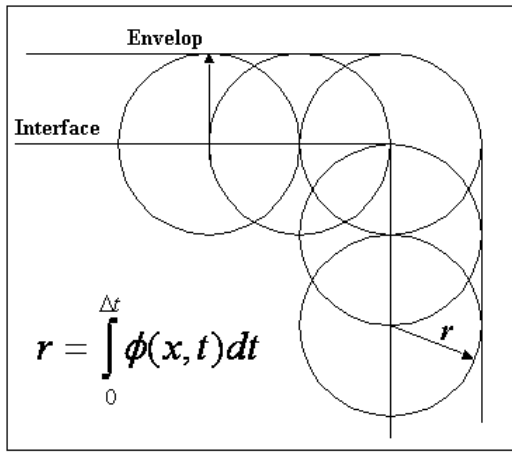


Figure 2.5 Huygens' Solution to Propagation

2.1.5 Boundary Value Formulation

If the speed of the interface is positive throughout the solution domain, a boundary value formulation can be established. In this method the time, $T(x, \phi)$, at which the interface arrives at each point in space is to be calculated. In 1-dimension the equation of motion is defined [7] as:

$$\phi \frac{dT}{dx} = 1 \quad (2.4)$$

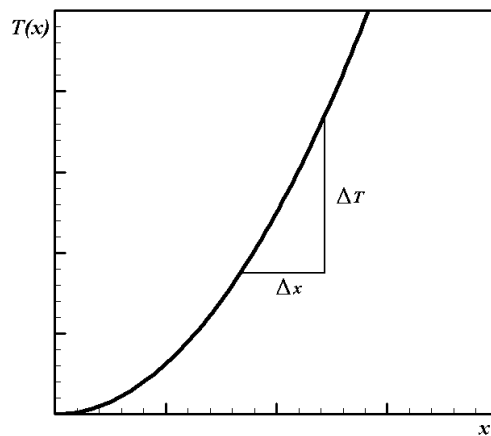


Figure 2.6 Wave Motion in one-dimension

which is similar to the motion of a wave front in one-dimension as shown in Figure 2.6. In multiple dimensions, $\vec{\nabla}T$ is orthogonal to level surfaces of T , and can be written as,

$$\phi|\vec{\nabla}T|=1 \quad (2.5)$$

with the boundary condition $T=0$ on Γ , where Γ is the initial location of interface. The type of this equation is referred to as Eikonal type [51].

2.1.6 Initial Value Formulation

If the speed by which the interface moves changes sign, initial value formulation should be adapted. The initial position of the interface is embedded as the zero level set of a higher dimensional function ψ , which transforms the governing equations into initial value problem. It is required that the level set value of a particle on the interface with path $x(t)$ must always be zero. Therefore

$$\psi(x(t),t) = 0 \quad (2.6)$$

By the chain rule,

$$\psi_t + \vec{\nabla}\psi \cdot \frac{d\vec{x}}{dt} = 0 \quad (2.7)$$

where $\vec{x}'(t) \cdot \vec{n} = \phi$ and $\vec{n} = \vec{\nabla}\psi / |\vec{\nabla}\psi|$. By inserting these relations into equation (2.7) the following evolution equation is obtained for ψ .

$$\psi_t + \phi|\vec{\nabla}\psi| = 0 \quad (2.8)$$

given $\psi(x, t = 0)$ as the initial condition. In order to obtain the position of the interface, the zero level set of the Equation (2.8) has to be solved.

2.1.7 Advantages of Both Formulations

The obvious advantages of these formulations, apart from both of them using Eulerian view are as follows:

- i. Both of the formulations are unchanged in all dimensions.
- ii. Changes in the topology are handled naturally. Breaking up or merging of curves and surfaces need no special attention.
- iii. Both formulations can be approximated by computational schemes, which use numerical solutions of hyperbolic conservation laws.
- iv. Both methods are made efficient by using adaptive methods.
- v. Both methods guarantee a unique, entropy satisfying weak solution.

2.1.8 Comparison of Formulations

Comparison between the two methods can be given as:

- i. Initial value formulation allows both positive and negative speed functions, whereas boundary value formulation allows only positive speed because it requires a single crossing time. The solid rocket grain burnback deals with positive speed functions only and therefore both methods can be applied.
- ii. On the other hand while handling positive speed function which depends on position, the boundary value formulation is advantageous because it requires no CFL condition due to the lack of time step. Furthermore the boundary value formulation can be made extremely computationally efficient.

Therefore for the retarding wall calculations described in this thesis, boundary value formulation is selected.

2.2 NUMERICAL ALGORITHMS FOR BOUNDARY VALUE PROBLEM

The general Hamilton-Jacobi equation is

$$\alpha T_t + \Theta(T_x, T_y, T_z, x, y, z) = 0 \quad (2.9)$$

The function Θ is the ‘‘Hamiltonian’’ and for the boundary value problem

$$\Theta = \phi \sqrt{T_x^2 + T_y^2 + T_z^2} - 1 \quad (2.10)$$

and α is zero.

A first order upwind finite difference scheme for the Hamiltonian is given by Sethian [7] as:

$$\left[\begin{array}{l} \max(D^{-x}_{ijk} T, 0)^2 + \min(D^{+x}_{ijk} T, 0)^2 \\ + \max(D^{-y}_{ijk} T, 0)^2 + \min(D^{+y}_{ijk} T, 0)^2 \\ + \max(D^{-z}_{ijk} T, 0)^2 + \min(D^{+z}_{ijk} T, 0)^2 \end{array} \right]^{1/2} = \frac{1}{\phi_{ijk}} \quad (2.11)$$

where D^{-x} is the backward difference and D^{+x} is the forward difference in x -direction and are given as

$$D^{+x} T = \frac{T(x + \Delta x) - T(x)}{\Delta x} \quad (2.12)$$

$$D^{-x}T = \frac{T(x) - T(x - \Delta x)}{\Delta x} \quad (2.13)$$

This is the scheme described by Sethian [7] for the solution of Eikonal equations.

2.3 FAST MARCHING METHOD

It is required to solve quadratic Equation (2.11) iteratively through the whole domain, which takes up $O(N^3)$ labor for one iteration in three dimensions where N is the number of nodes. Assuming that it will take another N iterations to reach a solution the iteration count will be close to $O(N^4)$.

Equation (2.11) constructs the solution using only upwind values, which means that the update occurs from smaller values of T . The information is propagated in one direction. The Fast Marching Method (FMM) exploits this property.

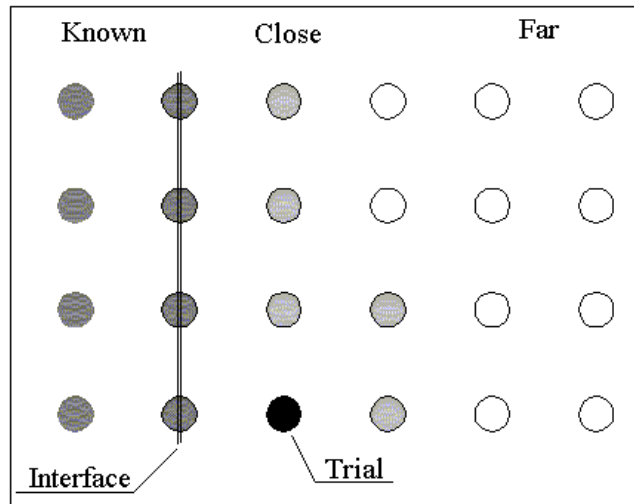


Figure 2.7 Fast Marching Method Update Procedure

The points in a thin zone around the interface are selected and the solution marches in this zone only where the downwind points are not solved until the interface reaches there.

2.3.1 The Update Procedure

The nodes where the boundary lies are tagged as *known*. The nodes, which are one grid point away as *close* points. Finally tag all the nodes which are in the downwind as *far* points since they will not be used for the recent iteration.

The solution is initialized by obtaining a T value for the *close* set points. The information comes from the interface initial position. Then the solution algorithm proceeds as follows:

- i. Select the *trial* point with the smallest value of T .
- ii. Add the *trial* point to the *known* set and all the neighbors of the *trial* point, which are not *known* to *close* points set.
- iii. Calculate values of T at each neighbor, which are in the *close* set by using Equation (2.11).
- iv. Return to the top of the loop.

In order to find the trial point with the smallest value of T , a sorting algorithm is used which requires $O(\log N)$ labor. Since each node is visited once, the total operation count is reduced to $O(N^3 \log N)$ for the three-dimensional case.

2.3.2 Approximation on Triangulated Domains

Previous definitions use fixed and logically rectangular meshes. Such techniques have high accuracy and programming ease. However, in our solution procedure, which will be described in this section, unstructured mesh is used due to the reasons given by Barth [50] as follows:

- i. When increased accuracy is required at certain locations, adaptation is applied which is easier to handle in unstructured meshes.

- ii. Interface fitted meshes can be applied similar to adaptation if required. This will help to define the properties easily on the interface and the solution of the flow in the gas side could be managed by regular unstructured flow solvers.
- iii. Due to the complexity of the geometry the mesh generation is easier in triangulated techniques. Nearly all geometries can be discretized by triangulation.

2.3.3 Derivative Approximations on Unstructured Mesh

The gradient is calculated as a combination of directional derivatives since there is no natural choice of the coordinate system for an unstructured mesh [51].

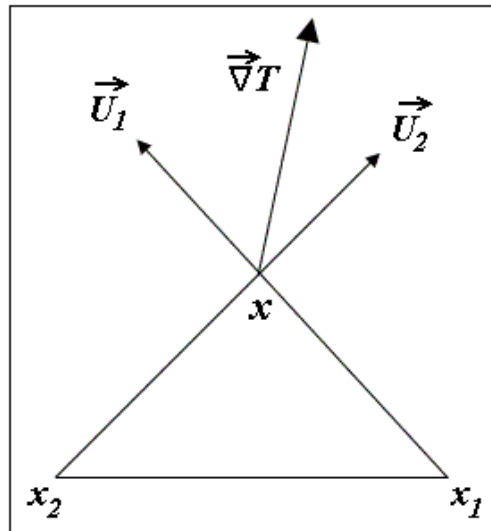


Figure 2.8 Directional Derivative Definitions

Consider the triangular mesh given in Figure 2.8, the unit vectors defining the directions are given as follows:

$$\vec{U}_i = \frac{\vec{x} - \vec{x}_i}{\|\vec{x} - \vec{x}_i\|} \quad (2.14)$$

The directional derivatives of interface arrival time in the direction of unit vectors U_1 and U_2 is be calculated as:

$$\vec{U} \cdot \vec{\nabla} T(x) = \mathbf{v}(x) \quad (2.15)$$

If Equations (2.5) and (2.15) are combined, the directional derivative for any unstructured mesh is obtained as:

$$\mathbf{v}(x)^T (\vec{U}\vec{U}^T)^{-1} \mathbf{v}(x) = \frac{1}{\phi^2(x)} \quad (2.16)$$

In order to discretize this equation, the directional derivative can be written in the linear form.

$$\mathbf{v} = \mathbf{a}T(x) + \mathbf{b} \quad (2.17)$$

where a and b are discretization constants. If Equation (2.17) is substituted into Equation (2.16) the following quadratic equation is obtained.

$$(\mathbf{a}^T (\vec{U}\vec{U}^T)^{-1} \mathbf{a})T^2 + (2\mathbf{a}^T (\vec{U}\vec{U}^T)^{-1} \mathbf{b})T + (\mathbf{b}^T (\vec{U}\vec{U}^T)^{-1} \mathbf{b}) - 1/\phi^2 = 0 \quad (2.18)$$

The solution of this quadratic equation will yield answers for time variable in the considered elements for any dimension.

In order to compute values of a and b in Equation (2.18), first or second order discretization may be used.

First order accurate solution defines directional derivative as:

$$\mathbf{v}_i = \frac{T(x) - T(x_i)}{\|\bar{x} - \bar{x}_i\|}$$

where a and b is obtained as:

$$\mathbf{a} = \frac{1}{\|\bar{x} - \bar{x}_i\|} \text{ and } \mathbf{b} = \frac{-T(x_i)}{\|\bar{x} - \bar{x}_i\|}$$

Second order accurate solution defines directional derivative as:

$$v_i = 2 \frac{T(x) - T(x_i)}{\|\bar{x} - \bar{x}_i\|} - \bar{U}_i \cdot \bar{\nabla} T(x_i)$$

where a and b is obtained as:

$$\mathbf{a} = \frac{2}{\|\bar{x} - \bar{x}_i\|} \text{ and } \mathbf{b} = \frac{-2T(x_i)}{\|\bar{x} - \bar{x}_i\|} - \bar{U}_i \cdot \bar{\nabla} T(x_i)$$

where

$$\bar{\nabla} T(x_i) = \bar{U}_i^{-1} [\mathbf{a}T(x_i) + \mathbf{b}]$$

The gradient values are not defined on the boundary. The solution starts with first order accurate discretization at the neighboring nodes of the boundary and converts back to second order solution once the gradient values are present for inner nodes. Therefore in order to have accurate solutions the mesh intensity near the boundary should be higher than rest of the domain.

2.3.4 Upwind Criteria

The updating procedure should follow certain criteria in order for the solution to protect the upwind nature of the problem and thus enabling Fast Marching Method

to work correctly. Assume that $T(x)$ is being updated from T_1 , T_2 and T_3 in a tetrahedron. The following criteria should hold true in order for $T(x)$ to be updated from these values.

Criteria 1 : The value of $T(x)$ calculated from the tetrahedron element should be larger than T_1 , T_2 and T_3 . This criterion is required so that the nature of Fast Marching updating procedure hold true. Otherwise accepting the trial value with the smallest value will be incorrect.

Criteria 2 : The gradient of T , which is calculated from the set of vertices should lie in the element as shown in Figure 2.9. Otherwise the information obtained from this element may be incorrect. This restriction is equivalent to

$$(\vec{U} \cdot \vec{U}^T)^{-1} \mathbf{v} \geq 0 \quad (2.19)$$

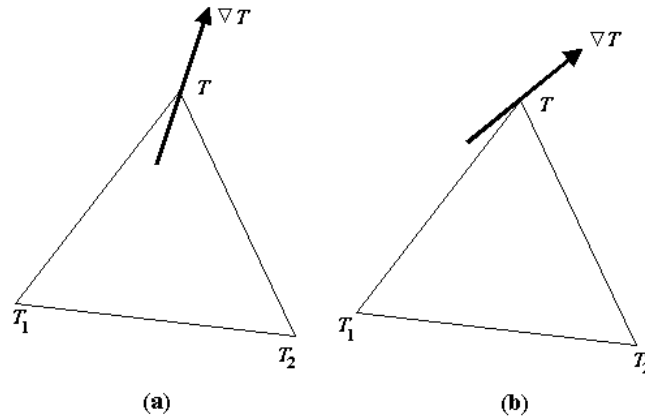


Figure 2.9 (a)Acceptable and (b)Non-Acceptable Gradient Approximations

Criteria 3 : While the above criteria hold, the calculated values from different elements may result in different values. At the vicinity of sharp corners and colliding interfaces this problem is encountered several times during the iterations. Therefore the result with the minimum value should be accepted as the true value.

The first arriving interface overtakes the next coming interface, which is physically what happens in real life.

All three criteria mentioned above should be met in order to accept any calculated value from the FMM. Sometimes these criteria can not be obtained by using regular solution method due to two reasons:

Problem 1: The triangular or tetrahedron element has inner angles larger than 90° . This brings conflicts with the second criteria.

Solution: The solution extends backwards towards the previously accepted vertices such that a new element is created which satisfies upwind criteria.

Problem 2: The element is on the boundary of the volume therefore insufficient information is arriving to that position due to the lack of neighboring elements on the free side. This brings conflicts with first and second criteria.

Solution: This problem is inherent only in three-dimensional problems. The boundaries are solved by two-dimensional FMM in three-dimensional space. The information obtained from the surface solution is used whenever the original solution encounters problems on the boundaries.

2.4 CUT-CELL METHODOLOGY

The flow-interface coupling is performed by a segregated approach where the flow is computed with a frozen interface and the new position of the interface is found using last computed flow properties. In order to apply boundary conditions at intermediate locations in the domain, cut-cell methodology is used. In this method the cells are allowed to be sliced by the interface and at the sliced surfaces of each element boundary conditions are applied. An intermediate tool is required to maintain the communication between the solvers and the burnback codes. In order

to achieve the mentioned communication, cut-cell methods are used to generate the control volume in which the flow solvers perform calculations for performance prediction. The cut-cell methods are used for three different types of solvers, which are given as follows:

1. Zero-dimensional Solvers: Combustion surface area versus propellant burned distance which is called *web* data is required.
2. One-dimensional Solvers: Burn perimeter versus *web* at stations along the axis of the motor is required.
3. 3-dimensional Unstructured Solvers: Unstructured volume mesh with boundary condition definitions is required.

In this section the algorithms, which are capable of generating mesh and data files for all the solvers above are described.

2.4.1 Interface in an Element

The elements, in which interface capturing is performed, are tetrahedrons only. Therefore the interface search is limited to this element type. The interface may slice an element in several ways creating scenarios to be handled. The number of scenarios for different element types and the amount of work they will require are listed in Table 2.1.

Table 2.1 Scenarios for Different Element Types

Element	No. of Scenarios	No. of Nodes	Virtual Work Required
Point	3	1	3
Line	6	2	12
Triangle	10	3	30
Tetrahedron	15	4	60
Hexahedron	28	8	224

As an example for a point element;

1. The time (T_{int}) value which defines the interface can be higher than the time value of the point,
2. It can be equal to time value of the point,
3. It can be below the time value of the point.

Therefore 3 scenarios are possible and with only one node to deal with, the work required is calculated to be 3. The similar discussion can be made for all element types. Table 2.1 is generated by a computer program which searched through all possibilities encountered in different element types.

The tetrahedron element has 15 scenarios. However some of the scenarios are impossible to occur with the tetrahedrons generated by a mesh generator; such as all values at the nodes being equal to interface value which suggests that the element has zero volume. Therefore we get rid of some of the work by deciding which scenarios are likely to occur. There are 8 scenarios, which are possible for real tetrahedrons.

Assume that the interface is recently at a time interval, which is equal to T_0 . Also assume that time values on the element's vertices are selected so that $T_D \geq T_C \geq T_B \geq T_A$ and it is required to find the interface with the value T_0 .

Scenario -1: If $T_0 \geq T_D$ then all the vertices of the element are considered to be inside the gas zone and added to the list of mesh where flow solutions will be carried out.

Scenario -2: If $T_0 \leq T_A$ then the element is considered to be inside the solid grain and added to the list of propellant mesh, where structure solutions will be performed.

Scenario -3: If $T_D > T_0 > T_C$ then a triangular surface element and a triangular prism volume element are created, which is shown in Figure 2.10.a.

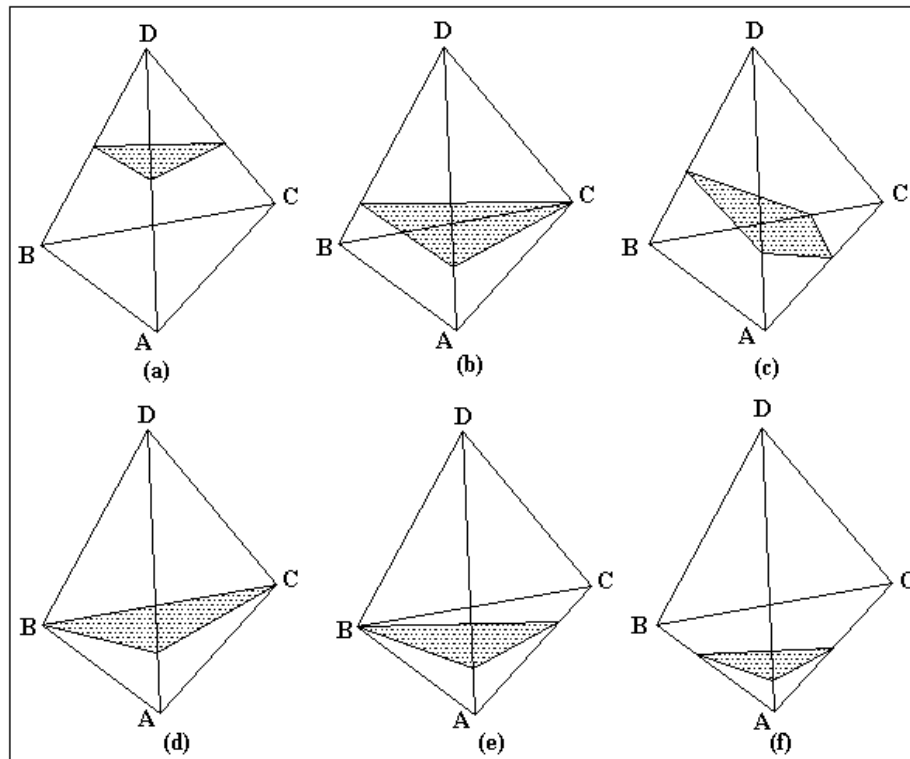


Figure 2.10 Scenarios of Interface in an Element

Scenario -4: If $T_D > T_C = T_0 > T_B$ then a triangular surface and a pyramid volume element are created as shown in Figure 2.10.b.

Scenario -5: If $T_C > T_0 > T_B$ then a quadrilateral surface element and a triangular prism volume element, which is shown in Figure 2.10.c are generated.

Scenario -6: If $T_D > T_C = T_B = T_0 > T_A$ then a triangular surface and a tetrahedron volume element are generated as shown in Figure 2.10.d.

Scenario -7: If $T_D > T_C > T_B = T_0 > T_A$ then a triangular surface and a tetrahedron volume element are generated as shown in Figure 2.10.e.

Scenario -8: If $T_B > T_0 > T_A$ then a triangular surface element and a tetrahedron volume element, which is shown in Figure 2.10.f are generated.

2.4.2 Capturing Interface for Zero-dimensional Solvers

Zero-dimensional flow solvers require web versus burn surface area information. As the solution proceeds, it marches through the web values by linear interpolation according to the rate of burn and at each time step the area of the burn surface obtained from the stationary data which is then used to calculate the amount of mass flux input to the control volume.

Each element is visited once and checked for the scenarios described above. If any of the above scenarios holds true for the element the interface area is calculated. For all the scenarios which are shown in Figure 2.10, except Scenario 5, the interface is a triangular element and by cross product the interface area is calculated.

$$A_{\text{int},i} = 0.5 \cdot |\mathbf{EF}_i \times \mathbf{GF}_i| \quad (2.20)$$

where E, F and G are shown in Figure 2.11.

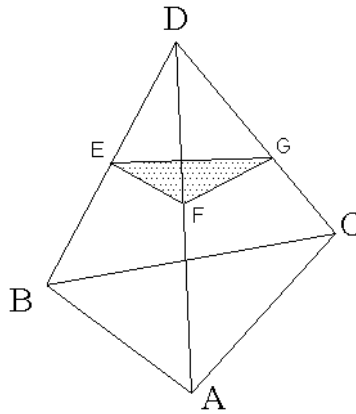


Figure 2.11 Area of Interface

For Scenario 5, the interface is a quadrilateral area, which is calculated with the same method. However it should be kept in mind that two triangles form a quadrilateral and summation of areas of two triangles is used in such a case.

2.4.3 Capturing Interface for One-dimensional Solvers

Similar to calculations for web vs. area, the perimeter information can also be obtained at stations along the motor axis as shown in Figure 2.12.

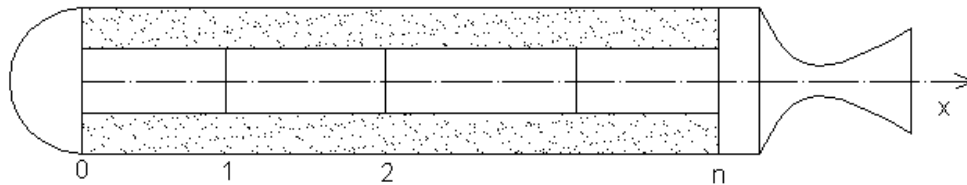


Figure 2.12 Stations along the Axis of the Motor

The interface is captured, and the elements, which carry the interface inside, are marked. Then a second iteration procedure visits only these elements for every station position along x -direction. The area information is sliced with the station location information, which yields a line with the length equal to the burn perimeter.

2.4.4 Capturing Interface for Three-Dimensional Flow

Three-dimensional flow solver used in this work requires boundaries to be defined at the interface and the domain in which solutions will be carried out. After interface capturing, the gas side and solid side elements are separated and boundaries are defined.

In order to obtain three-dimensional mesh for flow solver, the mesh is sliced through the interface. The mesh elements are divided into 3 parts as follows:

1. Mesh elements, which are inside the propellant side.
2. Mesh elements, which are inside the gas side.
3. Mesh elements, which are sliced by the interface.

Each element in the domain is visited once and the elements, which reside in the propellant side, are skipped and stored to an array where no solver calculations will be performed. If the element is inside the flow side, it will be stored into captured array, which also requires no modification.

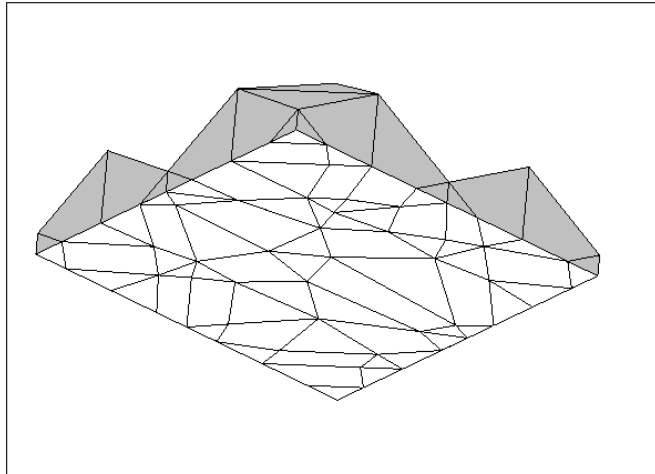


Figure 2.13 Hybrid Mesh near Interface

However, if the interface passes through an element, special care should be taken. According to the scenarios described above, the element is sliced. The original element is removed from the mesh file. The new node points (E, F, G, and H) which lay on the interface are used to create a new element. Thus this process may cause generation of unstructured elements with 4, 5 and 6 surfaces as can be seen in Figure 2.13.

While generating new elements, nodal points which coincide are also generated. These node points should be found and connectivity matrix should be re-organized accordingly. Also some elements may have 3 surfaces left in the mesh file, which suggest that the volume disappear with a slice coincident to one of the surfaces of

the element. The connectivity matrix and the node arrays should be re-arranged in order to preserve finite volumes and their neighboring information.

Three methods can be used to modify the sliced elements which are shown in Figure 2.14. If the elements are left as they are, hybrid mesh is created with different number of nodes creating each element (Figure 2.14.b).

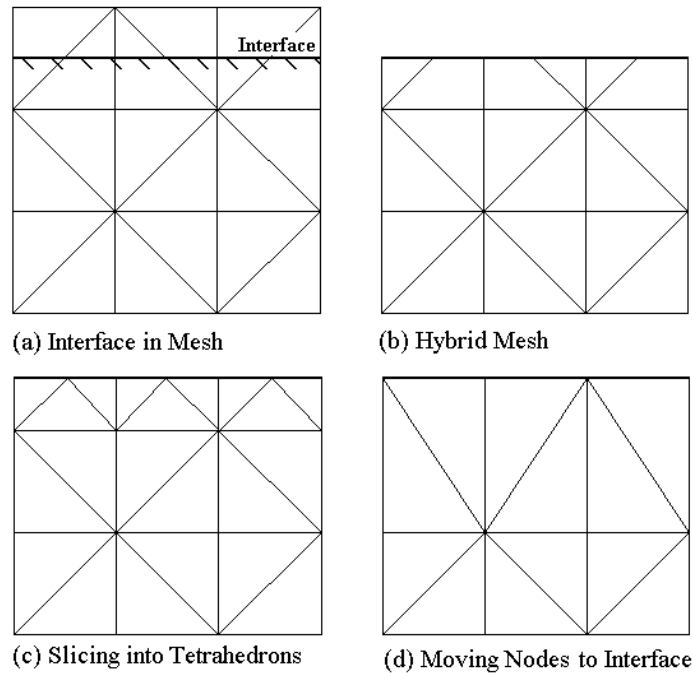


Figure 2.14 Methods of Mesh Modification

The hybrid elements can be divided into tetrahedrons therefore the homogeneity can be restored (Figure 2.14.c). However the number of elements is increased in such a case. For viscous solutions this may be advantageous. Also the hybrid part of the mesh can be pushed outside the flow control volume into the propellant side by moving the nodes on to the interface (Figure 2.14.d). This method requires that the node elements follow the boundary strictly which requires extra amount of work and in order to keep the generic form of the capturing extra boundary definitions may be required.

The hybrid mesh method is found to be efficient since no extra work is required to modify mesh elements. Since no reconstruction is performed, the errors are minimized and the number of elements is kept constant. The hybrid mesh requires a flow solver, which is capable of accepting any shaped element. This solver is described in the next chapter.

The hybrid mesh is composed of elements with different geometric shapes. The elements may have nodes between 8 and 4. 8 nodes create a rectangular prism, 6 nodes create a wedge, 5 nodes create a pyramid and 4 nodes create a tetrahedron. When a tetrahedron element is sliced, the new elements are forced to have 6 nodes at maximum. Therefore our application is sufficient for all scenarios.

Table 2.2 Hybrid Mesh Connectivity

Element Type	Connectivity							
Rectangular Prism	1	2	3	4	5	6	7	8
Wedge(Triangular Prism)	1	2	2	3	4	5	5	6
Pyramid	1	2	3	4	5	5	5	5
Tetrahedron	1	2	2	3	4	4	4	4

The definitions of connectivity are same for all elements. In the connectivity matrix 8 nodes are given for all elements. However some nodes are repeated if an element different than a rectangular prism is to be formed. The Figure 2.15 summarizes the procedure that is followed in describing the connectivity for each element and Table 2.2 shows the connectivity of each type of element.

However it should be mentioned that the hybrid mesh obtained by cutting the cells may have elements with zero or near to zero volume. In the present algorithm, these elements are searched for and they are removed from the grid.

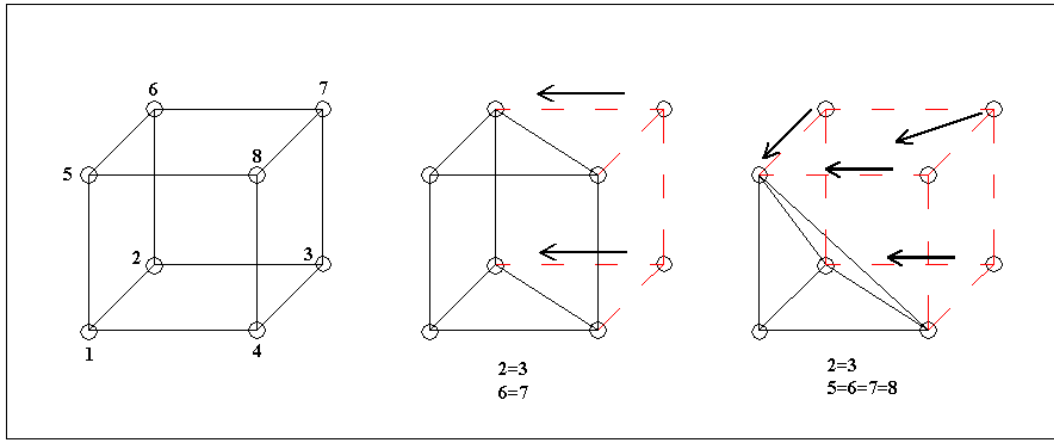


Figure 2.15 Hybrid Mesh Formation

2.5 NUMERICAL TEST CASES FOR FAST MARCHING METHOD

The test cases given in this section are solved by using the derivative approximation. These problems are selected such that their solution can be found analytically and a comparison can be made between the calculated values. Although these problems are simple, they inherit all the problems that may be encountered in real physical problems.

2.5.1 Error Definitions

Error is defined as the difference between the numerical calculation result and the analytical result at each nodal position when the solution is converged. The error definition is given below:

$$error_j = T_{j,cal} - T(x)_{j,analytic} \quad (2.21)$$

In order to define a single value defining the error, L_p -norms are used. L_2 and L_∞ norms which are used in this work and are defined as follows:

$$L_2 = \sqrt{\sum_{j=1}^N (error_j)^2} / N \quad (2.22)$$

$$L_\infty = \max |e_j| \quad j = 1..N \quad (2.23)$$

Since the area of the interface is more important for aerodynamic calculations, another error definition is used. The interface error is defined as the ratio of the difference of numerically calculated and analytically calculated interface area to the analytically calculated interface area. By applying interface error to our solutions, the total error, which is encountered during mass flux addition through the surface of the propellant, can be estimated. The interface error is given as follows:

$$IE = \frac{(A_{int,numeric} - A_{int,analytic})}{A_{int,analytic}} \quad (2.24)$$

2.5.2 Grid Size

The grid size is defined as the largest edge of an element in the domain. When the grid size of 100 units is defined for the domain, there can be no elements with an edge whose length is larger than 100 units. This definition is used in defining mesh intensity in the following test cases.

2.5.3 Two-Dimensional Test Cases

2.5.3.1 Test Case 0 – Flat Interface

This test case is two-dimensional and the geometry is given in Figure 2.16. The cube is 1000 units x 1000 units in size where the interface is placed at the centre of the cube. The upper portion is called the “Propellant” side because the interface

will move in this domain only whereas the lower part is called the “Fluid” part in which no solutions will be carried out by the FMM algorithms.

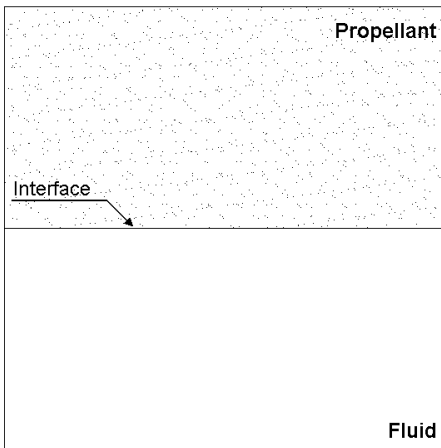


Figure 2.16 Test Case 1 Geometry

The results are shown in Figure 2.17 where the contours show the arrival time of the interface at each node point.

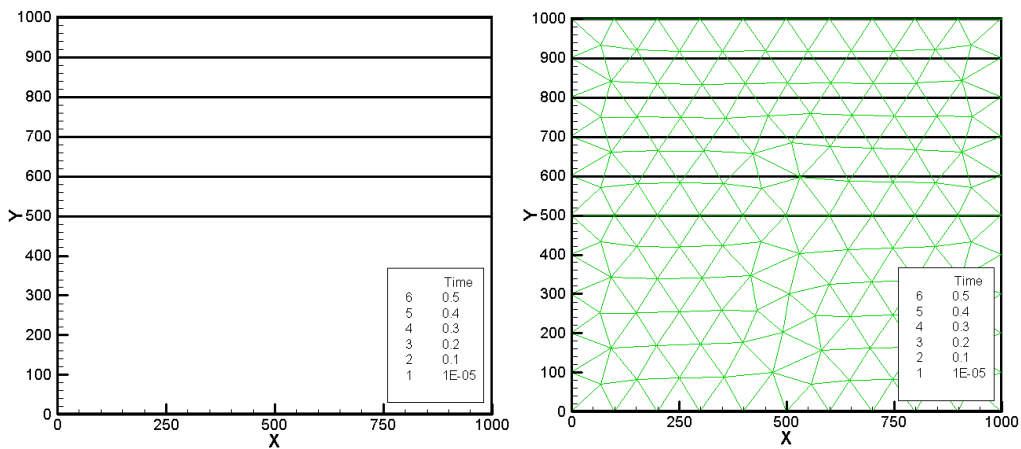


Figure 2.17 Test Case 0: Results

The error estimates are given are given in Table 2.3. It has been found that the solution possesses only numerical errors due to round off errors on an interface with no slope change.

Table 2.3 Two-dimensional Test Case 0: Error Estimates

L-2		$1.14 \cdot 10^{-9}$
L- ∞		$6.6 \cdot 10^{-10}$
I-E	Capture Time	Error
	0.1 sec.	$1.13 \cdot 10^{-8}$
	0.48 sec.	$1.14 \cdot 10^{-8}$

2.5.3.2 Test Case 1 – 90° Bend Interface

This test case is two-dimensional and the geometry is given in Figure 2.18. The cube is 1000 units x 1000 units in size where the fluid phase is a 500 units x 500 units cube placed at the lower left corner of the larger cube.

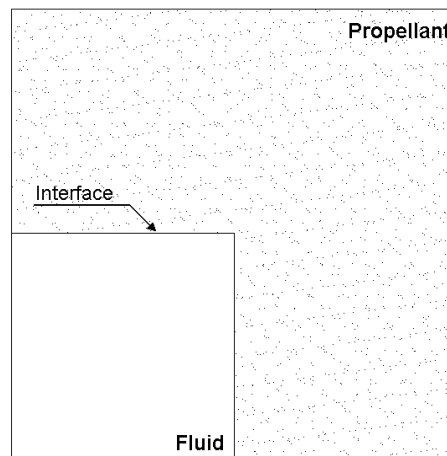


Figure 2.18 Test Case 1 Geometry

The results for test case 1 are shown in Figure 2.19, Figure 2.20 and Figure 2.21 for different mesh intensities. The contours show the arrival time of the interface at each node point. The dark lines correspond to the calculated values whereas the dashed lines correspond to real solution.

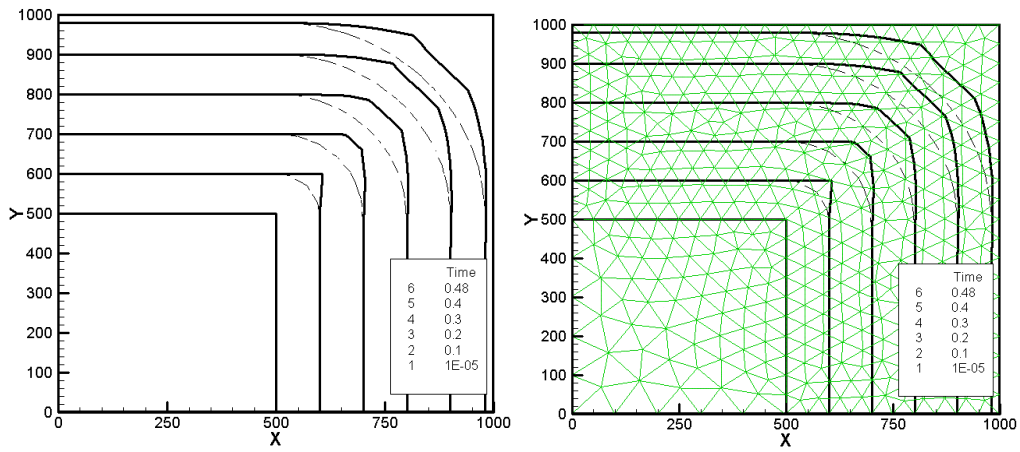


Figure 2.19 Test Case 1: Results for Grid Size of 100 units

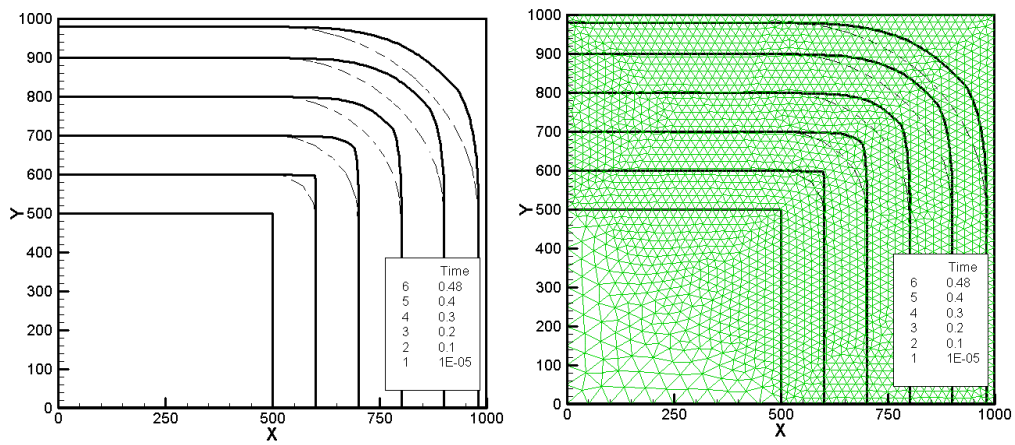


Figure 2.20 Test Case 1: Results for Grid Size of 50 units

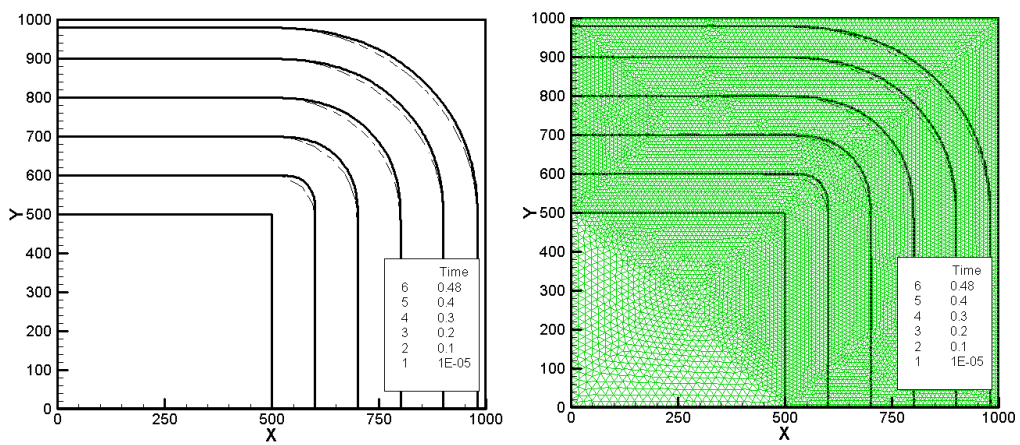


Figure 2.21 Test Case 1: Results for Grid Size of 10 units

The error estimates are given in Table 2.4. Discontinuities in a surface cause severe errors in all regression problems. The real solution is obtained by assuming that the viscous solution would be obtained by the FMM method. However if the mesh intensity is low, the viscous solution may not be obtained near the surface since the solution does not have enough space to pass the information at the sharp point of the interface. Therefore the mesh intensity is increased and L_2 error decreased considerably which suggests that increasing mesh intensity enhances the solution.

Table 2.4 2-dimensional Test Case 1 Error Estimates

Grid Type	L-2	L-∞	Capture Time	I-E
Outward Size = 100 units # of Elements =826	5.06×10^{-4}	0.07245	0.1	0.0414
			0.2	0.0457
			0.3	0.0415
			0.4	0.0391
Outward Size = 50 units # of Elements =4984	1.85×10^{-4}	0.0574	0.1	0.0345
			0.2	0.0412
			0.3	0.0365
			0.4	0.0326
Outward Size = 10 units # of Elements =20204	2.32×10^{-5}	0.0147	0.1	0.0116
			0.2	0.0094
			0.3	0.0081
			0.4	0.0071

The solutions to this test case exhibit Huygens ' Principle where sharp corners do not generate sharp corners. It can be assumed that no sharp corner exists in the nature but will always have a radius.

2.5.3.3 Test Case 2 – Circle Interface

In this test case the circle (radius = 250 units) is regressed outwards and inwards in a box 1000 units x 1000 units in size which is shown in Figure 2.22.

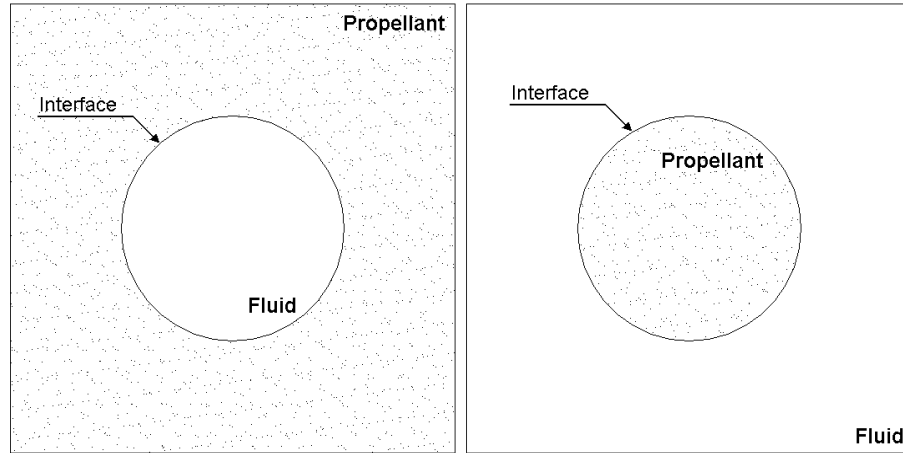


Figure 2.22 (a) Outward Burning Circle (b) Inward Burning Circle

The outward burning circle present no difficulty for the algorithms of Fast Marching since the curvature is constant through the whole domain. However the inward burning circle present difficulties since the interface disappears at the end of the process.

In most interface tracking algorithms the loss of any geometry is handled with complicated mesh re-structuring algorithms. The present work uses no extra handling procedure, the nature of the algorithm handles the surface loss naturally.

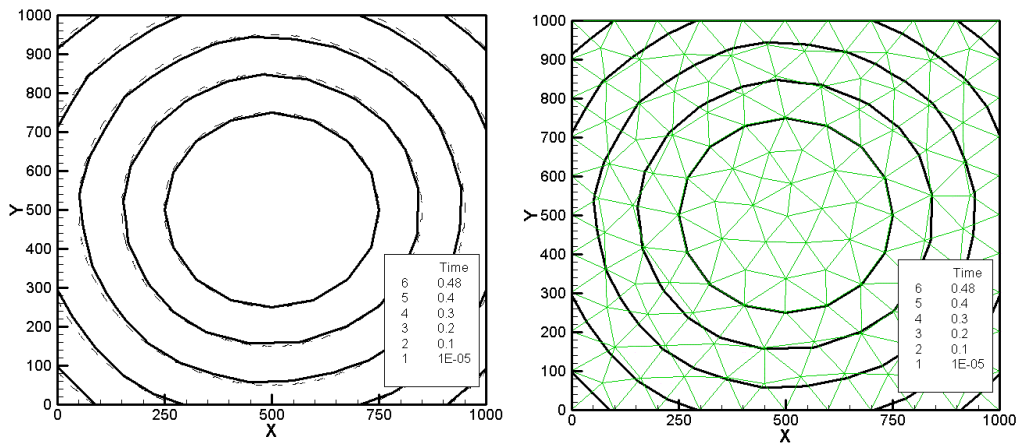


Figure 2.23 Test Case 2: Outward Burning Results with Grid Size of 100 units

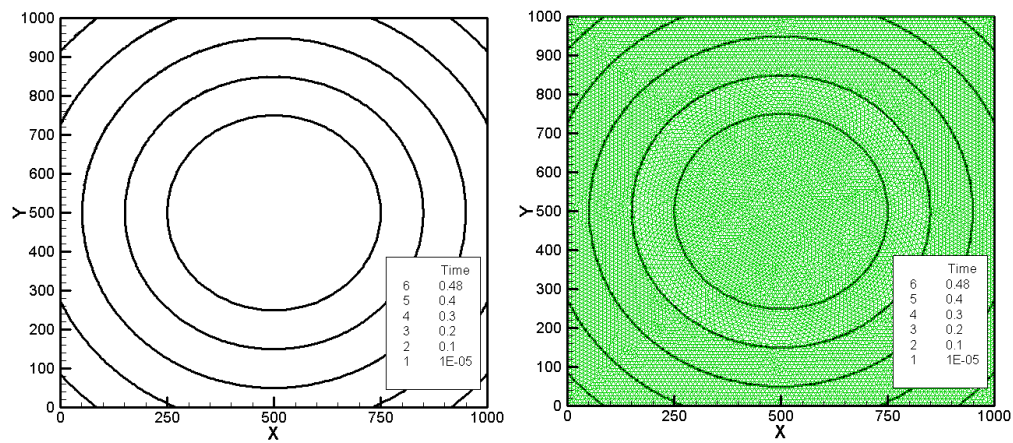


Figure 2.24 Test Case 2: Outward Burning Results with Grid Size of 10 units

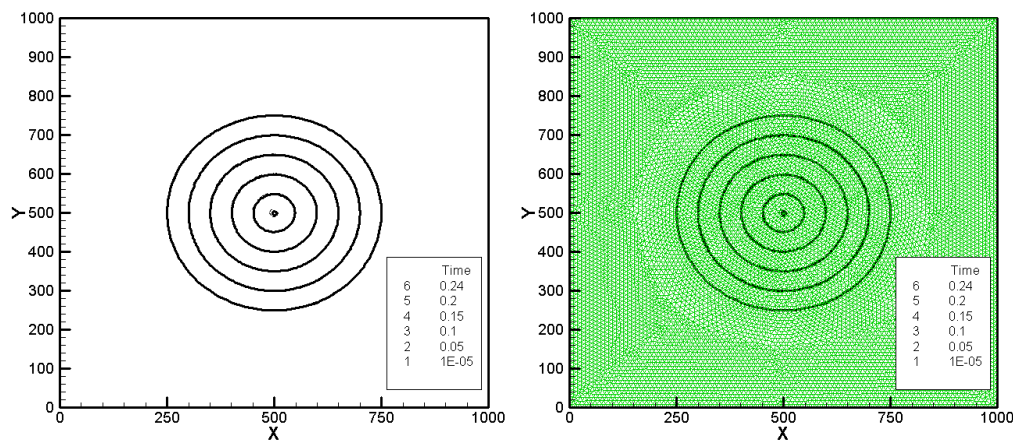


Figure 2.25 Test Case 2: Inward Burning Results with Grid Size of 10 units

The results are given in Figure 2.23, Figure 2.24 and Figure 2.25. The dark colored contour lines are calculated values whereas the dashed contour lines represent the actual value for the arrival time. The error estimates are given in Table 2.5.

Table 2.5 2-dimensional Test Case 2 Error Estimates

Grid Type	L-2	L-∞	Capture Time	I-E
Outward Size = 100 units # of Elements =244	2.31 x 10 ⁻⁴	0.01335	0.1	0.01339
			0.2	0.01321
			0.25	0.01377
Outward Size = 50 units # of Elements =916	6.78 x 10 ⁻⁵	0.006247	0.1	0.00769
			0.2	0.00821
			0.25	0.00822
Outward Size = 10 units # of Elements =22126	3.02 x 10 ⁻⁶	0.001147	0.1	0.00169
			0.2	0.00175
			0.25	0.001746
Inward Size = 100 units # of Elements =244	4.09 x 10 ⁻⁴	0.0541	0.05	0.04419
			0.1	0.143
			0.2	1.0
Inward Size = 50 units # of Elements =916	3.83 x 10 ⁻⁵	0.011208	0.05	0.009987
			0.1	0.0245
			0.2	0.2218
Inward Size = 10 units # of Elements =22126	1.58 x 10 ⁻⁶	0.003276	0.05	0.001433
			0.1	0.00416
			0.2	0.03528

2.5.3.4 Test Case 3 – Smooth Interface

The third test case consists of 3 quarter circles (radius = 250 units) combined smoothly in a box of 1000 units x 1000 units. The interface is regressed outwards and inwards as shown in Figure 2.26.

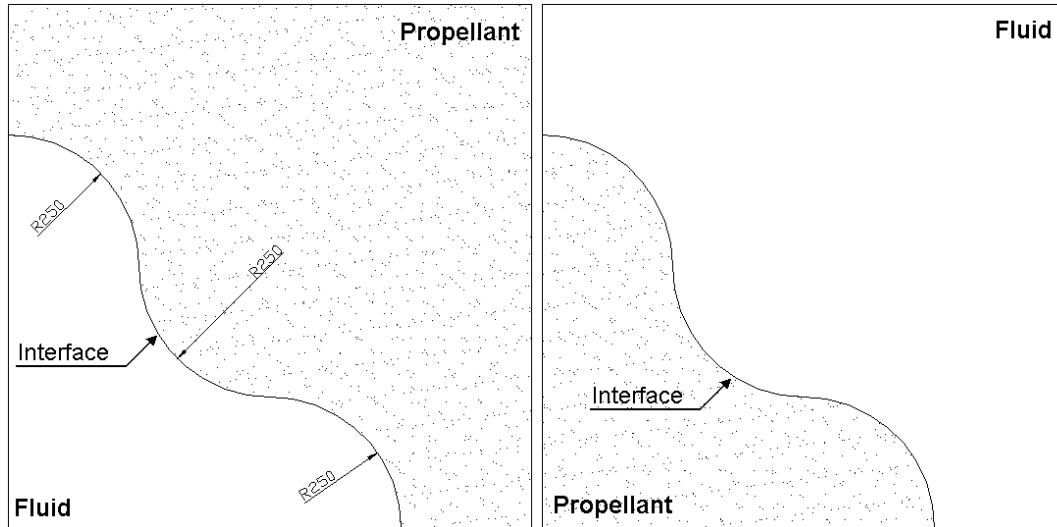


Figure 2.26 (a) Outward Burning Smooth (b) Inward Burning Smooth

The solution is problematic since the middle circle disappears after outward regression and side circles disappear after inward regression. All other methods especially interface tracking methods have to take special precautions in order to handle this problem. The solutions obtained are in good agreement with the actual values in all mesh intensities due to the smoothness of the problem. However with low intensity mesh at the interface, the circles are not defined properly. The results are given in Figure 2.27, Figure 2.28 and Figure 2.29. The dark colored contour lines are calculated values whereas the dashed contour lines represent the actual value for the arrival time. The error estimates are given in Table 2.6.

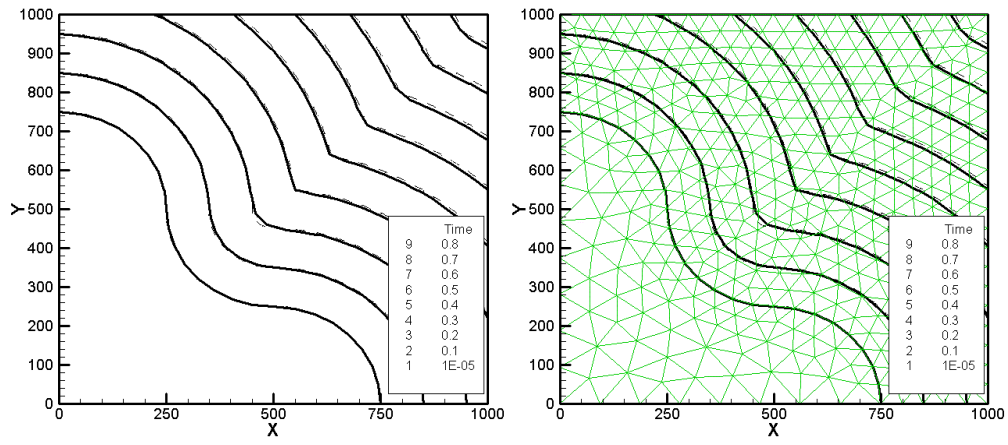


Figure 2.27 Test Case 3: Outward Regression Results with Grid Size of 50 units

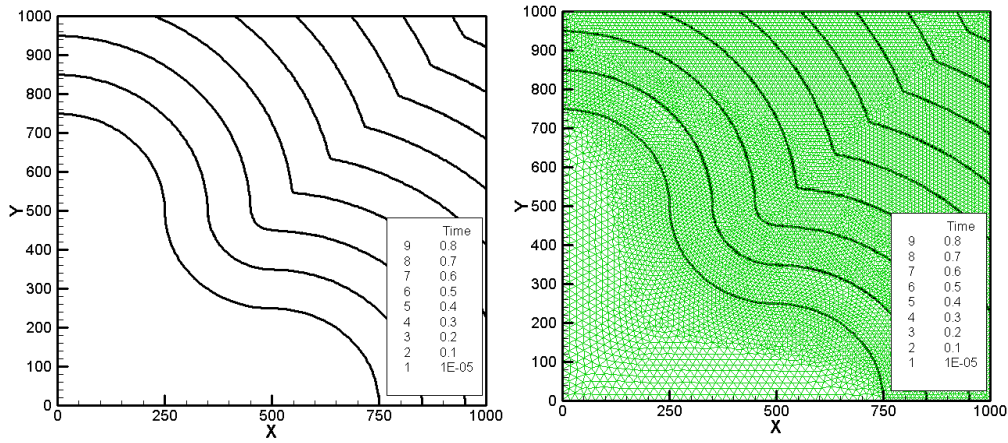


Figure 2.28 Test Case 3: Outward Regression Results with Grid Size of 10 units

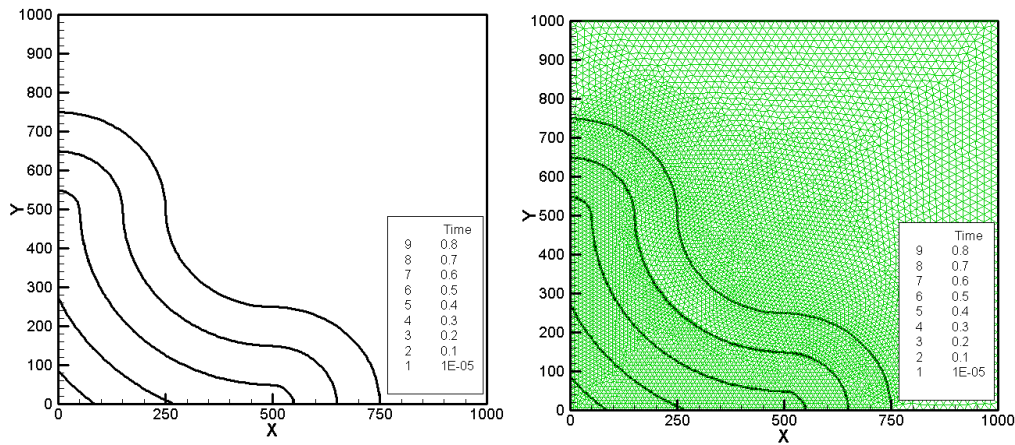


Figure 2.29 Test Case 3: Inward Regression Results with Grid Size of 10 units

Table 2.6 2-dimensional Test Case 3 Error Estimates

Grid Type	L-2	L-∞	Capture Time	I-E
Outward Size = 50 units # of Elements = 796	1.12 x 10 ⁻⁴	0.00912	0.05	0.00636
			0.1	0.00824
			0.2	0.0141
Outward Size = 20 units # of Elements =4944	1.83 x 10 ⁻⁵	0.0047	0.05	0.00239
			0.1	0.00375
			0.2	0.00603
Outward Size = 10 units # of Elements =19228	4.84 x 10 ⁻⁶	0.00438	0.05	0.00159
			0.1	0.00225
			0.2	0.00335
Inward Size = 50 units # of Elements =614	6.27 x 10 ⁻⁵	0.01123	0.05	0.00691
			0.1	0.0124
			0.2	0.028
Inward Size = 20 units # of Elements =3646	1.04 x 10 ⁻⁵	0.00461	0.05	0.00239
			0.1	0.0048
			0.2	0.0131
Inward Size = 10 units # of Elements =14030	2.86 x 10 ⁻⁶	0.003857	0.05	0.00128
			0.1	0.00294
			0.2	0.0058

2.5.3.5 Test Case 4 – Colliding Circles

The most difficult problem in interface regression is handled in this two-dimensional problem, where two circles (radius=500 units) at the upper right and lower left corners of a cube of 1000 units x 1000 units, expands towards each other.

The problem is encountered when two interfaces meet somewhere in space. The exact location is unknown, therefore no precautions can be taken before hand. The FMM algorithm has the solution in its nature since the smallest arrival time is

accepted for any node position and the node is not visited twice. By this way the interface which reaches a node first, claims the node for itself, thus the result is correct. The geometry is given in Figure 2.30.

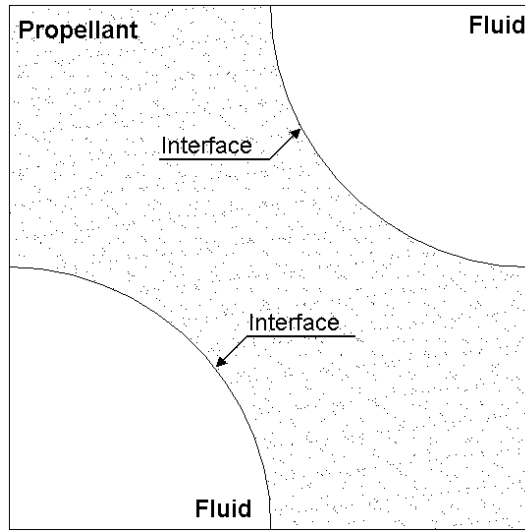


Figure 2.30 Colliding Circles

The results are given in Figure 2.31 and Figure 2.32. The dark colored contour lines are calculated values whereas the dashed contour lines represent the actual value for the arrival time.

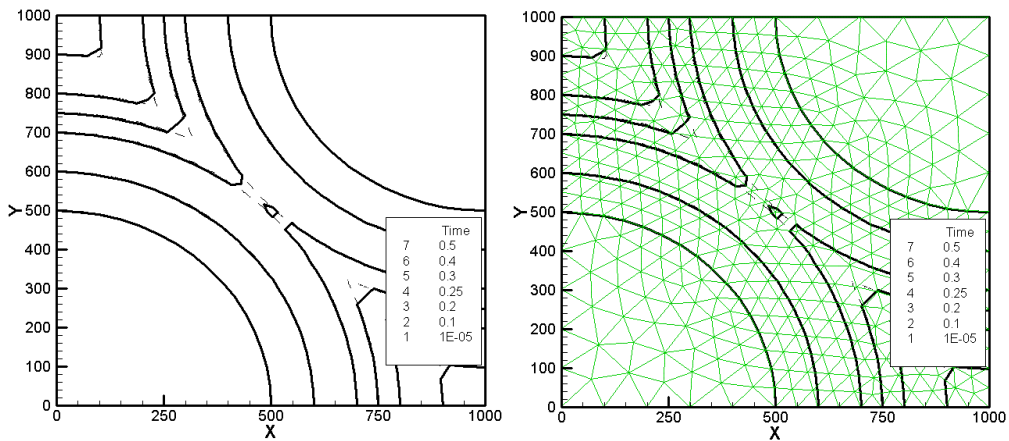


Figure 2.31 Test Case 4: Regression Results with Grid Size of 50 units

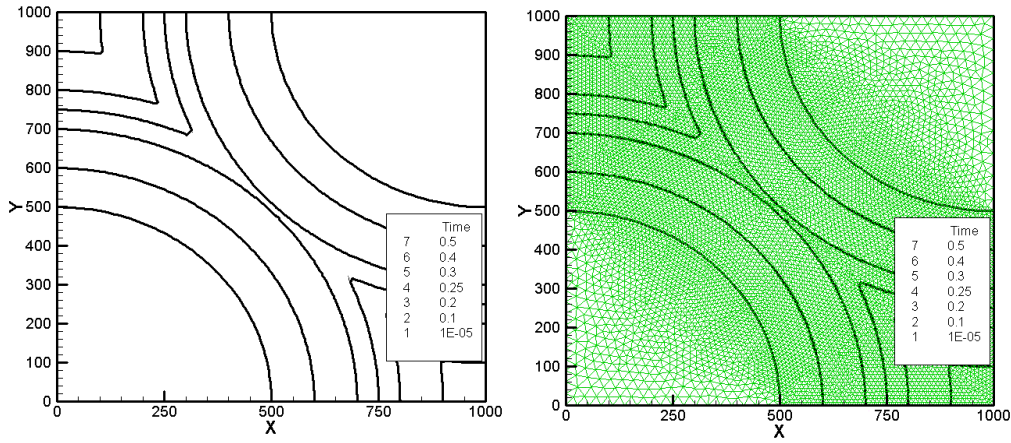


Figure 2.32 Test Case 4: Regression Results with Grid Size of 10 units

The contour island, which is seen in Figure 2.31, is due to a plotting error. While generating a contour plot, interpolation is performed at the site of collision. However the real answer can be found by extrapolation at this location. The only solution is to increase mesh intensity thus reducing the error to a minimum. The error estimates are given in Table 2.7.

Table 2.7 2-dimensional Test Case 4 Error Estimates

Grid Type	L-2	L-∞
Size = 50 units # of Elements = 760	9.53×10^{-5}	0.0417
Size = 20 units # of Elements = 4548	1.17×10^{-5}	0.0206
Size = 10 units # of Elements = 12252	3.17×10^{-6}	0.00838

2.5.4 Three-Dimensional Test Cases

In this section the similar test cases in three-dimension are investigated. Test Case – 0 is skipped, since the solver does not make any error for a constant slope surface.

The three-dimensional test cases are obtained by extruding the similar two-dimensional test cases in the z-direction by 500 units.

2.5.4.1 Test Case 1 – Cube In a Rectangular Prism

Test case 1 results are given in Figure 2.33 and Figure 2.34. The dark lines represent the calculated values whereas the dashed lines are exact values.

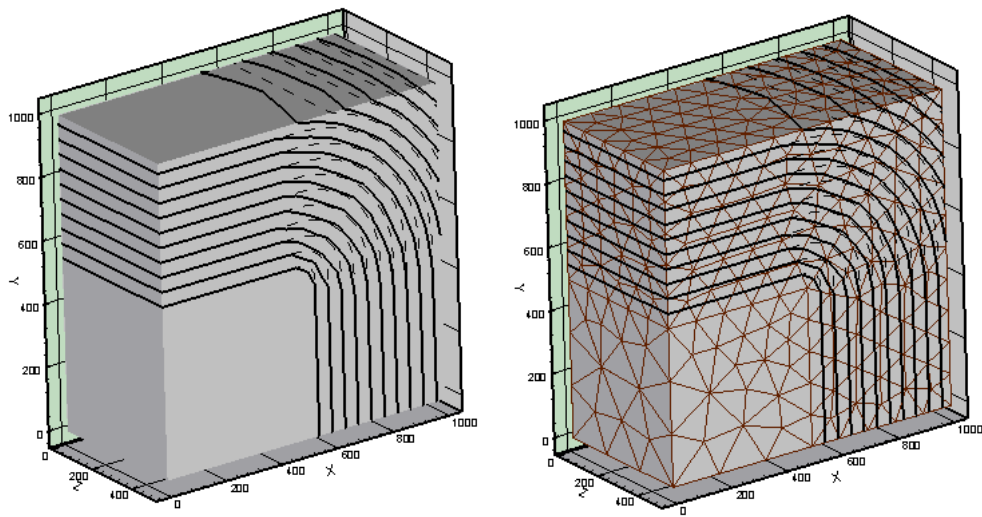


Figure 2.33 Test Case 1: Regression Results with Grid Size of 100 units

The error estimates are given in Table 2.8. It can be seen that as the mesh intensity increases the accuracy of the solution gets better at a cost of operation time.

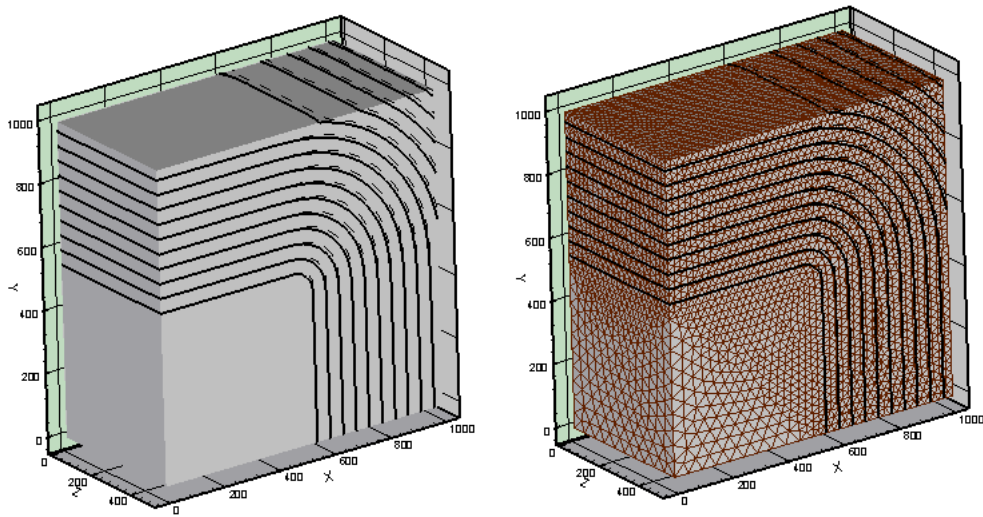


Figure 2.34 Test Case 1: Regression Results with Grid Size of 25 units

The L_2 error increases as the mesh intensity increases, due to the reason that the increase of number of elements is much higher than the two-dimensional cases where the accuracy is not sufficient to lower the overall effect.

The accuracy of the solutions for burn area may be achieved through increasing mesh intensity. The interface area comparison is given in Figure 2.35.

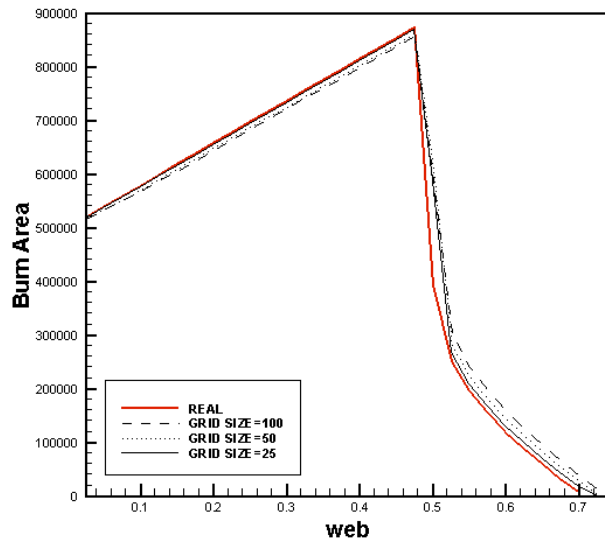


Figure 2.35 Burn area versus Web for Test Case 1

Table 2.8 3-dimensional Test Case 1 Error Estimates

Grid Type	L-2	L-∞
Outward Size = 100 units # of Elements = 3390	1.15×10^{-4}	0.04407
Outward Size = 50 units # of Elements = 24298	2.42×10^{-5}	0.02803
Outward Size = 25 units # of Elements = 187950	3.86×10^{-6}	0.01268

2.5.4.2 Test Case 2 – Cylinder in a Rectangular Prism

Test Case 2 results are given in Figure 2.36 for the outward regression and in Figure 2.37 and Figure 2.38 for inward regression.

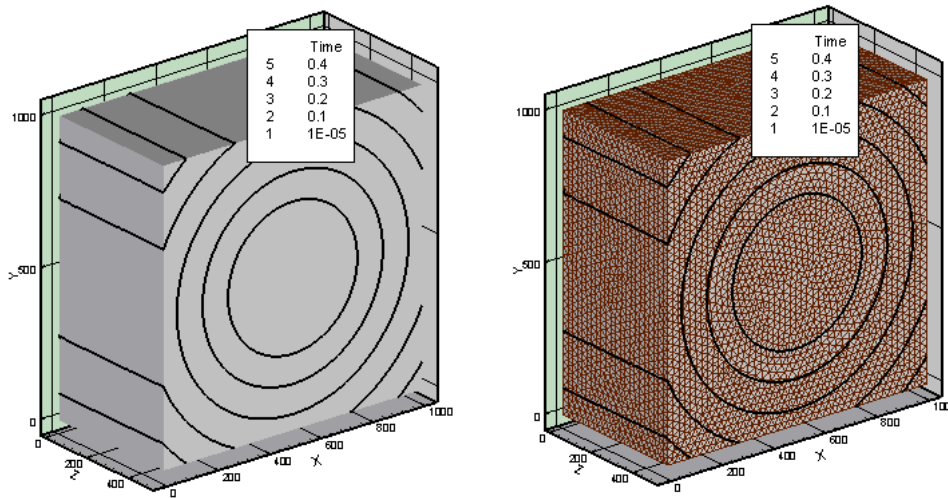


Figure 2.36 Test Case 2: Outward Regression Results with Grid Size of 25 units

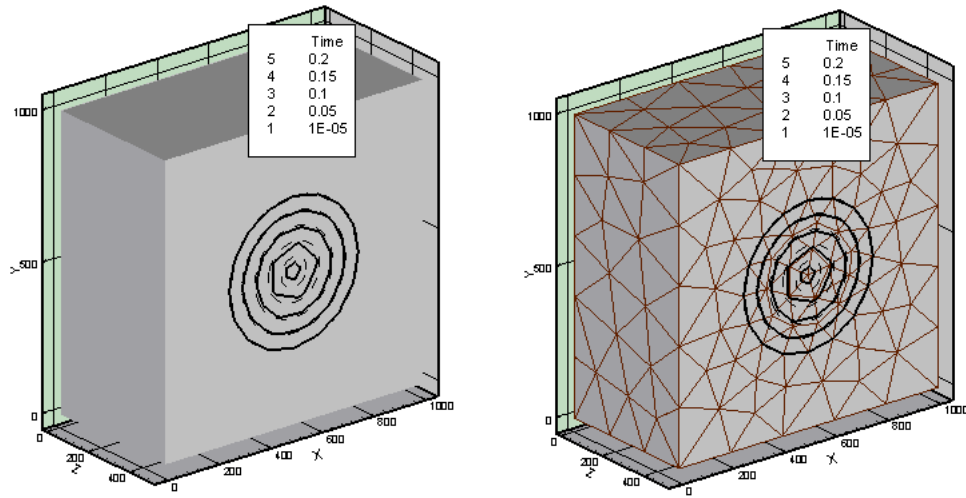


Figure 2.37 Test Case 2 Inward Regression Results with Grid Size of 100 units

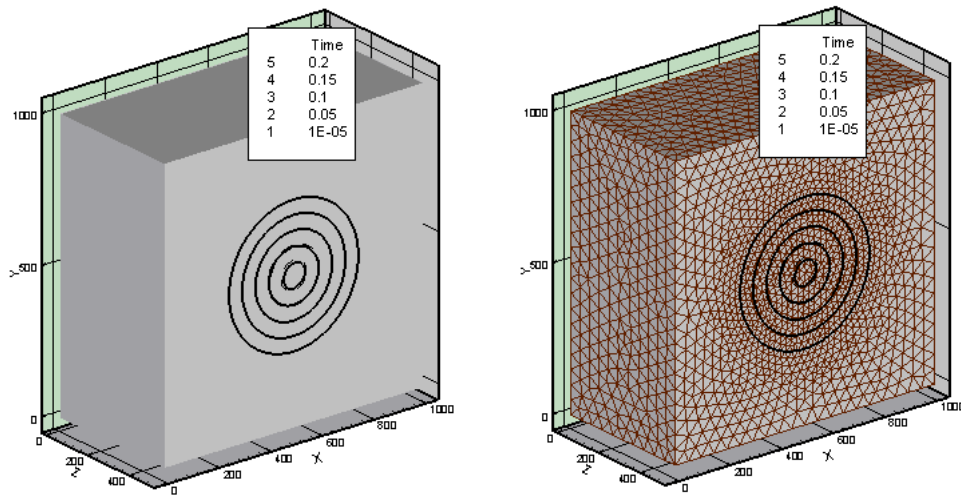


Figure 2.38 Test Case 2 Inward Regression Results with Grid Size of 25 units

The error estimates are given in Table 2.9. The L_{∞} error for inward regression with low mesh intensity is high due to the reason that the algorithm is unable to find enough space to have a mature solution.

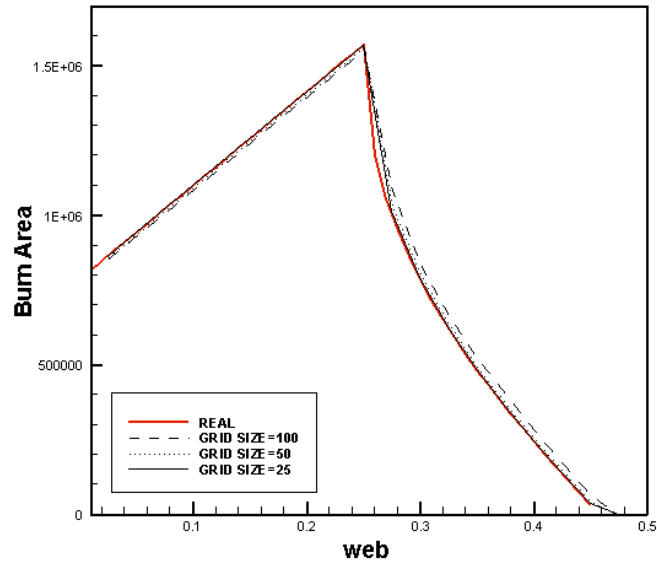


Figure 2.39 Burn area vs. Web for Test Case 2 –Outward

The interface error, which is made while solving inward burning, is considerably high when low intensity mesh is used. The locations where disappearing pieces exist in a model are critical, therefore the size of mesh elements should be decreased in such areas.

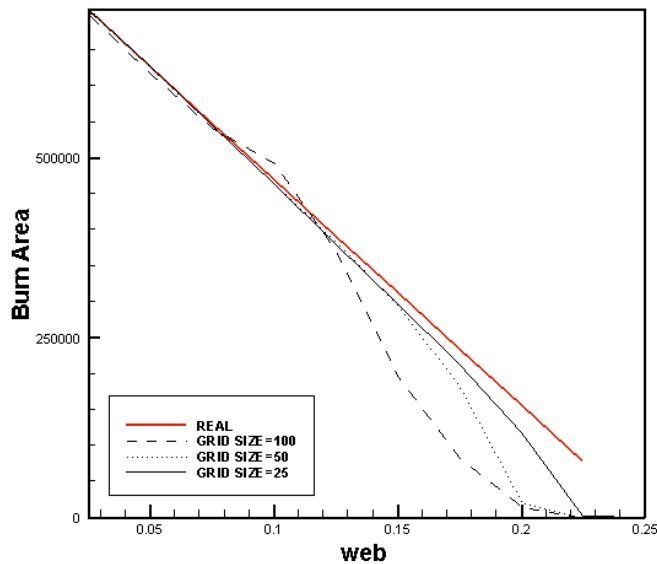


Figure 2.40 Burn area vs. Web for Test Case 2 –Inward

Table 2.9 3-dimensional Test Case 2 Error Estimates

Grid Type	L-2	L-∞
Outward Size = 100 # of Elements = 3967	4.76×10^{-5}	0.02038
Outward Size = 50 # of Elements = 27727	6.35×10^{-6}	0.007889
Outward Size = 25 # of Elements = 191127	9.52×10^{-7}	0.00323
Inward Size = 100 # of Elements = 2040	1.52×10^{-4}	0.129
Inward Size = 50 # of Elements = 13559	2.73×10^{-5}	0.07534
Inward Size = 25 # of Elements = 89702	6.94×10^{-6}	0.0561

2.5.4.3 Test Case 3 – Smooth Interface in a Rectangular Prism

The test case results are given in Figure 2.41 and Figure 2.42 for outward regression and in Figure 2.43 and Figure 2.44 for inward regression.

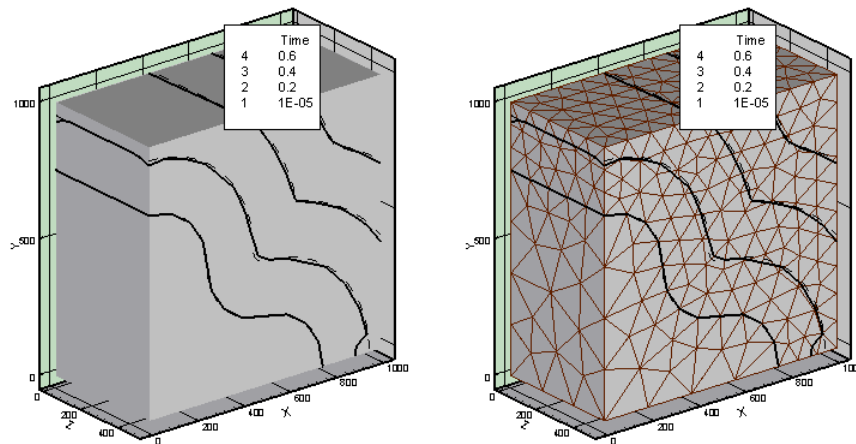


Figure 2.41 Test Case 3: Outward Regression Results with Grid Size of 100 units

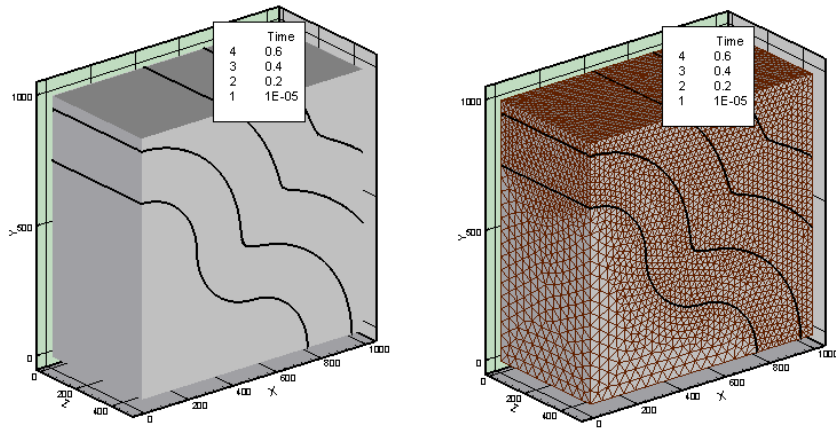


Figure 2.42 Test Case 3: Outward Regression Results with Grid Size of 25 units

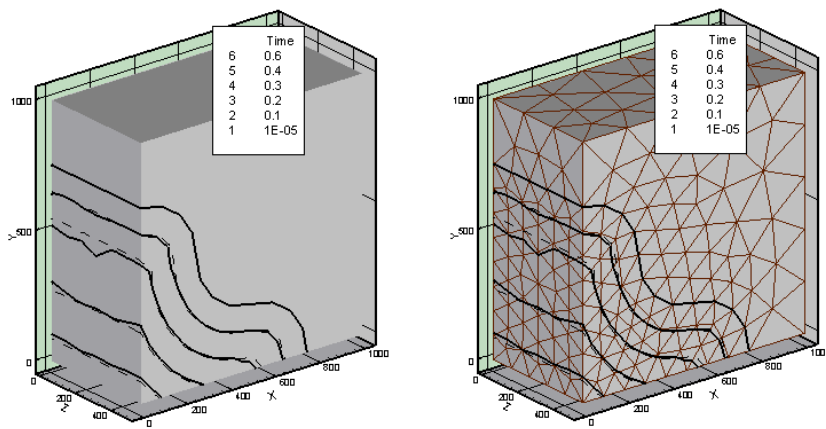


Figure 2.43 Test Case 3: Inward Regression Results with Grid Size of 100 units

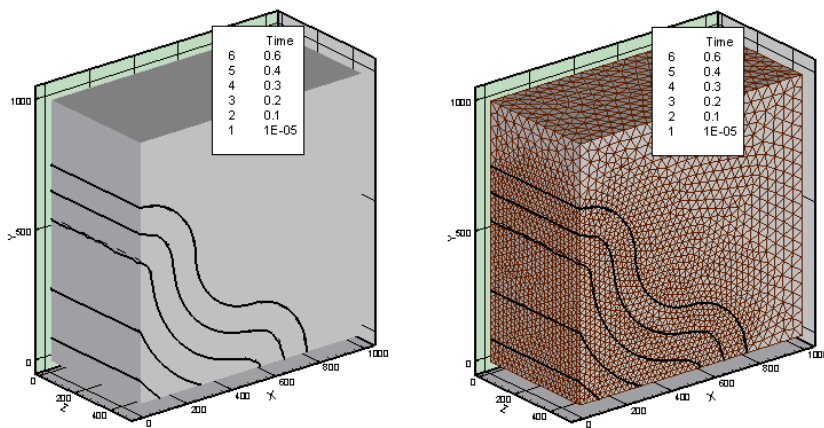


Figure 2.44 Test Case 3: Inward Regression Results with Grid Size of 25 units

The upwind criteria cause errors to occur on the boundaries of the solution since the information required for acceptance criteria may not be sufficient on the boundaries. Increasing the mesh intensity solves for the problem most of the time. The error estimates are given in Table 2.10.

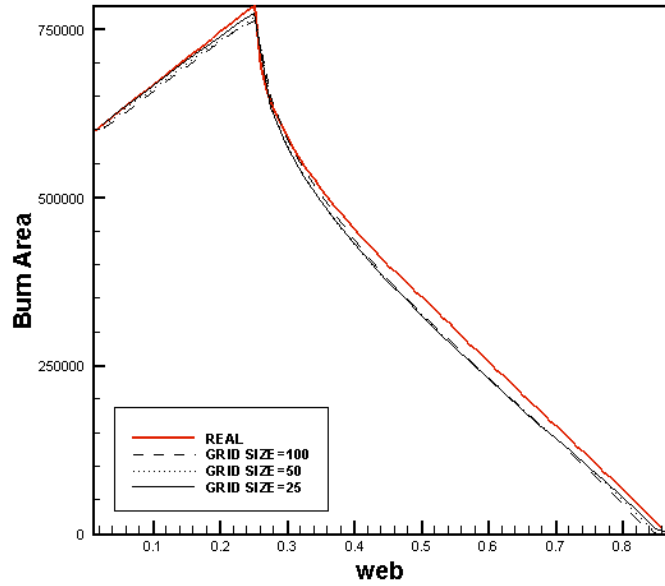


Figure 2.45 Burn area vs. Web for Test Case 3 –Outward

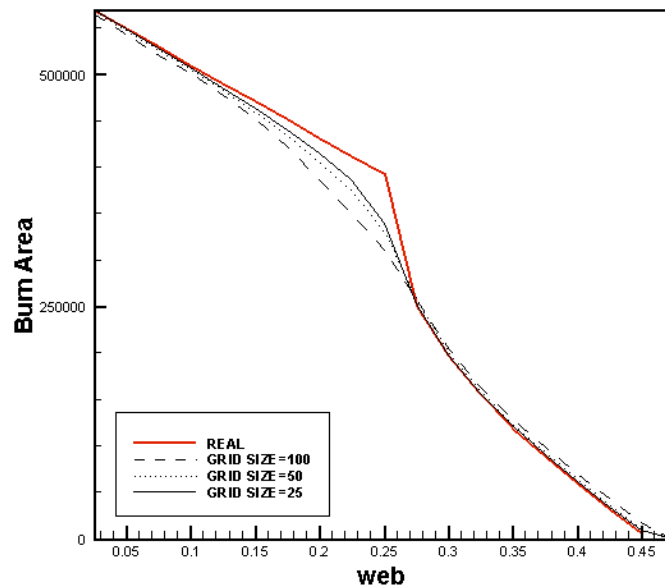


Figure 2.46 Burn area vs. Web for Test Case 3 –Inward

Table 2.10 3-dimensional Test Case 3 Error Estimates

Grid Type	L-2	L-∞
Outward, Size = 100 # of Elements = 3423	7.27×10^{-5}	0.0558
Outward, Size = 50 # of Elements = 22719	1.57×10^{-5}	0.0427
Outward, Size = 25 # of Elements = 162410	2.53×10^{-6}	0.0125
Inward, Size = 100 # of Elements = 2207	8.29×10^{-5}	0.0589
Inward, Size = 50 # of Elements = 13601	1.90×10^{-5}	0.0423
Inward, Size = 25 # of Elements = 89547	4.53×10^{-6}	0.0287

2.5.4.4 Test Case 4 – Colliding Cylinders in a Rectangular Prism

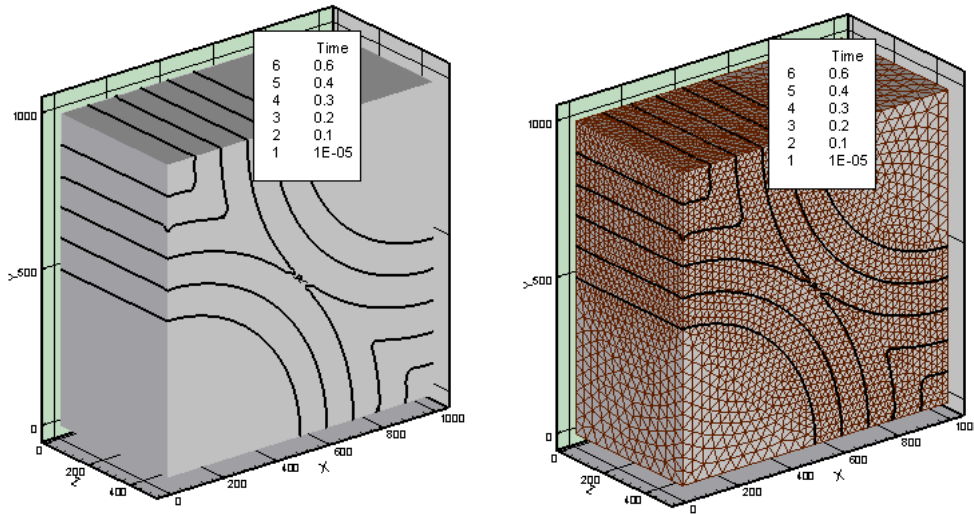


Figure 2.47 Test Case 4: Regression Results with Grid Size of 25 units

The test case results are given in Figure 2.47 and error estimates are given in Table 2.11. The $L-\infty$ error with mesh size 100 units is very high due to low mesh intensity. The solution is not capable of finding the correct gradient direction in such a case. The gradient at the point of collision has two values each coming from different burning surfaces. In order for the solution to give a decision more information is required which the high intensity mesh should supply. Mesh adaptation can be useful at the position of collision.

Table 2.11 Three-dimensional Test Case 4 Error Estimates

Grid Type	L-2	L-∞
Outward, Size = 100 # of Elements = 2900	1.82×10^{-4}	0.215
Outward, Size = 50 # of Elements = 22297	1.84×10^{-5}	0.0846
Outward, Size = 25 # of Elements = 131591	3.58×10^{-6}	0.0580

The interface errors and L - errors decrease as the mesh intensity increases. The run time increase is linearly proportional to the number of elements in the solution domain for the interface capturing algorithms. However for the three-dimensional flow solver the time required for convergence increases exponentially as the number of elements increase. Therefore a decision has to be made according to the accuracy requirement of the coupled solution.

An alternative is to solve the interface capturing on a fine mesh and transferring the interface data onto a different coarser mesh for the flow solutions or vice versa. This method is used in all applications given in next chapters which reduced run time for the flow solver and increased interface accuracy.

CHAPTER 3

FLOW SOLVERS FOR ROCKET MOTOR PERFORMANCE PREDICTION

For interface capturing purposes, the speed by which the interface moves have to be calculated, which is called the burning rate of the propellant. It is sensitive to several factors:

- Chemical composition of the propellant: The propellant is a mixture of several compounds and the combination of several products, which causes the burning rate to alter. Also, physical effects such as particle size distribution of the mixed materials may cause differences in the burn rate.
- Pressure: As the pressure inside the chamber increases the burning rate enhances. In the pressure range in which rocket motors operate, de Saint Robert's burning rate law is applicable. It is also possible to directly use plots of actually measured burning rates versus pressure [3].
- Temperature: It is necessary to know the burning rate sensitivity to initial propellant temperature.
- Acceleration: At acceleration levels higher than 10g, the propellant sensitivity to acceleration becomes significant.
- Internal Flow: Combustion products interact with propellant combustion and may lead local change in the burning rate. Also, the high speed of internal flow parallel to the burn surface may enhance burning, which is

called erosive burning, due to increase of heat transfer from the flame zone to the propellant surface. The pressure drop between forward and aft-end of the central port due to the increase in flow velocity also affects the burning rate, as described previously .

- Manufacturing Process: It has been widely reported that the ballistic response of solid rocket motors depends on the process used to manufacture it. Usually empirical parameters necessary to predict the performance of a motor (hump effect) are linked to the manufacturing process. Voids in the propellant, uneven distribution of propellant matrix, cracks in the propellant and mixture ratio differences in several batches are some of the problems that may be encountered [60].

Three flow solvers are proposed in this thesis. The zero-dimensional [95] and one-dimensional [92] flow solvers are previously validated codes. The last one is the three-dimensional flow solver, which is developed and validated in this work.

3.1 ZERO-DIMENSIONAL FLOW SOLVER

This program calculates the pressure-time and thrust-time history of the motor by using zero-dimensional equations of flow and empirical relations. The code requires burn area vs. web data from burnback solvers, thermo-chemical properties of propellants and dimensions of nozzle in order to achieve its purpose. The main assumptions are given as follows [95].

1. The combustion products are ideal gases.
2. Burning rate follows the empirical correlation for the pressure range described as $r_b = ap^n$ and pressure is constant throughout the motor.
3. Effects of transient mass addition and erosive burning can be neglected.
4. The chamber gases have negligible inertia.
5. Propellant burn rate may be corrected for ambient temperature of the propellant by the relation:

$$r_b = r_b(T_{ref}) \cdot \exp(0.01 \cdot \sigma_p \cdot (T - T_{ref})) \quad (3.1)$$

where $\sigma_p = (1-n)c_T$ and c_T is the propellant temperature sensitivity.

6. The properties of the gases in the chamber can be found by using weighted averages.

3.1.1 Governing Equations

In a radial burning solid rocket motor with subsonic flow ($M < 1.0$), it can be assumed that the properties of gases are constant ($d(\)/dx = 0$) along the grain length. The total pressure at the throat of the nozzle is also assumed to be equal to the chamber pressure.

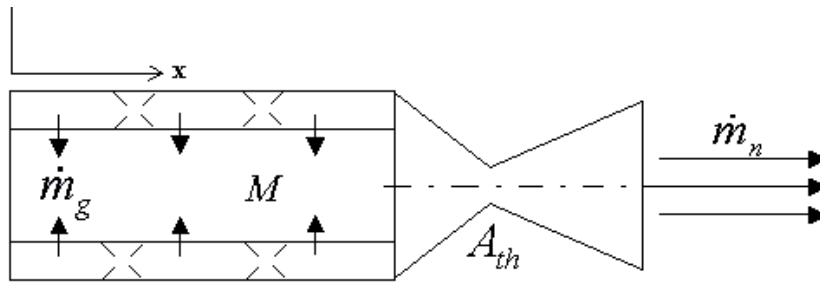


Figure 3.1 Zero-dimensional SRM Conservative Relations

The conservation of mass for isentropic flow is given by

$$\dot{m}_g = \frac{dM}{dt} + \dot{m}_n \quad (3.2)$$

where \dot{m}_g is the rate of mass addition by burning of propellant, M is the stored mass in the chamber and \dot{m}_n is the mass flow through the nozzle with $\dot{m}_n = 0$ until the pressure in the chamber reaches the closure blowout pressure.

The rate of mass generation is calculated by the relation

$$\dot{m}_g = \rho_p A_b r_b = \rho_p A_b a p_{ch}^n \quad (3.3)$$

where ρ_p is the propellant density and A_b is the burn area of the grain at that instant. The mass flow through the nozzle is calculated by the relation

$$\dot{m}_n = \frac{p_{ch} A_{th}}{C^*} \quad (3.4)$$

where p_{ch} is the chamber pressure, A_{th} is the throat area and C^* is the characteristic speed of the propellant. The throat area may change due to erosion of the nozzle insulation material and the characteristic speed is a function of chamber pressure.

The rate of change of mass stored in the chamber is given by the relation

$$\frac{dM}{dt} = \frac{d(\rho v)}{dt} = \rho \frac{dv}{dt} + v \frac{d\rho}{dt} \quad (3.5)$$

where ρ is the gas density and v is the gas volume.

Combining these equations, Equation (3.2) can be re-written as

$$\frac{dp_{ch}}{dt} = \frac{1}{v(t)} \left[RT_c \left(\rho_p A_b(t) a p_{ch}^n - \frac{p_{ch} A_{th}(t) g}{C^*} \right) - p_{ch} \frac{dv}{dt} \right] \quad (3.6)$$

where $\frac{dv}{dt} = r_b A_b$. Equation (3.6) is integrated with infinitesimal time steps as the propellant burn.

The thrust is calculated by the following simple relation,

$$Thrust(t) = C_f p_{ch}(t) A_t(t) C_d \quad (3.7)$$

where C_f is the thrust coefficient and C_d is the divergence loss coefficient.

3.2 ONE-DIMENSIONAL FLOW SOLVER

The flow solver, which is described in this section, is called IBS1D, which is developed [80] in order to predict performance of SRM, which uses empirical and analytical relations efficiently. The program requires the control volume data in the form of web vs. perimeter at stations along the axis of the motor.

3.2.1 Program Definitions

Assuming that the transient behavior of solid rocket motor consists of a small time interval, compared to the steady flow period, steady flow solutions are preferred for performance prediction purposes.

It can be assumed that the flow-field in the motor is parallel to the axis of the motor which is true with high *length-to-diameter* ratio motors. The vertical component of momentum does not have any effect on the thrust of the rocket motor. By this

assumption another simplification can be used: one-dimensional flow in a variable area duct.

The gas products are in multi-phase within a SRM. Most of the products are in gas phase, however liquid and solid particles exist in the flow. In order to keep the solution simple, ideal gas assumption is accepted.

Another simplification is to assume the flow as adiabatic and inviscid. At high flow velocities the heat transfer can be neglected. Since the gas phase has low viscosity the inviscid assumption can be acceptable. As the flow velocity increases the inviscid assumption is more acceptable.

Considering the assumptions described above, the conservation equations and equation of state may be written as:

$$d(\rho VA) = \rho_p r_b C_{per} dx \quad (3.8)$$

$$d(\rho V^2 A) + A dp = 0 \quad (3.9)$$

$$d[\rho VA(h + 0.5V^2)] = \rho_p r_b C_{per} h_0 dx \quad (3.10)$$

$$\gamma p = (\gamma - 1)h\rho \quad (3.11)$$

where C_{per} is the perimeter of the burning interface at each station, V is the velocity of the flow parallel to the axis, r_b is the propellant burn rate, ρ_p is the propellant density and A is the cross-sectional area. The above equations can be solved for p , ρ , V , and h .

3.2.2 Flow Solver Formulation

Equations (3.8), (3.9), (3.10) can be written as follows:

$$\rho VA = a \quad (3.12)$$

$$aV + \bar{A}p = b \quad (3.13)$$

$$h + \frac{1}{2}V^2 = c/a \quad (3.14)$$

where \bar{A} is the arithmetic average of port area at inlet and exit of each volume element. The parameters a , b , c are dependent on properties of combustion and flow-field. The initial values for these parameters at a reference point in the motor are assumed to be known. By using Equations (3.12), (3.13), (3.14), (3.15), h , p and ρ variables can be eliminated which leaves a quadratic equation in terms of V .

$$a \left(1 - \frac{\gamma - 1}{2\gamma} \frac{\bar{A}}{A} \right) V^2 - bV + c \left(\frac{\gamma - 1}{\gamma} \right) \frac{\bar{A}}{A} = 0 \quad (3.15)$$

The Equation (3.16) has two solutions. One solution is for the subsonic and the other solution is for the supersonic case. If the discriminant is zero, the flow is sonic.

After solving V and knowing the parameters a , b and c , the Equations (3.12), (3.13), (3.14) and (3.15) can be used to solve for ρ , p , h .

$$p = \frac{b - Va}{A} \quad (3.16)$$

$$h = \frac{c}{a} - \frac{1}{2}V^2 \quad (3.17)$$

$$\rho = \frac{a}{VA} \quad (3.18)$$

3.2.3 Solution Technique

The domain has to be discretized in order to solve for the flow variables. For this purpose, the motor is divided into segments in the axial direction (see Figure 3.2).

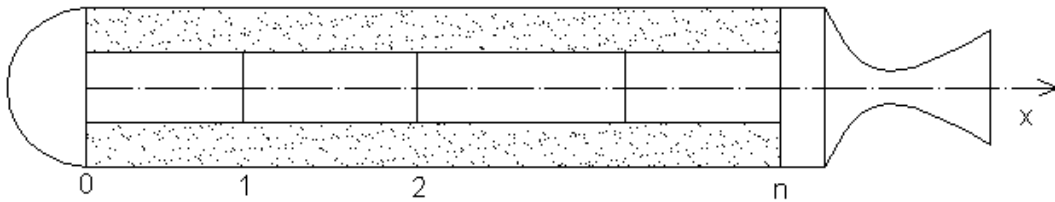


Figure 3.2 Stations for One-dimensional Solver

In order to start the solution a reference position is selected (node 0) and the flow properties at this point are supplied. This point is selected, as the head end of the motor since the properties at this location can be estimated.

Once the reference point is selected, the solution marches towards the end of the grain. The a , b , c parameters are calculated for the next point as follows:

$$a = \dot{m}_{port} + \dot{m}_{gen} \quad (3.19)$$

$$b = \dot{m}_{port} V + \bar{A} p \quad (3.20)$$

$$c = \dot{m}_{port} h + \dot{m}_{gen} h \quad (3.21)$$

where \dot{m}_{port} is the mass flow rate passing through the port area and \dot{m}_{gen} is the generated mass flow rate coming from the propellant. They are defined as:

$$\dot{m}_{port}(x + dx) = \dot{m}_{port}(x) + \dot{m}_{gen}(x) \quad (3.22)$$

$$\dot{m}_{gen} = C_{per} \rho_p r_b dx \quad (3.23)$$

$$h = c_p T_c \quad (3.24)$$

Equation (3.24) is written assuming that reference temperature is at 0 K. The calculated parameters are inserted into Equation (3.16), and V is obtained for the next point. By using Equations (3.17), (3.18), (3.19), the properties are calculated. The solution can proceed up to the end of the grain. The solution is steady up to this point.

3.2.4 Nozzle Flow and Thrust Calculations

In all SRM, a converging-diverging nozzle is used. In order to calculate the mass flow rate, it is necessary to find out if the nozzle is choked or not. If the throat is choked, then the flow rate is fixed. In this case the conditions inside the motor and the nozzle exit are not related to each other. If the throat is not choked, the mass flow rate is dependent on nozzle exit conditions.

The flow rate in a nozzle throat can be expressed as

$$\dot{m}_{th} = p_0 A^* (1/C^*) \quad (3.25)$$

In order to calculate the nozzle exit properties, the isentropic compressible flow equations can be used in order to find thrust

$$Thrust = A^* p_0 C_F \lambda \quad (3.26)$$

where

$$C_F = \left\{ \frac{2\gamma^2}{\gamma-1} \left(\frac{2}{\gamma+1} \right)^{\frac{\gamma+1}{\gamma-1}} \left[1 - \left(\frac{p_e}{p_0} \right)^{\frac{\gamma-1}{\gamma}} \right] \right\}^{1/2} + \left(\frac{p_e - p_a}{p_0} \right) \frac{A_e}{A_t} \quad (3.27)$$

where λ is nozzle divergence loss factor and.

3.2.5 Time Dependence

In this section, addition of time dependence is described by using the steady flow solution. The basic idea is to update the properties at the reference position at each time interval and steady solution at this time interval is calculated. The static pressure inside the chamber can be calculated by:

$$\frac{dp}{dt} = \frac{RT_c}{V} \left(\dot{m} - \frac{\rho}{\rho_p} \dot{m}_{gen} \right) \quad (3.28)$$

The change of mass of the control volume with respect to time is

$$\dot{m} = \dot{m}_{gen} - \dot{m}_{th} \quad (3.29)$$

By using Equation (3.29), at each time interval the pressure value at the reference point is updated.

3.3 THREE-DIMENSIONAL FLOW SOLVER

The motion of the interface is governed by the physical properties of the flow passing over the surface. In the case of solid propellant rocket motors, the burn rate by which the propellant surface regresses is affected by the static pressure of the flow by following the empirical de Saint Robert relation [94].

$$r_b = a \cdot p^n \quad (3.30)$$

where a and n are experimentally defined for a fixed propellant initial temperature.

The static pressure affecting the burn rate can be calculated locally and variation in the burn rate can be achieved on the surface of the combustion. The three-dimensional aerodynamic solver is coupled with the burnback code. The mesh required for the aerodynamic solution is obtained from the burnback code whereas the burn rate results at every node on the surface are obtained from the flow solver.

The code previously was a two-dimensional, inviscid, cell-centered finite volume, tetrahedron mesh, external flow solver [91]. The flow solver is upgraded to a three-dimensional internal flow solver, operating on hybrid mesh.

In order to obtain the flow field properties without viscosity and heat transfer considerations, Euler equations are used. In this section the governing equations are described.

3.3.1 Conservative Form of Euler Equations

In three dimensions the Euler equations can be written in differential form in terms of conservative flow variables as [88]:

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial K}{\partial z} = 0 \quad (3.31)$$

where

$$Q = [\rho \quad \rho u \quad \rho v \quad \rho w \quad \rho e_0]^T \quad (3.32)$$

where ρ is the density, u , v , w are three velocity components in x , y and z direction respectively and e_0 is the total energy defined as $e_0 = e + \frac{u^2 + v^2 + w^2}{2}$.

The flux vectors are given by

$$F = [\rho \quad \rho u^2 + p \quad \rho uv \quad \rho uw \quad \rho u h_0]^T \quad (3.33)$$

$$G = [\rho \quad \rho uv \quad \rho v^2 + p \quad \rho vw \quad \rho v h_0]^T \quad (3.34)$$

$$K = [\rho \quad \rho uw \quad \rho vw \quad \rho w^2 + p \quad \rho w h_0]^T \quad (3.35)$$

where h_o is the total enthalpy defined as $h_o = e_o + p/\rho$.

By using the ideal gas relation $p = \rho RT$, the static pressure can be calculated as:

$$p = (\gamma - 1)\rho \left[e_o - \frac{1}{2}(u^2 + v^2 + w^2) \right] \quad (3.36)$$

where γ is the specific heat ratio.

Assume that $\vec{H} = F\vec{i} + G\vec{j} + K\vec{k}$ and the conservative form of equations can be written in the gradient form as:

$$\frac{\partial Q}{\partial t} + \vec{\nabla} \cdot \vec{H} = 0 \quad (3.37)$$

Integrating the above equation over a control volume Ω and applying divergence theorem gives the integral form of governing equations.

$$\frac{\partial}{\partial t} \iiint_{\Omega} Q d\Omega + \iint_S (\vec{H} \cdot \vec{n}) ds = 0 \quad (3.38)$$

Equation (3.39) shows that the net flux into finite volume through the surfaces S is balanced with the rate of change of conservative variables in the finite volume Ω .

3.3.2 Jacobian Matrices

The governing Equation (3.38) can be expressed in quasi-linear form as follows:

$$\frac{\partial Q}{\partial t} + \frac{\partial \vec{H}}{\partial Q} \cdot \vec{\nabla} Q = \frac{\partial Q}{\partial t} + \mathbf{A} \frac{\partial Q}{\partial x} + \mathbf{B} \frac{\partial Q}{\partial y} + \mathbf{C} \frac{\partial Q}{\partial z} = 0 \quad (3.39)$$

where \mathbf{A} , \mathbf{B} and \mathbf{C} are Jacobian matrices in conservative variables.

$$\mathbf{A} = \frac{\partial F_i}{\partial Q_j} = [a_{i,j}] \quad (3.40)$$

$$= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \frac{\gamma-1}{2} V^2 - u^2 & (3-\gamma)u & (1-\gamma)v & (1-\gamma)w & (\gamma-1) \\ -uv & v & u & 0 & 0 \\ -uw & w & 0 & u & 0 \\ \frac{\gamma-1}{2} uV^2 - uh_0 & h_0 - (\gamma-1)u^2 & (1-\gamma)uv & (1-\gamma)uw & \gamma u \end{bmatrix}$$

$$\mathbf{B} = \frac{\partial G_i}{\partial Q_j} = [b_{i,j}] \quad (3.41)$$

$$= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ -uv & v & u & 0 & 0 \\ \frac{\gamma-1}{2} V^2 - v^2 & (1-\gamma)u & (3-\gamma)v & (1-\gamma)w & (\gamma-1) \\ -vw & 0 & w & v & 0 \\ \frac{\gamma-1}{2} vV^2 - vh_0 & (1-\gamma)uv & h_0 - (\gamma-1)v^2 & (1-\gamma)vw & \gamma v \end{bmatrix}$$

$$\mathbf{C} = \frac{\partial K}{\partial Q} = [c] \quad (3.42)$$

$$= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ -uw & w & 0 & u & 0 \\ -vw & 0 & w & v & 0 \\ \frac{\gamma-1}{2} V^2 - w^2 & (1-\gamma)u & (1-\gamma)v & (3-\gamma)w & (\gamma-1) \\ \frac{\gamma-1}{2} wV^2 - wh_0 & (1-\gamma)uw & (1-\gamma)vw & h_0 - (\gamma-1)w^2 & \gamma w \end{bmatrix}$$

where $V = \sqrt{u^2 + v^2 + w^2}$. It is complicated to obtain eigenvalues in conservative form, therefore primitive variable form of Euler equations will be obtained.

Similar to conservative matrix Q let us define $P = [\rho \ u \ v \ w \ p]^T$ in terms of primitive variables. The transformation matrices are defined as:

$$\mathbf{M} = \frac{\partial Q}{\partial P} = \tag{3.43}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ u & \rho & 0 & 0 & 0 \\ v & 0 & \rho & 0 & 0 \\ w & 0 & 0 & \rho & 0 \\ \frac{V^2}{2} & \rho u & \rho v & \rho w & \frac{1}{\gamma-1} \end{bmatrix}$$

$$\mathbf{M}^{-1} = \frac{\partial P}{\partial Q} = \tag{3.44}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -u/\rho & 1/\rho & 0 & 0 & 0 \\ -v/\rho & 0 & 1/\rho & 0 & 0 \\ -w/\rho & 0 & 0 & 1/\rho & 0 \\ (\gamma-1)\frac{V^2}{2} & (1-\gamma)u & (1-\gamma)v & (1-\gamma)w & \gamma-1 \end{bmatrix}$$

By inserting the transformation matrices into Equation (3.40) the following expression is obtained:

$$\mathbf{M} \frac{\partial P}{\partial t} + \mathbf{A}\mathbf{M} \frac{\partial P}{\partial x} + \mathbf{B}\mathbf{M} \frac{\partial P}{\partial y} + \mathbf{C}\mathbf{M} \frac{\partial P}{\partial z} = 0 \tag{3.45}$$

Multiply Equation (3.46) by \mathbf{M}^{-1} from left, the governing equations are obtained in primitive form:

$$\frac{\partial P}{\partial t} + \tilde{\mathbf{A}} \frac{\partial P}{\partial x} + \tilde{\mathbf{B}} \frac{\partial P}{\partial y} + \tilde{\mathbf{C}} \frac{\partial P}{\partial z} = 0 \quad (3.46)$$

where the Jacobians of primitive variables are

$$\tilde{\mathbf{A}} = \mathbf{M}^{-1} \mathbf{A} \mathbf{M} = \begin{bmatrix} u & \rho & 0 & 0 & 0 \\ 0 & u & 0 & 0 & 1/\rho \\ 0 & 0 & u & 0 & 0 \\ 0 & 0 & 0 & u & 0 \\ 0 & \rho c^2 & 0 & 0 & u \end{bmatrix} \quad (3.47)$$

$$\tilde{\mathbf{B}} = \mathbf{M}^{-1} \mathbf{B} \mathbf{M} = \begin{bmatrix} v & 0 & \rho & 0 & 0 \\ 0 & v & 0 & 0 & 0 \\ 0 & 0 & v & 0 & 1/\rho \\ 0 & 0 & 0 & v & 0 \\ 0 & 0 & \rho c^2 & 0 & v \end{bmatrix} \quad (3.48)$$

$$\tilde{\mathbf{C}} = \mathbf{M}^{-1} \mathbf{C} \mathbf{M} = \begin{bmatrix} w & 0 & 0 & \rho & 0 \\ 0 & w & 0 & 0 & 0 \\ 0 & 0 & w & 0 & 0 \\ 0 & 0 & 0 & w & 1/\rho \\ 0 & 0 & 0 & \rho c^2 & w \end{bmatrix} \quad (3.49)$$

3.3.3 Characteristics of Euler Equations

The eigenvalues and eigenvectors of the Euler Equations can be obtained from the primitive Jacobians obtained in previous section. The primitive Jacobians can be

summed up and the relations will still hold true. Since the aim is to formulate the relations for finite volume method, it is important to calculate characteristics along a direction, letting this direction be the normal to any surface.

$$\tilde{\mathbf{D}} = \tilde{\mathbf{A}}n_x + \tilde{\mathbf{B}}n_y + \tilde{\mathbf{C}}n_z = \begin{bmatrix} V_n & \rho n_x & \rho n_y & \rho n_z & 0 \\ 0 & V_n & 0 & 0 & n_x / \rho \\ 0 & 0 & V_n & 0 & n_y / \rho \\ 0 & 0 & 0 & V_n & n_z / \rho \\ 0 & \rho c^2 n_x & \rho c^2 n_y & \rho c^2 n_z & V_n \end{bmatrix} \quad (3.50)$$

where $V_n = \vec{V} \cdot \vec{n} = un_x + vn_y + wn_z$. The hyperbolic equation system would have linearly independent eigenvectors and real eigenvalues.

The eigenvalues of matrix $\tilde{\mathbf{D}}$ can be found from the relation:

$$\det|\tilde{\mathbf{D}} - \lambda \mathbf{I}| = 0 \quad (3.51)$$

which gives

$$\begin{aligned} \lambda_1 &= \vec{V} \cdot \vec{n} \quad , \quad \lambda_2 = \vec{V} \cdot \vec{n} \quad , \quad \lambda_3 = \vec{V} \cdot \vec{n} \quad , \\ \lambda_4 &= \vec{V} \cdot \vec{n} + c \quad , \quad \lambda_5 = \vec{V} \cdot \vec{n} - c \end{aligned} \quad (3.52)$$

The left eigenvectors of $\tilde{\mathbf{D}}$ can be found by solving

$$\tilde{\mathbf{L}}\tilde{\mathbf{D}} = \mathbf{\Lambda}\tilde{\mathbf{L}} \quad (3.53)$$

where $\mathbf{\Lambda}$ is the diagonal matrix with eigenvalues λ_i

$$\mathbf{\Lambda} = \begin{bmatrix} V_n & 0 & 0 & 0 & 0 \\ 0 & V_n & 0 & 0 & 0 \\ 0 & 0 & V_n & 0 & 0 \\ 0 & 0 & 0 & V_n + c & 0 \\ 0 & 0 & 0 & 0 & V_n - c \end{bmatrix} \quad (3.54)$$

After carrying out the calculations in Equation (3.54), the following eigenvectors are obtained.

$$\tilde{\mathbf{L}} = \begin{bmatrix} 1 & 0 & 0 & 0 & -1/c^2 \\ 0 & 0 & -n_z/(n_y^2 + n_z^2) & n_y/(n_y^2 + n_z^2) & 0 \\ 0 & 1 & -n_x n_y/(n_y^2 + n_z^2) & -n_x n_z/(n_y^2 + n_z^2) & 0 \\ 0 & n_x & n_y & n_z & 1/(\rho c) \\ 0 & -n_x & -n_y & -n_z & 1/(\rho c) \end{bmatrix} \quad (3.55)$$

The inverse of $\tilde{\mathbf{L}}$ will give the right eigenvectors of $\tilde{\mathbf{D}}$ as follows:

$$\tilde{\mathbf{R}} = \begin{bmatrix} 1 & 0 & 0 & \rho/(2c) & \rho/(2c) \\ 0 & 0 & n_y^2 + n_z^2 & n_x/2 & -n_x/2 \\ 0 & -n_x & n_x n_y & n_y/2 & -n_y/2 \\ 0 & n_y & n_x n_z & n_z/2 & -n_z/2 \\ 0 & 0 & 0 & \rho c/2 & \rho c/2 \end{bmatrix} \quad (3.56)$$

The left and right eigenvectors can be used to find the diagonal vector $\mathbf{\Lambda}$ such that

$$\tilde{\mathbf{L}} \tilde{\mathbf{D}} \tilde{\mathbf{R}} = \mathbf{\Lambda} \quad (3.57)$$

3.3.4 Finite Volume Solution

The normal components of velocity can be used to calculate the flux normal to each surface of the finite volume element Ω such that

$$H_n = \vec{H} \cdot \vec{n} \quad (3.58)$$

Equation (3.39) can be written as:

$$\frac{\partial}{\partial t} \iiint_{\Omega} Q d\Omega + \iint_S H_n ds = 0 \quad (3.59)$$

In order to discretize the above equation the flow properties are assumed to be constant in the finite volume element, and the property is calculated at the centre of this element. The second integral expression in Equation (3.41) is replaced by the summation operator and the resulting equation is given as follows:

$$\Omega_i \frac{\partial Q_i}{\partial t} + \sum_{j=1}^{N_i} [H_n]_{i,j} \Delta S_{i,j} = 0 \quad (3.60)$$

where $\Delta S_{i,j}$ is the i^{th} cell's j^{th} surface area and $[H_n]_{i,j}$ is the net flux flowing into i^{th} cell from j^{th} surface of the same cell.

The residual R_i as given below:

$$R_i = \sum_{j=1}^{N_i} [H_n]_{i,j} \Delta S_{i,j} \quad (3.61)$$

By coupling the residual term with Equation (3.42) the following explicit form can be obtained.

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Omega_i} R_i^n \quad (3.62)$$

The explicit solution method requires discretization of flow equations in time and in space.

3.3.5 Time Integration by Runge-Kutta Method

The solution at time step $n+1$ can be evaluated starting from the initial data at time step n by using an explicit algorithm through Runge-Kutta (RK) Method. The general RK scheme of order m is given by:

$$\begin{aligned} Q^{(0)} &= Q^n \\ Q^{(k)} &= Q^{(0)} - \alpha_k \frac{\Delta t}{\Omega} R(Q^{(k-1)}) \quad k = 1, 2, \dots, m \\ Q^{(n+1)} &= Q^{(m)} \end{aligned} \quad (3.63)$$

where the constant $0 < \alpha_k < 1$ and $\alpha_m = 1$.

In the present flow solver, fourth order scheme is used with the standard constants

$$\alpha_1 = 1/4 \quad , \quad \alpha_2 = 1/3 \quad , \quad \alpha_3 = 1/2 \quad , \quad \alpha_4 = 1$$

It is shown [88] that for different constants of RK the stability margin can be increased. The following optimal constants are proposed to increase the CFL number.

$$\alpha_1 = 0.12 \quad , \quad \alpha_2 = 0.26 \quad , \quad \alpha_3 = 1/2 \quad , \quad \alpha_4 = 1$$

where CFL is the Courant number coefficient and defined as follows

$$CFL = \frac{\Delta t}{\Delta x} V_{\max}^n$$

V_{\max}^n is defined as the maximum wave speed throughout the domain at time level n for each cell.

3.3.6 Van Leer Flux Splitting

In Van Leer's method [84] the flux vector H_n is expressed in terms of density, speed of sound and Mach number such that

$$H_n = H_n^+(Q^L) + H_n^-(Q^R) \quad (3.64)$$

where Q^L and Q^R are conservative information coming from within the cell and its neighbor respectively.

The splitting of the mass flux for $|M| < 1.0$ is achieved by

$$f_{mass}^+ = \frac{1}{4} \rho c (1 + M)^2 \quad (3.65)$$

$$f_{mass}^- = -\frac{1}{4} \rho c (1 - M)^2 \quad (3.66)$$

The splitting of the momentum flux is achieved by cubic polynomials

$$f^+_{mom} = f^+_{mass} \frac{2c}{\gamma} \left[\frac{(\gamma-1)}{2} M + 1 \right] \quad (3.67)$$

$$f^-_{mom} = f^-_{mass} \frac{2c}{\gamma} \left[\frac{(\gamma-1)}{2} M - 1 \right] \quad (3.68)$$

The splitting of the energy flux is achieved by

$$f^{\pm}_{energy} = \frac{\gamma^2}{2(\gamma^2 - 1)} (f^{\pm}_{momentum})^2 / (f^{\pm}_{mass}) \quad (3.69)$$

For three-dimensional Euler equations, the x -split flux formula is given in vector form for $|M_n| < 1.0$ as:

$$H_n^{\pm} = \pm \frac{1}{4} \rho c (1 \pm M) \begin{bmatrix} \frac{2c}{\gamma} \left(\frac{\gamma-1}{2} M \pm 1 \right) \\ v \\ w \\ \frac{2c^2}{\gamma^2 - 1} \left(\frac{\gamma-1}{2} M \pm 1 \right)^2 + \frac{1}{2} (v^2 + w^2) \end{bmatrix} \quad (3.70)$$

If the Mach number in the normal direction to surface is $M_n \geq 1.0$, then

$$H_n^+ = H_n \quad H_n^- = 0$$

If the Mach number in the normal direction to surface is $M_n \leq -1.0$, then

$$H_n^+ = 0 \quad H_n^- = H_n$$

Van Leer[84] defined a practical stability criterion, which can be given as:

$$CFL = \frac{\Delta t}{\Delta x} (|u| + c) \leq \frac{2\gamma + |M|(3 - \gamma)}{\gamma + 3}$$

3.3.7 Riemann Solver of Roe

In this section the algorithm of original Roe's method by following Toro [89] is described. The structure of the exact solution of the Riemann Problem for the x-split three-dimensional Euler equations is given in Figure 3.3. The vectors of conserved variables and fluxes are given as:

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix} \quad \text{and} \quad H_n = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(e_0 + p) \end{bmatrix} \quad (3.71)$$

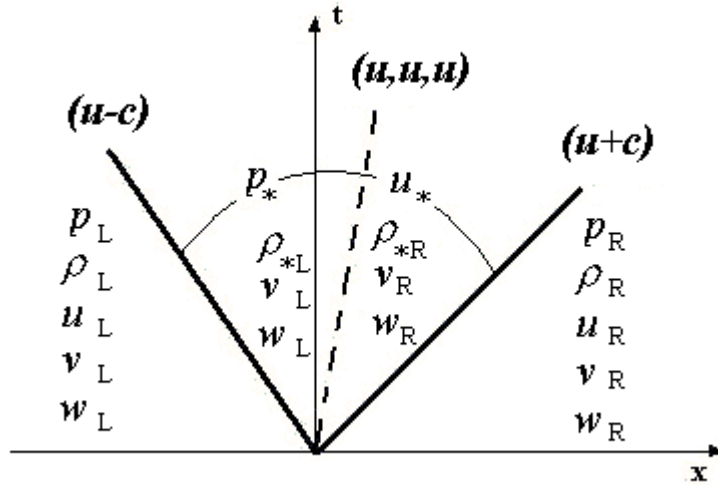


Figure 3.3 Structure of Riemann Problem for x-split Euler Equations

The piece-wise initial data is given as :

$$W_L = \begin{bmatrix} \rho_L \\ u_L \\ v_L \\ w_L \\ p_L \end{bmatrix} \quad \text{and} \quad W_R = \begin{bmatrix} \rho_R \\ u_R \\ v_L \\ w_L \\ p_L \end{bmatrix} \quad (3.72)$$

According to the original Roe's method, the inter-cell flux is given as:

$$H_n = 0.5 \cdot (H_L + H_R) - 0.5 \cdot \sum_{i=1}^m \bar{\alpha}_i |\bar{\lambda}_i| \bar{K}^{(i)} \quad (3.73)$$

The Roe averages of flow variables are given as

$$\bar{u} = \frac{\sqrt{\rho_L} u_L + \sqrt{\rho_R} u_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \quad (3.74)$$

$$\bar{v} = \frac{\sqrt{\rho_L} v_L + \sqrt{\rho_R} v_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \quad (3.75)$$

$$\bar{w} = \frac{\sqrt{\rho_L} w_L + \sqrt{\rho_R} w_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \quad (3.76)$$

$$\bar{h}_0 = \frac{\sqrt{\rho_L} h_{0L} + \sqrt{\rho_R} h_{0R}}{\sqrt{\rho_L} + \sqrt{\rho_R}} \quad (3.77)$$

$$\bar{c} = (\gamma - 1) \left(\bar{h}_0 - \frac{1}{2} \bar{V}^2 \right) \quad (3.78)$$

The eigenvalues of x-direction Jacobian matrix of A in terms of averaged values are

$$\bar{\lambda}_1 = \bar{u} - \bar{c} \quad , \quad \bar{\lambda}_2 = \bar{u} \quad , \quad \bar{\lambda}_3 = \bar{u} \quad , \quad \bar{\lambda}_4 = \bar{u} \quad , \quad \bar{\lambda}_5 = \bar{u} + \bar{c} \quad (3.79)$$

The averaged eigenvectors are

$$\bar{K}^{(1)} = [1 \quad \bar{u} - \bar{c} \quad \bar{v} \quad \bar{w} \quad \bar{h}_0 - \bar{u}\bar{c}]^T \quad (3.80)$$

$$\bar{K}^{(2)} = [1 \quad \bar{u} \quad \bar{v} \quad \bar{w} \quad \frac{1}{2} \bar{V}^2]^T \quad (3.81)$$

$$\bar{K}^{(3)} = [0 \quad 0 \quad 1 \quad 0 \quad \bar{v}]^T \quad (3.82)$$

$$\bar{K}^{(4)} = [0 \quad 0 \quad 0 \quad 1 \quad \bar{w}]^T \quad (3.83)$$

$$\bar{K}^{(5)} = [1 \quad \bar{u} + \bar{c} \quad \bar{v} \quad \bar{w} \quad \bar{h}_0 + \bar{u}\bar{c}]^T \quad (3.84)$$

The wave strengths are given as:

$$\bar{\alpha}_3 = \Delta u_3 - \bar{v} \Delta u_1 \quad (3.85)$$

$$\bar{\alpha}_4 = \Delta u_4 - \bar{w} \Delta u_1 \quad (3.86)$$

$$\bar{\alpha}_2 = \frac{\gamma - 1}{\bar{c}^2} [\Delta u_1 (\bar{h}_0 - \bar{u}^2) + \bar{u} \Delta u_2 - \bar{\Delta} u_5] \quad (3.87)$$

$$\bar{\alpha}_1 = \frac{1}{2\bar{c}} [\Delta u_1 (\bar{u} + \bar{c}) - \Delta u_2 - \bar{c} \bar{\alpha}_2] \quad (3.88)$$

$$\bar{\alpha}_5 = \Delta u_1 - (\bar{\alpha}_1 + \bar{\alpha}_2) \quad (3.89)$$

where

$$\overline{\Delta u_5} = \Delta u_5 - (\Delta u_3 - \bar{v} \Delta u_1) \bar{v} - (\Delta u_4 - \bar{w} \Delta u_1) \bar{w} \quad (3.90)$$

A modification to the above solution technique is required at sonic flow regions which is called entropy fix technique.

3.3.7.1 Entropy Fix

Linearized solutions to Riemann problem give results in discontinuous jumps which is preferable in shocks and contacts. However for expansion waves, this procedure creates errors. Roe's solver can be modified in order to avoid solutions with entropy violation. The Harten-Hyman entropy fix solution is described in this section.

At each surface the presence of a rarefaction wave is checked according to the relations given below. If a rarefaction wave is present the entropy fix process is used instead of the original Roe's method.

The left speeds are calculated as

$$\lambda_1^L = u_L - c_L \quad \text{and} \quad \lambda_1^R = u_{*L} - c_{*L}$$

where

$$\rho_{*L} = \rho_L + \bar{\alpha}_1 \quad (3.91)$$

$$u_{*L} = \frac{\rho_L u_L + \bar{\alpha}_1 (\bar{u} - \bar{c})}{\rho_L + \bar{\alpha}_1} \quad (3.92)$$

$$p_{*L} = (\gamma - 1) \cdot \left[e_{0L} + \bar{\alpha}_1 (\bar{h}_0 - \bar{u}\bar{c}) - \frac{1}{2} \rho_{*L} u_{*L}^2 \right] \quad (3.93)$$

$$c_{*L} = \sqrt{\frac{\gamma p_{*L}}{\rho_{*L}}} \quad (3.94)$$

If $\lambda_1^L < 0 < \lambda_1^R$ then the left wave is a transonic or a rarefaction wave. The following procedure is followed in such a case.

$$H_n = H_L + \hat{\lambda}_1 \bar{\alpha}_1 \bar{K}^{(1)} \quad (3.95)$$

where

$$\hat{\lambda}_1 = \lambda_1^L \left(\frac{\lambda_1^R - \bar{\lambda}_1^L}{\lambda_1^R - \lambda_1^L} \right) \quad (3.96)$$

The right speeds are calculated as

$$\lambda_5^L = u_{*R} + c_{*R} \quad \text{and} \quad \lambda_5^R = u_R + c_R$$

where

$$\rho_{*R} = \rho_R - \bar{\alpha}_5 \quad (3.97)$$

$$u_{*R} = \frac{\rho_R u_R - \bar{\alpha}_5 (\bar{u} + \bar{c})}{\rho_R - \bar{\alpha}_5} \quad (3.98)$$

$$p_{*R} = (\gamma - 1) \cdot \left[e_{0R} - \bar{\alpha}_5 (\bar{h}_0 + \bar{u}\bar{c}) - \frac{1}{2} \rho_{*R} u_{*R}^2 \right] \quad (3.99)$$

$$c_{*R} = \sqrt{\frac{\mathcal{P}_{*R}}{\rho_{*R}}} \quad (3.100)$$

If $\lambda_5^L < 0 < \lambda_5^R$ then the right wave is a transonic or a rarefaction wave. The following procedure is followed in such a case.

$$H_n = H_R - \hat{\lambda}_5 \bar{\alpha}_5 \bar{K}^{(5)} \quad (3.101)$$

where

$$\hat{\lambda}_5 = \lambda_5^R \left(\frac{\bar{\lambda}_5 - \lambda_5^L}{\lambda_5^R - \lambda_5^L} \right) \quad (3.102)$$

3.3.8 Initial and Boundary Conditions

In order to start the numerical solutions, initial conditions should be defined. In SRM the static pressure values are higher at the motor case whereas in the nozzle the pressure decreases rapidly. If both of the domains are started with the same initial pressure the convergence of the solution takes a long time for the motor case. Therefore for the domain with sub-sonic flow, high-pressure values estimated by

previous solutions are given. In the super-sonic flow region the exit pressure is assigned.

If there is no neighboring element on the surface of an element, then boundary condition implementation is applied. The boundary conditions used in this solver are injection boundary condition, subsonic flow exit boundary condition, supersonic flow exit boundary condition, symmetry boundary condition and wall boundary condition. These conditions are explained in the next sections.

3.3.8.1 Injection Boundary Condition

Zero pressure gradient at the injecting surface is assumed. The injected gas density is found through the solution of relations given below.

$$T_o = T + \frac{V^2}{2C_p} \quad (3.103)$$

$$T_o = \frac{p}{\rho R} + \frac{(G/\rho)^2}{2C_p} \quad (3.104)$$

where T_o is the flame temperature and G is the injection mass flux given as input data. Once the density is calculated, momentum flux terms can be calculated. The energy flux is calculated by using the flame temperature.

3.3.8.2 Symmetry Boundary Condition

The normal flux term passing through the surface is set to zero whereas the gradient of other flux terms are set to zero for the symmetry boundary condition. This is achieved through phantom elements on the symmetry where flux terms

tangent to the surface are kept as they are and the mirror image of the normal terms are taken.

3.3.8.3 Wall Boundary Condition for the Inviscid Flow

For an inviscid flow all the flux terms passing through the surface are set to zero, whereas the tangential flux terms are kept as they are.

3.3.8.4 Subsonic Outflow Boundary Conditions

Subsonic flow should reach the ambient pressure as the code converges. Therefore pressure value at the boundary is enforced by the ambient pressure. The velocity and temperature gradients are normal to the surface are kept zero.

$$\frac{\partial \vec{V}}{\partial \vec{n}} = 0 \quad \text{and} \quad \frac{\partial T}{\partial \vec{n}} = 0$$

The density value at the boundary should be computed with the ambient pressure and temperature.

3.3.8.5 Supersonic Outflow Conditions

Supersonic flows do not interact with outflow conditions. Thus, all the exit flow parameters are extrapolated from interior points.

3.3.9 Non-Dimensional Parameters

Given the reference states of T_{ref} , p_{ref} , other reference parameters are defined as :

$$c_{ref} = \sqrt{\gamma R T_{ref}}$$

$$\rho_{ref} = p_{ref} / R T_{ref}$$

The parameters are converted to non-dimensional form by the following relations:

$$u^* = u / c_{ref} \quad v^* = v / c_{ref} \quad w^* = w / c_{ref}$$

$$p^* = p / \rho_{ref} c_{ref}^2 \quad \rho^* = \rho / \rho_{ref} \quad T^* = T / T_{ref}$$

$$e^* = e / \rho_{ref} c_{ref}^2$$

3.3.10 Test Cases for Three-Dimensional Flow Solver

Zero-dimensional and one-dimensional solvers are validated previously. Therefore test cases present in this section are used in order to validate the three-dimensional flow solver.

The results, which are used for comparison, are validated by numerical results and experimental data of Onera [92]. For comparison purposes the flow solver of Yumusak [92] which is called IBS2D is used. The three-dimensional flow solver developed in this study is called *three-dimensional hybrid internal ballistic solver* (HIBS3D).

3.3.10.1 Residual Definitions

The definition of the residual, which is tracked as the convergence criteria, is given below:

$$L_2 = \sqrt{\sum_{i=1}^{N_{cell}} (flux(1)_i)^2} / N_{cell}$$

where N_{cell} is the number of elements in the domain and

$$flux(1) = \sum_{j=1}^6 (\rho_L \bar{u}_L + \rho_R \bar{u}_R)_j Area_j \quad (3.105)$$

for any hybrid element. The overall residual, which is used in test cases, is given as:

$$RESIDUAL = \log \left[\frac{(L_2)^n}{(L_2)^1} \right] \quad (3.106)$$

where $(L_2)^1$ is the initial L_2 .

3.3.10.2 CUBE TEST CASE

The cube test case is used to validate the solver during debugging level. The cube sides are 1000 mm. in length with the boundary conditions shown in Figure 3.4. The mass flux is $100 \text{ kg}/(\text{m}^2.\text{s})$ and outlet pressure is 100 kPa. The reference pressure and temperature are 100 kPa and 300 K respectively. The specific heat ratio is 1.4 and gas constant is $286.7 \text{ J}/\text{kg.K}$.

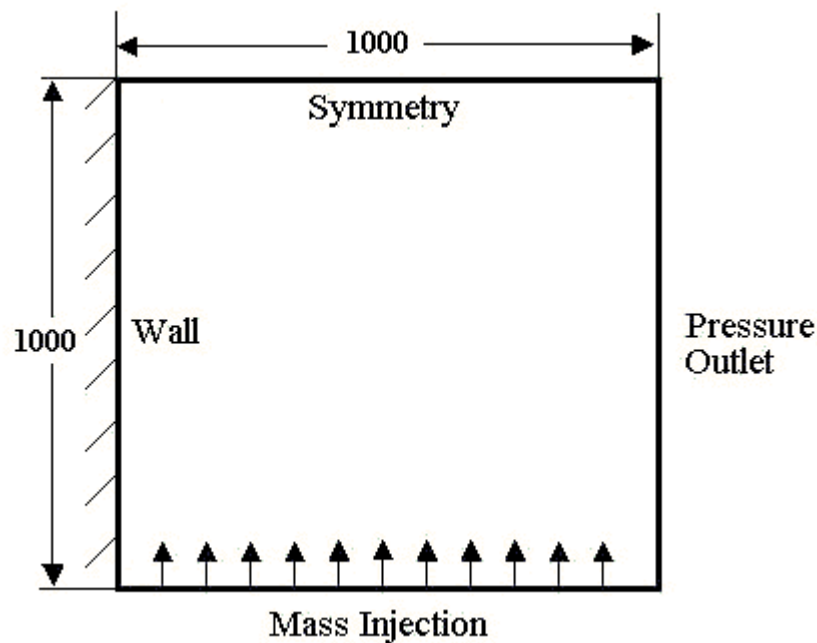


Figure 3.4 Cube Test Case Geometry

The solutions are performed on a hexahedron mesh with two intensities. First solutions are performed on a $10 \times 10 \times 10$ grid (Figure 3.5.a) and second solutions

are performed on a 20 x 20 x 20 grid (Figure 3.5 b). The IBS2D solver is second order accurate in space whereas the code which is developed in this study is first order accurate in space. The solutions used for comparison are taken from the first grid set only.

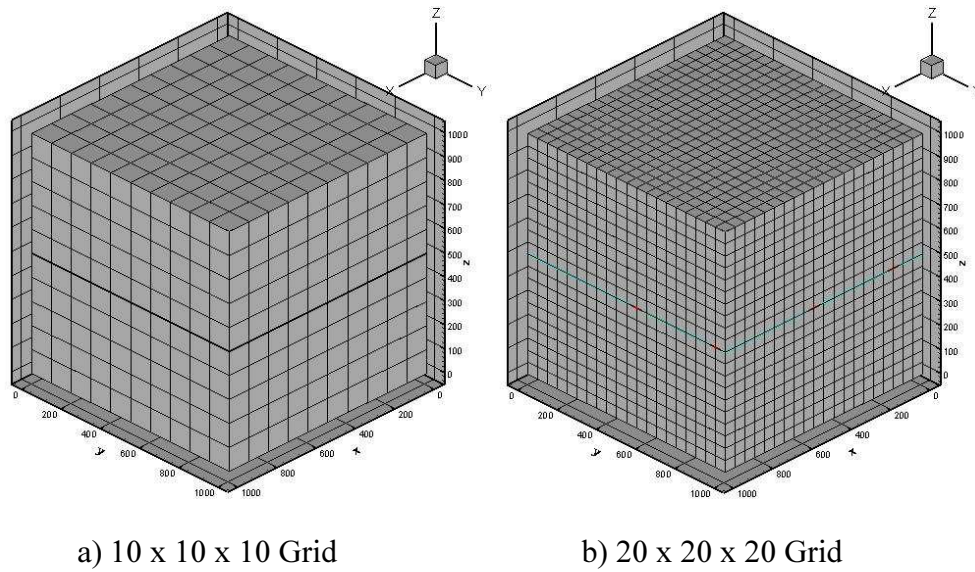


Figure 3.5 Cube Test Case Grid

The Mach contours are given in Figure 3.6 and static pressure along the symmetry axis is given in Figure 3.7 for low intensity grid. The dark line is obtained from IBS2D, thin solid line is HIBS3d with Roe's original method and the dotted line is HIBS3D with Van Leer flux splitting.

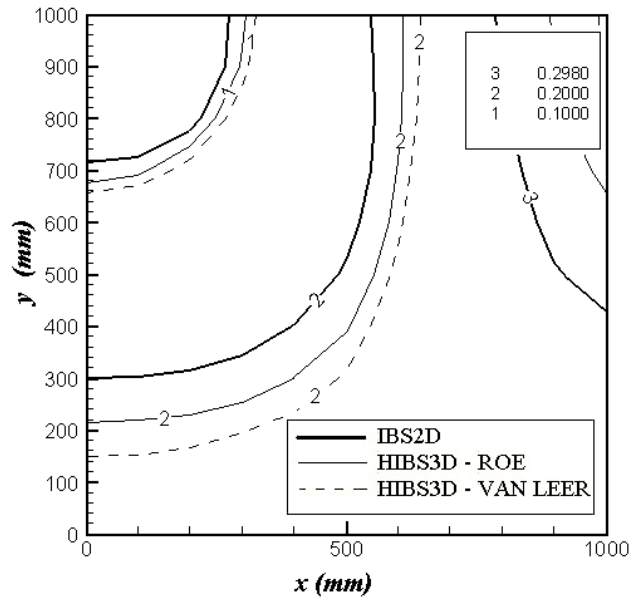


Figure 3.6 Cube Test Case (10x10x10)– Mach Contours

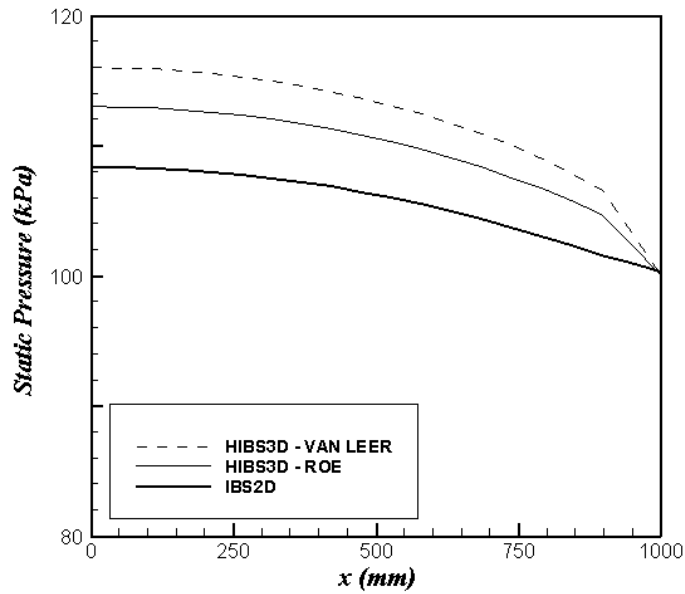


Figure 3.7 Cube Test Case (10x10x10)– Pressure at Symmetry Axis

The Mach contours are given in Figure 3.8 and static pressure along the symmetry axis is given in Figure 3.9 for high intensity grid.

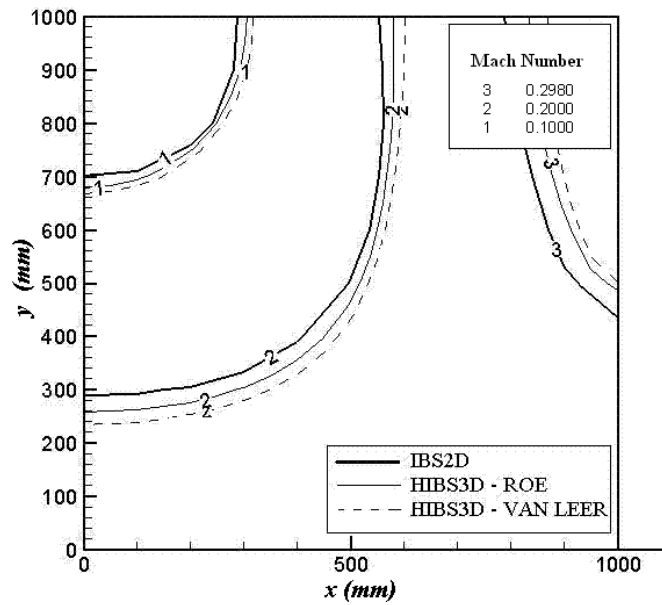


Figure 3.8 Cube Test Case (20x20x20) – Mach Contours

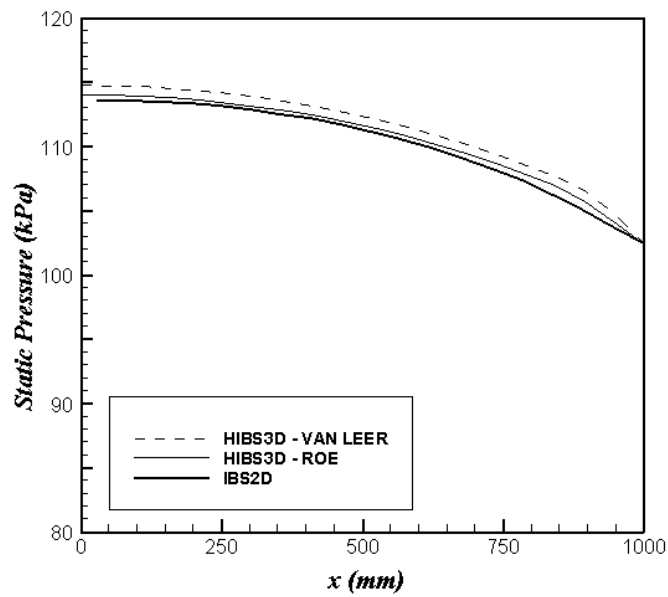


Figure 3.9 Cube Test Case (20x20x20) – Pressure at Symmetry Axis

From the results it can be seen that as the intensity of the grid increases the results are getting better. Also the Roe's method generates better results for this test case.

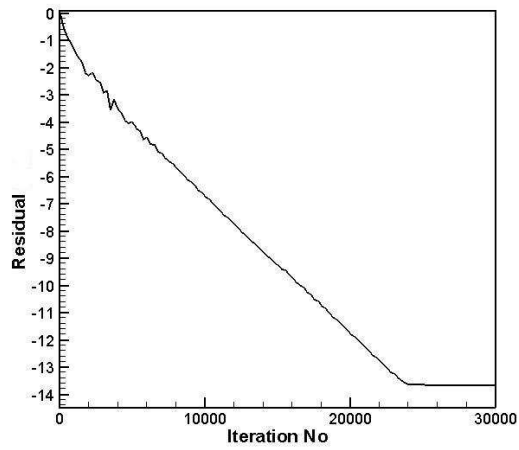


Figure 3.10 Residual for Cube Test Case (20x20x20) with Van Leer's Method

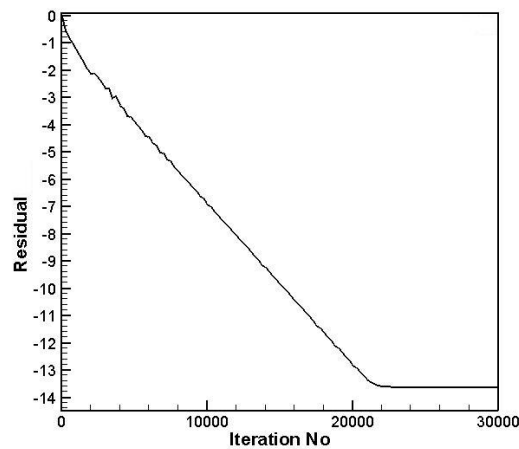


Figure 3.11 Residual for Test Case (20x20x20) with Roe's Method

The residuals are given in Figure 3.10 and Figure 3.11 for Van Leer and Roe respectively.

3.3.10.3 T-108 TEST CASE 1A

Test case 1A of T-108 project is a planar injection problem where the flow is subsonic. The geometry is given in Figure 3.12. The grid which is used by Yumusak [92] is given in Figure 3.13. The grid, which is used by our flow solver, is given in Figure 3.14.

In order to compare results with planar two-dimensional solutions, a three-dimensional grid with hexahedron elements is generated with the same initial and boundary conditions. Three-dimensional results are given as a two-dimensional slice at the centre of solution domain so that a comparison can be made.

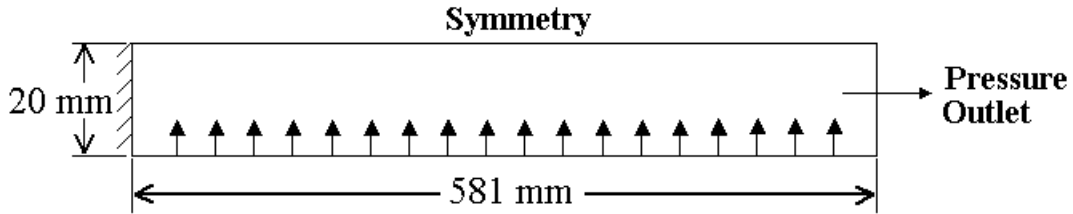


Figure 3.12 Geometry of Test Case 1

The physical properties of the flow and grid properties are given in Table 3..

Table 3.1 Geometric and Flow Properties for Test Case 1

	2-D Planar	3-D Solver
Geometric Parameters		
Length	581 mm	581 mm
Height	20 mm	20 mm
Width	-	20 mm
Grid Size	51 x 16	51 x 16 x 16
Flow Parameters		
Mass flux	2.42 kg/m ² /s	
Exit Pressure	150000 Pa	
Gamma	1.4	
Gas Constant	286.7 J/kgK	
Reference pressure	100000 Pa	
Flame Temperature	303 K	

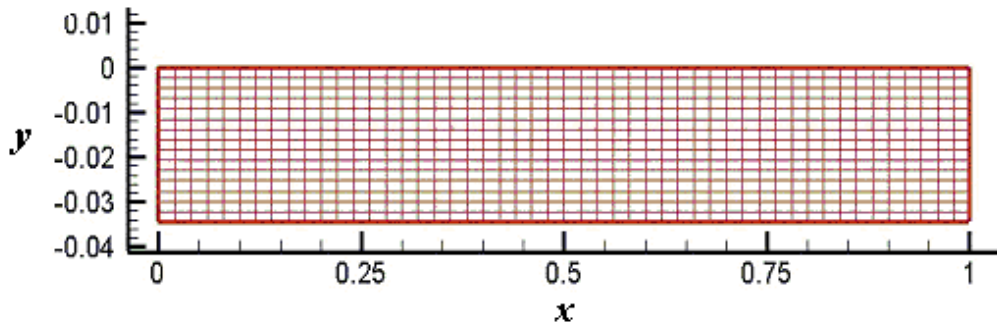


Figure 3.13 Two-dimensional Grid of Test Case 1

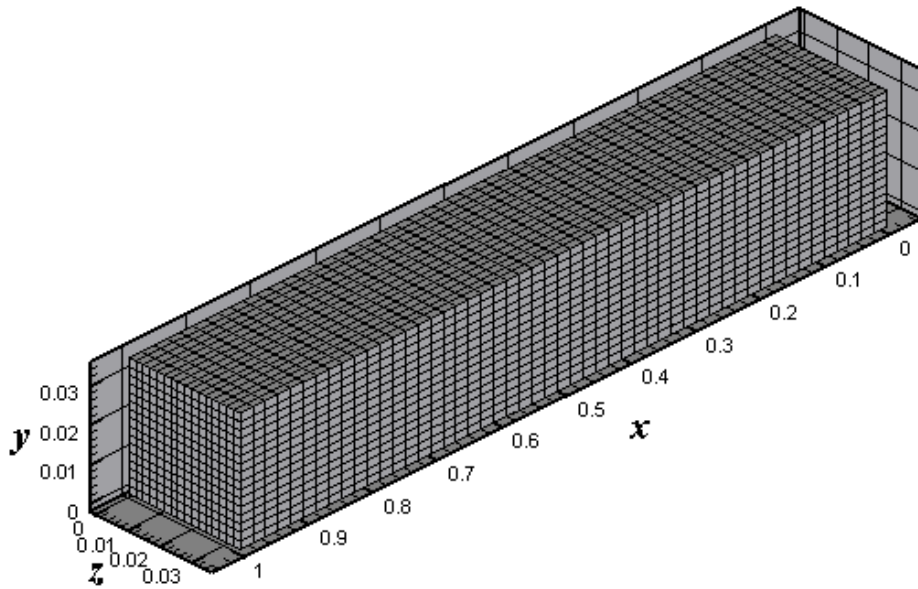


Figure 3.14 Three-dimensional Grid of Test Case 1

The Mach number contours are given in Figure 3.15 and static pressure along the x -axis is presented in Figure 3.16 for comparison.

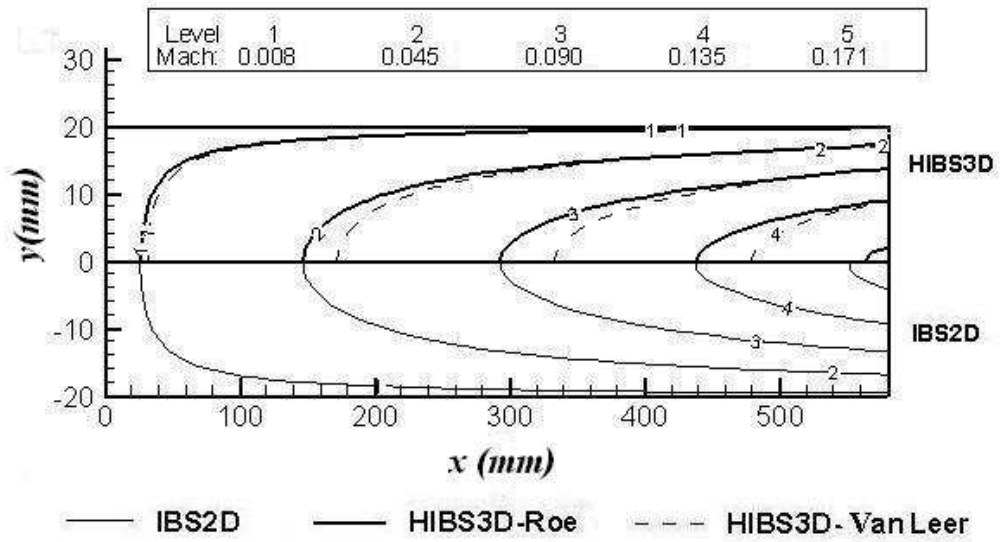


Figure 3.15 T-108 Test Case 1A- Mach Contours

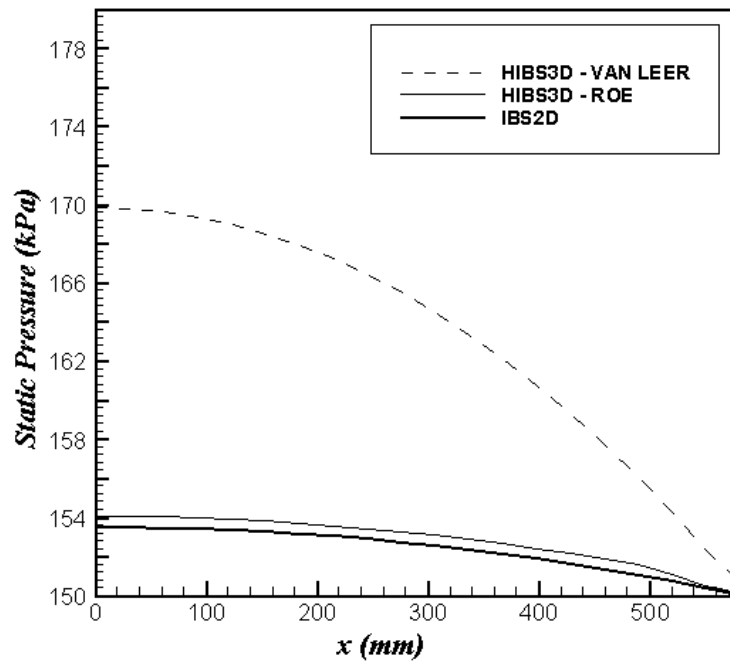


Figure 3.16 T-108 Test Case 1A - Static Pressure

The results of IBS2D and HIBS3D with Roe's method are in good agreement whereas the Van Leer flux splitting results are not satisfactory. It has been reported

that van Leer flux splitting scheme is considerably less accurate than Godunov's method with Roe's approximate Riemann solver [89]. The residuals for the solution are given in Figure 3.17 and Figure 3.18.

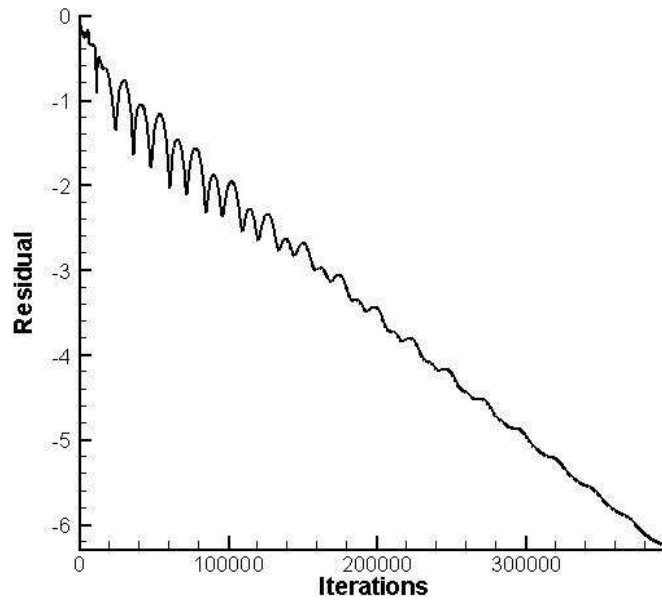


Figure 3.17 Residual for T-108 Test Case 1A with Van Leer's Method

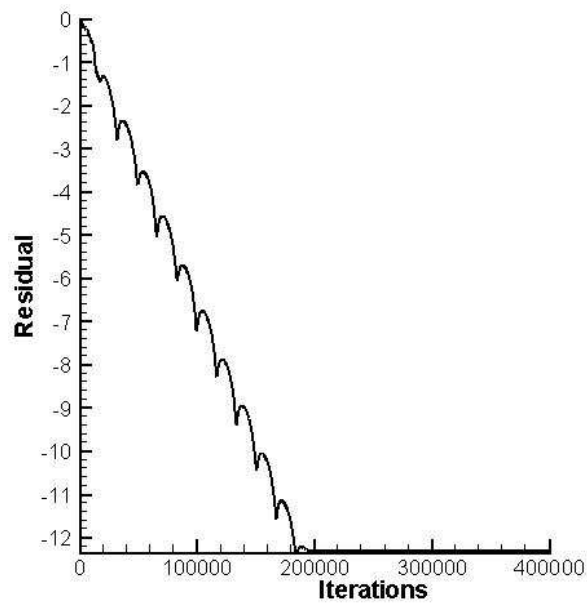


Figure 3.18 Residual for T-108 Test Case 1A with Roe's Method

3.3.10.4 T-108 TEST CASE 2

Test case 2 of T-108 project is an axi-symmetric mini SRM problem where the flow is subsonic to supersonic. The grid, which is used by IBS2D and HIBS3D, is given in Figure 3.19.

In order to compare results with axis-symmetric solutions two-dimensional solutions, a three-dimensional grid with tetrahedron elements is generated. Three-dimensional results are given as a two-dimensional slice at the x - y plane so that a comparison can be made.

The physical properties of the flow and grid properties are given in Table 3.2.

Table 3.2 Geometric and Flow Properties for Test Case 2

	2-D Axi-symmetric	3-D Solver
Geometric		
Length	270 mm	270 mm
Radius	20 mm	20 mm
Grid Size	99 x 16	7271 Nodes 32332 Elements
Flow		
Mass flux	11.39 kg/m ² /s	
Exit Pressure	100000 Pa	
Gamma	1.14	
Gas Constant	299.5 J/(kg.K)	
Reference pressure	100000 Pa	
Flame Temperature	3387 K	

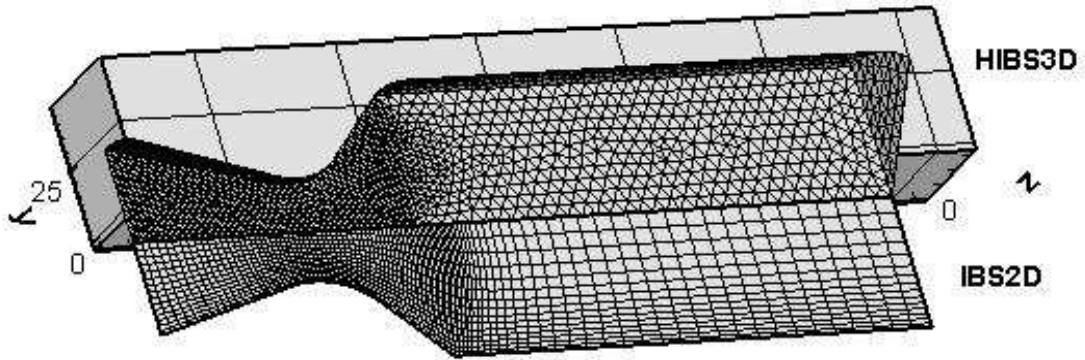


Figure 3.19 T-108 Test Case 2 – Grid Representation

The Mach number contours are given in Figure 3.20 and the pressure along the axis of symmetry is given Figure 3.21. The Van Leer's method is not considered for this case due to its inaccuracy and the solver with Roe's method is used for further applications.

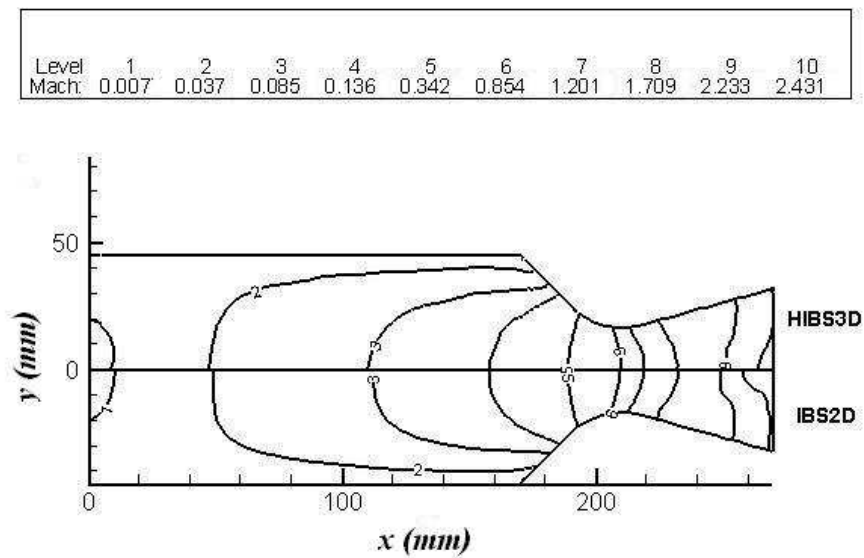


Figure 3.20 T-108 Test Case 2 – Mach Contours

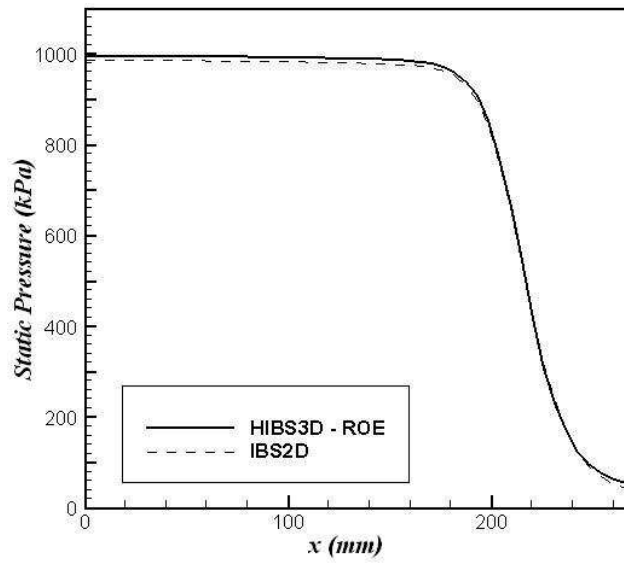


Figure 3.21 T-108 Test Case 2 – Pressure along Symmetry Axis

The convergence history for test case 2 is given in Figure 3.22.

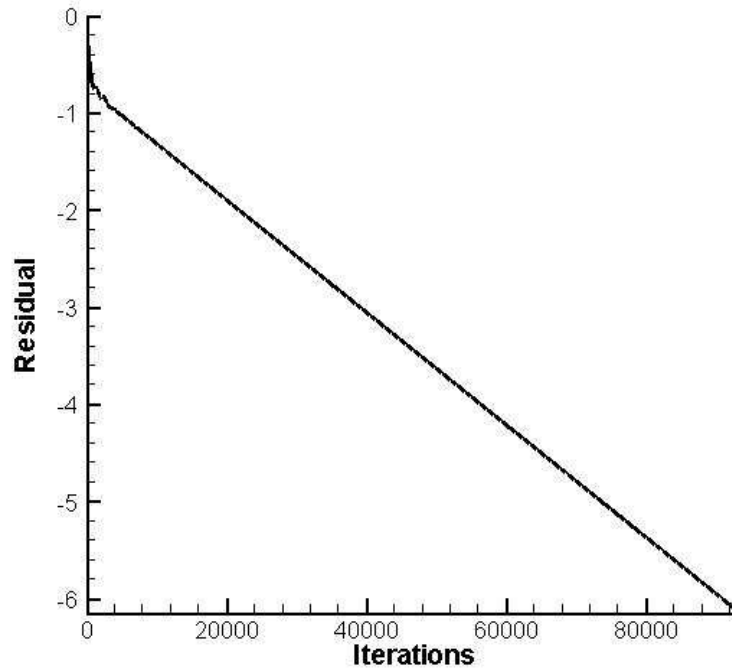


Figure 3.22 Residual for T-108 Test Case 2 – Roe's Method

CHAPTER 4

COUPLED TEST CASES AND RESULTS

The flow solutions and interface capturing solutions are coupled in a segregated manner. Each solution procedure takes turns, while the other is frozen. The following test cases are presented to show performance prediction capability of coupled solutions. The results are compared with the zero-dimensional program, one-dimensional program, analytical results and experimental data.

4.1 INCREASING INTERFACE CAPTURING ACCURACY

The interface solutions require high intensity grid whereas the flow solutions require less intense grid. Therefore the grids which are used by the interface capturing and the flow solvers are different. The results obtained from the interface solutions are transferred to the grid where flow solutions will be carried out.

Also the grid, where interface solutions are carried out, is separated. Since at location where high curvature and discontinuities exist, the solution requires high resolution grid. However if the surface is smooth the intensity requirement decreases rapidly. The problem is solved by separating the problem into several segments with different mesh resolution.

An algorithm based on the volume information of a cell is used to couple each segment with flow grid. The grid domains are placed on top of each other where

nodes do not coincide. The nodes in the flow domain are questioned in order to find in which cell in the interface modeling domain they reside in. This is done through forming tetrahedrons by connecting nodes in each cell of the interface grid, with the node under questioning. The procedure is shown in Figure 4.1.

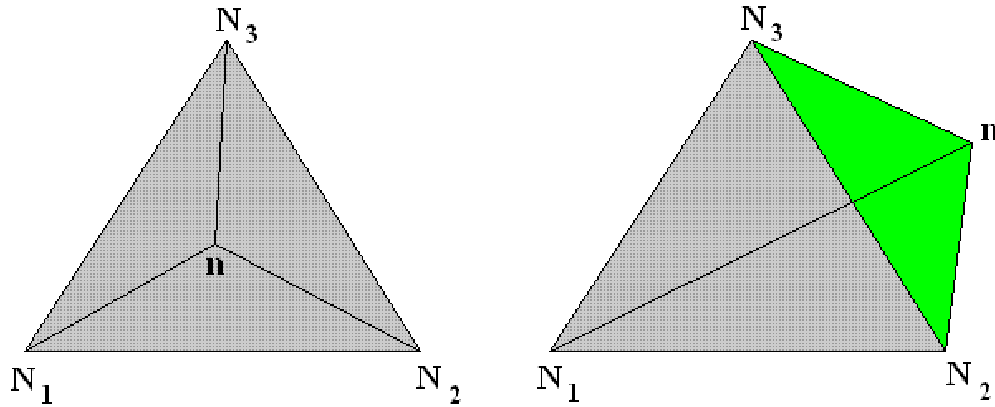


Figure 4.1 Locating Node Position

If the summation of the volumes of newly generated tetrahedrons is equal to the cell's original volume then the node is inside the element. If the summation of the volumes is greater than the original tetrahedron volume, the node is outside the element. Once locating the cells of the interface grid, in which the nodes of the flow grid are residing, interpolation according to a distance function is performed to transfer the interface arrival time.

The procedure is presented with an example which is shown in Figure 4.3. Similar to Test Case 1 of Chapter 2, a sharp corner is required to be solved whereas most of the domain has flat surfaces. The interface solutions are carried out in two different grids as shown in Figure 4.2. Once each solution is obtained for the interface modeling, interface arrival times are transferred to the flow grid.

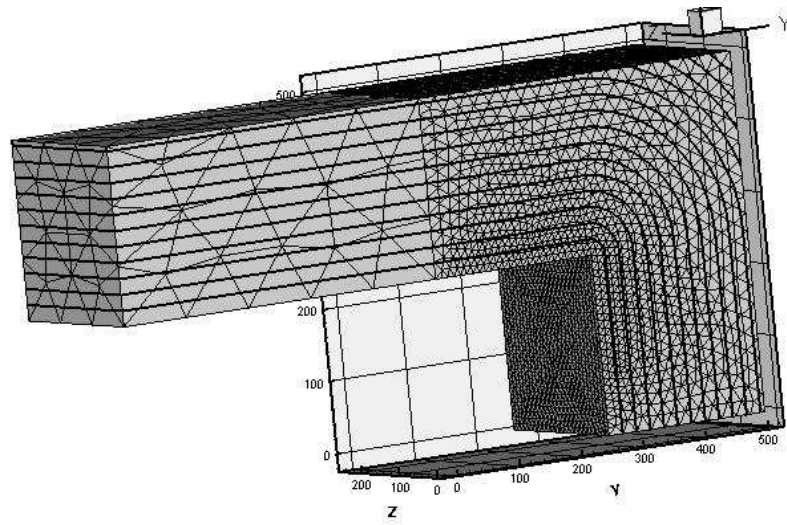


Figure 4.2 Segregated Interface Capturing

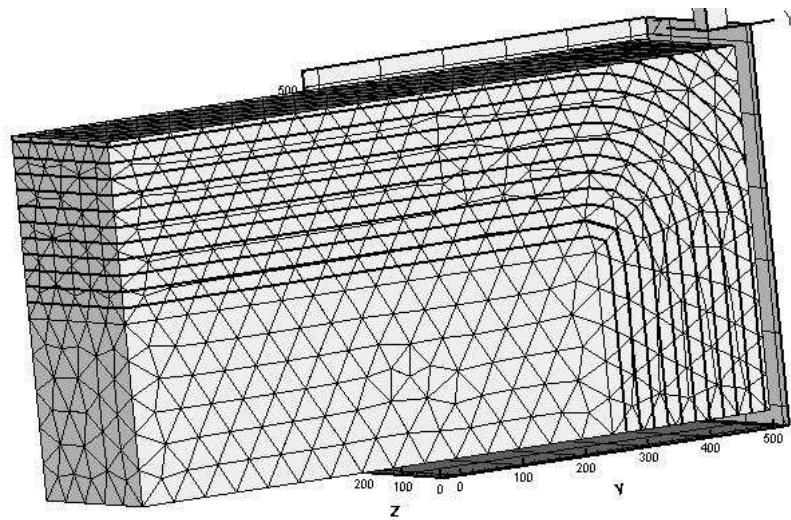


Figure 4.3 Coupled Interface Solution on Flow Domain

4.2 RELATED PARAMETERS

In order to define thermo-chemical parameters at least *specific heat at constant pressure* (C_p), *characteristic speed* (C^*) and *molecular mass* (M_m) is required. Given these values, other related parameters can be calculated as given in Table 4.1.

Table 4.1 Thermo-chemical Parameters

Gas constant per unit weight	$R = \frac{R_U}{M_m}$
Universal gas constant	$R_U = 8.314 \text{ kJ.kmol}^{-1}.\text{K}^{-1}$
Specific heat at constant volume	$C_v = C_p - R$
Specific heat ratio	$\gamma = C_p / C_v$
Flame temperature	$T_o = C^* \frac{\gamma}{R} \left/ \left(\frac{\gamma+1}{2} \right)^{(\gamma+1)/(\gamma-1)} \right.$

The burn rate is defined through the empirical relation of Saint Robert [1], which is given as:

$$r_b = ap^n \quad (4.1)$$

The pressure exponent is calculated by using two motor firing with constant pressure such that:

$$r_{b,1} / r_{b,2} = (p_{ch,1} / p_{ch,2})^n \quad (4.2)$$

When the pressure of the chamber and burn rate values are experimentally obtained, by using Equation (4.2) the pressure exponent n can be obtained. By using Equation (4.1) and one of the firing data the constant a , can be obtained. The pressure and burn rate which are used while obtaining the constant a , are called *reference pressure* and *reference burn rate*.

4.3 TEST CASE 1 – 6C4 TEST MOTOR

The 6C4 test motor has a cylindrical grain with two burning ends 4" inner diameter and outer diameter of 6". The test motor is used to analyze properties of the propellant at constant pressure. The grain geometry which is given in Figure 4.4 produces a neutral burning profile. By using different geometry nozzles the pressure adjustments can be performed.

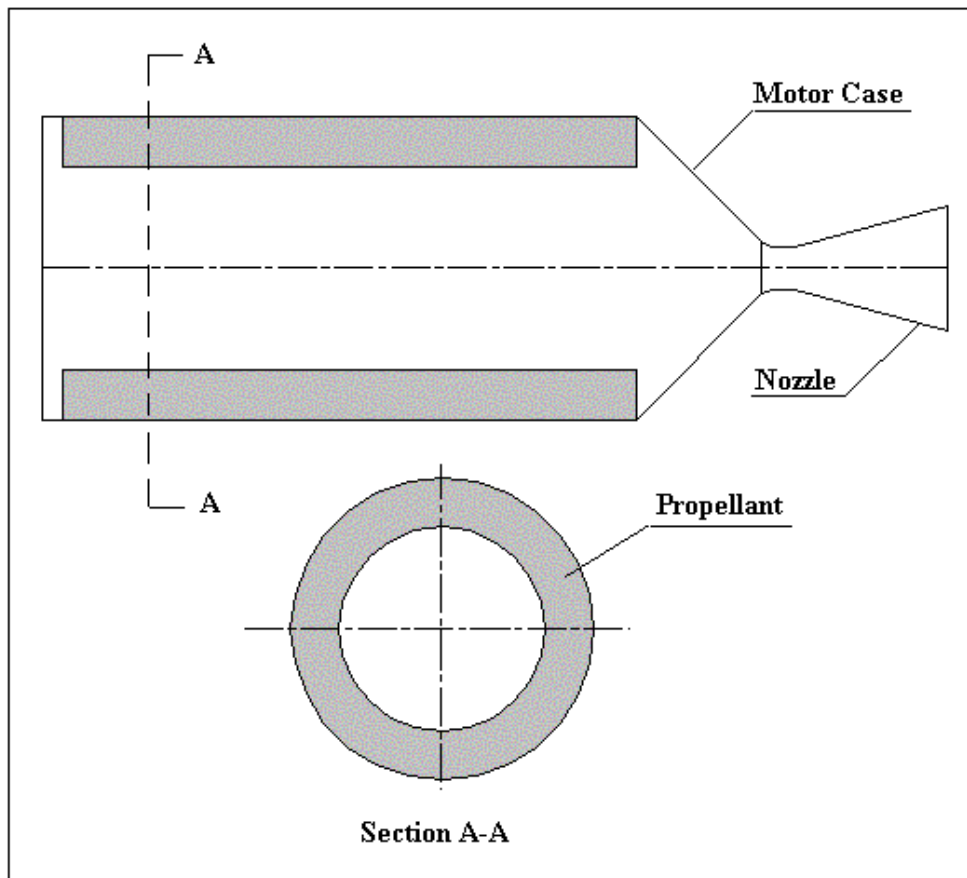


Figure 4.4 6C4 Test Motor

The Fast Marching algorithms are used to solve the interface motion and the hybrid flow solver is used to obtain flow parameters at each time interval. The properties of the propellant are given in Table 4.2 and mesh files are presented in Figure 4.5.

Table 4.2 Input Parameters for 6C4 Motor

Geometric Properties	Unit	
Outer Radius of Propellant	mm	76.2
Inner Radius of Propellant	mm	50.8
Length of Propellant	mm	285.0
Propellant Properties		
Flame Temperature	K	3078
Mass flux (at 3.0 MPa)	kg/m ² .s	16.8
Mass flux (at 6.0 MPa)	kg/m ² .s	21.8
Mass flux (at 9.0 MPa)	kg/m ² .s	25.4
Ratio of Specific Heat		1.169
Gas Constant	J/kg.K	319.8

The mesh file is generated by tetrahedron elements and the domain is divided into three parts; nozzle, propellant and subsonic flow side. The mesh has 10002 nodes and 45518 elements.

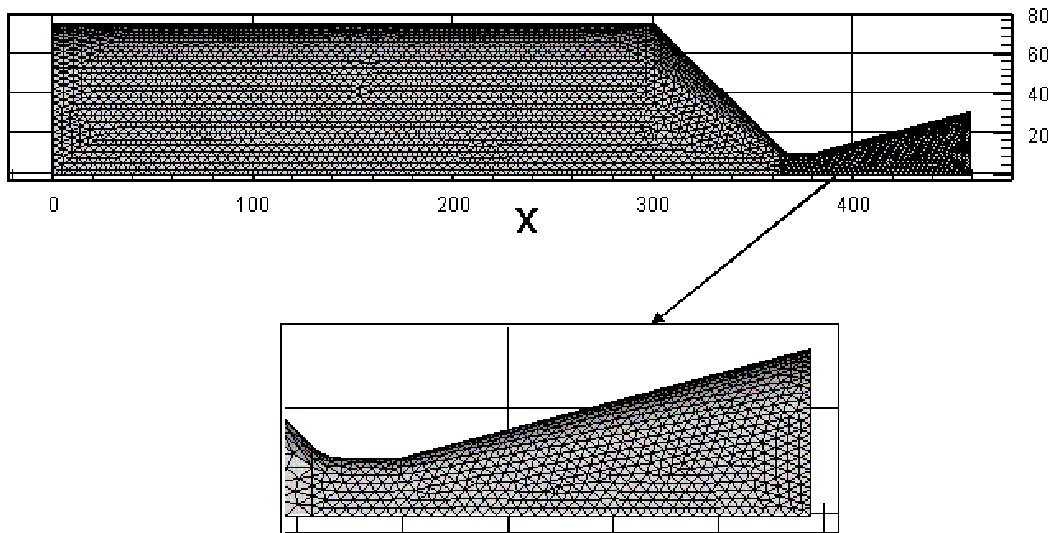


Figure 4.5 6C4 Flow Mesh

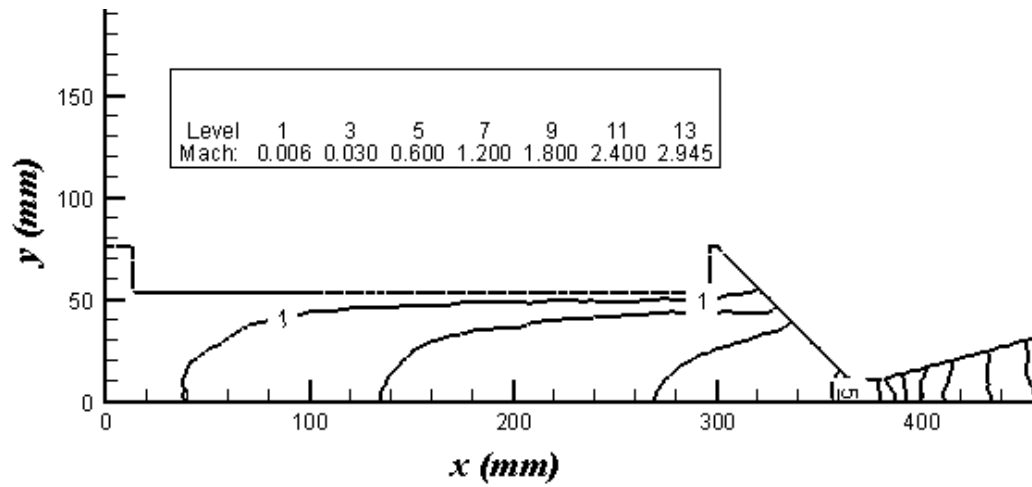


Figure 4.6 6C4 Mach Contours at 0.2 s.

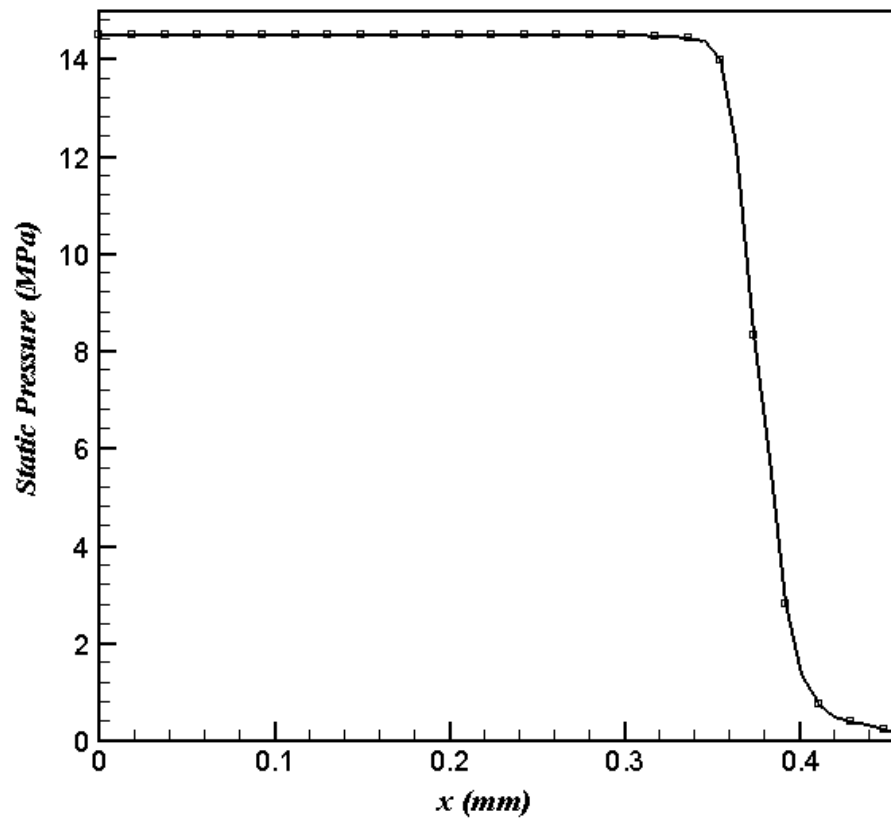


Figure 4.7 6C4 Pressure Along the x-axis at 0.2 s.

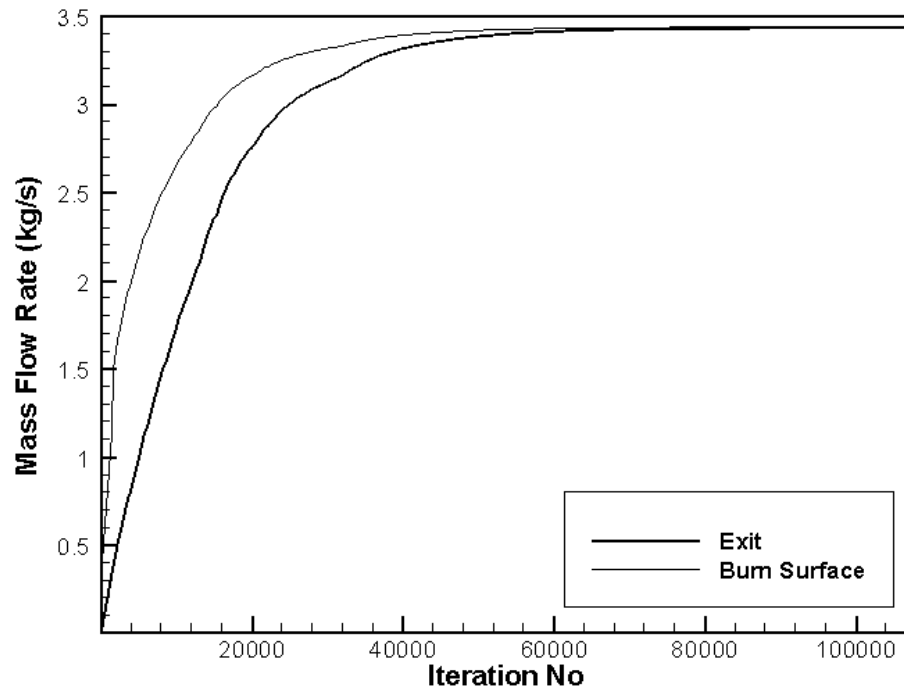


Figure 4.8 6C4 Mass Flow Rate Convergence at 0.2 s.

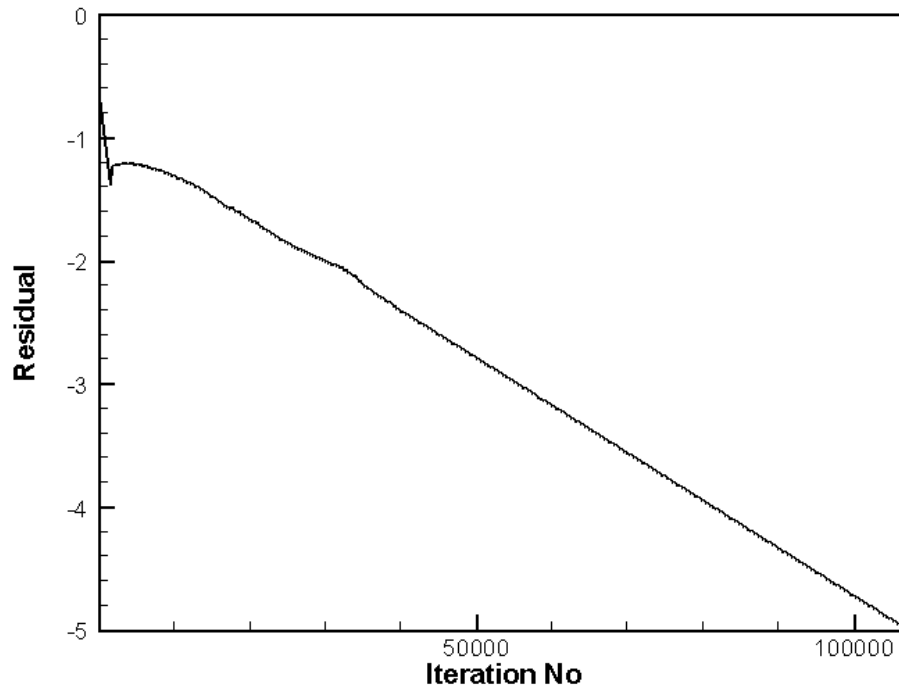


Figure 4.9 6C4 Convergence History at 0.2 s

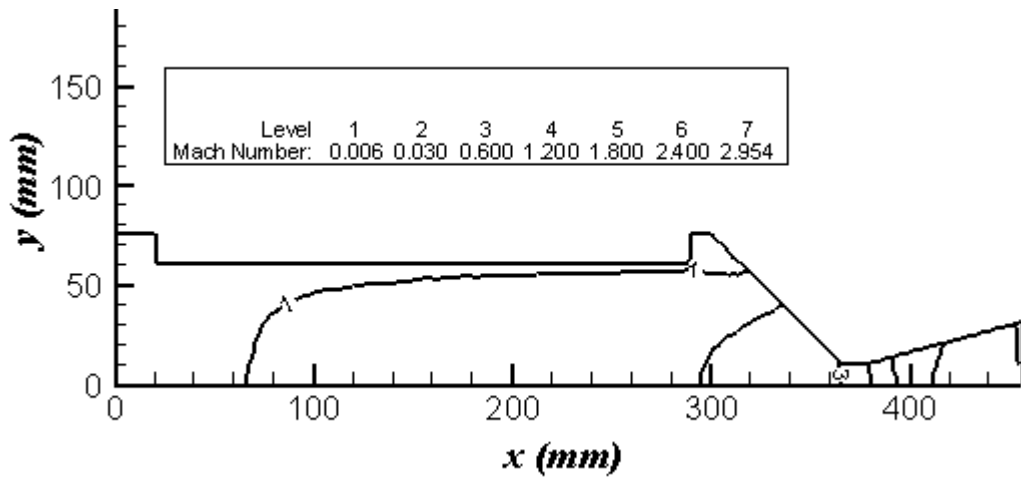


Figure 4.10 6C4 Mach Contours at 0.6 s.

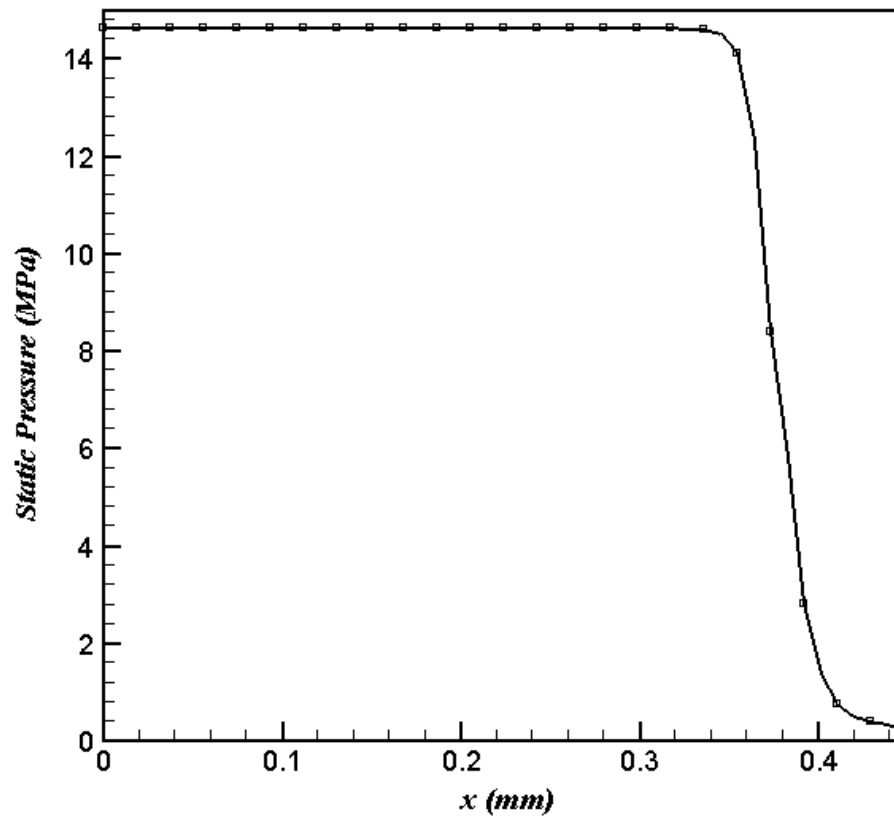


Figure 4.11 6C4 Pressure Along the x-axis at 0.6 s.

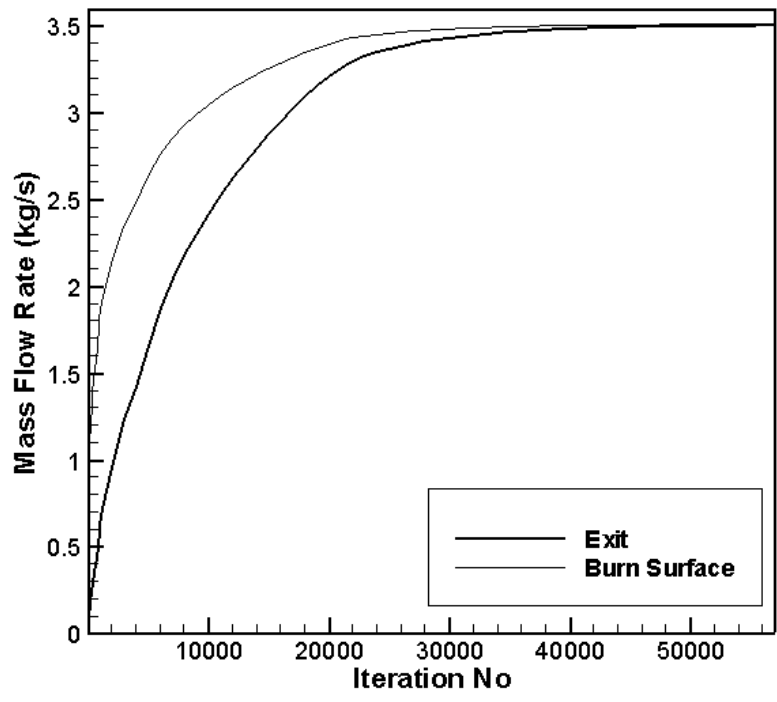


Figure 4.12 6C4 Mass Flow Rate Convergence at 0.6 s.

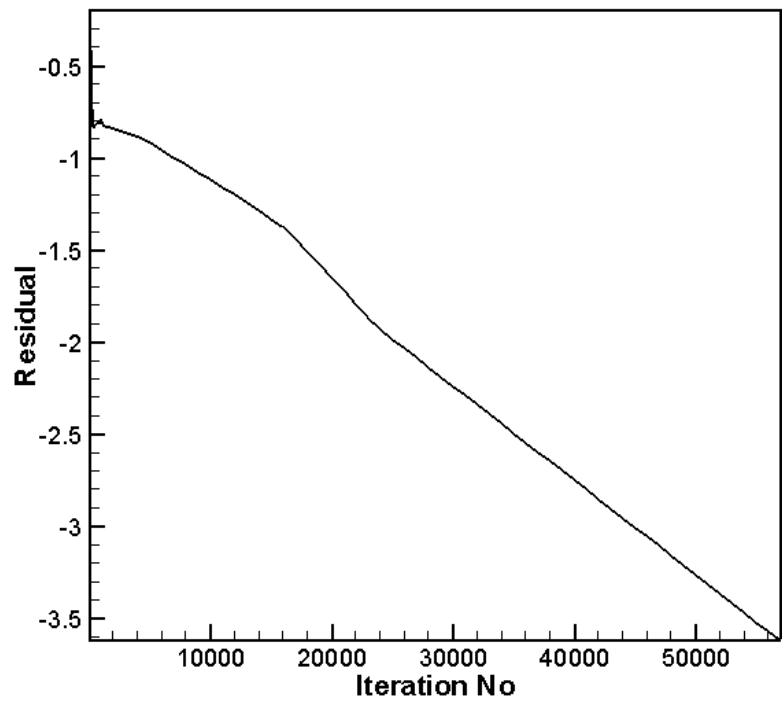


Figure 4.13 6C4 Convergence History at 0.6 s

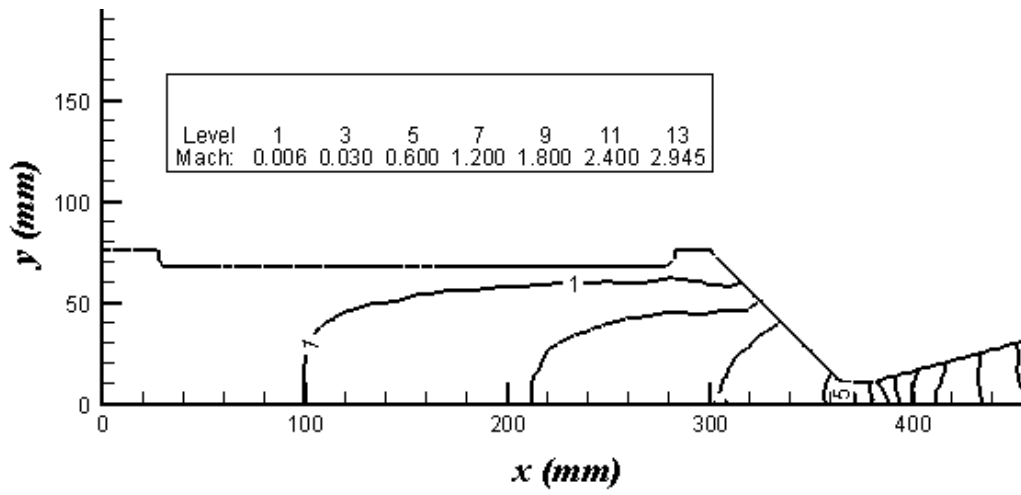


Figure 4.14 6C4 Mach Contours at 1.0 s.

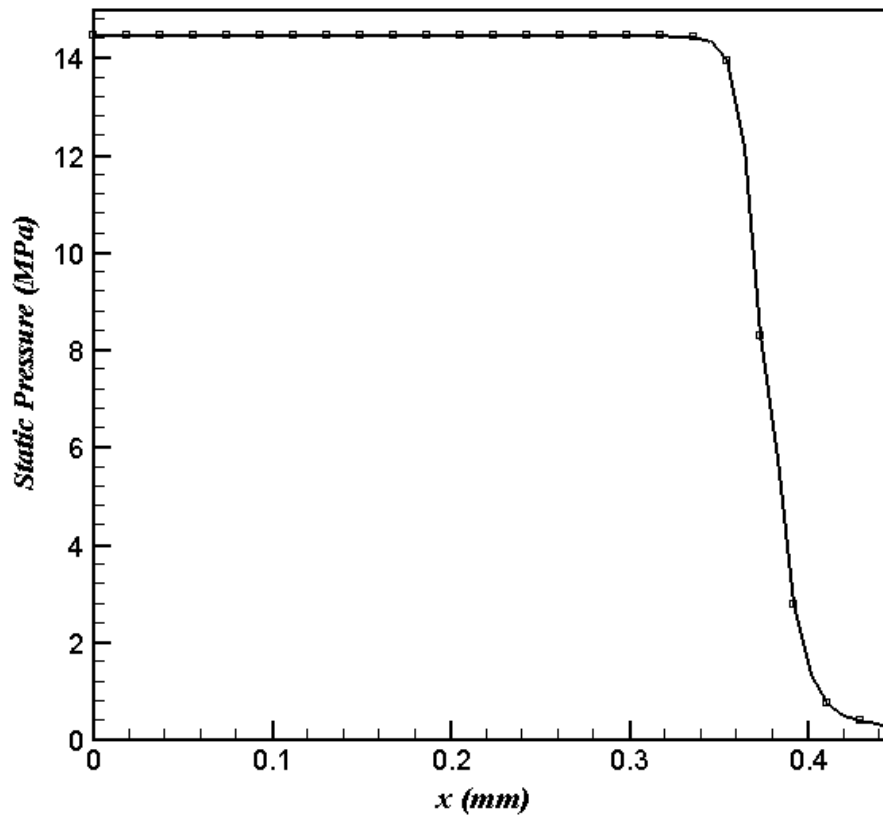


Figure 4.15 6C4 Pressure Along the x-axis at 1.0 s.

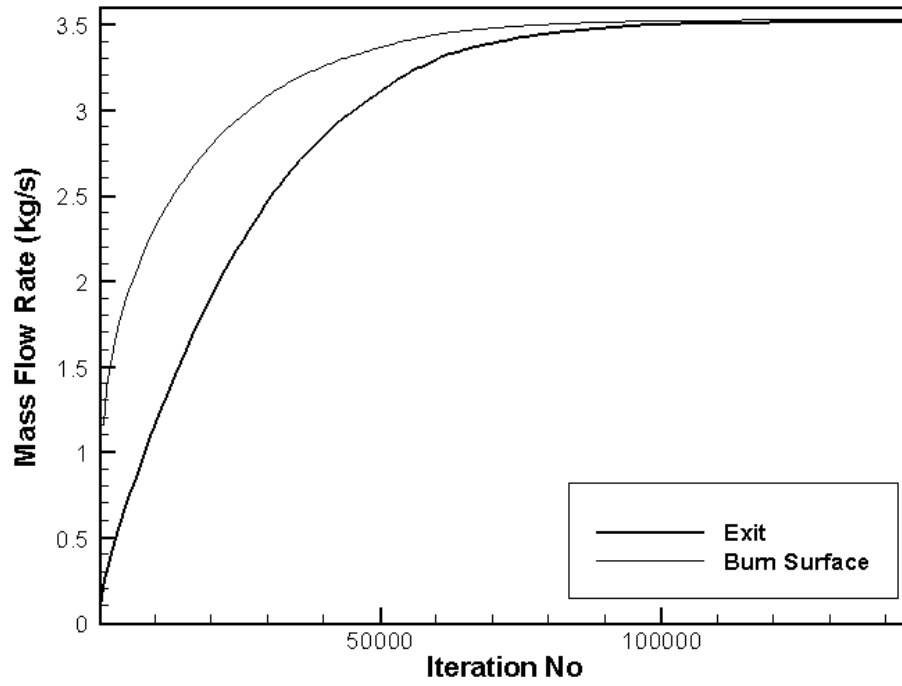


Figure 4.16 6C4 Mass Flow Rate Convergence at 1.0 s.

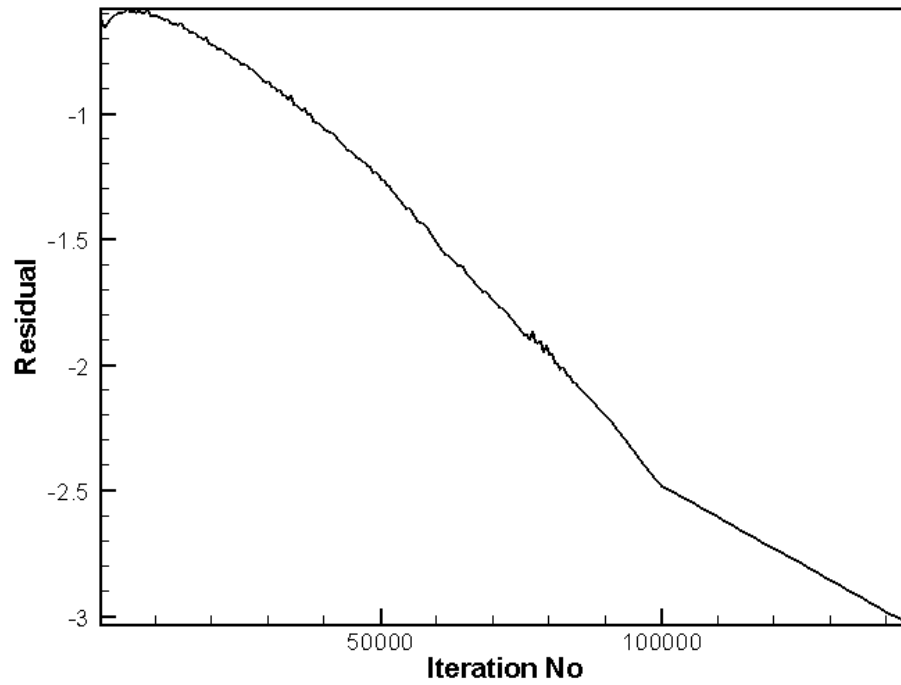


Figure 4.17 6C4 Convergence History at 1.0 s

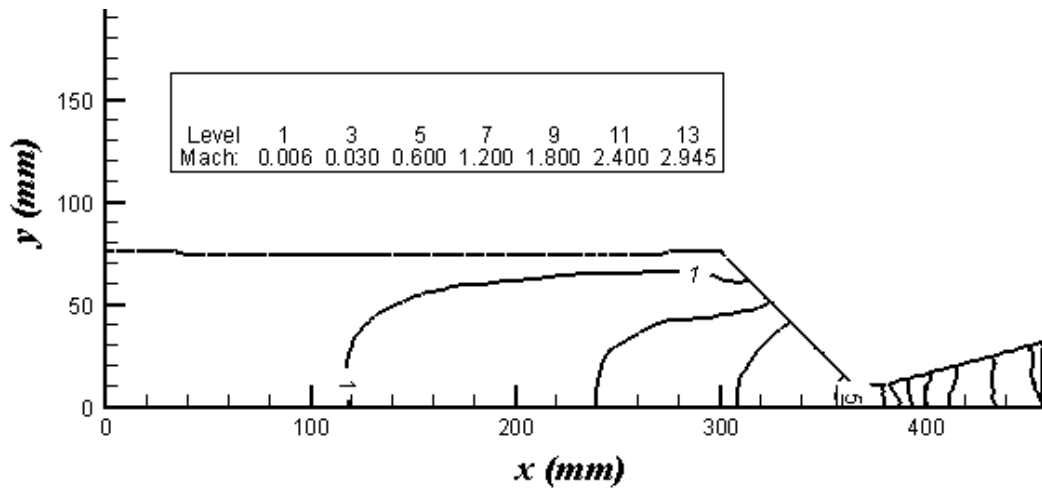


Figure 4.18 6C4 Mach Contours at 1.4 s.

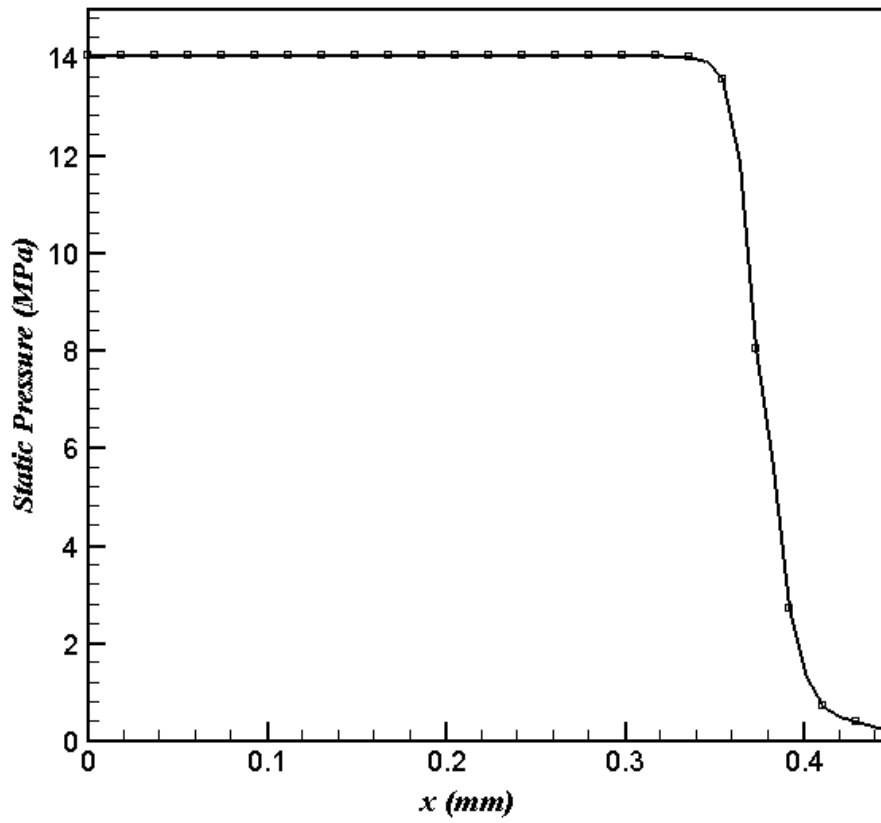


Figure 4.19 6C4 Pressure Along the x-axis at 1.4 s.

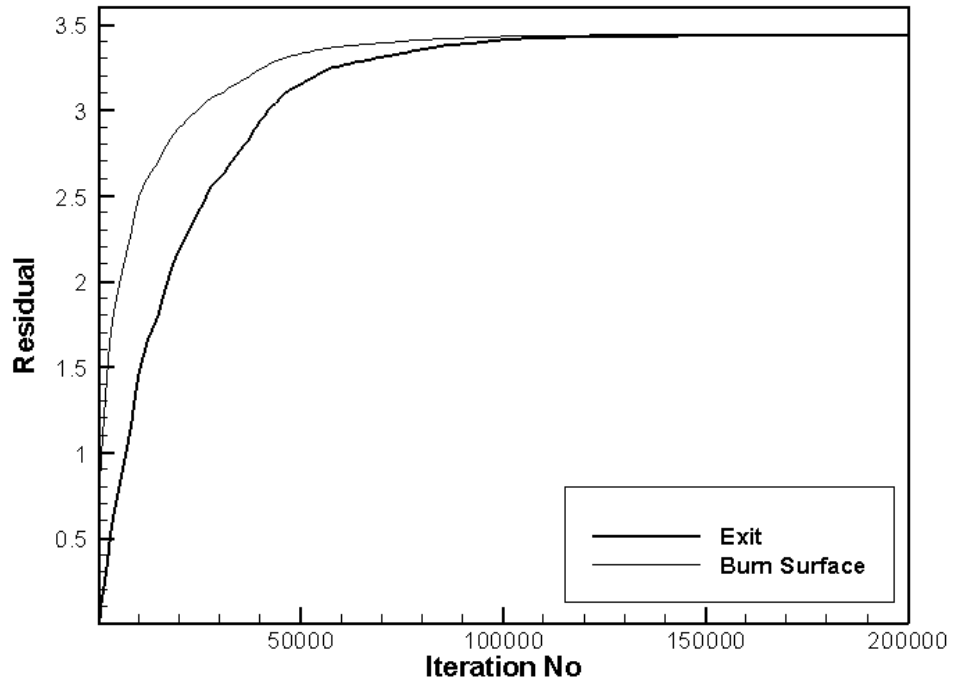


Figure 4.20 6C4 Mass Flow Rate Convergence at 1.4 s.

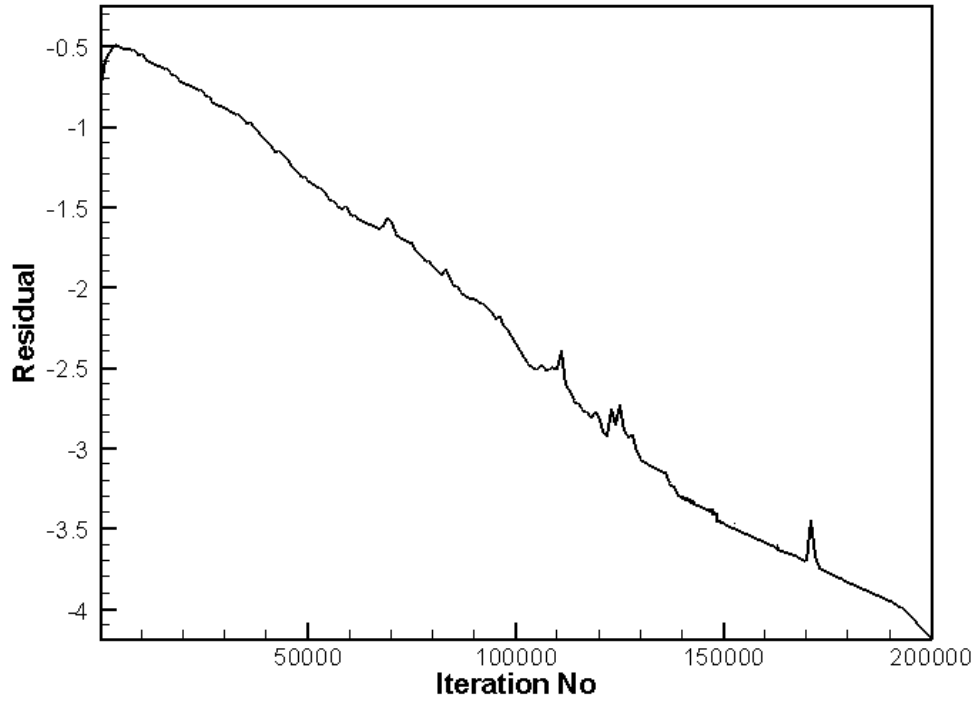


Figure 4.21 6C4 Convergence History at 1.4 s

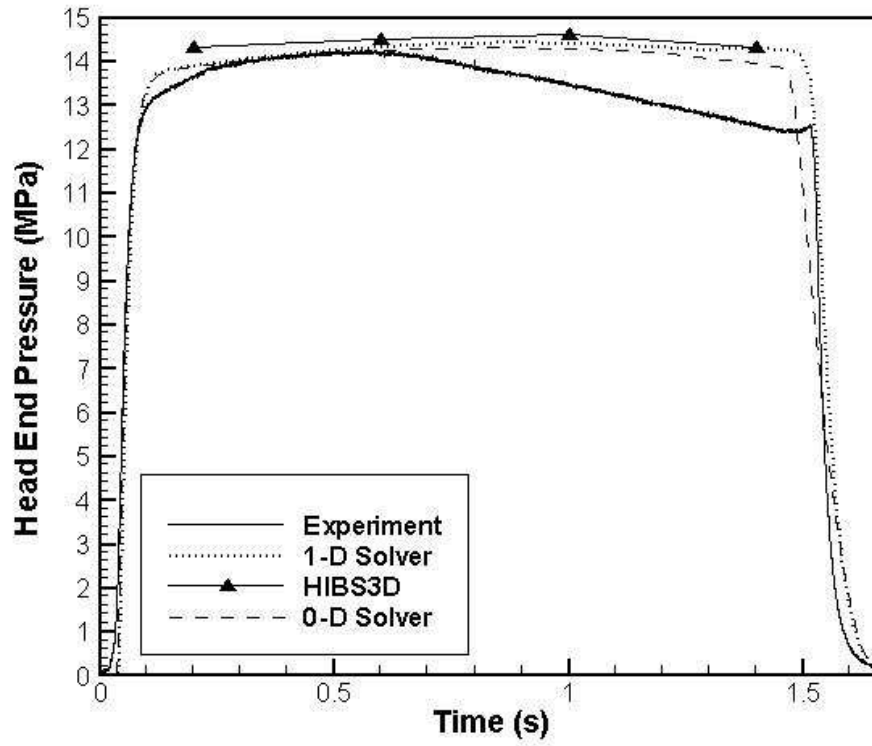


Figure 4.22 6C4 Head End Pressure versus Time Prediction

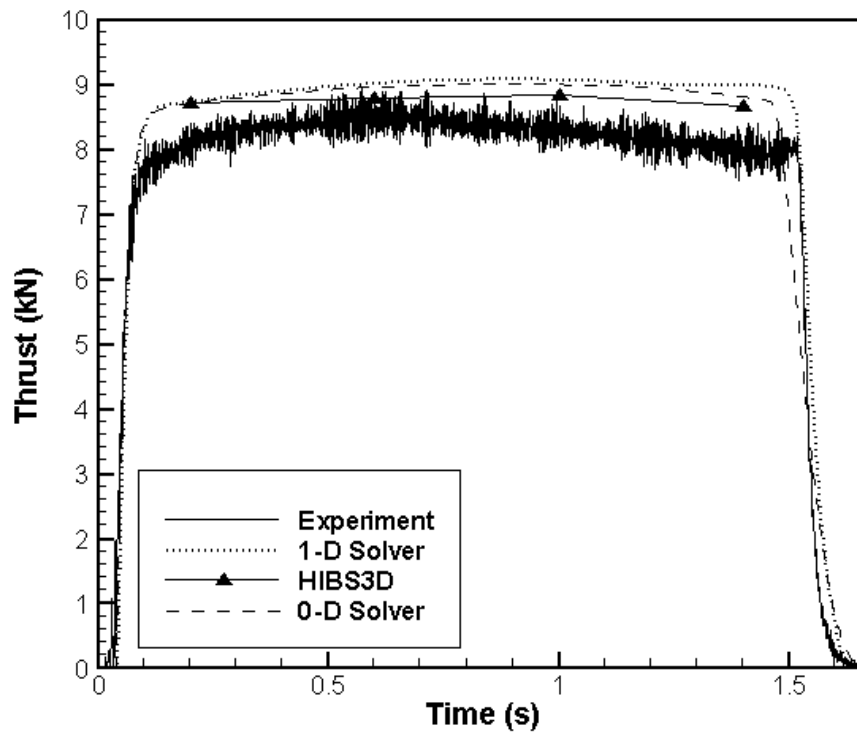


Figure 4.23 6C4 Thrust versus Time Prediction

The thrust is obtained from cell surfaces which are at the pressure outlet boundary and it is calculated by the following relation:

$$Thrust_i = \dot{m}V_n + A_i(P_i - P_{outlet}) \quad (4.3)$$

The total thrust is calculated as the summation of thrust coming from all the cell surfaces. It is observed that the thrust levels are in good agreement with the test results and with the prediction codes. The pressure levels are higher than the test results and prediction codes' estimations.

The nozzle throat area is given to the prediction codes exactly, however the grid intensity defines the throat area for the three-dimensional solver. Finite volumes generated at the nozzle throat do not follow the real shape, thus the throat is narrower. A test case is solved with different mesh intensity at the throat with constant mass flow rate. The result is given in Figure 4.24 and the grid size of 1 mm is used in nozzle throat for the present calculations.

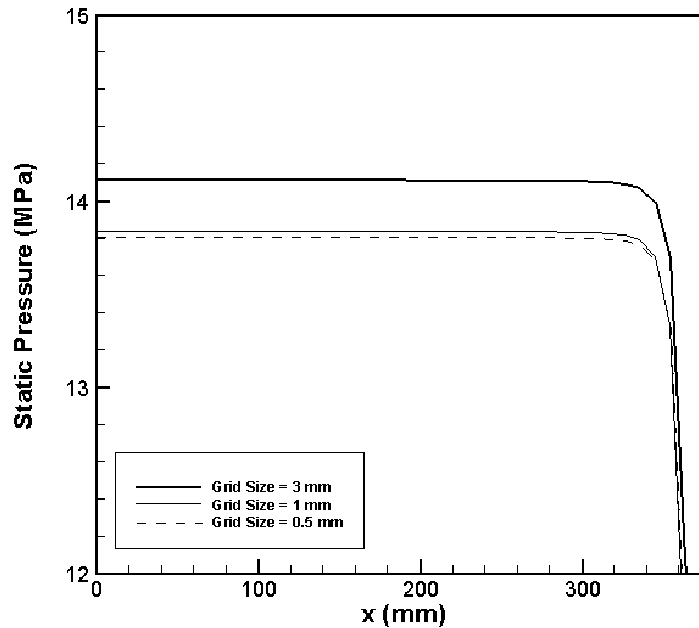


Figure 4.24 Effect of Mesh Intensity at the Throat

During the firing of the motor, the nozzle throat expands as the hot gas erodes and ablates material from the surface. This effect is excluded in all solutions, therefore the pressure drop seen at the end of the firing is not captured with prediction codes.

The real physics in a SRM is complicated with two-phase flow, turbulence and combustion instabilities. Using Euler equations idealizes the problem, thus the results obtained from such analysis would only be estimates and errors made are justified.

4.4 TEST CASE 2 – BOOSTER SRM

In this test case, a conceptual design of a booster SRM is investigated. Since the design is conceptual, the predictions are compared to previously validated codes of performance prediction.

The motor case has a diameter of 400 mm and a length of 3150 mm. The nozzle is a contoured nozzle, which is designed by NCDT code [100]. In order to obtain higher pressure to sustain ignition, 5 slots are placed at the end of the grain; therefore the geometry is not axis-symmetric. The grain and motor geometry are given Figure 4.25.

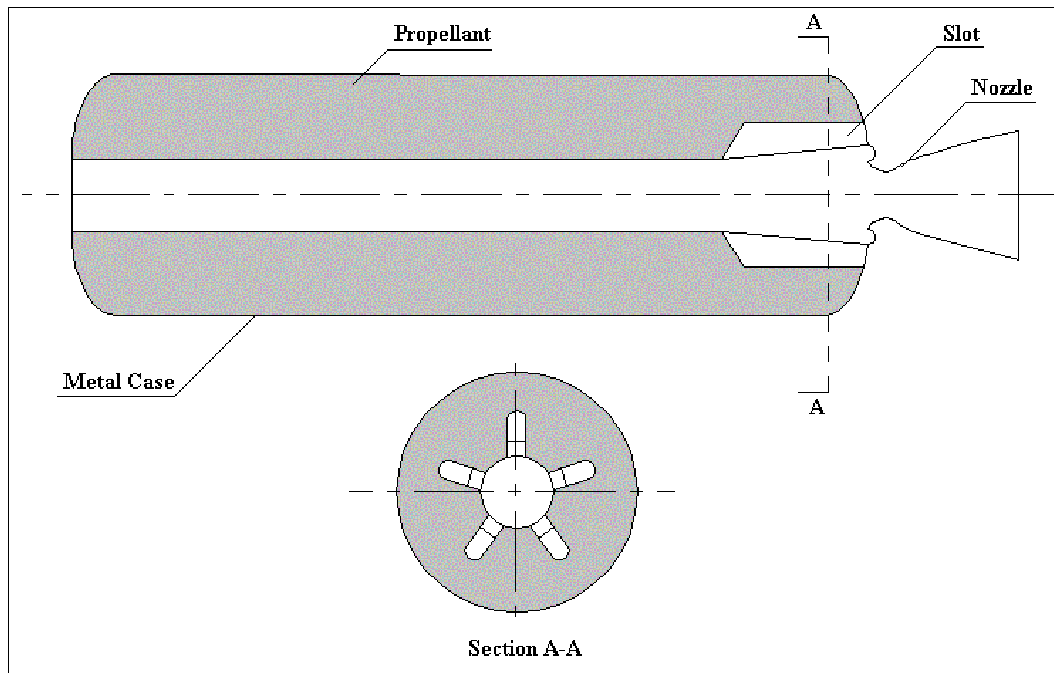


Figure 4.25 Geometry of Space Launcher Rocket Motor

The properties of the propellant are given in Table 4.3 and flow mesh is presented in Figure 4.26. The mach contour results for each captured time interval is given with pressure along the axis of the motor.

Table 4.3 Input Parameters for Booster SRM

Geometric Properties	Unit	
Radius of Propellant	mm	397
Length of Propellant	mm	2600
Length of Slotted Side	mm	400
Length of Cylindrical Side	mm	2200
Number of Slots		5
Width of Stars	mm	50
Propellant Properties		
Flame Temperature	K	2886.0
Mass flux (at 3.0 MPa)	kg/m ² .s	14.0
Mass flux (at 6.0 MPa)	kg/m ² .s	17.2
Mass flux (at 9.0 MPa)	kg/m ² .s	19.3
Ratio of Specific Heat		1.12
Universal Gas Constant	J/kg.K	325.4

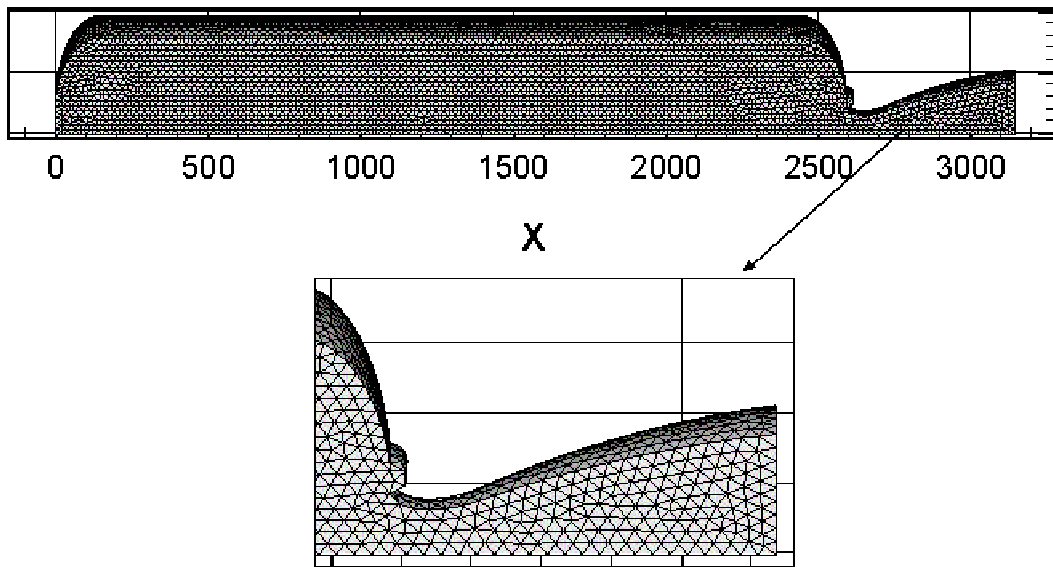


Figure 4.26 Booster SRM Flow Mesh

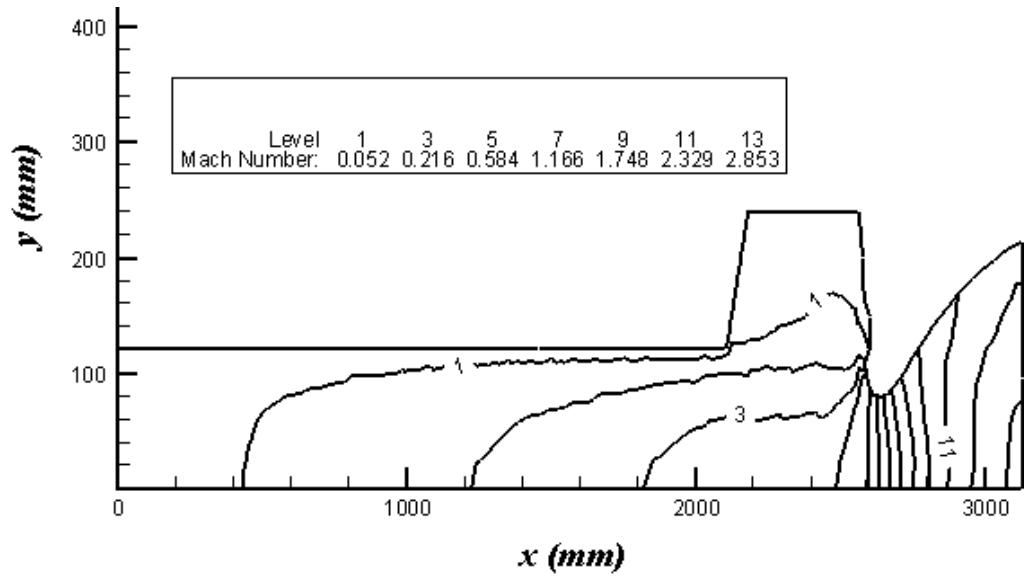


Figure 4.27 Mach Contours for Booster at 0.5 s.

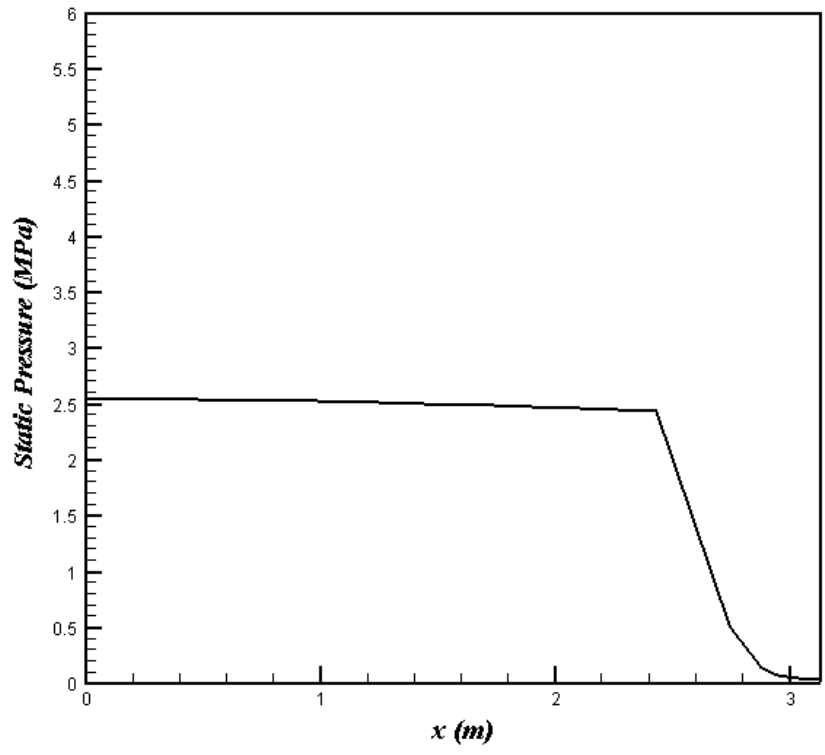


Figure 4.28 Booster Pressure along the Axis at 0.5 s.

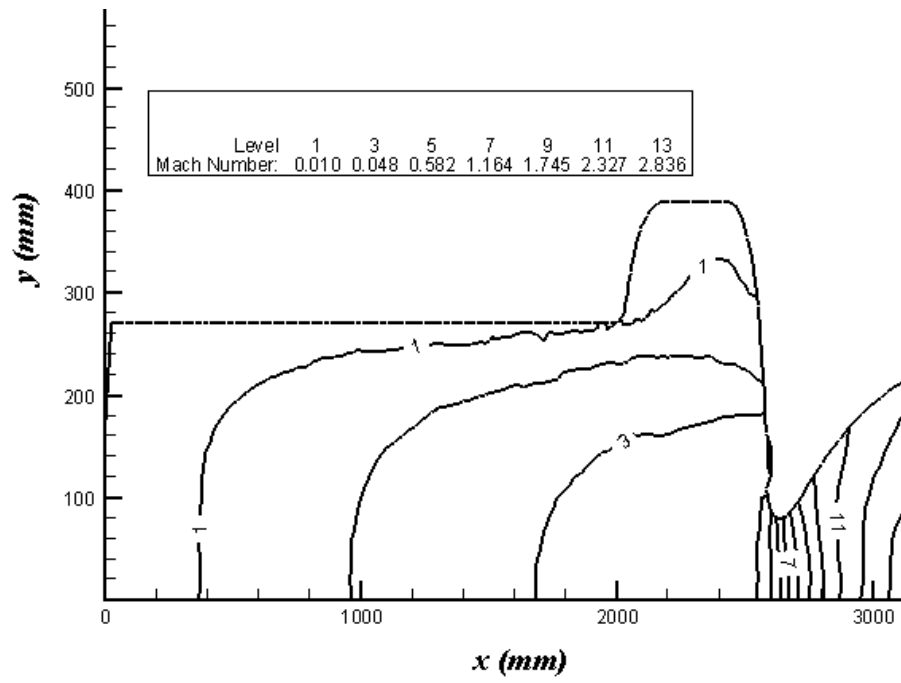


Figure 4.29 Mach Contours for Booster at 18.18 s.

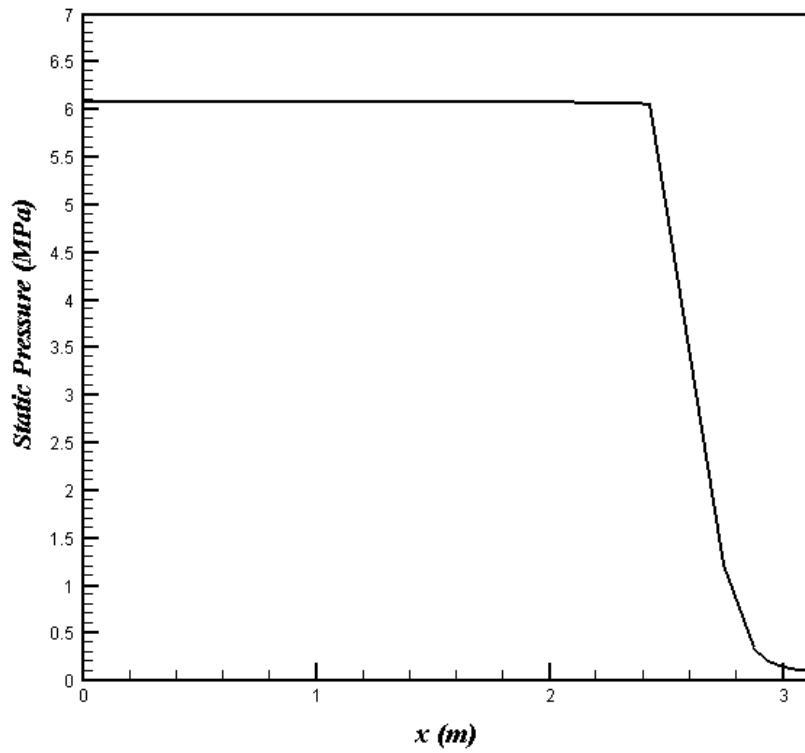


Figure 4.30 Booster Pressure along the Axis at 18.18 s.

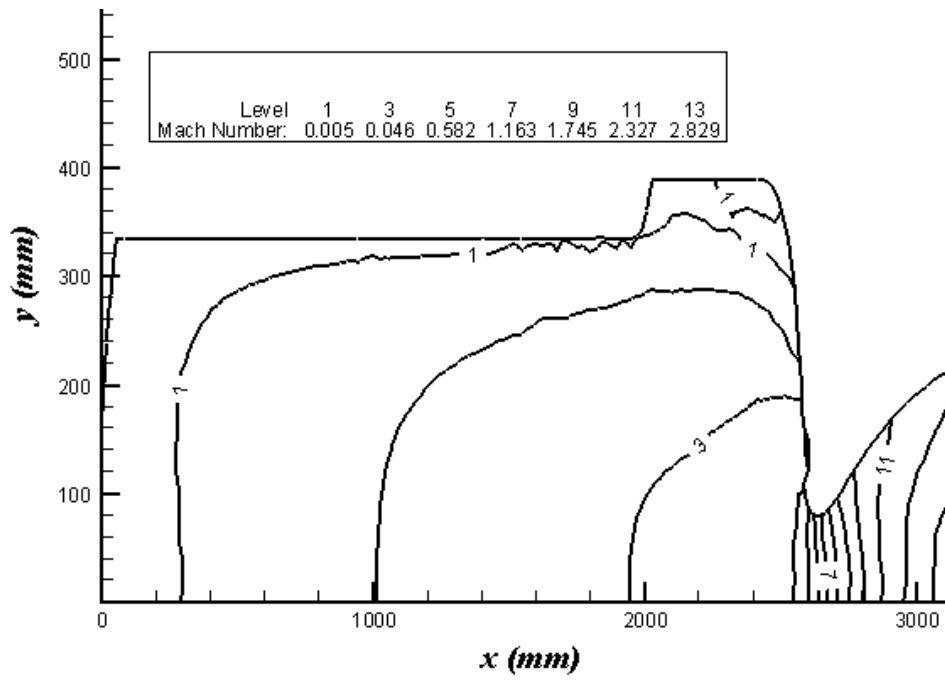


Figure 4.31 Mach Contours for Booster at 24.96 s.

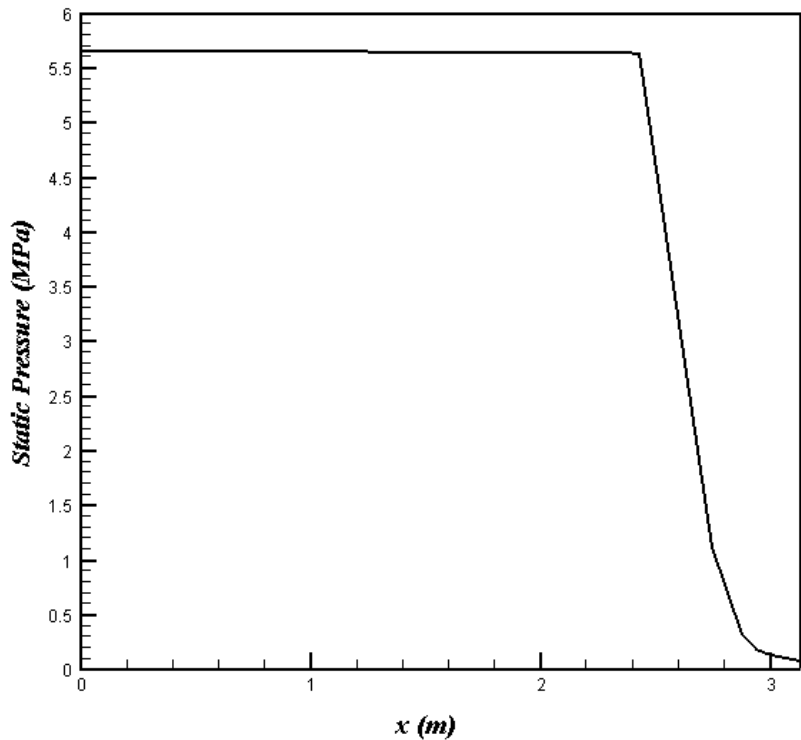


Figure 4.32 Booster Pressure along the Axis at 24.96 s.

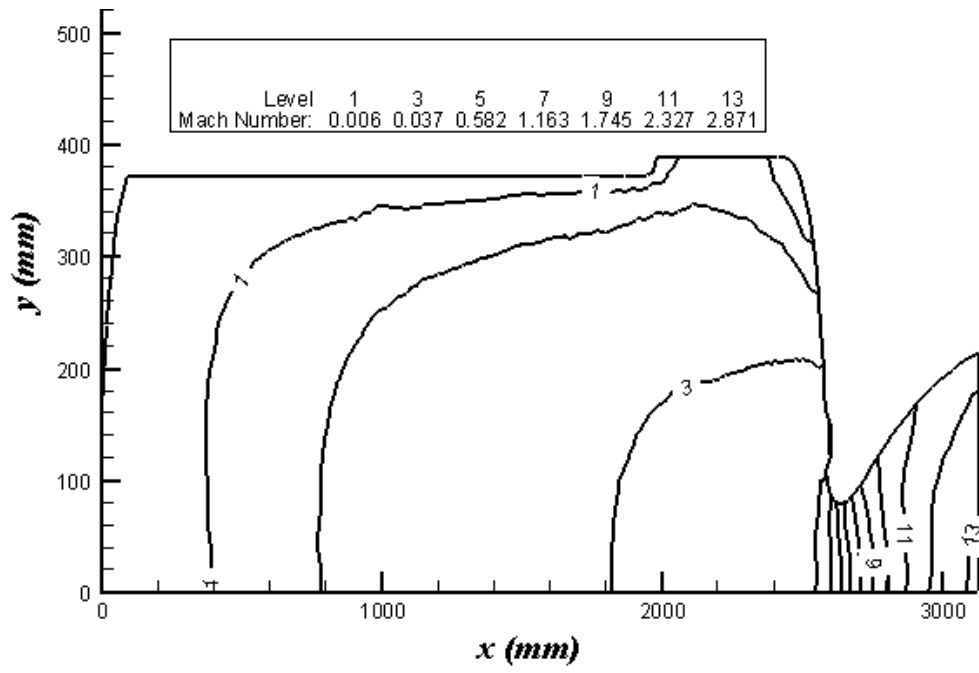


Figure 4.33 Mach Contours for Booster at 28.86 s.

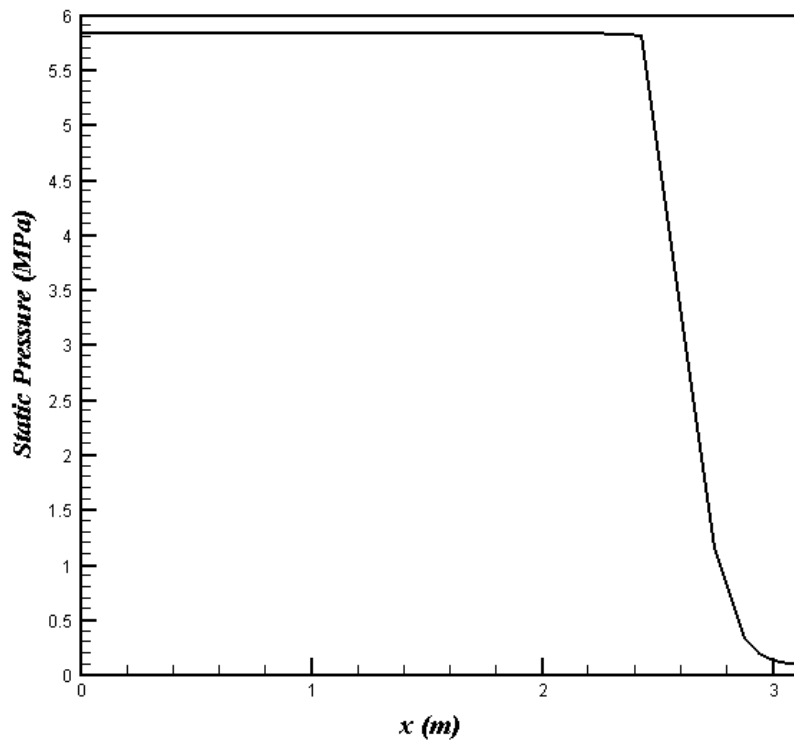


Figure 4.34 Booster Pressure along the Axis at 28.86 s.

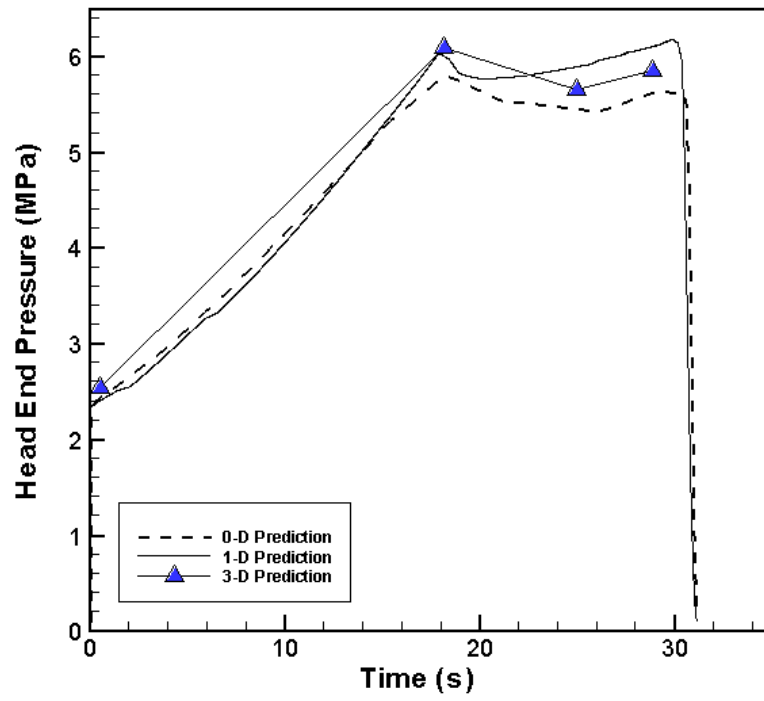


Figure 4.35 Booster Pressure – Time History Prediction

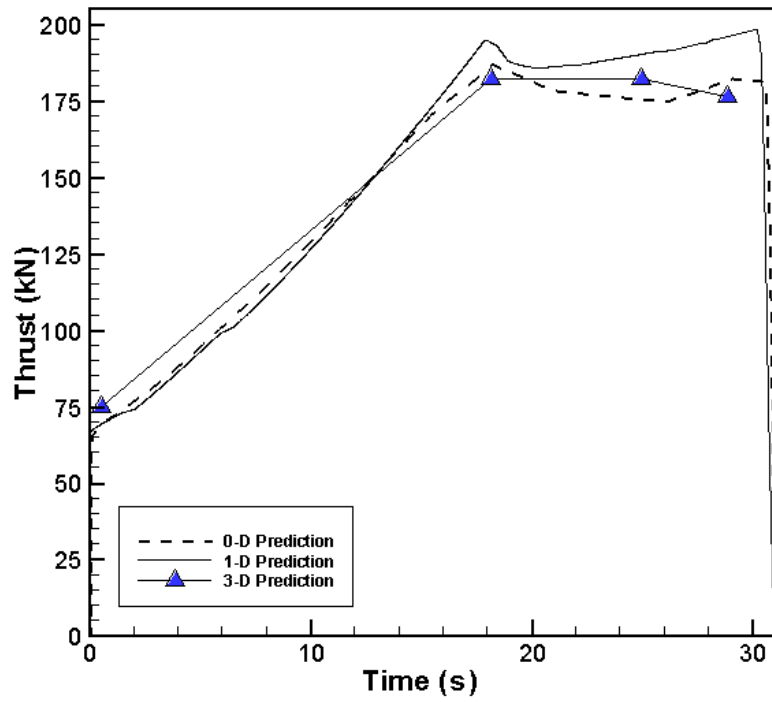


Figure 4.36 Booster Thrust – Time History Prediction

The booster SRM test case has typical three-dimensional grain geometry where analytical or two-dimensional numerical solvers fail to capture the combustion surface motion accurately. A contoured nozzle is used in order to increase thrust efficiency by reducing the radial component of momentum. The optimization of the nozzle geometry is performed based on the criteria that the convergence location of the expansion waves lays outside the nozzle. The expansion waves intercept the pressure outlet boundary and a lower CFL number is required for the solution to achieve stable convergence in the nozzle. Once the expansion waves are formed behind the choked region of the throat, the CFL number is increased.

In the early stages of the combustion, the cylindrical part of the grain is dominant and produces a progressive performance characteristic whereas the aft end takes over the regime during the late stages of the combustion. The comparison of results with zero-dimensional and one-dimensional solvers is in good agreement during the progressive region of the combustion because axi-symmetric assumptions are accurate in this region. However as the aft end of the grain becomes dominant, a difference in the results become visible.

4.5 TEST CASE 3 – DEFECTS IN GRAIN

This coupled solver can be used for Failure Modes and Effects Analysis for SRM manufacturing processes. Several failure modes related to grain structural defects such as voids or cracks in the grain can be modeled and their effects on the performance can be investigated.

In this test case one of the manufacturing problems where air voids are trapped in the grain during the filling operation is investigated. Assuming that the voids have spherical shape, two cases are investigated: 1 void with 10 mm diameter and 3 voids with diameters 10 mm and 6 mm are placed in the grain. Considering the grain model is 1/8 of the real model, the overall effect would be like having 8 voids and 24 voids respectively. The geometry and locations of voids are shown in

Figure 4.37 and Figure 4.38. The grid which is used for the solutions is given in Figure 4.39.

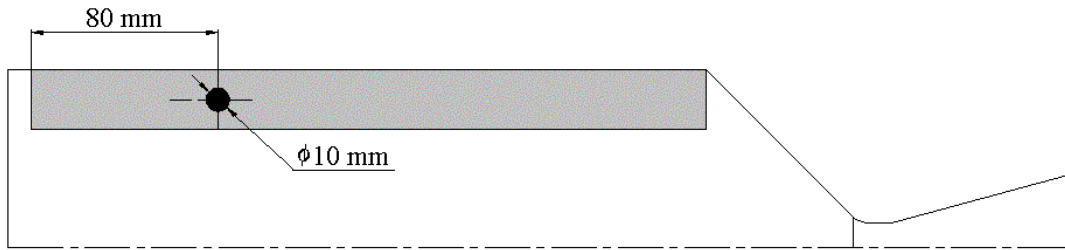


Figure 4.37 Location of Air Void for Test Case 3-A

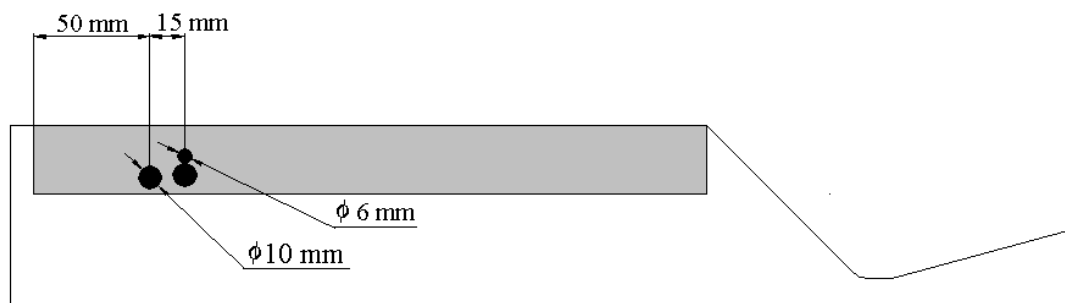


Figure 4.38 Location of Air Void for Test Case 3-B

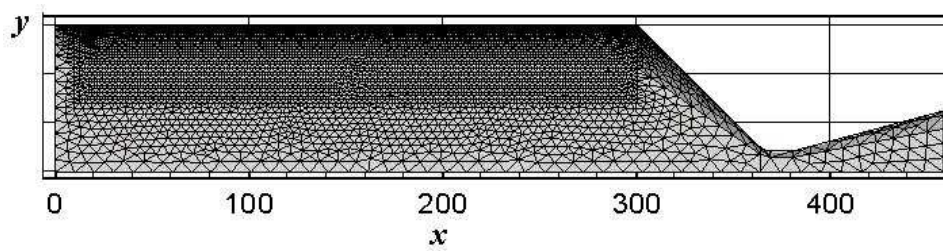


Figure 4.39 Grid for Test Case 3

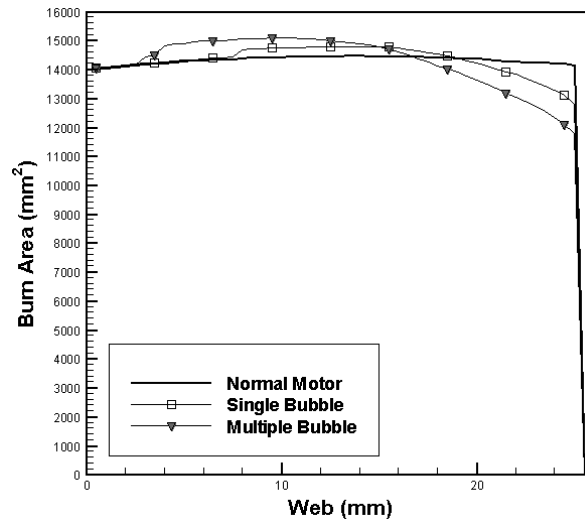


Figure 4.40 Burn Area Comparison for Voids in Grain

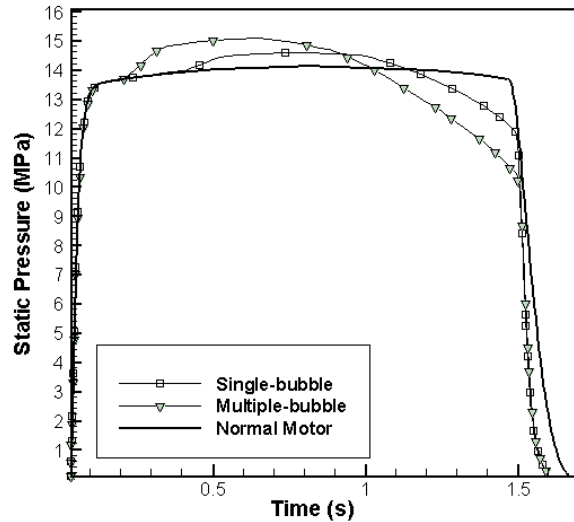


Figure 4.41 Comparison of Pressure for Test Case 3

The burn area results are given in Figure 4.40 and performance prediction results are given in Figure 4.41 and Figure 4.42. It is evident that having voids in the grain increases the burn surface area suddenly, thus changing the pressure and thrust history of the motor significantly. A detailed investigation can be made for quality control procedures of a SRM after the molding process by using X-ray imaging of the grain. No comparison is given in the context of this study

considering the voids in the grain. This test case is solved only to show the capabilities of the proposed model. The arrival time contours are given in Figure 4.43 and Figure 4.44. The three-dimensional results are given in Appendix B.

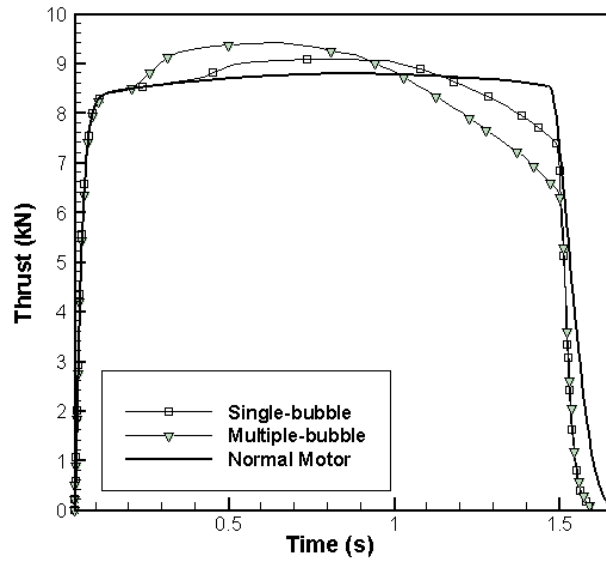


Figure 4.42 Comparison of Thrust for Test Case 3

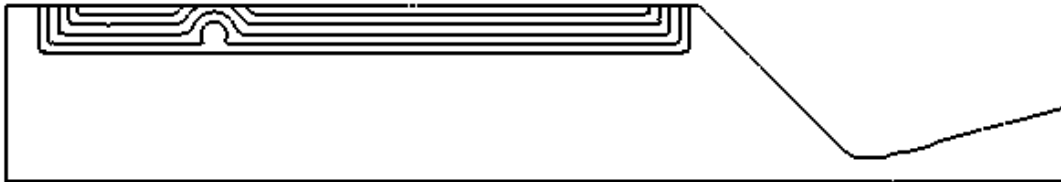


Figure 4.43 Burn Back Results for Test Case 3-A

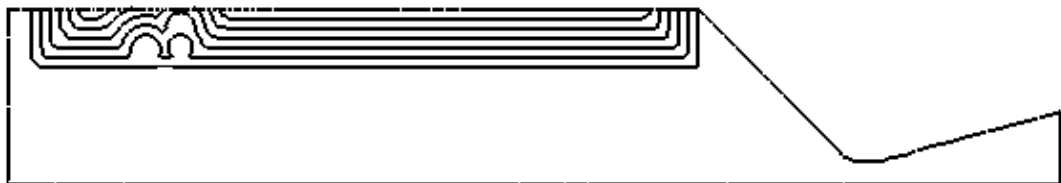


Figure 4.44 Burn Back Results for Test Case 3-B

4.6 TEST CASE 4 – HUMP EFFECT

At contact locations with walls of the mandrel and of the metal case, binder rich zones are created. This is due to the fact that during the casting processes the binders in the propellant stick to metal surfaces [3]. The binder rich zone burn slower, which explains the fact why the burn rate is dependent on the web, burned. Also the head end of the motor is initially closed by a metal piece during the filling process, therefore the burn rate decreases at these locations. The burn rate effect is defined through the relation

$$\frac{r_b}{r_o} = f(R, \theta, z) \quad (4.4)$$

where r_b is the actual burn rate, r_o is the known burning rate without hump effect, $f(R, \theta, z)$ is the function controlling the hump effect. The hump effect function can be found empirically as a rule, it varies between 0.95 and 1.05 and follows a sinusoidal shape.

In the R -direction the hump function is defined as

$$f(R) = 1 + 0.05 \cdot \sin\left(\frac{2\pi(R - R_i)}{(R_o - R_i)} - \frac{\pi}{2}\right) \quad (4.5)$$

where R_i is the inner radius of the grain and R_o is the outer radius of the motor.

In the z -direction the hump function is defined as

$$f(z) = 0.95 + 0.025 \cdot \left(\sin\left(\frac{\pi \cdot z}{R_o - R_i} - \frac{\pi}{2}\right) + 1 \right) \quad z \leq R_o - R_i \quad (4.6)$$

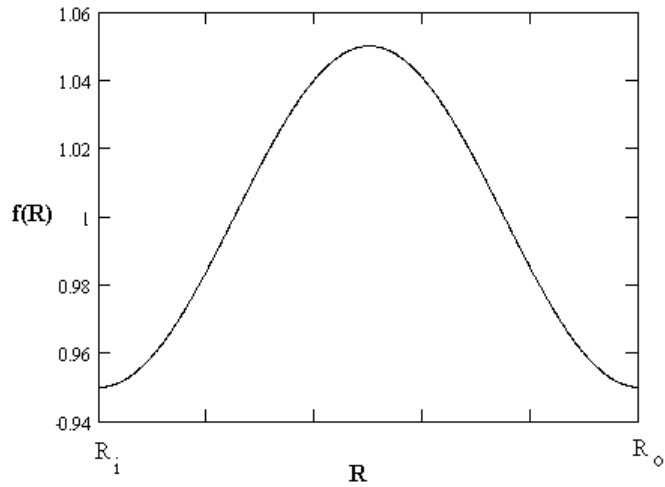


Figure 4.45 Hump Parameter as a Function of the Radial Direction

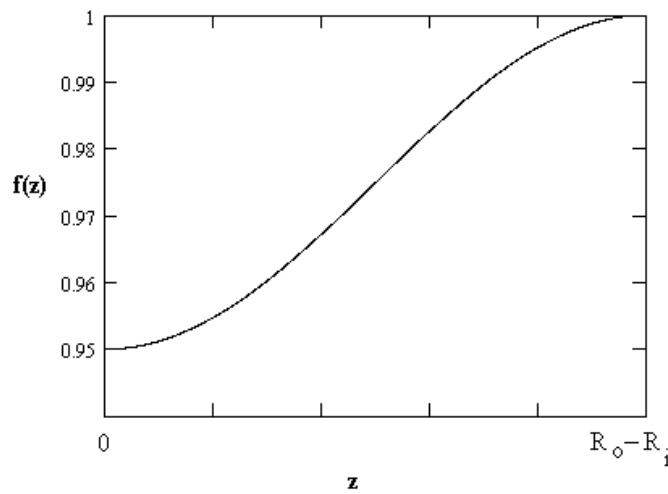


Figure 4.46 Hump Parameter as a Function of the Axial Direction

For 6C4 motor, the axial burn rate change is effective on 10 % of the whole motor length, and the maximum difference of 5 % occurs at $z=0$. Therefore total effect is less than 0.5 % in the overall performance prediction.

The change of burn rate along the radial direction is dominant in the flow solutions. The whole length of the motor is affected by this change of rate therefore the maximum change of 5 % can be seen in performance prediction.

The following results are obtained for the test case with hump effect.

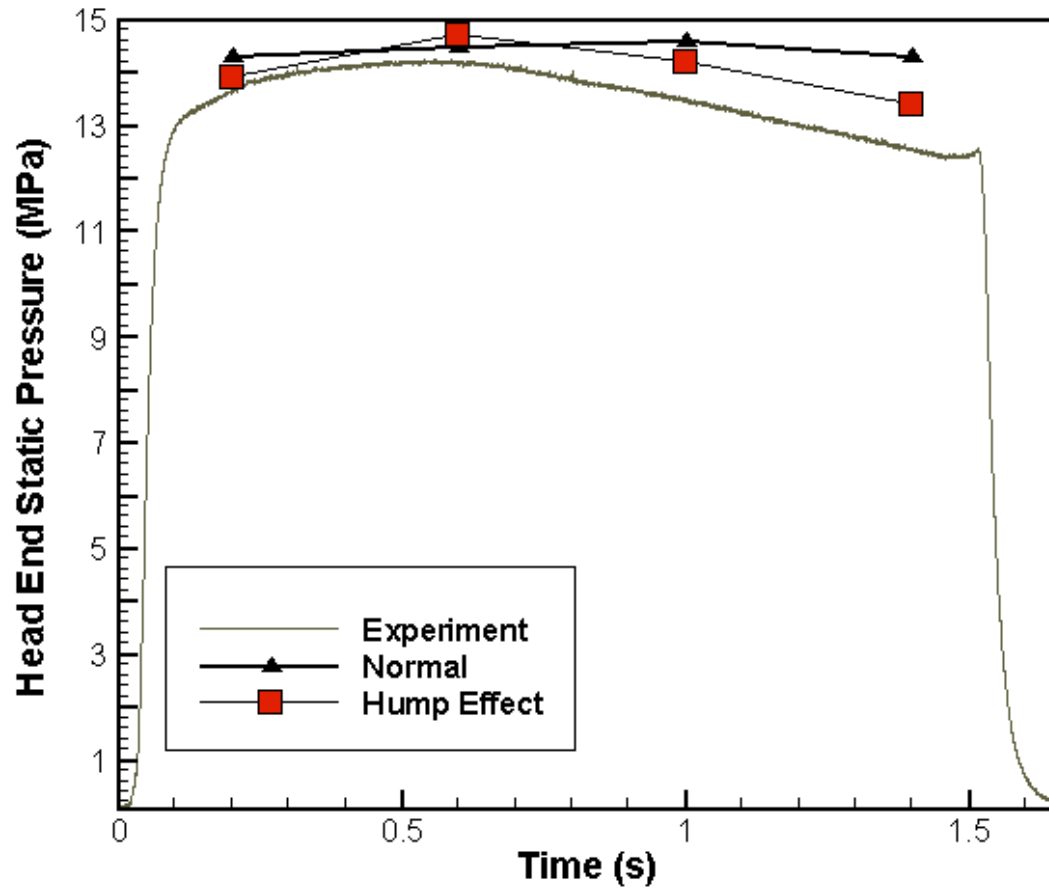


Figure 4.47 6C4 Pressure Time History with Hump Effect

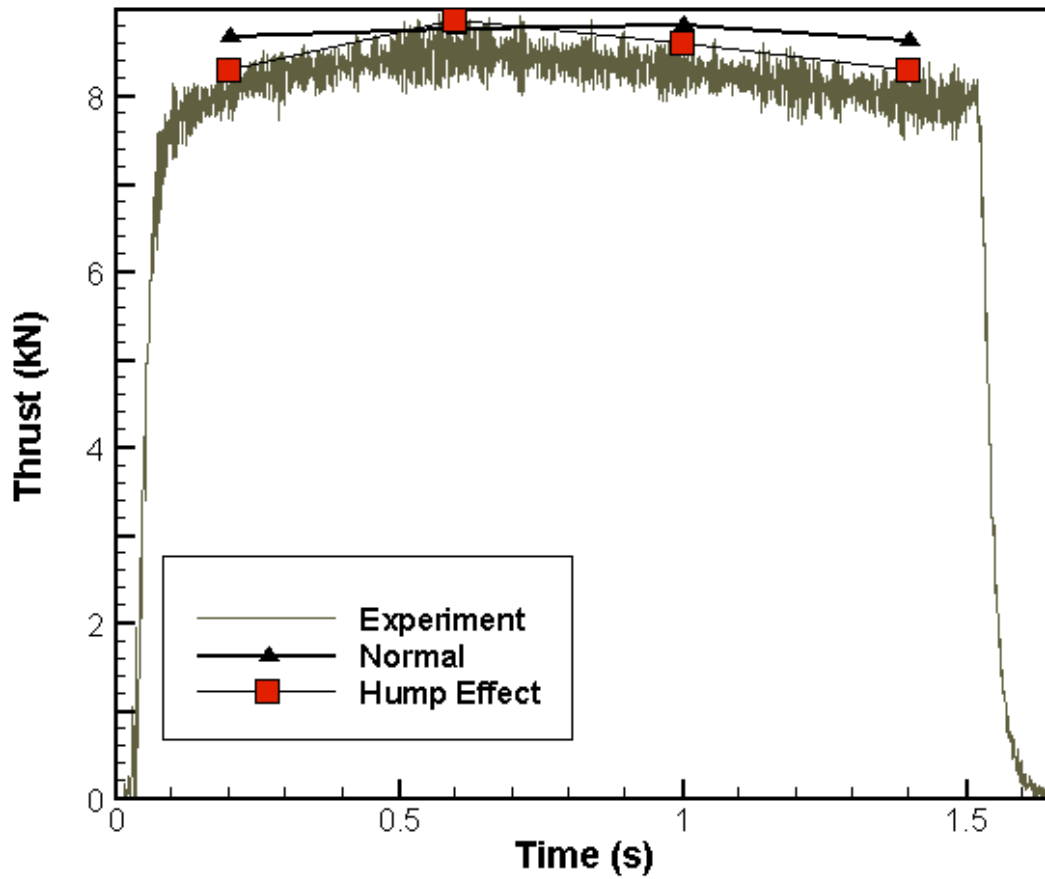


Figure 4.48 6C4 Thrust Time History with Hump Effect

Combining hump effect with the burn rate calculations, enhances the results significantly. The error which is made between the experimental and the prediction is mostly due to the fact that the nozzle erosion and ablation is not modeled in this test case. Another factor may be due to the expansion of the motor case and propellant grain due to high pressure built up in the motor. However the numerical analysis neglects any effect related to structural expansion.

CHAPTER 5

CONCLUSION AND DISCUSSION

The methodology described in this work is applicable to all moving boundary problems. However the intention is to solve three-dimensional grain burnback effects in a solid rocket motor. The interface capturing algorithm of Fast Marching Method is coupled to zero-, one-, and three-dimensional flow solvers in a segregated fashion. The results obtained from the present calculations and previous analyses are in good agreement.

The interface calculations are performed by an interface-capturing algorithm therefore the solutions are mesh dependent. Better accuracy can be achieved by using higher number of elements which will increase the run time. Otherwise the interface tends to smear out.

The three-dimensional flow solver (HIBS3D) is an explicit type flow solver therefore number of iterations for convergence is high. The elements generated on the boundaries are smaller than the ones in the rest of the solution domain. Therefore the flow solver loses time while trying to analyze these elements. The algorithm removes very small elements on the boundary however the real position of the boundary may be altered due to this process. The flow solver is first order accurate in space therefore in order to obtain the same results as the second order

accurate solvers, more elements are required. The problem can be solved through a second order implicit solver or a parallel solver.

Since the grid is stationary, previous results are used as initial conditions for the next calculation. This reduces the time for convergence drastically. However the results from the previous run should be ready for such a process, whereas at some test cases all the captured domains for each time interval are solved on different computers simultaneously without the initial values from previous runs. This way as the run time increases for each computer, overall run time is reduced.

The interface capturing methodology does not require keeping track of connectivity of elements. Therefore errors due to break up of surface are not present in our method. Since the grid is stationary all properties of propellant such as crack, voids and other manufacturing deficiencies may be implemented prior to run.

The flow solver of Roe with entropy fix method has been found to work effectively in transonic solutions. The solver is adapted to the cut-cell method used in tetrahedron mesh effectively by converting the tetrahedron solver into a hybrid solver. The hybrid solver is robust and the results are accurate where any type of mesh element can be used provided that the notation for the mesh is followed.

5.1 SUGGESTIONS FOR FUTURE WORK

The suggestions for future work can be proposed as follows:

- The flow solver should be upgraded to second order accurate in space.
- The flow solver should be converted to an implicit solver.
- The flow solver should be parallelized.
- Initial value formulation for interface capturing should be implemented and the interface solver should be applied to other moving boundary problems.

- The flow solver should be coupled to combustion, two-phase flow, turbulence and acoustic models as well.
- The interface-capturing program should implement mesh adaptivity in order to reduce smearing of the interface and reducing the amount of elements used.
- The nozzle erosion and ablation should also be modeled.
- The structural analysis should be coupled to the flow analysis, thus changes in the grain and motor case can be modeled.

5.2 CONCLUSION

The achievements and contributions of this study can be summarized as:

- The method of Fast Marching is applied to rocket motor grain burning surface modeling for the first time.
- The method of Fast Marching is applied on tetrahedron elements.
- The cut-cell method is applied on tetrahedron elements.
- A coupled model, which uses three-dimensional finite volume solver with a Fast Marching interface capturing method which uses cut-cell method, has been developed.

The coupled model can be used to solve rocket motor performance as well as any other problem with moving boundaries, since all the procedures are set in a generic manner.

REFERENCES

- [1] DE LUCA L.T., "*Burning Rate Fundamentals*", RTO Technical Report 43, pp.11-31, 2002
- [2] "Solid Propellant Grain Design and Internal Ballistics", NASA, SP-8076, 1972.
- [3] DAVENAS A., "*Solid Rocket Propulsion Technology*" Pergamon Press, 1993.
- [4] SMOLIANSKI A., "*Finite Element/Level-Set/Operator-Splitting (FELSOS) Approach for Computing Two-Fluid Unsteady Flows with Free Moving Interfaces*", Institute of Mathematics, University of Zurich, 1998..
- [5] SHYY W., UDAYKUMAR H.S., RAO M., SMITH R., "*Computational Fluid Dynamics with Moving Boundaries*", Taylor & Francis, 1996.
- [6] HOU T.Y., "*Numerical Solutions to free boundary problems.*" Acta Numerica, pp.335-415, 1995.
- [7] SETHIAN J.A. "*Level Set Methods and Fast Marching Methods*" Cambridge University Press, 2nd ed.,1999
- [8] OKAMOTO T., KAWAHARA M., "*Two-dimensional Sloshing Analysis by Lagrangian Finite Element Method*", Int. J. Num. Meth. Fluids, vol: 11, pp. 453-477, 1990.
- [9] FYFE D.E., ORAN E.S., FRITTS M.J., "*Surface Tension and Viscosity with Lagrangian Hydrodynamics on a Triangular Mesh*", J. Comput. Phys., vol: 76, pp. 349-384, 1988.
- [10] CHRISTODOULOU K.N., SCRIVEN L.E., "*Discretization of Free Surface Flows and Other Moving Boundary Problems*", J. of Comput. Phys., vol: 99, pp. 39-55, 1992.
- [11] CAMACHO G.T., ORTIZ M., "*Adaptive Lagrangian Modeling of Ballistic*

- Penetration of Metallic Targets*”, *Compt. Meth. App. Mech. Eng.*, vol:142, pp.269-301, 1997.
- [12] HIRT C.W., AMSDEN A.A., COOK J.L., “*An Arbitrary Lagrangian-Eulerian Computing Method for All Speeds*”, *J. Comput. Phys.* Vol: 14, pp.227-253, 1974.
- [13] CRISTINI V., BLAWSDEZIEWICZ J., LOEWENBERG M., “*An Adaptive Mesh Algorithm for Evolving Surfaces: Simulations of Drop Break-up and Coalescence.*” *J. of Comp. Physics*, 168, pp. 445-463, 2001
- [14] MONAGHAN J.J., “*Simulating Free Surface Flows with SPH*”, *J. Comput. Phys.*, vol: 110, pp.399-406, 1994
- [15] MORRIS J.P., “*Simulating Surface Tension with Smoothed Particle Hydrodynamics*”, *Int. J. Num. Meth. Fluids*, vol:33, pp.333-353, 2000.
- [16] HYMAN J.M., “*Numerical Methods for Tracking Interfaces*”, *Physica D*, vol: 12, pp. 396-407, 1984.
- [17] PESKIN C.S., “*The Dynamics of Heart Valves: Experimental, Theoretical and Computational Methods*”, *Ann. Rev. Fluid Mech.* Vol:14, pp.235, 1982.
- [18] GOLDSTEIN D., HANDLER R., SIROVICH L., “*Modeling a No-slip Flow Boundary with an External Force Field*”, *J. Comput. Phys.*, vol: 105, pp. 354, 1993.
- [19] UNVERDI S.O., TRYGGVASON G., “*A Front Tracking Method for Viscous, Incompressible, Multi-fluid Flows*”, *J. Comput. Phys.*, vol:100, pp.25-37, 1992.
- [20] JURIC D., TRYGGVASON G., “*A Front Tracking Method for Dendritic Solidification*”, *J. Comp. Phys.*, vol: 123, pp. 127-148, 1996.
- [21] KIM J., KIM D., CHOI H., “*An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries*”, *J. Comput. Fluids*, vol: 171, pp. 132-150, 2001.
- [22] UDAYKUMAR H.S., SHYY W. RAO M. M., “*ELAFINT-A Mixed Eulerian-Lagrangian Method for Fluid Flows with Complex and Moving Boundaries*”, *Int. J. Numer. Methods Fluids*, vol: 22, 691, 1996.

- [23] UDAYKUMAR H.S., MITTAL R., SHYY W., “*Computation of Solid-Liquid Phase Fronts in the Sharp Limit on Fixed Grids*”, J. of Comp. Phys., Vol: 153, pp.535-574, 1999.
- [24] TRYGGVASON G., BUNNER B., ESMAEELI A., JURIC D., “*A Front Tracking Method for the Computations of Multiphase Flow*”, J. Comput. Phys., vol: 169-2, pp. 708-759, 2001
- [25] GLIMM J., GROVE J.W., LI X.L., SHYUE K.M., ZHANG Q., ZENG Y., “*Three-dimensional Front Tracking*”, SIAM J. Sci. Comput., 1998, vol: 19, pp. 703-727
- [26] GLIMM J., GROVE J.W., LI X.L., TAN D.C., “*Robust Computational Algorithms for Dynamic Interface Tracking in Three Dimensions*”, SIAM J. Sci. Comput., Vol: 21, pp.2240-2256, 2000.
- [27] TORRES D.J., BRACKBILL J.U., “*The point-set Method : Front Tracking without Connectivity*”, J. Comput. Phys., vol: 165, pp. 620-644, 2000.
- [28] SHIN S., JURIC D., “*Modeling Three-dimensional Multiphase Flow Using a Level Contour Reconstruction Method for Front Tracking without Connectivity*”. J. of Comp. Physics, 180, 427-470, 2002
- [29] POPINET S., ZALESKI S., “*A Front Tracking Algorithm for Accurate Representation of Surface Tension*”, Int. J. Num. Meth. Fluids, vol: 30, pp. 775-793, 1999.
- [30] CRUCHAGA M., CELENTANO D., TEZDUYAR T., “*Computation of Mould Filling Processes with a Moving Lagrangian Interface Technique*”, Comm. Num. Meth. Eng., vol: 18, pp. 483-493, 2002
- [31] JOHNSON R.A., BELK D.M., “*Multi-grid Approach to Overset Grid Communication*”, AIAA J., vol: 33, pp.2305, 1995.
- [32] YANG G., CAUSON D.M., INGRAM D.M., “*Calculation of Compressible Flows about Complex Moving Geometries Using a Three-dimensional Cartesian Cut Cell Method*”, Int. J. Num. Meth. Fluids, vol: 33, pp. 1121-1151, 2000.
- [33] NAKAYAMA T., MORI M., “*An Eulerian Finite Element Method for Time-dependent Free Surface Problems in Hydrodynamics*” Int. J. Num.

- Meth. Fluids, vol: 22, pp. 175-194, 1996.
- [34] YORK A.R., SULSKY D., SCHREYER H.L., “*The Material Point Method for Simulation of Thin Membranes*”, Int. J. Numer. Meth. Eng., vol: 44, pp. 1429-1456, 1999
- [35] BRACKBILL J.U., RUPPEL H.M., “*FLIP: a Method for Adoptively Zoned, Particle-In-Cell Calculations in Two-dimensions*”, J. Comput. Phys., vol: 65, pp. 314-343, 1986.
- [36] BRACKBILL J.U., “*A Continuum Method for Modeling Surface Tension*”, J. Comput. Phys. Vol: 100, pp. 335-354, 1992.
- [37] DAI M., WANG H., “*A Numerical Method for Interface Tracking*” ILASS 15th Annual Conference on Liquid Atomization and Spray Systems, May 2002
- [38] HIRT C.W., NICHOLS B.D., “*Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries*”, J. Comput. Phys., vol: 39, pp.201-225, 1981.
- [39] ASHGRIZ N., POO J.Y., “*FLAIR-Flux Line Segment Model for Advection and Interface Reconstruction*”, J. Comput. Phys., vol:93, pp. 449-48, 1991.
- [40] RUDMAN M., “*Volume Tracking Methods for Interfacial Flow Calculations*”, Int. J. Num. Meth. Fluids, vol: 24, pp. 671-691, 1997.
- [41] PUCKETT E.G., ALMGREN A.S., BELL J.B., MARCUS D.L., RIDER W.J., “*A High Order Projection Method for Tracking Fluid Interfaces in Variable Density Compressible Flows*”, J. Comput. Phys., vol: 130, pp. 269-282, 1997.
- [42] RIDER W.J., KOTHE D.B., “*Reconstructing Volume Tracking*”, J. Comput. Phys. Vol:141, pp. 112-152, 1998.
- [43] ALIABADI S., TEZDUYAR T., “*Stabilized-finite-element / Interface-Capturing Technique for Parallel Computation of Unsteady Flows with Interfaces*”, Comput. Meth. Appl. Mech. Eng., vol: 190, pp. 243-261, 2000.
- [44] SCARDOVELLI R., ZALESKI S., “*Interface Reconstruction with Least-Square Fit and Split Eulerian-Lagrangian Advection*”, Int. J. Num. Meth.

- Fluids, vol: 41, pp. 251-274, 2003.
- [45] MARONNIER V., PICASSO M., RAPPAZ J., “*Numerical Simulation of Three-dimensional Free Surface Flows*”, Int. J. Num. Meth. Fluids, vol:42, pp. 697-716, 2003.
- [46] KELECY F.J., PLETCHER R.H., “*The Development of a Free Surface Capturing Approach for Multidimensional Free Surface Flows in Closed Containers*”, J. Comput. Phys., vol: 138, pp. 939-980, 1997.
- [47] OSHER S., SETHIAN J.A., “*Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations.*” J. of Comp. Phy., 79, pp.12-49, 1988.
- [48] OSHER S., FEDKIW R.P., “*Level Set Methods: An overview and Some Recent Results*”, J. Comput. Phys., vol:169, pp. 463-502, 2001
- [49] SETHIAN J.A., “*Fast Marching Methods and Level Set Methods for Propagating Interfaces*”, VKI Lecture Series, 3, 1998
- [50] BARTH T., SETHIAN J.A., “*Numerical Schemes for the Hamilton-Jacobi and Level Set Equations on Triangulated Domains*” J. of Comp. Phy. 145, 1-40, 1998
- [51] SETHIAN J.A., VLADIMIRSKY A., “*Fast Methods for the Eikonal and Related Hamilton Jacobi Equations on Unstructured Meshes*” PNAS, vol: 97, No: 11, pp. 5699-5703, 2000.
- [52] ADALSTEINSSON D., SETHIAN J.A., “*The Fast Construction of Extension Velocities in Level Set Methods*” J. of Comp. Phy. 148,2-22,1999
- [53] SUSSMAN M., SMEREKA P., OSHER S., “*A Level Set Approach for Computing Solutions to Incompressible Two-phase Flows*”, J. Comput. Phys., vol: 114, pp. 14-149, 1994
- [54] ADALSTEINSSON D., SETHIAN J.D., “*Transport and Diffusion of Material Quantities on Propagating Interfaces Via Level Set Methods*”, J. Comput. Phys., vol: 185, pp. 271-288, 2003.
- [55] SUKUMAR N., CHOPP D.L., MORAN B., “*Extended Finite Element Method and Fast Marching Method for Three-dimensional Fatigue Crack Propagation*”, Eng. Fracture Mech., vol: 70, pp. 29-48, 2003.

- [56] KIMMEL R., SETHIAN J.A., "Computing Geodesic Paths on Manifolds", Proc. Nat. Academy Sci., 1997.
- [57] KOHNO H., TANAHASHI T., "Numerical Analysis of Moving Interfaces Using a Level Set Method Coupled with Adaptive Mesh Refinement", Int. J. Num. Meth. Fluids, vol: 45, pp. 921-944, 2004.
- [58] KOBAYASHI R., "Modeling and Numerical Simulation of Dendritic Crystal Growth", Physica D, vol: 3, pp. 410-423, 1993
- [59] KOBAYASHI R., WARREN J.A., CARTER W.C., "A Continuum Model of Grain Boundaries", Physica D, vol: 140, pp. 141-150, 2000.
- [60] LE BRETON P., RIBERAU D., "Casting Process Impact on Small-Scale Solid Rocket Motor Ballistic Performance" J. of Propulsion and Power, Vol.18, No: 6, pp. 1211-1217, 2002
- [61] HEJL R.J., HEISTER S.D., "Solid Rocket Motor Grain Burnback Analysis Using Adaptive Grids. "J. of Prop. Power, 1995, Vol.11 No: 5, pp1006-1011.
- [62] STONE M.W., "A practical Mathematical Approach to Grain Design "Jet Propulsion, pp. 236-244, April 1958.
- [63] ZARDA P.R., "Computer Aided Propulsion Burn Analysis" Martin Marietta Electronics and Missile Group, AIAA-88-3342
- [64] SAINTOUT E., RIBERAU D., PERRIN P., "ELEA: A Tool for 3D Surface Regression Analysis in Propellant Grains", AIAA89-2782
- [65] DUNN S.S., COATS D.E., "3-D Grain Design and Ballistic Analysis Using the SPP97 Code". AIAA 97-3340
- [66] ALAVILLI P., TAFTI D., NAJJAR F., "The Development of an Advanced Solid-Rocket Flow Simulation: Program ROCFLO." AIAA-2000-0824, 2000
- [67] SHYY W., FRANCOIS M., UDAYKUMAR H.S., "Computational Moving Boundary Problems in Engineering and Biomechanics". 7th National Computational Fluid Dynamics Conference, 2000.
- [68] ALAVILLI P., BUCKMASTER J., JACKSON T.L., SHORT M., "Ignition-Transient Modeling for Solid Propellant Rocket Motors." AIAA-2000-3567,

- [69] PIRZADEH S.Z. "*Unstructured Viscous Grid Generation by Advancing-Front Method*", NASA Contractor Report 191449, April 1993.
- [70] FRENCH J.C., COATS D.E., "*Automated 3-D Solid Rocket Combustion Stability Analysis.*" AIAA-99-2797, 1999
- [71] HECKEL B., UVA A.E., HAMANN B., JOY K., "*Surface Reconstruction Using Adaptive Clustering Methods*". Dept. of Comp. Sci., University of California
- [72] JEONG J. H., YANG D. Y., "*Three-dimensional Finite Element Analysis of Transient Fluid Flow with Free-Surface Using Marker Surface Method and Adaptive Grid Refinement*", Int. J. Numer. Meth. Fluids, Vol: 29, pp. 657-684, 1999.
- [73] KARIMIAN S.M.H., AMOLI A., "*Two-dimensional Simulation of SRM Internal Ballistics on a Moving Grid*" AIAA-99-2801, 1999
- [74] KARLSEN K.H., RISEBRO N.H., "*Unconditionally Stable Methods for Hamilton-Jacobi Equations.*" J. of Comp. Physics, 180, 710-735, 2002
- [75] MAJDALANI J., "*An Improved Time-dependent Flow field Solution for Solid Rocket Motors*" AIAA-97-2717, 1997
- [76] PIRZADEH S.Z. "*An Adaptive Unstructured Grid Method by Grid Subdivision, Local Re-meshing, and Grid Movement*" AIAA-99-3255, 1999
- [77] SHEWCHUK J. R., "*Delaunay Refinement Mesh Generation*", PhD. Thesis, Submitted to the School of Computer Science, Carnegie Mellon University, May 1997.
- [78] TANG H.Z., TANG T., ZHANG P., "*An Adaptive Mesh Redistribution Method for Non-linear Hamilton-Jacobi Equations in Two- and Three-dimensions.*"
- [79] THOMPSON J.F., WARSI Z.U., WAYNE MASTIN C. "*Numerical Grid Generation*" Elsevier Science Publishing Co. 1985
- [80] TINAZTEPE T., ÇAKIROĞLU M., RIBEREAU D., UHRIG G., "*Technical Report of AGARD-PEP T71 Support Project*" Roketsan A.Ş., 1995

- [81] ZHOU C., MAJDALANI J., *"Improved Mean-flow Solution for Slab Rocket Motors with Regressing Walls."* AIAA-2000-3191, 2000
- [82] KAKAŞCI U. *"Three-dimensional Unstructured Grid Generation by Advancing Layer Method."* M.S. Thesis, Submitted to the Graduate School of Natural and Applied Sciences of the Middle East Technical University, Dec. 2001
- [83] STEGER J.L., WARMING R.F., *"Flux Vector Splitting of the Inviscid Gas Dynamic Equations with Application to Finite Difference Methods"*, J. of Comp. Physics, 40, 263-293, 1981.
- [84] VAN LEER B., *"Flux-vector Splitting for The Euler Equations"* Lecture Notes in Physics, Springer-Verlag, 1982, vol: 70.
- [85] ANDERSON W.K., THOMAS J.L., *"Comparison of Finite Volume Flux Vector Splittings for The Euler Equations"* AIAA Journal, vol.24, No.9, 1986, 1453-1460.
- [86] LUO H., BAUM J.D., *"An Improved Finite Volume Scheme for Incompressible Flows on Unstructured Grids"* AIAA-95-0348
- [87] JAMESON A., SCHMIDT W., TURKEL E., *"Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes"*, AIAA-81-1259.
- [88] MANNA M., *"A Three-dimensional High Resolution Upwind Finite Volume Euler Solver"*, Von Karman Institute Technical Notes 180, 1992.
- [89] TORO E.F., *"Riemann Solvers and Numerical Methods for Fluid Dynamics"*, Springer-Verlag, 1997.
- [90] LANEY C.B., *"Computational Gas Dynamics"*, Cambridge University Press, 1998.
- [91] TUNCER I.H., *AE 546 CFD on Unstructured Grid - Lecture Notes*, METU, Ankara, Turkey, 2003
- [92] VUILLOT F., YUMUSAK M., TASKINOGLU E.S., TINAZTEPE T., *"Technical Report of RTO-AVT T-108 Support Project"*, Roketsan A.S., 1998.
- [93] HIRSCH C., *"Numerical Computation of Internal and External Flows"*,

- vol.2, John-Wiley & Sons, 1988.
- [94] SUTTON G.P., *"Rocket Propulsion Elements"*, Wiley, NY, USA, 6th Edition, 1992.
- [95] NETZER D.W., *"Propulsion Analysis for Tactical Solid Propellant Rocket Motors"*, Lecture Notes, 1990.
- [96] I-deas Help Files, *"File Formats"*, Structural Dynamics Research Corporation, 1999
- [97] VENKATAKRISHNAN V., *"Perspective on Unstructured Grid Flow Solvers"*, AIAA J., vol:34, pp.533-547, 1996
- [98] SHYY W., FRANCOIS M., UDAYKUMAR H.S., *"Cartesian and Curvilinear Grid Methods for Multi-domain, Moving Boundary Problems"*, 13. International Conference on Domain Decomposition Methods, pp. 109-125, 2001
- [99] KAWARADA H., SUITO H., *"Numerical Method for a Free Surface Flow on the Basis of the Fictitious Domain Method"*, East-West J. Num. Math., vol: 5-1, pp. 57-66, 1997.
- [100] SEÇKİN B., *"Rocket Nozzle Design and Optimization"*, M.S. Thesis, Submitted to the Graduate School of Natural and Applied Sciences of the Middle East Technical University, Jan 2003

APPENDIX A

USER'S MANUAL

In this section the input files and the programs for the coupled solutions is presented briefly. The pre-processing procedures, processing procedures and post processing procedures are described separately.

A.1 PRE-PROCESSING STEP

The aim of the pre-processing is to prepare mesh files for the regression codes to process. These files include node positions, connectivity of elements and boundary conditions for each node. The solid model is generated with the I-DEAS[®] Master Modeler and unstructured tetrahedron mesh is generated with I-DEAS[®] Simulation module. The boundary conditions are set and a text file in universal file format is exported from I-DEAS, which is imported to the regression code.

A.1.1 I-Deas Universal File Format

Universal files are ASCII data files, that can be used to store selected information from a model file to interface with programs of your own or to transfer information between different types of computers [96]

Universal files are designed so that they may be easily read and written using user-written programs. Each universal file is a sequential formatted file with records of

maximum 80 characters. The file is compatible with the text editors on PC and UNIX systems.

Blocks of information, called data sets make up the basic structure of a universal file. Each block begins with a row, containing a minus sign in column 5 and a 1 in column 6 (-1). The second row has a number, which is the data set number. For example, the connection grid specifications for mesh volumes are written with data set number 785. Following the data type record, the body of the data set contains data, which is dependent on the data set type. The final record of the data set is the similar row with a "-1". The following is the beginning statement of all the universal files (unv.). "151" is the data set number, which is the header data set. The model file from which the mesh is generated, the modeler version number and date of processing are given. The data set ends with the -1 as usual.

```
-1
151
/disk1/for/example/purposes/dtc1
I-DEAS Master Series 21m1: Simulation
02-Apr-03 10:07:33
I-DEAS Master Series 21m1: Simulation
03-Apr-03 07:36:12 8 1 3 1 3
-1
```

Figure A.1 Data Set Example for UNV Format

The surface and volume meshes are generated on I-DEAS and the mesh files are exported using universal file format. Most of the information is skipped since it involves properties, which are not relevant to the investigation such as material properties. The node positions and element connectivity is removed from the universal file and written to a separate file with ".haz" extension. The data set code numbers for these data are 2411 for the "node positions" and 2412 for the "element connectivity" information. The format rules for these data set numbers are listed here from the I-DEAS help files for convenience.

If the mesh generation is done by parts in the I-DEAS program then the mesh numbers are not sorted and some of the unused meshes are deleted. Therefore a sorting algorithm has been added to the “mesh_3d” program.

A.1.2 Universal Data Set Number 2411

Node positions in any coordinate system are given under this data set.

Table A.1 Format for Data Set 2411

Record 1	FORMAT (4I10)	
	Field 1	node label
	Field 2	export coordinate system number
	Field 3	displacement coordinate system number
	Field 4	color of each node
Record 2	FORMAT (1P3D25.16)	
	Fields 1-3	node coordinates

A.1.3 Universal Data Set Number 2412

Connectivity of each element is defined under this data set.

Table A.2 Format for Data Set 2412

Record 1	FORMAT (6I10)	
	Field 1	element label
	Field 2	FE descriptor ID (111 tetrahedron)
	Field 3	physical property table number (1-2d, 2-3d)
	Field 4	material property table number
	Field 5	color
	Field 6	number of nodes on the element
Record 2	FORMAT (8I10)	
	Fields 1 to n	node labels defining element

A.1.4 Defining Boundary Conditions

The boundary conditions are required separately for the flow solver (H-IBS3D) and the FMM solver. They are defined by modifying the color of each node and surface element, which lays on the boundary. Similarly the inner nodes are also defined by using the color codes in I-DEAS. For a list of color codes see Table below.

Table A.3 Color Codes for Boundary Condition Definitions in I-DEAS

HIBS3D BC	Color	No
Pressure Exit	Gray Blue	2
Symmetry	Light Blue	3
Mass Flux	Green	7
Flow Inlet	Orange	10
Propellant Symmetry Surface	Light Blue	3
Wall	Light Magenta	13
Propellant on Wall Surface	Light Magenta	13
Metal Case	Light Magenta	13
Wall with Surface Regression	Pink	14
FMM3 VOLUME NODE COLORS		
Flow field	Blue	1
Propellant itself	Red	11
Nozzle Insulation	Magenta	12
FMM3 AREA NODE COLORS		
Void and Crack	Black	0
Flow Field	Blue	1
Propellant Burn Surface	Green	7
Nozzle Insulation Burn Surface	Yellow	8
Metal Case	White	15

A.1.5 Program “ MESH_3D ”

This FORTRAN program reads the node positions and element connectivity information from universal file format and converts the mesh information into a

compressed form for the Fast Marching Algorithms to work on. This new mesh file has extension of “ha2”.

The “ha2” format contains information on node positions, connectivity of elements and neighboring elements. This information is gathered from two sources:

1. In the “unv” file format the connectivity is used to gather information about neighboring cells to each face of the tetrahedron element.
2. In the “unv” file format the color codes of each surface element is defined according to the boundary conditions they represent. This information is used and if a boundary condition exists, negative of the color code is adopted as the neighborhood information.

```

6c4.ha2 - Notepad
File Edit Format Help
INPUT MESH FILE= 6c4_a11.unv
NEIGH= 1, DIM= 3, SOURCE = IDEAS
12311/NUMBER OF NODES.
56529/NUMBER OF ELEMENTS.
1/THE NODE POSITIONS
1 2581.000000000000 559.96581570000 25.000000000000 7
2 2581.000000000000 547.90199458000 25.000000000000 7
3 2181.000000000000 515.32129032000 25.000000000000 7
2/CONNECTIVITY OF EACH ELEMENT
1 157 338 333 1836 48416 1150 -3 2
2 157 162 338 1836 4 1149 1 3
  
```

Figure A.2 Example HA2 File

The neighboring information is given in the connectivity list. Negative values represent boundary conditions and positive numbers represent elements in the neighborhood. Neighbors are sorted as follows:

For a tetrahedron element

- Neighbor 1 : Nodes (1 – 2 – 3)
- Neighbor 2 : Nodes (2 – 3 – 4)
- Neighbor 3 : Nodes (3 – 4 – 1)

Neighbor 4 : Nodes (4 – 1 – 2)

For a hexahedron element

Neighbor 1 : Nodes (1 – 2 – 3 – 4)

Neighbor 2 : Nodes (5 – 6 – 7 – 8)

Neighbor 3 : Nodes (1 – 2 – 5 – 6)

Neighbor 4 : Nodes (2 – 3 – 6 – 7)

Neighbor 5 : Nodes (3 – 4 – 7 – 8)

Neighbor 6 : Nodes (4 – 1 – 8 – 5)

For hybrid elements the neighboring connectivity for hexahedrons is used strictly at all times.

A.2 PROCESSING STEP

The HA2 format file is the mesh file for the interface capturing algorithms. The input file given in Figure A.3 is used for the interface capturing code FMM3. The variables are described in Table A.4.

Table A.4 Parameters for Fast Marching Input File

TVALUE	Initial Time Value
FTYPE	Burn rate type (0=Constant burn rate / 1=Pressure dependent burn rate)
F	Constant burn rate
RBREF	Reference burn rate
PREF	Reference pressure for burn rate calculations
PEX	Pressure exponent for burn rate calculations
RHO_PROP	Propellant density

Table 6.4 (Continued) Parameters for Fast Marching Input File

PINIT	Initial pressure in the motor
IT_SHOW	Iteration show interval
SCALE	Mesh dimension multiplier (Used for unit conversion)
I_CAPT	Volume capturing. (0=N0 / 1=Yes)
IMOVE	Movable inner nodes. (0=N0 / 1=Yes)
N_T0	Number of captured time
T0	Capture time
IRUN	Flow solver coupled . (0=N0 / 1=Yes)
IPRE	Previous run to be used for later runs
IAREA	Capture area vs. web data. (0=N0 / 1=Yes)
DT	Capturing time interval
IPERI	Capture perimeter vs. web for stations data. (0=N0 / 1=Yes)
IAXIS	Along which axis the stations are placed for perimeter capturing. (1=x-axis / 2=y-axis / 3=z-axis)
ISKIN	Solve interface on the boundaries for 3-d surfaces (0=N0 / 1=Yes)
IWHAT	Solve for interface capturing. (0=N0 / 1=Yes)
IORDER	Order of accuracy for fast marching method (1=first order / 2=second order)

```
input_FM33.txt - Notepad
File Edit Format Help
*****
*FM3 INPUT FILE
*****
c_6c4_4.ha2      /Mesh File
SIL_out.plt     /output file
sil.ha4         /HA4 file
SIL_c.plt       /Captured tecplot file
sil_c.unv       /Captured unv file
50000          /ITER:
5000.0d00       /TVALUE:
1.0001         /IVM
0              /FTYPE
10.2329d00     /F
10.2329d00     /RBREF
13.7900d06     /PREF
0.100d00       /PEX
1111.9d0       /RHO_PROP
99.90d03       /PINIT
250            /IT_SHOW
1.00d0         /SCALE
1              /I_CAPT
0              /IMOVE
1              /NUMBER OF CAPTURED TIME STEPS
0.1D0         /CAPTURE TIME 1
0              /RUN_SOLVER
0              /IPRE
***CAPTURE AREA VS WEB
0              /IAREA
0.025         /DT
SIL_AREA.PLT   /Captured AREA file
***CAPTURE PERIMETER VS WEB FOR STATIONS
0              /IPERI
SIL_PER.PLT    /PERIMETER FILE
SIL_PORT.PLT   /PORT AREA
3              /AT WHICH AXIS SLICES WILL BE TAKEN 1=X,2=Y,3=Z
2              /# OF STATIONS
198.0         /1
400.0         /2
***SOLVER NUMERIC OPTIONS
1              /ISKIN
```

Figure A.3 Fast Marching Code Input File

The input file name is “input_fm3.txt” at all times. When the FMM3 program runs it generate several output files. These files are given as:

Table A.5 Output files of Fast Marching Program

*_out.plt	The mesh file with time values to define interface location
*.ha4	Capture mesh file for flow solver
*_c.plt	Captured domain mesh file for Tecplot presentation
*_c.unv	Universal file for captured domain.
*_area.plt	Area vs. web data file
*_per.plt	Perimeter vs. web along stations data file
*_por.plt	Port area at each station data file

APPENDIX B

THREE DIMENSIONAL SOLUTIONS

B.1 6C4 RESULTS

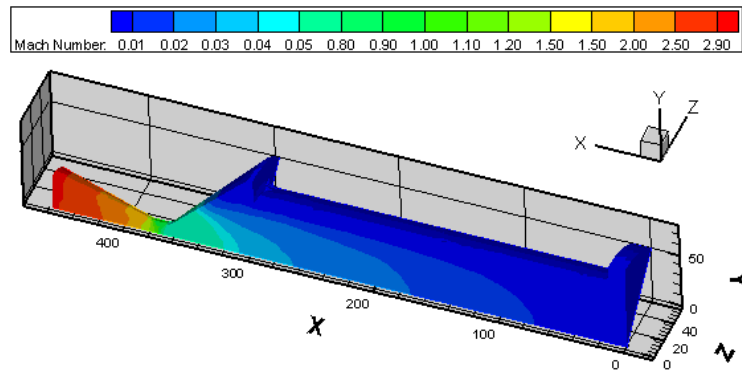


Figure B.1 6C4 Mach Contours at 0.2 s

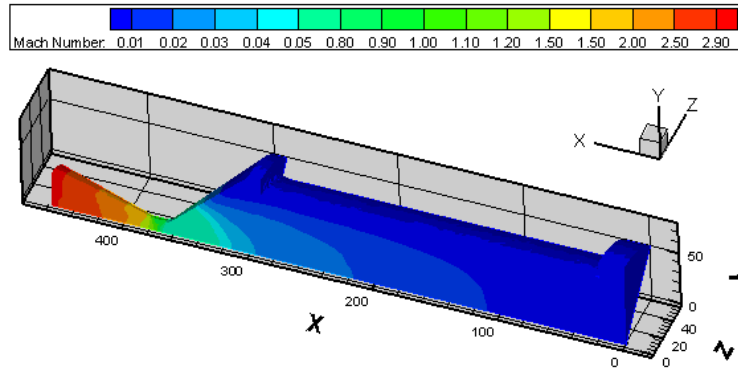


Figure B.2 6C4 Mach Contours at 0.6 s.

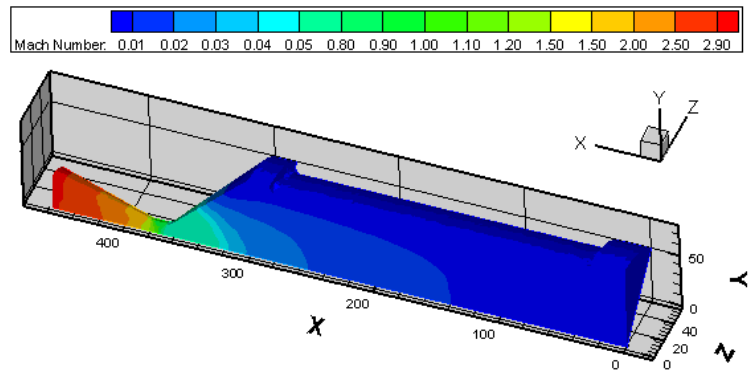


Figure B.3 6C4 Mach Contours at 1.0 s

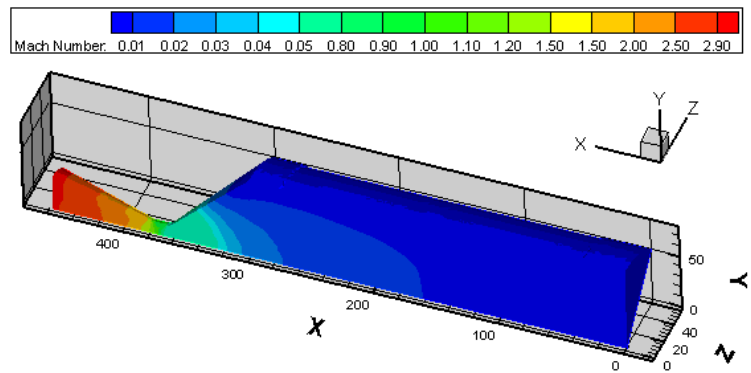


Figure B.4 6C4 Mach Contours at 1.4 s

B.2 BOOSTER SRM RESULTS

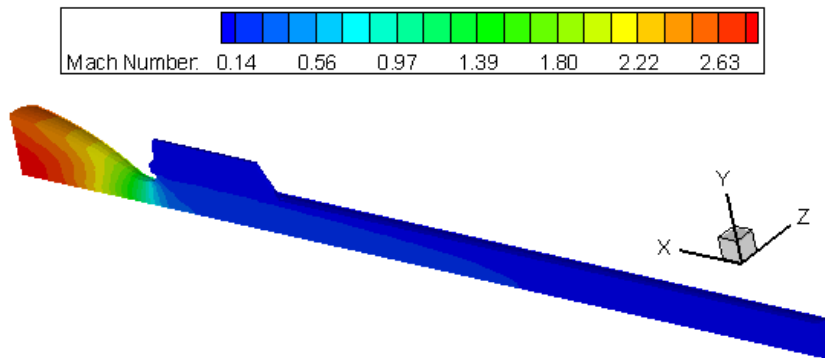


Figure B.5 Booster Mach Contours at 0.5 s

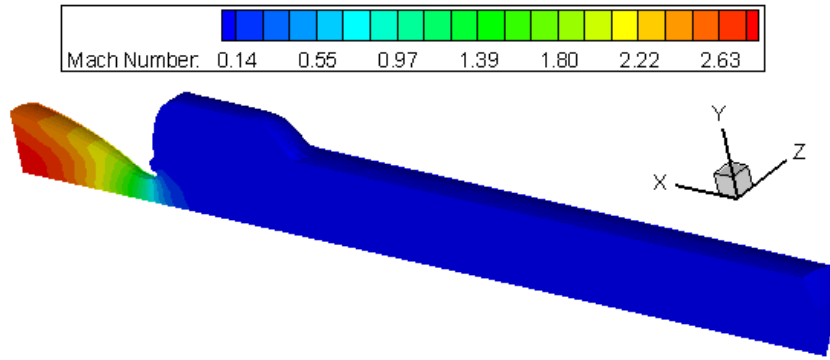


Figure B.6 Booster Mach Contours at 18.18 s

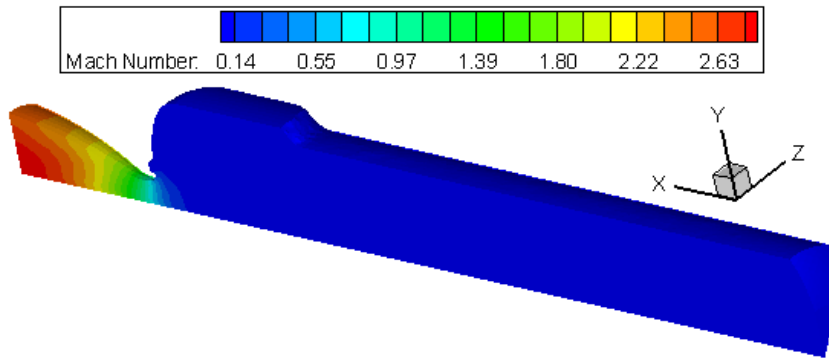


Figure B.7 Booster Mach Contours at 24.96 s

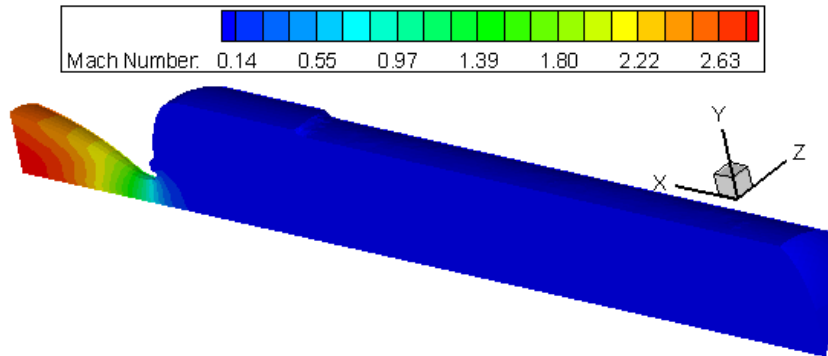


Figure B.8 Booster Mach Contours at 28.86 s

B.3 BUBBLE IN PROPELLANT RESULTS

Grain burn surface arrival time, for a burn rate of 1.0 mm/s is given below.

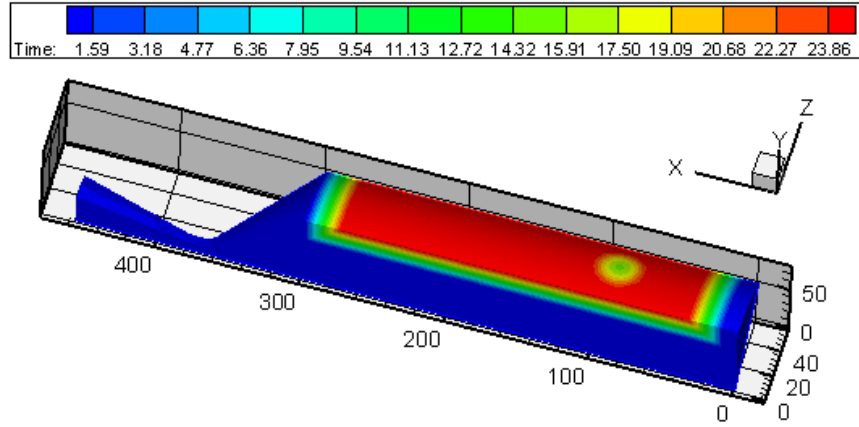


Figure B.9 Single Void in 6C4 Grain

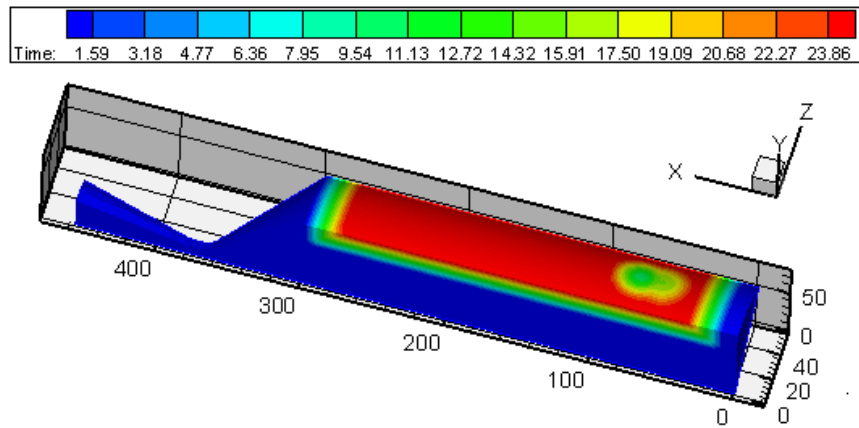


Figure B.10 3 Voids in 6C4 Grain

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: TOKER, Kemal Atilgan
Nationality: Turkish (TC)
Date and Place of Birth: 23 May 1974 , Ankara
Marital Status: Married
Phone: +90-312-8634200
Fax: +90-312-8634208
E-mail: atoker@rkts.com.tr / keattok@yahoo.com

EDUCATION

Degree	Institution	Year of Graduation
MS	METU, Mechanical Engineering	2000
BS	METU, Mechanical Engineering	1997
High School	Bursa Anadolu Lisesi, Ankara	1992

WORK EXPERIENCE

Year	Place	Enrolment
2000- Present	Roketsan A.Ş.	Eng. Development Group
1997-2000	METU, Mechanical Engineering	Research Assistant

FOREIGN LANGUAGES

English

PUBLICATIONS

1. K.A. TOKER, U. ARKUN, T. TINAZTEPE, “*A Method for Estimating Space Launch Vehicle Parameters and Technological Assessment*”, Recent Advances in Space Technology, Istanbul, Turkey, 2003
2. K.A. TOKER, U. ARKUN, T. TINAZTEPE, “*Roketsan’s Technological Capabilities for Space Propulsion Applications*”, Istanbul Space Propulsion Workshop, Istanbul, Turkey, 2004
3. . K.A. TOKER, H. AKSEL, T. TINAZTEPE “*3-dimensional Propellant Grain Burnback Calculations on Tetrahedron Mesh by Fast Marching Method*”, AIAA-2004-4960, 22nd Applied Aerodynamics Conference, RI, USA, 2004