

Internal Ballistic Modelling of Solid Rocket Motors Using Level Set Methods for Simulating Grain Burnback

by

Mvuyisi Humphrey Tshokotsha

Thesis presented in partial fulfilment of the requirements for the degree
Master of Science in Applied Mathematics at
Stellenbosch University



Supervisor: Dr. M.F. Maritz
Faculty of Science
Division of Applied Mathematics
Department of Mathematical Sciences

2016

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Signature: Mvuyisi Humphrey Tshokotsha

Date: March 2016

Abstract

Solid Rocket Motors are propulsive systems providing thrust from products exiting through a nozzle, obtained by burning a solid fuel (the grain) inside a chamber. Various thrust profiles can be obtained by different initial grain geometries. The purpose of numerical modelling, is to predict the thrust curve, given an initial geometry.

A simplified model for Solid Rocket Motor behaviour is developed. The grain geometry is represented by a signed distance function, and the burning surface geometry is calculated using a numerical solution of the burning process based on level-set methods.

The gas flux is modelled using a 1-D approximation of the gas dynamics inside the chamber, following a method proposed by Lamberty.

2D and 3D models for modelling the regression of the grain surface over time are developed. The evolution of different grain geometries are simulated and the results obtained are compared to the experimental data available.

The burning surface and port area for some grain geometries is calculated analytically and the results found are compared with the ones found numerically using 2D model of the level set method.

Opsomming

Vastevuurpyl motore is 'n aandrywingstelsel wat stukrag verskaf vanaf gas wat by 'n spuitstuk uitspuit, waar die gas van die verbranding van 'n vaste brandstof (genoem die grein) in 'n ontbrandingskamer afkomstig is. Verskeie stukrag krommes kan deur verskillende aanvanklike grein geometrie's verkry word. Die doel van numeriese modellering is om die stukrag kromme te voorspel uit die aanvanklike grein-geometrie.

'n Vereenvoudigde model vir Vastevuurpyl motorgedrag word ontwikkel. Die grein geometrie word deur 'n tekenafhanklike afstandsfunksie voorgestel en die brandoppervlak word deur numeriese oplossing van die brandproses, gebaseer op kontoervlak-metodes, bereken.

Die gasvloei word met 'n 1D benadering van die gasdinamika binne die ontbrandingskamer gemodelleer, geskoei op 'n metode voorgestel deur Lamberty.

2D en 3D modelle vir die tydsafhanklike wegbrand van die greinoppervlak word ontwikkel. Die evolusie van verskillende grein geometrie's word gesimuleer en die resultate word met beskikbare eksperimentele data vergelyk.

Die area van die brandoppervlak, sowel as die kanaalvloeiarea word analities bereken vir 'n aantal grein-geometrie's en dit word vergelyk met die ooreenstemmende areas soos numeries bereken deur die 2D model van die kontoervlak-metode.

Dedications

To Eve and my family

Acknowledgements

I would like to express my deepest thanks and gratitude to Dr Milton F Maritz for his supervision, understanding and constant guidance.

I would like to thank Dr Willie Brink for his crucial suggestions during the study.

I would like to express my sincere appreciation to Dr Werner Rousseau for supplying experimental data and partially supporting this study.

I would like to thank my colleagues Eyaya Eneyew and Jan-At Engelbrecht for their invaluable suggestions and supports.

Generous support from FLUXION STUDY GRANT GAUF600.11214.08400.084AA.013-GRANTS is acknowledged.

Nomenclature

Abbreviations and Acronyms

SDF	Signed Distance Function
SLIC	Simple Line Interface Calculation
SRM	Solid Rocket Motor
STL	Stereolithography file
VOF	Volume of Fluid

Roman letters

A	Area
A_t	Throat Area
C^*	Characteristic Velocity
c	Burn rate coefficient
D_{port}	Port diameter
D_{out}	Outer diameter
d	Euclidean distance function
L	Length of the propellant grain
l	Length of the slot in a slotted grain
M	Mach number
\dot{m}	Mass flow rate
n	number of slots
\mathbf{n}	Normal vector
\bar{n}	Burn rate exponent

P	Pressure
\bar{P}	Pressure ratio
R	Gas Constant
r	slot diameter
\dot{r}	Burn rate velocity
T	Temperature
T_f	Flame Temperature
u	Velocity of the flow field
V	Speed to advance the interface
w	Web diameter
\boldsymbol{x}	Position vector

Greek Symbols

β	Web length
γ	Specific heat ratio
Γ	General Interface
ν	Volume
ρ	Density
τ	Chamber filling time
ϕ	Signed distance function
Ω^+	Region outside the interface
Ω^-	Region inside the interface

Superscripts

*	Quantity at the throat
---	------------------------

Subscripts

a	Quantity along the axis of the SRM
s	Quantity at stagnation/burning surface
noz	Quantity at the nozzle
tot	Total

Contents

Declaration	i
Abstract	ii
Opsomming	iii
Dedication	iv
Acknowledgements	v
Nomenclature	vi
List of Figures	xvii
List of Tables	xviii
1 Introduction	1
1.1 Aims and Objectives of the Study	2
1.2 Motivation of the Study	3
1.3 Parts of the Solid Propellant Rocket Motor	4
1.3.1 Motor Case	4
1.3.2 Igniter	5

1.3.3	Nozzle	5
1.3.4	Propellant Grain	5
1.4	Typical Configurations of the Propellant Grain	6
1.4.1	Internal-Burning Tube (Tubular) Grain	6
1.4.2	Star Grain	7
1.4.3	Slotted Grain	8
1.5	Anchor Grain	9
1.6	Stereolithography (STL) Representation of the Surface	10
1.6.1	Vertex to Vertex Rule	12
1.6.2	Common STL Errors	12
1.7	Thesis Layout	15
2	Burnback Modelling using Level Set Approach	17
2.1	Methods Used for the Evolution of Interfaces	17
2.2	Equations of Motion for Moving Interfaces	18
2.2.1	A Boundary Value Formulation	20
2.2.2	The Initial Value Formulation	23
2.3	Advantages of Using Level Set Methods	25
2.3.1	Topological Changes	26
2.4	Theory of Curves and Surface Evolution	30
2.4.1	The Lagrangian Formulation of a Moving Interface	30
2.4.2	The Marker Point Methods for the Evolution of Interfaces	31

2.4.3	Volume of Fluid Technique	35
2.5	General Method of Finding Shortest Distance Function	39
2.6	Analytical Calculation of the Burning Surface Area and Port Area	42
2.6.1	Equations to Find Burning Surface Area and Port Area of a Tubular Grain Analytically	42
2.6.2	Equations to Find Burning Surface Area and Port Area of a Star Grain Analytically	43
2.7	Calculation of the Signed Distance Function in 2D	46
2.8	A Simple Method to Calculate the Signed Distance Function in 3D	51
2.9	Calculation of the Signed Distance Function in 3D Using Projection Geometry	52
2.10	2D Modelling of the Burnback in MATLAB	55
2.10.1	Modelling the Casing	60
2.11	Implementation of the Level Set Method in 3D to Model burnback of the Solid Grain	61
3	Gas Dynamics and Internal Ballistics	63
3.1	Stagnation Properties	63
3.2	Isentropic Flow of Gas	64
3.3	Mass Flow Rate	66
3.4	Area Ratio	69
3.5	Flow Behaviour in the Converging-Diverging Nozzles	71
3.6	Calculation of Flow in Nozzles	74
3.7	Internal Ballistics Parameters	75

3.7.1	Burning Rate	75
3.7.2	Characteristic Velocity	75
3.7.3	Chamber Pressure	76
3.8	Internal Flow Modelling of Solid Rockets Using Gas Dynamics	76
3.8.1	Grid Generation and the Burned Away Distance	77
3.8.2	Gas Dynamics Used to Model the Flow Inside the Combustion Chamber	78
3.8.3	Gas Dynamics in the Nozzle to Obtain Head Pressure for the Next Time Step	79
4	Results	81
4.1	Internal Burning Grain	82
4.2	Star Grain	83
4.3	Slotted Grain	86
4.4	Anchor Grain	92
5	Conclusions	95
	Bibliography	97
	Appendix	100
A	Projection Geometry	100
A.1	The Calculation of the Signed Distance Function in 2D Using Projection Geometry	100
A.2	The Calculation of the Signed Distance Function in 3D Using Projection Geometry	102

B Zero Contour Code

List of Figures

1.1 Solid Rocket Motor	2
1.2 Thrust-time curves for tubular (left) and End-burning (right) grain configurations	3
1.3 Tubular grain configuration	7
1.4 Star grain configuration	7
1.5 Slotted grain configuration	8
1.6 Anchor grain configuration	9
1.7 STL representation of a sphere	11
1.8 A violation of the vertex to vertex rule	12
1.9 Crossed facets in 3D space.	13
1.10 Mismatching surface patches	14
1.11 Co-linear vertices	14
1.12 Vertices at the same point	15
2.1 Curve (Γ) propagating with speed V in normal direction.	19
2.2 Calculation of crossing time at (x, y) for expanding interface with $V > 0$. Γ is the interface separating the outside region (Ω^+) from the inside region (Ω^-).	21

2.3	Setup for boundary value formulation.	21
2.4	Transformation of an interface motion into boundary value problem.	22
2.5	Transformation of an interface motion into initial value problem.	23
2.6	Interface made up of two circles that are evolving with a speed V	26
2.7	ϕ at $t = 0$ for the two circles.	27
2.8	The approximation of the original surface given by the values of $\phi = 0$	27
2.9	The surface made up of two circles merging to form one interface.	28
2.10	$\phi(x, y, t)$ for the two circles merging to form one interface.	29
2.11	The contour on the plane $\phi = 0$ approximating the surface time t	29
2.12	Parametrized view of a propagating curve	31
2.13	The discrete parametrization of a propagating curve	32
2.14	The evolution of set of points on the interface at time $t = 0$ and at time $t = 1$	34
2.15	The evolution of a cusp of an interface at time $t = 0$ and at time $t = 1$	34
2.16	The evolution of a corner of an interface at time $t = 0$ and at time $t = 1$	35
2.17	A-original surface separating two regions and associated volume fractions. B& C-Reconstructed interfaces using rectangular and piecewise linear frac- tional volume methods	36
2.18	Interface evolution using volume of fluid technique	37
2.19	\mathbf{x}_c as the shortest distance to both \mathbf{x} and \mathbf{y}	39
2.20	Grid points (red) and their shortest distances shown by a blue line	42
2.21	Tubular grain evolving with a distance function f	43
2.22	Star Geometry and its defining parameters (figure taken directly from [14]).	44

2.23	Star defining parameters when the burning front hits the casing (figure taken directly from [14]).	45
2.24	Node \mathbf{x}_j as the closest point to the grid point \mathbf{x} labelled A	47
2.25	Shortest distance $\phi(\mathbf{x})$ obtained by projection	48
2.26	Original interface and the distances of the grid points from the interface . .	49
2.27	Original interface (blue) and its approximation (red).	50
2.28	Evolving interface	50
2.29	\mathbf{x}_c as the centre of the closet triangle to the grid point \mathbf{x}	51
2.30	Point \mathbf{x} , its closest point \mathbf{x}_0 , and its neighbours $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_5$	52
2.31	A projection of vector \mathbf{q} onto vector \mathbf{a}_k	53
2.32	An Icosahedron an its normals pointing out of the surface (red line segments) 54	
2.33	Reconstructed surface using <code>isosurface</code> that approximates the original Icosahedron	55
2.34	Star grain made up of planar triangular mesh	56
2.35	Mandrel of a star grain made up of a triangular mesh	56
2.36	SDF values of a star grain and the interface obtained from them	57
2.37	Data points (left) of the contour (right) at zero level obtained using <code>ZeroContour</code>	58
2.38	Patched surface (right) between polygons (left) of a cell	58
2.39	Burning surface of a cell	59
2.40	Contour obtained with <code>ZeroContour</code> (left) and interpolated equidistant data points (right)	59
2.41	Casing modelling (magenta), evolving star (black), remaining surface (blue) 60	

2.42	The burning surface area left in a burning star grain	60
2.43	Polygon (in blue) representing a port area (left) and the burning surface area of a cell (right)	61
2.44	Portion of the triangle inside	62
3.1	Nozzle operating at steady-state	64
3.2	Flow through a converging nozzle	68
3.3	Area as a function of Mach number	70
3.4	Pressure as a function of position for flow in converging-diverging nozzle with different values of the nozzle exit pressure	71
3.5	Flow patterns in the nozzle	72
3.6	Internal Ballistics element method	77
4.1	Evolution of a tubular grain	82
4.2	Pressure-Time curve for a tubular grain, red is obtained from experimental data, tubular grain modelled using 2D level sets (green) and 3D level sets (blue)	82
4.3	Burning surface area as a function of time (left) and Port area as a function of time (right) for a tubular grain	83
4.4	Polygons (blue) and their approximations (red) of the mandrel of a star grain	84
4.5	Evolution of a star grain	84
4.6	Pressure versus time graph of the star grain	85
4.7	Burning surface area as a function of time (left) and Port area as a function of time (right) for a star grain	86
4.8	Slotted grain	87

4.9	Mandrel of the Slotted grain	88
4.10	Slotted grain cells (left) and polygons obtained (right)	88
4.11	Slotted grain burned for some time	89
4.12	Evolution of part A	89
4.13	Evolution of part B	90
4.14	Evolution of part C	90
4.15	Evolution of part D	91
4.16	Pressure versus time graph of the slotted grain, red is the experimental data, green comes from the 2D model for burnback	91
4.17	The mandrel of anchor divided into cells (left) and polygons obtained from the intersection of the plane and the mandrel (blue) and their approxima- tions (red) (right).	92
4.18	The anchor grain evolution	93
4.19	Pressure versus time graph of the anchor grain	93
4.20	Burning anchor grain at Point A	94
A.1	The point closest to \mathbf{b} on the line determine by \mathbf{a}	100
A.2	The point closest to \mathbf{b} on the plane determined by \mathbf{a}_1 and \mathbf{a}_2	102

List of Tables

1.1	Parameters defining a slotted propellant grain	9
1.2	Parameters defining an anchor grain	10
2.1	Summary of boundary value and initial value formulation of the level set method	25
4.1	Constants used for modelling the burnback in all examples	81
4.2	Dimensions of the slotted propellant grain	86
4.3	Values of the parameters of the double anchor grain	92

Chapter 1

Introduction

A Solid Rocket Motor (SRM), as its name implies, is a rocket where the propellant of the motor is in the solid state. The oxidizer and the fuel are premixed and are contained and stored directly in the combustion chamber. Since the solid propellant includes both fuel and oxidizer, solid propellant rocket motors can operate in all environmental conditions. In comparison to other types of rockets, solid propellant rocket motors have simple design, are easy to operate and require little or no maintenance [9].

When compared to other types of rocket motors, solid propellant rocket motors are the most commonly used one due to its relatively simple design, high reliability, ease of manufacture and has a relative lower price. Solid Rocket Motors (SRM's) can be used for a wide variety of applications requiring a wide range of size and duration of the thrust [20].

The schematic diagram of a solid propellant rocket motor is shown in Figure 1.1, showing the ignition charge (initiates combustion), casing (protects the inner contents of the motor), grain (propellant that burns to obtain thrust) and nozzle (discharges the hot gases coming from inside the combustion chamber). These parts of the SRM are discussed in detail in section 1.3.

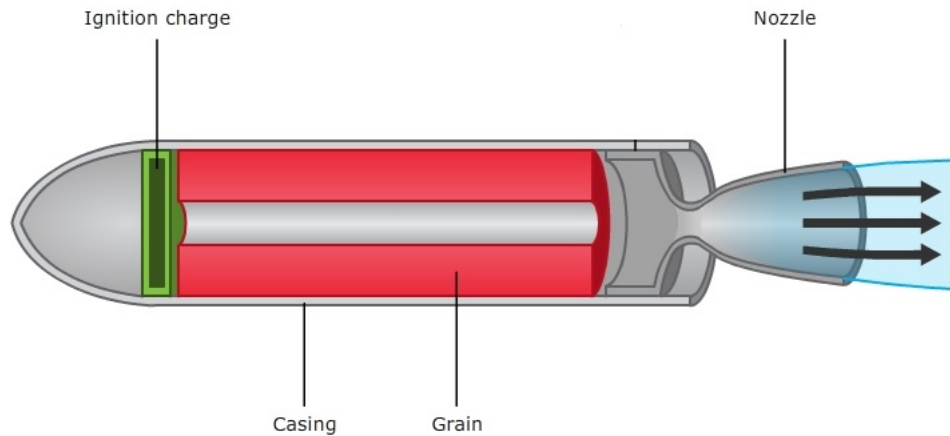


Figure 1.1. Solid Rocket Motor

1.1 Aims and Objectives of the Study

The main objectives of this study is

- to explore some existing models of SRM burnback,
- to implement one such a model in computer code,
- for a given grain geometry, to simulate burnback together with a simplified internal (1D) ballistics model for the gas flow in order to arrive at a predicted pressure-time curve,
- to compare the results obtained using level set methods with the analytical results for the port area and the burning surface area,
- to compare the predicted pressure-time curve with some available experimental data.

In particular, modelling of the burnback with Level Set Methods based on level surfaces of the Signed Distance Function (SDF) is investigated. Two different surface representation

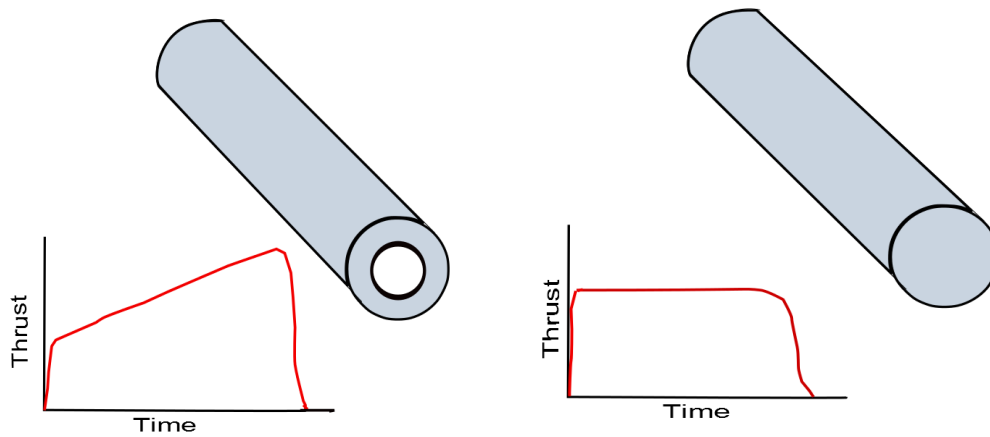


Figure 1.2. Thrust-time curves for tubular (left) and End-burning (right) grain configurations

schemes are implemented, viz. (a) a fully 3D triangular mesh representation and (b) a simplified 2D representation based on polygonal approximation of slices through the grain (orthogonal to the motor axis). The second representation is believed to be a novel contribution.

Gas flow is simulated with a 1D model, provided by Lamberty [21]. The thrust-time curves for tubular and end-burning grain configurations are shown in Figure 1.2

1.2 Motivation of the Study

Many studies have been conducted in order to optimize motor design and operation. These studies constitute the design and development phase of the Solid Rocket Motors (SRM's). Technological goals as well as commercial aims have impelled the research of SRM.

When the design bases are defined, the optimum configuration is selected and then the critical review of the design configuration begins. In this phase, in order to predict the full-scale motor operation, selected propellant, grain design, and motor configuration are analysed in detail [23]. The basic tool to perform this analysis combines the theoretical performance prediction methods with numerical models implemented in simulation codes

that simulate the SRM grain burnback and motor performance. To verify the results obtained, it is necessary to match it with experimental data obtained from test firings which are often risky and expensive.

Therefore, in the design and development of SRM's, the prediction of grain burnback as well as internal ballistics prediction capabilities can enhance SRM reliability.

In this project analytical methods are used to calculate the burning surface area and port area in order to validate the accuracy of the 2D model obtained using level set methods.

1.3 Parts of the Solid Propellant Rocket Motor

1.3.1 Motor Case

In general, a motor case is a cylindrical cover containing the solid propellant, igniter and insulator. The combustion takes place inside the motor case; it is therefore sometimes referred to it as a combustion chamber.

The case must be capable of withstanding the internal pressure resulting from the motor operation, 3-25 MPa, with a sufficient safety factor. Therefore, the motor case is usually made either from metal (high-resistance steels or high strength aluminium alloys) or from composite materials (glass, kevlar, carbon) [9]. In addition to the stresses due to the pressure in the chamber, thermal stresses may sometimes be critical and, when the case also serves as the flight vehicle body, bending loads and inertial forces also play an important role in determining the thickness and the material of the motor case.

The combustion process produces high temperatures, ranging from approximately 2000 K to 3500 K, therefore the motor case needs to be insulated. Typical insulator materials have low thermal conductivity, high heat capacity and they are usually capable of ablative cooling. The most commonly used insulation material is EPDM (Ethylene Propylene Diene Monomer) with addition of reinforcing materials [9].

1.3.2 Igniter

The ignition system gives the energy to the propellant surface necessary to initiate combustion. Ignition usually starts with an electrical signal. The ignition charge have a high specific energy and it is designed to release either gases or solid particles. Convectonal heat releasing compounds are usually pyrotechnic materials, black powder, or metal-oxidant formulations [17, 8, 5].

1.3.3 Nozzle

High temperature, high pressure combustion products are discharged through the converging-diverging nozzle. In this way, the chemical energy of the propellant is converted to kinetic energy and thrust is obtained. The geometry of the nozzle directly determines how much of the total energy is converted to kinetic energy. Therefore, nozzle design plays a very important role with respect to the performance of a rocket motor [4].

Nozzles are usually classified according to their structural mounting technique or shape of the contour, such as a submerged nozzle, a movable nozzle, or a bell-shaped nozzle.

Combustion products have an erosive effect because of their high temperature and high velocity and also because of the high concentration of liquid and solid particles such as metal oxides inside them. Material selection of the nozzle is a very important step of the nozzle design, especially for the throat region where erosive effects are more dominant. Refractory metal, composite materials with a high carbon content or graphite and reinforced plastic that will withstand erosive effects are commonly used as throat material.

1.3.4 Propellant Grain

Solid propellant is cast in a certain configuration and is called the propellant grain.

The propellant grains can be sub-categorized into two main configurations; case-bonded grain and free-standing grain. Case-bonded grains are directly cast into the motor case

already provided with thermal insulation. After the curing time, the propellant grain is complete and this motor case with the propellant grain is ready to be mounted with the other components of the motor. Free-standing grains are not directly cast into the motor case, but instead, the propellant is cast in some special mould or extruded. When the cure process of the propellant is completed, the grain is extracted from this mould. Free-standing grains are loaded to the insulated motor case on the assembly line. That is why they are sometimes called cartridge-loaded grains [20].

The burning surface of the grain changes during motor operation as the propellant burns [5]. Burnback analysis determines this change in the grain geometry. The geometric design of the grain ultimately defines the performance characteristics that can be obtained with a given propellant type and nozzle.

1.4 Typical Configurations of the Propellant Grain

For different system missions, different thrust-time profiles are required for the rocket motors, such as progressive, neutral or boost and sustain mode. By changing the propellant grain configuration, different thrust-time profiles can be obtained from a rocket motor. Grain configurations can be categorized in several different ways [6]:

- Inner geometry (star, wagon, internal burning tube, etc)
- Outer shape of the grain (Tubular or Spherical)
- The propellant used (single propellant or dual propellant)

The details of the most commonly used grain configurations in the SRM application are given in the subsequent sections.

1.4.1 Internal-Burning Tube (Tubular) Grain

Figure 1.3 shows the internal-burning tube which is one of the most practical and preferred grain configurations. It is a radially burning grain that burns progressively. It is typically

case bonded which inhibits burning of the outer surface. The internal-burning tube is defined by length L and two diameters D_{out} and D_{port} . Tubular grain produces a progressive thrust-time curve.

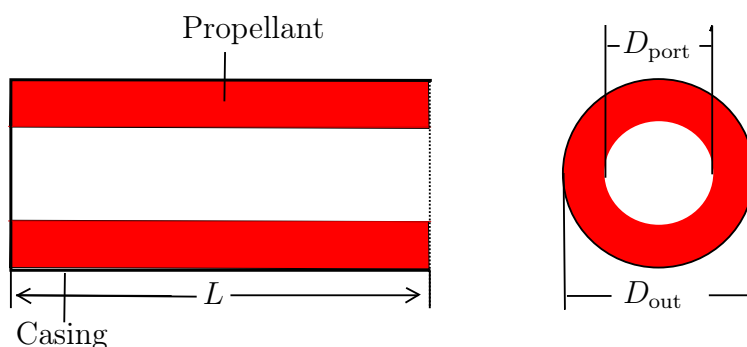


Figure 1.3. Tubular grain configuration

1.4.2 Star Grain

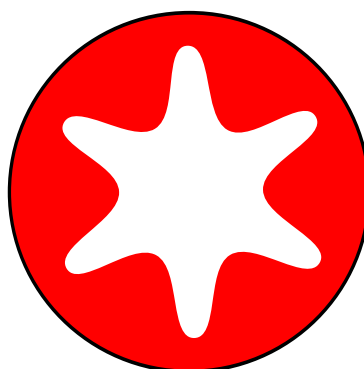


Figure 1.4. Star grain configuration

The star burning grain have a series of points protruding inward, as shown in Figure 1.4, such that while the points burn off, they keep the area roughly constant. In a first approximation one can see that the periphery of the star should be equal to the outer (circular) periphery of the grain, so that the burning area is equal at the beginning and the end [7, 37]. The thrust-time curve of a star grain is neutral.

1.4.3 Slotted Grain

The slotted-tube configuration, as shown in Figure 1.5, consists of a cylindrical tube of propellant which has been cast into, or which has been cut into a number of slots. These

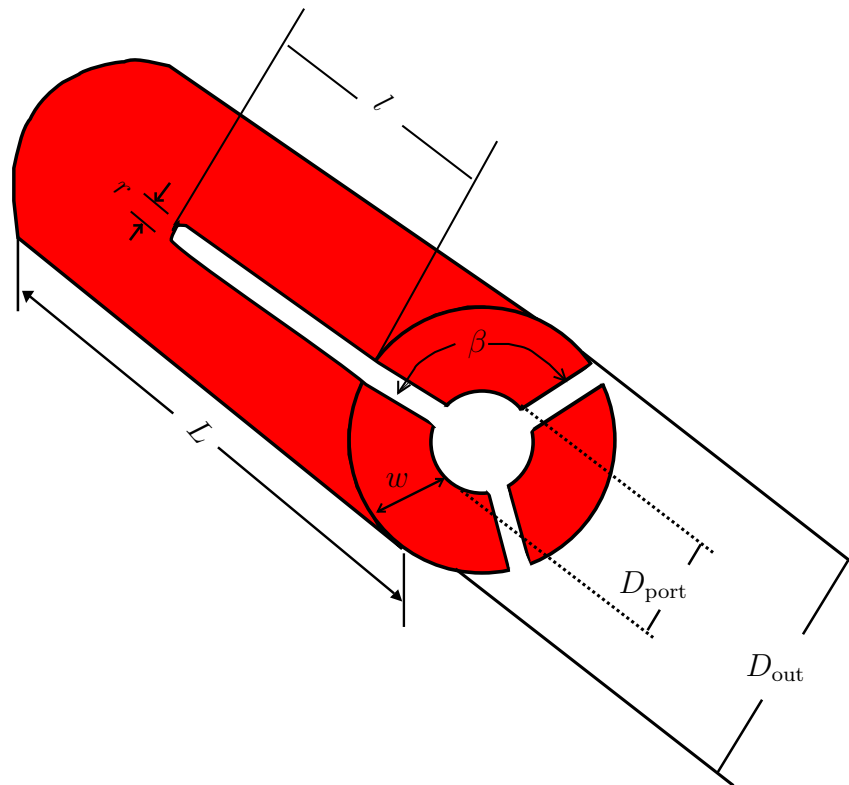


Figure 1.5. Slotted grain configuration

slots connect the inner and outer surface of the tube and extend to part of its length. The configuration offers some striking advantages for the ballistician. Perhaps the most obvious one is its inherent lack of sliver (small remaining pieces of the grain as it burns to completion) in a non-erosive situation since it is basically an internal-burning cylinder [34]. The parameters that describe the slotted grain configuration are given in Table 1.1 with their descriptions: Slotted propellant grain can produce a neutral thrust-time curve.

Symbol	Description
L	Length of the slotted propellant grain
l	Length of the slot
w	Web thickness
β	Web length
r	Diameter of the slot
D_{port}	Port diameter
D_{out}	Outer diameter of the grain
n	number of slots

Table 1.1. Parameters defining a slotted propellant grain

1.5 Anchor Grain

The double anchor grain is a grain propellant with two anchor spokes [36]. The inner part of this grain is defined by five independent variables, see Figure 1.6. Table 1.2, gives the description of these variables. This type of grain produces a regressive thrust-time curve.

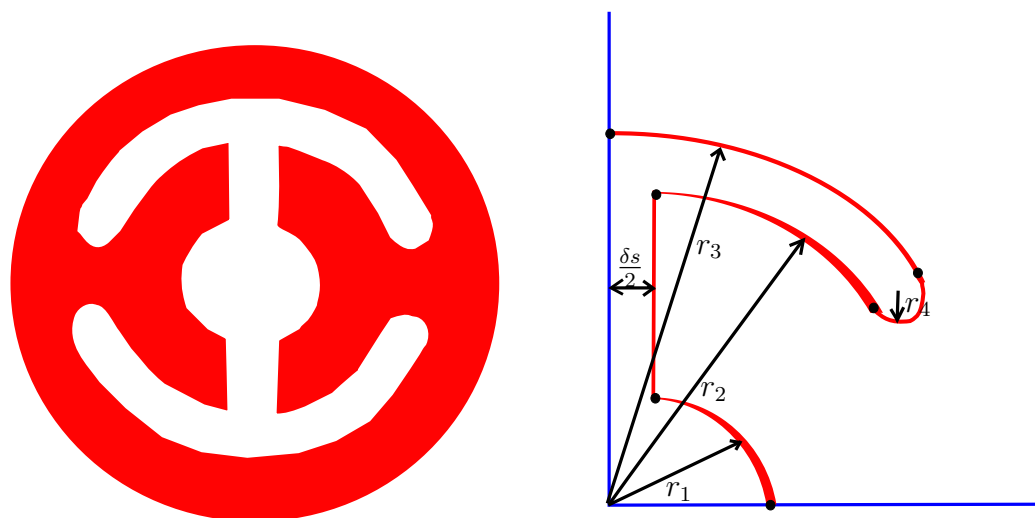


Figure 1.6. Anchor grain configuration

Symbol	Description
r_1	Radius of the inner circular region
r_2	Inner radius of the spoke
r_3	Outer radius of the spoke
r_4	Radius of the circular end of the spoke
δs	Width of the part joining the circular region and the spoke

Table 1.2. Parameters defining an anchor grain

1.6 Stereolithography (STL) Representation of the Surface

In order to simulate the operation of a SRM, the designer must supply the shape of the grain. The surface of the grain is represented by a triangular mesh, which is supplied as an STL file. This file consists of a list of face data. Each face is uniquely identified by a unit normal and three vertices. The normal and each vertex are specified by three coordinates each, so there is a total of 12 numbers for each facet.

The STL standard includes two data formats, ASCII and binary. Only the ASCII format that is described here.

```

solid name
{
  facet normal  $n_i$   $n_j$   $n_k$ 
    outer loop
      vertex  $v_1$   $v_1$   $v_1$ 
      vertex  $v_2$   $v_2$   $v_2$ 
      vertex  $v_3$   $v_3$   $v_3$ 
    endloop
  endfacet
}
endsolid name

```

Bold face print indicates a keyword; these must appear in lower case. Note that there is a space in “**facet normal**” and in “**outer loop**”, while there is no space in any of the keywords beginning with “**end**”. Indentation must be with spaces; tabs are not allowed. The notation $\{ \dots \}$ means that the contents of the brace brackets can be repeated one or more times. Words in italics are variables which are to be replaced with user-specified values. The numerical data in the **facet normal** and **vertex** coordinates are single precision floating point numbers. A **facet normal** coordinate may have a leading minus sign; a **vertex** coordinate may not. The STL surface representing a sphere is shown in Figure 1.7.

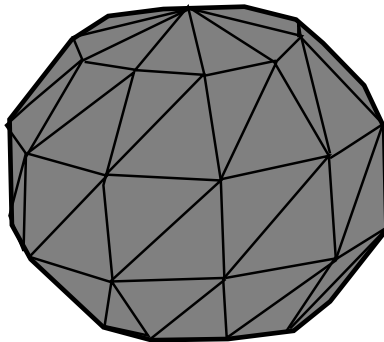


Figure 1.7. STL representation of a sphere

The facets define the surface of a 3-dimensional object. As such, each facet is part of the boundary between the interior and the exterior of the object. The orientation of the facets is specified in two ways which must be consistent. Firstly the direction of the normal is outward. Secondly, the vertices are listed in counter-clockwise order when looking at the object from the outside (right-hand rule).

1.6.1 Vertex to Vertex Rule

A correct STL file has to follow the vertex to vertex rule. The vertex to vertex rule says that each triangle in the STL file has to share two vertices with its adjacent triangles. In other words, a vertex of one triangle cannot lie on the edge of another triangle. Figure 1.8(a) shows the violation of the vertex to vertex rule and Figure 1.8(b) shows the corrected surface.

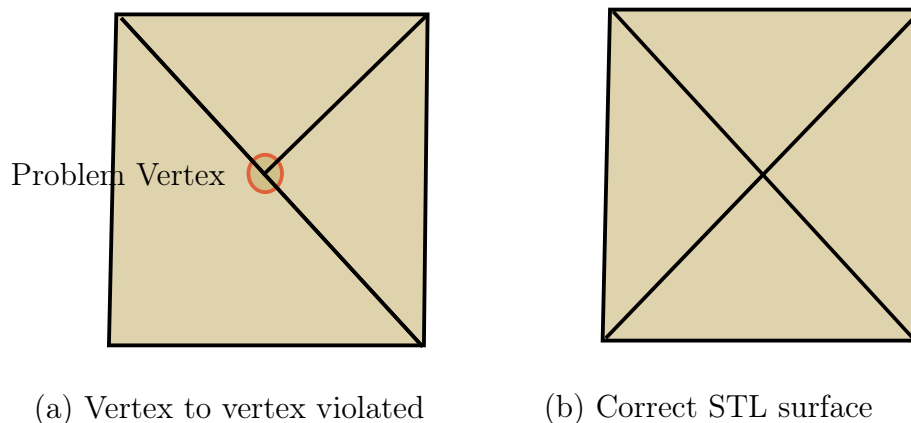


Figure 1.8. A violation of the vertex to vertex rule

It is because of the vertex to vertex rule that we know that a legal closed solid of genus = 0 will have $(3/2)E$ edges for each face (i.e each face has three edges, but each edge is shared by two faces). This gives us the three consistency rules against which to check:

Let F and E be the number of faces and edges respectively,

1. F must be even,
2. E must be a multiple of three,
3. $2 \times E$ must be equal $3 \times F$.

1.6.2 Common STL Errors

Although it is not explicitly specified in the STL data standard, all facets in a STL data file should construct one entity according to Euler's rule for legal closed solids:

$$F - E + V = 2B \quad (1.1)$$

Where F , E , V , and B are the number of faces, edges, vertices, and separate solid bodies respectively. If the relation does not hold, we say that the STL model is ‘leaky’.

The types of leaks commonly found are:

1. *Two facets crossed in the 3D space*

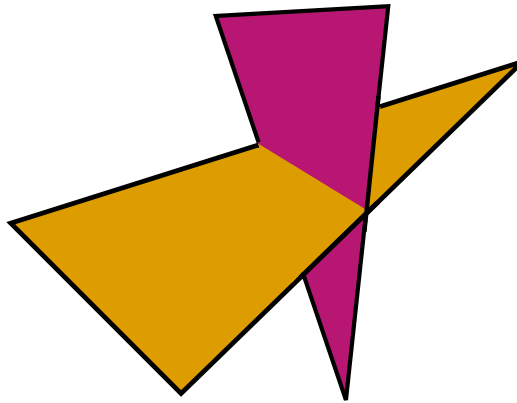


Figure 1.9. Crossed facets in 3D space.

This type of error shown in Figure 1.9 is very common for a low quality solid, created using boolean operators to generate an STL file [10].

2. *The triangulated edge of two surface patches does not match, therefore producing gaps between faces.*

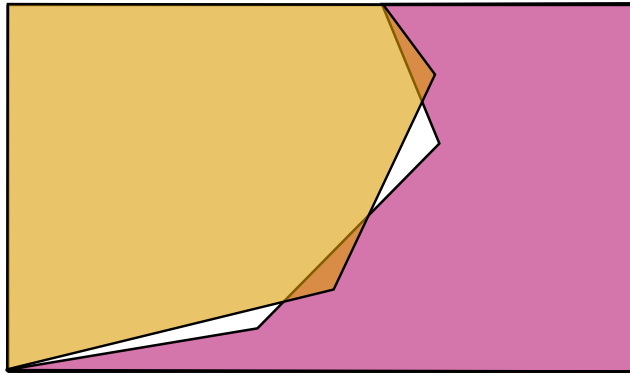


Figure 1.10. Mismatching surface patches

The error in Figure 1.10 is mostly caused by software bugs in the applications, or an ill-configured STL generation routine.

3. *Degenerated facets*

Degenerated facets are less critical errors in STL data. Unlike topological errors that are listed above, degenerated facets occur seldom and can cause serious build failure in the rapid prototyping process.

The types of degenerated facets include:

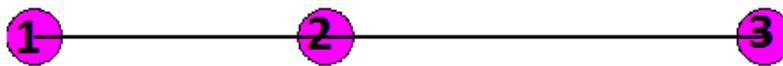


Figure 1.11. Co-linear vertices

The three vertexes of the facet are co-linear, or become co-linear when the previously non

co-linear coordinates are truncated by the algorithm of the importing routine.

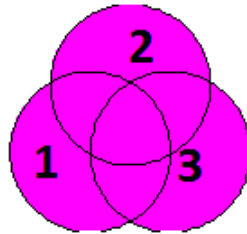


Figure 1.12. Vertices at the same point

The three vertices of the facet have the same coordinate or coincide. They coincide when the previously non coincidal coordinates are truncated by the algorithm of the importing routine. Although the problem of degenerate facets is not critical, it does not mean that they can be ignored. The facets data take up file space.

The solid bodies that are used in this project to represent the solid propellant grain are open solid bodies and they do not follow Euler's rule for representing a solid body but every triangle has to share at least two edges.

1.7 Thesis Layout

In chapter 2, ways of tracking moving interface are discussed. Three methods are discussed, namely:

- marker-point method,
- Volume of fluid technique,
- level set method.

The drawbacks of marker point and volume of fluid technique will be mentioned. Level set method is chosen to be used to model the surface propagation of the grain. This method is implemented in 2D and in 3D.

Chapter 3 gives the gas dynamics equations for simulating the internal flow inside the combustion chamber and the nozzle. The 1D internal ballistics model by Lamberty [21] is discussed, which is used together with the level set method in order to model the surface propagation of the grain.

In chapter 4, numerical results obtained using 2D or 3D model are shown. Some of these results are compared with the results that are obtained from experimental data.

In chapter 5, conclusions on the study are made and highlights of possible future work are mentioned.

Chapter 2

Burnback Modelling using Level Set Approach

2.1 Methods Used for the Evolution of Interfaces

Propagating interfaces are found in every area of science and engineering, from a burning flame to the diffusion of substance into another. Wherever there is a moving boundary separating two spaces, there is a propagating-interface problem.

In this chapter methods used to formulate equations for moving interfaces, will be discussed. These methods use different approaches to formulate these equations. Some use Lagrangian approach, i.e. this method divides the interface into small line segments using points and follow the movement of each point as the interface moves with time. Other methods use an Eulerian approach, i.e. these methods embed the interface on fixed grid and calculate the motion of the interface using the fixed grid points. This is the approach we will also use when we are tracking the interface of the solid propellant as it burns. We use the initial value formulation of the level set methods which is discussed in detail in section [2.2.2](#).

The level set method is a computational technique for tracking a propagating interface over time, which in many problems has proven more accurate in handling topological complexities such as corners and cusps. It is a robust scheme that is relatively easy to

implement. This method will be explained in detail in the following subsections using the signed distance function (SDF). This is the method that we use in modelling the burnback of the grain.

Another simple method for modelling a moving interface involves planting marker points along the surface and following their motion [24]. However, small initial errors quickly accumulate, and awkward subjective methods must be used to add or remove marker points as they get too far apart or too close together. This will be explained in detail in the following sections. And like many other simpler approaches to model propagating interfaces, the marker-point method fails in modelling some of the the more complex motions of the surface. Since this method fails to model the propagation of an interface that changes topology as it moves we will not use it to model the burnback.

A third method that is used to track moving interfaces, is the fractional marker volume method. The fractional marker volume method (sometimes called the volume of fluid method (VOF) [15, 16, 25], or simple line interface calculation, SLIC [26]) defines the surface by calculating the fractional volume of each material occupying in a computational cell [18]. These numbers range from zero (no material) to one (completely filled with that material). The interface occurs in the cells with fractional volumes. This method has its own drawbacks and we will not use it to track the interface of the burning surface of the solid grain. These drawbacks will be clear when we discuss these methods in detail in the following subsections.

2.2 Equations of Motion for Moving Interfaces

We shall present here the partial differential equations of moving interfaces. One approach leads to boundary value partial differential equation for evolving interface, the other leads to time dependant initial value problem [24].

Consider a boundary, either a curve in two dimensions or a surface in three dimensions, separating one region from another. Imagine that this curve or surface moves in the direction normal to itself (where the normal direction is oriented with respect to an inside and

outside) with a known speed function V , see Figure 2.1.

The goal is to track the motion of this interface as it evolves. We are only concerned with the motion of this interface in its normal direction throughout, and we shall ignore motions of the interface in its tangential directions.

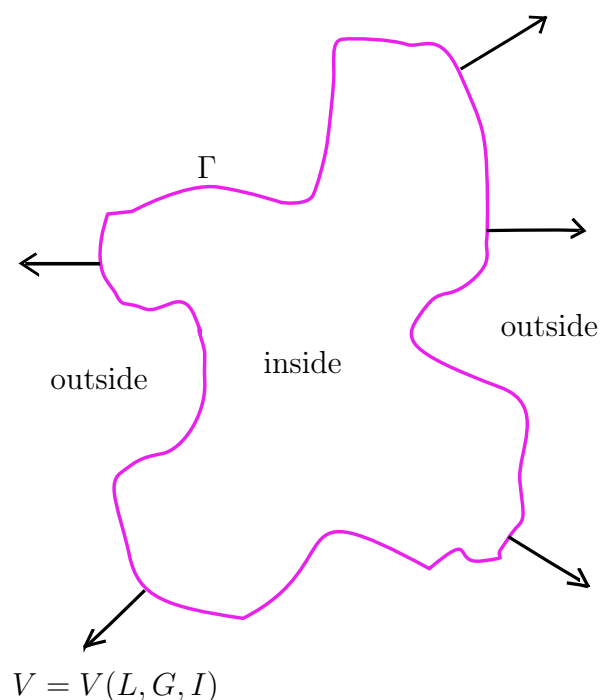


Figure 2.1. Curve (Γ) propagating with speed V in normal direction.

The speed function V , which may depend on many factors, can be written as

$$V = V(L, G, I), \quad (2.1)$$

where L, G, I represent the Local properties, Global properties and Independent properties respectively. Local properties are those determined by the local geometric information, such as curvature of the interface and normal direction. Global properties are those prop-

erties that depend on the shape and position of the interface. For example, the speed might depend on the integrals along the front (moving interface) and/or associated differential equations. As a particular case, if the interface is a source of heat that affects diffusion on either side of the interface, and a jump in the diffusion in turn influences the motion of the interface, then this would be characterized as a global property [32]. Independent properties are those that are independent of the shape of the interface, such as the underlying fluid velocity that passively transports the interface.

Now assuming that we are given the speed V and the position of the interface, the objective is to track the evolution of the interface. Our first task is to formulate this evolution problem in an Eulerian approach, that is the one in which the underlying coordinate system remains fixed.

2.2.1 A Boundary Value Formulation

We shall assume that $V > 0$ throughout the entire motion of the interface, hence the interface always moves outwards. One way to characterize the position of this expanding interface is to compute the arrival time $T(x, y)$ of the interface as it crosses each point (x, y) as shown in Figure 2.2.

The equation for this arrival time $T(x, y)$ is easily derived. In one dimension, using the fact that *distance = velocity \times time* (see Figure 2.3), we have that

$$V \frac{dT}{dx} = 1. \quad (2.2)$$

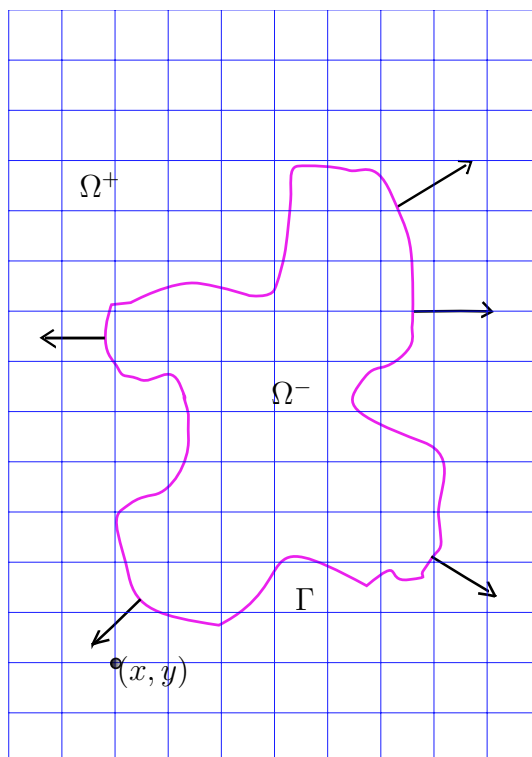


Figure 2.2. Calculation of crossing time at (x, y) for expanding interface with $V > 0$. Γ is the interface separating the outside region (Ω^+) from the inside region (Ω^-).

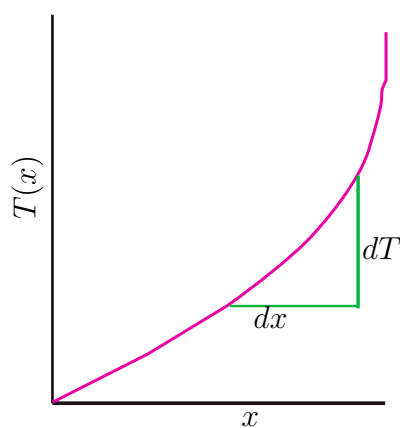


Figure 2.3. Setup for boundary value formulation.

In multiple dimensions, ∇T is orthogonal to the level sets of T , and similar to the one

dimensional case, its magnitude is indirectly proportional to the speed V [2], Therefore we have

$$|\nabla T|V = 1. \quad (2.3)$$

Hence the interface motion is characterized as the solution to a boundary value problem. If the speed V depends only on position, then the equation reduces to what is known as the ‘‘Eikonal’’ equation [33]. As an example, the moving surface $T(x, y)$ for a circular interface expanding with unit speed $V = 1$ is shown in Figure 2.4.

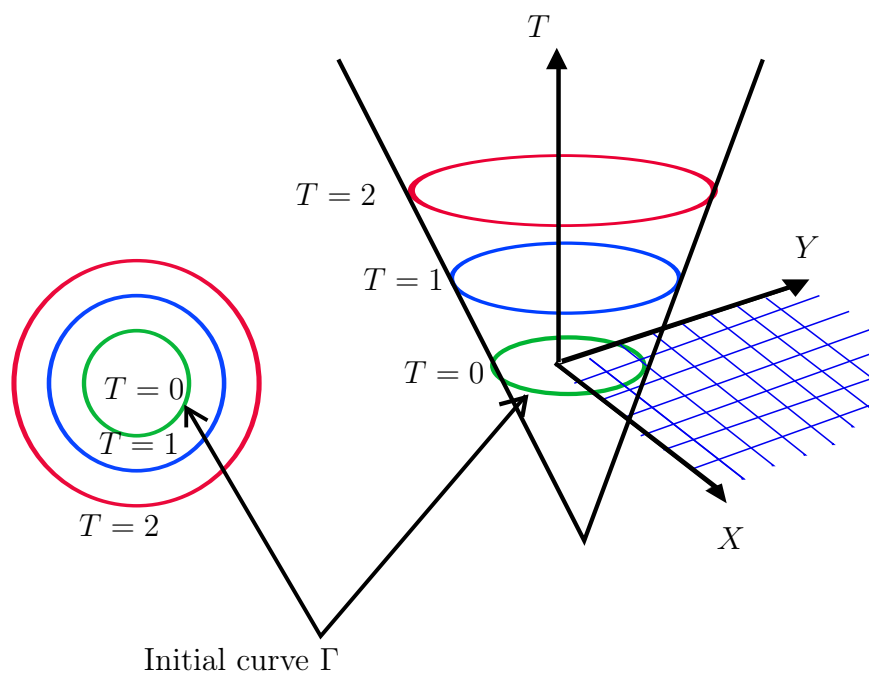


Figure 2.4. Transformation of an interface motion into boundary value problem.

This boundary value method will not be implemented in modelling the propagation of the burning surface of the grain, since it requires only a positive velocity field and each grid point cannot be revisited to find $T(x, y)$ because it is determined once for each grid point. Since the burning surface of the grain has a changing topology as a function of time, $T(x, y)$ is changing and therefore the boundary value method cannot be implemented.

2.2.2 The Initial Value Formulation

Another approach of applying the level set method, is the initial value formulation that allows a function $T(x, y)$ which may change, and a point can be passed several times.

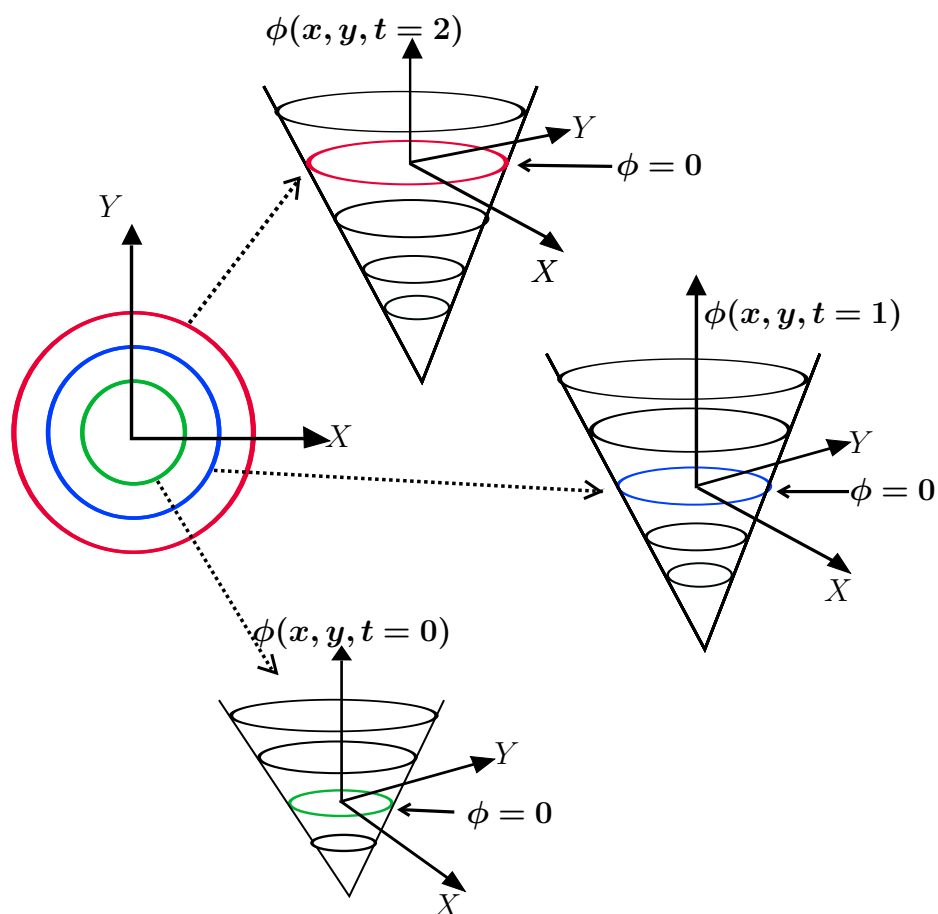


Figure 2.5. Transformation of an interface motion into initial value problem.

Suppose that the interface moves with speed V , and V depends on independent properties such as the pressure of the flow inside the combustion chamber. Therefore the rate at which the interface moves is different in every position in the chamber because we have different burning rates in different positions in the chamber. The difference in burning rate is caused by the changing pressure and flow conditions inside the chamber. For complex

grain geometries, we may pass over a point (x, y) more than once, hence crossing time $T(x, y)$ is not a single valued function. The way of taking care of this is to embed the initial position of the interface as the zero level set of a higher dimensional function ϕ . We can link the evolution of this function ϕ to the propagation of the interface itself through a time-dependant initial value problem. Then, at any time t , the interface is given by the zero level set of the time-dependant level set function ϕ (see Figure 2.5). The function ϕ has the following properties:

$$\begin{aligned}\phi(\mathbf{x}, t) &> 0 \quad \forall \mathbf{x} \in \Omega^+, \\ \phi(\mathbf{x}, t) &< 0 \quad \forall \mathbf{x} \in \Omega^-, \\ \phi(\mathbf{x}, t) &= 0 \quad \forall \mathbf{x} \in \partial\Omega = \Gamma,\end{aligned}$$

where $\mathbf{x} = [x_1, \dots, x_n] \in \mathcal{R}^n$.

In order to derive an equation of the motion for this level set function ϕ and match the zero level set of ϕ with the evolving interface, we first require that the level set value of a particle on the interface with path $x(t)$ must always be zero, hence

$$\phi(\mathbf{x}(t), t) = 0. \quad (2.4)$$

Then by the chain rule of differentiation, we have,

$$\phi_t + \nabla\phi(\mathbf{x}(t), t) \cdot \mathbf{x}'(t) = 0. \quad (2.5)$$

Since V supplies the speed in the outward normal direction, then $\mathbf{x}'(t) \cdot \mathbf{n} = V$, where $\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}$. This yields an evolution equation for ϕ , namely

$$\phi_t + V|\nabla\phi| = 0. \quad (2.6)$$

This equation requires an initial condition $\phi(\mathbf{x}, 0)$.

This is a level set equation given by Osher and Sethian [28]. Equation (2.6) describes

the time evolution of the level set function ϕ in such a way that the zero level set of this evolving function is always identified with a propagating interface; see Figure 2.5.

We can summarize the two approaches for modelling moving interface by letting Γ be a curve in the plane propagating in a direction normal to itself with speed V . $\Gamma(t)$ is the set of points of \mathbf{x} that gives the position of the interface at any time t . Table 2.1 gives the summary of the two methods.

Boundary Value Formulation	Initial Value Formulation
$ \nabla T V = 1$ Interface = $\Gamma(t) = \{\mathbf{x} T(\mathbf{x}) = t\}$ Requires $V > 0$	$\phi_t + V \nabla\phi = 0$ Interface = $\Gamma(t) = \{\mathbf{x} \phi(\mathbf{x}, t) = 0\}$ Applies for an arbitrary V

Table 2.1. Summary of boundary value and initial value formulation of the level set method

2.3 Advantages of Using Level Set Methods

There are certain advantages of using level set methods compared to the marker points method and volume of fluid technique which will be discussed in detail later in this chapter.

- Level set methods are easily applied to higher dimensions.
- Topological changes in the evolving interface Γ are handled naturally. This is shown in detail in section 2.3.1
- The initial value formulation of level set method for moving interfaces can also be approximated using computational schemes [31, 13, 12].

For example schemes may be developed by using a discrete grid in (x, y) domain and substituting finite difference approximations for spatial derivatives [1]. As an illustration, using a uniform mesh of spacing h , with grid node (i, j) , and employing

the standard notation that ϕ_{ij}^n is the approximation of the solution at x_i, y_j , at time step n , one might write equation (2.6) as follows

$$\frac{\phi_{ij}^{n+1} - \phi_{ij}^n}{\Delta t} + V|\nabla_{ij}\phi_{ij}^n| = 0, \quad (2.7)$$

where Δt is the time step. Here a forward difference scheme in time is used and $|\nabla_{ij}\phi_{ij}^n|$ represents some appropriate finite difference operator for the spatial derivative. Therefore, an explicit finite difference approach is possible.

- Using level set methods, geometric properties of the interface are determined. For example at any point of the interface, the normal vector is given by,

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}. \quad (2.8)$$

2.3.1 Topological Changes

The position of the interface at time t is given by a zero level set, i.e $\phi(x, y, t) = 0$ for the evolving function ϕ . This need not to be a single curve, and it can break or merge as t advances. The level set function ϕ remains a single-valued function.

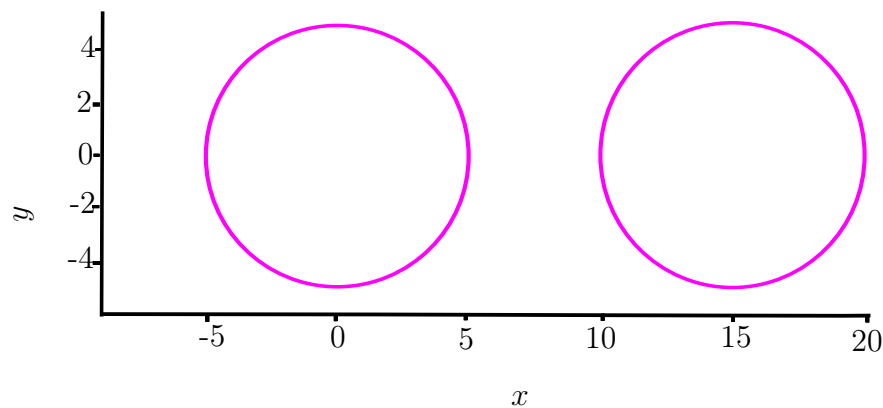


Figure 2.6. Interface made up of two circles that are evolving with a speed V .

To explain this change in topology consider a surface that is made up of two circles that are evolving with the speed V (assuming that $V > 0$) in normal direction and the interface

of the surface propagates outwards, then if these circles are positioned close together at some time t they will come into contact and form one interface.

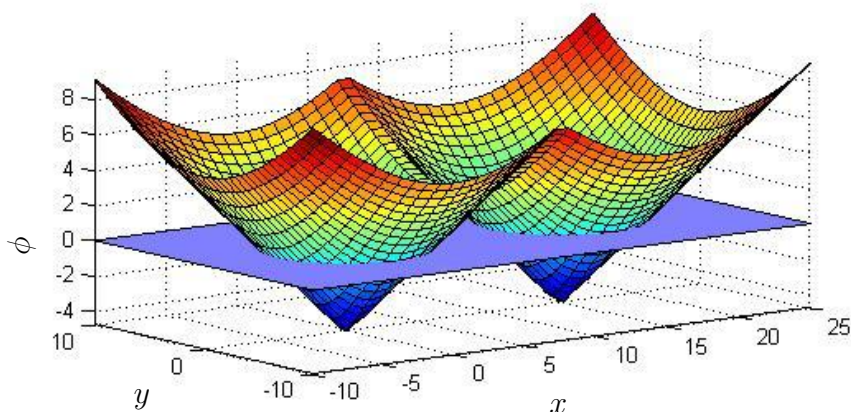


Figure 2.7. ϕ at $t = 0$ for the two circles.

Figure 2.7 shows $\phi(x, y, 0)$ for the surface given by the two circles. The plane plotted in blue represent the values where $\phi(x, y, 0) = 0$, and this gives us a surface that yields the approximation of the original interfaces see Figure 2.8.

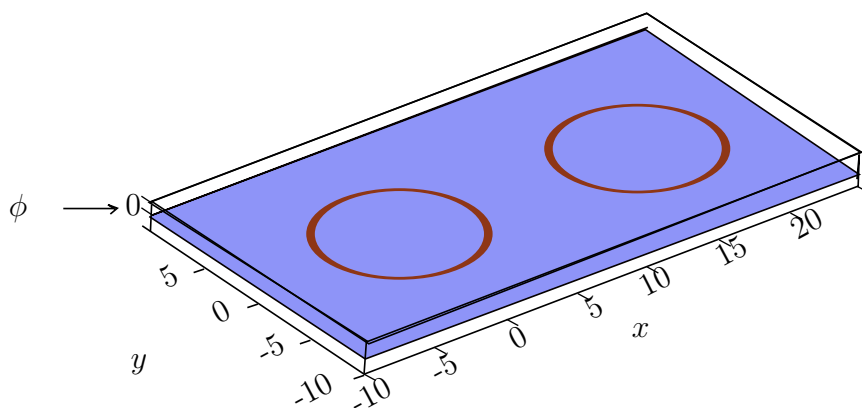


Figure 2.8. The approximation of the original surface given by the values of $\phi = 0$.

Using the surface approximation in Figure 2.8 given by $\phi(x, y, 0) = 0$ and propagating it using the velocity V and equation (2.6), the two circles after some time t will merge and become one closed interface (see Figure 2.9)

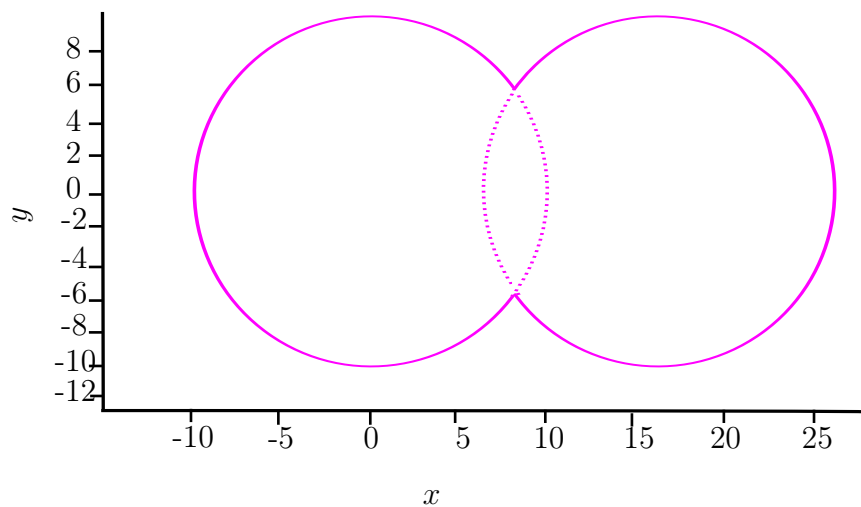


Figure 2.9. The surface made up of two circles merging to form one interface.

The interface in Figure 2.9 can be found by solving $\phi(x, y, t) = 0$ see Figure 2.10 and Figure 2.11.

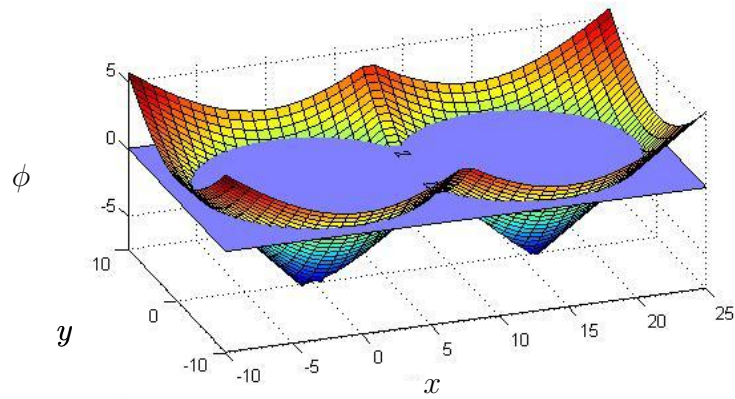


Figure 2.10. $\phi(x, y, t)$ for the two circles merging to form one interface.

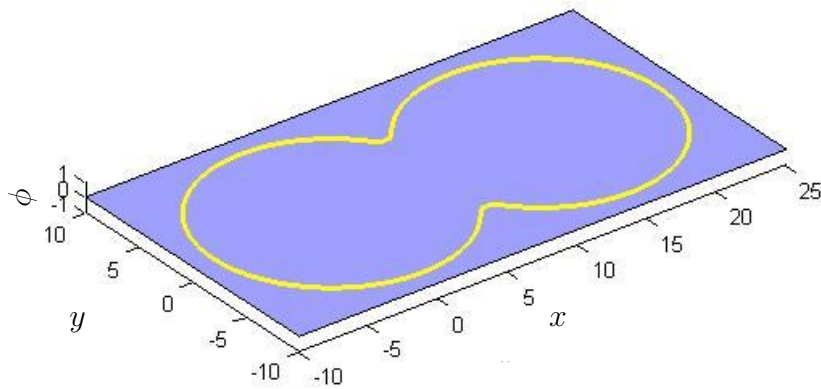


Figure 2.11. The contour on the plane $\phi = 0$ approximating the surface time t .

The contour found on the plane (see Figure 2.11) will give the approximation of the surface in Figure 2.9.

2.4 Theory of Curves and Surface Evolution

In order to simulate the burnback, a method for tracking the interface as it evolves is required. In the previous section, we have mentioned that initial value formulation of the level set method is used to model burnback. In this section, alternative methods for tracking moving interfaces are discussed and the reasons why they not implemented in order to model the grain burnback are also mentioned.

In this section we shall discuss the marker point and the fractional volume method for moving interfaces which are theoretical methods that model a moving interface. These methods formulate the equation of the motion of a propagative curve using both Eulerian and Lagrangian formulations [32, 30]. These methods are not stable and singularities in the curvature can develop as the interface evolves. These singularities are often analogous to shocks formation in the solution.

2.4.1 The Lagrangian Formulation of a Moving Interface

Let Γ be a simple, smooth, closed initial curve in \mathcal{R}^2 , and let $\Gamma(t)$ be the one parameter family of curves generated by moving Γ along its normal vector field with speed V , where V is a given scalar function. This means that $\mathbf{n} \cdot \mathbf{x}_t = V$, where \mathbf{x} is the position vector of the curve, t is time, and \mathbf{n} is the unit normal vector to the curve.

This approach considers a parametrized form of equations. In the discussion, we will assume that the speed function V is constant [1, 30]. Let the position vector $\mathbf{x}(s, t)$ parametrize Γ at time t , where $0 < s < S$, and assume periodicity $\mathbf{x}(0, t) = \mathbf{x}(S, t)$. The curve is parametrized so that the interior is on the left in the direction of increasing s (see Figure 2.12). If we let $\mathbf{n}(s, t)$ be the parametrization of the outward normal. The equation of motion of the curve can then be written in terms of individual components $\mathbf{x} = (x, y)$ as

$$x_t = V \left(\frac{y_s}{(x_s^2 + y_s^2)^{\frac{1}{2}}} \right), \quad y_t = V \left(\frac{x_s}{(x_s^2 + y_s^2)^{\frac{1}{2}}} \right). \quad (2.9)$$

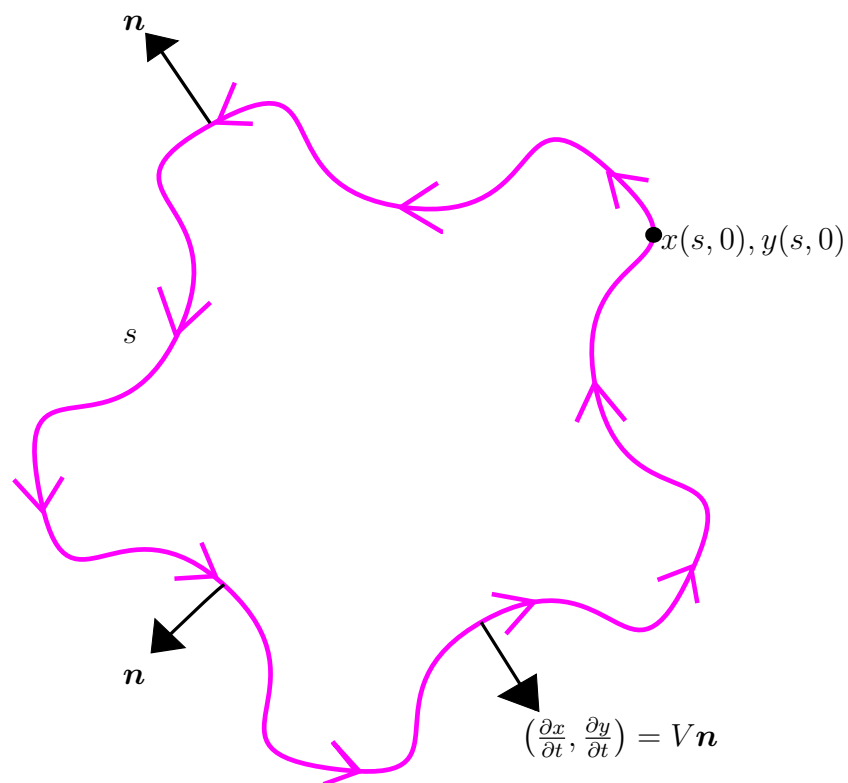


Figure 2.12. Parametrized view of a propagating curve

The normal is given by

$$\mathbf{n} = \frac{(y_s, -x_s)}{(x_s^2 + y_s^2)^{\frac{1}{2}}}.$$

Here $(x(s, t), y(s, t))$ describes a moving interface.

2.4.2 The Marker Point Methods for the Evolution of Interfaces

This is a standard approach to modelling moving interfaces which comes from discretizing the Lagrangian form of equations of motion given in equation (2.9). In this approach, the parametrization is discretized into a set of marker particles whose positions at any time are used to reconstruct the interface. This technique is known under many different names that include marker particle techniques, string methods, and nodal methods. In two dimensions, the interface is reconstructed as line segments, and in three dimensions triangles

are often used.

This approach can be illustrated through a straightforward scheme that constructs a simple difference approximation to the Lagrangian equations of motion. The parametrization interval $[0, S]$ is then divided into M equal intervals of size Δs , yielding $M + 1$ mesh points, $s_i = i\Delta s$, where $i = 0, \dots, M$, as shown in Figure 2.13. The time is also divided into equal intervals of length Δt . The image point (point that moves with the moving interface) of each mesh point $i\Delta s$ at each time step $n\Delta t$ is a marker point (x_i^n, y_i^n) on the moving interface. The goal that we need to achieve, is to find a numerical scheme that will produce new values (x_i^{n+1}, y_i^{n+1}) from the previous positions by following the method given by Sethian [30] and the ones given by Hyman [19].

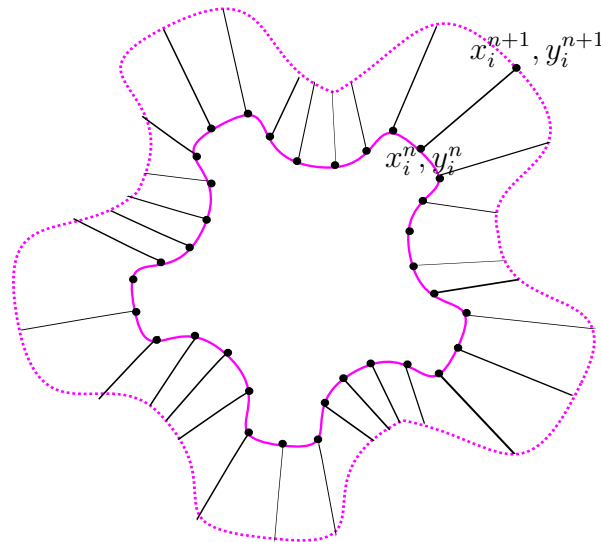


Figure 2.13. The discrete parametrization of a propagating curve

First, we approximate parameter derivatives at each marker point by using neighbouring mesh points. Central difference approximations based on Taylor series expansions give

$$\frac{dx_i^n}{ds} \approx \frac{x_{i+1}^n - x_{i-1}^n}{2\Delta s}, \quad \frac{dy_i^n}{ds} \approx \frac{y_{i+1}^n - y_{i-1}^n}{2\Delta s}, \quad (2.10)$$

$$\frac{d^2x_i^n}{ds^2} \approx \frac{x_{i+1}^n - 2x_i^n + x_{i-1}^n}{\Delta s^2}, \quad \frac{d^2y_i^n}{ds^2} \approx \frac{y_{i+1}^n - 2y_i^n + y_{i-1}^n}{\Delta s^2}. \quad (2.11)$$

Similarly, time derivatives may be replaced by the forward difference approximations

$$\frac{dx_i^n}{dt} \approx \frac{x_i^{n+1} - x_i^n}{\Delta t}, \quad \frac{dy_i^n}{dt} \approx \frac{y_i^{n+1} - y_i^n}{\Delta t}. \quad (2.12)$$

Substituting these approximations into equation of motion of the interface given in equation (2.9) produces the following scheme

$$(x^{n+1}, y^{n+1}) = (x_i^n, y_i^n) + \Delta t V \frac{(y_{i+1}^n - y_{i-1}^n, -(x_{i+1}^n - x_{i-1}^n))}{((x_{i+1}^n - x_{i-1}^n)^2 + (y_{i+1}^n - y_{i-1}^n)^2)^{\frac{1}{2}}}. \quad (2.13)$$

This method finds it difficult to deal with cusp and corners. For example one way to keep track of a few points $(p(x, y))$ is to advance them in the normal direction to the interface (arrows in Figure 2.14) and determine where there interface ends up at time t (see Figure 2.14).

The marker point method have some serious drawbacks. For a V-shaped interface (or a cusp) propagating with the speed V , some of the marker points are absorbed and they need to be removed (see Figure 2.15)

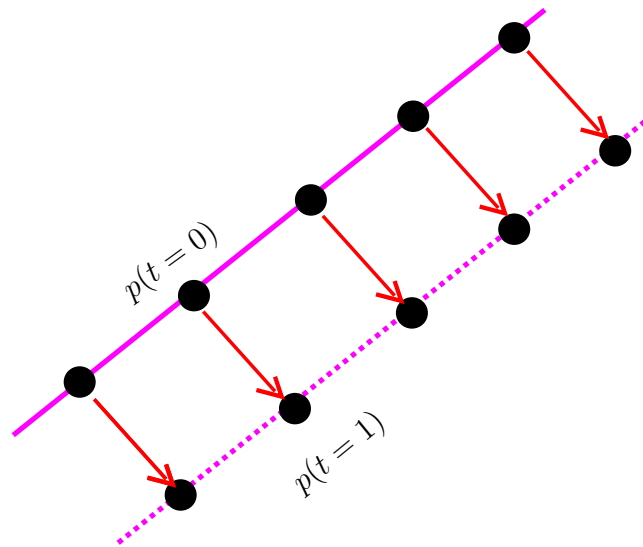


Figure 2.14. The evolution of set of points on the interface at time $t = 0$ and at time $t = 1$.

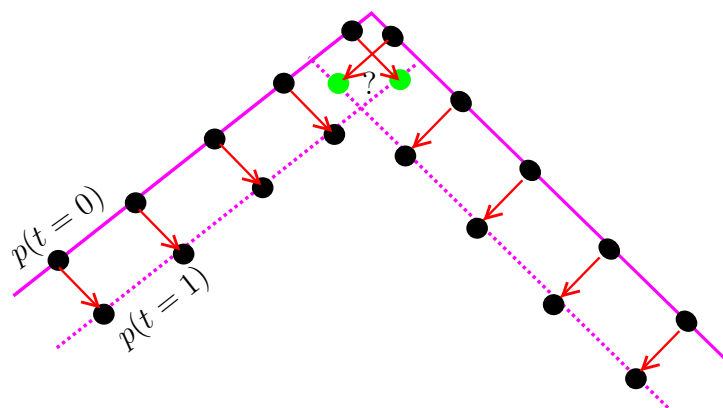


Figure 2.15. The evolution of a cusp of an interface at time $t = 0$ and at time $t = 1$.

Also, if the interface has a corner that is expanding, the initial set of points may not be sufficient to define the evolving interface (see Figure 2.16). Therefore some points will need

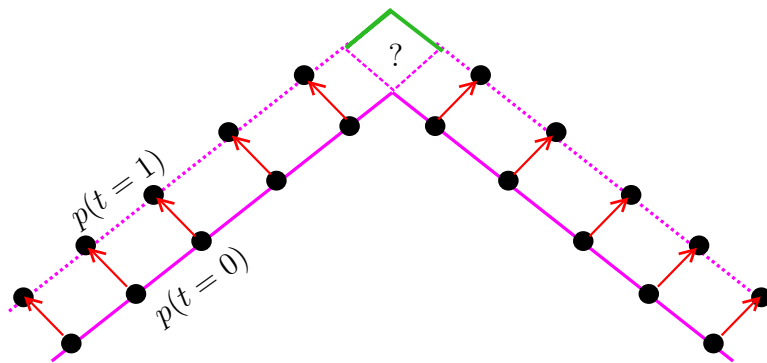


Figure 2.16. The evolution of a corner of an interface at time $t = 0$ and at time $t = 1$.

to be added. The distance between the points should be kept small enough in order to obtain a smooth curve. This method is very troublesome during implementation because during topology change points will need to be added or removed depending on whether the interface merges or breaks. This complicates the scheme.

2.4.3 Volume of Fluid Technique

A different approach to model moving interface is provided by volume of fluid technique, that was introduced by Noh and Woodward [26], and is based on an Eulerian view. This method has been introduced under many different names such as the cell method and method of partial fractions.

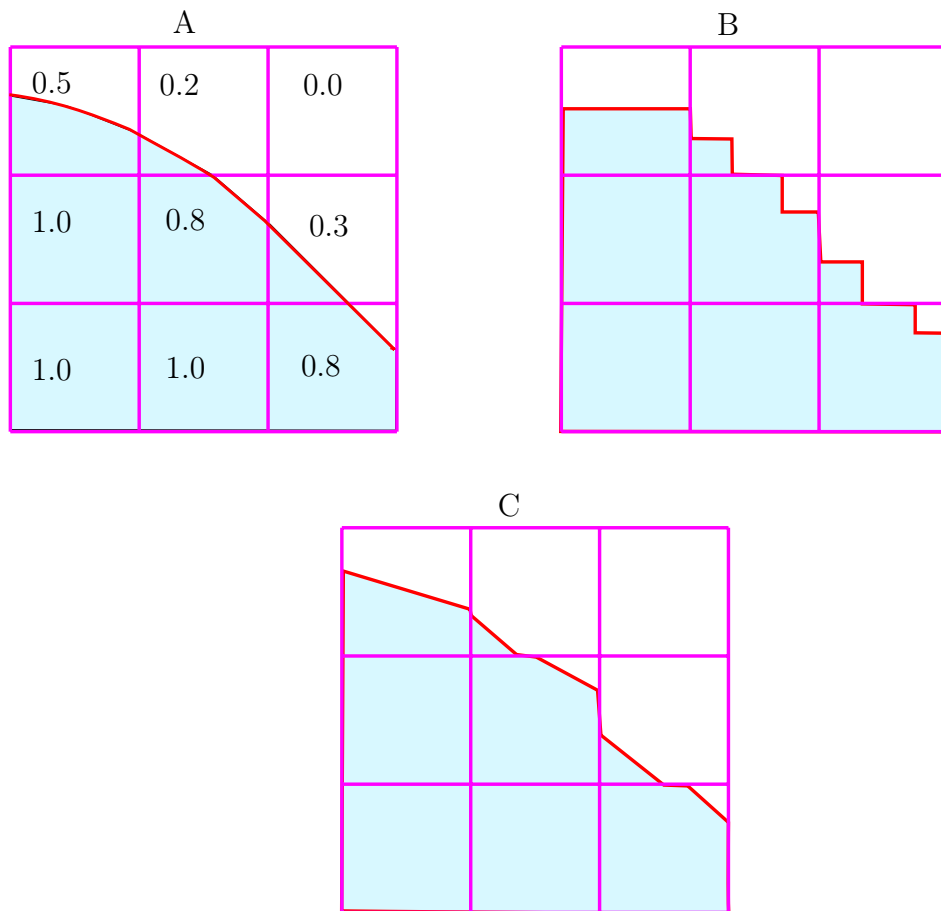


Figure 2.17. A-original surface separating two regions and associated volume fractions. B& C-Reconstructed interfaces using rectangular and piecewise linear fractional volume methods

The idea is as follows (see Figure 2.17A): assuming that we have a fixed grid on the computational domain, and assign values to each grid cell based on the fraction of that cell containing material inside the interface. If we assume that we are given a closed curve, then a value of one is assigned to those cells that are completely inside the curve (or interface), and a cell value of zero is assigned to those cells that are completely outside the interface, and a fraction between zero and one is assigned to those cells that the interface crosses, based on the area of the cell that is inside the interface.

The idea then relies solely on these cell fractions shown in Figure 2.17A to characterize the interface location. Approximation techniques are used to reconstruct the interface

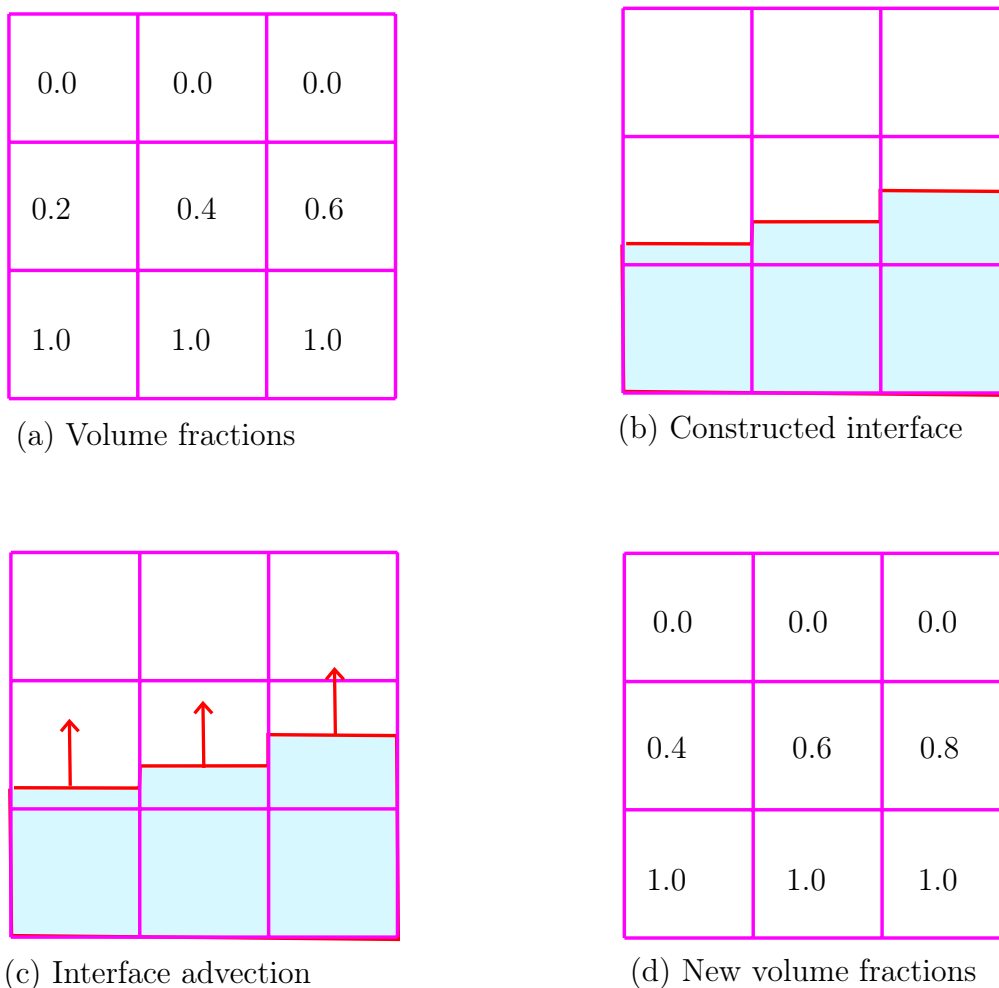


Figure 2.18. Interface evolution using volume of fluid technique

from these cell fractions. The original idea of Noh and Woodward was known as Simple Line Interface Calculation (SLIC), since the interface is reconstructed using vertical and horizontal lines (see Figure 2.17B).

In order to evolve the interface, the cell fractions on the fixed grid are updated to reflect the progress of the interface as it moves. Suppose that one wishes to move the interface under velocity V (here V can be a speed which is a transport term that is not necessarily

in the normal direction of the interface). Noh and Woodward provide a methodology in which the value in each cell is updated under the velocity in each coordinate direction by locally reconstructing the interface and then changing the material in the neighbouring cells under this motion. After completing coordinate sweeps, one has produced new cell fractions at the next time step corresponding to the updated moving interface. In Figure 2.18, we show the motion of the interface with the simple vertical velocity field V .

Since the development of the volume of fluid technique, many reconstruction development techniques have been developed to include slope [3, 16]. Then rectangles and triangles were used during the reconstruction of the interface (see Figure 2.17C)

However, we are not going to use this technique in modelling the propagation of the grain due to the following drawbacks:

- This technique is inaccurate, since the approximation of the interface is through volume fractions which are relatively rough (or crude) and a large number of cells are often required to obtain reasonable results.
- Evolution under complex speed functions is always problematic. The results depend on the underlying orientation of the grid. Therefore these problems become worse in the presence of directional velocity fields.
- Calculation of intrinsic geometric properties of the interface, such as curvature and normal direction may be inaccurate.
- Much work may be required to develop a similar algorithm for higher dimensions.

2.5 General Method of Finding Shortest Distance Function

The initial value formulation of the level set method in section 2.2.2, uses the signed distance function ϕ to represent the moving interface. The values of ϕ are found by calculating the shortest distance between the grid point and the interface. In this section we present the general method used to find the minimum distance between a grid point and the interface.

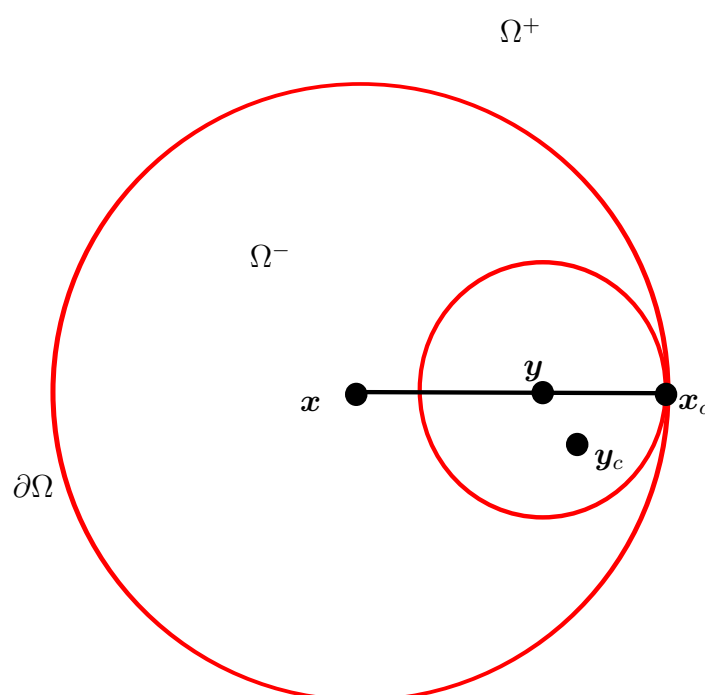


Figure 2.19. x_c as the shortest distance to both x and y .

Let Ω be a region with boundary $\partial\Omega$, which is a partition between the region inside and the region outside the interface. Therefore a **distance function** $d(\mathbf{x})$ is defined as

$$d(\mathbf{x}) = \min(\|\mathbf{x} - \mathbf{x}_I\|), \quad \forall \mathbf{x}_I \in \partial\Omega, \quad (2.14)$$

implying that $d(\mathbf{x}) = 0$ on the boundary where $\mathbf{x} \in \partial\Omega$ [27]. Geometrically d may be

constructed as follows. If $\mathbf{x} \in \partial\Omega$, then $d(\mathbf{x}) = 0$, otherwise, for a given \mathbf{x} we have to find the point on the boundary set $\partial\Omega$ closest to \mathbf{x} and we label this point \mathbf{x}_c . Therefore we have

$$d(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_c\|, \quad (2.15)$$

as the distance between \mathbf{x} and \mathbf{x}_c . For any given point \mathbf{x} , suppose that \mathbf{x}_c is the point on the interface closest \mathbf{x} . Then for every point \mathbf{y} on the line segment connecting \mathbf{x} and \mathbf{x}_c , \mathbf{x}_c is the point on the interface closest to \mathbf{y} as well. To see this clearly consider the graph shown in Figure 2.19, where \mathbf{x} , \mathbf{x}_c , and an example of \mathbf{y} are shown. Since \mathbf{x}_c is the closest interface point to \mathbf{x} , no other interface points can be inside the large circle drawn about \mathbf{x} passing through \mathbf{x}_c . Since the small circle lies inside the larger circle, no interface points can be inside the smaller circle, and thus \mathbf{x}_c is the interface point closest to \mathbf{y} . The line segment from \mathbf{x} to \mathbf{x}_c is the shortest path from \mathbf{x} to the interface. Any local deviation from this line segment increases the distance from the interface to \mathbf{x} . In other words, the path from \mathbf{x} to \mathbf{x}_c is the path of steepest descent for the function d . Evaluating $-\nabla d$ at any point on the line segment from \mathbf{x} to \mathbf{x}_c gives a vector that points from \mathbf{x} to \mathbf{x}_c . Furthermore, since d is Euclidean distance,

$$|\nabla d| = 1. \quad (2.16)$$

Proof of equation (2.16)

Let $f(\mathbf{x})$ be a function in \mathcal{R}^3 that defines a surface that separates two regions. Let $\mathbf{x}(x, y, z)$ be a grid point, and $\mathbf{x}_c(x_c, y_c, z_c)$ be a point on the surface defined by f . Then the distance between these two points is given by,

$$d = \sqrt{(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2}.$$

Therefore, ∇d is given by,

$$\nabla d = \frac{x - x_c}{\sqrt{(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2}} \mathbf{i} + \frac{y - y_c}{\sqrt{(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2}} \mathbf{j} + \frac{z - z_c}{\sqrt{(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2}} \mathbf{k}.$$

Hence,

$$|\nabla d| = \frac{(x - x_c)^2}{(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2} + \frac{(y - y_c)^2}{(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2} + \frac{(z - z_c)^2}{(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2} = 1.$$

The above argument leading to equation 2.16 is true for any \mathbf{x} as long as there is a unique closest point \mathbf{x}_c . That is, equation 2.16 is true except at points that are equidistant from (at least) two distinct points on the interface. Unfortunately, these equidistant points can exist, making equation 2.16 only partially true. It is also important to point out that equation 2.16 is generally approximating the gradient numerically. One of the triumphs of the level set method involves the ease with which these degenerate points are treated numerically.

We will use initial value formulation of the Level set methods (explained in section 2.2.2) to describe the propagation of the burning surface using a signed distance function. A **signed distance function** (SDF) is an implicit function ϕ with $|\phi(\mathbf{x})| = d(\mathbf{x})$ for all \mathbf{x} . Therefore, $\phi(\mathbf{x}) = d(\mathbf{x}) = 0$ for all $\mathbf{x} \in \partial\Omega$, $\phi(\mathbf{x}) = -d(\mathbf{x})$ for all $\mathbf{x} \in \Omega^-$ (inside the interface), and $\phi(\mathbf{x}) = d(\mathbf{x})$ for all $\mathbf{x} \in \Omega^+$ (outside the interface). Signed distance functions also possess the property,

$$|\nabla\phi| = 1, \tag{2.17}$$

for all the points that are not equidistant from two points on the interface.

The method presented here is not feasible in calculating the shortest distance between the grid point and the interface because some grid points are closer to the line segments between the nodes of the interface. This is shown in Figure 2.20.

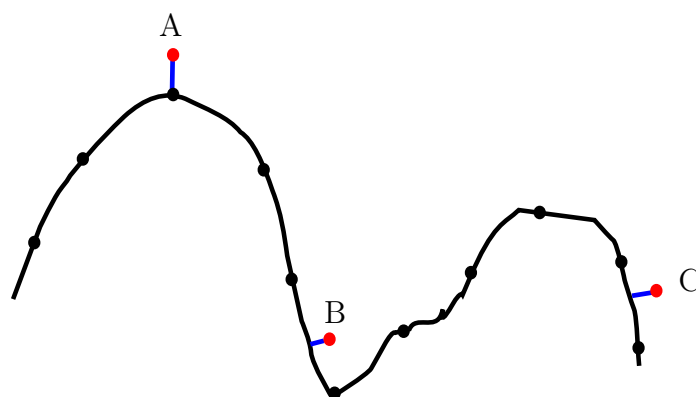


Figure 2.20. Grid points (red) and their shortest distances shown by a blue line

For example the shortest distance of point A in Figure 2.20 to the interface, is to the node of the interface. This is not true for point B and C because their shortest distances to the interface are to the line segments between nodes of the interface.

2.6 Analytical Calculation of the Burning Surface Area and Port Area

In this section the burning surface area and port area for tubular and star grain geometries are calculated analytically. The equations to find the burning surface area and port area for tubular grain geometry are derived by the author and for the star grain geometry equations given by Hartfield et al. [14] are used.

2.6.1 Equations to Find Burning Surface Area and Port Area of a Tubular Grain Analytically

Tubular grain has the initial burning cylinder known as the mandrel of the burning surface. If we have r_0 as the radius of the mandrel, the port area and the burning surface area are given by (also see Figure 4.3),

$$A_s = 2\pi l(r_0 + f), \quad (2.18)$$

$$A_a = \pi(r_0 + f)^2, \quad (2.19)$$

where A_s , A_a , l are burning surface area, port area and length of the grain respectively.

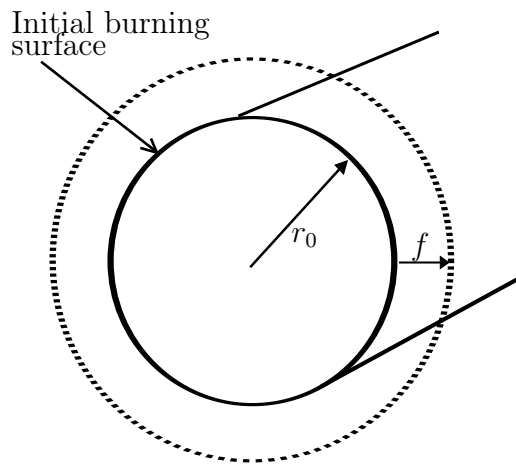


Figure 2.21. Tubular grain evolving with a distance function f

The function f is the burn away distance which gives the position of the interface as the surface evolves.

2.6.2 Equations to Find Burning Surface Area and Port Area of a Star Grain Analytically

In this subsection equations defining the burning surface area and port area of a star grain geometry are given. These equations are discussed in detail in [14]. Grain geometries are generally described using lengths and the angles defined in the cross-section. Figure 2.22 shows a sample of a star grain burnback and its geometric definition on the right. The burning surface area and port area are determined as follows:

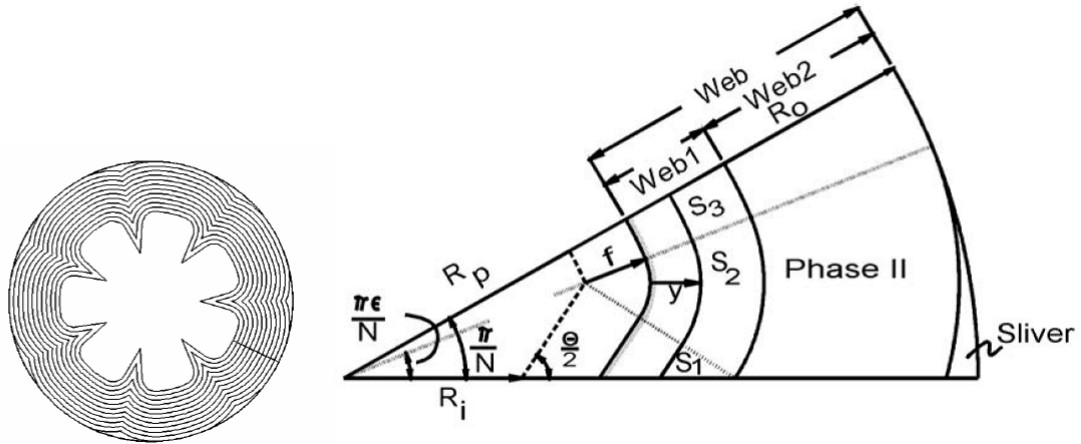


Figure 2.22. Star Geometry and its defining parameters (figure taken directly from [14]).

Before reaching the casing

The geometric relationship between the two primary angles in the geometry definition can be written as:

$$\frac{\theta}{2} = \tan^{-1} \left\{ \frac{R_p \sin\left(\frac{\pi\epsilon}{N}\right) \tan\left(\frac{\pi\epsilon}{N}\right)}{R_p \sin\left(\frac{\pi\epsilon}{N}\right) - R_i \tan\left(\frac{\pi\epsilon}{N}\right)} \right\}, \quad (2.20)$$

where N is the number of spokes. Therefore, the burning surface area is given by the two arcs and straight line multiplied the length of the grain. The burning perimeter is given by,

$$S = 2N(S_1 + S_2 + S_3) \quad (2.21)$$

$$S = 2N \left\{ \frac{R_p \sin\left(\frac{\pi\epsilon}{N}\right)}{\sin\left(\frac{\theta}{2}\right)} - (y + f) \cot\left(\frac{\theta}{2}\right) + (y + f) \left(\frac{\pi}{2} - \frac{\theta}{2} + \frac{\pi\epsilon}{N}\right) + (R_p + y + f) \left(\frac{\pi}{N} - \frac{\pi\epsilon}{N}\right) \right\},$$

hence $A_s = S \times l$.

The port area is given by the area under the two arcs plus the area of the triangle. Math-

ematically it is given by,

$$A_a = 2N \left\{ \begin{array}{l} \frac{1}{2} R_p \sin \left(\frac{\pi \epsilon}{N} \right) \left[R_p \cos \left(\frac{\pi \epsilon}{N} \right) + R_p \sin \left(\frac{\pi \epsilon}{N} \right) \tan \left(\frac{\theta}{2} \right) \right] \\ - \frac{1}{2} \left(\frac{R_p \sin \left(\frac{\pi \epsilon}{N} \right)}{\sin \left(\frac{\theta}{2} \right)} - (y + f) \cot \left(\frac{\theta}{2} \right) \right)^2 \tan \left(\frac{\theta}{2} \right) \\ + \frac{1}{2} (y + f)^2 \left(\frac{\pi}{2} - \frac{\theta}{2} + \frac{\pi \epsilon}{N} \right) + \frac{1}{2} (R_p + y + f)^2 \left(\frac{\pi}{N} - \frac{\pi \epsilon}{N} \right) \end{array} \right\} \quad (2.22)$$

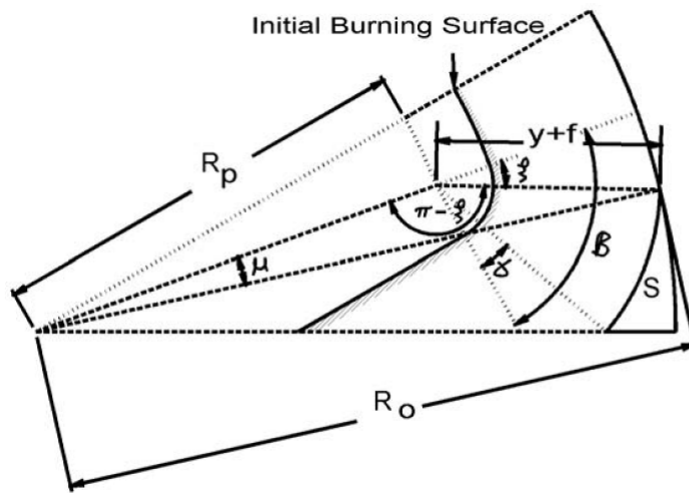


Figure 2.23. Star defining parameters when the burning front hits the casing (figure taken directly from [14]).

When burning surface hits the casing

The angles β and γ in Figure 2.23 are used in the development of the surface as it evolves and can be represented as:

$$\beta = \left(\frac{\pi}{2} - \frac{\theta}{2} + \frac{\pi \epsilon}{N} \right), \quad (2.23)$$

$$\gamma = \tan^{-1} \left(\frac{\sqrt{(y + f)^2 - R_p \sin \left(\frac{\pi \epsilon}{N} \right)^2}}{R_p \sin \left(\frac{\pi \epsilon}{N} \right)} \right) - \frac{\theta}{2}. \quad (2.24)$$

Using the law of cosines, ξ can be expressed as follows:

$$\xi = \pi - \cos^{-1} \left(-\frac{R_0^2 - R_p^2 - (y + f)^2}{2R_p(y + f)} \right). \quad (2.25)$$

The burn perimeter is given by a section of the arc as,

$$S = 2N[(y + f)(\beta - \gamma - \xi)]. \quad (2.26)$$

To find the port area, the angle μ is calculated using the law of sine's as:

$$\mu = \sin^{-1} \left(\frac{y + f}{R_0} \sin(\pi - \xi) \right). \quad (2.27)$$

Hence the port area is given by,

$$A_a = N \left\{ \begin{array}{l} R_0^2 \left(\frac{\pi}{N} (1 - \epsilon) + \mu \right) + (y + f)^2 (\beta - \gamma - \xi) \\ + R_p \sin \left(\frac{\pi \epsilon}{N} \right) \left[R_p \cos \left(\frac{\pi \epsilon}{N} \right) + \sqrt{(y + f)^2 - R_p \sin \left(\frac{\pi \epsilon}{N} \right)^2} \right] \\ - R_p \sin \mu (R_p \cos \mu + \sqrt{(y + f)^2 - R_p \sin(\mu)^2}) \end{array} \right\} \quad (2.28)$$

2.7 Calculation of the Signed Distance Function in 2D

We find the shortest distance using the projection geometry in 2D (explained in detail in Appendix A.1) in the following way:

If we have a grid point \mathbf{x} (see Figure 2.24), a point on the interface that is closest to \mathbf{x} is found, in this scenario \mathbf{x}_j is that point. Then we look for two neighbouring points of \mathbf{x}_j , which are given by \mathbf{x}_{j-1} and \mathbf{x}_{j+1} . Three vectors are determined as follows,

$$\begin{aligned}\mathbf{q} &= \mathbf{x} - \mathbf{x}_j, \\ \mathbf{a} &= \mathbf{x}_{j-1} - \mathbf{x}_j, \\ \mathbf{b} &= \mathbf{x}_{j+1} - \mathbf{x}_j.\end{aligned}$$

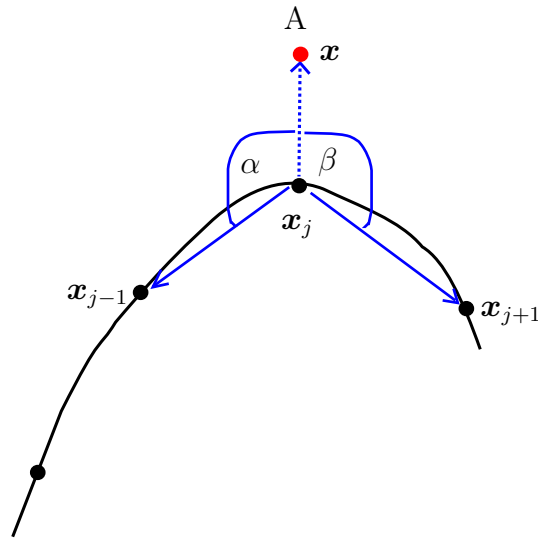


Figure 2.24. Node \mathbf{x}_j as the closest point to the grid point \mathbf{x} labelled A

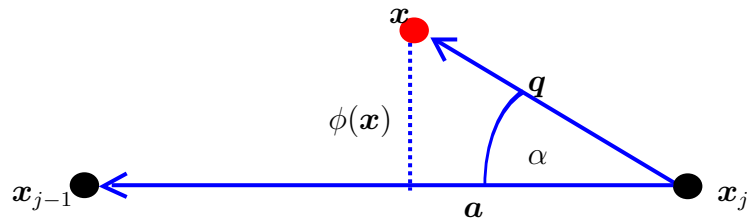
After we found these vectors, we continue to calculate the dot product between vector \mathbf{q} and the vectors on the interface, given by

$$A = \cos \alpha = \hat{\mathbf{a}}^T \mathbf{q}, \quad B = \cos \beta = \hat{\mathbf{b}}^T \mathbf{q},$$

where $\hat{\mathbf{a}} = \frac{\mathbf{a}}{\|\mathbf{a}\|}$, and $\hat{\mathbf{b}} = \frac{\mathbf{b}}{\|\mathbf{b}\|}$.

Case 1: If we have $A < 0$, and $B < 0$, then the distance between the grid point and the interface is to the nodal point of the interface, and is given by

$$\phi(\mathbf{x}) = \|\mathbf{q}\|$$

Figure 2.25. Shortest distance $\phi(\mathbf{x})$ obtained by projection

Case 2: If $A > 0$, $B < 0$, the distance between the grid point \mathbf{x} and the interface is to line segment between \mathbf{x}_{j-1} and \mathbf{x}_j (see Figure 2.25). The line segment is defined by vector \mathbf{a} . Therefore the distance is given by,

$$\phi(\mathbf{x}) = \|\mathbf{q} - A\hat{\mathbf{a}}\|.$$

If $A < 0$, $B > 0$, then

$$\phi(\mathbf{x}) = \|\mathbf{q} - B\hat{\mathbf{b}}\|,$$

which is the distance from the grid point \mathbf{x} to the line segment between the nodal points \mathbf{x}_{j+1} and \mathbf{x}_j . Vector \mathbf{b} defines this line segment.

Case 3: If $A > 0$, $B > 0$, we project \mathbf{x} onto the line segments defined by \mathbf{a} and \mathbf{b} . Then the distance between the grid point and the interface is given by the shortest distance between the two projections. Mathematically, this is given by

$$\phi(\mathbf{x}) = \min(\|\mathbf{q} - A\hat{\mathbf{a}}\|, \|\mathbf{q} - B\hat{\mathbf{b}}\|).$$

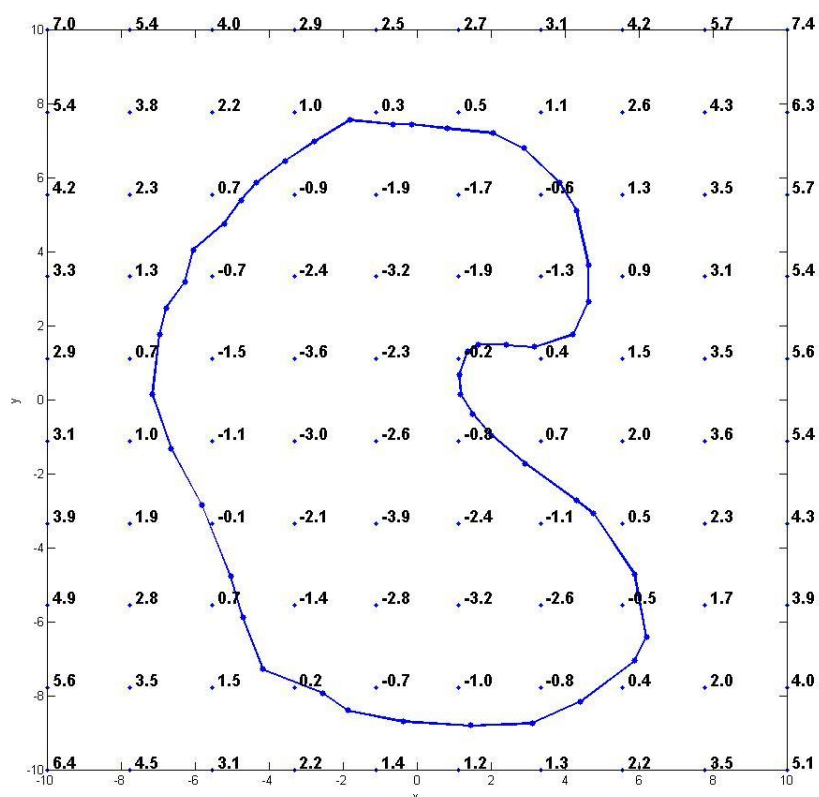


Figure 2.26. Original interface and the distances of the grid points from the interface

To obtain the sign of the distance, we use a function `inpolygon` in MATLAB which decides whether the points are inside the interface (polygon) and we assign a negative sign to all the distances of the grid points that are inside the polygon.

Figure 2.26 shows an interface with its nodal points and a few grid points and their distances from the interface. The distance is negative if the point is inside the interface, and is zero if the grid point is on the interface, and is positive if the grid point is on the outside of the interface.

In order to find the approximation of this interface using the ϕ values shown in Figure 2.26, we need to find the contour at zero level. This contour is given by,

$$\phi(x, y) = 0. \quad (2.29)$$

Figure 2.27 shows the contour obtained at zero level which give the approximation of the original interface (shown in red), and original interface (shown in blue). The red interface in Figure 2.27 is the one used to evolve the interface using equation (2.6).

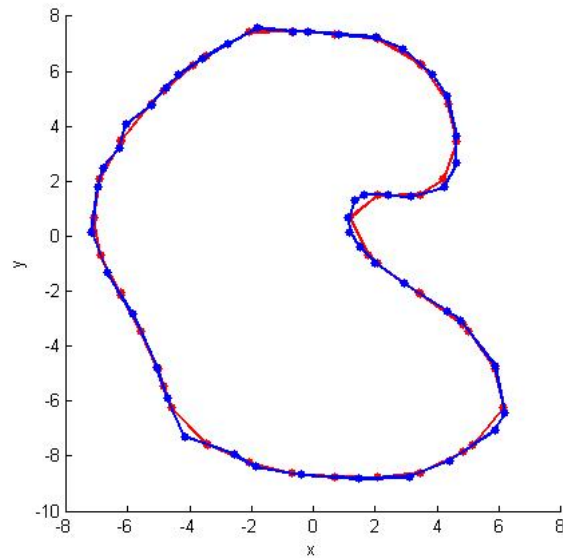


Figure 2.27. Original interface (blue) and its approximation (red).

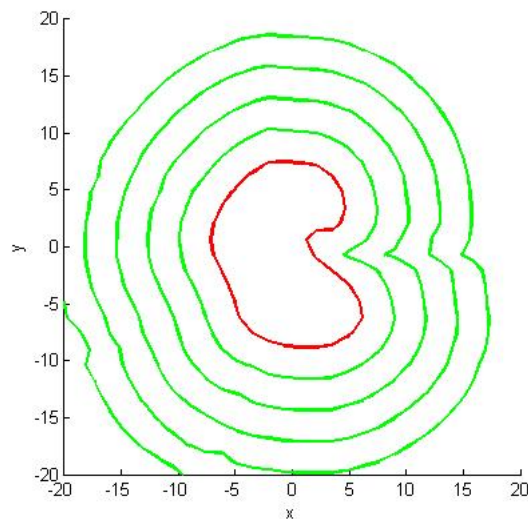


Figure 2.28. Evolving interface

If we find the contours of other levels starting from level one to the next, shows how the interface will evolve with time under constant velocity. This is shown in Figure 2.28

2.8 A Simple Method to Calculate the Signed Distance Function in 3D

In this section a simple method used to find the signed distance function (SDF), ϕ , in 3D is presented. A triangular mesh is a type of polygonal mesh in computer graphics which comprises of triangles that are connected by their common edges to form a surface.

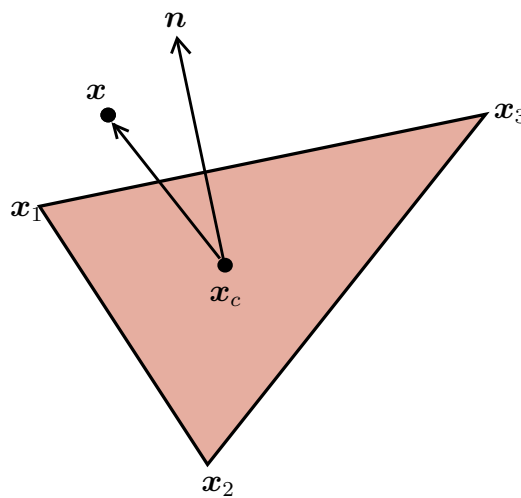


Figure 2.29. \mathbf{x}_c as the centre of the closet triangle to the grid point \mathbf{x} .

A surface that is given by a triangular mesh where the coordinates of the vertices of triangles are given, the shortest distance from the surface to the grid point can be determined in the following manner.

Firstly, we have to evaluate the centre (\mathbf{x}_c) of the triangle that is closest to the grid point by taking the mean of all the coordinates of the vertices that form the triangle (see Figure 2.29). Then the shortest distance is given by,

$$\phi(\mathbf{x}) = |\mathbf{x} - \mathbf{x}_c|, \quad (2.30)$$

where $\mathbf{x}_c = \frac{\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3}{3}$.

In order to obtain the sign that will be assigned to the distance, we use the fact that all normals are pointing out of the surface, this means that the orientation of the facets of the triangle is very important. Therefore, $[\mathbf{x} - \mathbf{x}_c] \cdot \mathbf{n} > 0$ if the point \mathbf{x} is outside the surface, and $\phi(\mathbf{x})$ is chosen positive, otherwise if $[\mathbf{x} - \mathbf{x}_c] \cdot \mathbf{n} < 0$, then point \mathbf{x} is inside

the interface and $\phi(\mathbf{x})$ will have a negative distance.

When the interface is given by triangular mesh, some points are closer to the edge of the triangle, some are closer to the plane of the triangle, some are closer to the vertices of triangle than to the center of triangle. Therefore, a better method for calculating distance between the surface and the grid points is required.

2.9 Calculation of the Signed Distance Function in 3D Using Projection Geometry

In the previous section we presented a simple method for calculating distances in 3D, and we have mentioned that some points are closer to other parts of the triangle other than the centre. Therefore, taking this into account projection geometry in linear algebra (shown in Appendix A.2) is used to calculate the distances between the grid points and the interface.

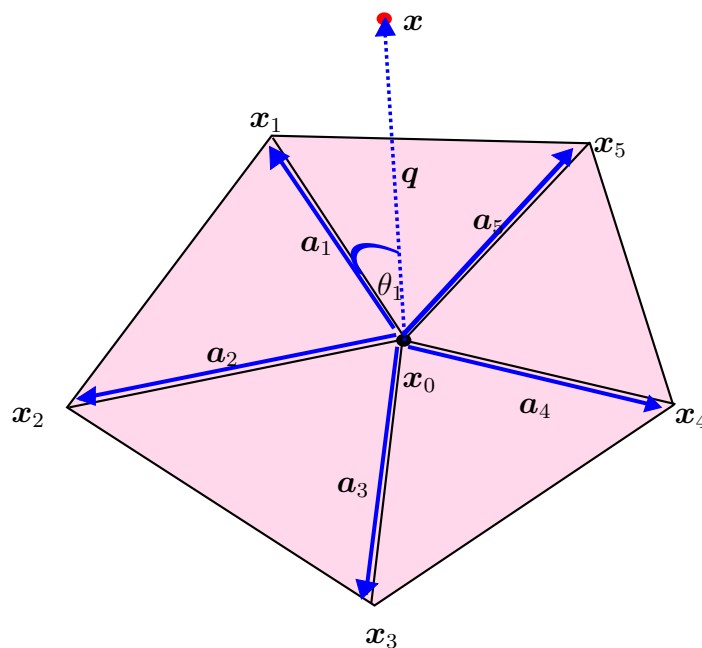


Figure 2.30. Point \mathbf{x} , its closest point \mathbf{x}_0 , and its neighbours $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_5$.

If we have a grid point \mathbf{x} , we look for the vertex on the surface (which is given by a triangular mesh) that is closest to \mathbf{x} . Call this point \mathbf{x}_j , so that $\|\mathbf{x} - \mathbf{x}_j\|$ is the minimum distance between the grid point \mathbf{x} and the surface. Since \mathbf{x}_j is the point that is a vertex for more than one triangle, we then look for all the triangles that have \mathbf{x}_j as a vertex.

Therefore all the triangles that have \mathbf{x}_j as a vertex will have the neighbours of \mathbf{x}_j as other vertices. We denote these neighbours of \mathbf{x}_j by $\mathbf{x}_1, \dots, \mathbf{x}_p$.

In Figure 2.30, we shall illustrate the procedure with $p = 5$ and \mathbf{x}_j is denoted by \mathbf{x}_0 . Denote the vectors between these points by,

$$\begin{aligned}\mathbf{q} &= \mathbf{x} - \mathbf{x}_0 \\ \mathbf{a}_1 &= \mathbf{x}_1 - \mathbf{x}_0 \\ &\vdots \\ \mathbf{a}_5 &= \mathbf{x}_5 - \mathbf{x}_0,\end{aligned}$$

which are shown in Figure 2.30. The sign of the dot product between the vector \mathbf{q} and the vectors on the plane of the surface are then calculated as follows,

$$c_k = \cos \theta_k = \hat{\mathbf{a}}_k^T \mathbf{q} \quad \text{for } k = 1, \dots, 5,$$

where $\hat{\mathbf{a}}_k = \frac{\mathbf{a}_k}{\|\mathbf{a}_k\|}$. Note that $c_k \hat{\mathbf{a}}_k$ is the projection of vector \mathbf{q} onto the line define by vector \mathbf{a}_k (see Figure 2.31).

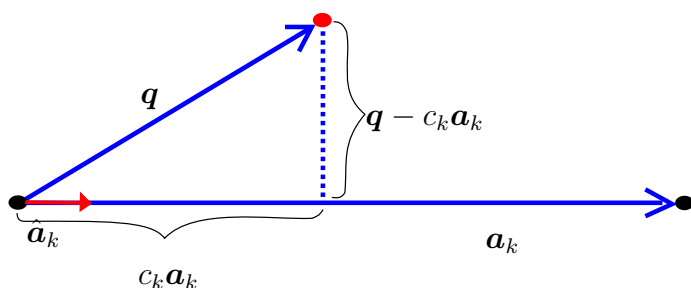


Figure 2.31. A projection of vector \mathbf{q} onto vector \mathbf{a}_k

Case 1: If $c_k < 0$ for $k = 1, \dots, p$, then the shortest distance of point \mathbf{x} to the surface is to the vertex \mathbf{x}_0 , and is given by

$$\phi(\mathbf{x}) = \|\mathbf{q}\|.$$

Case 2: If we have only one angle that is an acute angle, then it means that the shortest distance to the surface is to the edge of the triangle. For example, if we have $c_1 > 0$ and $c_k < 0$ for $k = 2, \dots, p$, then the shortest distance between the grid point and the surface

is given by,

$$\phi(\mathbf{x}) = \|\mathbf{q} - c_1 \hat{\mathbf{a}}_1\|.$$

Case 3: If we have more than one angle that is acute, we first project to all the edges with positive dot product in the following way; assuming that $c_1, c_2, c_3 > 0$, the shortest distance to the edge is

$$\phi_1(\mathbf{x}) = \min(\|\mathbf{q} - c_1 \hat{\mathbf{a}}_1\|, \|\mathbf{q} - c_2 \hat{\mathbf{a}}_2\|, \|\mathbf{q} - c_3 \hat{\mathbf{a}}_3\|). \quad (2.31)$$

We also project to all the triangles that have both angles positive. In the scenario where we have $c_1, c_2, c_3 > 0$, the shortest distance to the triangle is given by,

$$\phi_2(\mathbf{x}) = \min(\|\mathbf{q} - \mathbf{p}_1\|, \|\mathbf{q} - \mathbf{p}_2\|), \quad (2.32)$$

where $\mathbf{p}_1 = P_1 \hat{\mathbf{x}}_1$, $\mathbf{p}_2 = P_2 \hat{\mathbf{x}}_2$. P_1, P_2 are projection matrices defined in Appendix A.2.

Note that P_1 has the basis vectors \mathbf{a}_1 and \mathbf{a}_2 , and P_2 has the basis vectors \mathbf{a}_2 and \mathbf{a}_3 . In order to ensure that the projection is inside the triangle $0 < \tilde{x}_1 + \tilde{x}_2 < 1$, and $\tilde{x}_1, \tilde{x}_2 > 0$ for $\tilde{x}_1, \tilde{x}_2 \in \hat{\mathbf{x}}_1$. Hence the shortest distance to the surface is,

$$\phi(\mathbf{x}) = \min(\phi_1(\mathbf{x}), \phi_2(\mathbf{x})). \quad (2.33)$$

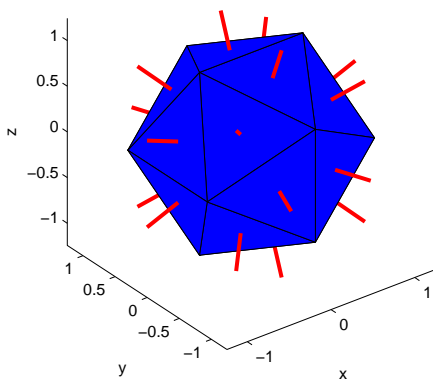


Figure 2.32. An Icosahedron and its normals pointing out of the surface (red line segments)

In order to obtain the sign to the distance, we use the fact that all the normals of the triangles are pointing out of the surface, then we set $\phi(\mathbf{x})$ equal to $-\phi(\mathbf{x})$ if $\mathbf{n} \cdot \mathbf{q} < 0$. This means that point \mathbf{x} is inside the surface. If $\mathbf{n} \cdot \mathbf{q} > 0$, it means that point \mathbf{x} is outside the surface and $\phi(\mathbf{x})$ remains positive.

Figure 2.32 shows an Icosahedron which is given as an STL solid body with its normals plotted at the centre of the triangles pointing out of the surface. The distance function $\phi(x, y, z)$ is calculated for the grid points and the surface is obtained by solving,

$$\phi(x, y, z) = 0. \quad (2.34)$$

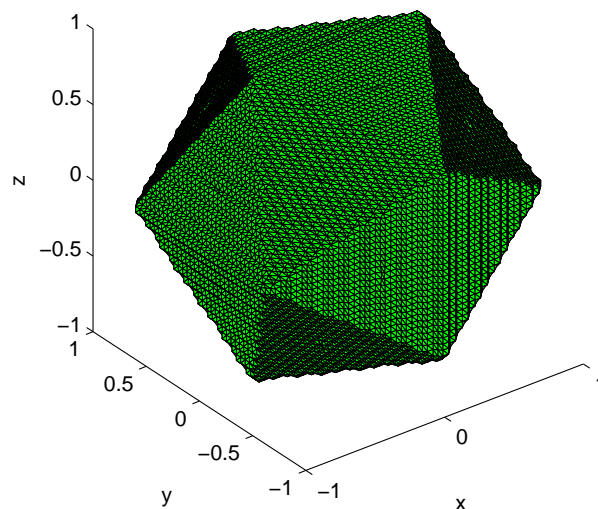


Figure 2.33. Reconstructed surface using `isosurface` that approximates the original Icosahedron

Using `isosurface` in MATLAB, the reconstructed surface is shown in Figure 2.33.

2.10 2D Modelling of the Burnback in MATLAB

The surface is given as an STL file which consists of a planar triangular mesh (see Figure 2.34).

The burnback of the grain is modelled using the inner part of the grain called the mandrel [29]. As an example of this, a star grain is shown in Figure 2.35(a). In order to calculate the internal ballistics explained in chapter 3, the grain is divided into cells (see Figure 2.35(b)). The SDF ϕ , is then calculated in order to find the approximation of the mandrel which will be taken as the initial condition for surface evolution.

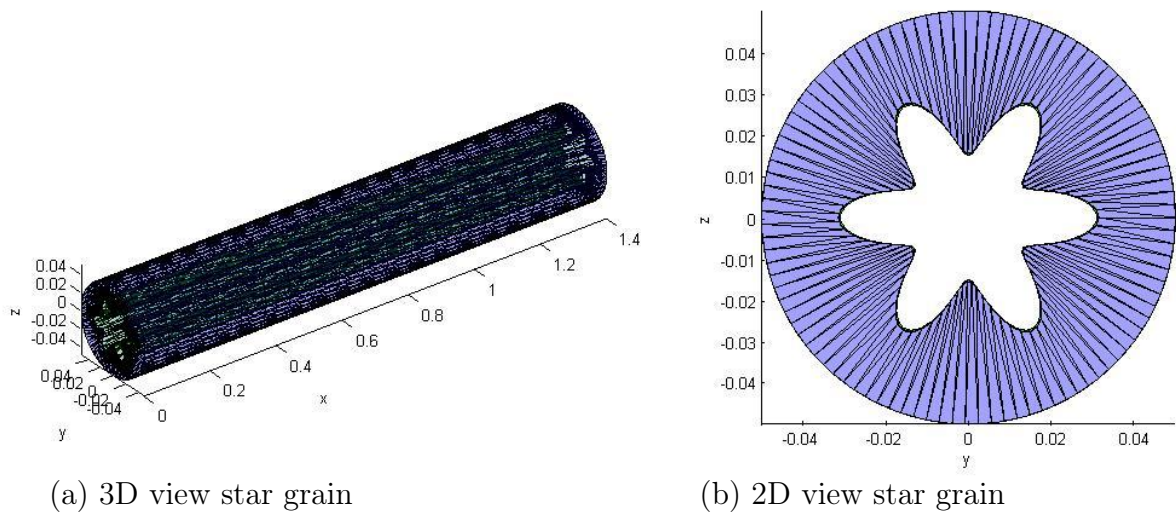


Figure 2.34. Star grain made up of planar triangular mesh

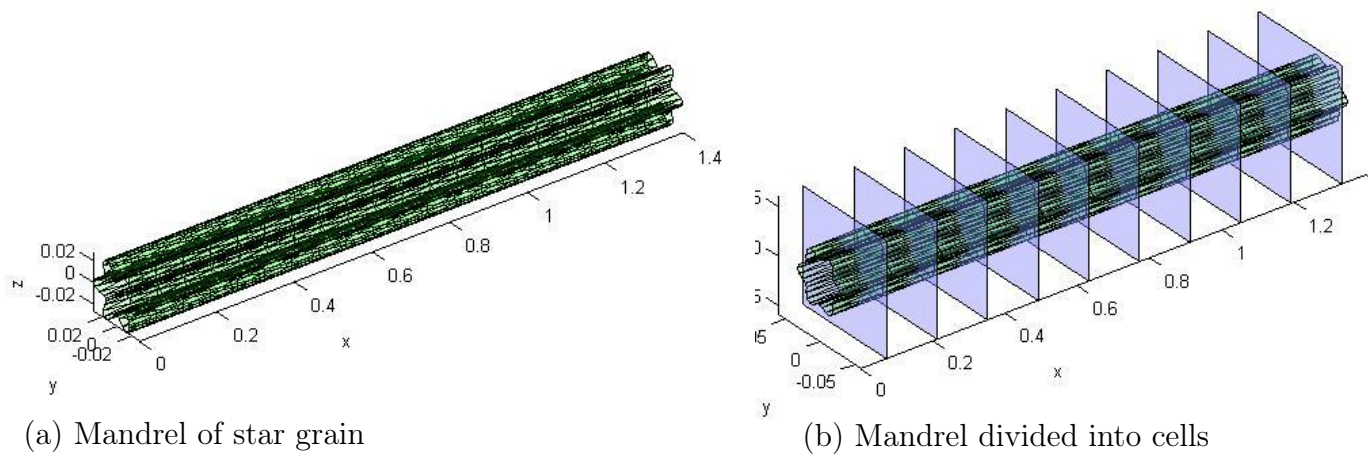


Figure 2.35. Mandrel of a star grain made up of a triangular mesh

The SDF values of the star grain polygon are shown in Figure 2.36(a) and the interface that approximates the original polygon is shown in Figure 2.36(b) plotted in red. To obtain the interfaces that approximates the original polygons, we solve

$$\phi(x, y) = 0, \quad (2.35)$$

on each plane. Initially we used “`contourc`” of MATLAB to obtain the contour at level

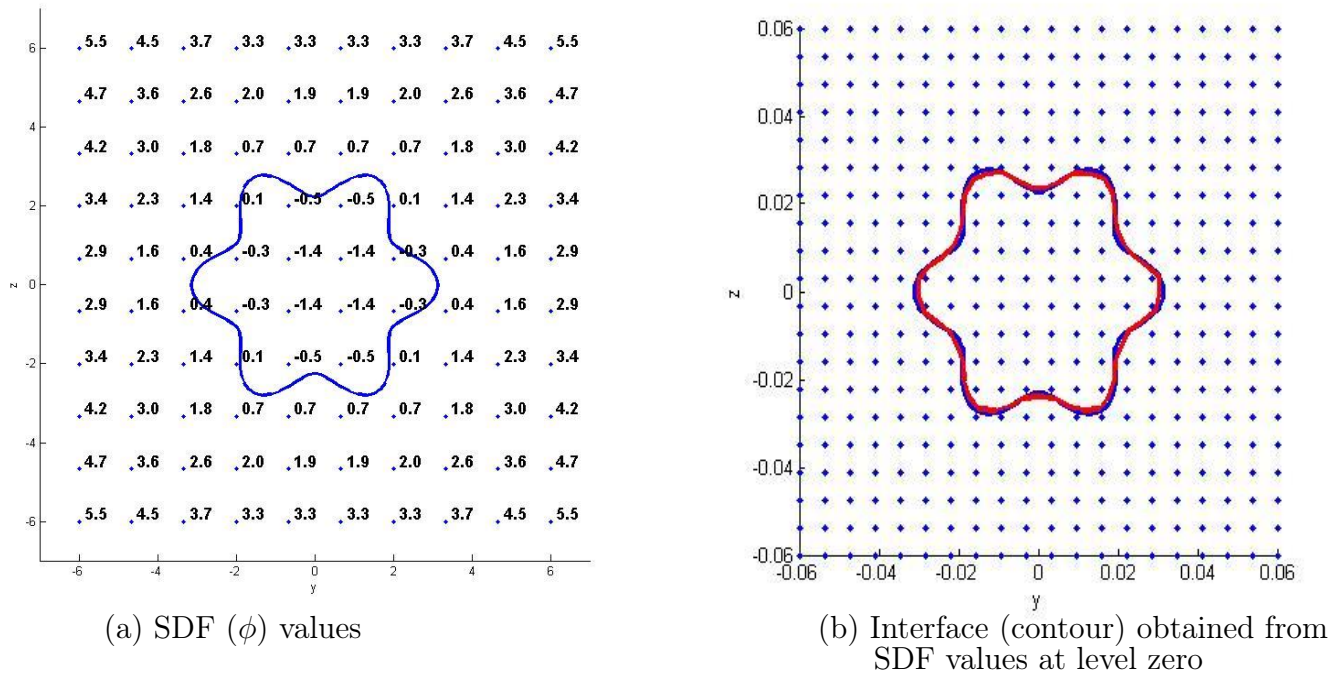


Figure 2.36. SDF values of a star grain and the interface obtained from them

zero. However, the function `contourc` has the following drawbacks:

- direction of obtaining the contour is inconsistent (i.e the contour is obtained in a clockwise direction and sometimes in counter-clockwise),
- can start anywhere when obtaining the contour,
- since the direction is inconsistent incorrect burning surface area is obtained,
- data points for level zero are sometimes divided into many data sets.

In order to address the problems of `contourc`, the code “`ZeroContour`” (see Appendix B) was created to replace `contourc`. `ZeroContour` gets the same data points as “`contourc`” of MATLAB. `ZeroContour` obtains the contour at zero level set by interpolation between negative and positive values of ϕ . The contour is obtained in a counter-clockwise direction starting from the positive x-axis. The contour data points obtained using `ZeroContour` for

a star grain are shown in Figure 2.37 (left) with white dots and the contour is plotted in Figure 2.37 (right) in blue. The black part outside the contour in the Figure 2.37 represent the positive values of ϕ , while the grey part inside the contour represent the negative values of ϕ .

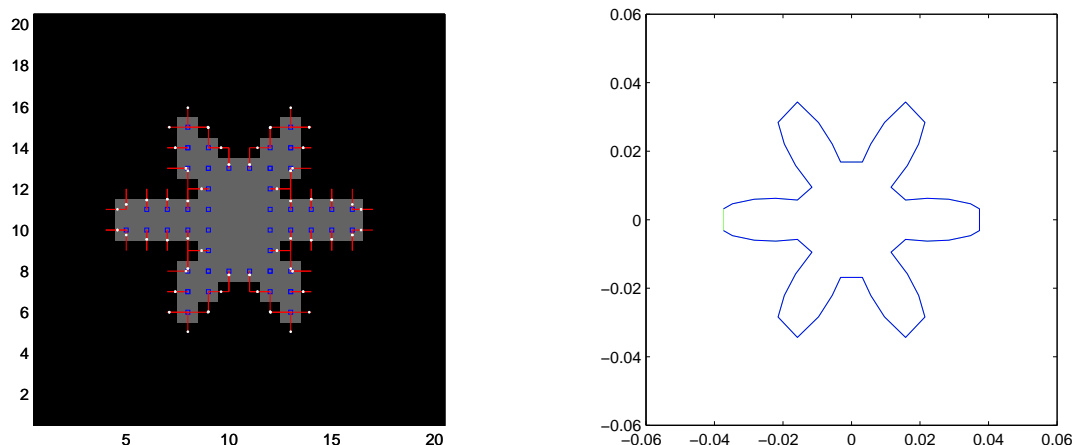


Figure 2.37. Data points (left) of the contour (right) at zero level obtained using Zero Contour.

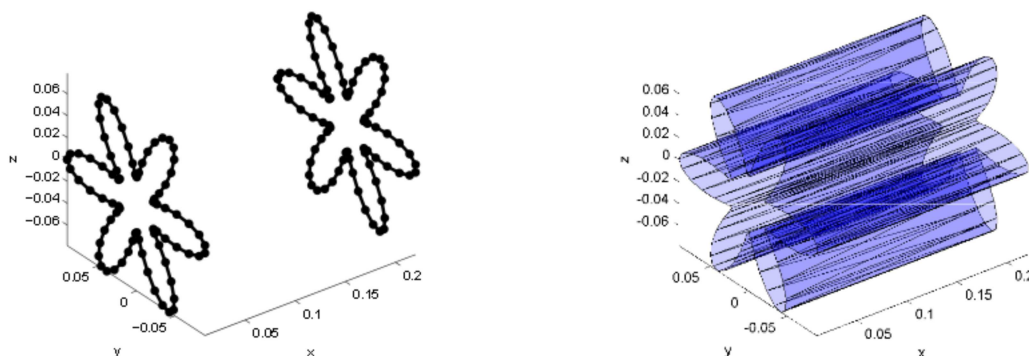


Figure 2.38. Patched surface (right) between polygons (left) of a cell

The grid points are only found in the places where the plane intersects the grain. In order to model the surface burnback we evolve the polygons found on the planes and form a surface by patching triangles between the polygons (see Figure 2.38). The polygons that we find with “ZeroContour” do not have the same number of data points and the spacing between them is not the same which leads to incorrect burning surface area calculation because the triangles become skew; this is shown in Figure 2.39 (left) and the correct

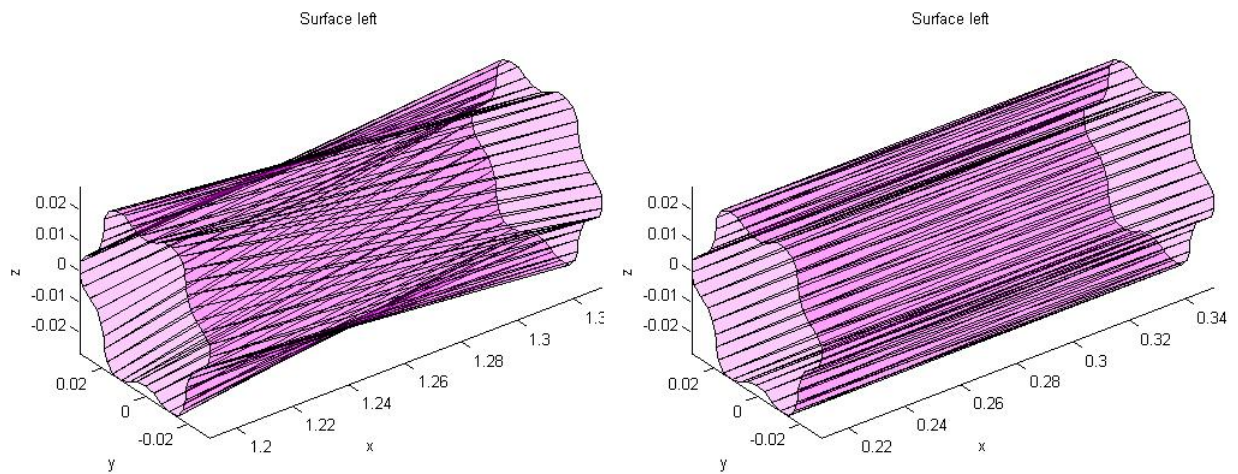


Figure 2.39. Burning surface of a cell

surface is shown on the right of Figure 2.39.

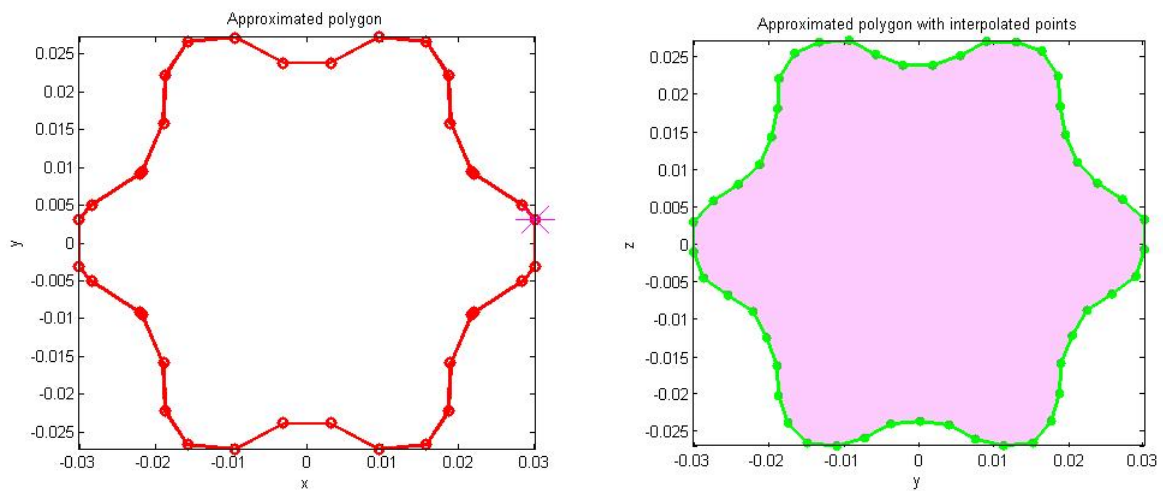


Figure 2.40. Contour obtained with ZeroContour (left) and interpolated equidistant data points (right)

In order to solve this problem we use interpolation to get the same number of points for each polygon and these points are equidistant to each other. Figure 2.40 (left) shows data points obtained using ZeroContour. These points are then interpolated to be equidistant to each other (see Figure 2.40 (right)). The shaded part shows the port area of the polygon. The correct surface resulting from equidistant data points of polygons is shown in Figure 2.39 (right).

2.10.1 Modelling the Casing

The grain burns until it reaches the casing before it burns out. Therefore, in order to model the casing, a circle with constant radius is used in 2D, and if the surface evolves until some points of the interface protrude out of the casing they are discarded by taking them to the casing. Figure 2.41 shows a casing plotted in magenta, burning star with some points outside the casing (black), and the remaining surface (blue) as the star burns.

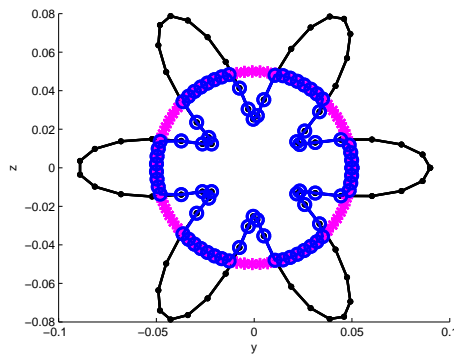


Figure 2.41. Casing modelling (magenta), evolving star (black), remaining surface (blue)

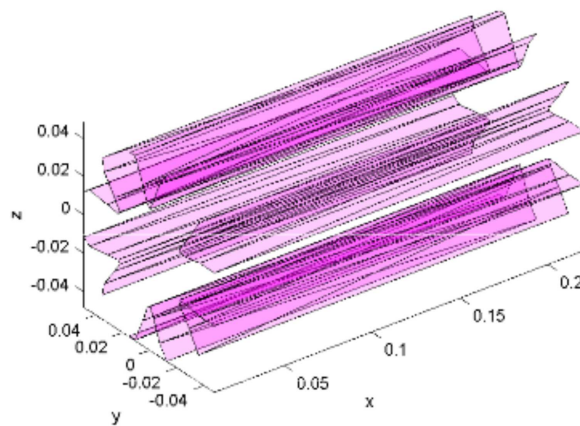


Figure 2.42. The burning surface area left in a burning star grain

The polygon in blue in Figure 2.41 helps to obtain the port area (A_a). In order to obtain the burning surface area, the triangles on the casing are discarded see Figure 2.42 then the remaining triangles give the burning surface area (A_s)

2.11 Implementation of the Level Set Method in 3D to Model burnback of the Solid Grain

In order to implement the level set methods in 3D, the grain is also divided into cells. Whenever the plane intersects the grain surface a polygon is obtained (see Figure 2.43 (left) plotted in blue). This polygon gives the port area of the cell. The function “polyarea” in MATLAB is used to find the port area (A_a). In order to obtain the polygon, we look for an edge of a triangle that has an x -coordinate on either side of \mathbf{x}_0 , where \mathbf{x}_0 is the plane that intersects with the grain. The triangle in which the edge belongs is identified. Then we look for the next edge of the same triangle that also intersects with the plane \mathbf{x}_0 . We then continue in this way until all the points that intersect with \mathbf{x}_0 are obtained.

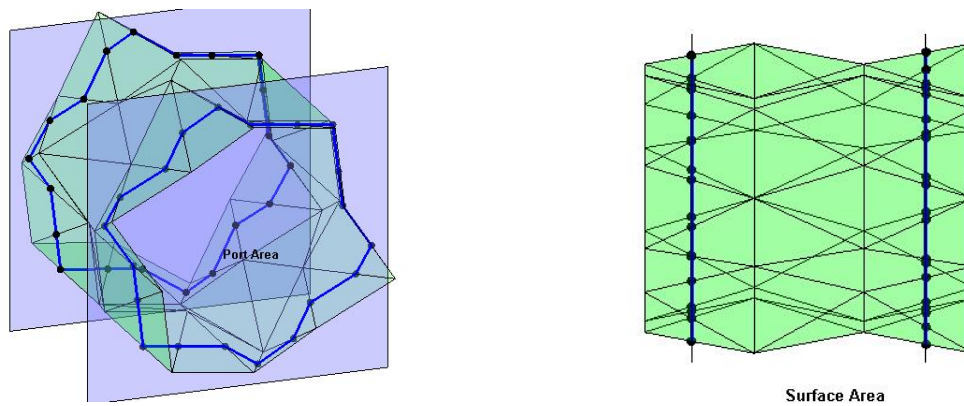
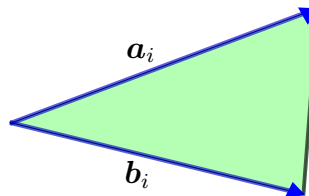


Figure 2.43. Polygon (in blue) representing a port area (left) and the burning surface area of a cell (right)

In order to calculate the internal ballistics process, we also need to know the burning surface area (A_s) as the grain burns. We use the method illustrated below to calculate the burning surface area:

An i^{th} triangle formed by vectors \mathbf{a}_i and \mathbf{b}_i has the area given by,

$$(A_s)_{\text{inside}} = \sum_{i=1}^n \frac{1}{2} \|\mathbf{a}_i \times \mathbf{b}_i\|,$$



where n is the number of triangles that are completely inside the cell. Then for the triangles with some part of them inside the cell and the other part outside the cell as shown in Figure 2.44, their area is calculated as follows: Let

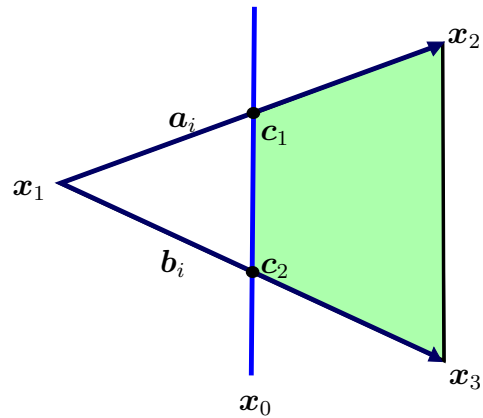


Figure 2.44. Portion of the triangle inside

$$\lambda = \frac{\|\mathbf{c}_1 - \mathbf{x}_1\|}{\|\mathbf{x}_2 - \mathbf{x}_1\|}, \quad (2.36)$$

and

$$\mu = \frac{\|\mathbf{c}_2 - \mathbf{x}_1\|}{\|\mathbf{x}_3 - \mathbf{x}_1\|}, \quad (2.37)$$

then

$$(A_s)_{\text{partly inside}} = \sum_{j=1}^k \frac{1}{2} (1 - \lambda\mu) \|\mathbf{a}_i \times \mathbf{b}_i\|, \quad (2.38)$$

therefore the burning surface area is given by,

$$A_s = (A_s)_{\text{inside}} + (A_s)_{\text{partly inside}}. \quad (2.39)$$

The 3D burnback modelling was only applied to the tubular grain.

Chapter 3

Gas Dynamics and Internal Ballistics

In order to simulate the surface evolution of the grain, we need to determine the burn speed V which is given by [4],

$$V = cP^{\bar{n}}, \quad (3.1)$$

where P is the pressure of the gas at the burning surface. This pressure is obtained from the gas that flows inside the combustion chamber. The flow inside the chamber is subsonic (i.e velocity of the gas is less than Mach 1). In order to determine the pressure inside the casing, it is necessary to study gas dynamics inside the combustion chamber. This chapter discusses the gas dynamics equations used to obtain the pressure of the gas, which is then used to find the burn rate velocity in equation (3.1).

The gas that flows in the combustion chamber is compressible and can undergo density changes at high speeds. These changes in density are accompanied with changes in pressure and temperature.

3.1 Stagnation Properties

Nozzles typically operate at steady-state with one inlet and outlet, as shown in Figure 3.1. Let us consider a nozzle in which no shocks occur. Therefore, the outlet pressure must be either equal to the critical pressure or the design pressure (these are explained later in this chapter). The mass balance requires that the mass flow rate at the inlet must be equal to the mass flow rate at the exit:

$$\dot{m} = \frac{A_{in}u_{in}}{\nu_{in}} = \frac{A_{out}u_{out}}{\nu_{out}}, \quad (3.2)$$

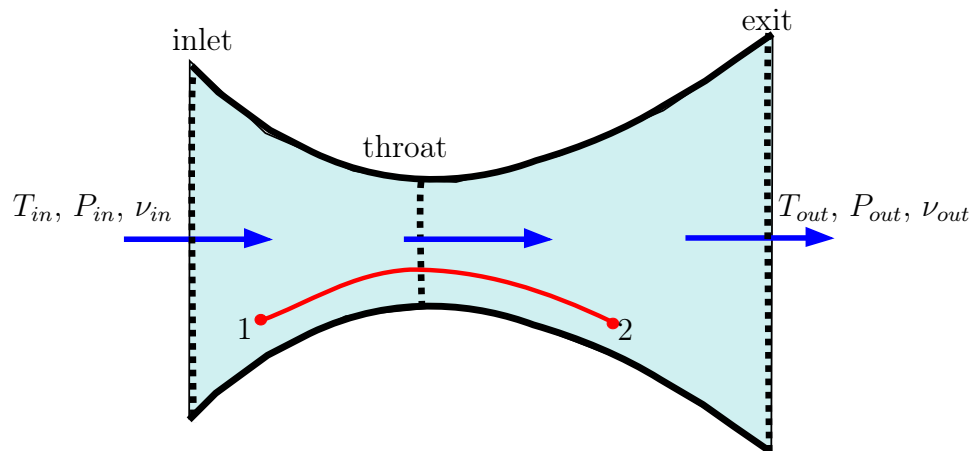


Figure 3.1. Nozzle operating at steady-state

where A_{in} and A_{out} are cross-sectional areas at the nozzle inlet and outlet respectively. The specific volume of the gas is defined by ν . This specific volume of gas can vary significantly between the inlet and outlet, depending on the differences in temperature and pressure that occur in the nozzle.

It is customary to define the *stagnation state* in compressible flow applications. The stagnation state is defined as the state where the gas velocity becomes zero and all the kinetic energy has been converted into pressure energy.

3.2 Isentropic Flow of Gas

Isentropic flow in simple terms means flow at constant entropy. This means that the gas flows isentropically up to a shock along a streamline. If we have T_1, P_1 and T_2, P_2 denoting the temperature and pressure at position 1 and 2 respectively in Figure 3.1, then the relation between pressure and temperature of an ideal gas with constant specific heat capacities undergoing an isentropic process, is given by,

$$\frac{T_2}{T_1} = \left(\frac{P_2}{P_1} \right)^{\frac{\gamma-1}{\gamma}}, \quad (3.3)$$

where γ is the ratio of specific heat capacities $\frac{c_p}{c_v}$. The specific heat for mono-atomic gases is high since their energy is stored only in one form, i.e. as a chemical potential energy. These gases include Neon, Argon, and Helium all with $\gamma = 1.667$. As the number of atoms

increases in a molecule the specific ratio value drops because there are intermolecular forces between the atoms that cause them to vibrate. Therefore, for diatomic molecules like Oxygen and Nitrogen have $\gamma = 1.4$.

We shall derive a number of useful equations below relating to various quantities at two positions: at the stagnation point and a point somewhere else. A gas that is reversibly and adiabatically decelerated to its stagnation state (T_s, P_s) undergoes an isentropic process; therefore the temperature and pressure are related to the stagnation temperature according to:

$$\frac{T}{T_s} = \left(\frac{P}{P_s} \right)^{\frac{\gamma-1}{\gamma}}. \quad (3.4)$$

The velocity of an ideal gas (a gas in which collisions between atoms or molecules are perfectly elastic because there are no intermolecular attractive forces) with constant c_p can be expressed in terms of temperatures as

$$u^2 = 2c_p(T_s - T). \quad (3.5)$$

The speed of sound is the distance per unit time travelled by sound waves propagating through an elastic medium. In dry air, the speed of sound is 343.2 m.s^{-1} . The ratio of the local velocity of the gas to the speed of sound is called the Mach number (M) [38],

$$M = \frac{u}{a} \quad (3.6)$$

where u is the local velocity of the gas and a is the speed of sound.

Equation (3.5) expressed in terms of the Mach number yields,

$$M^2 = \frac{2c_p(T_s - T)}{a^2}, \quad (3.7)$$

where a is the speed of sound and T_s is the stagnation temperature.

For an ideal gas, the speed of sound is given by $a = \sqrt{\gamma RT}$, which is then substituted into equation (3.7). The resulting equation that expresses the temperature ratio is given by,

$$\frac{T}{T_s} = \left(1 + \frac{\gamma R}{2c_p} M^2 \right)^{-1}, \quad (3.8)$$

where T is the temperature and T_s is the stagnation temperature. Substituting $R = c_p - c_v$ into equation (3.8), gives:

$$\frac{T}{T_s} = \left(1 + \frac{\gamma - 1}{2} M^2 \right)^{-1}. \quad (3.9)$$

The temperature ratio on the left hand side of equation (3.9) can be replaced by the ratio of the local pressure to the stagnation pressure using equation (3.4):

$$\frac{P}{P_s} = \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{-\gamma}{\gamma - 1}}. \quad (3.10)$$

3.3 Mass Flow Rate

A simple setup for studying flow through a nozzle consists of mass flow rate which can be derived in terms of local pressure and stagnation pressure ratio [4].

When considering an isentropic compressible flow from the reservoir where the stagnation conditions prevail, the Bernoulli's equation is given by

$$\frac{1}{2}u^2 + \frac{a^2}{\gamma - 1} = \frac{a_s^2}{\gamma - 1}, \quad (3.11)$$

where u , a , a_s is velocity of the gas, acoustic speed (speed of sound), and stagnation acoustic speed. When using the equation $a^2 = \frac{\gamma P}{\rho}$ (where P is the pressure, ρ denotes the density of the gas and γ is the specific heats ratio) to substitute for a^2 , we have

$$\frac{1}{2}u^2 + \left(\frac{\gamma}{\gamma - 1}\right) \frac{P}{\rho} = \left(\frac{\gamma}{\gamma - 1}\right) \frac{P_s}{\rho_s}, \quad (3.12)$$

so that

$$u = \left[\frac{2\gamma}{\gamma - 1} \frac{P_s}{\rho_s} \left(1 - \frac{P}{P_s} \frac{\rho_s}{\rho}\right) \right]^{\frac{1}{2}}, \quad (3.13)$$

where P_s , ρ_s are stagnation pressure and stagnation density respectively. The mass flux in a tube across the cross-sectional area A per unit time from the continuity equation is given by,

$$\dot{m} = \rho u A. \quad (3.14)$$

When using the relation $\frac{P}{P_s} = \left(\frac{\rho}{\rho_s}\right)^\gamma$, the mass flow equation is given by,

$$\dot{m} = \rho_s u A \left(\frac{P}{P_s}\right)^{\frac{1}{\gamma}}. \quad (3.15)$$

Substituting equation (3.13) into equation (3.15), yields

$$\dot{m} = A \left[\frac{2\gamma}{\gamma-1} P_s \rho_s \left(\frac{P}{P_s} \right)^{\frac{2}{\gamma}} \left\{ 1 - \left(\frac{P}{P_s} \right)^{\frac{\gamma-1}{\gamma}} \right\} \right]^{\frac{1}{2}}, \quad (3.16)$$

which is the mass flow rate as the function of the ratio of pressure and stagnation pressure.

The mass flow rate defined by equation (3.16) reaches a maximum. This maximum can be found as follows:

Let $\bar{P} = \frac{P}{P_s}$ in equation (3.16), we then square the result and differentiate it with respect to \bar{P} [4]. This yields,

$$2\dot{m} \frac{d\dot{m}}{d\bar{P}} = \frac{K(\gamma+1)}{\gamma} \bar{P}^{\frac{(2-\gamma)}{\gamma}} \left[\left(\frac{2}{\gamma+1} \right) - \bar{P}^{\frac{\gamma-1}{\gamma}} \right]. \quad (3.17)$$

The above equation shows that $\frac{d\dot{m}}{d\bar{P}} = 0$ when $\bar{P} = \left(\frac{2}{\gamma+1} \right)^{\frac{\gamma}{\gamma+1}}$, therefore the maximum is reached when

$$\frac{P}{P_s} = \left(\frac{2}{\gamma+1} \right)^{\frac{\gamma}{\gamma+1}}, \quad (3.18)$$

which is given by,

$$\dot{m}_{\max} = A \left[\left(\frac{2\gamma}{\gamma+1} \right) P_s \rho_s \left(\frac{2}{\gamma+1} \right)^{\frac{2}{\gamma-1}} \right]^{\frac{1}{2}}. \quad (3.19)$$

It is convenient to study mass flow rate as a fraction of the maximum mass flow, therefore dividing equation (3.16) by equation (3.19) yields,

$$\frac{\dot{m}}{\dot{m}_{\max}} = \left(\frac{\gamma+1}{\gamma-1} \right)^{\frac{1}{2}} \left[\frac{1}{2} (\gamma+1)^{\frac{1}{\gamma-1}} \left(\frac{P}{P_s} \right)^{\frac{1}{\gamma}} \right] \left[1 - \left(\frac{P}{P_s} \right)^{\frac{\gamma-1}{\gamma}} \right]^{\frac{1}{2}}, \quad (3.20)$$

where \dot{m}_{\max} is the maximum mass flow rate.

We shall now interpret the meaning of equation (3.20). Consider a converging nozzle connected to a reservoir where the stagnation conditions prevail, see Figure 3.2. The pressure level P_b at the exit of the nozzle is referred to as the back pressure (or ambient pressure) which is the pressure of the medium and it is the pressure that determines the flow in the nozzle. In Figure 3.2, the ratio $\frac{P}{P_s}$ in equation (3.20) is given by $\frac{P_b}{P_s}$, where P_s is the pressure inside the reservoir.

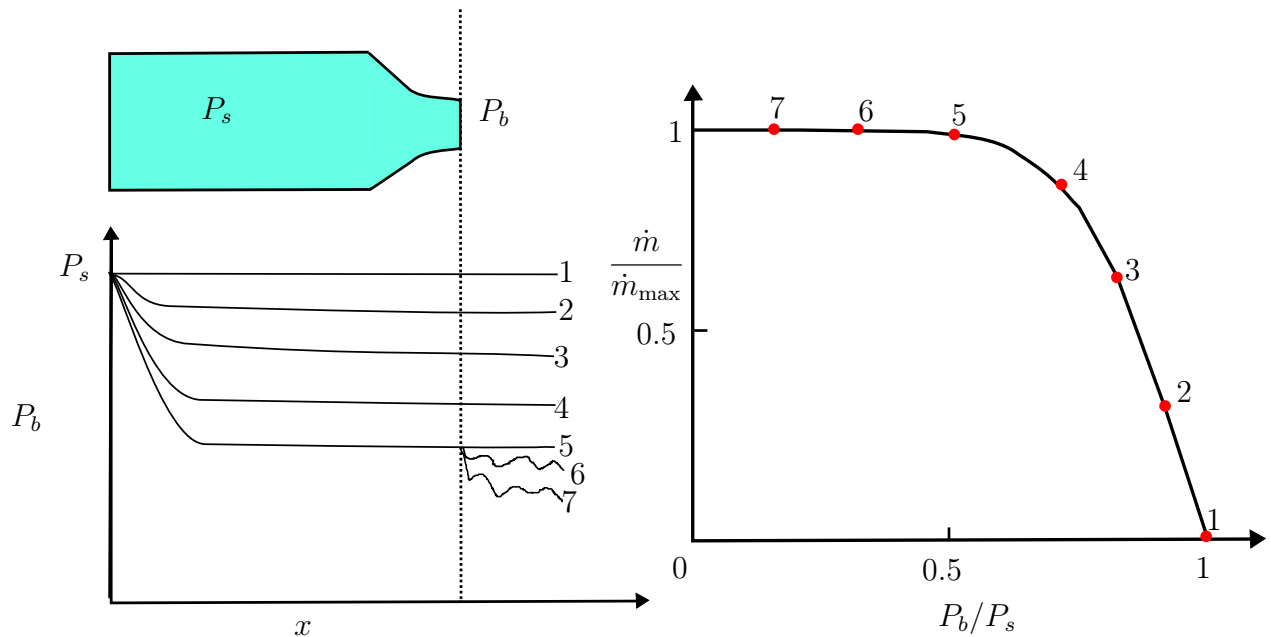


Figure 3.2. Flow through a converging nozzle

When the back pressure is equal to the reservoir pressure, there is no flow in the nozzle. This is condition 1 on the graph. Reducing P_b slightly (condition 2 on the graph), the flow is induced in the nozzle. It can be seen from the graph that for relatively high values of P_b , the mass flow rate is small, meaning that the flow is subsonic throughout (gas velocity lower than the local speed of sound). A further reduction in back pressure results in subsonic flow still, but with high values of Mach number close to Mach 1 because of the increased mass flow rate. It is clear from the graph that the mass flow rate increases as the speed increases. When P_b is reduced we have an increased Mach number at the exit with an increased mass flow rate. At a particular value of back pressure the flow reaches sonic conditions (gas velocity equal to the local speed of sound) [4].

When the back pressure is further reduced, an interesting phenomenon emerges: if the Mach number at the exit tries to increase, then it demands an increase in the mass flow rate, but the condition is sonic, therefore this means there is no propagation of pressure information upstream. The reservoir is “unaware” of the conditions downstream and it does not send any more mass flow. Consequently the flow patterns remain unchanged in the nozzle. Any adjustments to the back pressure takes place outside the nozzle. The nozzle is said to be choked [35]. This occurs because the mass flow rate through the nozzle has reached its maximum possible value.

3.4 Area Ratio

Most SRM's operate at supersonic speeds (gas velocity at the nozzle outlet greater than the speed of sound). This speed is achieved because of the nozzle design. Therefore, in the nozzle design process it is advantageous to study the area ratio.

If we have \dot{m}_{\max} denoting the maximum mass flow rate, using equation (3.16) we can write the area of any section in the nozzle as,

$$A = \frac{\dot{m}_{\max}}{\left[\frac{2\gamma}{\gamma-1} P_s \rho_s \left(\frac{P}{P_s} \right)^{\frac{2}{\gamma}} \left\{ 1 - \left(\frac{P}{P_s} \right)^{\frac{\gamma-1}{\gamma}} \right\} \right]^{\frac{1}{2}}}, \quad (3.21)$$

where P is the pressure inside the nozzle. At the throat where $A = A_t$, we have

$$A_t = \frac{\dot{m}_{\max}}{\left[\frac{2\gamma}{\gamma-1} P_s \rho_s \left(\frac{P^*}{P_s} \right)^{\frac{2}{\gamma}} \left\{ 1 - \left(\frac{P^*}{P_s} \right)^{\frac{\gamma-1}{\gamma}} \right\} \right]^{\frac{1}{2}}}, \quad (3.22)$$

where A_t is the throat area and P^* is the pressure at the throat. The \star denotes that the sonic conditions prevail. If we square equations (3.21) and (3.22) and get their quotient, we obtain

$$\left(\frac{A}{A_t} \right)^2 = \frac{\left(\frac{P^*}{P_s} \right)^{\frac{2}{\gamma}} \left[1 - \left(\frac{P^*}{P_s} \right)^{\frac{\gamma-1}{\gamma}} \right]}{\left(\frac{P}{P_s} \right)^{\frac{2}{\gamma}} \left[1 - \left(\frac{P}{P_s} \right)^{\frac{\gamma-1}{\gamma}} \right]}. \quad (3.23)$$

Now when using equation (3.10) and (3.18), equation (3.23) is given by,

$$\left(\frac{A}{A_t} \right)^2 = \frac{1}{M^2} \left[\frac{2}{\gamma+1} \left(1 + \frac{\gamma-1}{2} M^2 \right) \right]^{\frac{\gamma+1}{\gamma-1}}. \quad (3.24)$$

Equation (3.24) helps to determine changes in Mach number (M) as the area changes.

For air with $\gamma = 1.4$, equation (3.24) becomes

$$\frac{A}{A_t} = \frac{1}{M} \left(\frac{5}{6} + \frac{1}{6} M^2 \right)^3. \quad (3.25)$$

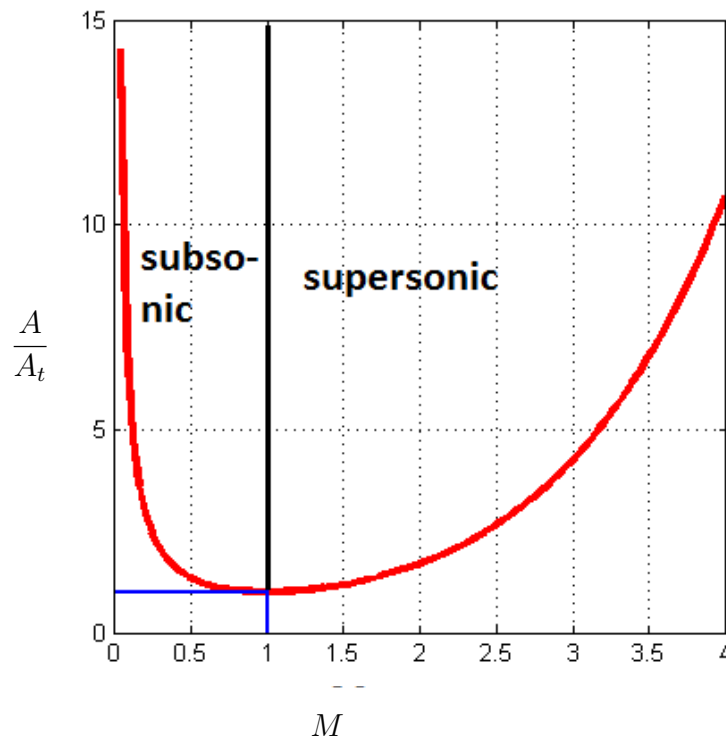


Figure 3.3. Area as a function of Mach number

The graph in Figure 3.3 shows the area ratio $\frac{A}{A_t}$ from equation (3.25) plotted as function of Mach number. It is clear from the graph that for subsonic flows, the Mach number increases as the area decreases and it decreases as the area increases. On the contrary, for supersonic flows, the Mach number decreases as the area decreases and it increases as the area increases.

3.5 Flow Behaviour in the Converging-Diverging Nozzles

The chemical potential energy of the SRM is changed into kinetic energy by the nozzle. The rate at which this happens will determine the thrust the SRM can achieve. A nozzle must have a converging-diverging geometry in order to achieve supersonic velocities, as shown by the analysis in the above section using Figure 3.3. The converging-diverging nozzle is also known as a *de Laval* nozzle [22]. The gas velocity at the exit of a converging nozzle is limited to the speed of sound.

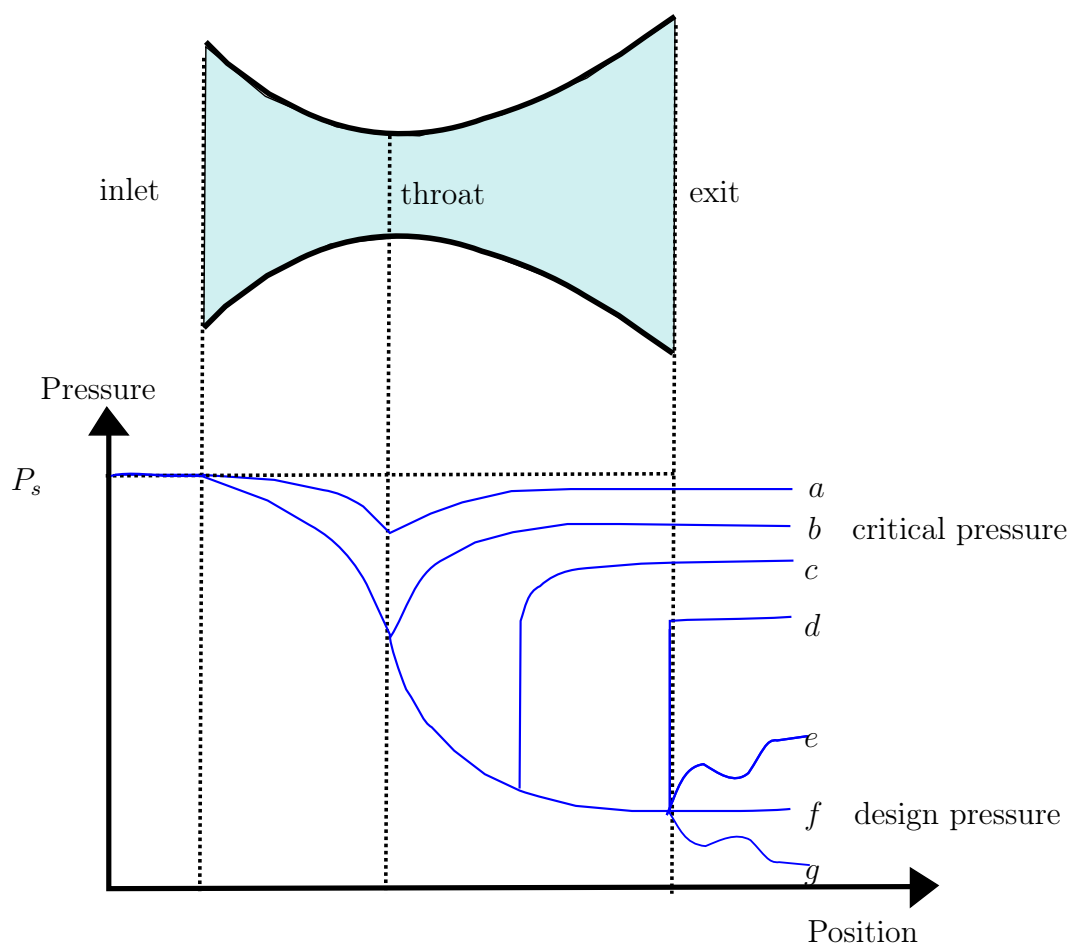


Figure 3.4. Pressure as a function of position for flow in converging-diverging nozzle with different values of the nozzle exit pressure

Figure 3.4 illustrates the pressure as function of position for a compressible flow through

a converging-diverging nozzle. The geometry of the nozzle, which has a converging section followed by a diverging section, is illustrated at the top of the figure. The different graphs in Figure 3.4 (labelled *a* through *g*) are each associated with different operating conditions. Each operating condition has the same inlet pressure (P_s) but a different back pressure. The back pressure is highest for case *a* and lowest for case *g*.

When the back pressure is slightly lower than the inlet pressure (corresponding to case *a*), the flow is subsonic throughout the nozzle. In this case, the lowest pressure and highest velocity of the gas occur at the nozzle throat where the cross-sectional area reaches a minimum. As the back pressure is decreased, the velocity at the throat continues to increase until eventually it reaches the local speed of sound at the throat, corresponding to case *b*.

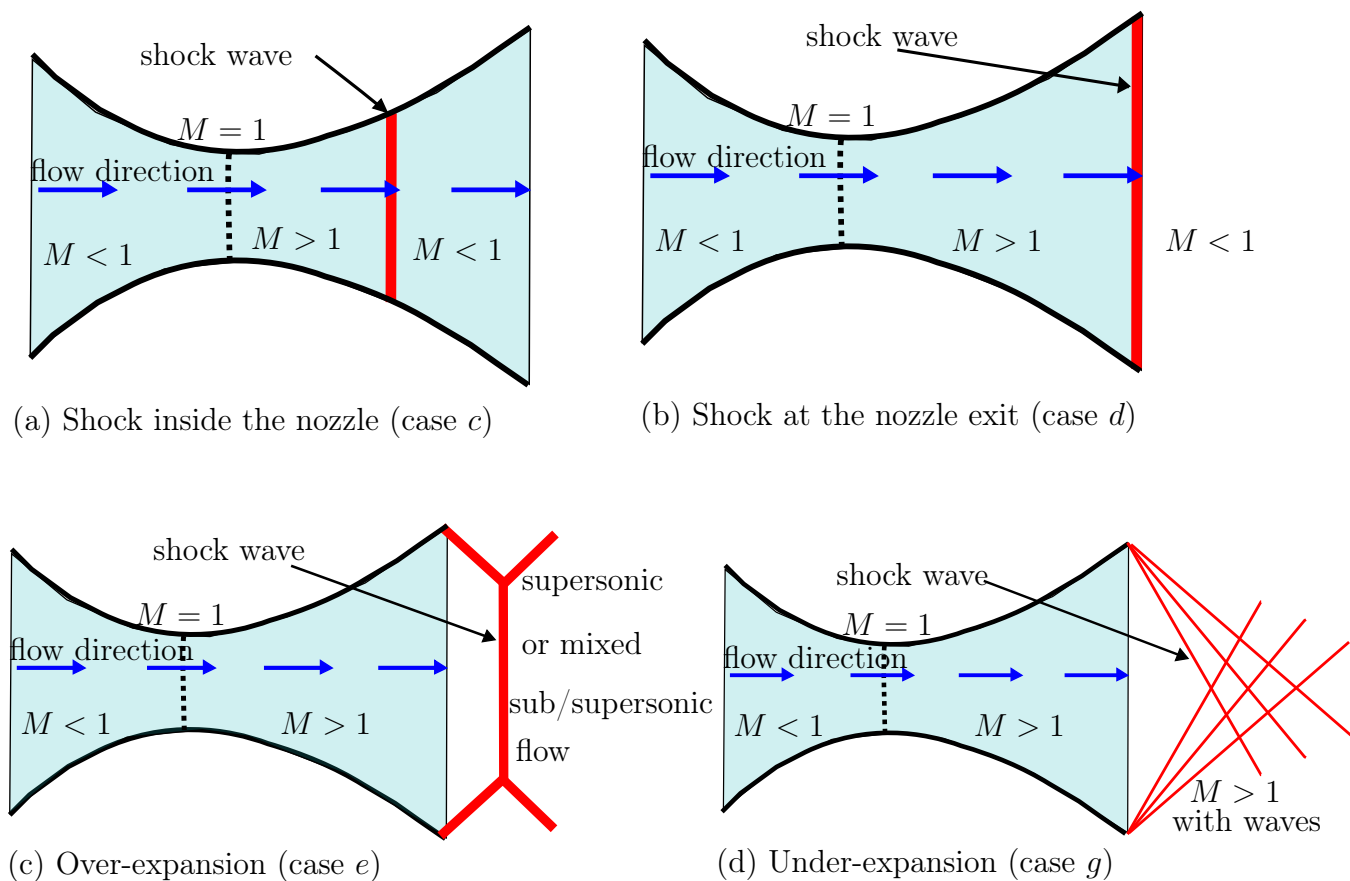


Figure 3.5. Flow patterns in the nozzle

The Mach number at the throat for case b is unity; this corresponds to the highest velocity that the gas can achieve in a converging nozzle. The highest back pressure for which the Mach number is unity is referred to as the *critical back pressure* [4]. When the Mach number is unity, the nozzle is choked and the mass flow reaches its maximum value for a set of inlet conditions. Under these conditions, the mass flow rate is insensitive to the back pressure. As the gas progresses through the diverging section of the nozzle for case b , its pressure slowly increases as velocity decreases until the gas exits the nozzle at the critical back pressure. No discontinuities are observed for case b .

As the back pressure is reduced still further (case c in Figure 3.4) the mass flow rate through the nozzle remains constant but the flow accelerates from speed of sound at the throat to supersonic velocities in the diverging section of the nozzle. At some point downstream of the throat, the flow suddenly changes from supersonic to subsonic. This change in velocity is accompanied by an abrupt increase in pressure and an entropy change. The behaviour occurring at this pressure discontinuity is referred to as a *shock wave* [4] (see Figure 3.5(a)). The shock wave occurring inside the nozzle is called *normal shock wave* since it is perpendicular to flow direction. Note that shock waves can only occur in locations where the upstream flow is supersonic. Downstream of the shock wave, the flow is subsonic. Thereafter the pressure smoothly increases and the velocity decreases as the flow travels from the shock wave to the nozzle exit. The flow emerges as subsonic because of the imposed back pressure.

The location of the normal shock wave towards the nozzle exit as the back pressure is further reduced, changes. Case d in Figure 3.4 illustrates the behaviour for the back pressure value that causes a shock wave exactly at the nozzle exit (see Figure 3.5(b)). If the back pressure is decreased still further, as shown in case e in Figure 3.4, an *oblique shock wave* occurs outside the nozzle as the pressure adjusts from the supersonic condition at the nozzle exit to the imposed back pressure (see Figure 3.5(c)). The oblique shock wave will, in general, not be perpendicular to the flow. This condition has a discharge pressure (pressure at the nozzle exit) that is lower than the imposed back pressure and is referred to as *over-expansion*.

When the back pressure is reduced to the *design pressure* corresponding to case f in Figure 3.4, supersonic flow leaves the nozzle with an exit pressure that is exactly equal to the applied back pressure and there is no shock that occurs. The flow is isentropic throughout the nozzle. If the back pressure is reduced to a value that is below the design pressure, corresponding to case g , an *oblique expansion waves* results (see Figure 3.5(d)). This behaviour is not a result pressure discontinuity, but rather a wave that is a result of the flow exiting the nozzle coming to equilibrium with the low external pressure. This situation is called *under-expansion*.

3.6 Calculation of Flow in Nozzles

In order to derive a model for the internal ballistics process discussed in Lamberty [21], we have to express the mass flow rate in terms of Mach number and other quantities.

The nozzles that are used for flow measurement ordinarily consist of a converging section only. The reason nozzles are used for flow measurement is that an analytical equation can be determined for the flow rate that requires only knowledge of the throat diameter, the nozzle inlet and outlet pressures, and the inlet air density. If sonic velocity occurs at the nozzle throat (i.e the nozzle choking), the mass flow rate is independent of the outlet pressure and therefore the inlet pressure, temperature and throat diameter completely determine the mass flow rate [38].

The analytical equations for the mass flow rate can be derived using isentropic equations, assuming ideal gas behaviour with constant specific heat capacities. The mass flow rate at any point in the nozzle is:

$$\dot{m} = Au\rho = A(Ma) \left(\frac{P}{RT} \right). \quad (3.26)$$

We will first determine an expression for the mass flow rate at the critical area, A^* , which is the area at which the Mach number is 1. If the nozzle is choked, the critical area is at the throat of the nozzle (i.e $A^* = A_t$); note that the throat of a converging nozzle is the exit area. If the back pressure is sufficiently high, so that sonic velocity is not achieved at the exit, then the critical area does not physically exist. However, the critical area will be useful for deriving an expression for mass flow rate in any case.

At the critical area, $M = 1$, pressure and temperature at this speed are denoted by P^* and T^* . The local speed of sound at the location of the critical area, is given by $a^* = \sqrt{\gamma RT^*}$. Substituting this information into equation (3.26), the mass flow rate is expressed as

$$\dot{m} = A_t M P^* \left(\sqrt{\frac{\gamma}{RT^*}} \right)^{\frac{1}{2}}. \quad (3.27)$$

Assuming isentropic flow, the temperature and pressure at any point in the nozzle are given by equations (3.9) and (3.10). Applying these equations at the critical area determines T^* and P^* ,

$$\frac{T^*}{T_s} = \left(1 + \frac{\gamma - 1}{2} M^2 \right)^{-1}, \quad (3.28)$$

$$\frac{P^*}{P_s} = \left(1 + \frac{\gamma - 1}{2} M^2 \right)^{\frac{-\gamma}{\gamma - 1}}. \quad (3.29)$$

Substituting equations (3.28) into equation (3.27) gives,

$$\dot{m} = A_t M P^* \left[\frac{\gamma}{RT_s} \left(1 + \frac{\gamma - 1}{2} M^2 \right) \right]^{\frac{1}{2}}. \quad (3.30)$$

3.7 Internal Ballistics Parameters

3.7.1 Burning Rate

It is assumed that the propellant grain burns in a direction perpendicular to the grain surface. The rate at which a propellant burns is usually described by a reference value at a specific pressure [35]. This value is called the burning rate and its units is meters per second. Since it is an independent parameter, the burning rate is one of the propellant properties.

Apart from the propellant formulation and propellant manufacturing process, burning rate can be increased by the following [20]:

- Combustion chamber pressure
- Initial temperature of the solid propellant
- Combustion gas temperature

The relation between burning rate and the chamber pressure is governed by the following empirical equation, also known as Saint Robert's burn rate law [4],

$$\dot{r} = c P^{\bar{n}}, \quad (3.31)$$

where \dot{r} is the burn rate velocity, P is the pressure of gas, c is the burn rate coefficient, \bar{n} is the burn rate exponent. This empirical expression defines the burning rate of the propellant, the values of c and \bar{n} usually derived from small sub-scale burning rate test motor firings at different operating pressures.

3.7.2 Characteristic Velocity

The characteristic velocity C^* is a function of the propellant characteristics and combustion chamber design, and it is independent of nozzle characteristics. The C^* is used in comparing the relative performance of different chemical rocket propulsion system designs

and propellants. It is easily determined from measured data of mass flow rate, chamber pressure and throat area. The characteristic velocity can be formulated as [35],

$$C^* = \frac{PA_t}{\dot{m}}. \quad (3.32)$$

3.7.3 Chamber Pressure

Chamber pressure is the gas pressure inside the combustion chamber during motor operation. In the grain design process a limit on maximum pressure is usually established. Concurrent with grain design, the motor case and other structural components are being designed and analysed with this maximum pressure. This constraint on chamber pressure is usually named as *maximum expected operating pressure* [11].

3.8 Internal Flow Modelling of Solid Rockets Using Gas Dynamics

To understand the internal flow of the SRM, we need to have a method to calculate the internal ballistics. We will employ the method described by Lamberty [21]. The grain is divided into a number of cells that are necessary to obtain satisfactory prediction of the internal flow velocities and pressure, stagnation pressure loss, and individual surface area locations. Figure 3.6 shows the method applied to a simple grain and description of the flow conditions in one of the elements.

The grain is divided into N cells and we shall use i as the cell index so that $i = 1, 2, 3, \dots, N$. The variables in (e.g. pressure) in two adjacent cells are given by,

$$P_{i-1} \quad \text{and} \quad P_i \quad (3.33)$$

for the previous cell and the i^{th} cell respectively.

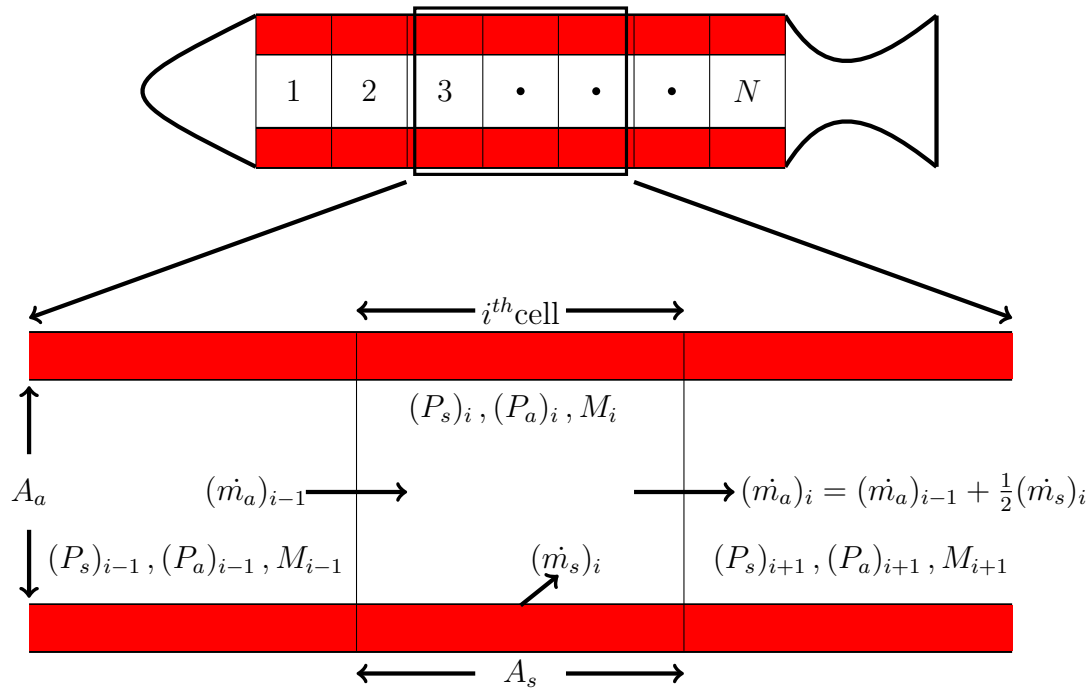


Figure 3.6. Internal Ballistics element method

The subscripts s and a represent the quantity at the surface and axial direction respectively. This means that A_a is the port area and A_s is the burning surface area. The symbols $(\dot{m}_s)_i$ and $(\dot{m}_a)_i$ denote mass flow rate at the surface and the mass flow rate along the axial direction for the i^{th} cell respectively.

3.8.1 Grid Generation and the Burned Away Distance

Since the gases of the solid rocket move towards the nozzle as the grain burns, we will label this direction in which the gases move as the x direction. Therefore the x -axis is the central axis of the rocket. The head of the rocket will always be at $x = 0$.

To generate the grid used in the level set method, N_x is the number of grid points in the x direction which must be always be greater than the number of cells. N_y, N_z are the number of grid points along y and z directions respectively. For convenience sake, we shall assume that the number of grid points in the y and z directions are always equal.

In terms of advancement in time, we shall use n as the time step index, so that,

$$t^{n+1} = t^n + \Delta t, \quad (3.34)$$

where t is the time in seconds, and Δt is the time step.

In order to start burning the grain, we set the initial starting surface pressure (P_s) to 3 MPa at starting time ($t = 0$ s). Then we use the burning rate equation,

$$\dot{r} = cP^{\bar{n}}, \quad (3.35)$$

to determine the initial velocity at which the grain burns. The surface is burned away at the rate \dot{r} . Note that here \dot{r} is the same as V , which is the velocity to evolve the interface in the level sets method.

3.8.2 Gas Dynamics Used to Model the Flow Inside the Combustion Chamber

In order to analyse the flow of gases in the combustion chamber, we will use the 1-Dimensional approach proposed by Lamberty [21]. The burning rate that we found using the burn rate equation (3.31) with the use of the initial starting pressure at $t = 0$ s, is used to calculate the mass flux going out of the combustion chamber, which also determines the pressure, which in turn determines the next burn rate.

The mass flux from the burning surface of the i^{th} cell is given by,

$$(\dot{m}_s)_i = \rho(A_s)_i(\dot{r}_i). \quad (3.36)$$

The flux introduced into the i^{th} cell (taken at the centre of the cell) is therefore

$$(\dot{m}_a)_i = (\dot{m}_s)_1 + (\dot{m}_s)_2 + \cdots + (\dot{m}_s)_{i-1} + \frac{1}{2}(\dot{m}_s)_i, \quad (3.37)$$

which in short can be written as,

$$(\dot{m}_a)_i = (\dot{m}_a)_{i-1} + \frac{1}{2}(\dot{m}_s)_i, \quad (3.38)$$

where $(\dot{m}_s)_i$, $(\dot{m}_a)_i$ are the mass fluxes from the surface and along the central axis respectively. The first cell does not get mass from a cell before it, therefore

$$(\dot{m}_a)_1 = \frac{1}{2}(\dot{m}_s)_1. \quad (3.39)$$

To calculate the pressure contribution to cell i depends on the mass addition, \dot{m}_s , and the mass that is lost to the next cell, \dot{m}_a , the current stagnation pressure in the cell, $(P_s)_i$, as

well as the speed of the gas in the cell, M_i . Lamberty [21] proposes that,

$$(\Delta P_s)_i = \gamma(M_i)^2 \frac{(\dot{m}_s)_i}{(\dot{m}_a)_i} (P_s)_i. \quad (3.40)$$

The mean stagnation pressure per cell is obtained from the differences between the adjacent cells as well as the contribution above. It is given by,

$$(P_s)_i = (P_s)_{i-1} + \left[\frac{(\Delta P_s)_{i-1} + (\Delta P_s)_i}{2} \right]. \quad (3.41)$$

In order to calculate the mean pressure per cell (denoted by P_a , since it is measured along the axis) we use equation (3.10) and is given by,

$$(P_a)_i = (P_s)_i \left(1 + \frac{1}{2}(\gamma - 1)M_i^2 \right)^{\frac{-\gamma}{\gamma-1}}. \quad (3.42)$$

As the grain burns, we also calculate the Mach number at each cell and this is obtained using the following equation,

$$M_i = \frac{(\dot{m}_a)_i}{(P_a)_i(A_a)_i} \sqrt{\frac{RT_f}{\gamma}} \left(1 + \frac{1}{2}(\gamma - 1)M_i^2 \right)^{-\frac{1}{2}}, \quad (3.43)$$

which is obtained from the rearrangement of equation (3.30), where T_f is the flame temperature. An equilibrium for the equations (3.36) to (3.43) must be obtained (*i.e* these equations must all be solved simultaneously). This not done, instead, these equations are solved iteratively until the Mach number (M_i) has converged.

3.8.3 Gas Dynamics in the Nozzle to Obtain Head Pressure for the Next Time Step

The surface pressure in the last cell is used to calculate the stagnation pressure in the nozzle, which is

$$P_{noz} = (P_s)_N + \frac{1}{2}(\Delta P_s)_N. \quad (3.44)$$

Using equation (3.32) the nozzle mass rate is,

$$\dot{m}_{noz} = \frac{P_{noz}A_t}{C^*}, \quad (3.45)$$

where A_t is the throat cross sectional area of the nozzle, and C^* is the nozzle characteristic velocity.

The total mass flow from the grain is,

$$\dot{m}_{tot} = \sum_{i=1}^N (\dot{m}_s)_i. \quad (3.46)$$

The condition of the equilibrium $\dot{m}_{noz} = \dot{m}_{tot}$, is not forced upon the solution. Instead, the time rate pressure change is calculated and used to determine the head pressure for the first cell which is used for the next time step as follows,

$$\frac{dP}{dt} = \frac{RT_f}{\nu_c} (\dot{m}_{tot} - \dot{m}_{noz}) - \frac{1}{2} \frac{P_1}{\rho} \frac{\dot{m}_{tot}}{\nu_c}, \quad (3.47)$$

where ν_c is the volume of the free space (the space inside the combustion chamber that is not occupied by the propellant grain where the gas flow). The head pressure in the chamber is then calculated using

$$P_1(t + \Delta t) = P_1(t) + \Delta t \frac{dP}{dt}. \quad (3.48)$$

The time interval is continuously readjusted depending on the chamber conditions and is given by,

$$\Delta t = \frac{1}{5} \tau, \quad (3.49)$$

where τ is a chamber filling time constant calculated from

$$\tau = \frac{P_{noz} \nu_c}{(1 - \bar{n})(RT_f \dot{m}_{noz})}. \quad (3.50)$$

The 1-D internal ballistics model described by Lamberty is complete. This model is used with the Level Set Method to model the grain propagation with time. The pressure of gas gives the burn rate velocity used by Level Set to evolve the interface. Therefore, as the interface evolves port and burning surface area are calculated which are then used by the internal ballistics to determine the pressure and burn rate of the next time step.

Chapter 4

Results

In this chapter, we will discuss some numerical results obtained using the level set method discussed in chapter 2 coupled with 1-D internal ballistics which is discussed in chapter 3. The grains that were investigated include tubular grain, star grain, slotted grain and anchor grain. For some grains, experimental data was available to validate the results that were obtained from numerical experiments but unfortunately for some grains it was not available because test firing of SRM's is expensive.

We shall model the SRM with the constants shown in Table 4.1:

Description	Symbol	Value	Units
Length of the SRM grain	L	1.4	m
Casing radius	R_{out}	0.05	m
Throat area	A_t	0.001257	m ²
Universal gas constant	R	333	J.K ⁻¹ .kg ⁻¹
Flame Temperature	T_f	2900	K
Characteristic Velocity	C^*	1500	m.s ⁻¹
Specific heat ratio	γ	1.2	-
Density	ρ	1650	kg.m ⁻³
Burning rate coefficient	c	4.5×10^{-5}	kg ⁻¹ .m ² .s
Burning rate exponent	\bar{n}	0.319	-

Table 4.1. Constants used for modelling the burnback in all examples

These constants are used for all the different grain geometries presented in this chapter.

In all examples, we shall model the process twice, first using the 2D-model for grain burnback described in section 2.10, and then using the 3D-model for grain burnback discussed in section 2.11.

4.1 Internal Burning Grain

The internal burning grain with the mandrel radius of 0.0315 m was investigated and the propagation of the surface with time is shown in Figure 4.1. Using the 1-D internal ballistics we have managed to produce the two curves plotted in blue (SDF values calculated in 3D) and green (SDF values calculated in 2D) shown in Figure 4.2.

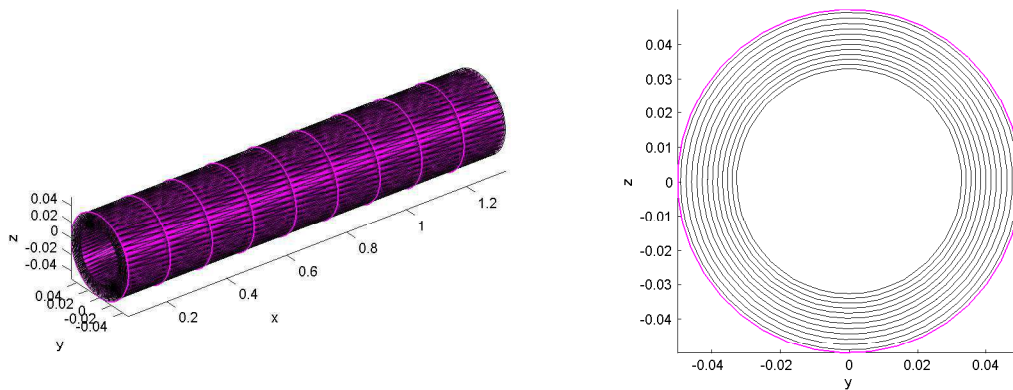


Figure 4.1. Evolution of a tubular grain

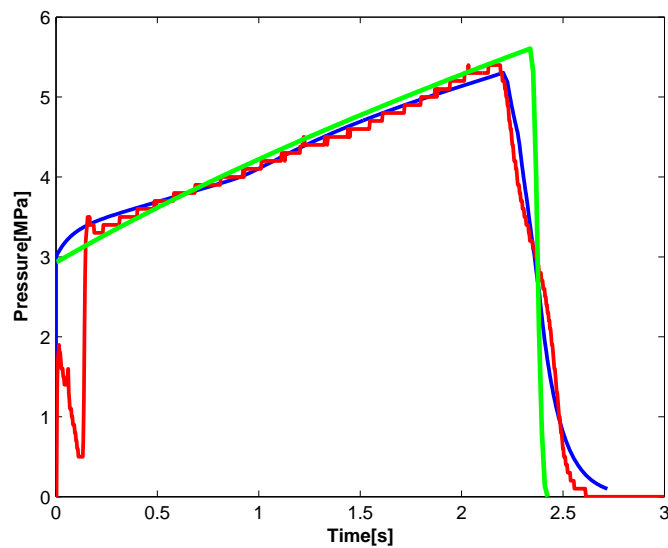


Figure 4.2. Pressure-Time curve for a tubular grain, red is obtained from experimental data, tubular grain modelled using 2D level sets (green) and 3D level sets (blue)

The two curves show a progressive pressure-time curve of the tubular grain. The progressive increase in the graphs is due to the increasing port and surface area of the cells as the grain evolves. These graphs give a good approximation of the red curve in Figure 4.2, which represents the experimental data provided.

As the grain burns, the pressure rises until it reaches 5.6 MPa. Then after 2.3 s, it burns out, this will cause the pressure to drop until the atmospheric pressure and pressure inside the combustion chamber are equal. The gas flow then stops because there is no longer any mass flow from the burning surface.

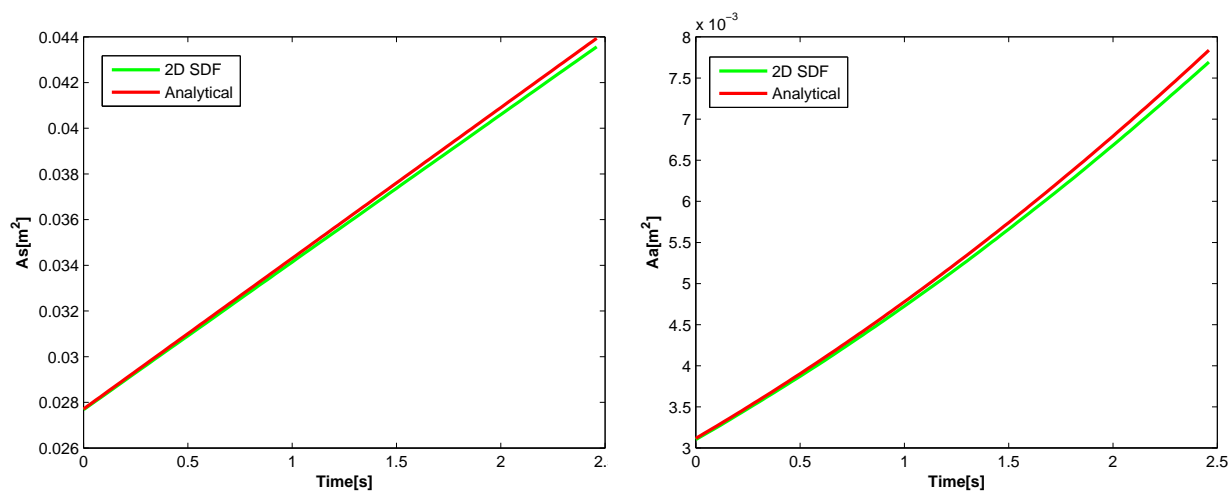


Figure 4.3. Burning surface area as a function of time (left) and Port area as a function of time (right) for a tubular grain

Figure 4.3 shows the comparison between the analytical method and the 2D level set for determining the burning surface area (A_s) and the port area (A_a). It is evident from the graphs the 2D model of the level sets gives a good approximation of the analytical solution. The burning surface increases linearly with time, while the port area increases quadratically with time for a tubular grain.

4.2 Star Grain

Using the star given in Hartfield et al. [14] defined by the following parameters $N = 6$, $R_p = 0.037$, $R_i = 0.023$ and $\epsilon = 0.7$ (see Figure 2.22). The polygons obtained when dividing the grain into cells (plotted in blue) and their approximations (plotted in red) are shown in Figure 4.4. The propagation of the star grain with time is shown in Figure 4.5, where the blue polygon shows the mandrel of the star grain.

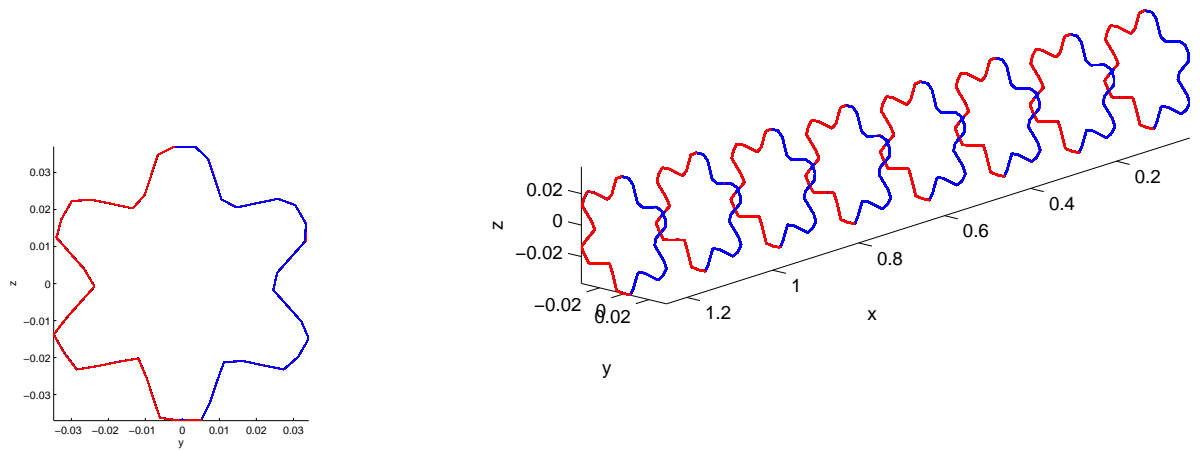


Figure 4.4. Polygons (blue) and their approximations (red) of the mandrel of a star grain

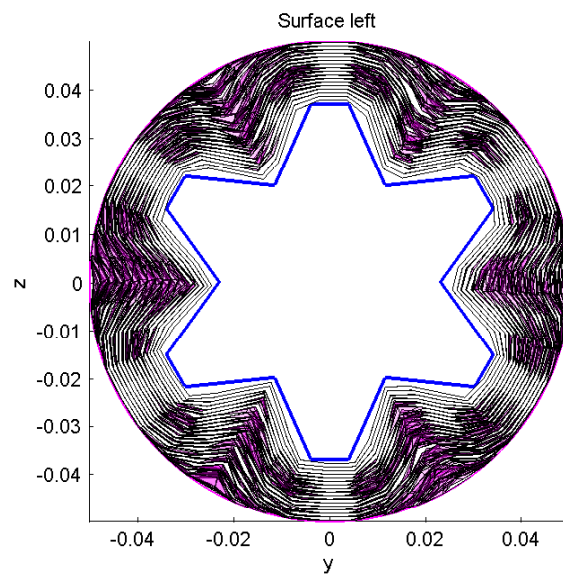


Figure 4.5. Evolution of a star grain

Figure 4.5 shows that as the star grain burns, there are slivers (pieces left) attached to the casing that burn for some time before the grain burns out. The pressure-time curve produced by the star grain is shown in Figure 4.6. We did not have experimental data to compare our results with.

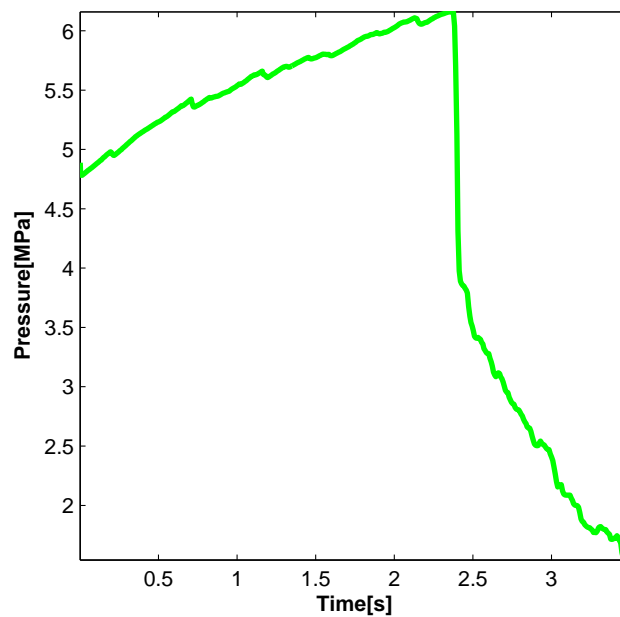


Figure 4.6. Pressure versus time graph of the star grain

From Figure 4.6, it is evident that since the star is given by a star shaped cylindrical tube, there is a rise in pressure at the start because of the increasing burning surface area and port area for each cell. After some time it hits the casing then the pressure starts to drop because the burning surface area for each cell decreases as it burns. The pressure drop is slow compared to the tubular grain, because in the final stage there are still bits of pieces that are still burning.

Figure 4.7 shows the comparison between analytical solution and 2D model of the level set method for the star grain geometry. The burning surface area (A_s) and port area (A_d) are plotted as a function of time. The graph on the left shows that the burning surface increases as the grain evolves, then when it reaches the casing it decreases because there are small slivers left which are still burning before the propellant burns out. The graph on the right shows that the port area increases quadratically with time until the burning front hits the casing, then the rate at which the port area increases decreases. The 2D model of the level set method gives a good approximation of the analytical results for the star grain.

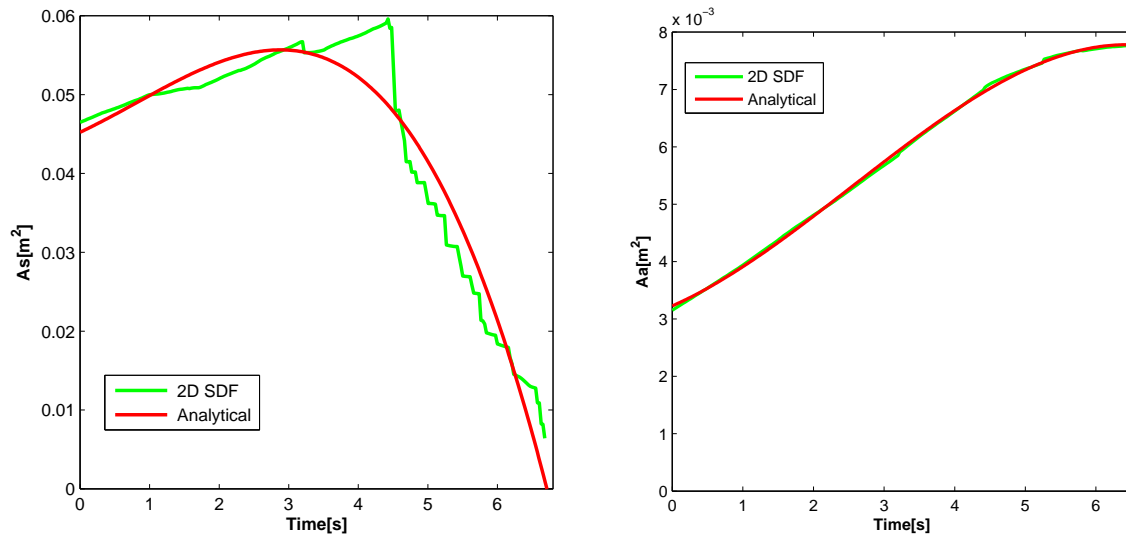


Figure 4.7. Burning surface area as a function of time (left) and Port area as a function of time (right) for a star grain

4.3 Slotted Grain

The slotted grain with the dimensions given in Table 4.2 was also investigated,

Description	Symbol	Value	Units
Length of the slot	l	0.445	m
Web length	β	0.0394	m
Web thickness	w	0.0185	m
Slot diameter	r	0.01	m
Port diameter	D_{port}	0.063	m
Number of slots	n	4	-

Table 4.2. Dimensions of the slotted propellant grain

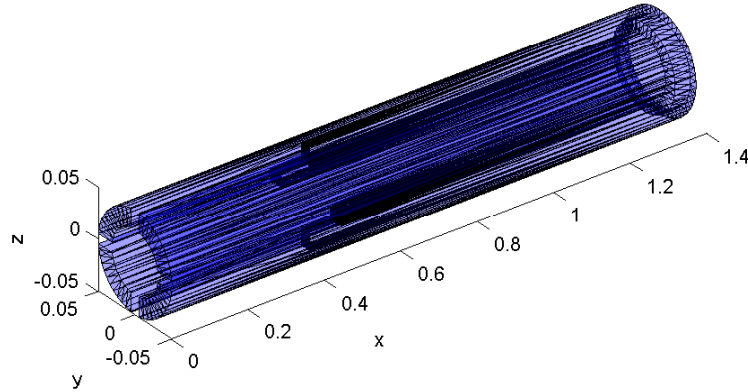


Figure 4.8. Slotted grain

Since the slotted grain (shown in Figure 4.8) has slots along its length, the mandrel with fins which are longer than the casing radius was created (shown in Figure 4.9) using 3D SDF values.

The mandrel was then divided into cells shown in Figure 4.10, left. Since the grain geometry of the mandrel has some sharp bends that are orthogonal to the axis of the grain at the end of the slot fins and at the back of the mandrel, the cells were made to be smaller in size in these places so as to reduce the error when modelling the propagation of the grain.

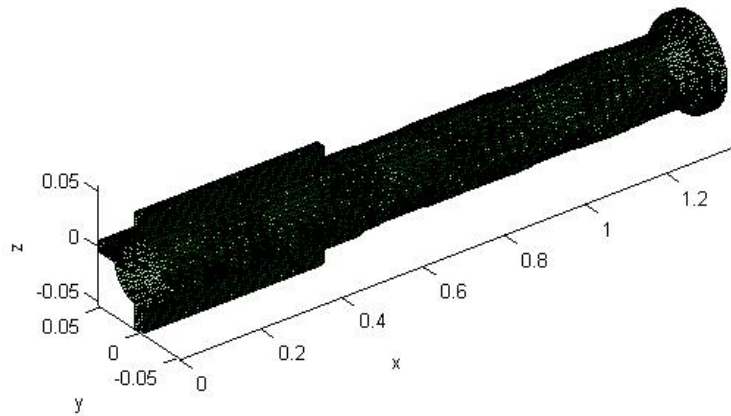


Figure 4.9. Mandrel of the Slotted grain

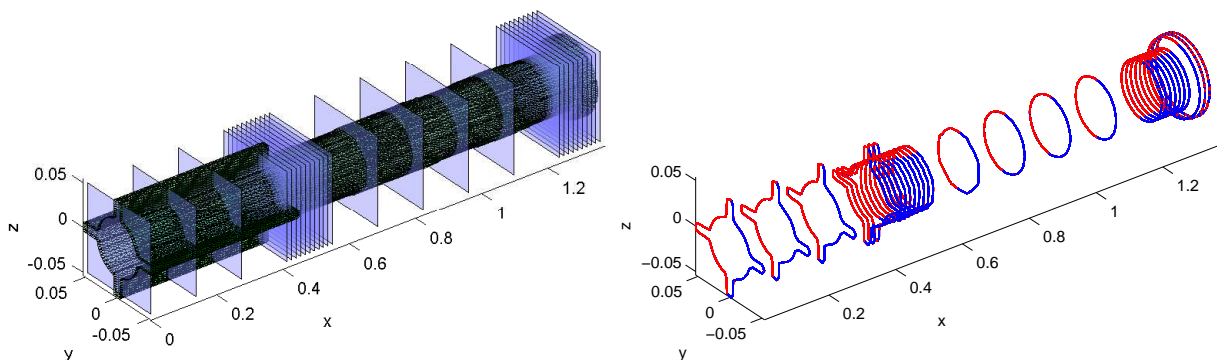


Figure 4.10. Slotted grain cells (left) and polygons obtained (right)

The polygons obtained and their approximations are plotted in Figure 4.10 on the right hand side. The mandrel approximation polygons plotted in red in Figure 4.10 were used to model the evolution of the slotted grain. The evolution of the slotted grain with time is shown starting from Figure 4.11 to Figure 4.15.

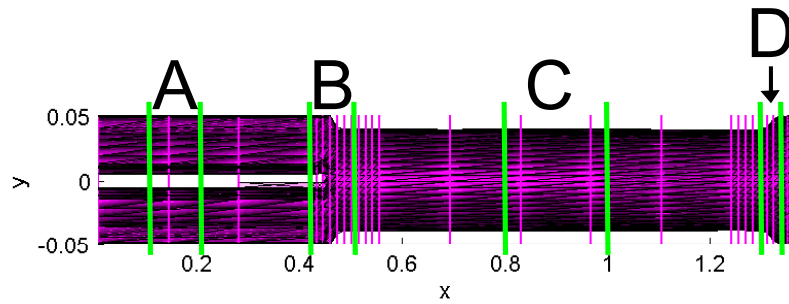


Figure 4.11. Slotted grain burned for some time

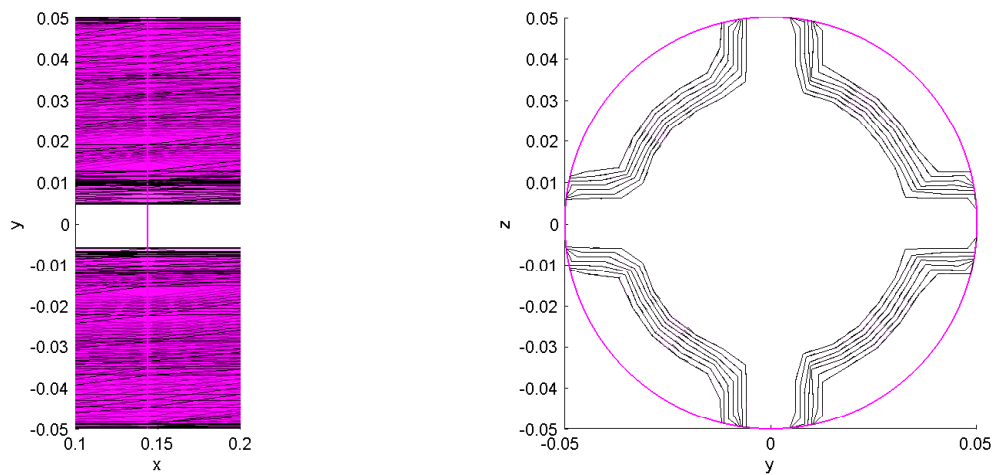


Figure 4.12. Evolution of part A

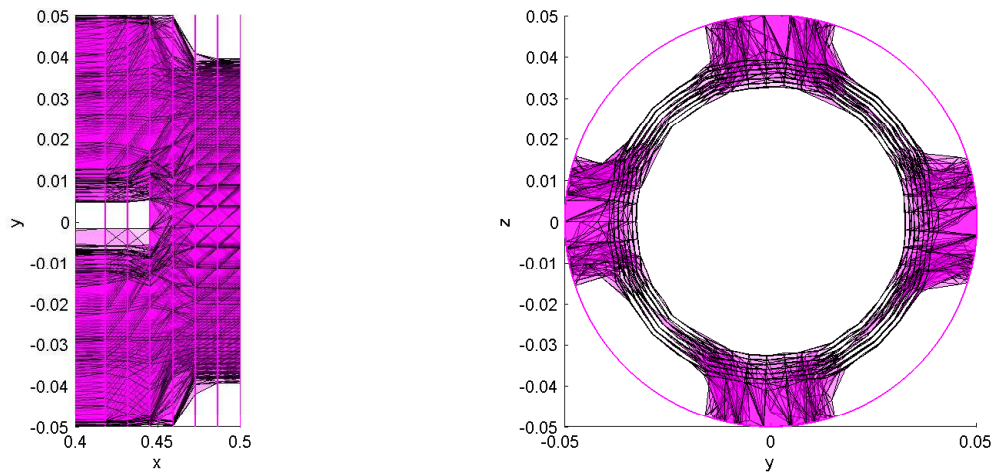


Figure 4.13. Evolution of part B

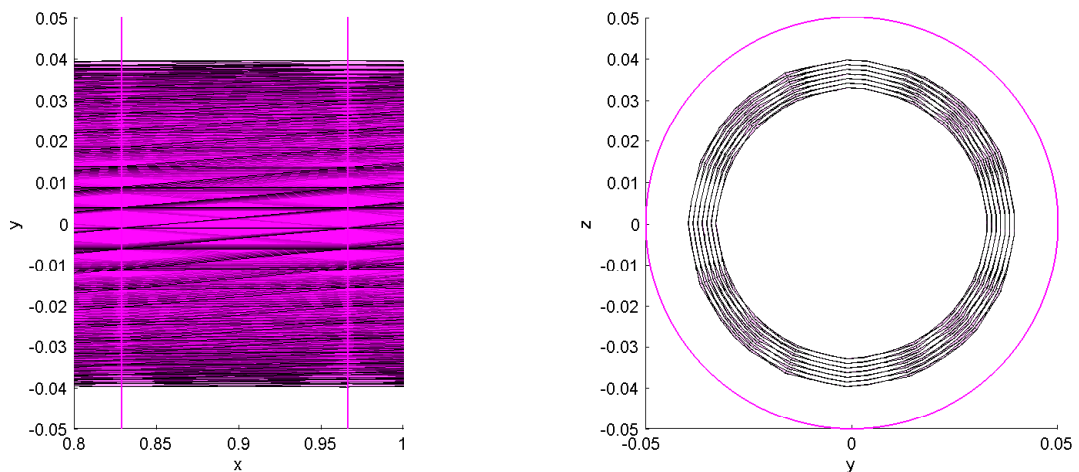


Figure 4.14. Evolution of part C

The pressure time curves for the slotted grain propellant are shown in Figure 4.16. The red graph is plotted from the experimental data, while the green curve is obtained using our 2D model. It can be seen from the graph a slotted grain propellant produces a fairly constant pressure-time curve. The initial gap between the graphs is due to erosive burning not being included in the model.

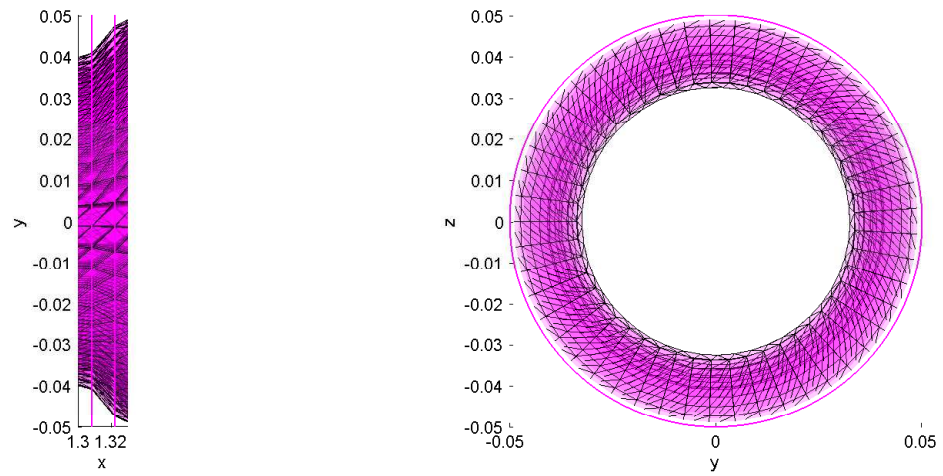


Figure 4.15. Evolution of part D

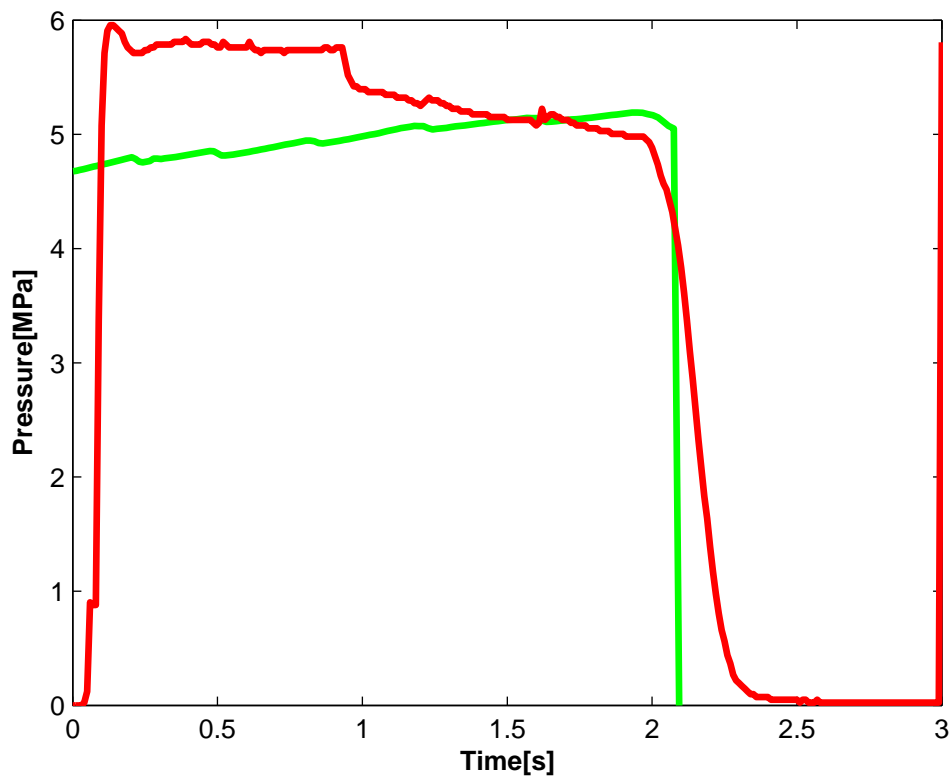


Figure 4.16. Pressure versus time graph of the slotted grain, red is the experimental data, green comes from the 2D model for burnback

4.4 Anchor Grain

The double anchor grain with the mandrel defined by the parameters (shown in chapter 1, Figure 1.6) listed in Table 4.3, was investigated. This mandrel is then divided into a number of cells shown in Figure 4.17, on the left hand side. Where the plane intersects with the grain mandrel polygons on the boundaries of the cell were obtained, shown in blue in Figure 4.17, on the right hand side. The SDF values of the interfaces given by the blue polygons are then calculated. The polygons that approximates the original polygons are then determined, shown in red in Figure 4.17 on the right hand side.

Description	Symbol	Value	Units
Radius of the inner circular region	r_1	0.015	m
Inner radius of the spoke	r_2	0.03	m
Outer radius of the spoke	r_3	0.043	m
Radius of the circular end of the spoke	r_4	0.0065	m
Width of the part joining the circular region and the spoke	δs	0.006	m

Table 4.3. Values of the parameters of the double anchor grain

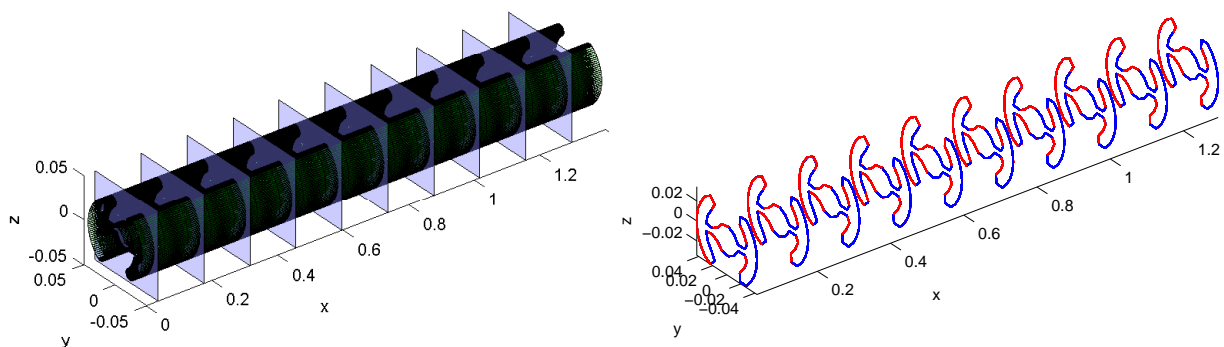


Figure 4.17. The mandrel of anchor divided into cells (left) and polygons obtained from the intersection of the plane and the mandrel (blue) and their approximations (red) (right).

The evolution of the double anchor grain is shown in Figure 4.18.

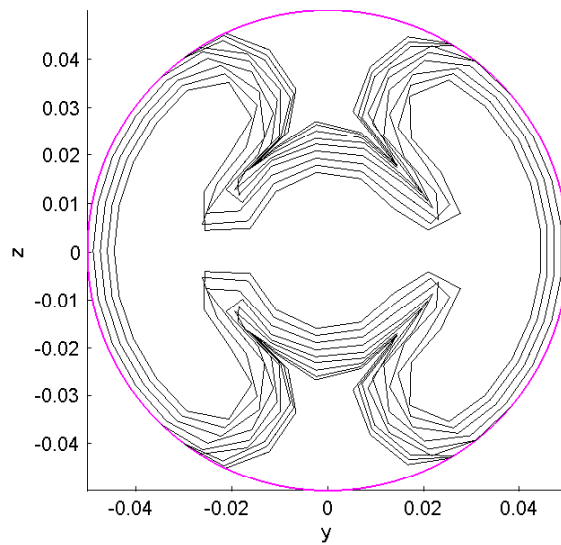


Figure 4.18. The anchor grain evolution

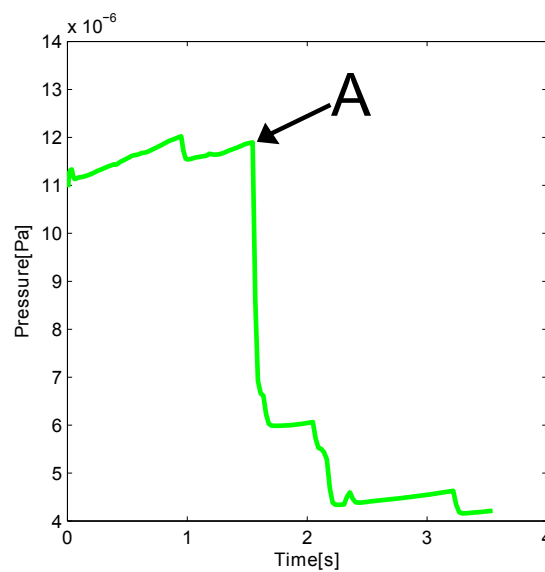


Figure 4.19. Pressure versus time graph of the anchor grain

Figure 4.19 shows the pressure-time curve of the anchor grain. At first the grain burns progressively and it reaches point A, at point A the anchor spokes reaches the casing (see Figure 4.20), then it starts to burn regressively until the grains burn out.

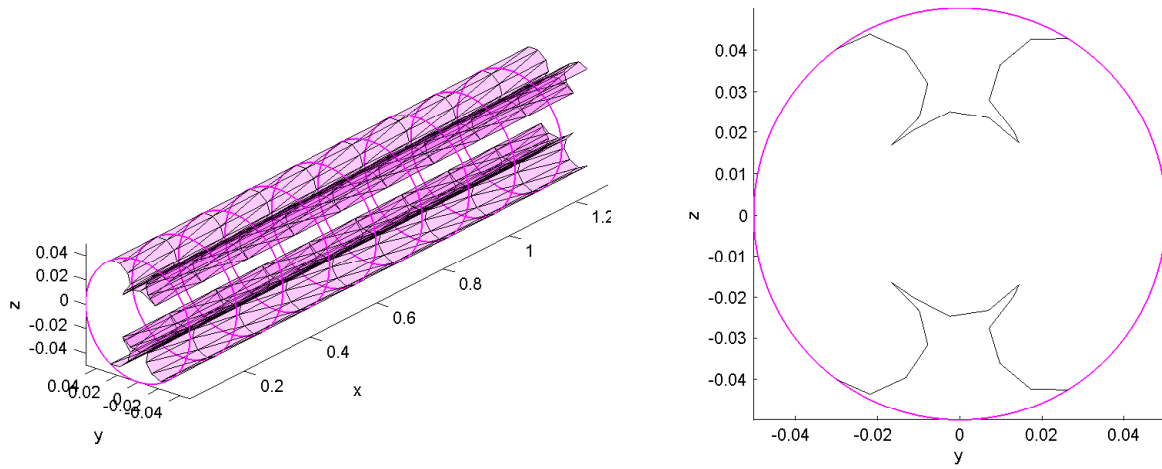


Figure 4.20. Burning anchor grain at Point A

As the anchor grain evolves the spokes meet and form slivers that are attached to the casing and the ones that are detached from the casing, this makes it difficult for zero contour to obtain a closed contour. This gives rise to instability in the solution.

Chapter 5

Conclusions

The purpose of this project was to develop a model used to simulate the propagation of the solid propellant grain inside the solid rocket motor. In order to simulate this propagation 1-D internal ballistics needs to be evaluated.

A model that is used for the evolution of the burning surface has been developed and presented. This method uses a signed distance function to represent the burning surface, and the level set method is used for the surface propagation. This method gives satisfying results when modelling the cusps and corners as the surface evolves. The 1-D internal ballistics model from Lamberty [21] was implemented in order to model the flow of gases inside the solid rocket motor.

Four grain geometries were investigated and the results found using the model match those obtained from available experimental data.

The `Zero Contour` function that was written to replace `contourc` of MATLAB was successful in obtaining the contour at level zero, and the drawbacks of `contourc` discussed in chapter 2 section 2.10 were dealt with.

The 2D model for surface burnback was validated by comparing its results with those of analytical solutions. We observed that the model gives good results when calculating the burning surface area and port area of the surface as the front evolves, therefore this method can be recommended for calculation burning surface area and port area of an evolving front. The way signed distance function is calculated takes less computational time compared to the methods used before in literature.

When investigating the burnback of the anchor grain some numerical instabilities arises. These numerical instabilities are formed because the anchor spokes evolve until they meet giving rise to sliver that is attached to the casing and the ones that are detached from the

casing, then zero contour finds it difficult to deal with these islands. The detached and attached slivers are also mentioned by Umbel [36].

In reality SRM's turn to burn faster near the nozzle than at the front, this is due to erosive burning. Erosive burning is caused by high velocity gases that move parallel to the burning surface, and this tends to increase the burning rate. The model presented here does not include the erosive burning, therefore, for future work this phenomenon should be investigated and included.

The internal ballistic model presented in this project is 1-D, therefore other models such as 2D-symmetric and 3D internal ballistic models should be included in the model.

In this project, six star shaped grain, four slotted grain, and double anchor grains were investigated. These grain geometries can be generalised to have different number slots and spokes [36], therefore this can be investigated and see how the thrust-time curve and evolution of the grain changes.

Bibliography

- [1] R Abgrall. Numerical discretization of the first-order hamilton-jacobi equation on triangular meshes. *Communications on pure and applied mathematics*, 49(12):1339–1373, 1996.
- [2] S Chen, B Merriman, S Osher, and P Smereka. A simple level set method for solving stefan problems. *Journal of Computational Physics*, 135(1):8–29, 1997.
- [3] A.J Chorin. Flame advection and propagation algorithms. *Journal of Computational Physics*, 35(1):1–11, 1980.
- [4] F Chorlton. *Textbook of fluid dynamics*. Van Nostrand, 1967.
- [5] A Davenas. *Solid rocket propulsion technology*. Newnes, 2012.
- [6] H.W Douglass, J.H Collins, W.T Brooks, and R.B Keller. Solid propellant grain design and internal ballistics, space vehicle design criteria. *NASA-sp8076*, 1972.
- [7] S.S Dunn and D.E Coats. 3-D grain design and ballistic analysis using spp97 code. *AIAA paper*, 1997.
- [8] A Espagnacq, N Forichon-Chaumet, and T Rodriguez. Multilaye pyrotechnic materials comprising boron, potassium nitrate, nitrocellulose, aluminium, copper oxide, April 9 2002. US Patent 6,368,434.
- [9] V.I Feodosiev and G.B Siniarev. *Introduction to rocket technology*. Academic Press, 2014.
- [10] J.D Foley, A Van Dam, S.K Feiner, J.F Hughes, and R.L Phillips. *Introduction to computer graphics*, volume 55. Addison-Wesley Reading, 1994.
- [11] M.F Gomez. *Internal Ballistics Simulation of a Solid Rocket Motor*. PhD thesis, NIVERSIDADE DA BEIRA INTERIOR, 2013.
- [12] A Harten. High resolution schemes for hyperbolic conservation laws. *Journal of computational physics*, 49(3):357–393, 1983.

-
- [13] A Harten and S Osher. Uniformly high-order accurate nonoscillatory schemes. i. *SIAM Journal on Numerical Analysis*, 24(2):279–309, 1987.
- [14] R Hartfield, R Jenkins, J Burkhalter, and W Foster. A review of analytical methods for solid rocket motor grain analysis. In *39th Joint Propulsion Conference and Exhibit*, 2003.
- [15] C.W Hirt and B.D Nichols. A computational method for free surface hydrodynamics. *Journal of pressure vessel technology*, 103(2):136–141, 1981.
- [16] C.W Hirt and B.D Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of computational physics*, 39(1):201–225, 1981.
- [17] G Hussain and GJ Rees. Combustion of black powder. part 1: Thermo-analytical studies. *Propellants, Explosives, Pyrotechnics*, 15(2):43–47, 1990.
- [18] J.M Hyman. Numerical methods for tracking interfaces. *Physica D: Nonlinear Phenomena*, 12(1):396–407, 1984.
- [19] J.M Hyman. Numerical methods for tracking interfaces. *Physica D: Nonlinear Phenomena*, 12(1):396–407, 1984.
- [20] B.V Kit and D.S Evered. *Rocket propellant handbook*. Macmillan, 1960.
- [21] J.T Lamberty. A report on the grain design and internal ballistics module of the improved solids performance program. In *AIAA 19th Aerospace Sciences Meeting, AIAA Paper*, volume 34, 1981.
- [22] J Li, Z Xin, and H Yin. Transonic shocks for the full compressible euler system in a general two-dimensional de laval nozzle. *Archive for Rational Mechanics and Analysis*, 207(2):533–581, 2013.
- [23] P.A Lorente. Study of grain burnback and performance of solid rockets motors. Master’s thesis, Universitat Politècnica de Catalunya, 2013.
- [24] R Malladi, J.A Sethian, and B.C Vemuri. Shape modeling with front propagation: A level set approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(2):158–175, 1995.
- [25] B.D Nichols and C.W Hirt. Methods for calculating multi-dimensional, transient, free surface flows past bodies. In *Proc., 1st Int. Conf. Ship Hydrodynamics*, pages 253–277. DTIC Document, 1975.
- [26] W.F Noh and P Woodward. Slic (simple line interface calculation). In *Proceedings of the Fifth International Conference on Numerical M Methods in Fluid Dynamics June 28–July 2, 1976 Twente University, Enschede*, pages 330–340. Springer, 1976.

-
- [27] S Osher and R Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer Science & Business Media, 2006.
- [28] S Osher and J.A Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.
- [29] K.O Reddy and K.M Pandey. Burnback analysis of 3-D star grain solid propellant. *International Journal of Advanced Trends in Computer Science and Engineering*, 2(1):215–223, 2013.
- [30] J.A Sethian. Curvature and the evolution of fronts. *Communications in Mathematical Physics*, 101(4):487–499, 1985.
- [31] J.A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.
- [32] J.A Sethian et al. Numerical algorithms for propagating interfaces: Hamilton-jacobi equations and conservation laws. *Journal of differential geometry*, 31(1):131–161, 1990.
- [33] J.A Sethian and A Vladimirovsky. Fast methods for the eikonal and related hamilton-jacobi equations on unstructured meshes. *Proceedings of the National Academy of Sciences*, 97(11):5699–5703, 2000.
- [34] M.W Stone. Slotted tube grain design. *ARS Journal*, 31(2):223–228, 1961.
- [35] P.G Sutton. *Rocket Propulsion Elements-An Introduction to the Engineering of Rockets*. John Wiley and Sons, 1986.
- [36] M.R Umbel. An exact geometric analysis of the generalized anchor grain configuration. In *44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, page 4697, 2008.
- [37] M.A Willcox, M.Q Brewster, K.C Tang, and D.S Stewart. Solid propellant grain design and burnback simulation using a minimum distance function. *Journal of propulsion and power*, 23(2):465–475, 2007.
- [38] M.J Zucrow and J.D Hoffman. *Gas dynamics*, volume 1. John Wiley and Sons, 1976.

Appendix A

Projection Geometry

In order to calculate the signed distance function, ϕ , a method of the shortest distance between the grid point and the interface is required. Here is a 2D and 3D method used to find the shortest distance between a grid point and the interface (or surface).

A.1 The Calculation of the Signed Distance Function in 2D Using Projection Geometry

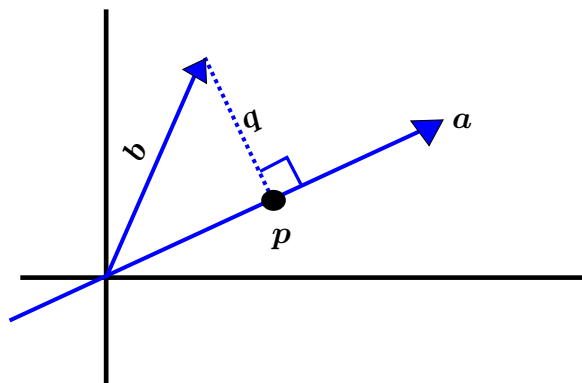


Figure A.1. The point closest to \mathbf{b} on the line determined by \mathbf{a}

If we have a vector \mathbf{b} and a line determined by a vector \mathbf{a} , then we can find a point on the line that is closest to \mathbf{b} . We can see from the Figure A.1 that this closest point \mathbf{p} is at the intersection formed by a line through \mathbf{b} that is orthogonal to \mathbf{a} . If we think of \mathbf{p} as an approximation of \mathbf{b} , then the length of \mathbf{q} is the error in that approximation.

Since \mathbf{p} lies on the line through \mathbf{a} , we know $\mathbf{p} = x\mathbf{a}$ for some number x . We also know that \mathbf{a} is perpendicular to $\mathbf{q} = \mathbf{b} - x\mathbf{a}$. Therefore,

$$\begin{aligned}\mathbf{a}^T(\mathbf{b} - x\mathbf{a}) &= 0 \\ x\mathbf{a}^T\mathbf{a} &= \mathbf{a}^T\mathbf{b} \\ x &= \frac{\mathbf{a}^T\mathbf{b}}{\mathbf{a}^T\mathbf{a}},\end{aligned}$$

and $\mathbf{p} = x\mathbf{a} = \mathbf{a}\frac{\mathbf{a}^T\mathbf{b}}{\mathbf{a}^T\mathbf{a}}$.

Now we would like to write this projection in term of a *projection matrix* P : $\mathbf{p} = P\mathbf{b}$.

$$\mathbf{p} = x\mathbf{a} = \frac{\mathbf{a}\mathbf{a}^T\mathbf{b}}{\mathbf{a}^T\mathbf{a}},$$

therefore, the matrix is:

$$P = \frac{\mathbf{a}\mathbf{a}^T}{\mathbf{a}^T\mathbf{a}}.$$

Note that $\mathbf{a}\mathbf{a}^T$ is a two by two matrix and not a number, and matrix multiplication is not commutative. The column space of P is spanned by \mathbf{a} because for any \mathbf{b} , $P\mathbf{b}$ lies on the line determined by \mathbf{a} . The rank of the projection matrix P is 1. Also we need to notice that P is symmetric. Furthermore $P^2\mathbf{b} = P\mathbf{b}$ because the projection of a vector already on the line through \mathbf{a} is just that same vector. In general the projection matrix has the following properties:

1. $P^T = P$ because P is symmetric.

2. $P^2 = P$

Proof :

$$P^2 = P \cdot P = \left(\frac{\mathbf{a}\mathbf{a}^T}{\mathbf{a}^T\mathbf{a}}\right) \left(\frac{\mathbf{a}\mathbf{a}^T}{\mathbf{a}^T\mathbf{a}}\right) = \frac{\mathbf{a}\mathbf{a}^T\mathbf{a}\mathbf{a}^T}{\mathbf{a}^T\mathbf{a}\mathbf{a}^T\mathbf{a}} = \frac{\mathbf{a}\mathbf{a}^T}{\mathbf{a}^T\mathbf{a}} = P$$

3. P is independent of the length of \mathbf{a} .

Proof :

Let $\mathbf{a} = \lambda\mathbf{a}$ where λ is a scalar multiple of the vector \mathbf{a} . Then,

$$P = \frac{(\lambda\mathbf{a})(\lambda\mathbf{a})^T}{(\lambda\mathbf{a})^T(\lambda\mathbf{a})} = \frac{\lambda\mathbf{a}\lambda\mathbf{a}^T}{\lambda\mathbf{a}^T\lambda\mathbf{a}} = \frac{\lambda^2\mathbf{a}\mathbf{a}^T}{\lambda^2\mathbf{a}^T\mathbf{a}} = \frac{\mathbf{a}\mathbf{a}^T}{\mathbf{a}^T\mathbf{a}} = P$$

4. $P' = I - P$ is also projection matrix

Proof :

$$\begin{aligned}(P')^2 &= (I - P)(I - P) = I - P - P + P^2 \\ &= I - 2P + P \\ &= I - P = P'.\end{aligned}$$

Therefore, P' projects on orthogonal compliment of space spanned by \mathbf{a} ,

$$\begin{aligned}\mathbf{q} &= \mathbf{b} - \mathbf{p} \\ &= \mathbf{b} - P\mathbf{b} \\ &= (I - P)\mathbf{b} \\ &= P'\mathbf{b}.\end{aligned}$$

A.2 The Calculation of the Signed Distance Function in 3D Using Projection Geometry

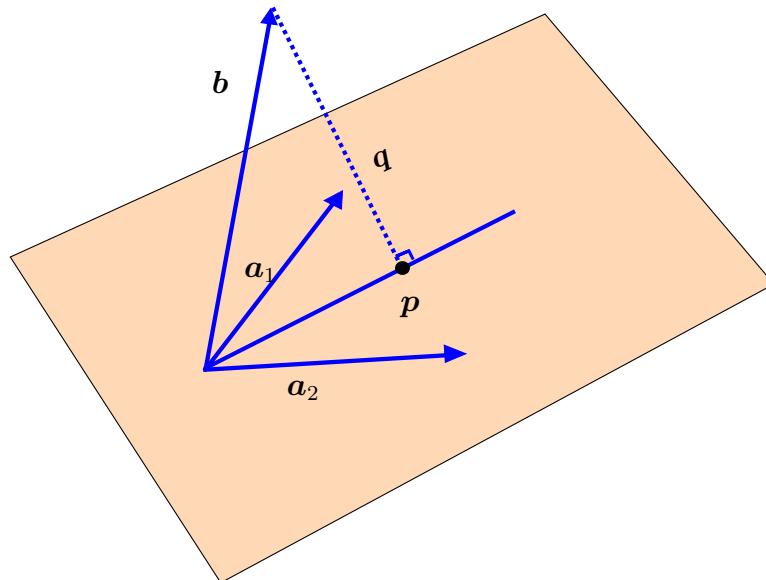


Figure A.2. The point closest to \mathbf{b} on the plane determined by \mathbf{a}_1 and \mathbf{a}_2

We want now to project a vector \mathbf{b} onto the closest point \mathbf{p} in a plane in \mathcal{R}^3 . If we have vectors \mathbf{a}_1 and \mathbf{a}_2 that form a basis for the plane, then the plane is the column space of

the matrix $A = [\mathbf{a}_1 \quad \mathbf{a}_2]$.

Then $\mathbf{p} = \hat{x}_1 \mathbf{a}_1 + \hat{x}_2 \mathbf{a}_2 = A\hat{\mathbf{x}}$ for some scalars \hat{x}_1 and \hat{x}_2 , and we want to find $\hat{\mathbf{x}}$. When using Figure A.2 the vector \mathbf{q} can be written in the form

$$\mathbf{q} = \mathbf{b} - \mathbf{p} = \mathbf{b} - A\hat{\mathbf{x}},$$

and is orthogonal to the plane we are projecting onto, after which we can use the fact that \mathbf{q} is perpendicular to \mathbf{a}_1 and \mathbf{a}_2 :

$$\mathbf{a}_1^T(\mathbf{b} - A\hat{\mathbf{x}}) = 0 \quad \text{and} \quad \mathbf{a}_2^T(\mathbf{b} - A\hat{\mathbf{x}}) = 0.$$

In matrix form, $A^T(\mathbf{b} - A\hat{\mathbf{x}}) = \mathbf{0}$. Notice that $\mathbf{q} = \mathbf{b} - A\hat{\mathbf{x}}$ is in the left nullspace of A .

Therefore, we can rewrite the equation $A^T(\mathbf{b} - A\hat{\mathbf{x}}) = \mathbf{0}$ as:

$$A^T A \hat{\mathbf{x}} = A^T \mathbf{b}.$$

When we were projecting onto a line $A^T A$ was just a number, now it is a square matrix. Therefore instead of dividing by $\mathbf{a}^T \mathbf{a}$ we now have to multiply by $(A^T A)^{-1}$.

Hence in n dimensions,

$$\begin{aligned} \hat{\mathbf{x}} &= (A^T A)^{-1} A^T \mathbf{b}, \\ \mathbf{p} &= A\hat{\mathbf{x}} = A(A^T A)^{-1} A^T \mathbf{b}, \\ P &= A(A^T A)^{-1} A^T. \end{aligned}$$

The matrix P has the same properties (i.e. symmetric, idempotent, independent of length \mathbf{a}_1 and \mathbf{a}_2 , etc) as P before.

Appendix B

Zero Contour Code

```

function ZC=ZeroContour(X,Y,U,y0)
%X,Y are grid coordinates
%U are the value of phi
%y0 is starting position of y
S=U>0;
y=find(Y(:,1)>y0,1);
x=find(S(y,:)>0,1);
SHOW=0;
jj=3;

si=size(S);
Ny=si(1); Nx=si(2);
if SHOW==1,
    figure(200); hax=axes; image(S*100); colormap gray(256); axis image; hold on
    set(hax,'YDir','normal');
end
PP=[0 1 0 -1; -1 0 1 0];
ZC=[];
p=[y; x];
p0=p;
if FirstPosition(S,p)==0
    E=p([2 1])' - [1 1];
    E=[E; E];
    return
end

E=p0([2 1])' - [1 1];

% ==== do first point ====
[p,jj]=NextPosition(S,p,jj);
% ==== iterate until loop is closed ====
while ~(p(1)==p0(1) & p(2)==p0(2))
    pp0=p;

```

```

    if SHOW==1, plot(pp0(2),pp0(1),'sb','Markersize',3); end
    XY=SInterpContour(X,Y,U,S,pp0,jj,SHOW);
    [p,jj]=NextPosition(S,p,jj);

    ZC=[ZC; XY];
end
% ==== do last point ====
pp0=p;
XY=SInterpContour(X,Y,U,S,pp0,jj,SHOW);
[p,jj]=NextPosition(S,p,jj);
ZC=[ZC; XY];

% ==== plot result if SHOW==1 ====
if SHOW==1
    figure(201);
    contour(X(1,:),Y(:,1),U,[0 1]);
    hold on;
    plot(ZC(:,1),ZC(:,2),'b')
end

% =====
function [p,jj]=NextPosition(S,p,jj)
jj0=jj;
NN=[0 1 0 -1; 1 0 -1 0];
jj=mod(jj-1,4);
n=NN(:,jj+1);
q=p+n;
for j=1:4
    if S(q(1),q(2))==0
        jj=mod(jj+1,4);
        n=NN(:,jj+1);
        q=p+n;
    end
end
p=p+n;
% =====
function D=FirstPosition(S,p)
NN=[0 1 0 -1; 1 0 -1 0];
D=0;
for j=1:4
    q=p+NN(:,j);
    if S(q(1),q(2))==1, D=1; end
end

% =====
function XY=SInterpContour(X,Y,U,S,p,jj,SHOW);

hx=X(1,2)-X(1,1);
hy=Y(2,1)-Y(1,1);
NN=[0 1 0 -1; 1 0 -1 0];

```

```
if jj==0; sp=[3 0 1]; end
if jj==1; sp=[0 1 2]; end
if jj==2; sp=[1 2 3]; end
if jj==3; sp=[2 3 0]; end

XY=[]; j=0;
n=NN(:, sp(1)+1);
q=p+n;

while S(q(1),q(2))==0,
    u0=U(p(1),p(2)); u1=U(q(1),q(2));
    s=-u0/(u1-u0);
    XY =[XY; X(p(1),p(2))+n(2)*s*hx, ...
        Y(p(1),p(2))+n(1)*s*hy];
    if SHOW==1
        plot([p(2),q(2)], [p(1),q(1)], 'r');
        plot(p(2)+n(2)*s,p(1)+n(1)*s, 'w.', 'Markersize',5);
        pause
    end
    j=j+1;
    if j==4, break; end
    n=NN(:, sp(j)+1);
    q=p+n;
end
```