

VALIDATION OF THE
KURGANOV-TADMOR/KURGANOV-NOELLE-PETROVA
SCHEME FOR ROTATING DETONATION ENGINE
SIMULATIONS USING OPENFOAM

CHIBUIKEM UCHENNA AJAERO

A THESIS
IN
THE DEPARTMENT
OF
MECHANICAL, INDUSTRIAL AND AEROSPACE ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF APPLIED SCIENCE
CONCORDIA UNIVERSITY
MONTRÉAL, QUÉBEC, CANADA

JANUARY 2019

© CHIBUIKEM UCHENNA AJAERO, 2019

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: **Chibuikem Uchenna Ajaero**
Entitled: **Validation of the Kurganov-Tadmor/Kurganov-Noelle-
Petrova Scheme for Rotating Detonation Engine Simu-
lations using OpenFOAM**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Mehdi Hojjati	_____	Chair
Dr. Hoi Dick Ng	_____	Examiner
Dr. Xiaolei Wang	_____	Examiner
Dr. Charles Basenga Kiyanda	_____	Supervisor

Approved _____
Chair of Department or Graduate Program Director

_____ 20 _____

Dr. Amir Asif, Ph.D, PEng Dean
Gina Cody School of Engineering and Computer Science

Abstract

Validation of the Kurganov-Tadmor/Kurganov-Noelle-Petrova Scheme for Rotating Detonation Engine Simulations using OpenFOAM

Chibuikem Uchenna Ajaero

Detonation waves are a challenging field of study given the short time and length scales involved in the phenomenon. Such waves exhibit a complex structure consisting of a lead shock and shock waves travelling transversely to a detonation's normal propagation direction. The interaction between the shocks and the rapid chemical reactions they trigger results in the emergence of a natural length scale, the detonation cell size. There are still no complete theory or model that can accurately predict the cell size or a detonation waves initiation, propagation and failure dynamics. The numerical simulation of detonation waves is also challenging, due to the rapid reaction rates encountered. The Open source Field Operation And Manipulation (OpenFOAM) framework, commonly referred to as OpenFOAM, Computational Fluid Mechanics (CFD) package is increasingly used and referenced. One drawback of the stock OpenFOAM package is that the only finite volume numerical scheme available for the solution of the Euler equations in conservative form is the Kurganov-Tadmor (KT)/Kurganov-Noelle-Petrova (KNP) numerical scheme. Moreover, combustion is not implemented, hence which needs to be modified to simulate detonation waves; a coupling of compressible flows and reaction. This particular scheme is 2nd order accurate in smooth region based on the idea behind the Lax-Friedrichs scheme and which does not involve the solution to a Riemann problem in order to evaluate the intercell fluxes. This is unlike the methods currently used in detonation research, which nearly always consist of Godunov-type schemes with an approximate Riemann Solver such as Harten-Lax-van Leer-Contact (HLLC). OpenFOAM, with the KNP scheme, was recently used to simulate the two-dimensional structure of detonation waves despite having not been fully validated for the detonation simulation. Efforts to get access to the codes used proved abortive. In this work, we create a custom solver named `rhoCentralFoamreac` which we used to evaluate (validate) the KNP scheme for detonation cases, by simulating a standard 1D detonation case that usually results in pulsating propagation with a single mode. Metrics for detailed comparison and convergence studies are the oscillation peak pressure and period. Using the KNP scheme, we then examine OpenFoam as a CFD tool for the simulation of a detonation

based engine, where an initiated wave propagates circumferentially in a combustion chamber, commonly referred to as a rotating detonation engine. We study the effect of different ignition methods, and initiation flow fields (subsonic and supersonic) on the formation of these rotating detonation waves.

Acknowledgments

I cannot thank my parents enough for the role you have constantly played in my life, for even making this adventure possible in the first place. I owe you a world of gratitude, but that in itself is not enough. Thank you for support and for always being there. I wish to thank my supervisor, Professor Charles Kiyanda, for his immense assistance and guidance during the course of this work. He kept the door open every time I needed to walk through it for clarifications. Considering that I did use the door a lot, I just want to say thank you. Finally, thank you Lord for being my strength.

To a most dear friend, Emmanuel Ugbaja. You will forever be in my heart.

Contents

List of Figures	x
List of Tables	xiv
1 INTRODUCTION	3
1.1 DETONATION	3
1.1.1 PULSE DETONATION ENGINE	4
1.1.2 STANDING DETONATION WAVE ENGINE	5
1.1.3 ROTATING DETONATION ENGINE	5
1.2 DETONATION AND DEFLAGRATION	6
1.2.1 WHAT IS A DEFLAGRATION WAVE	6
1.2.2 WHAT IS A DETONATION WAVE	7
1.2.3 STRUCTURE OF A DETONATION WAVE	7
1.2.4 ESSENTIAL GEOMETRIC RELATIONS	8
1.2.5 THERMAL EFFICIENCY OF A DETONATION WAVE	9
1.2.6 ONE-DIMENSIONAL DETONATION DYNAMICS	10
1.3 THESIS OVERVIEW	12
2 THE KURGANOV-TADMOR/KURGANOV-NOELLE-PETROVA SCHEME AND OpenFOAM	14
2.1 REVIEW OF NUMERICAL DISCRETIZATION	14
2.2 GOVERNING EQUATIONS	15
2.2.1 CONSERVATION LAWS IN INTEGRAL FORM	16
2.3 TYPES OF NUMERICAL SCHEMES	17
2.3.1 GODUNOV'S METHOD (FIRST-ORDER UPWIND METHOD)	17
2.3.2 MUSCL-HANCOCK APPROACH FOR HIGHER RESOLUTION	17
2.4 CENTERED SCHEMES (LAX-FRIEDRICHS FAMILY OF SCHEMES	18

2.4.1	LAX-FRIEDRICHS SCHEME	18
2.4.2	NESSYAHU-TADMOR (NT) SCHEME	19
2.4.3	DESCRIPTION OF THE KNP NUMERICAL SCHEME	20
2.5	OVERVIEW OF OpenFOAM	23
2.5.1	WHAT IS OpenFOAM	23
2.5.2	FILE STRUCTURE OF OpenFOAM	24
2.5.3	POST PROCESSING	27
2.6	CREATION OF CUSTOM SOLVER FOR DETONATION SIMULATIONS (rhoCentralFoamreac)	27
3	VALIDATION OF THE KT/KNP SCHEME FOR DETONATION SIMULATION	30
3.1	NON-REACTIVE VALIDATION	31
3.1.1	CASE DESCRIPTION AND SET-UP	31
3.1.2	BOUNDARY AND INITIAL CONDITIONS	31
3.1.3	RESULTS	32
3.2	REACTIVE VALIDATION	37
3.2.1	CASE SETUP DESCRIPTION	37
3.2.2	INITIAL CONDITIONS	38
3.2.3	SOLUTION METHOD	39
3.3	RESULTS	40
3.4	ANALYSIS AND CONCLUSION	40
4	SIMULATION OF COLD ENGINE START	46
4.1	PROBLEM DESCRIPTION/CASE SET UP	46
4.1.1	MESH GENERATION	47
4.1.2	GEOMETRY SETUP	51
4.1.3	ACTUAL ENGINE GEOMETRY	52
4.1.4	AREA RATIO	52
4.2	BOUNDARY CONDITIONS	53
4.2.1	FIXED VALUE BOUNDARY CONDITION	53
4.2.2	INLET-OUTLET BOUNDARY CONDITION	54
4.2.3	TOTAL PRESSURE/TEMPERATURE BOUNDARY CONDITION	55
4.2.4	PLENUM PRESSURE BOUNDARY CONDITION	56
4.3	CHOICE OF FLOW CONDITIONS	56

4.3.1	SUPERSONIC FLOW FIELDS: BACK PRESSURE OF 124kPa and 100kPa	60
4.3.2	SUBSONIC FLOW FIELDS: CASES WITH BACK PRESSURE OF 619 kPa and 650 kPa	60
4.3.3	MACH NUMBER	60
4.4	INVESTIGATION OF MASS FLOW RATE	60
4.4.1	RESULTS AND ANALYSIS	62
4.5	SELECTED FLOW FIELDS FOR INITIATION	66
4.5.1	FLOW FIELD FOR SUBSONIC INITIALIZATION	66
4.5.2	FLOW FIELD FOR SUPERSONIC INITIALIZATION	66
5	SIMULATION OF ROTATING DETONATION ENGINE INITIATION	67
5.1	PROBLEM DESCRIPTION/CASE SET UP	67
5.1.1	MESH DESCRIPTION	67
5.1.2	BOUNDARY CONDITION	68
5.1.3	INITIAL CONDITIONS	68
5.1.4	DIFFERENT TYPES OF INITIATION	68
5.2	RESULTS WITH LOW RESOLUTION MESH	68
5.2.1	SUBSONIC FLOW FIELD INITIATION	70
5.2.2	INITIATION WITH SUPERSONIC FLOW FIELD	72
5.3	EFFECT OF HIGHER RESOLUTION ON DETONATION INITIATION	77
5.3.1	SUBSONIC CASE	77
5.3.2	SUPERSONIC CASE	78
5.3.3	CONCLUSION	79
6	CONCLUSION	85
6.1	RECOMMENDATION	86

Appendices

List of Figures

1	Images are of detonation waves in a mixture of $\text{CH}_4 + 2\text{O}_2$ from [KH13] showing coupling effect.	4
2	Rotating Detonation Engine from [LB14]	6
3	Detonation Cell Size with complex fish scale structure [VJ14]	8
4	Control volume for 1-D dynamics [Tat15]	10
5	Rayleigh-Hugoniot curve showing different regions	12
6	Representation of the scheme in finite volume formulation	22
7	Overview of the structure of OpenFOAM [Ope16]	23
8	OpenFOAM case formulation tree [Ope16]	24
9	Pressure file at time 0 (P)	25
10	Inclusion of Transport Equation to rhoCentralFoam. Fig10h shows the advection of "Y" after some time	28
11	Images are from simulation of detonation waves in [MTE17b]	31
12	Case set-up for non-reactive validation (shock tube problem)	32
13	Initial State (t = 0.000s)	32
14	Simulated State (t = 0.007s)	32
15	Initial condition for P , U and T (t = 0.000 s).	33
16	End state showing P, U, T and ρ (t = 0.007 s)	33
17	Pressure vs position for the shock tube problem using the minmod limiter CFL = 0.09	34
18	Flow simulations at different Courant-Friedrichs-Levy number (CFL) numbers	35
19	Flow simulations using superbee limiter, indicating intense amplitude of oscillation	35
20	Disturbance amplitude DP (Pa) vs CFL number for the minmod, van Alabada, van Leer and superBee limiters for a pressure ratio of 10.	36
21	Disturbance amplitude DP (Pa) vs CFL number for the minmod, van Alabada, van Leer and superBee limiters for a pressure ratio of 1.5.	36

22	Disturbance amplitude DP vs CFL number for the minmod, van Alabada, van Leer and superBee limiters for a pressure ratio of 2.	37
23	Disturbance amplitude vs CFL number for the minmod, van Alabada, van Leer and superBee limiters for a pressure ratio of 5.	38
24	Case set-up for reactive validation (piston problem)	39
25	Result from octave codes for Zeldovich-Von Neumann-Döring (ZND) model	41
26	Shock peak pressure vs x axis for $\Delta x =$ half reaction zone length ($L_{1/2}$) /10 and $\Delta x = L_{1/2}/20$ for a domain 200 $L_{1/2}$ long	42
27	P_{shock}/P_0 vs position for $\Delta x = L_{1/2}/10$ for ZND initiated case (600 $L_{1/2}$ long).	42
28	P_{shock}/P_0 vs time for $\Delta x = L_{1/2}/10$ for ZND initiated case (600 $L_{1/2}$ long).	42
29	P_{shock}/P_0 vs position for $\Delta x = L_{1/2}/20$. (ZND)).	43
30	P_{shock}/P_0 vs time for $\Delta x = L_{1/2}/20$. (ZND).	43
31	P_{shock}/P_0 vs position for $\Delta x = L_{1/2}/40$. (ZND)	43
32	P_{shock}/P_0 vs time for $\Delta x = L_{1/2}/40$. (ZND).	43
33	P_{shock}/P_0 vs position for $\Delta x = L_{1/2}/60$. (ZND)	43
34	P_{shock}/P_0 vs time for $\Delta x = L_{1/2}/60$. (ZND).	43
35	P_{shock}/P_0 vs position for $\Delta x = L_{1/2}/80$. (ZND)	44
36	P_{shock}/P_0 vs time for $\Delta x = L_{1/2}/80$. (ZND).	44
37	P_{shock}/P_0 vs position for $\Delta x = L_{1/2}/100$. (ZND)	44
38	P_{shock}/P_0 vs time for $\Delta x = L_{1/2}/100$. (ZND).	44
39	P_{shock}/P_0 vs position for $\Delta x = L_{1/2}/80$ for an extended domain. (ZND)	44
40	P_{shock}/P_0 vs time for $\Delta x = L_{1/2}/80$ for an extended domain. (ZND).	44
41	Image shows single mode pulsating detonation [NNL00].	45
42	P_{shock}/P_0 vs Number of points per $L_{\frac{1}{2}}$ [NNL00].	45
43	Rotating Detonation Engine [LB14]	46
44	Representaion of an RDE [Sha15]	46
45	Image shows unwrapping of an RDE (3D to 2D).	47
46	2D RDE representaion	47
47	2D RDE representaion	47
48	2D Mesh with one obstacle	48
49	Sketch of domain with 2 obstacles	48
50	2D mesh generation with two obstacles	49
51	Initial pressure for the case	49
52	2D mesh with three obstacles	49
53	Image showing mesh blocks(11 in this case)	49
54	Vertices for 2 obstacles using octave code.	50

55	Image of mesh with 10 obstacles (code generated)	50
56	Image of mesh with 100 Obstacles (code generated)	50
57	A resulting flow field using engine A.	51
58	Pressure flow fields in the "slip" BC test with 1 obstacle	54
59	Pressure flow fields in the "slip" BC test with 2 obstacle	54
60	InletOutlet boundary condition [Cfd]	55
61	Plots showing fixed boundary condition with field variables indicating unphysical behaviour in the boundary	57
62	Plots obtained using total pressure and temperature boundary condition . .	58
63	Plots showing plenum boundary condition with field variables	59
64	Steady state subsonic flow field using a back pressure of 650 kPa	61
65	Mach number for subsonic and supersonic initiation flow fields	62
66	Mass flow rate across boundaries for $P_b = 100$ kPa.	63
67	Mass flow rates across boundaries for $P_b = 250$ kPa.	63
68	Mass flow rate across at boundaries for $P_b = 500$ kPa	64
69	Mass flow rate against time across boundaries for $P_b = 650$ kPa	65
70	Subsonic flow field using a back pressure of 619 kPa	69
71	Various initiation techniques considered	70
72	Detonation initiated with a high temperature using a subsonic flow field . . .	71
73	Detonation initiated with a high pressure-temperature zone using a subsonic flow field	73
74	Detonation initiated further downstream with a high temperature zone using a subsonic flow field	74
75	Detonation initiated with a high pressure and temperature zone using a subsonic Flow field	75
76	Detonation initiated with a high temperature zone using a supersonic flow field	76
77	Detonation initiated with a high temperature and pressure zone using a supersonic flow field	80
78	Temperature based initiation for supersonic flow with a pressure ratio of 1.2857	81
79	Pressure-Temperature based initiation for subsonic flow with a pressure ratio of 5	81
80	Pressure-Temperature based initiation for supersonic flow with a pressure ratio of 5	82
81	Initial unstable detonation wave (Subsonic case)	82
82	Image indicates decoupling of shock pressure-reaction(Unsustained Detonation for subsonic case)	83

83	Pressure-reaction coupling indicating a detonation wave (supersonic case) . .	83
84	Coupling remains as wave travels further (supersonic case)	84

List of Tables

- 1 Table of CFL and amplitude of pressure oscillations for the 4 chosen limiters for a pressure ratio of 10.
- 2 Table of CFL and amplitude of pressure oscillations for the 4 chosen limiters for pressure ratio of 2.
- 3 Table of CFL and amplitude of pressure oscillations for the 4 chosen limiters using pressure ratio of 5
- 4 Table of CFL and amplitude of pressure oscillations for the 4 chosen limiters using pressure ratio of 1.5

Abbreviations

CFD Computational Fluid Mechanics

CFL Courant-Friedrichs-Levy number

CJ Chapman-Jouguet

D_{CJ} detonation velocity at Chapman-Jouguet (CJ) point

DDT Deflagration to Detonation Transition

FVM Finite Volume Methods

HLL Harten-Lax-van Leer

HLLC Harten-Lax-van Leer-Contact

KNP Kuganov-Noelle-Petrova

KT Kuganov-Tadmor

ODW Oblique Detonation Wave

OpenFOAM Open source Field Operation And Manipulation

OWDE Oblique Wave Detonation Engine

PDE Pulse Detonation Engine

RDE Rotating Detonation Engine

SDE Standing Detonation Engine

WENO Weighted Essentially Non-oscillatory Schemes

ZND Zeldovich-Von Neumann-Döring

Nomenclature

c unwrapped circumference. m

Δ_R total length of the reaction zone. m

L length of engine. m

E_a activation energy. kJ/kg

l injection width. m

\dot{m} mass flow rate. kg/s

n number of obstacles

γ adiabatic index

H_f heat of formation. kJ/kg

h height of fluid. m

e internal energy per unit mass. kJ/kg

k pre-exponential constant 1/s

$L_{1/2}$ half reaction zone length. m

M molecular mass of mixture. kg/kmol

Q heat release. kJ/kg

u velocity. m/s

t_s obstacle thickness. m

Chapter 1

INTRODUCTION

1.1 SHOCK INTERACTIONS AND COMBUSTION AS A PREREQUISITE FOR DETONATION

It is common practice to generate power from low-speed (subsonic), constant pressure combustion processes. Of growing interest however has been the idea of realizing this highly craved demand for power generation through a rather more spontaneous heat release process, as is the case with detonation waves [EGL91]. The interaction of shock waves in a compressible medium, and exothermic chemical reactions, both occurring on short time scales, gives rise to detonation waves, wherein we have a coupling of hydrodynamic and chemical processes. The self-propagating waves similar to that shown in figure 1 occur on short time and length scales, with propagation speeds in the order of 2–3 km/s in reactive gases [FD79]. These waves generate pressures 6–8 times the initial reactant pressure, and high temperatures (above 2000 K). High and rapid energy conversion is a significant component of a detonation process which is quite promising particularly as it relates to the field of propulsion. The period between 1940s/50s saw an increase in activities in the detonation field, aimed at exploiting its promise of low weight to high power ratio. These efforts mainly focused on understanding the phenomena, with huge interest shown to the development of waves from slow moving combustion waves to fast travelling (detonation), as well as assessing the actual practicability of the process. Three interesting areas of application have been examined in the course of utilizing detonation waves for propulsion. These include the Pulse Detonation Engine (PDE), the Oblique Wave Detonation Engine (OWDE) and the Rotating Detonation Engine (RDE). We examine these 3 briefly.

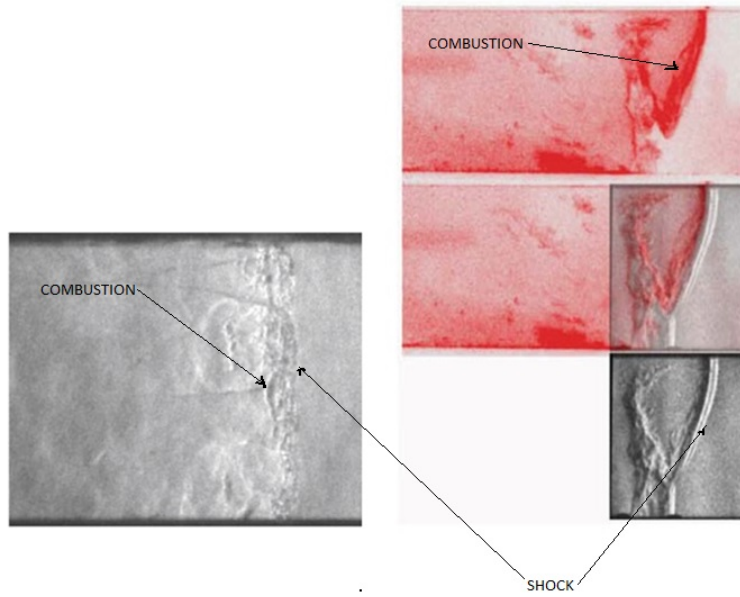


Figure 1: Images are of detonation waves in a mixture of $\text{CH}_4 + 2\text{O}_2$ from [KH13] showing coupling effect.

1.1.1 PULSE DETONATION ENGINE

PDE utilize a detonation wave for propulsion through the use of an intermittent detonation cycle process, where a detonation wave is initiated at one end of a mixture filled chamber [EGL91]. They are based on axially propagating detonation waves mainly in the downstream direction. A lot of studies have been done on this engine. Thomas [Bus96] received a patent for his work on the PDE coupled to a fuel and air source in 1996. Nichols et al. looked at the practicability of a reaction device functioning on intermittent gaseous detonation waves [Nic57] by examining the thrust, fuel-air flow among others. PDE research continues to this day. We note here that a challenge facing the PDE is the fact that its operation is intermittent, needing re-initiations of high frequency, which could be energy demanding. Add that to the low thrust it generates, which in itself results from its mode (pulsed) of operation, necessitated by the need to "refill" the engine, during which time there is zero thrust generation [ZWW]. The intermittent nature of the device also makes it impossible to design a practical nozzle. Hence the focus on PDEs these days gears towards deflagration to detonation transition studies; using obstacles to achieve flame acceleration thereby reducing energy demand [Tat15]. There has been a test flight on an engine modified to run on pulsed detonation [MMM98].

1.1.2 STANDING DETONATION WAVE ENGINE

In this detonation-based type of engine, a detonation wave is stabilized within a chamber, oblique or normal to a wedge [ZWW], with fuel injected and mixed with supersonic air flow, ahead of the wave. The mixture is then compressed by the resulting shock wave after which detonation occurs, and the combustion products used to generate thrust. The Standing Detonation Engine (SDE) requires high inflow velocities and can only work with the inflow of high mach numbers [Kai09] hence even though the idea seems quite promising from a conceptual stand point, the SDE is faced with certain difficulties that challenge its practical implementation. Studies show that it has been quite challenging to sustain detonation waves in the chamber for long [ZWW]. Normal waves are more difficult to stabilize and they usually result in higher stagnation losses in pressure than with oblique waves, hence the desire to rather utilize oblique shock waves for detonation. Braun [BLWC13] and others describe oblique detonation as a combustion process initiated by an oblique shock wave. Powers et al. [PBJJ94] also explains an oblique detonation wave as one wherein the pressure field induced by combustion behind the shock wave influences the wave itself, resulting in a rotation of the oblique wave to a normal orientation, the limiting case being an oblique Chapman-Jouguet detonation, in which there is a mach 1 normal velocity component behind the wave. An increase in heat addition ultimately means a decoupling from the anchoring point, thereby making it rotate to a more normal orientation [PBJJ94]. The idea that by utilizing an oblique shock to initiate a detonation, we can achieve the stabilization of a detonation wave in a supersonic flow field has escalated interest in its use for hypersonic flight [PBJJ94]. Carrier et al. [CFM⁺92] looked at utilizing lasers to initiate such waves by analyzing the practicability of a supersonic combustor which utilizes a stabilized, conically configured Oblique Detonation Wave (ODW). We note that in spite of efforts like this, and many years of research, there remains the challenge of stabilizing a steady oblique detonation in high speed, combustible mixtures.

1.1.3 ROTATING DETONATION ENGINE

This represents an approach to ensuring pressure gain combustion and has been receiving increased attention in the last 10 years, mainly because of its ability to provide significant improvements in thermal efficiency, and the promise of engine design simplification. Far back as in the early 1960s, Voitsekhovskii et al. carried out experiments on continuously rotating detonations [VM63]. Tobita et al. [TFW10] made an application for both Japanese and US based patents for their rotating detonation based device. The RDE mode of operation presents a certain advantage over the PDE concept. The RDE requires a single initial

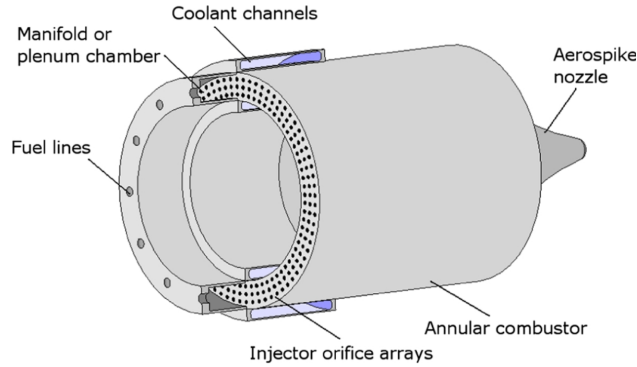


Figure 2: Rotating Detonation Engine from [LB14]

activation/ignition energy to have its four main processes occurring at very high frequencies (in order of 10^3 Hz) [PFHS15] during operation (introduction of reactants, mixture blow-down in the chamber, purging of combustion products) since the detonation wave propagates circumferentially in an annular chamber while consuming combustible products. Hence it is always generating thrust and it doesn't require Deflagration to Detonation Transition (DDT) devices. Figure 2 shows an overview of the RDE design for rocket mode operation. Here we basically see an annular chamber of a given diameter, through which we have fluid injection into the combustion chamber. Depending on the configuration, it could be premixed or separately injected, in which case the oxidizer and the fuel enter through different flow channels. Engine dynamics and geometry give rise to a large number of governing parameters [BZV06]. The goal is that a RDE functions as one, by having at least one stable detonation wave travelling circumferentially in the annular chamber.

1.2 DETONATION AND DEFLAGRATION

Generally speaking, there are two types of self-propagating combustion waves: deflagration and detonations [Lee08]. We provide some insight on these modes of combustion.

1.2.1 WHAT IS A DEFLAGRATION WAVE

Deflagration is a low speed (subsonic) mode of combustion involving a mixture of a fuel and an oxygen source, that results in heat release and rise in temperature. It is common to think of it as a constant pressure process since it usually occurs with little changes in pressure. Common applications include the internal combustion engine and in gas turbines. They are used to create useful work and generate power when initiated, using some spark mechanism, and allowing the flame front to propagate through the fuel-Air mixture usually at speeds

close to 1 m/s [Tat15].

1.2.2 WHAT IS A DETONATION WAVE

Detonation on the other hand can be described as a supersonic burning process involving a combustible mixture, wherein a shock wave triggers a chemical reaction, and is sustained by this exothermic reaction. This enables it to propagate at speeds in the order of 2000 m/s [Tat15]. The wave thickness and whether it can be considered as a shock wave accompanied by a chemical reaction depends on its structure [FD79]. We note therefore that the supersonic wave with a shock structure raises the temperature and pressure of the mixture, [Tat15] hence the reason devices utilizing a detonation wave are often referred to as self-pressurizing. Generally speaking, detonation velocities obtained in gaseous mixtures are usually about 4–7 times higher than the corresponding sound speed in their reactant media [GSMK12] and the velocity depends on the mixture composition, its temperature and pressure. The use of detonation as a combustion process is motivated by the expectation of improved performance, with great expectations for jet propulsion, considering that chemical energy release occurs in a very short time over a small region in space, in comparison to the deflagration mode.

1.2.3 STRUCTURE OF A DETONATION WAVE

Generally speaking, detonations are unstable, possessing a cellular frontal structure [Lee08] with cell boundaries defined by the connecting points of transversely propagating shock waves fed by energy release in the front that is non uniform. This implies, in contradiction to the CJ theory, that there exists an interactive relationship between the wave front and the ongoing energy-release happening behind it [Lee08]. Figure 3 shows a typical representation of this detonation wave structure with the intersection between the normal, lead shocks and the transverse shock waves denoted, as they are commonly called, as triple points. The average width of successive triple points is the cell width, λ . This length scale λ , in geometries where the overall scale is large compared to the detonation cell width, is a characteristic of the detonable mixture (its constituents and stoichiometry) and of its initial thermodynamic state (mainly its initial pressure, but also its initial temperature). In the one-dimensional analysis of detonation waves (the ZND model), the only emerging length scale is the steady detonation thickness, i.e. the distance between the lead front and the end of reaction. Since, for some reaction models, the detonation thickness can be infinite, it is customary to use the half-reaction zone length, $L_{1/2}$ as a reference detonation length scale. $L_{1/2}$ corresponds to the physical extent between the lead shock wave and the point, within the detonation structure, where half of the chemical potential energy has been released as heat. The detonation cell

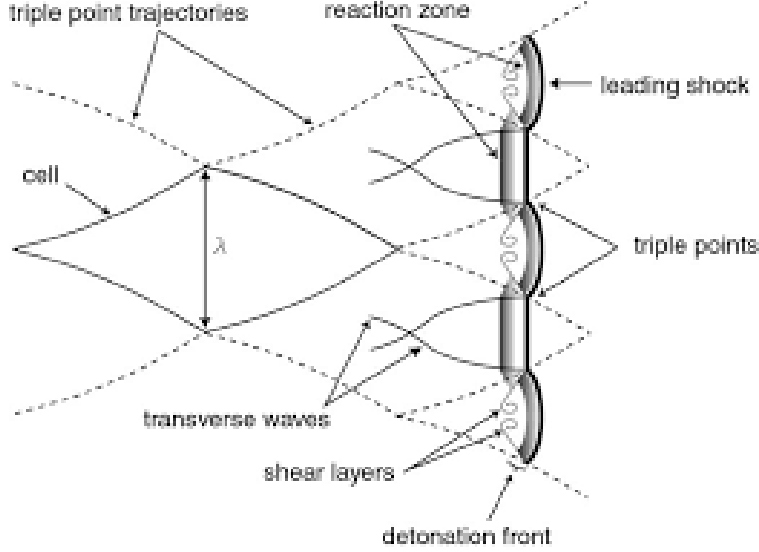


Figure 3: Detonation Cell Size with complex fish scale structure [VJ14]

width can be broadly correlated to the chemical length scale, either the overall detonation thickness or $L_{1/2}$ [ST65].

1.2.4 ESSENTIAL GEOMETRIC RELATIONS

As stated earlier, there needs to be a steady renewal of the chemical reactants, and hence, the layer of combustible mixture, to sustain a rotating/continuous spin detonation wave [BZV06]. Bykovskii et al. in [BZV06] highlighted that this layer, h , should not be any smaller than certain critical value h^* . They showed that a change in the length of engine (L) of an annular cylindrical chamber does not strongly influence the detonation process [BZV06]; making L large enough results in development of boundary layer by lowering effective cross sectional area which in turn aids momentum heat losses [BZV06]. It can be seen from experiments that L should be between 1.5 to 2 times higher than h for obtaining stable transverse detonation waves (TDWs). The critical value the of height of fluid (h), is related to the total length of the reaction zone (Δ_R), in the detonation wave as shown in [BZV06]. Therefore the expression below is presented;

$$h \approx h^* \approx (17 \pm 7)\Delta_R. \quad (1)$$

In the absence of physical processes (premixed gaseous mixture) [BZV06] or in cases where their duration can be neglected,

$$\Delta_R \approx 0.7\lambda \quad (2)$$

where λ the width of the self-excited cell of the detonation front, the detonation cell size, readily measured and known for many mixtures. Combining 1 and 2 as in [BZV06] gives

$$h \approx (12 \pm 5)\lambda. \quad (3)$$

which agrees well with experimental values.

Another useful geometric parameter [BZV06] is given as

$$K = l/h = \pi d_c/nh. \quad (4)$$

which has been shown to be roughly the same for all annular cylindrical chambers using a gaseous oxidizer, given as $K = 7 \pm 2$. The parameter n is the number of transverse detonation waves rotating in the chamber. Thus with K known, for $n = 1$, from 4, we obtain that for a prescribed mean pressure in a chamber, the minimum chamber diameter [BZV06] must be such that

$$(d_c)_{min} = hK/\pi. \quad (5)$$

For chambers operating on a gaseous oxidizer, we have $(d_c)_{min} = 40\Delta_R$ with the value of the minimum diameter known, the minimum and optimal length of the chambers can be determined approximately as:

$$L_{min} \approx 2h \approx d_c/n. \quad (6)$$

$$L_{opt} \geq 4h \approx 0.71 \approx 2d_c/n. \quad (7)$$

These equations are important since they serve as benchmarks for engine design. They also highlight the usefulness of knowing the cell size of a mixture.

Thus given a cell size which is a function of a given mixture composition. We can determine the geometry needed such that an engine designed to run on such mixture can support a rotating detonation wave. We can also simulate such an engine numerically since we know geometric parameters and can specify an equivalence ratio. The minimum diameter and channel width have been shown to be approximately $d \leq 30\Delta_R$ and $\delta \leq 2.5\Delta_R$, respectively [LB14]. This implies that the RDE can be scaled up or down depending on how reactive the mixture components are.

1.2.5 THERMAL EFFICIENCY OF A DETONATION WAVE

Cycles based on detonations are more thermodynamically efficient. Detonations, which are modelled as constant volume combustion (Humphrey and Fickett-Jacobs thermodynamic

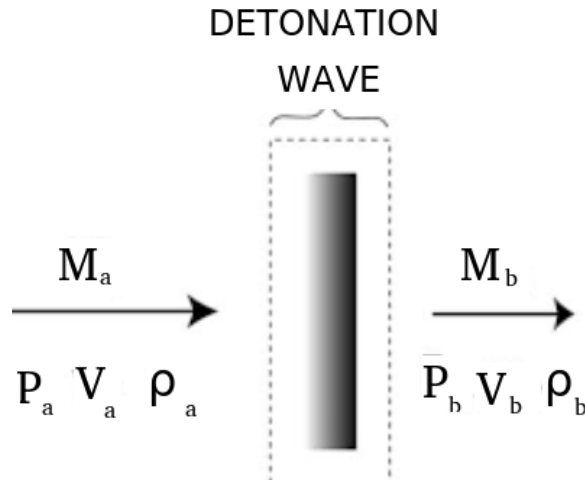


Figure 4: Control volume for 1-D dynamics [Tat15]

cycles) processes produce a higher thermal efficiency (1.3 -1.5 times more) [Tat15] than that of a constant pressure combustion cycle at an equivalent pressure ratio, meaning that this can result in a similar increase in fuel efficiency provided other mechanical and related efficiencies can be maintained [BP94]. The formation and propagation of a detonation wave compresses the gas ahead of it causing a dramatic increase in pressure and temperature after the combustion process. This can be described by the one dimensional Chapman-Jouguet theory and the Zeldovich-Von Neumann-Döring (ZND) model.

1.2.6 ONE-DIMENSIONAL DETONATION DYNAMICS

The conservation equations are presented for a combustion wave with a constant area flow. Figure 4 describes a control volume for analysis. The following four assumptions are pertinent:

1. the detonation is steady,
2. the flow is one-dimensional and planar,
3. the products approach a state of equilibrium at a distance behind the detonation front,
4. the detonation velocity obtained is the minimum permitted by the conversion conditions.

The mass, momentum and energy balances across the cross sectional area suggest that a relationship exists between the states on two sides of a shock [Dah11]. This results in the following expressions based on the assumptions made earlier:

$$\rho_a V_a = \rho_b V_b \quad (8)$$

$$P_a + \rho_a V_a^2 = P_b + \rho_b V_b^2 \quad (9)$$

$$\rho_a V_a h_a + \rho_a \frac{V_a^3}{2} = \rho_b V_b h_b + \rho_b \frac{V_b^3}{2} \quad (10)$$

These are known as the "energy and mass balances". They express a relationship between thermodynamic quantities before and after a shock, with the left and right values representing low and high pressure regions in this case. A combination of these equations result in

$$\frac{\gamma - 1}{\gamma} \left(\frac{P_b}{\rho_b} \right) - \frac{P_a}{\rho_a} - \frac{P_b - P_a}{2} \left(\frac{1}{\rho_b} + \frac{1}{\rho_a} \right) = -(h_b - h_a) \quad (11)$$

It is pertinent to state that these relations have been derived without considering the specific processes within the control volume. Thus the distinguishing factor between a detonation and a shock is that whereas for a shock wave the chemical properties remain the same, in the case of a detonation, there is an energy release due to changes in chemical composition. Thus, the main difference in analysis would be the enthalpy behind the detonation, i.e. at the exit of the control volume, h_b [Dah11]. For a given value of P_a and change in enthalpy ($-(h_b - h_a)$), a representative plot of P_b against $1/\rho_b$ is known as the Hugoniot curve, and point $(P_a, 1/\rho_a)$ refers to the unreacted state. It represents the connection of all points representing a solution of the conservation equations as restated in equation 11. Two points known as the Chapman-Jouguet points exist by drawing tangents through the Hugoniot origin. The mass and momentum equations can be combined to obtain the expression:

$$\rho_a^2 v_a^2 = \frac{P_b - P_a}{\frac{1}{\rho_a} - \frac{1}{\rho_b}} \quad (12)$$

This yields the Rayleigh relation. A combination of equations 11 and 12 results in the Rayleigh-Hugoniot relation which shows the regions within which we can have a detonation or deflagration. Figure 5 shows the various regions of the Rayleigh-Hugoniot curve. We note that regions 1 and 2 are possible areas for strong and weak detonation, while regions 3 and 4 are those for strong and weak deflagration. However, looking at region 5, we can see that $\frac{1}{\rho_a} - \frac{1}{\rho_b}$ results in a negative number, and $P_a - P_b$ gives a positive result. These two answers are thermodynamically impossible; therefore a solution to the Rayleigh-Hugoniot line for detonation and deflagration at these points does not exist [Dah11]. Chapman [Cha99] and Jouguet [Jou05] independently stated that detonations propagate at a given velocity, which is the minimum for the *branch* that can possibly result in a detonation. This velocity is known

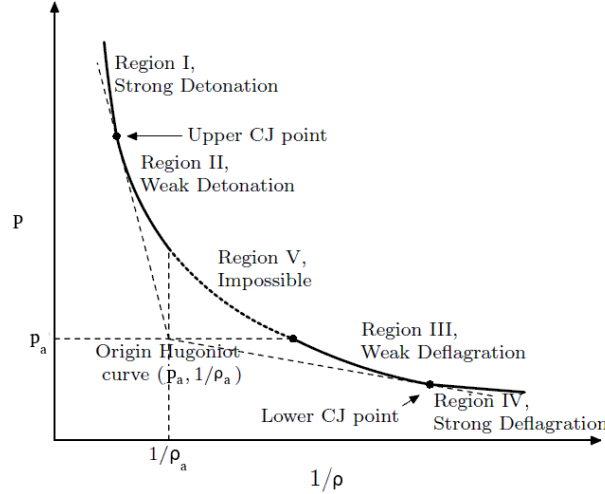


Figure 5: Rayleigh-Hugoniot curve showing different regions
[hys16]

as the Chapman-Jouguet (CJ) velocity. Further manipulations of equation 11 show that the mach number behind the shock wave is unity ($M_b = 1$) at the Chapman-Jouguet points. For emphasis, the CJ theory is arrived at with complete disregard for the detonation structure. It is essentially a consideration of the possible solutions of the steady one-dimensional conservation equations that link the upstream and downstream equilibrium states of the reactants and products, respectively. In the absence of some description for the structure however, the propagation mechanism of the detonation wave cannot be known. Although shock-induced ignition was known to the early pioneers who discovered the phenomenon, it was Zeldovich (1940), Von-Neumann (1942), and Döring (1943) [Zel40, VN42, Dör43] who independently described the model of the detonation structure as comprising a lead shock closely followed by a finite chemical reaction zone. Their model is known as the ZND model and is presented in chapter 3.

1.3 THESIS OVERVIEW

Detonation as a concept is introduced, discussing various efforts at utilizing it for power branch generation in chapter 1. Chapter 2 discusses some select numerical schemes and their formulations and presents results obtained from in-house implementation of the schemes using Octave. It also presents the implementation of the Kuganov-Tadmor (KT)/Kuganov-Noelle-Petrova (KNP) scheme in the Open source Field Operation And Manipulation (OpenFOAM). Chapter 3 evaluates the KNP scheme with a view to determine the conditions under which the scheme is suitable for the simulation of detonation waves. This was done through

a reactive and non reactive validation process using a one-dimensional shock tube case. Chapter 4 is concerned with the simulation of the cold engine starting process for an RDE. Chapter 5 shows different methods of initiation of an RDE. Chapter 6 presents conclusion and recommendations based on findings made, and also provides some of the challenges during the course of this work. Certain tables/codes generated during the course of this work are included in the appendix section.

Chapter 2

THE KURGANOV- TADMOR/KURGANOV-NOELLE- PETROVA SCHEME AND OpenFOAM

2.1 REVIEW OF NUMERICAL DISCRETIZATION

For partial differential equations, it is common to employ numerical solution schemes that consist of discretizing the spatial extent, discretizing time, and finally expressing derivative quantities approximately on those discretized grids. The approximate integrated functions of the differential system are then obtained at discrete time values and either (a) at discrete points in space, in which case they are called finite difference or finite element methods, or (b) over a certain region of space (i.e. as an average value over a given spatial cell), in which case they are referred to as finite volume methods. For problems involving shock wave and other discontinuities, finite volume methods are the almost exclusively appropriate methods. Finite difference schemes are, except to some degree for 1st order accurate schemes, incapable of capturing discontinuities. Finite volume methods discretize the integrated form of a system of PDE, making them able to capture discontinuities in function values, also called weak solutions.

In this chapter, an introduction to the Euler equations that describe the compressible flow is done. Next is a review of the Godunov-type schemes, most commonly employed for compressible flow problems, as well as the Lax-Friedrichs family of numerical schemes. A summary of the KT and KNP numerical schemes for compressible flow, which are an extension of

the Lax-Friedrichs scheme is presented. Finally, the OpenFOAM numerical framework is reviewed.

2.2 GOVERNING EQUATIONS

The Euler equations are the balance of mass, momentum and energy, and govern the dynamics of inviscid, compressible flow.

$$\rho_t + (\rho u)_x + (\rho v)_y + (\rho w)_z = 0, \quad (13)$$

$$(\rho u)_t + (\rho u^2 + p)_x + (\rho uv)_y + (\rho uw)_z = 0, \quad (14)$$

$$(\rho v)_t + (\rho uv)_x + (\rho v^2 + p)_y + (\rho vw)_z = 0, \quad (15)$$

$$(\rho w)_t + (\rho uw)_x + (\rho vw)_y + (\rho w^2 + p)_z = 0, \quad (16)$$

$$E_t + [u(E + p)]_x + [v(E + p)]_y + [w(E + p)]_z = 0. \quad (17)$$

In equations 13–17, the Euler equations are expressed in terms of the primitive or physical variables density, $\rho(x, y, z, t)$, pressure $P(x, y, z, t)$, velocity $V(x, y, z, t) = [u, v, w]$ and total energy $E(x, y, z, t)$, where

$$E = \rho \left(e + \frac{V^2}{2} \right). \quad (18)$$

The primitive variables are more convenient to describe the problem's initial and boundary conditions, but, as shown by Toro [Tor] and Hou et al. [HL94], numerical methods that discretize the primitive-based, differential form of the equations are unable to accurately capture discontinuities. It is necessary to integrate the PDE system in its conservative form where the integrated functions are the conserved variables

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix} \quad F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E + p) \end{bmatrix} \quad G = \begin{bmatrix} \rho v \\ \rho v^2 + p \\ \rho vu \\ \rho vw \\ v(E + p) \end{bmatrix} \quad H = \begin{bmatrix} \rho w \\ \rho w^2 + p \\ \rho wu \\ \rho wv \\ w(E + p) \end{bmatrix} \quad (19)$$

and the system can be expressed as

$$U_t + F(U)_x + G(U)_y + H(U)_z = 0, \quad (20)$$

Only one equation is still required to close the set, the equation of state that relates $e = e(P, \rho)$. The ideal gas equation of state is utilized for analysis (i.e $\rho = P/RT$).

2.2.1 CONSERVATION LAWS IN INTEGRAL FORM

A useful form of expressing conservation is its integral form. In one-dimension, the Euler equation is:

$$U_t + F(U)_x = 0 \quad (21)$$

where subscript t indicates time dependence. This continuous mathematical expression can be confined within a finite extent by discretization in time and space. In terms of volume considerations, a domain can split this into m computing cells (volumes) $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ with each of width $\Delta x = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$, where subscript "i" ranges from 1 to m . A control volume $V_c = I_i \times [t^n, t^{n+1}]$ is also defined [ZM10]. By evaluating the spatial integral of equation 21 above given by the expression,

$$\frac{d}{dt} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} U(x, t) dx = F(U(x_{i-\frac{1}{2}}, t)) - F(U(x_{i+\frac{1}{2}}, t)) \quad (22)$$

as well as the temporal integration between t^n and $t^n < t^{n+1}$ represented by

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} U(x, t^{n+1}) dx = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} U(x, t^n) dx + \int_{t^n}^{t^{n+1}} F(U(x_{i-\frac{1}{2}}, t)) dt - \int_{t^n}^{t^{n+1}} F(U(x_{i+\frac{1}{2}}, t)) dt \quad (23)$$

It is possible therefore to represent equation 21 in integral conservation form [ZM10]. By defining the variable in a cell U_i^n as equal to some integral average within the cell $\frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(x, t^n) dt$ and the flux through an interface, $F_{x_{i+\frac{1}{2}}}$ as given by the expression $\frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} F[U_{(x_{i+1/2}, t)}] dt$, equation can be rewritten 21 as:

$$U_i^{n+1} = U_i^n + \frac{\Delta t}{\Delta x} (F_{i-\frac{1}{2}} - F_{i+\frac{1}{2}}) \quad (24)$$

The above mathematical expression has been derived purely from calculus, without any applied constraints imposed so far. It has been assumed that knowledge of the analytical functions $U(x, t)$ and $F(x, t)$ exists. [ZM10]. However what is of interest to us is a numerical formulation, which is bounded by some finite domain. Hence there is need to determine the constituents of the expression in equation 24 above by some form of estimation. Numerical methods differ in the way these constituents are evaluated. Bearing this in mind, a presentation of some select numerical schemes studied in the course of this work is done.

2.3 TYPES OF NUMERICAL SCHEMES

A description of the two types of numerical methods mentioned above is presented: Godunov based methods and central schemes, with some examples.

2.3.1 GODUNOV'S METHOD (FIRST-ORDER UPWIND METHOD)

We note the expression in figure 24 evolves from a mere mathematical expression to a numerical scheme [ZM10] when we make approximations for the evaluation of the numerical fluxes and cell averages, U_i . Fundamental to Godunov's (First order) approach is the idea that at a certain time step, n , the control volume has a piecewise constant grid data [God59] of the form $u_i^n = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(x, t^n) dx$. Doing this results in a loss of some level of cell data accuracy, which gets improved upon by utilizing higher order reconstructions, of which a number of methods exist. Sergei Godunov [God59] is acknowledged to be the first to successfully carry out a conservative extension of the first-order upwind scheme of Courant, Isaacson and Rees (CIR) [CIR52] to nonlinear systems of hyperbolic conservation laws. We note that the main highlight of upwind methods is that the discretization depends on the sign, and hence direction of wave propagation [ZM10]. Godunov's first-order upwind method is a conservative method of the form of equation 21, and numerical fluxes ($F_{i-\frac{1}{2}}$, $F_{i+\frac{1}{2}}$) are evaluated by solving a local Riemann problem.

2.3.2 MUSCL-HANCOCK APPROACH FOR HIGHER RESOLUTION

Van Leer [VL76], is credited with the idea of modifying the piecewise constant proposed by Godunov as a necessary step in achieving high resolution schemes (second order and above). Thus one can for instance perform a linear interpolation of cell data to the interface. His idea is known today as the MUSCL (Monotone Upstream-Centered Schemes for conservation laws) or variable extrapolation approach [VL76], for conservation Laws. Godunov's order barrier theorem implies that one cannot have a scheme higher than first order that is free from spurious oscillations [God57]. Therefore, to enjoy the benefit of higher resolution methods, they must be used together with a limiter. A piecewise linear, local reconstruction of cell data, u_i^n is achieved using the linear expression $u_i(x) = u_i^n + \frac{(x-x_i)}{\Delta x} \Delta_i$, and the ratio $\frac{\Delta_i}{\Delta x}$ refers to a slope of $u_i(x)$ in cell I_i . At the extreme ends, $u_i^L = u_i(0) = u_i^n - \frac{1}{2} \Delta_i$, $u_i^R = u_i(\Delta x) = u_i^n + \frac{1}{2} \Delta_i$ are usually called boundary extrapolated values. Toro [Tor] states that most approaches neglect the generalized Riemann problem, utilizing the boundary extrapolated

values u_i^L , u_i^R for each function of $u_i(x)$. Therefore, the result is a constant data Riemann problem;

$$u(x, 0) = \begin{cases} u_i^R, & \text{if } x < 0. \\ u_{i+1}^L, & \text{if } x > 0. \end{cases} \quad (25)$$

for the cells. The Muscl-Hancock approach requires that these extrapolated values are evolved by half a time step according to the relations below

$$\bar{u}_i^L = u_i^L + \frac{1}{2} \frac{\Delta t}{\Delta x} [F(u_i^L) - F(u_i^R)] \quad (26)$$

$$\bar{u}_i^R = u_i^R + \frac{1}{2} \frac{\Delta t}{\Delta x} [F(u_i^L) - F(u_i^R)] \quad (27)$$

after which the solution is evolved to the next time step by evaluating the piece-wise constant data Riemann problem. Simulation of test cases for the MUSCL scheme for the Burgers equation using the same discretized domain as with the Godunov first order test cases to evaluate the effectiveness of different numerical schemes were also conducted and have not been included in this work. Octave codes have been uploaded to GitHub and are available on request.

2.4 CENTERED SCHEMES (LAX-FRIEDRICHS FAMILY OF SCHEMES)

A characteristic trait of Godunov-type schemes is that the evaluation of flux quantities, a necessary step in solving finite volume based problems, involves the solution to a Riemann problem. This demands the characteristic decomposition to enable the evolution of the problem in space and time. However, the centered schemes achieve the solution of a quantity at the next time step without solving a Riemann problem, rather they do this by integrating over the Riemann fan. This integration implies that both the left and right travelling waves are catered for in this method. Hence there is no need to solve a Riemann problem at every cell interface, thus these methods could be computationally inexpensive. A presentation of some of the schemes belonging to this family and their numerical formulations.

2.4.1 LAX-FRIEDRICHS SCHEME

The Lax-Friedrichs scheme applied to the linear advection equation [Tor] is given as

$$u_i^{n+1} = \frac{1 + CFL}{2CFL} (u_{i-1}^n) + \frac{CFL - 1}{2CFL} (u_{i+1}^n) \quad (28)$$

where CFL is the Courant-Friedrichs-Levy number (CFL). Given the 1D conservative equation for linear advection,

$$u_t + a(u)_x = 0 \quad (29)$$

the flux term is evaluated as

$$F_i^{n+\frac{1}{2}} = \frac{1 + CFL}{2CFL} F(u_i^n) + \frac{CFL - 1}{2CFL} F(u_{i+1}^n) \quad (30)$$

and its solution at a cell interface, i can be obtain. Looking at equation 28, it is seen that it is a weighted integral average of a corresponding Riemann problem with different left and right states at time $t = \frac{1}{2}\Delta t$. For a nonlinear system of conservation laws

$$U_t + F(U)_x = 0 \quad (31)$$

considering the scheme to be an integral average within cells, a conservative flux generalization for the LaxFriedrichs scheme yields :

$$F_{i+\frac{1}{2}} = \frac{1}{2}(F_i^n + F_{i+1}^n) + \frac{1}{2} \frac{\Delta x}{\Delta t} (U_i^n - U_{i+1}^n) \quad (32)$$

and hence obtain the general solution of the Lax-Friedrichs scheme evolved in time as

$$U_i^{n+1} = \frac{1}{2}(U_{i-1}^n + U_{i+1}^n) + \frac{1}{2} \frac{\Delta t}{\Delta x} (F_{i-1}^n - F_{i+1}^n) \quad (33)$$

Equation 32 is the numerical flux for the Lax-Friedrichs scheme when applied to systems of conservation laws. It can be easily noted that the flux evaluation and the evolution of the solution in time is done without solving a Riemann problem.

2.4.2 NESSYAHU-TADMOR (NT) SCHEME

Consider again the conservation equation in 1D given below:

$$\frac{dU}{dt} + \frac{\partial}{\partial x}(F(U)) = 0 \quad (34)$$

As stated earlier, Godunov type schemes are upwind in nature. This can be demonstrated as follows: if the characteristic speeds within all the neighbouring finite control volumes are

positive (or negative), an integration of equation 34 reduces to

$$u_j(t + \Delta t) = u_j(t) - \lambda[F(u_j(t)) - F(u_{j-1}(t))] \quad (35)$$

$$u_j(t + \Delta t) = u_j(t) - \lambda[F(u_{j+1}(t)) - F(u_j(t))] \quad (36)$$

where the second equation corresponds to the case with negative speed [NT90]. In this convention of NT scheme, $\lambda = \frac{dt}{dx}$. For cases where there are existing waves with both signs for the characteristic speeds (right and left going waves) in the control volume, this becomes complicated and the flux evaluation demands the "direction of the wind" is taken into account [NT90] so as to differentiate both waves within the Riemann fan, hence the exact solution to the Riemann problem in this case becomes challenging. Nessyahu et al. [NT90] proposed the NT scheme as a way of by-passing the need to solve multiple Riemann problems for such cases. Starting from the staggered version of the Lax-Friedrichs scheme given as:

$$u_{j+\frac{1}{2}}(t + \Delta t) = \frac{1}{2}[v_j + u_{j+1}]\lambda[F(u_{j+1}(t)) - F(u_j(t))] \quad (37)$$

they developed a central difference scheme that integrates over the Riemann fan. We note that such integration will cater to waves in both directions. Hence, it eliminates the need to solve a Riemann problem. However since the Lax-Friedrichs scheme is quite diffusive and first order, they introduced MUSCL type interpolants thereby making the scheme of higher order(second order). The reader is referred to [NT90], however a summary of the process results in the NT scheme described by a first order predictor step:

$$u_j(t + \frac{\Delta t}{2}) = u_j(t) - \frac{1}{2}\lambda F'_j \quad (38)$$

and the second order corrector step

$$u_{j+\frac{1}{2}}(t + \Delta t) = \frac{1}{2}[u_j(t) + u_{j+1}(t)] + \frac{1}{8}[u'_j - u'_{j+1}] - \lambda[F(u_{j+1}(t + \frac{\Delta t}{2}) - F(u_j(t + \frac{\Delta t}{2}))) \quad (39)$$

where the prime, " ' " indicates a derivative. The non-oscillatory properties of the scheme have been demonstrated in [NT90] . We note that there are two versions of the scheme: the staggered and its non-staggered forms. Codes were implemented for both versions in Octave.

2.4.3 DESCRIPTION OF THE KNP NUMERICAL SCHEME

The reader is referred to the works of Kurganov et al. in [KBS15, KNP01, KT00], where a detailed description of the mathematical formulation of the second order numerical scheme and its implementation in OpenFOAM has been described. While the KT is a central

scheme based on the Lax-Friedrichs scheme, the KNP, its generalization, is central- upwind; central since it involves integration over the Riemann fan with no decomposition required, and upwind because it takes into account the wave direction of travel by measuring the "one-sided local" speeds of propagation which for a one dimensional case results in the eigenvalue of the flux Jacobian, as opposed to the spectral radius for KT, which is less precise [KNP01]. Our simulations used the KNP scheme. As mentioned earlier, conservative formulation of the Euler equation is of particular interest, in which case it is mandatory that some flux evaluation technique is used to evolve the equation in time. The finite volume formulation for flux evaluation through a cell interface for KNP as presented in [KBS15] is therefore highlighted in this section. The system solved is usually the reactive inviscid Euler equations in conservation form (here in its 1D, single reaction form):

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = S(G) \quad (40)$$

where

$$U = \{\rho, \rho u, \rho(e + u^2/2 + yQ), \rho y\} \quad (41)$$

is the vector of conserved variables, with the primitive variables being density, (ρ), velocity (u), internal energy per unit mass (e), and extent of reaction y . The vectors

$$F = \{\rho u, \rho u^2, \rho(e + P/\rho + u^2/2 + yQ), \rho u y\} \quad (42)$$

and

$$s(U) = \{0, 0, 0, r\} \quad (43)$$

are the fluxes, and the source terms associated with the one-step Arrhenius chemical reaction rate, r . The system is most often simulated employing finite volume methods, with numerical schemes capable of handling large, rapid gradient changes, i.e., shock waves. The implementation of the algorithm has been discussed above. The KT/KNP scheme is a Riemann-free flux evaluation scheme where the flux, $F(U)$ of the quantity U , is computed as

$$F(U) = \psi_R \phi_R = \psi_R^P ((\alpha_R^P \phi_R^P + \alpha_R^P \alpha_R^{min}) + \psi_R^N (\alpha_R^N \phi_R^N - \alpha_R^P \alpha_R^{min})). \quad (44)$$

where the above is in the original KNP notation from [KBS15] as implemented in OpenFOAM. The subscript R implies that a quantity is computed at the cell interface, and the superscripts, "P" and "N", indicate two different cells, from which the cell values are interpolated to the cell face, i.e. the two sides of a cell face as represented in figure 6. ψ is the vector of conserved quantities, U , \vec{S} is the cell face normal area, and ϕ is $\dot{m}/\rho = \vec{V} \cdot \vec{S}$, the

volumetric flow rate. In a more familiar notation, 44 can be rewritten as:

$$F(U) = U_R^P((\alpha_R^P(\vec{V}_R^P \cdot \vec{S})_R^P + \alpha_R^P \alpha_R^{min}) + U_R^N(\alpha_R^N(\vec{V}_R^N \cdot \vec{S})_R^N - \alpha_R^P \alpha_R^{min})). \quad (45)$$

The quantity $(\alpha_R^P \alpha_R^{min})$ can be defined as ω_f , a product of a weighting and the volume flux, where $\alpha_R^P = \alpha_R^{max}/(\alpha_R^{max} + \alpha_R^{min})$, and $\alpha_R^N = \alpha_R^{min}/(\alpha_R^{max} + \alpha_R^{min})$. We note that for the KT variant, the value of α_R^P and ω_f is 0.5, making the scheme central.

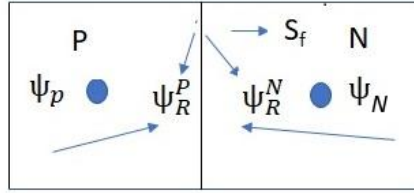


Figure 6: Representation of the scheme in finite volume formulation

The wave speeds are evaluated as follows:

$$\alpha_R^{max} = \max(\phi_R^P + c_R^P |\vec{S}_R|, \phi_R^N + c_R^N |\vec{S}_R|) \quad (46)$$

$$\alpha_R^{min} = -\min(\phi_R^P - c_R^P |\vec{S}_R|, \phi_R^N - c_R^N |\vec{S}_R|) \quad (47)$$

or in more familiar notation in the detonation community,

$$\alpha_R^{max} = \max(\vec{V}_R^P \cdot |\vec{S}_R| + c_R^P |\vec{S}_R|, \vec{V}_R^N \cdot |\vec{S}_R| + c_R^N |\vec{S}_R|) \quad (48)$$

$$\alpha_R^{min} = -\min(\vec{V}_R^P \cdot |\vec{S}_R| - c_R^P |\vec{S}_R|, \vec{V}_R^N \cdot |\vec{S}_R| - c_R^N |\vec{S}_R|) \quad (49)$$

where V represents the macroscopic or particle velocity within a given cell, c is the thermodynamic sound speed, ($c_R^P = \sqrt{\gamma_R T_R^P}$, $c_R^N = \sqrt{\gamma_R T_R^N}$) and \vec{S}_R is the surface area normal of the interface. The details of the scheme are available in [KBS15]. The conditions at the cell faces are evaluated using a linear, slope-limited interpolation. For non-reactive cases, the inviscid Euler equations are solved, while for the reactive cases, a single step, irreversible reaction with heat release (Q) and a rate $r = ke^{[-Ea/RT]}$ is coupled to the system and solved.

2.5 OVERVIEW OF OpenFOAM

2.5.1 WHAT IS OpenFOAM

Simulations for this work have been based on the OpenFOAM framework, commonly referred to as OpenFOAM. It is a library of over 100 C++ codes [Ope16] that can be grouped together as needed, to create executables, referred to in OpenFOAM as applications. OpenFOAM has over 250 of them pre-built [Ope16]. These can be categorized into two: solvers heavily employed for problem formulation and analysis in continuum mechanics, and utilities that execute some form of data manipulation. OpenFOAM is such that users are allowed the freedom to design their own custom solvers and utilities that relate better to their desired case or specific problem. This is one of the major reasons it is gaining traction in the CFD community [WTJF98]. A representation of the basic structure of OpenFOAM is shown in figure 7, where it can be seen that there are three phases to an OpenFOAM problem: pre-processing, solving and post-processing. OpenFOAM, when installed, comes with a

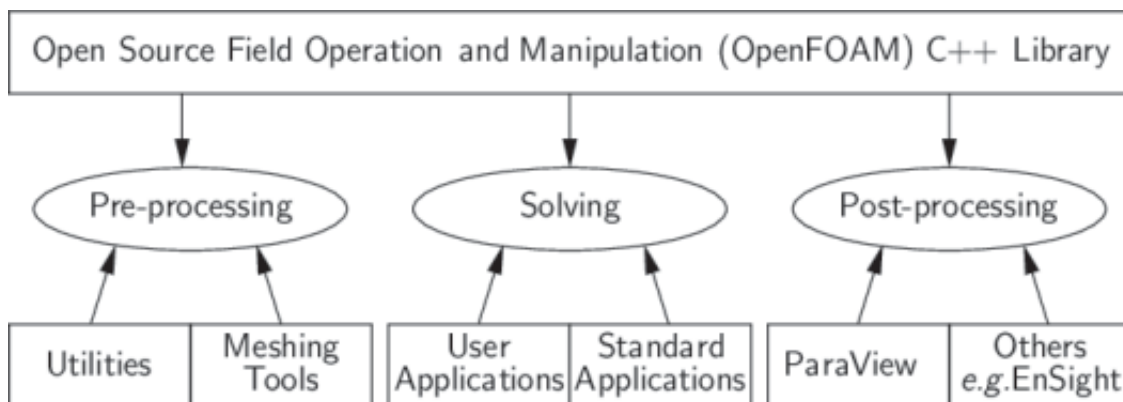


Figure 7: Overview of the structure of OpenFOAM [Ope16]

case set-up and result analysis section, referred to as the pre-processing and post-processing environments, which in actual terms employ the use of these utilities in their operation. For instance, to define a hexagonal mesh, one will have to specify the co-ordinates in a text file named `blockMeshDict`, which can be called by the utility "blockMesh" to generate the mesh data. One can then use, for instance, the `setFields` utility to specify initial conditions within two regions of the domain as with a basic multiphase flow problem for instance. The whole essence of this is to guarantee some form of consistency particularly with regards to data utilization across all environments [Ope16].

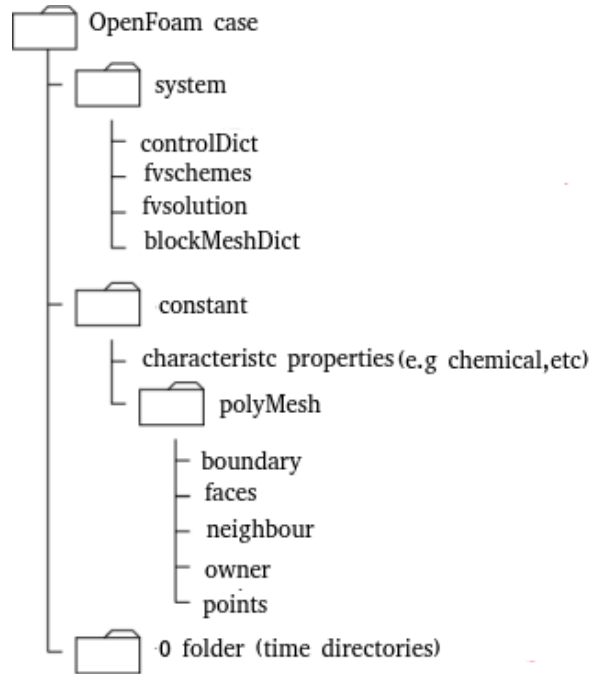


Figure 8: OpenFOAM case formulation tree [Ope16]

2.5.2 FILE STRUCTURE OF OpenFOAM

To run an OpenFOAM case, a basic file description is presented in figure 8. Included here as well are some utilities to help facilitate understanding, particularly as they relate to this thesis. We see that the we can define an OpenFOAM case or problem by populating the 3 major file folders or directories included in figure 8. These will be discussed, making reference to a shock tube case that has been built in OpenFOAM. We go over these 3 folders described below.

0 FOLDER(TIME DIRECTORY)

The time directory contains the initial and boundary conditions of field variables we are seeking solutions to, or those they depend on. e.g. we may not need to initialize density since we can obtain it by specifying temperature and gas constant. Initial fields in OpenFOAM simulations must be specified as a requirement even for cases that are within the domain of steady flow. We note that the directory is called a 0 folder as we ordinarily begin our simulations from a "0" time. However as we advance in time, OpenFOAM writes out other time folders based on a user specified output time and interval. Referencing our shock tube case, we need to evolve the velocity, pressure and temperature as we advance in time. Hence in our zero (0) folder, we have specified the U, P and T files which contain the initial and boundary conditions for the case. Figure 9 shows a sample pressure file, known as a P file,

```

/*-----* C++ *-----*/
=====
Field      | OpenFOAM: The Open Source CFD Toolbox
Operation  | Version: 5
And        | Web: www.OpenFOAM.org
Antipulation
FoamFile
{
  version      2.0;
  format       ascii;
  class        volScalarField;
  object       p;
}
// *****

dimensions    [1 -1 -2 0 0 0];

internalField uniform 0;

boundaryField
{
  sides
  {
    type        zeroGradient;
  }

  empty
  {
    type        empty;
  }
}
// *****

```

Figure 9: Pressure file at time 0 (P)

for our case. A number of things are striking. We see an array specified as dimensions. In OpenFOAM, dimensions are specified by considering the S.I unit. The first element of the array is always the unit of mass, in kg, and the next being length in meters. The rest of the elements are time (s), temperature (K), number of moles (mol), current (A) and luminous intensity (cd). Thus, derived quantities are a combination of two or more of these as seen in the unit of pressure kg/ms^2 . OpenFOAM data is specified using dictionaries: an entity containing data entry(ies) that can be called upon by utilizing keywords [Ope16]. Most OpenFOAM data files are themselves dictionaries, and can contain one or more data entries. Thus the key word "InternalField" is a dictionary containing one keyword, and is assigned the value 0 to represent the pressure values within the domain initially. The boundary field is a dictionary containing the boundary condition for the field variable (P in this case). Here we see that a "zeroGradient" boundary condition has been assigned to the field at a region defined as "sides" and a boundary named empty is assigned an "empty" boundary condition. There are a host of boundary conditions available in OpenFOAM. In the case discussed above, the internal field is constant or uniform. OpenFOAM also allows for non-uniform internal fields. Thus we can use a non-uniform field obtained from another software (e.g. octave), or from an existing simulation, as initial data for another case.

CONSTANT FOLDER

A basic constant folder will contain 3 elements: A thermophysical properties (thermophysicalProperties in OpenFOAM) file, a turbulence properties (turbulenceProperties in OpenFOAM) file, and a folder known as the PolyMesh. A thermophysical properties file describes

the thermophysical models utilized in the OpenFOAM case in question, where the thermal energy, compressibility and/or mass transfer is of particular interest [Ope16]. These properties can be constant, or can be a function of thermodynamic properties such as T,P. The thermophysicalProperties dictionary is called upon by any solver that utilizes the thermophysical model library [Ope16]. OpenFOAM creates these models as pressure-Temperature system which in turn can be used to create other models. A lot of these can be found in OpenFOAM and are readily described in the user guide. The thermophysical properties (keyword thermophysicalProperties) begins with a "thermotype", which defines the basic equation of state and models for fluid transport and mixture modelling among others. The turbulenceProperties file contains the turbulenceProperties dictionary wherein we can specify the turbulence model we wish to use. For our non-turbulent cases, the "simulationType" dictionary is always set to laminar, and inviscid flow is enforced by specifying a "0" value of viscosity in the chemical properties file. The polyMesh folder is home to the mesh. Simple OpenFOAM meshes are generated by defining a blockMeshDict file (usually found in the system folder) which, when called by the blockMesh utility, generates the mesh as specified in the file. Here, we can specify the co-ordinates of the domain, and indicate the mesh blocks and desired number of cells within each block. Once we run the "blockMesh" utility, OpenFOAM populates the PolyMesh folder with "points", "Faces", "owner" "neighbour" and "boundary" files, where Points refers to a vector of cell vertices, faces being a list of all the existing faces, with each face being made up a list of vertices sourced from "points". Owner and Neighbour represent owner and neighbour cells respectively. The boundary folder describes a list of patches, that is being a list of faces grouped together to act as a boundary. E.g, an inlet boundary is a group of faces that have been formed into a patch.

SYSTEM FOLDER

The system folder contains utilities that are used to carry out mesh manipulation and control simulation cases. Some dictionary entries found here include the blockMeshDict, which houses the geometrical data, the controlDict where we can specify the time step, CFL criteria among others, and the fvSchemes where we can describe the choice of discretization, interpolation and reconstruction methods for field variables. A useful dictionary to highlight as well is the decomposeParDict which enables one to decompose a domain into a number of units which can then be run in parallel on different processors. This is a useful tool for running OpenFOAM cases in computer clusters.

2.5.3 POST PROCESSING

OpenFOAM comes with an open source visualization tool known as paraFoam, (a fork of paraview) which enables visualization of simulation results. Usually we tell OpenFOAM the time it should output our results; for instance we can say every 1×10^6 s of run time. We can then visualize these results and produce animations based on them. We can also perform numerous operations on the results with the help of various tools it comes with. We can also compute derived fields which is a function of two or more output fields.

2.6 CREATION OF CUSTOM SOLVER FOR DETONATION SIMULATIONS (rhoCentralFoamreact)

OpenFOAM consists of a significant amount of "solvers" which refers to a collection of C++ codes written and compiled to solve a given type of problem. These include among others, combustion solver, compressible and incompressible flow solvers, etc. Detonation waves involve a complex coupling of shocks and reactions, hence any detonation solver must be able to capture shocks effectively and also have the ability to solve for chemical reactions. There exist compressible flow solvers e.g., rhoCentralFoam, sonicFoam, etc. The details of these solvers are described in [Ope16]. In OpenFOAM5, however, there is not a flux evaluating solver for compressible flows with chemical reactions that can capture this phenomenon effectively. Of all the compressible flow schemes available, only the rhoCentralFoam solver evaluates the fluxes in a true finite volume manner. The rhoCentralFoam solver is written such that it can be employed for cases whose flow behaviour can be described by the Euler equations. However a solver was needed for the reactive Euler equations which describe the behaviour of detonation waves. Hence a new solver was created for that purpose which was named rhoCentralFoamreact. This needed to be compiled in all servers and or clusters where it was utilized. Starting off with the rhoCentralFoam solver already in existence, a copy of the C++ based codes that included the various header files necessary to successfully execute the application was created. Considering it was not intended to solve for full chemistry as the focus was in one-step governed reaction kinetics, there was need to modify the code executable to account for this. First the transport equation was included into our code which is given by the equation below;

$$\frac{\partial \rho Y}{\partial t} + \frac{\partial \rho u Y}{\partial x} = 0 \quad (50)$$

where Y accounts for fluid transport. To evaluate this, a shock tube case was run with the newly modified solver to see how it behaves with the inclusion of the transport term Y. This

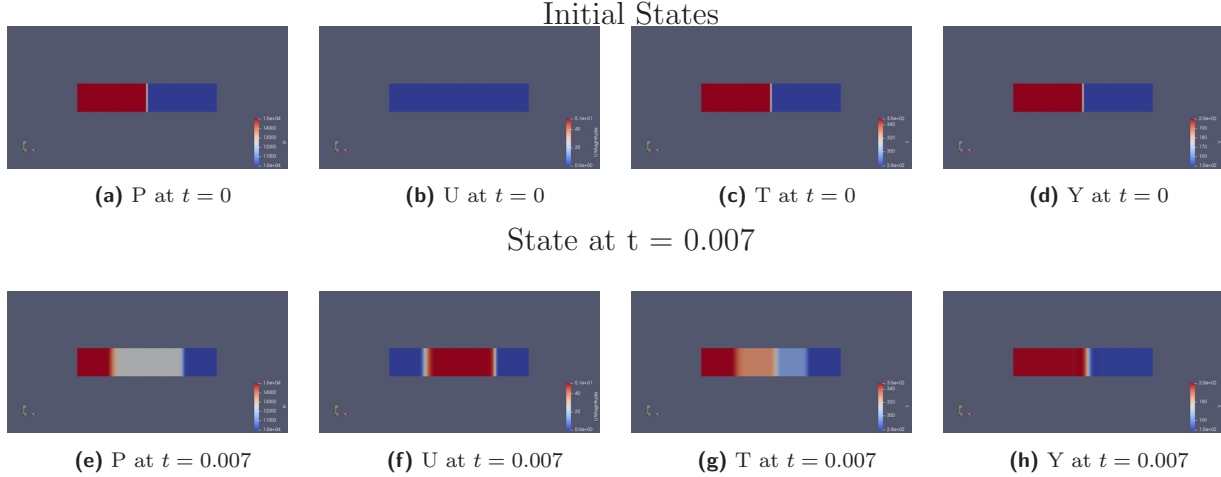


Figure 10: Inclusion of Transport Equation to rhoCentralFoam. Fig10h shows the advection of "Y" after some time

was done by modifying the source code to enable us solve for an imaginary property being convected through our domain, defined by the transport equation, and then compiling a new solver know as rhoCentralFoamnor using the openFoam "wmake" command. It can be seen that for a given transport quantity when solved using this solver, based on certain initial condition, the property gets advected through the domain, and the simulated result gives the expected behaviour as shown in figure 10. This inclusion was done such that the user has the option of employing the KNP or the KT scheme (its slightly more diffusive version) as implemented in OpenFOAM. The flux terms of the transport equation were evaluated using the scheme as well after interpolating cell values to faces following the implementation described in [KBS15]. With the solver modified to handle fluid transport, the reactive source terms were then included. Simulations were based on the one step Arrhenius kinetics equation, hence the source code was modified to account for that. The procedure as well as a brief overview of how the solver works has been enumerated in 7:

1. Create copy of rhoCentralFoam solver.
2. Create Y in the zero folder.
3. Rename source code and modify to include transport terms.
4. Modify createFields.H to include volScalarField Y to be obtained from 0 folder.
5. Modify "createFields.H" to include conserved volScalarField rhoY, ($\rho \times Y$).
6. Include concentration change and integrate rate law equations $rhoY * (e^{-(dt/rkA)*e^{-Ea*psi}} - 1)$ and correct boundary condition

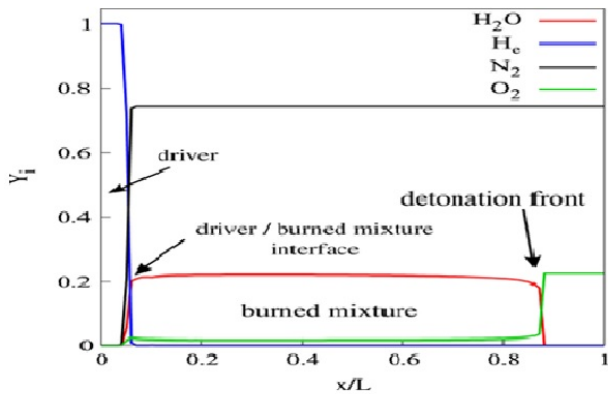
7. Compile solver by running "wmake" within the directory using a terminal(Ubuntu in this case) .

The new solver, rhoCentralFoamreact was then used to validate the KNP/KT scheme for detonation simulation, and to perform RDE simulations, as discussed in the following chapters.

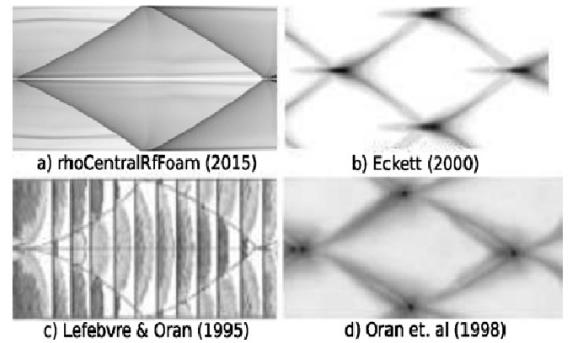
Chapter 3

VALIDATION OF THE KT/KNP SCHEME FOR DETONATION SIMULATION

The open source Computational Fluid Mechanics (CFD) package, OpenFOAM, as it is commonly known, is increasingly used and referenced. One drawback of the stock OpenFoam package is that the only finite volume numerical scheme available for the solution of the Euler equations in conservative form is the KT, and its variant, KNP, numerical scheme. This family of scheme is 2nd order accurate, and is based on the idea behind the Lax-Friedrichs scheme which does not involve the solution to a Riemann problem in order to evaluate the intercell fluxes. This is unlike the methods currently used in detonation research, which are mainly Godunov-type schemes with an approximate Riemann solver [Tor] at each cell interface. Common modern flux evaluation methods encountered include the class of Weighted Essentially Non-oscillatory Schemes (WENO), and schemes derived from the Harten-Lax-van Leer (HLL) method, such as the Harten-Lax-van Leer-Contact (HLLC). OpenFoam, with the KNP scheme, was recently used to simulate the two-dimensional structure of detonation waves in [MTE17b] despite having not been fully validated for the detonation simulation. Figure 11a and 11b show the 1D simulated detonation structure and the 2D simulated detonation cells obtained in that work. They created a solver known as rhoCentralRfFoam using the OpenFOAM framework which they used to simulate planar detonations in H₂-O₂-Ar, and also [MTE17a] simulated a two-dimensional detonation cellular structure in H₂-O₂-Ar mixtures with OpenFOAM employing full chemistry. While qualitatively correct, those results are based on a scheme that was not validated for detonation simulations. The validation of the KT/KNP scheme for detonation waves is done by simulating a 1D problem whose expected result is a pulsating detonation with a single oscillation mode. This is done



(a) Concentration vs space for a H₂-O₂-Ar Mixture



(b) Simulated Detonation cell sizes

Figure 11: Images are from simulation of detonation waves in [MTE17b]

by carrying out reactive and non-reactive validation studies. Useful metrics for detailed comparison and convergence studies are the oscillation peak pressure and period. Firstly, an examination of how it handles non-reactive flows with shock waves is performed, and thereafter, the 1-D dynamics of a propagating detonation wave is investigated.

3.1 NON-REACTIVE VALIDATION

3.1.1 CASE DESCRIPTION AND SET-UP

This approach was first implemented using a one-dimensional shock tube problem implemented within the rhoCentralFoam solver in OpenFoam. This is a compressible non-Riemann based solver. The case was set up building from the already existing shock tube case in the OpenFoam tutorial. The domain was set to be 0.1m long. This was meshed with 10000 cells with uniform cell distribution across the domain. A sketch of the domain with boundary conditions is shown in figure 12

3.1.2 BOUNDARY AND INITIAL CONDITIONS

A transmissive (zero gradient) boundary condition was applied on the sides for pressure, temperature and velocity, while the other sections of the domain, e.g., front face (relevant for 2D and 3D) were specified as empty to enforce a one-dimensional simulation in x . The density is evaluated using the perfect gas equation of state. Two regions of distinct, thermodynamic

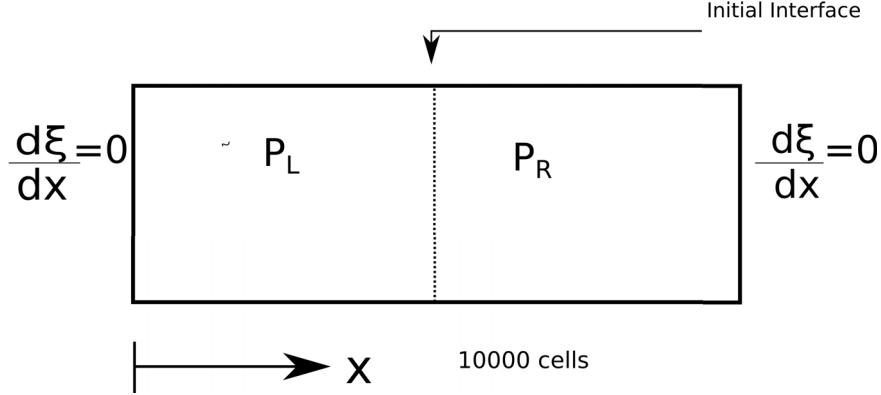


Figure 12: Case set-up for non-reactive validation (shock tube problem)

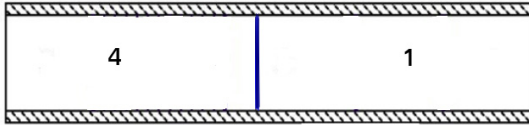


Figure 13: Initial State ($t = 0.000s$)

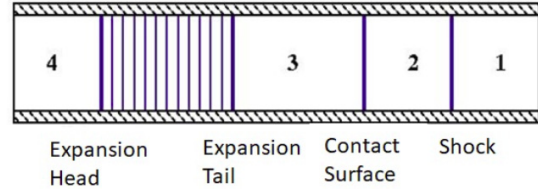


Figure 14: Simulated State ($t = 0.007s$)

properties were specified as initial condition: a driver and a driven section within the tube, separated by a contact surface. The initial pressure was 100 kPa and 10 kPa, initial velocity is 0 everywhere, initial temperature was set to 348.432 K and 278.746 K for the driver and driven ends respectively. The mixture inside the tube is treated as an ideal gas with zero viscosity, a molecular mass of mixture (M) of 28.98 g/mol and a heat of formation (H_f) of 2.544e+06 kJ/kmol. The performance of four different limiters: Minmod, Van Leer, Van Albada and the superBee limiters in OpenFOAM, are evaluated. The simulation was run for 0.007s and the initial and end states presented in fig 13 and fig 14).

3.1.3 RESULTS

Figure 16 presents images obtained from paraview, prior to and at the end of simulation cases for non-reactive validation. This indicates the four different states obtained in the solution to the shock tube problem. where figures 16a, 16b, 16c and 16d are the resulting solutions obtained by specifying P , U and T initially as shown in figures 15a, 15b and 15c. ρ is obtained from the ideal gas equation of state as stated earlier.

For each limiter, the maximum CFL number allowed in a computation was gradually increased, starting with an initial value of 0.05. For low enough CFL numbers, the solution

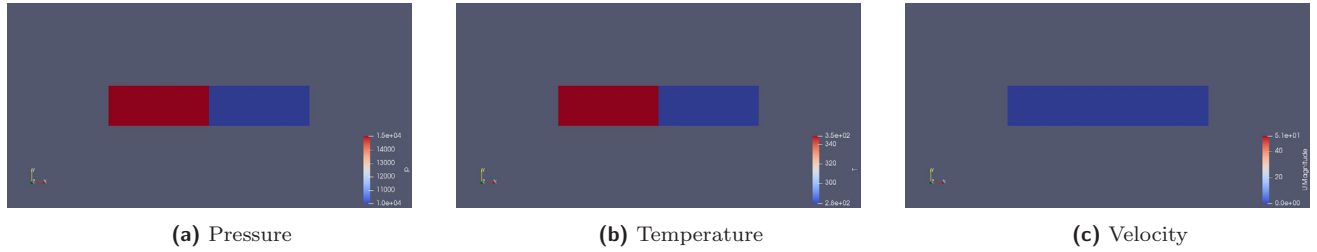


Figure 15: Initial condition for P , U and T ($t = 0.000$ s).

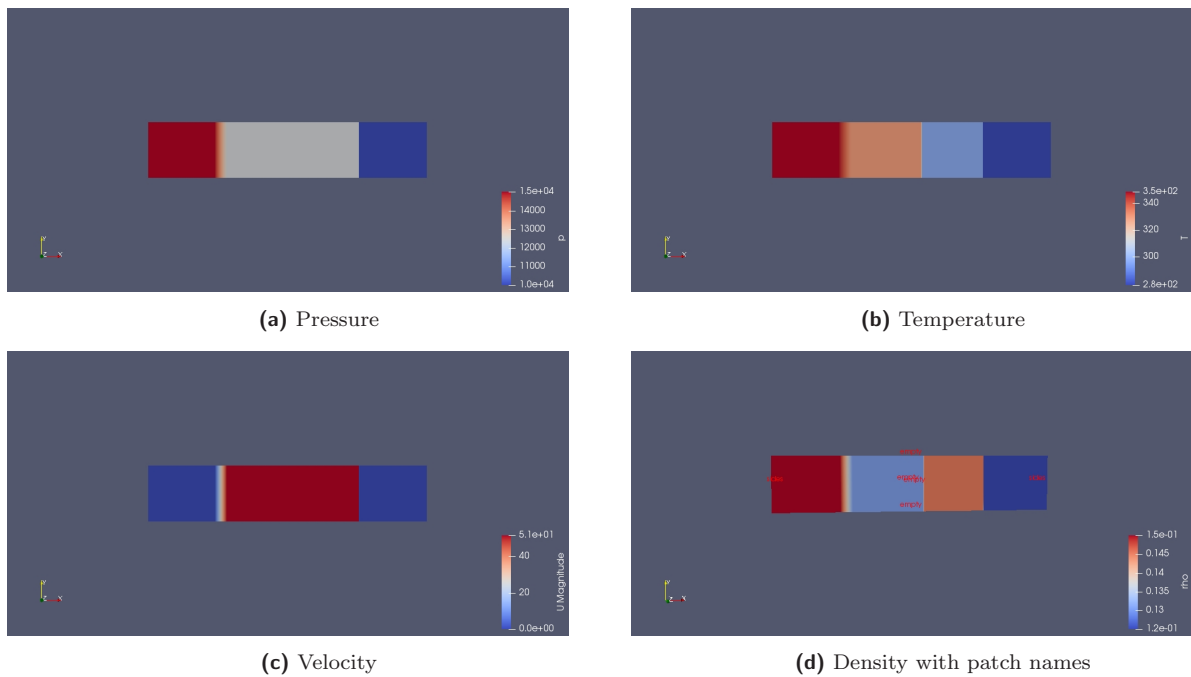


Figure 16: End state showing P , U , T and ρ ($t = 0.007$ s)

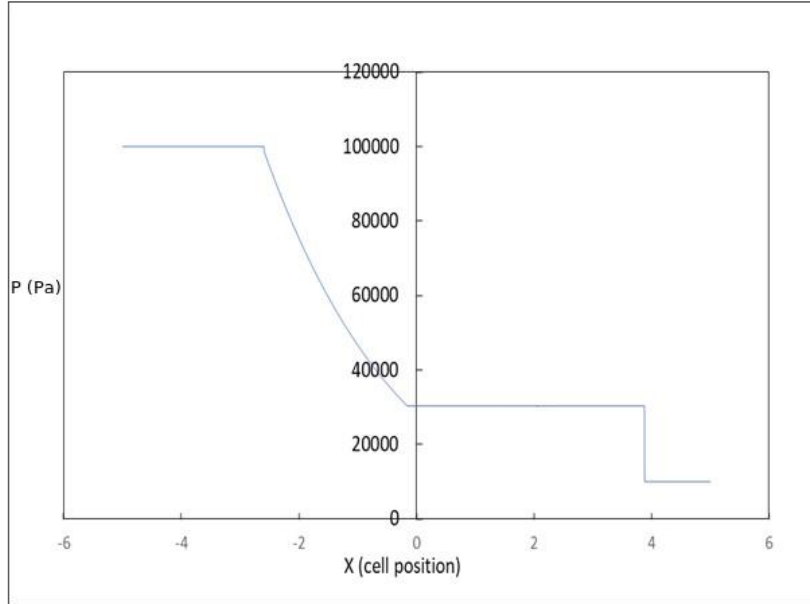
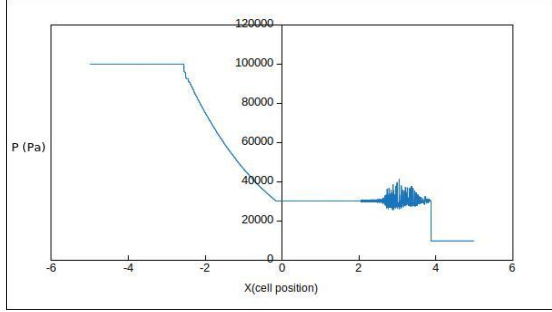
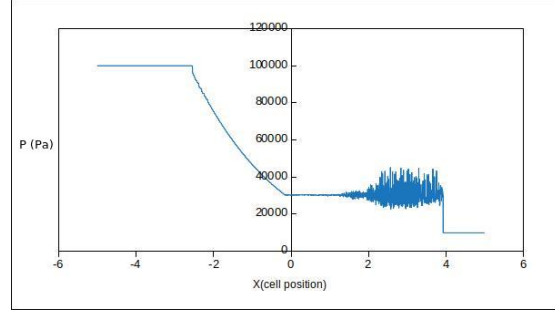


Figure 17: Pressure vs position for the shock tube problem using the minmod limiter $CFL = 0.09$

behaves as expected, with four apparently constant states: the low pressure state on the far right, the high pressure state on the far left and the two intermediate states around the contact surface. In figure 17, the distribution of pressure against space is shown at the end of the simulation, for the minmod limiter with a CFL number of 0.09. As the CFL number is increased further, non-physical oscillations are readily visible and degrade the solution, as shown in figures 18a and 18b, for the minmod limiter with CFL of 0.23 and 0.33. It is important to determine under which conditions those unphysical oscillations appear, since it is important to understand how the KT/KNP scheme performs when simulating unreactive flows, before it can be utilized for simulations involving detonation wave. Careful examination of the solutions obtained show these unphysical oscillations originate from the contact surface and propagate both towards the shock front and through the expansion fan. It can be noted that there are measurable oscillations even for cases that appear oscillation free, such as that shown in figure 17. The maximum amplitude of the oscillations, in the pressure field, after a simulation has been run, are taken as a metric for the level of degradation of the solution. As all the computed cases are at the same initial conditions, and for the same duration, Δt , the absolute disturbance maximum amplitude is used in dimensional scale. An example of a solution dominated by unphysical oscillations is shown for the case of superbee limiter in figure 19a and figure 19b, with $CFL = 0.05$ and CFL of 0.16. It is evident that

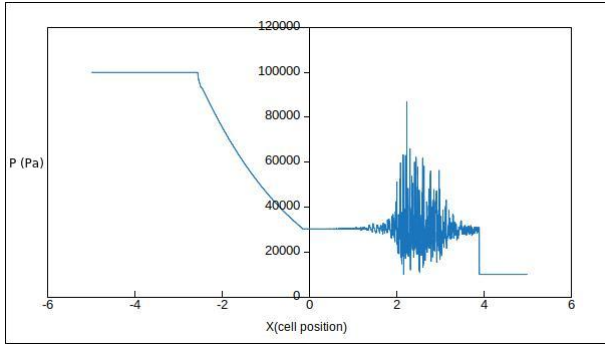


(a) Pressure vs cell position, (CFL = 0.23)(minmod)

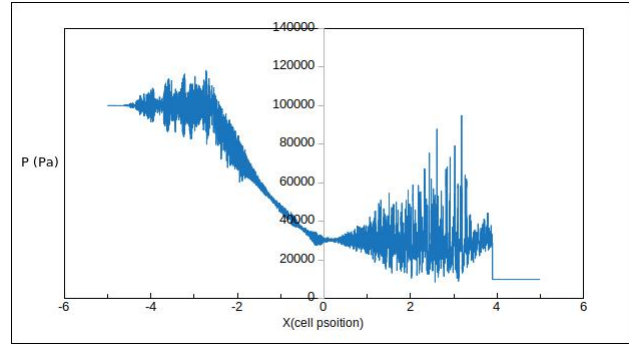


(b) Pressure vs cell position, (CFL = 0.33)(minmod)

Figure 18: Flow simulations at different CFL numbers



(a) Pressure (Pa) vs cell position, (CFL = 0.05)(superbee)



(b) Pressure (Pa) vs cell position, (CFL = 0.16)(superbee)

Figure 19: Flow simulations using superbee limiter, indicating intense amplitude of oscillation

the choice of limiter can be significant as even for very low CFL number as shown in fig 19a there exists large amplitude oscillations. In the case shown in fig 19b, the unphysical oscillations have even reached the right propagating shock and have started to influence the high pressure state on the left side of the domain. Thus, an examination of these oscillations and the corresponding effect changes in pressure ratios between the driver and driven pressures could have on them was performed. In figure 20, a presentation of the pressure disturbance amplitude as a function of CFL number for a pressure ratio of 10 for the 4 limiters considered is shown. For the test conditions evaluated, the minmod limiter is the best performer while the superbee limiter fails to accurately compute the solution at almost all the CFL numbers considered. There are easily recognizable outlier points, for example at CFL = 0.11 and 0.17 for the minmod limiter. Those correspond to cases where a numerical instability in the platform resulted in a failure of the code to complete the simulation. The source of this error has not been successfully identified and may not originate with the numerical solver itself. The van Leer and van Albada limiters perform only slightly worse than the minmod limiter, except above limiting CFL numbers of 0.12 and 0.22 respectively. Above these critical values, the algorithm fails to compute a solution to the problem that is not

dominated by those spurious oscillations. It can be seen that, even for the minmod limiter, to maintain an accuracy of 1% in the states on either side of the contact surface, a CFL number below a value of about 0.15 should be used. This critical CFL is very low. With more commonly used solvers in the detonation field, the CFL number is most often in the range of 0.5–0.8. The process was repeated for various pressure ratios, each time evaluating

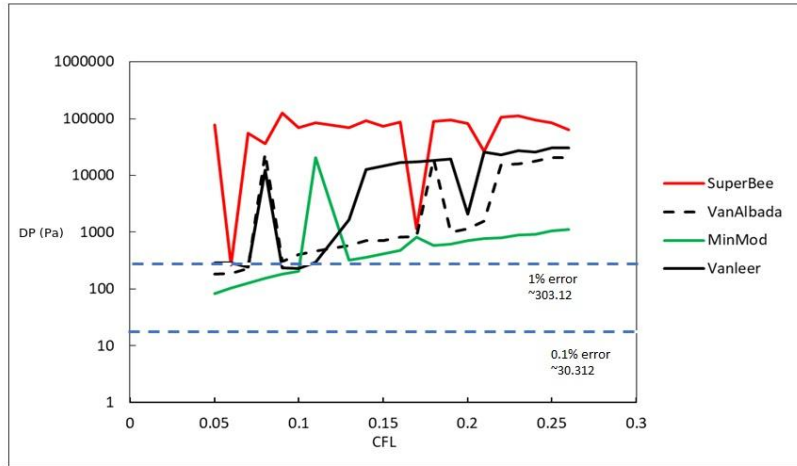


Figure 20: Disturbance amplitude DP (Pa) vs CFL number for the minmod, van Alabada, van Leer and superBee limiters for a pressure ratio of 10.

the amplitude of oscillation resulting from the numerical method. The results for a pressure ratio of 1.5 and 2 are shown in figure 21 and figure 22, and that for a pressure ratio of 5 presented in figure 23.

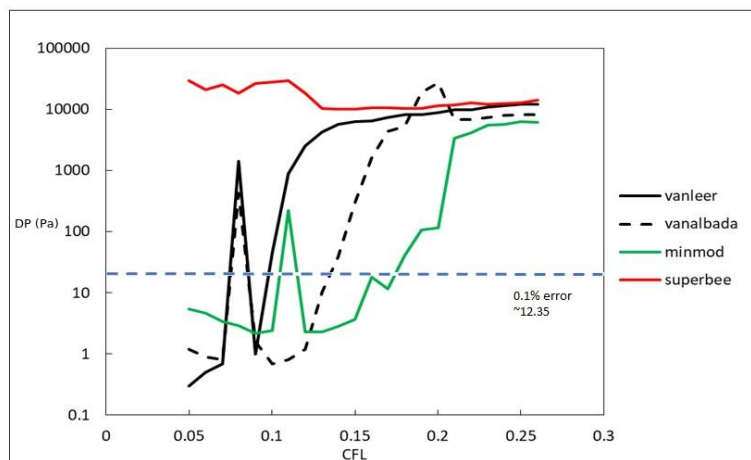


Figure 21: Disturbance amplitude DP (Pa) vs CFL number for the minmod, van Alabada, van Leer and superBee limiters for a pressure ratio of 1.5.

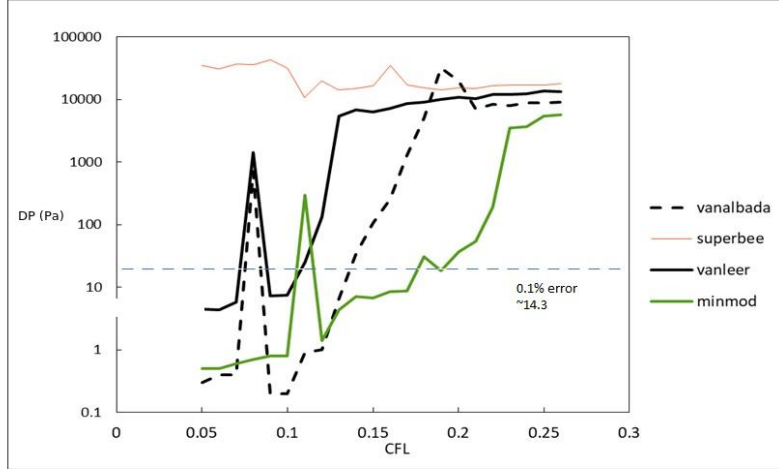


Figure 22: Disturbance amplitude DP vs CFL number for the minmod, van Alabada, van Leer and superBee limiters for a pressure ratio of 2.

3.2 REACTIVE VALIDATION

Having determined the conditions under which the KT/KNP scheme can reproduce the shock tube validation case, the simulation of reactive compressible flow can be evaluated. This is done by simulating a standard detonation case often used in the literature as a validation case, corresponding to a one-dimensionally unstable, over-driven detonation. The chemical rate parameters are $Q = Ea = 50RT_o$, adiabatic index (γ) = 1.2, detonation overdrive factor $f = \frac{D^2}{D_{CJ}^2} = 1.6$, where D_{CJ} is the detonation velocity at CJ point (\mathbf{D}_{CJ}), and k is the pre-exponential constant 1/s (k), which is arbitrary. It was chosen here to correspond to the combustion of methane and oxygen, such that the half reaction zone length ($L_{1/2}$) = 1.9×10^{-4} m. As described in chapter 2, the 1-step Arrhenius kinetics is used to model the reaction.

3.2.1 CASE SETUP DESCRIPTION

An over-driven, unstable detonation is simulated by considering the case of a piston pushing on the gas at a velocity higher than the CJ particle speed. The flow is simulated in the frame of reference of the piston. Two initial conditions were considered:

1. A small, high pressure and temperature initiation region is initialized near the piston and the dynamics of the initiation process are simulated;
2. The ZND structure is initiated near the piston and the wave instability is allowed to develop.

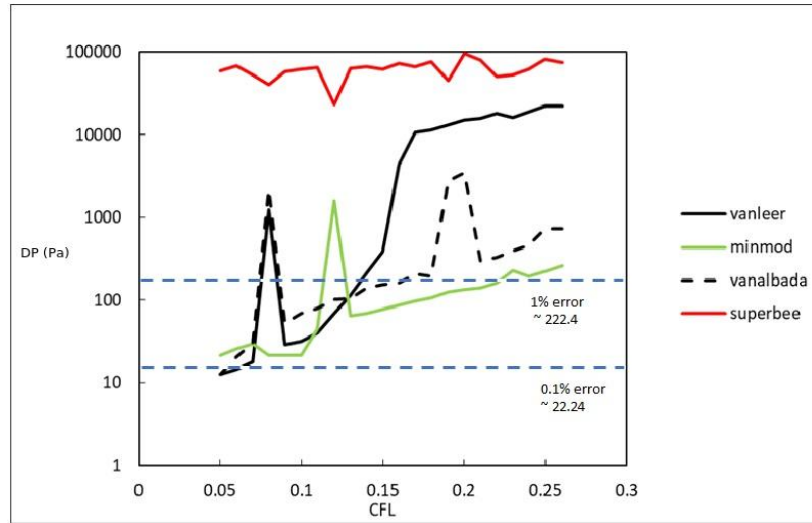


Figure 23: Disturbance amplitude vs CFL number for the minmod, van Alabada, van Leer and superBee limiters for a pressure ratio of 5.

The domain size was set to be a multiple of the half-reaction zone length, most often $200L_{1/2}$ (0.038 m in physical terms) to $600L_{1/2}$ (0.101410 m).

Boundary Conditions

The piston face is treated as a solid wall. The opposite end of the domain is an inlet providing fresh reactants flowing in at the imposed particle velocity, $U = -1811.6$ m/s

3.2.2 INITIAL CONDITIONS

Fig 24 shows the overall initial conditions. In a small region near the piston, a driver section was initiated.

INITIATING PISTON

The driver section was initially static ($U = 0$), with a high pressure of 100 kPa, and a temperature of 298 K. The heat release term, Y , was set to 4.2×10^6 KJ/Kg everywhere in the domain. The velocity elsewhere was set at the same imposed particle velocity $U = -1811.6$ m/s.

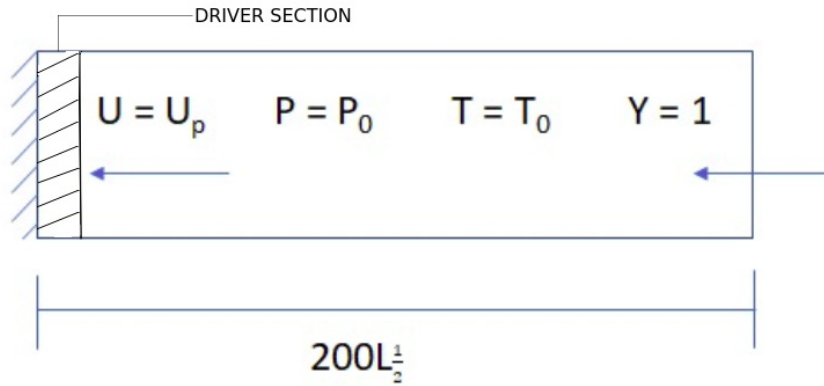


Figure 24: Case set-up for reactive validation (piston problem)

1-D ZND STRUCTURE

An alternative initiation method that bypasses the initiation transient associated with the localized high pressure and temperature driver section consists in initiating, near the piston, the 1D ZND profile of the steady wave and letting instabilities grow. The details of the ZND model can be found in [FD79]. In summary, the gas is compressed by a strong shock to a high pressure, the von Neumann spike, followed by chemical reaction. The reaction rate occurs within a finite time frame and over a certain physical extent, instead of in an instantaneous process. For a steady wave, a reaction progress variable can be used to describe the state points completely along the Rayleigh-Hugoniot curve. Thus, at each point in the reaction, the value of pressure, P , specific volume, v , and velocity, U can be obtained. An Octave code was written to integrate the 1D Euler equations for a steady detonation wave and calculate the physical extent between the front of the shock and the point where half of the reaction is complete, $L_{1/2}$. This integration is performed assuming a single Arrhenius chemical reaction rate with specified values of heat release Q and activation energy (Ea). This wave profile is then used as the initial condition within the driver section near the piston. The ZND wave profile was calculated with the same spatial grid extent as the computation considered, i.e. $L_{1/2}/\Delta x = 10, 20, 40$, etc. Sample results of the ZND structure are shown in figure 25 and the Octave code used to generate the ZND structure is reproduced in the appendices section.

3.2.3 SOLUTION METHOD

The 1D CFD Euler calculations were performed with varying grid spacing corresponding to fractions of the half-reaction zone length, i.e. $\Delta x = L_{1/2}/C$, where $C = 10, 20, 40, 80$, etc up

to 100. The OpenFOAM postprocessing utility sampleDict was used to produce 1D results data sets (remember that OpenFOAM always, by default, produces 3D data sets) that were themselves post-processed using a python script to extract the shock pressure as a function of time and space.

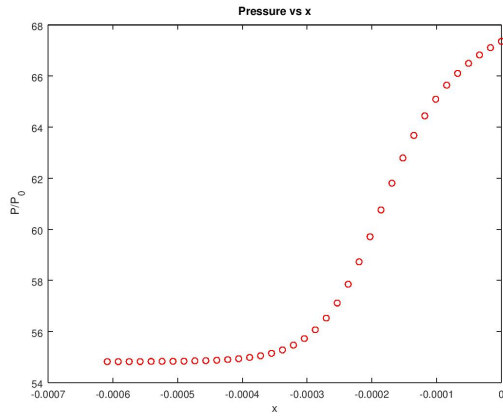
3.3 RESULTS

The results obtained from the piston set up and the ZND initiated cases are presented. The domain was also extended to $600L_{1/2}$ (0.101418 m) from the initial $200L_{1/2}$ used for the piston problem. In each simulated case, the mesh size was adjusted to reflect the desired number of cells per half reaction zone length, e.g 12000 cells in x gives 20points per $L_{1/2}$ for the mesh $600L_{1/2}$ long. 6000 cells corresponds to 10 points, etc. The peak shock pressure, (P_{shock}) is plotted against time and space. It can be seen that the scheme reproduces the oscillatory structure of a detonation wave as shown in Figures 27, 29, 31, 33, 35, and 37 which shows results obtained for 10, 20, 40, 60, 80 and 100 points per half reaction zone length initiated using results from the ZND code with a CFL number of 0.1. After a certain time, t , the detonation wave hits the domain boundary, at which point it no longer is a detonation wave, hence the reason for the artifact seen towards the right end of the domain as in 34 and 38 for plots against time.

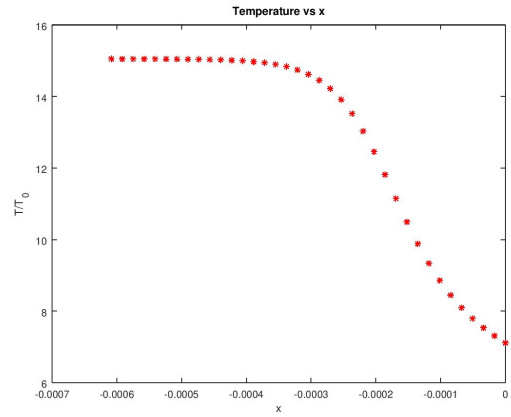
The domain length was doubled thereby allowing the detonation wave travel longer. This was done to see if the propagating detonation wave would result to a single mode of oscillation at a distance further away. This however was still not the case as can be seen in figures 39 and 40 which show dimensionless pressure profiles plotted against space and time.

3.4 ANALYSIS AND CONCLUSION

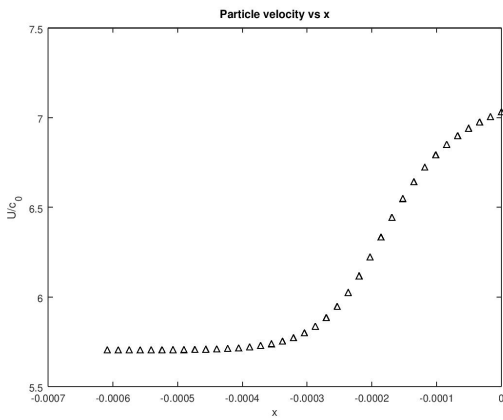
For the simple, overdriven detonation problem, the KNP scheme recovers the oscillating solution. Findings from a number of linear stability analysis [BM95, NNL00] show that the resulting ZND solution gives a single mode instability for the chosen values of heat release (Q) and Ea as shown in figure 41. However, a single mode of oscillation is not obtained, and even after increasing the resolution up to 100 cells per half reaction zone length it still fails to converge to a single mode. Instead, a profile that is rather irregular with multiple modes is the resulting solution. Figure 42 shows the comparison of the oscillation peak pressure (P_{shock}) for various other schemes as seen in [NNL00]. A comparison of the KT/KNP scheme, with other schemes is made. It can be noted that the peak pressure values obtained for each case varied with space, with the maximum values peaking above 100 kPa (≈ 108 kPa), quite



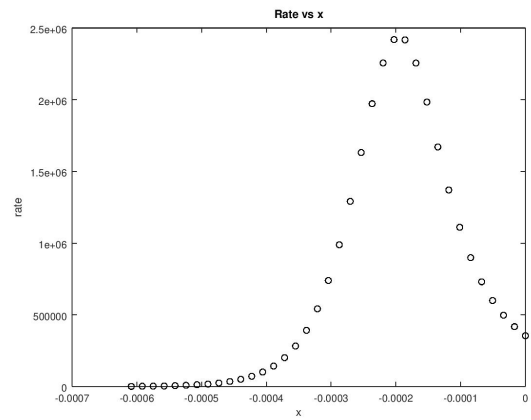
(a) Pressure, P against cell position in space



(b) Temperature, T against cell position in space

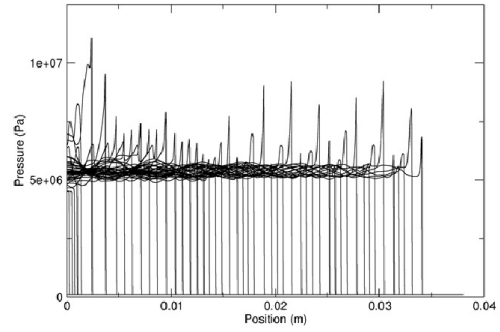
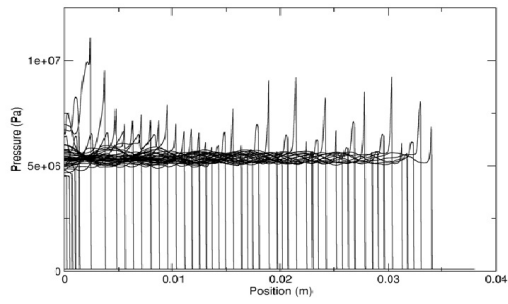


(c) Particle velocity, U_p against cell position in space



(d) Reaction rate against cell position in space

Figure 25: Result from octave codes for ZND model



Detonation Pressure profile for $\Delta x = L_{1/2} / 10$ and $\Delta x = L_{1/2} / 20$ for a domain $200 L_{1/2}$ long

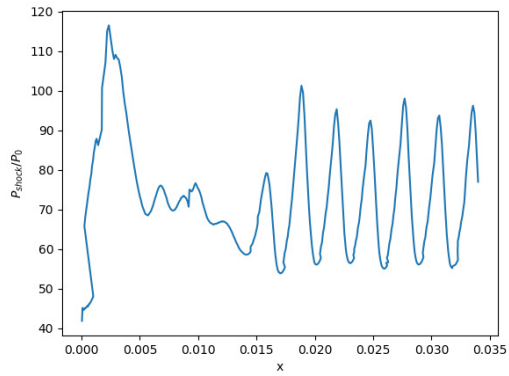
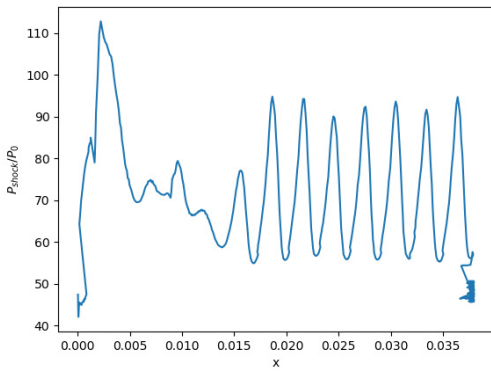


Figure 26: Shock peak pressure vs x axis for $\Delta x = L_{1/2} / 10$ and $\Delta x = L_{1/2} / 20$ for a domain $200 L_{1/2}$ long

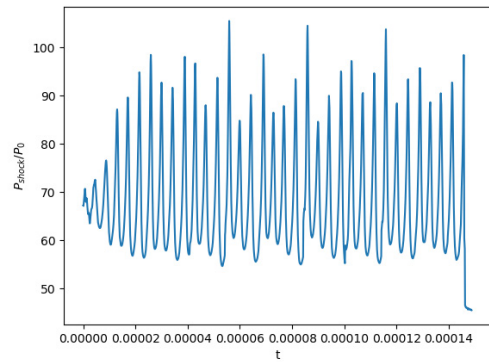
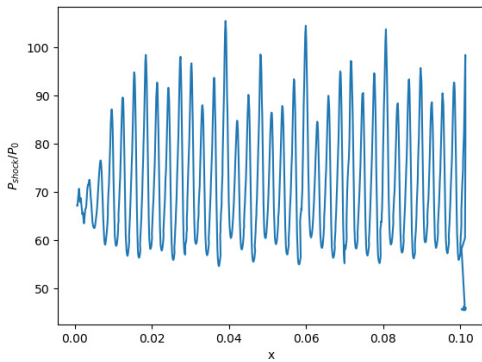


Figure 27: P_{shock}/P_0 vs position for $\Delta x = L_{1/2} / 10$ for ZND initiated case ($600 L_{1/2}$ long).

Figure 28: P_{shock}/P_0 vs time for $\Delta x = L_{1/2} / 10$ for ZND initiated case ($600 L_{1/2}$ long).

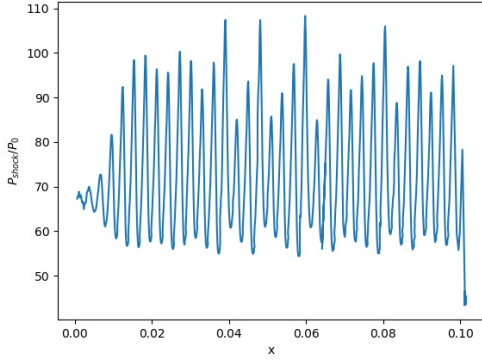


Figure 29: P_{shock}/P_0 vs position for $\Delta x = L_{1/2}/20$. (ZND).

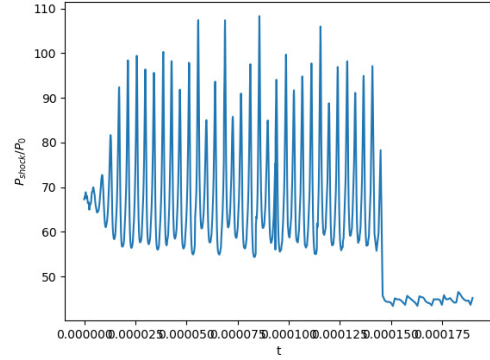


Figure 30: P_{shock}/P_0 vs time for $\Delta x = L_{1/2}/20$. (ZND).

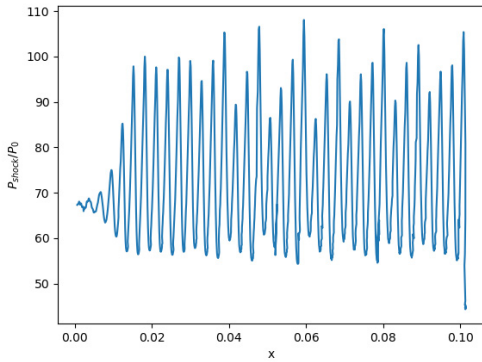


Figure 31: P_{shock}/P_0 vs position for $\Delta x = L_{1/2}/40$. (ZND).

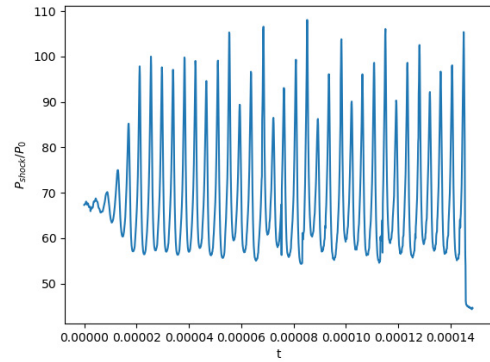


Figure 32: P_{shock}/P_0 vs time for $\Delta x = L_{1/2}/40$. (ZND).

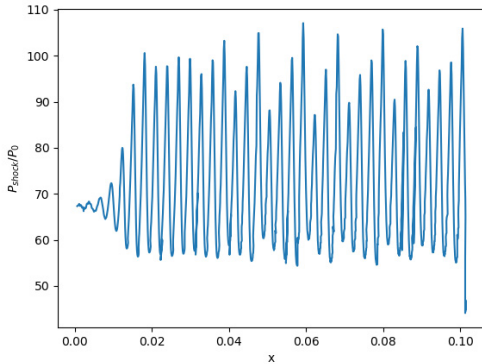


Figure 33: P_{shock}/P_0 vs position for $\Delta x = L_{1/2}/60$. (ZND).

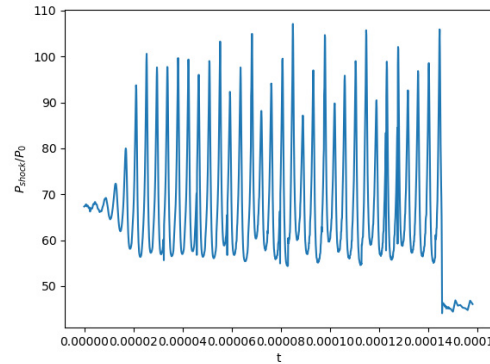


Figure 34: P_{shock}/P_0 vs time for $\Delta x = L_{1/2}/60$. (ZND).

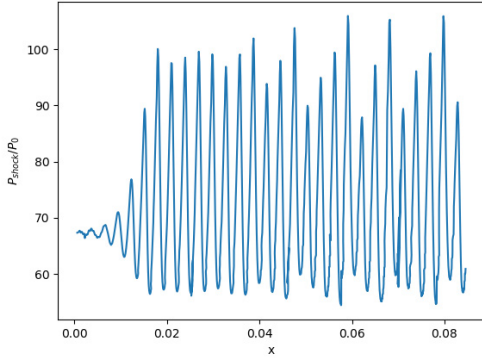


Figure 35: P_{shock}/P_0 vs position for $\Delta x = L_{1/2}/80$. (ZND) .

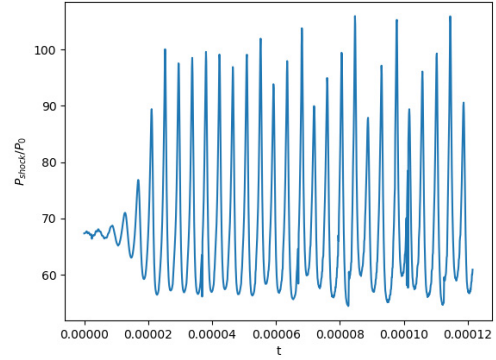


Figure 36: P_{shock}/P_0 vs time for $\Delta x = L_{1/2}/80$. (ZND).

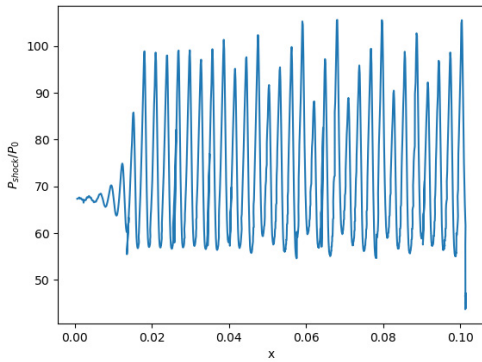


Figure 37: P_{shock}/P_0 vs position for $\Delta x = L_{1/2}/100$. (ZND) .

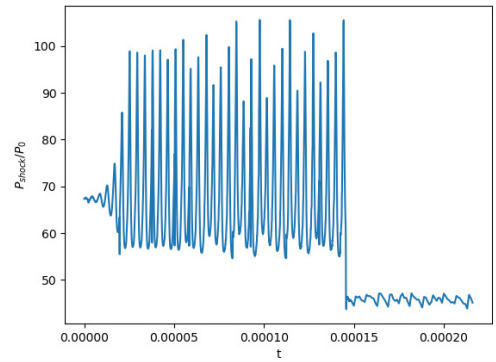


Figure 38: P_{shock}/P_0 vs time for $\Delta x = L_{1/2}/100$. (ZND).

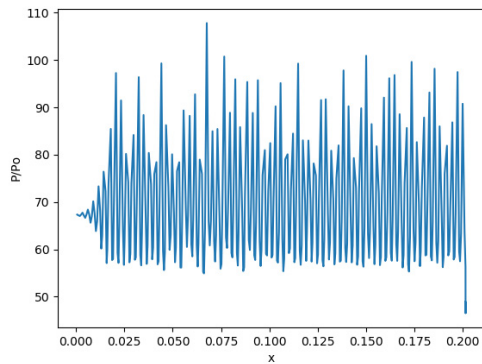


Figure 39: P_{shock}/P_0 vs position for $\Delta x = L_{1/2}/80$ for an extended domain. (ZND) .

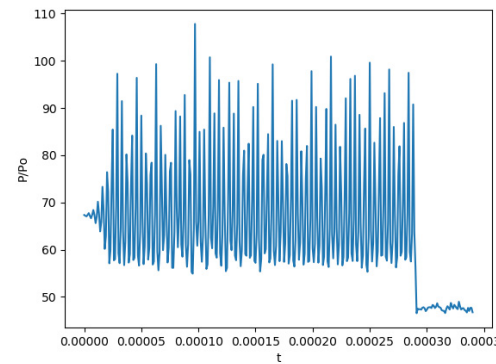


Figure 40: P_{shock}/P_0 vs time for $\Delta x = L_{1/2}/80$ for an extended domain. (ZND).

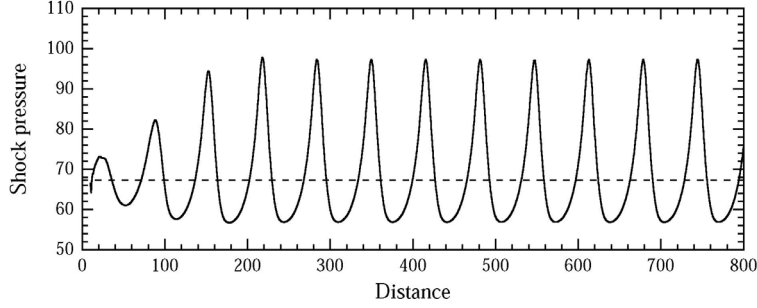


Figure 41: Image shows single mode pulsating detonation [NNL00].

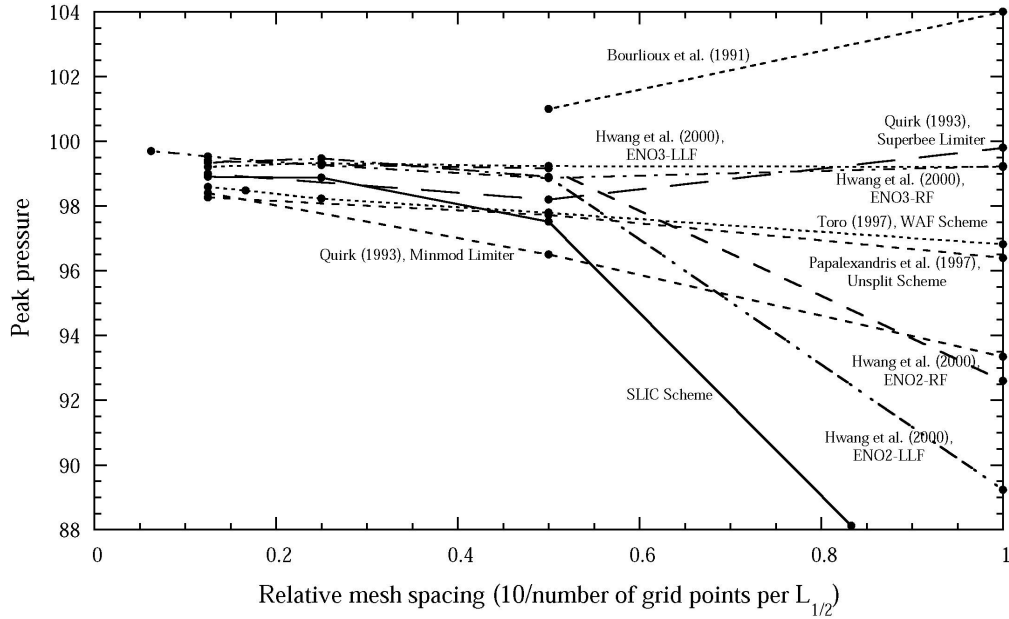


Figure 42: P_{shock}/P_0 vs Number of points per $L_{\frac{1}{2}}$ [NNL00].

a high value considering that most of the schemes discussed in [NNL00] show a convergence to a peak pressure value of ≈ 98.6 kPa. Out of curiosity, a reduction in the CFL number for some of the simulation cases to 0.05 was done, and test cases carried out, to see if there would be a different result; that however was not the case. In conclusion, the ZND solution of a detonation wave solved employing the KNP scheme cannot recover the regular, single mode, periodic oscillation for an overdriven pulsating detonation with an overdrive factor of 1.6, which is usually the case with numerical schemes employed in problems that deal with hydrodynamic-chemical reaction problems, used for the detonations studies, e.g., the SLIC scheme [NNL00].

Chapter 4

SIMULATION OF COLD ENGINE START

4.1 PROBLEM DESCRIPTION/CASE SET UP

The evaluation process is continued by studying the RDE in this section. We see that it is possible to obtain a circumferentially propagating detonation wave within an annular combustor by flowing combustible mixtures through the chamber as demonstrated in [BLWC13] and [Sha15]. Figure 43 presents a visual representation of the concept as applicable to rocket propulsion, and figure 44 shows a rotating detonation wave propagating circumferentially along the chamber. Though the KT/KNP scheme has not performed as we would expect it to in the 1-D validation, it is possible that the detonation structure details are irrelevant to 2D RDE simulations. The suitability of OpenFOAM is also evaluated for these simulations.

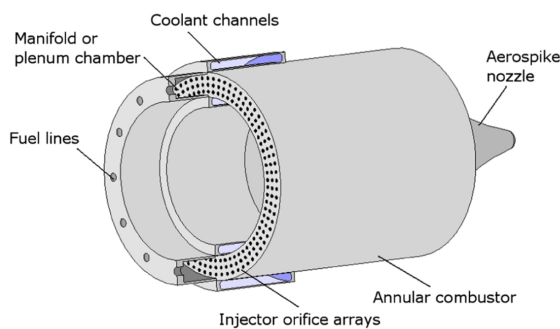


Figure 43: Rotating Detonation Engine [LB14]

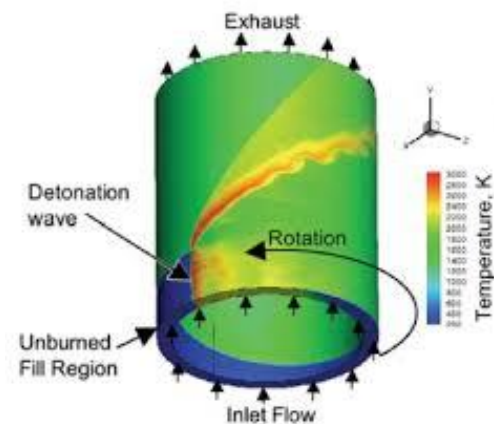


Figure 44: Representaion of an RDE [Sha15]

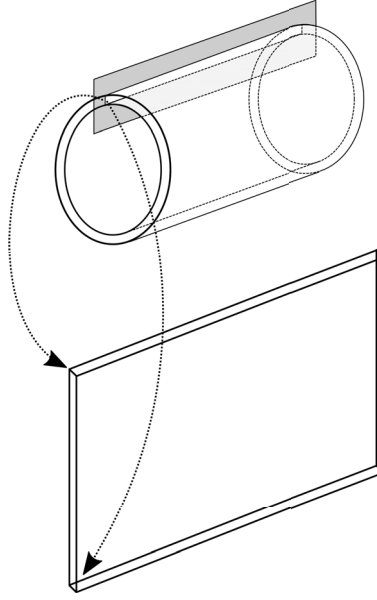


Figure 45: Image shows unwrapping of an RDE (3D to 2D).

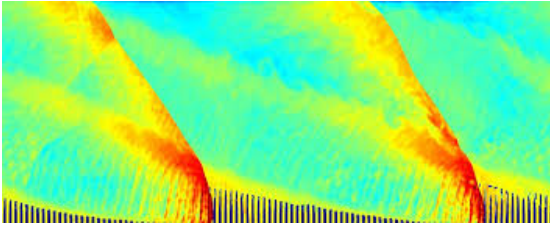


Figure 46: 2D RDE representaion
[TSS⁺14]

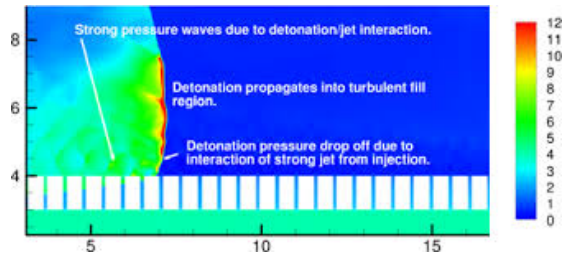


Figure 47: 2D RDE representaion
[KS13]

For ease of analysis, it is common to see simplified, 2-D simulations of the RDE phenomenon in literature. Towery et al. [TSS⁺14] for instance, show an unwrapped 2D model of the detonation wave. This approach is elaborated upon by presenting fig 45, where the unwrapping process has been demonstrated. This approach minimizes the resource requirements needed for such simulations, and simplifies the analysis involved. Hence a 3D combustion channel can be reduced to an "unwrapped" 2D version for analysis. Figures 46 and 47 show similar approaches as employed in RDE simulations in [TSS⁺14] and [KS13].

4.1.1 MESH GENERATION

The RDE mesh was developed from that used for the validation processes. This adaptation was undertaken because it is desired to simulate the injection of an RDE, which in our model involves a premixed fluid being injected into a combustion chamber from a plenum. To achieve this, the domain was taken as it were and modified the blockMeshDict found in

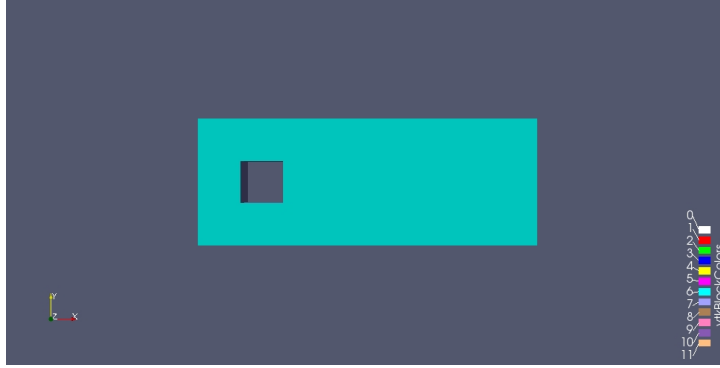


Figure 48: 2D Mesh with one obstacle

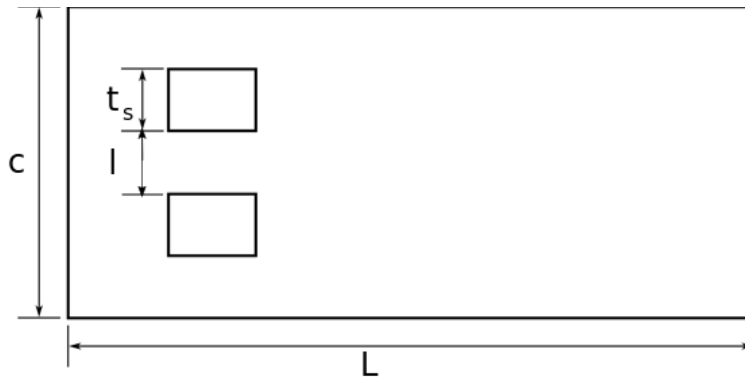


Figure 49: Sketch of domain with 2 obstacles

the system folder of our built case, by adjusting the vertex points to include a square-shaped void 0.05 m wide, (referred to as obstacles going forward) placed at a distance a little further away from the upstream section of our computational mesh. The resulting domain is shown in fig 48. The obstacle is to serve as the solid boundary within the injection cross sectional area, and the spaces below and above the void serving as the injection flow part (orifice in this case). Using this mesh, and by employing similar approach as described above, the vertex points were further adjusted to include two and three obstacles as shown in figures 49, 50 and 52, such that they are placed above each other, with the fluid flow part in-between the obstacles as desired. It can be seen that this has the effect of increasing the number of blocks and faces in the resulting mesh data written into the constant folder in OpenFoam, when the "blockMesh" command is run through a terminal. It can be seen that the number of blocks generated when the "blockMesh" command is run on OpenFoam increases as the number obstacles are increased. This results because the mesh is such that there are strips of blocks extending from the inlet region to an obstacle, and then from behind it to the end of the domain (the outlet) for each case. The OpenFoam command "paraFoam-block" was run to generate images of the number of mesh points/vertices as well as the faces to help visualize

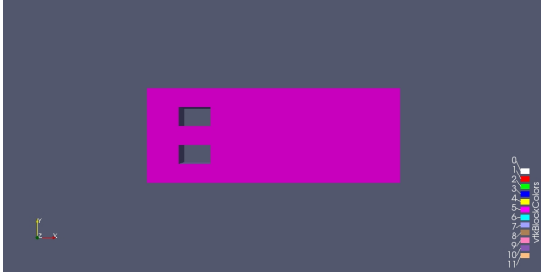


Figure 50: 2D mesh generation with two obstacles

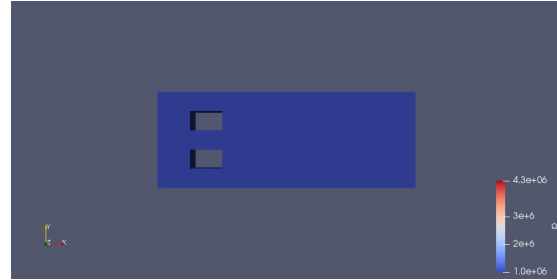


Figure 51: Initial pressure for the case

these points. The obstacles were extended from the boundaries for the case with 3 walls, hence the reason it looks a little different. Details of these are found in figures 52 and 53. The mesh vertices were generated, for the 3 obstacle cases above, by hand, so that a program

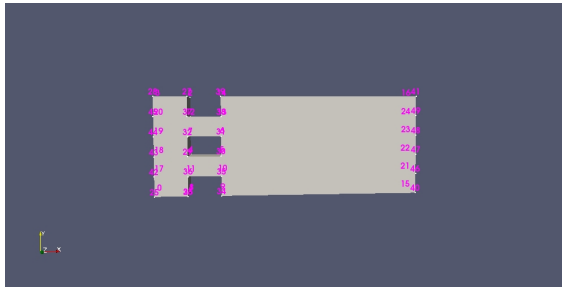


Figure 52: 2D mesh with three obstacles

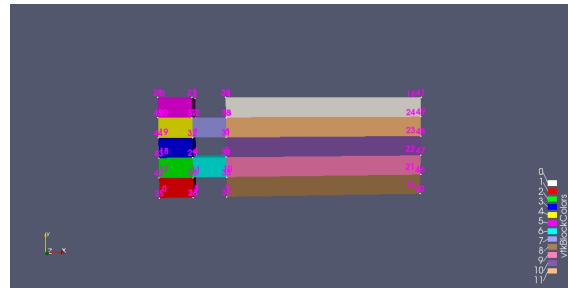


Figure 53: Image showing mesh blocks(11 in this case)

can be written that determines the obstacles width, for a choice of domain configuration, by taking in as inputs the number of desired obstacles, the void dimensions, and the orifice diameter(or width). Certain trends were established through this. For instance, we see that the total number of points needed to generate a mesh expressed as a function of number of obstacles (n) is $4 * (2 * n + 2)$, where n is the desired number of obstacles (solid walls between orifices). It can also be seen that the total number of blocks needed in the mesh is given as $3 * (2 * n + 1) - n$. A block in OpenFoam is described by eight(8) points. Starting from the origin, a numbering system was chosen that respected OpenFoam's format for face description (e.g outward pointing normal for faces) and generated an 8 column matrix, which was then populated using a code written in Octave, depending on the desired number of obstacles. Each face therefore is a resulting combination of four numbers, from the first four elements of the matrix (representing the point where $z = 0$) or from the last four column elements in each row representing the z direction ($z \neq 0$). Using this, the mesh input file, blockMeshDict, can be generated to be used with the blockMesh command. Figure 54 shows the generated meshes for the case with 3 obstacles in octave. The code was improved upon

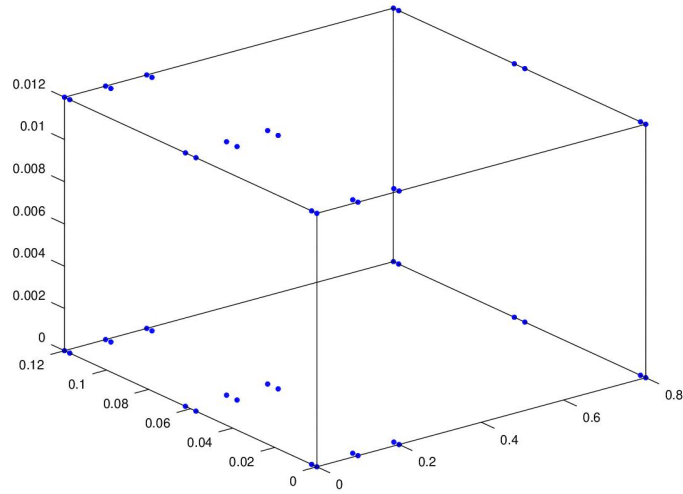


Figure 54: Vertices for 2 obstacles using octave code.

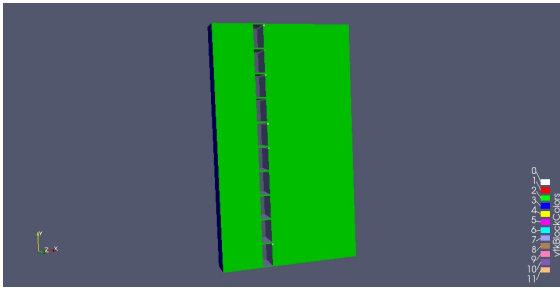


Figure 55: Image of mesh with 10 obstacles (code generated)

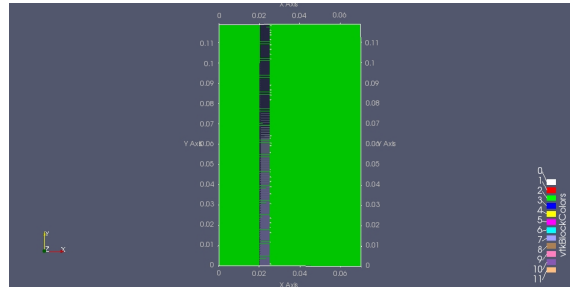


Figure 56: Image of mesh with 100 Obstacles (code generated)

by making it capable of grouping the faces into Hexagonal blocks, and also generate a text file with the same format as the custom blockMeshDict, OpenFoam’s text file for mesh data, which we can use to generate the PolyMesh folder, for a simulation case. The text file should be copied into the system folder in the case directory. The updated code behaved as expected and in figures 55 and 56, images showing cases with 10 and 100 obstacles generated using it are presented. The obstacle width is given as obstacle thickness (t_s) = injection width (l), where n is the chosen number of obstacles, c is the unwrapped circumference (c) and l is the injection width. To achieve periodicity of the unwrapped 2D domain geometry, it was specifically ensured that for each mesh gotten using this code, the injection width at the top-most and bottom-most parts of the domain sum up to l (one injection width) (i.e $l = \frac{1}{2}l + \frac{1}{2}l$). The sum of all the injection widths (orifices), l , give the total orifice area of the RDE, in 2D.

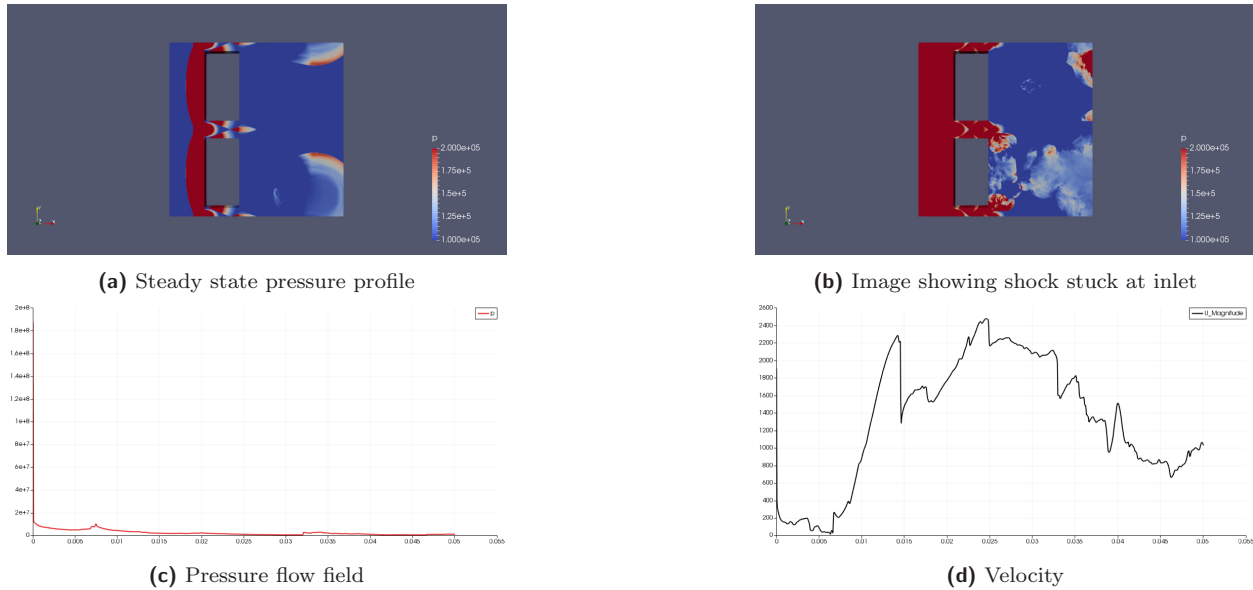


Figure 57: A resulting flow field using engine A.

4.1.2 GEOMETRY SETUP

The simulations in this section are based studies on two engine configurations, an imaginary engine with equal width and span, which has been named "engine A", and then an actual configuration from the literature. Using the assumed engine configuration, preliminary simulations were conducted to determine the effect ramping up the injection velocity (and hence mass flow rate) will have on the flow field of an operating RDE, beginning with 100 m/s in steps of 100, up to 700 m/s. These were run on the Compute Canada Cedar and Graham cluster, using a 10 points per $L_{1/2}$ refined mesh. The initiation of a detonation wave was attempted in some cases. Two observations are worthy of note using the assumed engine:(1) It can be seen that there is a pressure rise in the inlet section of the domain resulting from a reflected shock travelling back to the inlet, and "sticking" there, which is completely unphysical. This was the case even after ramping up the velocity, and initializing the domain with lower pressures set at the discharge end with the hope that this shock gets "sucked" in at some point.(2) It is important to stabilize the flow field before initiating a detonation. A fixed value inlet with a zero gradient outlet condition for the field variables was imposed as boundary condition. Considering that the exact reason for the unphysical behaviour was not known, it was also thought necessary to examine the boundary condition. Some obtained results for a case with injection velocity of 700 m/s are presented in Figure 57, where it is seen that the pressure and velocity profiles indicate no detonation wave(s) present.

4.1.3 ACTUAL ENGINE GEOMETRY

This case is based on the works of J. Kindracki and P. Wolanski in [KWG11]. They carried out an experimental study of the rotating detonation engine with most necessary details reported. It has an internal diameter of 38 mm. The length of the channel was varied from 23-50 mm; the stagnation pressure and temperature are 700 kPa and 293 K respectively. Fuel was injected through an orifice of diameter between 0.7-1 mm, while oxygen came in through a 0.5-1 mm diameter orifice. For the purpose of our simulations which are based upon premixed reactants within the combustion chamber, using the same stagnation conditions, it was assumed that the total injection diameter was 8mm (l), and the span was 50 mm. Using this l value, and making use of our mesh-generator code, it is possible to generate a mesh for n number of obstacles of our choice. The simulations performed were based on 10 injectors given a total injection area of $0.08 \times h$ m², and neglecting thickness (for 2-D), h as in our case, 0.08 m. Each injector area has a throat of 0.008 m. We will refer to this engine as the Wolanski engine going forward. By assuming a 1-D converging diverging nozzle configuration problem, we can make useful assumptions that aide engine modeling.

4.1.4 AREA RATIO

Having successfully devised a means to generate n number of obstacles, one may want to know the ratio between the injection area and the total area of the inlet, including the obstacles. This is a very useful metric in compressible flow problems. It can be seen that for 2D cases, this becomes a length ratio. Of interest usually is the mass flow rate, \dot{m} for any condition of flow, including situations where the flow is choked at the orifice. For such cases, we can define a throat t , and its area A_t , known as throat area. We can determine the throat area using 1D isentropic flow. In actual terms, since we are considering 2D cases, we are most interested in the throat width. Given the width of an orifice or nozzle, and the total number available, we can determine the total injection width. The area of the unwrapped RDE with width (circumference) c , for a given annular thickness δ , is given as $A_{RDE} \times \pi \times c \times \delta$. Hence the area ratio becomes,

$$\phi = \frac{\text{AreaofThroat}}{\text{AreaofRDEAnulus}} = \frac{A_t}{A_{RDE}} = \frac{l}{c} \quad (51)$$

This is important since by considering that mass flow rate, $\dot{m} = \rho \times U \times A$, where A refers to the cross sectional area of flow, we can compute the injection velocity, U for choked flow, which we can specify as input in the initial condition folder, (named "0") for OpenFoam. We can also determine the area ratio needed to support a given mass flow rate.

4.2 BOUNDARY CONDITIONS

Our geometry consists of 5 main regions where we can expect fluid-boundary interactions; Top, bottom, obstacle, inlet and outlet. These interactions are usually modelled by imposing boundary conditions which reflect the physical behaviour of the region, and hence must make realistic choices. The mode of operation of our engine is thus; Fluid flows into the chamber through the inlet boundary at certain conditions, undergoes an expansion process driven by a detonation wave within the chamber, and then exits the chamber via its outlet boundary end. It is required that these two regions are the only pathways for entrance or exit of fluid particles through the domain. Of mention again at this point is the fact that OpenFoam meshes by default are in 3D. Hence one must define two other boundaries, front and back which represent the front and rear faces of our domain. However a boundary condition known as "empty", in OpenFoam lingu, is imposed which enforces a wall boundary condition on the front and back faces, and also guaranties a 2d simulation. A periodic boundary condition, (known in OpenFoam terms as cyclic boundary condition) is imposed at the top and bottom boundary, such that whatever flow in the Y direction "reappears" in the domain from the other end. We want the obstacles to act as solid walls with no fluid flow through, as well as have the ability to reflect off shocks that bounce at them, hence the slip conditions are applied on those. In using OpenFoam and the custom solver, rhoCentralFoamreac, for these simulations, it is required to specify boundary conditions for the following four (4) thermodynamic properties (pressure, P , temperature, T , velocity, U and heat release, Y). It is also assumed that the engine is constantly fed with combustible reactants, hence the reason for imposing the fixed value BC. The outflow BC is a linear extrapolation with zero gradient $\frac{d\zeta}{dx} = 0$, where $\zeta = \{P, T, U, Y\}$ These boundary conditions were tested with a simple case involving a shock travelling into the domain which encounters an obstacle (or obstacles) within it. Figures 58 and 59 show the pressure profile of a reflected shock off for test cases having one (1) and two (2) obstacles, with slip boundary condition imposed which is expected. However this shock travels back to the inlet and "sticks" to it, leading to a constant pressure rise in the region. This result, together with that from the assumed engine configuration above, implied that it became necessary to explore other conditions, particularly for the inlet boundary. Some of the cases considered are described below.

4.2.1 FIXED VALUE BOUNDARY CONDITION

This implements the Dirichlet condition where face values are defined according to a specified reference value: $\phi_f = \phi_{ref}$. The syntax is defined in [Ope16] and was widely used for the

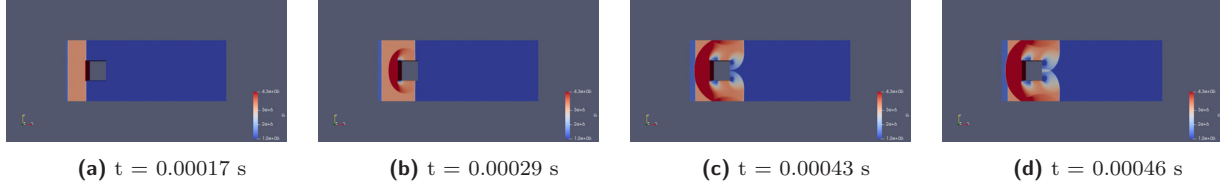


Figure 58: Pressure flow fields in the "slip" BC test with 1 obstacle

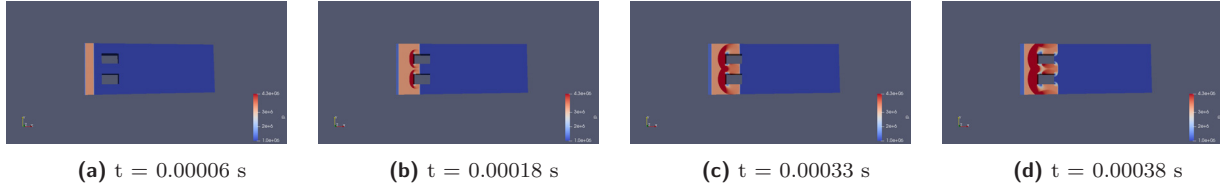


Figure 59: Pressure flow fields in the "slip" BC test with 2 obstacle

validation process. It is assumed that the mixture is injected into the engine at a certain (fixed) thermodynamic state, hence the reason the fixed value BC was imposed.

RESULT

Using this BC results in a unphysical behaviour in which a reflected shock from the obstacles travels to the inlet patch and remains there as shown in figures 57c and 61b. This raises the pressure, and affects the temperature and velocity at this boundary patch during simulation.

4.2.2 INLET-OUTLET BOUNDARY CONDITION

This is usually applied in situations where one wishes to account for back flows. It applies a zero gradient condition to the forward moving flow and then applies a specified (fixed value) constant to the backward flow. Figure 60 gives a representation of this. This was applied at the inlet for the pressure field, and for other fields as well.

RESULT

The inlet-outlet performed more like the fixed value boundary condition, failing to provide any outflow in the inlet boundary. Thus the reflected shock off the obstacle travels back towards the inlet and remains there.

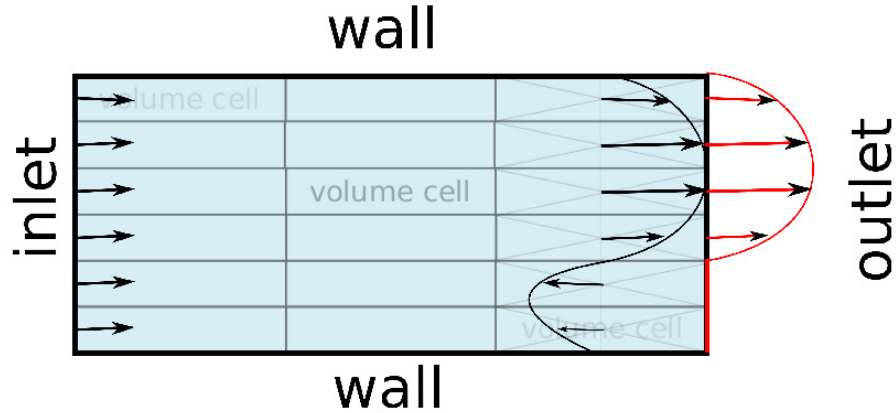


Figure 60: InletOutlet boundary condition [Cfd]

4.2.3 TOTAL PRESSURE/TEMPERATURE BOUNDARY CONDITION

The total pressure (total temperature) boundary condition is an extension of the fixed value boundary condition in OpenFoam, used to specify the static pressure at the boundary based on the user defined total or stagnation pressure and total temperature as well. It is useful for modelling plenum conditions. Variants for compressible and incompressible flows exist in OpenFoam. For the cases involving supersonic flow, for instance, it evaluates the static conditions by using the isentropic relations. Thus, $T_p = \frac{T_o}{1 + \frac{\gamma-1}{2\gamma}|u|}$, Where T_p is defined as the temperature of the patch (static), and γ is the adiabatic index. Figures 62 show the results obtained for this case.

RESULTS

The total pressure condition performed well for our tests. A back pressure value was specified for the domain based on isentropic flow calculations. Also considered was a case with zero gradient boundary condition at the outlet. In all cases, we see that we do not obtain the unphysical solution as before, in the inlet boundary, and depending on the back pressure

value, we are able to obtain a steady solution. Figure 62 presents plots for the pressure, temperature and velocity.

4.2.4 PLENUM PRESSURE BOUNDARY CONDITION

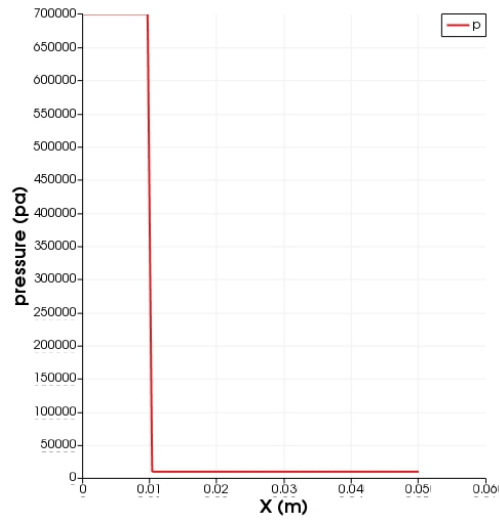
OpenFOAM uses this boundary conditions to set a plenum inlet condition. It models a zero dimensional volume of gas enclosed ahead of the inlet patch and the static pressure exerted on the inlet depends on the thermodynamic state within the enclosed volume. This model neglects momentum and advances the density and temperature of the plenum with run-time. The smaller the volume of the plenum, the more responsive it is to changes in mass flow rate and the further the model approximates a fixed mass flow system. If we significantly increase the volume, we obtain a fixed mass flow rate situation.

RESULTS

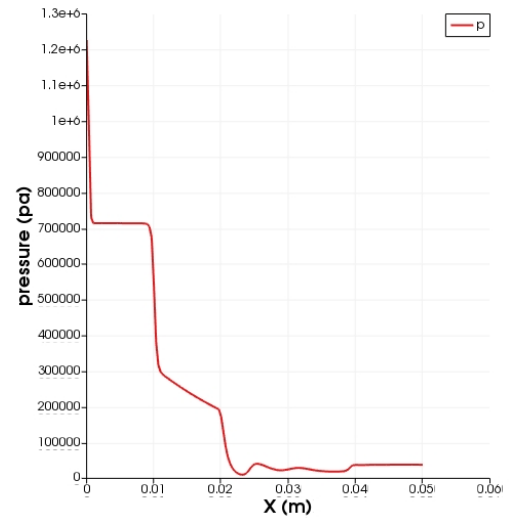
The plenum boundary condition also eliminates the jump in pressure observed at the inlet. As implied from our definition earlier, when it is assumed that a large enough volume is involved, this condition approximates to a fixed total pressure condition. Since a plenum that significantly maintains the stagnation conditions is desired, and in order not to introduce extra dynamics into our system, the total pressure and temperature boundary conditions was utilized for our simulations. Figure 63 represents field parameters obtained using the plenum condition.

4.3 CHOICE OF FLOW CONDITIONS

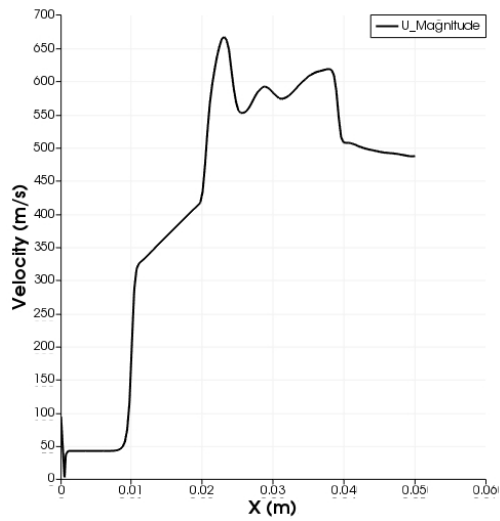
Using the total pressure and temperature boundary conditions, four different cases were set up, reflecting four possible solutions for an isentropic converging-diverging nozzle problem, by varying the back pressure in each case to determine the flow field. Using the stagnation conditions, the limiting back pressure values were determined for the subsonic (619 kPa) and supersonic (124 kPa) flow solutions with shock at the exit. Two other cases (100 kPa and 650 kPa) were chosen to enable us to see the results for other scenarios. The resulting flow fields are discussed in more detail.



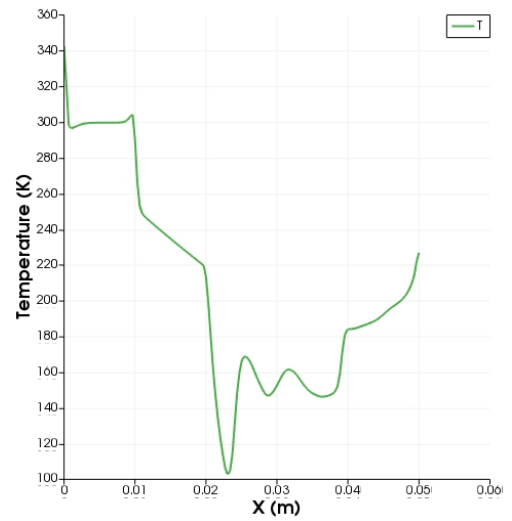
(a) Initial pressure state



(b) pressure plotted against distance, x.

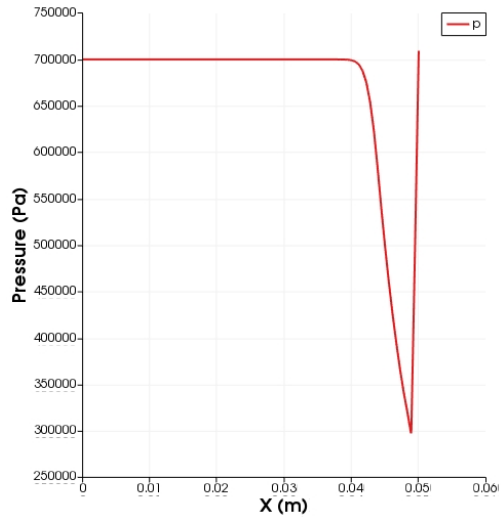


(c) Velocity using fixed boundary condition

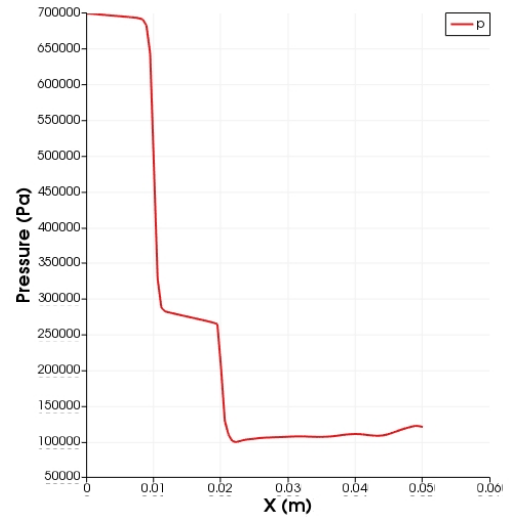


(d) Temperature plotted against distance, x

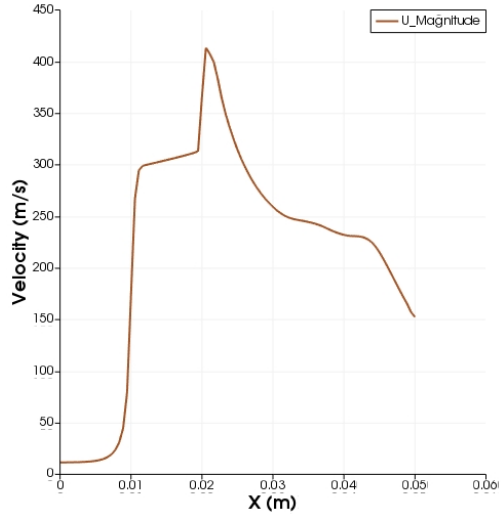
Figure 61: Plots showing fixed boundary condition with field variables indicating unphysical behaviour in the boundary



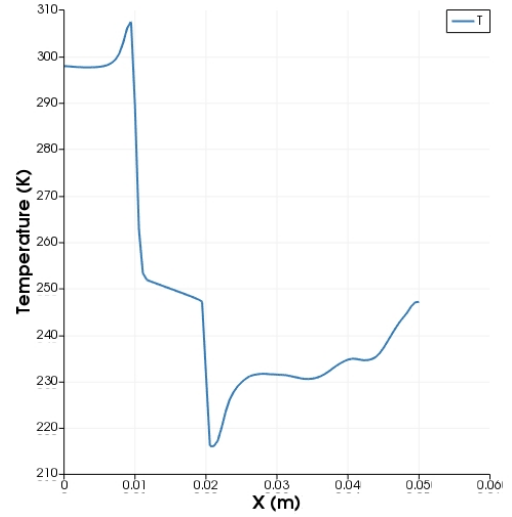
(a) Pressure plot showing shock and expansion wave.



(b) Pressure after 25 time steps

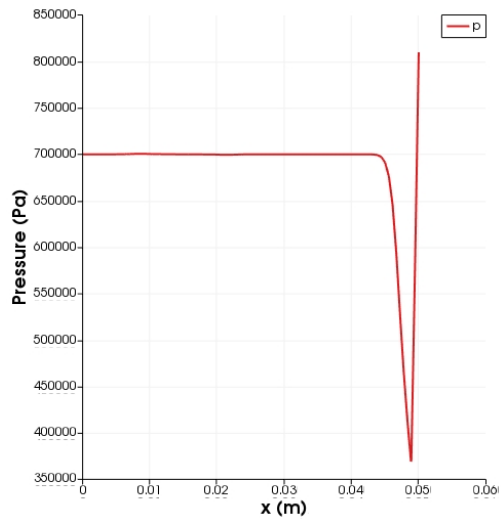


(c) Velocity after 25 time steps

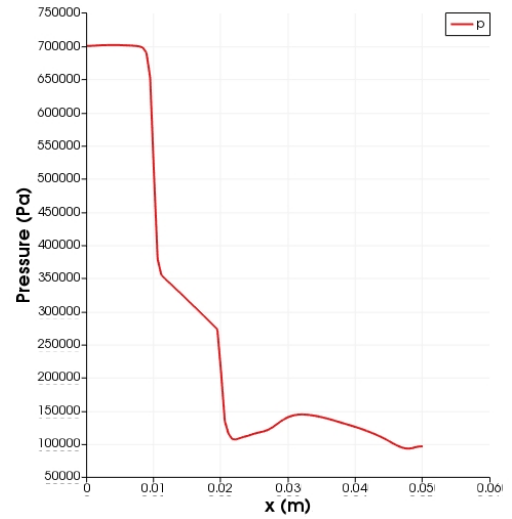


(d) Temperature

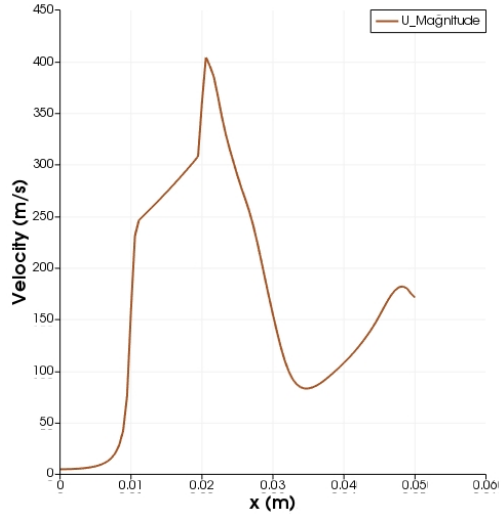
Figure 62: Plots obtained using total pressure and temperature boundary condition



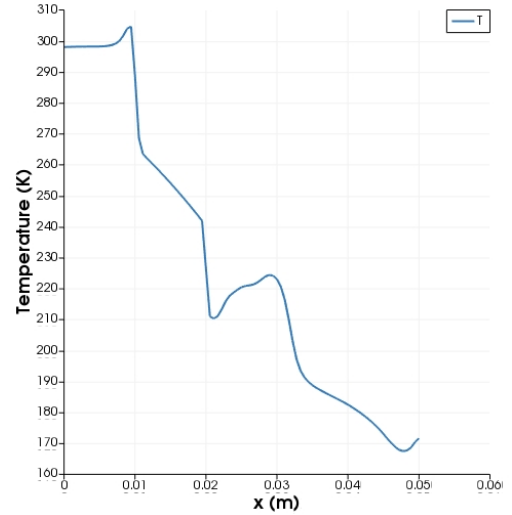
(a) Initial pressure state



(b) Pressure profile using plenum boundary condition



(c) Velocity after 25 time outputs



(d) Temperature after 25 time outputs

Figure 63: Plots showing plenum boundary condition with field variables

4.3.1 SUPERSONIC FLOW FIELDS: BACK PRESSURE OF 124kPa and 100kPa

These back pressures result in a supersonic flow at the nozzle exit, as shown in figure 62. However we do not get to a stable solution partly because the solver cannot handle the vortices that develop as the simulation is advanced in time. The case with back pressure of 124 kPa is the boundary case for a shock at the exit of the diverging end, while the case with a back pressure of 100 kPa should in reality result in oblique shocks at the exit. We encounter unphysical inflows into the domain from the outlet after the simulation runs for a long time, which eventually makes it fail. Hence we were not able to achieve a steady solution of the supersonic cases. We highlight that in view of these, we set up another case with an expected supersonic solution having a back pressure of 250 kPa. This was able to run for an extended period of time, without failing, until we ended the simulation. Despite reaching a metastable condition, disturbances eventually grew from the back boundary and polluted the solution.

4.3.2 SUBSONIC FLOW FIELDS: CASES WITH BACK PRESSURE OF 619 kPa and 650 kPa

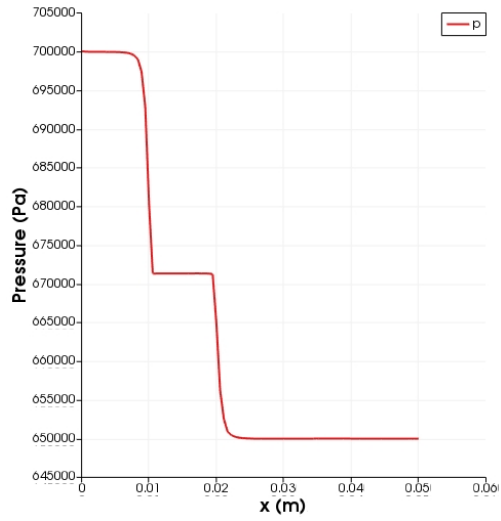
These result in subsonic flow fields, with 619 kPa being the limit based on 1-D isentropic flow calculations. Steady solutions of these cases were obtained by allowing the simulations to run for a long time . The corresponding pressure and velocity profiles for the case with back pressure of 650 kPa is presented in figure 64.

4.3.3 MACH NUMBER

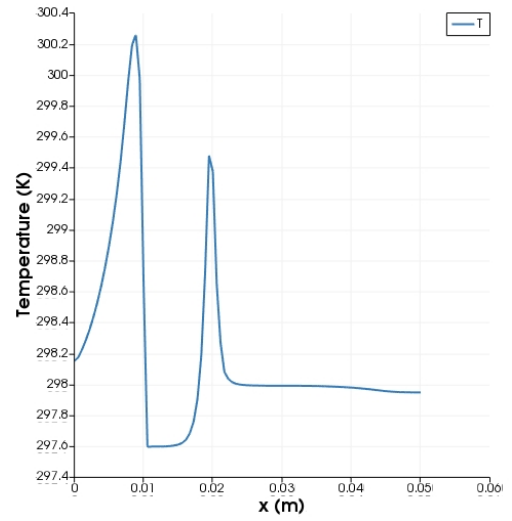
Here the Mach number for the selected flow fields for both the subsonic and supersonic cases are presented as shown in figures 65a and 65b.

4.4 INVESTIGATION OF MASS FLOW RATE

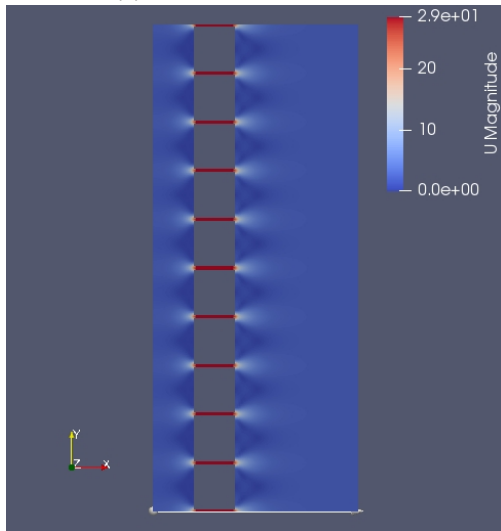
From continuity, the mass flow rate at the outlet should equal that at the inlet, for a steady flow. Some anomalies were noticed when using the post processing utility of determining mass flow rate inbuilt in OpenFOAM, at the preliminary stage of simulation, where different values for the inlet and outlet patches were obtained. We want to be able to examine the impact variations in supply mass flow rate could have on the operations of an RDE, hence the reason this is of importance. This lead us to investigate the accuracy of the method of



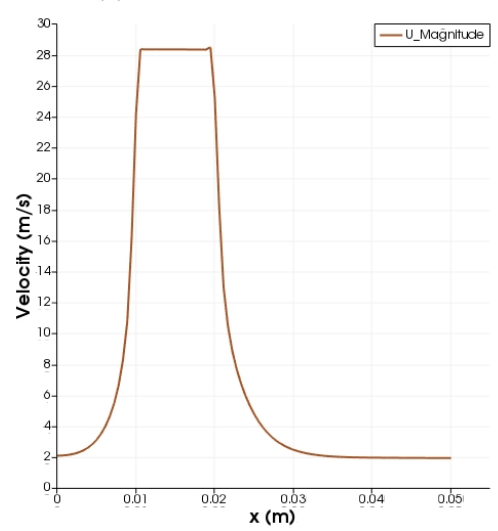
(a) Pressure profile vs x axis



(b) Temperature profile vs x axis

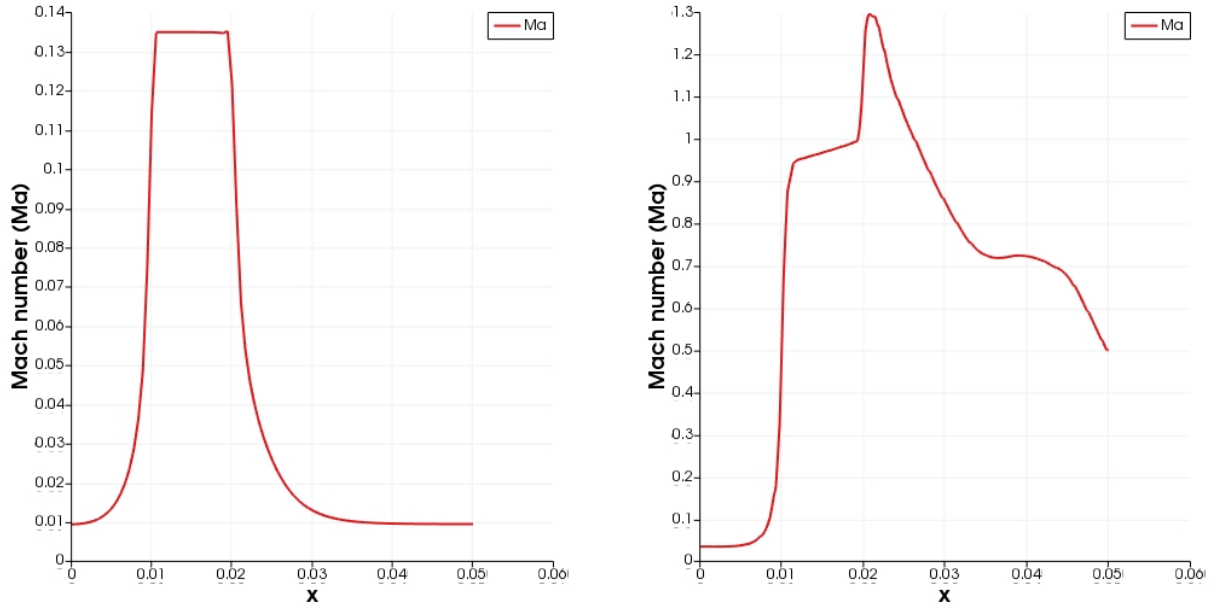


(c) Velocity field



(d) Velocity plot vs x

Figure 64: Steady state subsonic flow field using a back pressure of 650 kPa



(a) Mach number for subsonic initiation flow field

(b) Mach number for supersonic initiation flow field

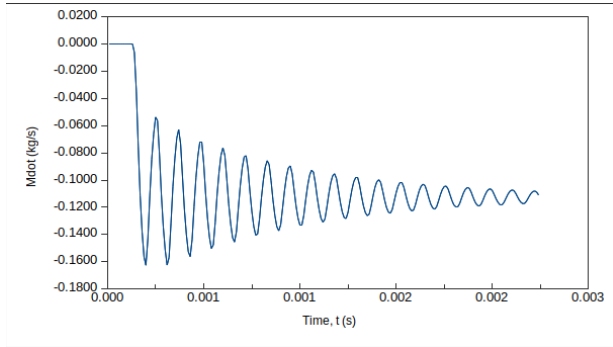
Figure 65: Mach number for subsonic and supersonic initiation flow fields

calculation implemented in OpenFoam. To do this, the value at the inlet and outlet of our domain were evaluated for 6 back pressure test cases: 100 kPa, 124 kPa, 250 kPa, 650 kPa, 500 kPa and 619 kPa. Plots of plots mass flow rate (\dot{m}) against time for the 3 cases: 250 kPa, 650 kPa and 500 kPa are presented. The "integrate patch" utility was utilized to determine the surface integral over the patch faces in question (inlet and outlet) as a function of time. A comparison of these values with that obtained from 1-D isentropic flow analysis. The mass flow rate into a control volume is given by the product of the density, velocity and the surface normal to it. In OpenFoam, a negative surface normal indicates an inlet area.

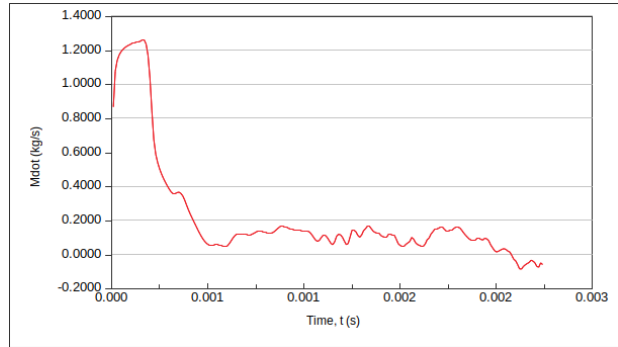
4.4.1 RESULTS AND ANALYSIS

The mass flow rate results obtained for the 4 cases in figures 66 to 69 are shown. The plots in blue indicate the inlet mass flow rate, while that for the outlet boundary are in red.

The supersonic cases do not achieve a steady solution and does not result in a convergence of mass flow rate values at both inlet and outlet patches even after a long time. They start off with a large amplitude of oscillation which decreases with time, until the simulation eventually fails, due to the development of intense vortices which cannot be handled by the outlet boundary condition, as shown in figures 66a. However it can be seen that at some time steps before the solution goes unstable, the same values are obtained approximately, implying a reasonable degree of convergence at that point. From the outlet flow profile, it

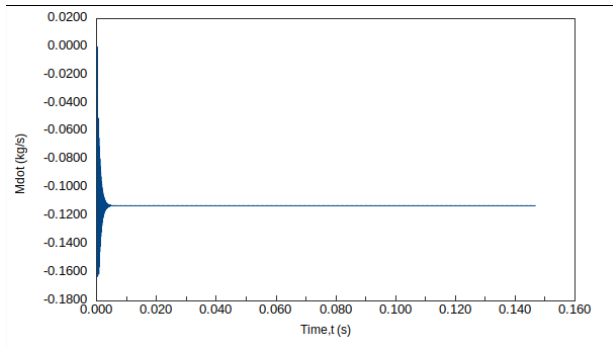


(a) Mass flow rate against time at inlet boundary for $P_b = 100$ kPa.

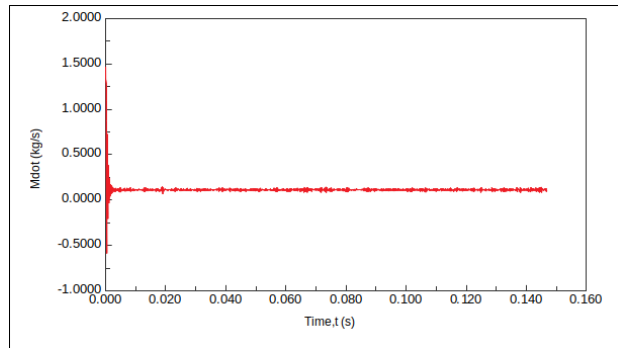


(b) Mass flow rate against time at outlet boundary for $P_b = 100$ kPa.

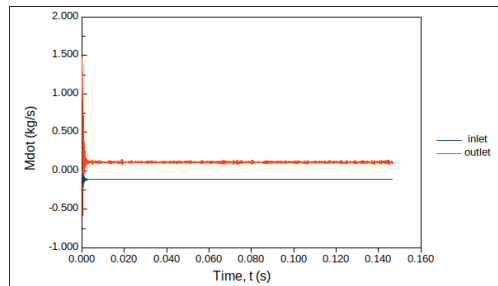
Figure 66: Mass flow rate across boundaries for $P_b = 100$ kPa.



(a) Mass flow rate against time at inlet boundary for $P_b = 250$ kPa.

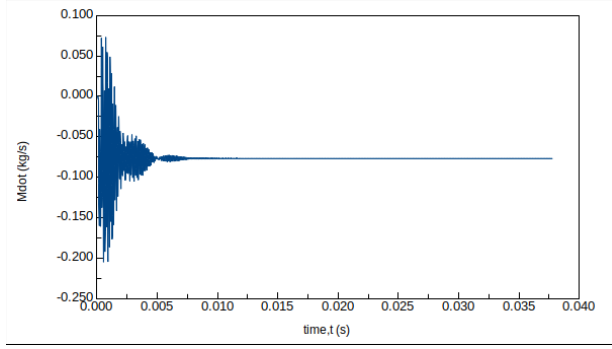


(b) Mass flow rate against time at outlet boundary for $P_b = 250$ kPa.

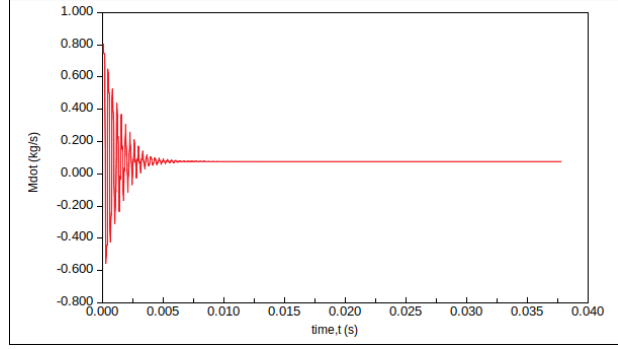


(c) Inlet and outlet flow fields plotted together showing some oscillations

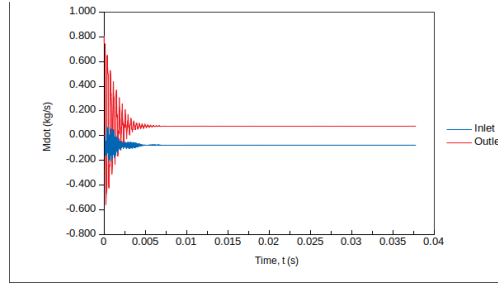
Figure 67: Mass flow rates across boundaries for $P_b = 250$ kPa.



(a) Mass flow rate against time at inlet boundary for $P_b = 500$ kPa.



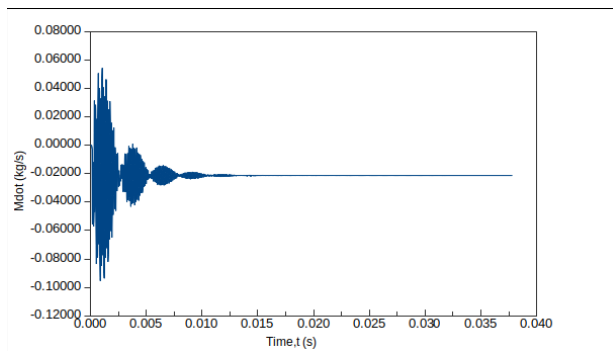
(b) Mass flow rate against time at outlet boundary for $P_b = 500$ kPa



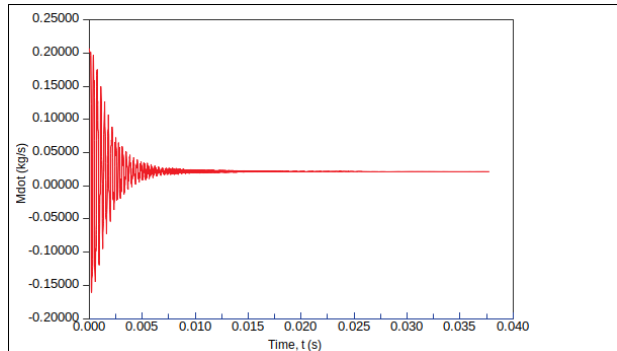
(c) Inlet and outlet flow fields plotted together showing flow is steady

Figure 68: Mass flow rate across at boundaries for $P_b = 500$ kPa

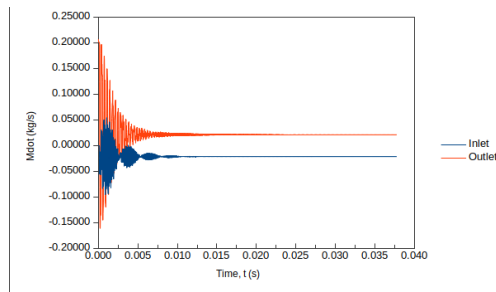
can be seen that the mass flow rate dips to a negative value which suggests an inflow into the boundary at some point, after which the simulation fails. The case with a 250 kPa back pressure as shown in figure 67 gives a more stable result. The velocity goes supersonic at the nozzle exit, but even then, the flow swirls after some time, and a final steady solution is not achieved. The calculated mass flow rates for the supersonic flow with a back pressure of 100 and 250 kPa is 1.225800×10^{-1} kg/s and 1.124639×10^{-1} kg/s respectively, while for the subsonic cases with a back pressure of 619 and 650 kPa, the resulting mass flow rate is 3.397538×10^{-2} kg/s and 2.128058×10^{-2} kg/s respectively. The cases with back pressures of 250 kPa, 500 kPa and 650 kPa are shown in figures 67, 68 and 69. The subsonic cases resulted in a convergence of inlet and outlet values to a constant value, after running for a long time (over 3700 times steps).



(a) Mass flow rate against time at inlet boundary for $P_b = 650$ kPa.



(b) Mass flow rate against time at outlet boundary for $P_b = 650$ kPa



(c) Inlet and outlet flow fields plotted together showing flow is steady

Figure 69: Mass flow rate against time across boundaries for $P_b = 650$ kPa

4.5 SELECTED FLOW FIELDS FOR INITIATION

4.5.1 FLOW FIELD FOR SUBSONIC INITIALIZATION

The case with the back pressure set at 619 kPa was selected, and its steady state solution using a coarse mesh obtained. The CFL of 0.1 was maintained, and the calculated mass flow rate outputted at every time step. This served as our stability and convergence criteria. The corresponding pressure profile is shown in figure 70a.

4.5.2 FLOW FIELD FOR SUPERSONIC INITIALIZATION

The selected flow field to be tested for supersonic initiation is that obtained using the case with a back pressure of 124 kPa. The same process as in the subsonic case above was repeated. The flow simulation fails eventually due to vortices and oscillations in the exit, and also because our boundary condition does not account for the solution of a Converging-Diverging nozzle with an expansion or oblique shock at the exit. The profile obtained is similar to the one shown in figure 62. To conclude this section, at this point two flow fields, one subsonic and one supersonic have been chosen. These become the initial condition for the initiation of a rotating detonation wave, discussed in the proceeding chapter, where a description of some of the various initiation techniques attempted are presented.

Chapter 5

SIMULATION OF ROTATING DETONATION ENGINE INITIATION

5.1 PROBLEM DESCRIPTION/CASE SET UP

It has been established that the flow field for the Wolanski engine, with a certain number of injectors, can be modelled as a converging-diverging nozzle problem from gas dynamics, using the equations of 1-D isentropic nozzle. Solutions for subsonic and supersonic cold start conditions for this engine depending on the choice of back pressure have been obtained. In this chapter, it is desired to initiate the RDE, and evaluate different initiation patterns, considering a case with high temperature, another with high pressure and temperature, and one with burnt mixture within the domain.

5.1.1 MESH DESCRIPTION

The same domain as with the cold start has been retained; i.e., the unwrapped 2-D domain. However the mesh was refined to have 0.7 points per $L_{1/2}$, resulting in 86850 cells, and the case with 10 points per half reaction zone length, resulted in 17 million cells (17042520). This was run on Compute Canada high performance clusters Graham and Cedar. OpenFOAM allows for parallel decomposition of mesh for computing, hence the domain was decomposed into 32 units for the same number of processors, each having over 500,000 computational cells. It was split along the x-axis.

5.1.2 BOUNDARY CONDITION

The same boundary conditions as with the cold start case was used to generate the mesh, i.e., a total pressure and temperature condition with a back pressure value depending on if the initiating flow is to be subsonic or supersonic. First the mesh was generated with these and for the actual initiation flow simulation, after which the outlet boundary condition was changed from the back pressure condition (fixed value pressure) to a boundary extrapolated flow condition (zero gradient). This was done to prevent expansion waves at the divergent section, which degrade the strength of the initiation zone, thereby preventing a transition to a detonation wave, particularly for the supersonic case.

5.1.3 INITIAL CONDITIONS

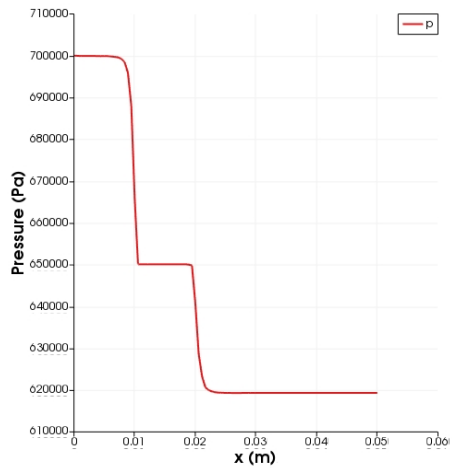
For the purpose of simulating the RDE, two initiation flow fields were considered; subsonic and supersonic cases. The selected flow field results in chapter 4 were utilized as our initial flow fields. For both cases, the results were mapped from a steady time step obtained from the coarse mesh cold flow simulations, using the "mapFields" utility, which had 16200 cells, corresponding to 0.2 points per half reaction zone length, to the finer meshes considered. It should be noted that both meshes are "consistent", i.e., of similar geometry and boundaries, hence there was no need to create a "mapFieldsDict" in the system folder for these cases.

5.1.4 DIFFERENT TYPES OF INITIATION

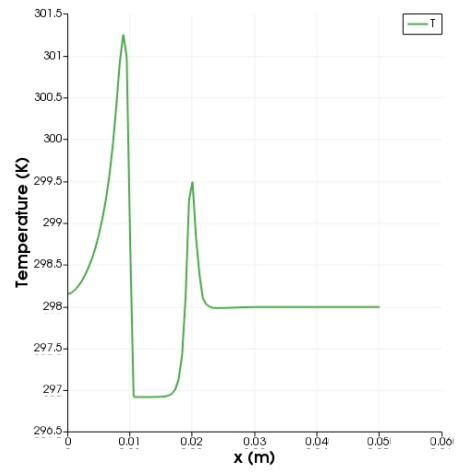
Four different initiation techniques were considered as shown in fig 71; a purely temperature based initiation in which case the temperature within a small box or zone is set to an elevated value as in fig 71a, and an initiation pattern with both elevated temperature and pressure values as in fig 71b, another with burnt gases in the combustion chamber as in fig 71d, and finally a case where the initiation was done further downstream of the domain as in fig 71c. These were tried for both the supersonic and subsonic flow conditions, for both the high and low resolution cases that were simulated.

5.2 RESULTS WITH LOW RESOLUTION MESH

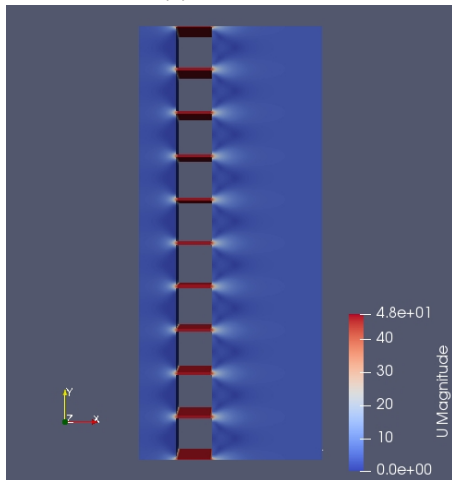
Results for the low resolution mesh with 0.7 points per $L_{1/2}$ simulated on a local device are presented and discussed below.



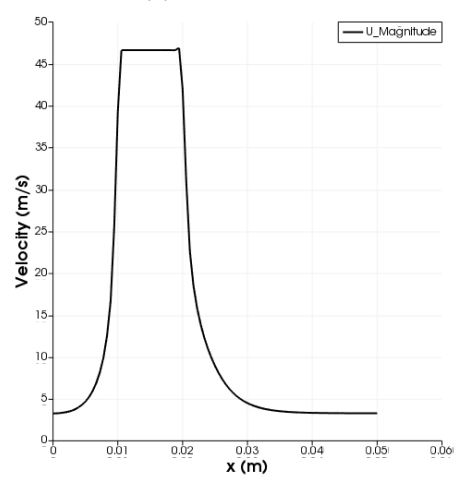
(a) Pressure profile



(b) Temperature profile



(c) Velocity flow field



(d) velocity profile

Figure 70: Subsonic flow field using a back pressure of 619 kPa

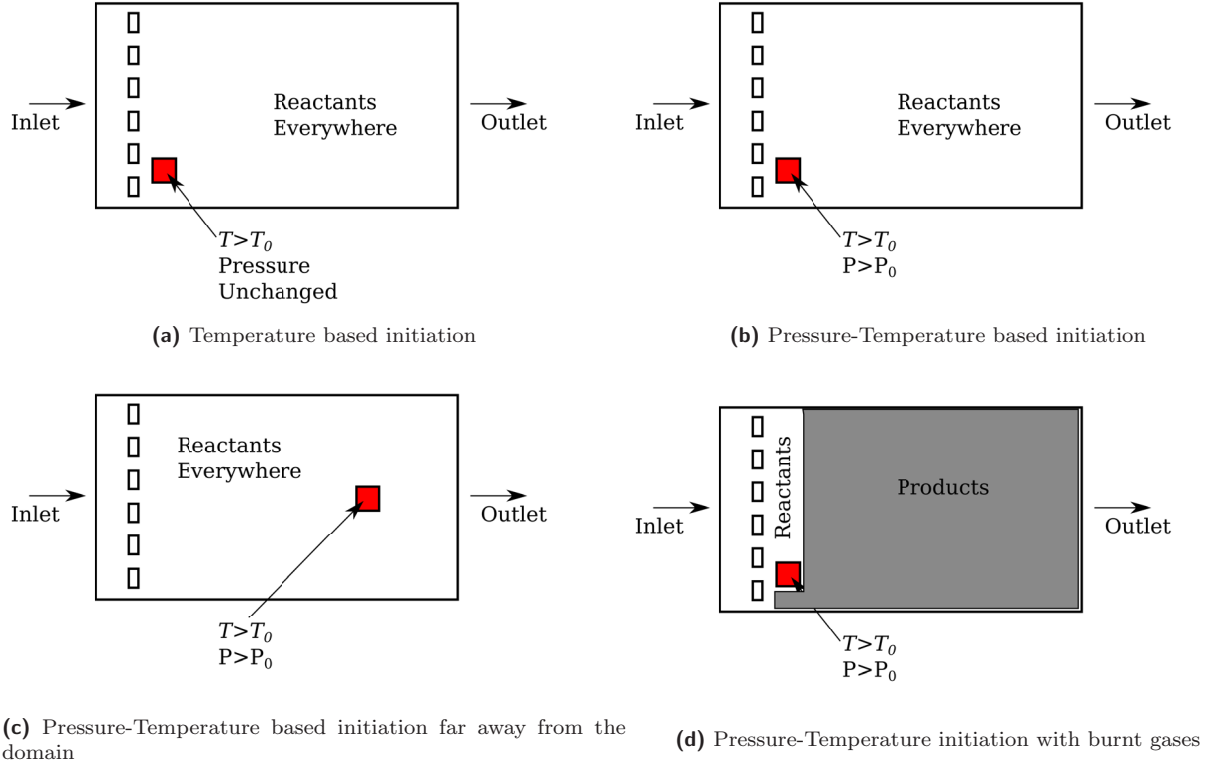


Figure 71: Various initiation techniques considered

5.2.1 SUBSONIC FLOW FIELD INITIATION

The "setFields" utility was used to set a high temperature region of 2000 K within a few cells contained in a small box, by using the "boxToCell" utility, after which the simulation was advanced in time. The hot spot (small box) was set to be 0.005 m away from the injectors and was of same magnitude in length. It makes sense to do this before decomposing the domain using the "decomposePar" command, otherwise one will have to map the cells from the individual processors to those contained in the finer mesh. Detonation waves that propagated in both the axial and circumferential directions in the combustion chamber were initiated. These waves are symmetric. The von Neumann spike pressure, as outputted from the simulation after 9 time outputs is 2.69168×10^7 kPa, while the pressure behind the detonation wave is about 8×10^6 kPa, roughly 10 times the initial value, which correlates with what is obtainable in literature. The shock pressure is 2.77533×10^7 kPa just before the simulation ends. We note that the simulation fails as there exists a detonation wave travelling back axially to the inlet boundary, through the injector holes, at a very high pressure. The pressure, temperature and velocity plots are shown in figure 72. Results for pressure-temperature based initiations are presented in figure 73 for a hot spot introduced

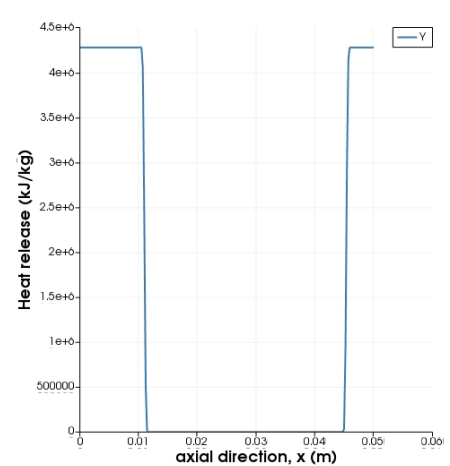
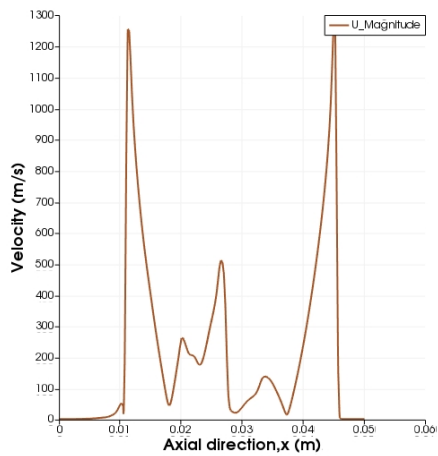
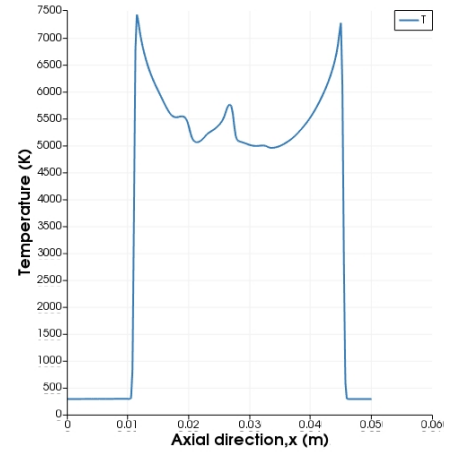
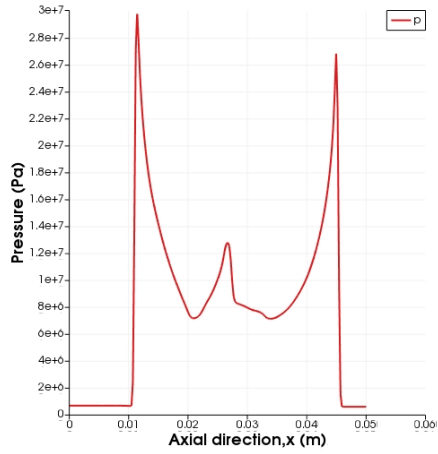
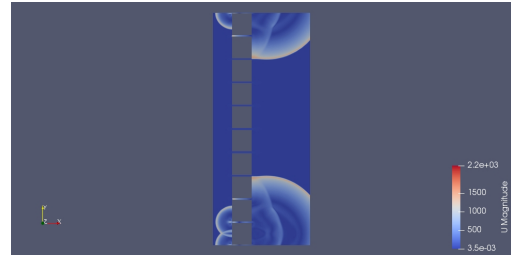
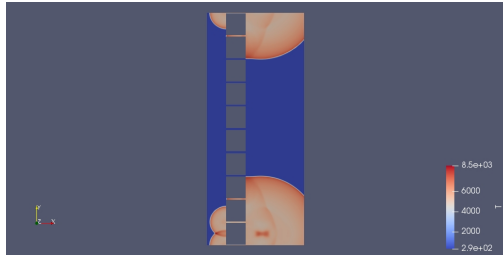
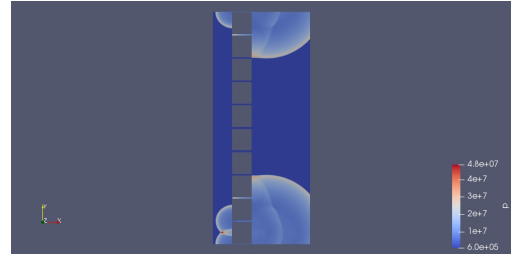
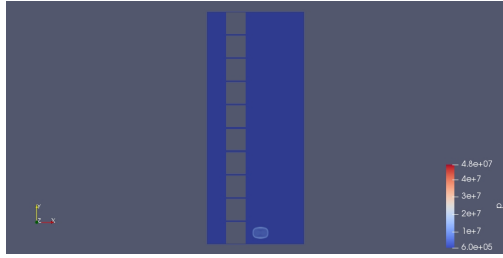


Figure 72: Detonation initiated with a high temperature using a subsonic flow field

in a region close to the nozzle outlet. The same temperature was maintained as with the temperature based initiation, of 2000 K. The pressure in the region was however set to 900 kPa, about 1.3 (1.2857) times higher than the stagnation pressure. It can be seen that the initiation of a rotating detonation wave was achieved, which had the same von Neumann peak pressure (2.69168×10^7 kPa) value after nine (9) time outputs, as with the temperature based initiation, with the pressure behind the detonation also remaining fairly the same value. This suggests that the temperature seems to be the main driver of the shock strength for this choice of pressure ratio.

A temperature based initiation with the hot spot placed further away from the domain was performed. Plots in the y direction showing two detonation waves travelling with the same peak pressures are presented. Figure 74 presents results for the field variables as the detonations travels in space. Of significance is the observed shock-reaction coupling effect which is the hallmark of a detonation. Initiation cases in which the reactive mixture from certain regions of the domain are cut off were considered to see if that results to obtain one initial detonation wave travelling in the y-direction. In practice, this can be viewed as igniting a detonation chamber filled with a mixture of burnt and un-burnt gases. Figure 75 presents results obtained, and we see from the image for the pressure shown in figure 75b, that this approach results in one detonation wave travelling faster than the other.

5.2.2 INITIATION WITH SUPERSONIC FLOW FIELD

Using the case with a back pressure of 124 kPa, we initialized the supersonic cases to run with a CFL of 0.1, and with Δt set as 1×10^6 s. The supersonic cases never achieved a steady flow solution as stated earlier, however, by making a comparison between the inlet and outlet mass flow rates, and also visualizing the flow field using Paraview, it was possible to pick out a time step that appeared stable to some extent. Hence the simulation results were mapped from that time step into our finer mesh. The process of initiation using the temperature, and pressure-temperature based approaches were considered. The setFields utility, was used to set up hot spots as described earlier. The same temperature of 2000 K was maintained for this case. Figure 76 presents results obtained for a temperature based initiation for the supersonic flow field. We see that a symmetric detonation wave propagating in the x and y directions is achieved, and one also recovers the oscillating behaviour of a detonation wave. It can be seen also that the shock pressure for this case was of the same order of magnitude with the activation energy, and the propagating detonation is oscillatory in nature. The peak shock pressure after 10 seconds is 5.7013×10^6 kPa, with the highest particle velocity being 1305.01 m/s Results for pressure-temperature based initiations for this flow field are

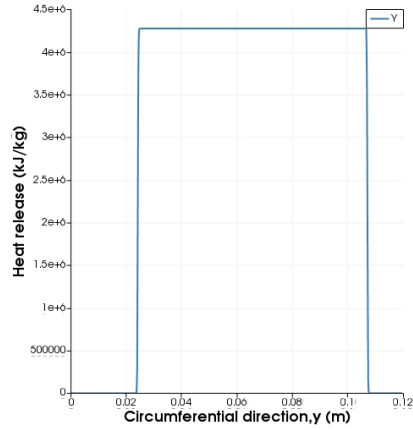
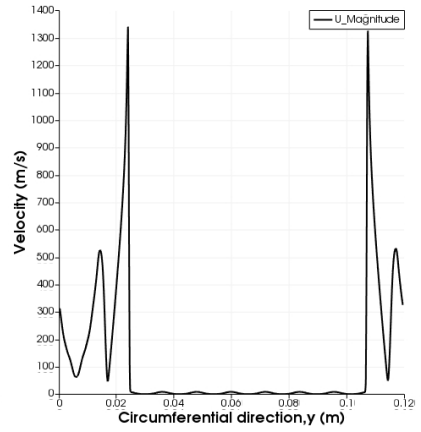
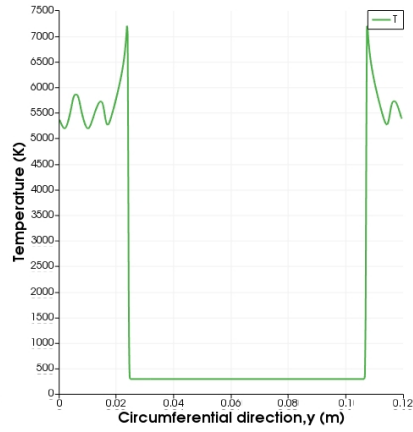
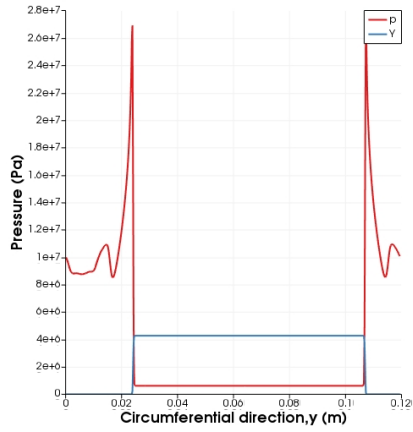
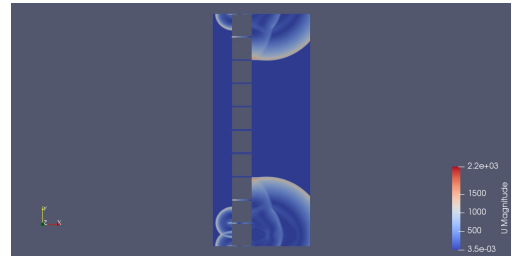
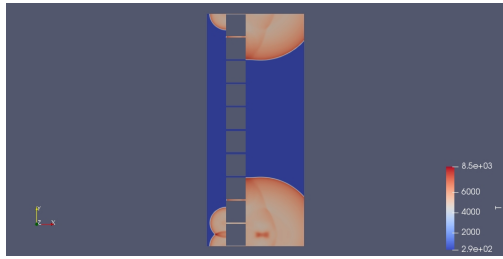
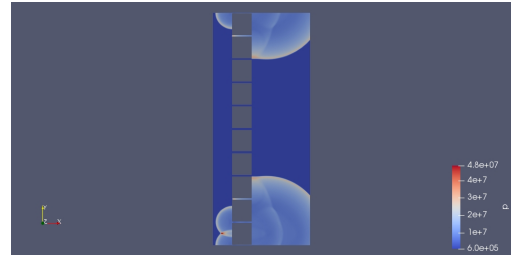
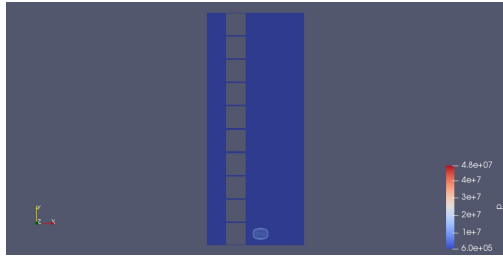
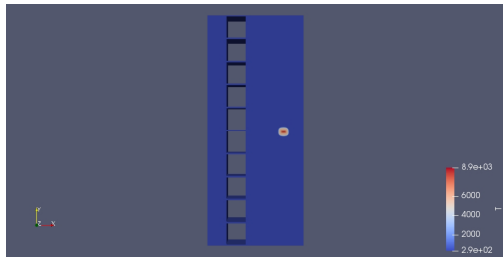
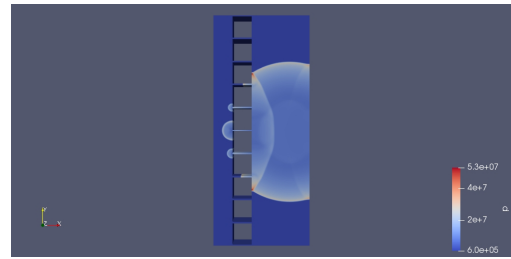


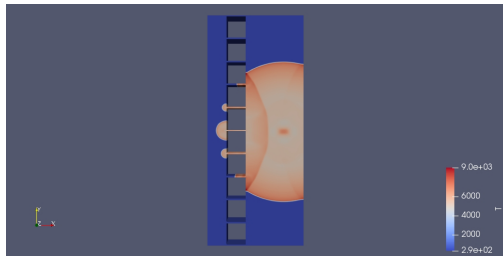
Figure 73: Detonation initiated with a high pressure-temperature zone using a subsonic flow field



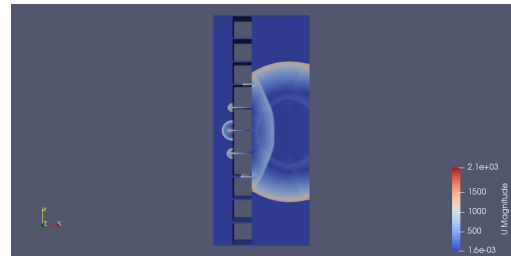
(a) Initiation spot (T)



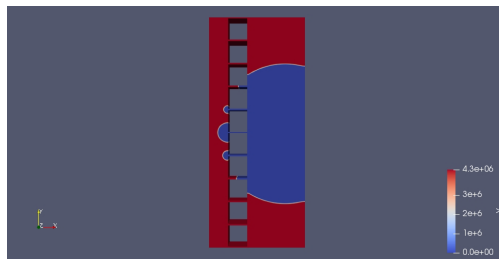
(b) Two travelling detonation waves



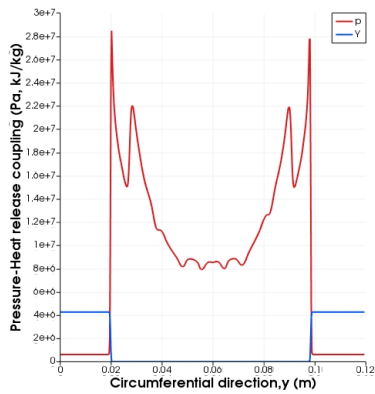
(c) Temperature



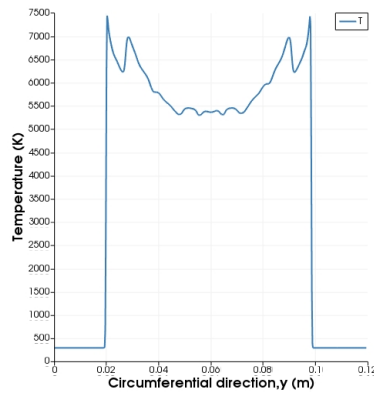
(d) Velocity



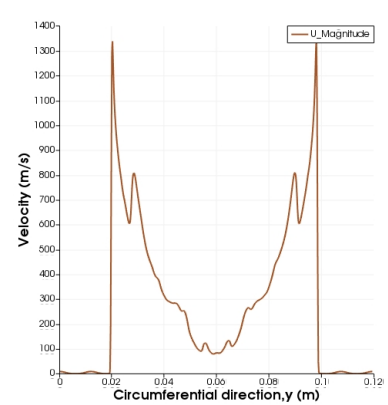
(e) Heat release



(f) Shock pressure- reaction coupling in y direction

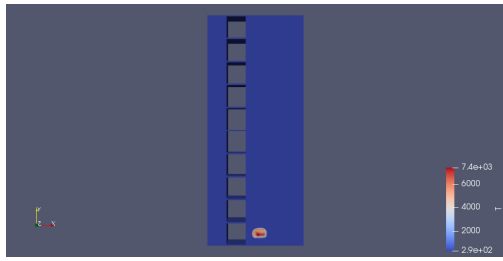


(g) Detonation Temperature in y direction

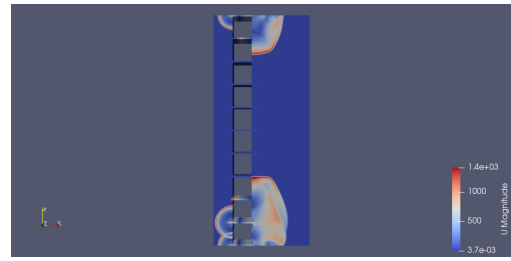


(h) Velocity in y direction

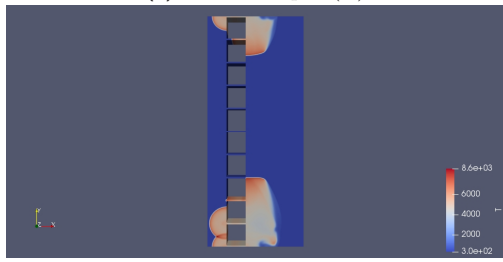
Figure 74: Detonation initiated further downstream with a high temperature zone using a subsonic flow field



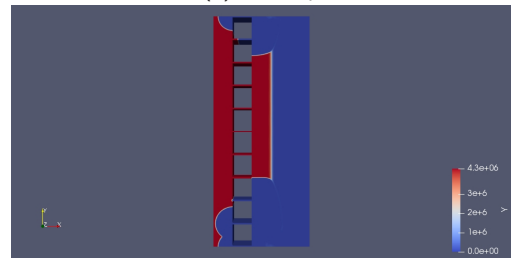
(a) Initiation spot (T)



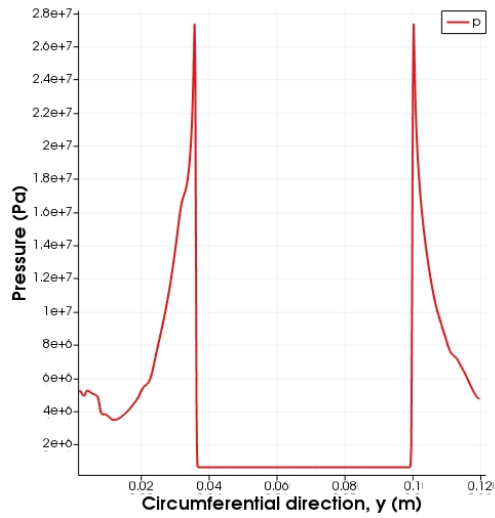
(b) Velocity



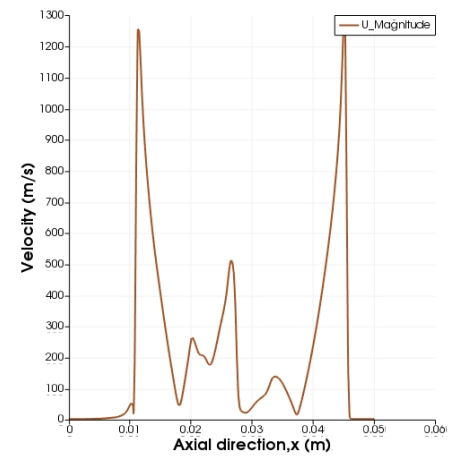
(c) Temperature



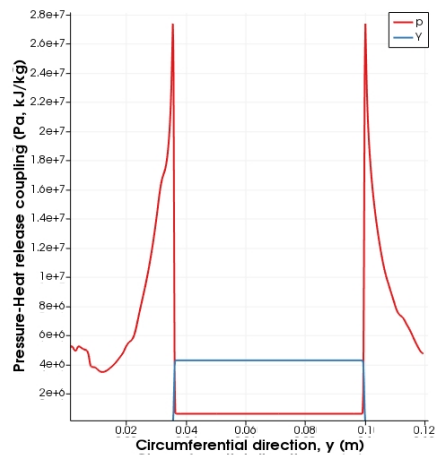
(d) Heat release



(e) Pressure in x direction

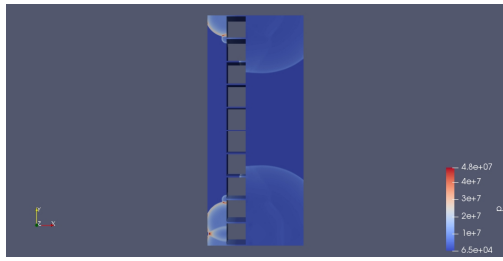


(f) Velocity in x direction

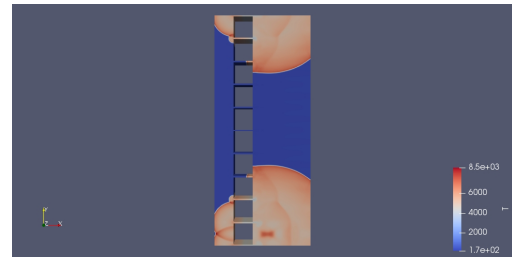


(g) Shock-reaction coupling in x direction

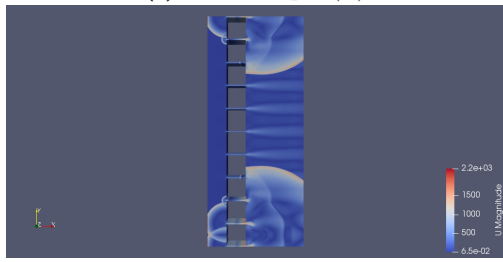
Figure 75: Detonation initiated with a high pressure and temperature zone using a subsonic Flow field



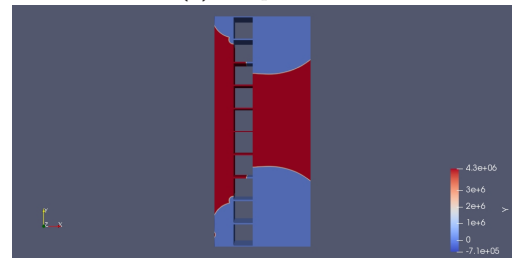
(a) Initiation spot (T)



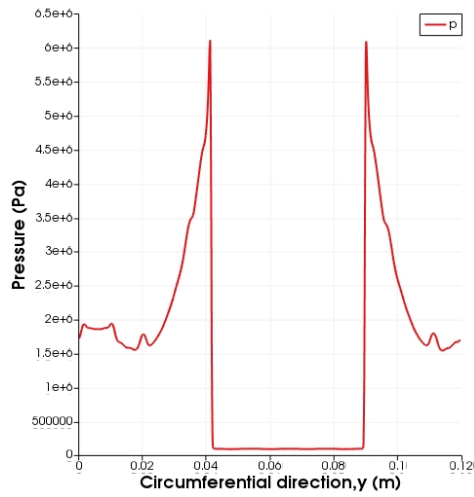
(b) Temperature



(c) Velocity



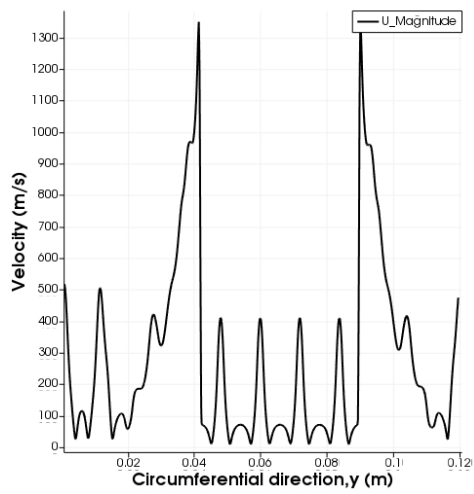
(d) Reaction



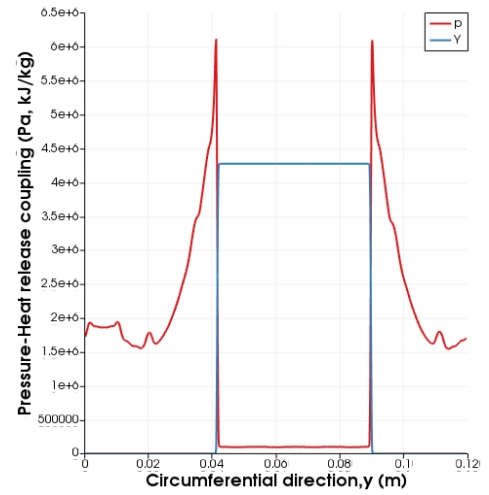
(e) Pressure profile in y direction flow field



(f) Temperature in y direction



(g) Velocity profile in Y direction



(h) Shock pressure-reaction coupling in y direction

Figure 76: Detonation initiated with a high temperature zone using a supersonic flow field

shown in Figure 77. The same ratio as with the subsonic cases was utilized. The peak pressure value after 10 time outputs was 5.7013×10^6 kPa, slightly lower than that for the temperature based initiation. We see that two symmetric detonation waves propagating in the y direction is the resulting solution. Another mode of initiation for the supersonic flow mixture was considered, in this case, a certain section of the domain is filled with burnt gases. This was done by setting a zero value to the activation energy at the exit end of the domain. The pressure and temperature values within the hot spot was set to the same value as with those for the pressure-temperature based initiation, by using the setFields utility. Results for this initiation procedure are presented in figure 78

5.3 EFFECT OF HIGHER RESOLUTION ON DETONATION INITIATION

Using both flow fields (subsonic and supersonic), simulations were performed on a high resolution mesh, with 10 points per $L_{1/2}$, in Compute Canada, resulting in over 17 million cells. The domain was decomposed into 32 strips, using the "decomposePar" utility in OpenFoam, with the mesh split in x-direction, each having over 500,000 cells. Again attempt was made at initiating a detonation wave using temperature and pressure-temperature based initiation techniques as with the low resolution cases. The results obtained for each mode of initiation for both the subsonic and supersonic flow fields considered are discussed below.

5.3.1 SUBSONIC CASE

First the subsonic high resolution mesh was initiated with a hot spot of temperature of 2000 K. However, a detonation wave was not achieved. It is important to restate here that the hallmark of a detonation wave is a coupling between the shock and reaction, thus one should expect to see this coupling when a visualization of simulation results is performed in the post processing software used for this work, Paraview. It is worthy of mention that even though one gets an initial jump to a high pressure value, and the reactants are burnt out after ignition, the flow does not transition into a detonation wave. Another initiation approach for the RDE, done by setting a high pressure-temperature region within the combustion chamber just after the nozzle as in figure 71b, using the setFields utility as above. Four (4) pressure ratios were considered: 1.3 (as above), 1.5, 2 and 5, each case maintaining the same temperature. A sustained detonation wave for the subsonic flow field for all the ratios considered was not achieved. There temporarily existed a detonation wave propagating for a few time steps, however this quickly decouples. It is important to highlight that

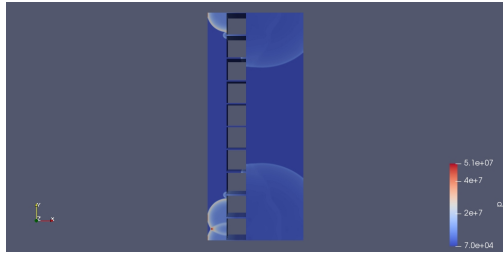
even though combustion continues, the engine doesn't function as one that is detonation based. Eventually, the simulation fails after some time as there is an axially back-travelling detonation wave, with a shock pressure value higher than the stagnation pressure, and hence the inlet boundary values, which reaches the inlet, thereby preventing further injection. Results for the subsonic case with a pressure ratio of 5 are presented, with a look at figures 79a and 79c showing the decoupling between the shock and chemical reaction behind it, indicating that a sustained propagating detonation wave is not achieved. A presentation of 1-D slice images as shown in figures 81 and 82 highlight this further, where by a coupling of the pressure and reactive terms exist initially, decoupling after some time.

5.3.2 SUPERSONIC CASE

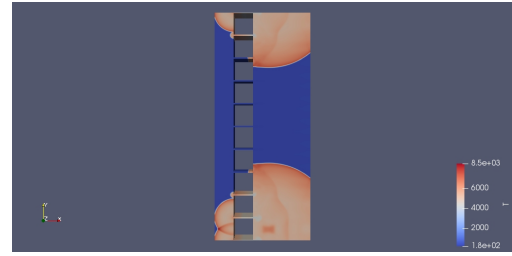
For this case, the high resolution mesh was initially initiated with a temperature of 2000 K. A detonation wave for the supersonic case with this technique was also not initiated. It is a struggle to see even a significant jump in pressure after ignition, as is usually the case when a detonation is formed, and there exists a coupling between the shock pressure and reaction. One would also expect to see this coupling when a visualization of the solution is done in Paraview. It is seen that even though this results to a jump in pressure value, and the reactants are burnt out after ignition, it does not transition into a detonation wave. The RDE was then initiated by setting a high pressure-temperature region within the combustion chamber using the setFields utility as above. Four (4) pressure ratios were also considered: 1.2857(as above), 1.5, 2 and 5, each time maintaining the same temperature of 2000 K. It was observed that this fails to achieve a detonation (or sustained) wave for both the subsonic and supersonic flow fields for all the ratios except for the supersonic case with a pressure ratio of 5. This case results in a sustained detonation wave circumferentially propagating in our domain. However the simulation fails as well after some time as we have an axial back-travelling detonation wave with a pressure higher than the inlet boundary pressure, which hits the inlet thereby preventing further injection. Results for the supersonic flow initiations using a pressure ratio 5 are presented in figures 83 and 84, where it can be seen that the propagating pressure wave remains coupled with the chemical reaction occurring behind it. A rotating detonation wave is achieved and sustained for the supersonic case using this pressure ratio, with a Von-Neumann peak pressure of 1.1×10^7 kPa after 14 time outputs.

5.3.3 CONCLUSION

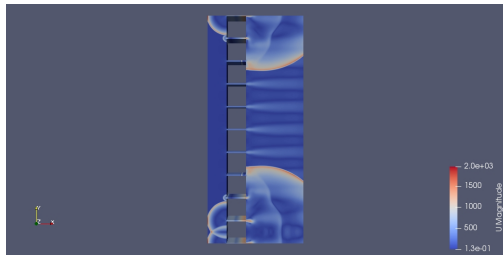
In conclusion, circumferentially propagating detonation waves have been initiated by using different initiation techniques; a high temperature spot of 2000 K, and a high pressure-temperature initiation zone as well. For the low resolution mesh, the effect of utilizing a high pressure-temperature initiation instead of only a high temperature zone didn't seem to be obvious, but became so when the high resolution (10 points per $L_{1/2}$) mesh was utilized, and for the same ignition pressure ratio (of 5) and temperature, a transition to detonation waves was achieved using a supersonic flow field as the initial flow field.



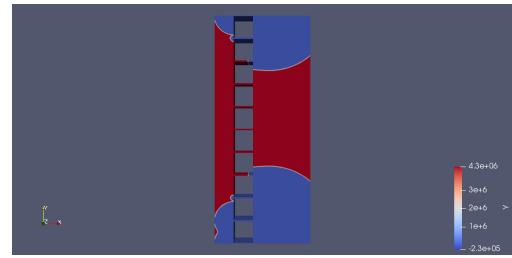
(a) Pressure



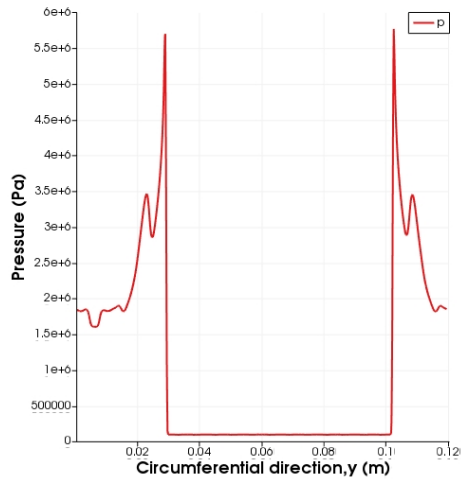
(b) Temperature



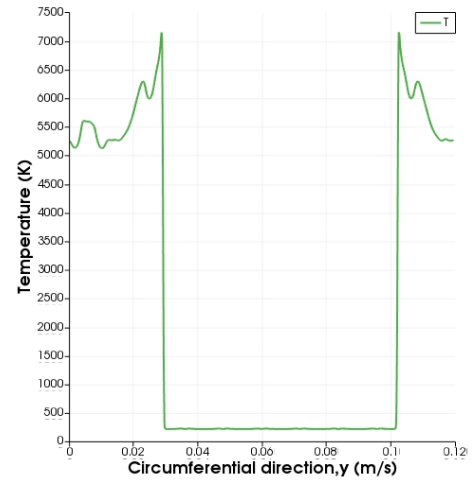
(c) Velocity



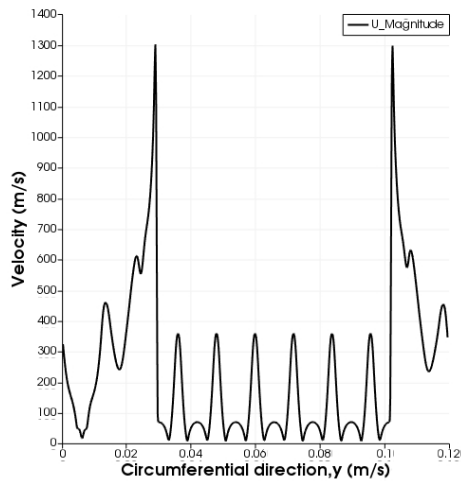
(d) Heat release



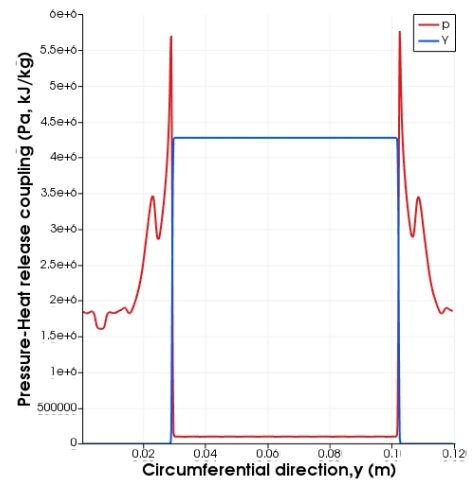
(e) Pressure profile in y direction



(f) Temperature profile in y direction



(g) Velocity profile in y direction



(h) Shock-Reaction coupling in y direction

Figure 77: Detonation initiated with a high temperature and pressure zone using a supersonic flow field

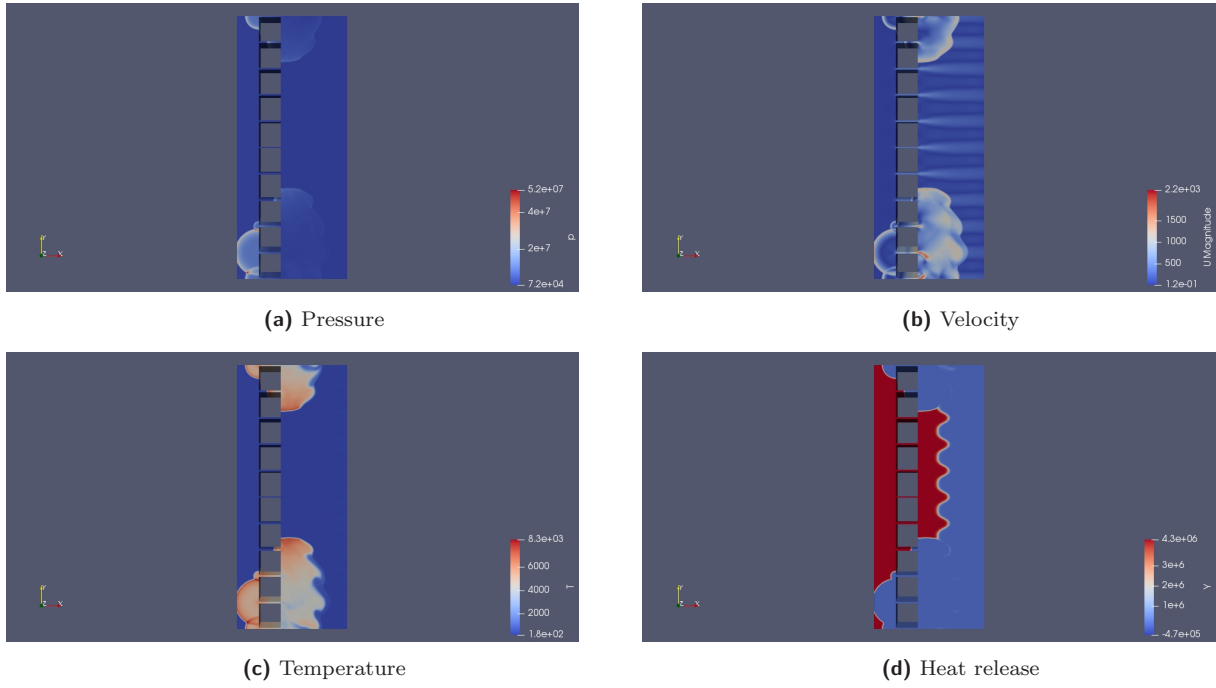


Figure 78: Temperature based initiation for supersonic flow with a pressure ratio of 1.2857

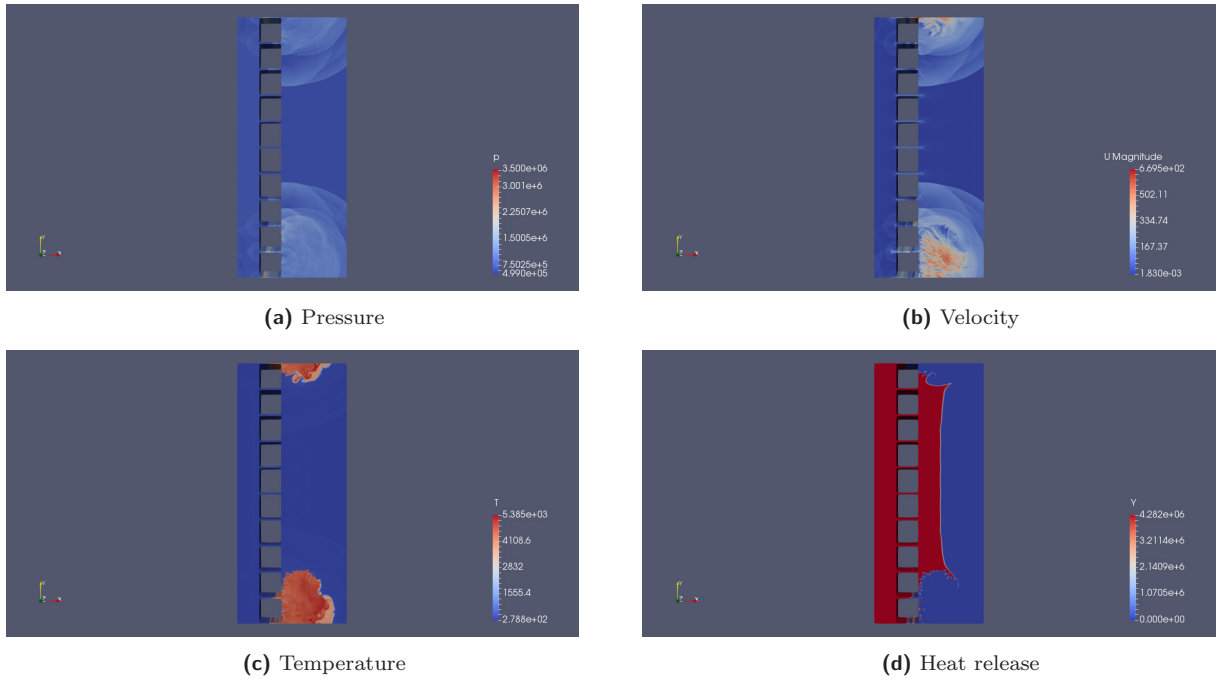


Figure 79: Pressure-Temperature based initiation for subsonic flow with a pressure ratio of 5

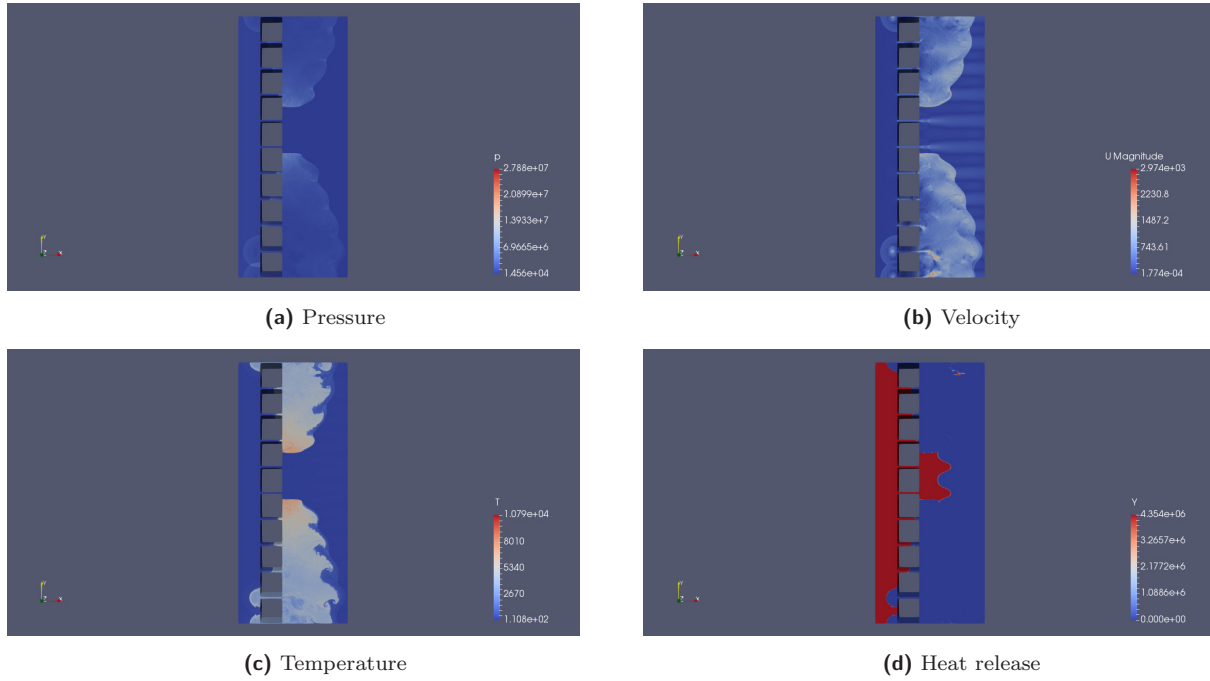


Figure 80: Pressure-Temperature based initiation for supersonic flow with a pressure ratio of 5

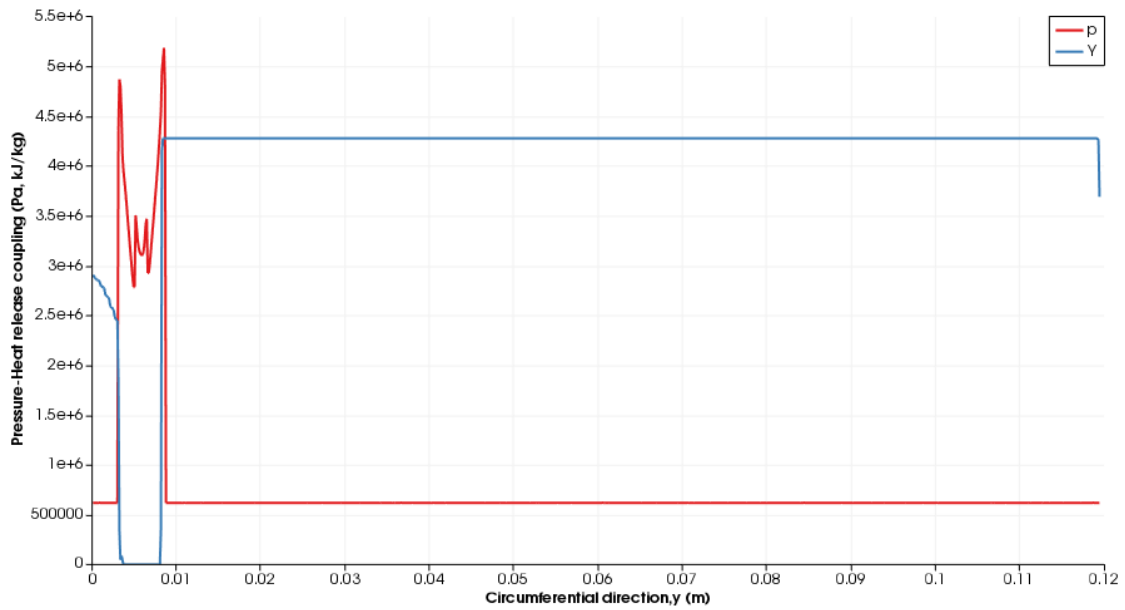


Figure 81: Initial unstable detonation wave (Subsonic case)

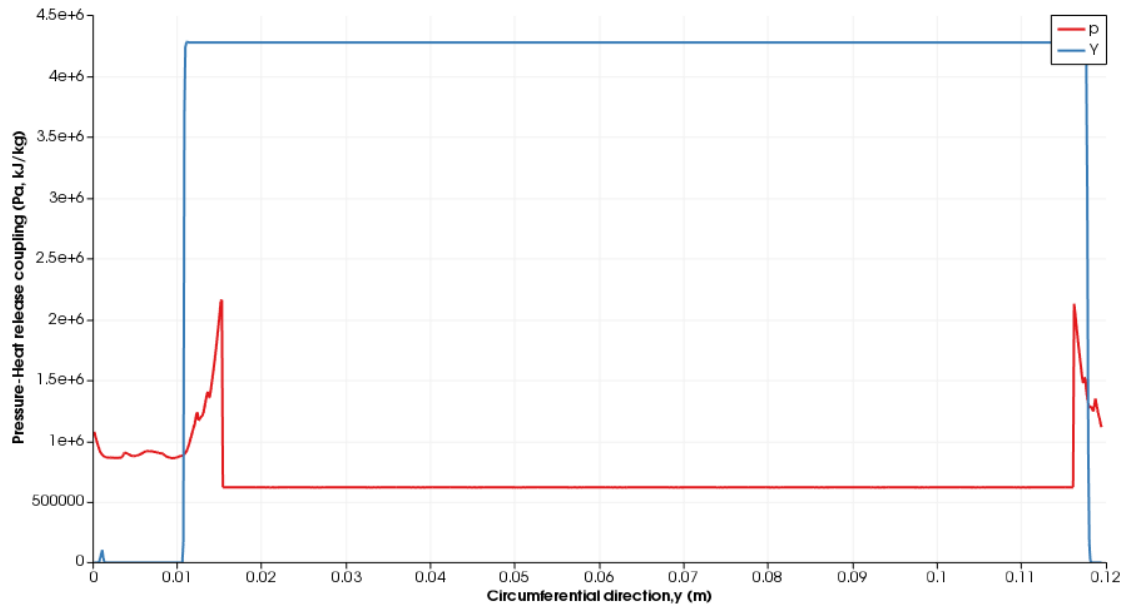


Figure 82: Image indicates decoupling of shock pressure-reaction(Unsustained Detonation for subsonic case)

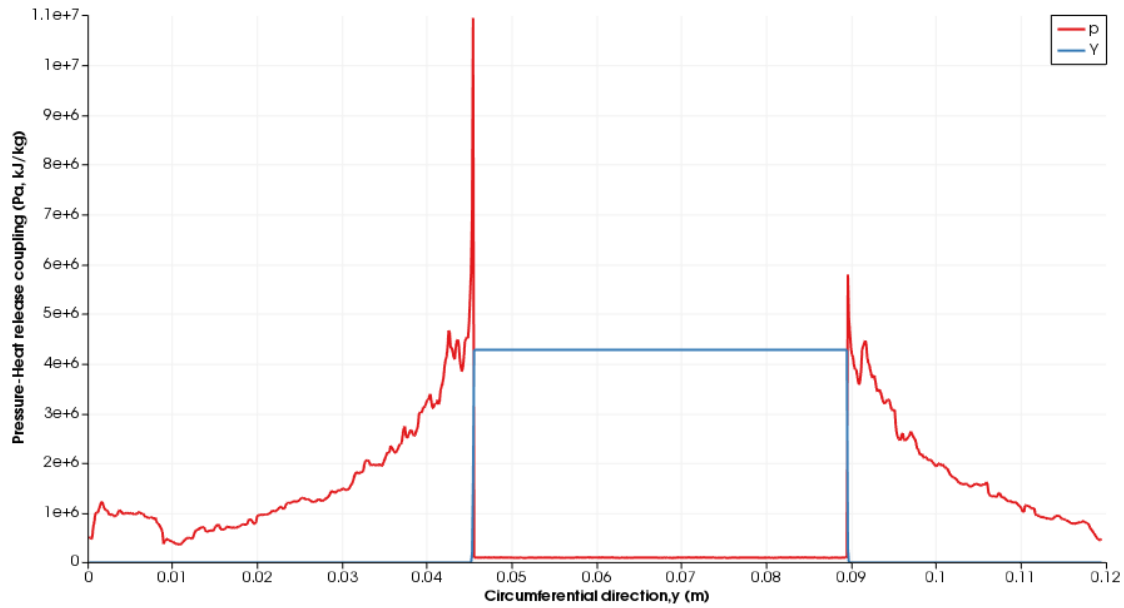


Figure 83: Pressure-reaction coupling indicating a detonation wave (supersonic case)

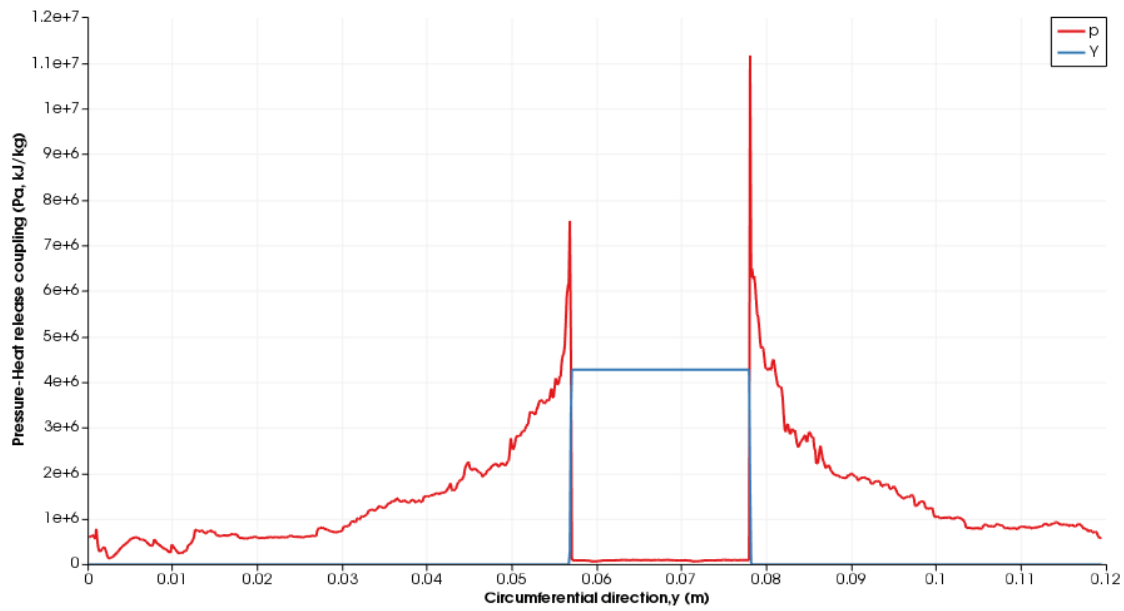


Figure 84: Coupling remains as wave travels further (supersonic case)

Chapter 6

CONCLUSION

A validation study has been conducted for the KT/KNP scheme. We were able to show that the minmod limiter produces the least disturbance amplitude for the various limiters considered, with superbee performing the worst. Critical CFL before a significant jump in disturbance amplitude is observed increases with increasing pressure ratio. For pressure ratios up to 5, a maximum CFL number of $\simeq 0.15$ gives a 1% degree of accuracy using the minmod limiter, with lower ratios allowing for higher accuracy. $CFL \leq 0.1$ agrees with rules of thumb disseminated throughout the community. For the simple, over-driven detonation problem, KNP scheme recovers the oscillating solution of a detonation wave. Findings from a significant number of linear suitability analysis show however that the resulting ZND solution gives a single mode instability [NNL00] for the chosen choice of parameters for a pulsating detonation wave, with an overdrive factor of 1.6. However, this mode of oscillation is not seen in this case, even after increasing the resolution up to 100 cells per $L_{1/2}$. Instead, we obtain a profile that is rather irregular in pattern. We note that the peak pressure values obtained for each case was above 100 kPa (about 108 kPa), quite a high value considering that most of the schemes discussed in [NNL00] show a peak pressure value below 100 kPa, for the various cells considered.

Cold start conditions for initiation using the Wolanski engine have been determined, and the initiation of a propagating detonation wave demonstrated. It has been established that regardless of the initial flow field-be it supersonic or subsonic-, a rotating detonation wave can be achieved. For the same stagnation pressures however, one would require a higher pressure ratio to initiate a detonation using a subsonic flow field, as opposed to a corresponding supersonic flow field. The significance of a highly resolved mesh in detonation simulations has been demonstrated, as this is crucial to determining physically acceptable results.

It was thus determined that under certain conditions, the KT/KNP can accurately solve the standard shock tube problem. What has so far been an anecdotal rule-of-thumb has been

formalized. It was also determined that a reactive solver based on the KT/KNP scheme is unable to accurately capture the 1-D dynamics of detonation waves. Detonations were initiated in a premixed RDE setup, but always resulted in a detonation travelling back into the plenum section. A conditional reactive solver (for e.g one that suppresses reaction for spatial positions upstream of the injection holes) could be implemented to mimic non-premixed RDE operation.

6.1 RECOMMENDATION

Higher resolution cases (with more points per $L_{1/2}$) need to be tested for the initiation process, to see how the scheme performs under such cases, as well as to compare peak pressure and velocity values. Also, the inlet boundary condition needs to be modified to enable it adapt to on-coming flows to it, since a detonation wave will always cause a rise in pressure, which in turn will seek to drive the flow upstream of it backwards. This will entail creating a custom boundary condition, within the OpenFoam framework, to that effect. Having been constrained by the validation results to a CFL number of 0.1, it becomes necessary to implement a numerical scheme that allows one run at even higher numbers than that, without spurious oscillations. Overall, we do not recommend using the KT/KNP scheme for detonation simulation.

Appendices

Table 1: Table of CFL and amplitude of pressure oscillations for the 4 chosen limiters for a pressure ratio of 10.

CFL	SuperBee(dp)	VanAlbada(dp)	MinMod (dp)	Vanleer(dP)
0.05	77237	115.9	83	73
0.06	267.2(*)	180.1	104.3	81.7
0.07	54742.06	230.8	127.8	117.9
0.08	36153.5	Ftr	152.9	12219.8
0.09	125380	306.8(*)	184.3	229
0.1	69167.2	407.8	204	228.6
0.11	84169.2	469.4	20716.5	297.7(*)
0.12	Ftr	512.3	297.3	406
0.13	69481.2	561.5	317.6(*)	1666.1
0.14	91258.35	703.8	364.9	12845.6
0.15	72417.5	612.3	419.4	14749.8
0.16	86432.71	716.1	484.5	17085.9
0.17	1163.2	826.7	826.7	17223.3
0.18	90264.49	842.4	581.1	18106.3
0.19	94596.69	19256.8	615.3	19600.7
0.2	83276.64	1000.2	710.8	2057.3
0.21	26541.2	1163.5	767.5	25534.1
0.22	106738	1555.1	785.5	22977.9
0.23	110998.5	15613	883.4	27202.3
0.24	93775.5	15831.4	915.9	25785.4
0.25	85220.3	17712.5	1047.2	30827.4
0.26	62853.4	20305	1121.7	30570.2

(*)represents CFL numbers that guarantee 1% error

Table 2: Table of CFL and amplitude of pressure oscillations for the 4 chosen limiters for pressure ratio of 2.

CFL	VanAlbada (dp)	SuperBee (dp)	Vanleer (dp)	MinMod (dP)
0.05	0.3	34643.01	4.4	0.5
0.06	0.4	30547.9	4.3	0.5
0.07	0.4	37165.35	5.7	0.6
0.08	703	35529.21	1419.9	0.7
0.09	0.2	43249.41	7.3	0.8
0.10	0.2	31232.47	7.5(*)	0.8
0.11	0.9	10756.2	25.3	294.5
0.12	0.1	19732.37	133.1	1.4
0.13	6.5(*)	14130.24	5408.8	4.3
0.14	34.4	14826.14	6794.1	7
0.15	106.1	16620.48	6322.3	6.8
0.16	253.8	34643.01	7206.5	8.5
0.17	1278.8	17212.33	8632.1	8.8(*)
0.18	4971.9	15320.68	9014.1	30.8
0.19	31784.9	14339.22	9914.7	18.4
0.20	19509.9	15363.5	10941.7	36.7
0.21	7001.4	14913.28	10251.1	54.2
0.22	8341.1	16516.96	12105.3	192.5
0.23	7961.2	17131.22	12138.38	3491.7
0.24	8774.5	16803.63	12444.5	3666.6
0.25	8739.9	16969.06	13510.27	5429.9
0.26	9045.1	17732.7	13384.94	5741.3

(*)represents CFL numbers that guarantee 0.1% error

Table 1 presents the various Finite Volume Methods (FVM) test cases evaluated, their CFL numbers, and pressure disturbance amplitude values(dp) for each, with a number of cases failing to run (Ftr). We also note that although all cases were set to end at the same time, (0.007),some didn't, particularly with the superbee limiter. We mark with *, the critical CFL number for the 4 limiters.

Table 3: Table of CFL and amplitude of pressure oscillations for the 4 chosen limiters using pressure ratio of 5

CFL	Vanleer(dp)	Minmod(dp)	VanAlbada (dp)	SuperBee(dP)
0.05	12.8	21.5	12.6	59117
0.06	14.7	25.6	20.7	67428
0.07	17.8	29.5	30.5	52553
0.08	1250	21.5	2212.4	40368.5
0.09	28.9	21.5	52.5	57997
0.1	32.1	21.5	69.9	61737
0.11	40.3	44.6	78.9	64700
0.12	69.8	1609.8	103	23176
0.13	114.5	64.3	104.7	63612.7
0.14	213.3(*)	69.3	142.4	66364.8
0.15	384.6	77.9	155.5	62244
0.16	4404.5	86.2	162.5	72266
0.17	10573.6	96.8	208.1	66927
0.18	11310.1	107.3	195.8(*)	76708
0.19	13127.9	124.2	2776.9	45177.8
0.2	14894.3	134.8	3472.4	96736.7
0.21	15727.9	141.5	325.2	51001
0.23	15950.7	225.6	392.4	53346
0.24	18912.9	194.6	459.8	62656
0.25	22197.9	222.7(*)	732.7	81719
0.26	21926.6	263.1	732.7	73943

(*)represents CFL numbers that guarantee 1% error

Table 4: Table of CFL and amplitude of pressure oscillations for the 4 chosen limiters using pressure ratio of 1.5

CFL	Vanleer(dp)	VanAlbada (dp)	Minmod (dp)	SuperBee(dP)
0.05	0.3	1.2	5.4	29367.75
0.06	0.5	0.9	4.7	21181.44
0.07	0.7	0.8	3.4	24962.17
0.08	1423.5	424.3	2.9	18244.87
0.09	1(*)	1.6	2.2	26378.03
0.1	45.5	0.7	2.4	28265.94
0.11	877.7	0.8	223	29249.06
0.12	2545.5	1.2	2.3	18437.71
0.13	4209.5	10.3(*)	2.3	10251.05
0.14	5618.81	38.8	2.8	10174.27
0.15	6252.36	294.7	3.7	10212.92
0.16	6537.8	1641.2	18	10523.67
0.17	7295.02	4411.2	11.7(*)	10766.38
0.18	8195.42	5256.17	41.2	10482.85
0.19	8274.45	18755.54	108.1	10448.48
0.2	8972.97	27324.18	117	11591.58
0.21	9958.36	6914.77	3386.4	11888.02
0.22	9825.24	6892.92	4181.4	12837.94
0.23	11021.36	7387.47	5482.85	12001.94
0.24	11436.22	7988.1	5624.99	12581.73
0.25	12286.66	8155.86	6345.65	12890.19
0.26	12167.49	8291.08	6166.28	14205.06

(*)represents CFL numbers that guarantee 0.1% error.

Listing 1: Codes utilized for ZND solution

```

1  gamma = 1.2;
2  P_0 = 100; #kPa
3  T_0 = 298; #K
4  #D = 2000
5  f = 1.6; #overdrive = D^2/(D_cj)^2
6  #k=0.69315
7  k = (1/2.5e-9)
8  R = 8.314; #KJ/KMolK
9  M = 28.96; #Kg/KMol
10 Rsp = R / M; # kJ/KgK
11 rho_0 = P_0 / (Rsp *T_0) #Kg/m^3;
12 c_0 = sqrt(gamma * Rsp*1000 * T_0);
13 lambda = 1;
14 Q = 50;
15 Ea= 50;
16 q = Q* Rsp *1000*T_0; # KJ/Kg
17 ea= Ea* Rsp *1000*T_0; # KJ/Kg
18 [D_cj,P_cj,rho_cj, V_cj, C_cj, M_cj, T_cj, u_cj] = znd.polyfn(P_0, rho_0, T_0, Q, gamma,Rsp,c_0);
19 D = sqrt(f)*D_cj;
20
21 tspan = [0 0.5 0.9999]
22 opt = odeset ("RelTol", 1e-12, "InitialStep", 1e-1, "MaxStep", 1e-1);#, "AbsTol", 1e-8);
23 [lambda, xt] = ode45 (@(lambda,xt) znd_integrate(lambda,xt,P_0, rho_0, T_0, Q, gamma,Rsp,c_0,D,Ea,k),
    tspan, [0,0]);
24 t12 = xt(2,2)
25 x12 = xt(2,1)
26 xr = xt(3,1)
27
28 N12 = 40; #Number of points per half-reaction zone length
29 dx = x12/N12;function [dL, dt] = znd_integrate_x(x,y,P_0, rho_0, T_0, Q, gamma,Rsp,c_0,D,Ea,k)
30 [P, U, c, rho,V,x,M,T] = overdrivenD(P_0, rho_0, T_0, Q, gamma,Rsp,c_0,D,y(1));
31 dL = ((1-y(1)).*k.*e.^(-Ea*T_0./(T)))/(U.-D);
32 dt = 1.0/(U.-D);
33 endfunction
34 xspan = 0:dx:xr;
35 #N = floor(xr/dx)+1;
36 #xspan = linspace(0,xr,N);
37 opt = odeset ("RelTol", 1e-12, "InitialStep", 1e-1, "MaxStep", 1e-1);#, "AbsTol", 1e-8);
38 [xsol, lt] = ode45 (@(x,lt) znd_integrate_x(x,lt,P_0, rho_0, T_0, Q, gamma,Rsp,c_0,D,Ea,k), xspan,
    [0,0]);
39 x = xsol;
40 lambda = lt(:,1);
41 t = lt(:,2);
42
43 P = 0*lambda;
44 U = 0*lambda;
45 c = 0*lambda;
46 rho = 0*lambda;
47 V = 0*lambda;
48 xFD = 0*lambda;
49 M = 0*lambda;
50 T = 0*lambda;
51 j = length(lambda);
52
53 for i = 1:j;
54 [P(i), U(i), c(i), rho(i),V(i),xFD(i),M(i),T(i))] = overdrivenD(P_0, rho_0, T_0, Q, gamma,Rsp,c_0,D,
    lambda(i));
55 end
56 rate = k*(1-lambda).*e.^(-Ea*T_0./T);
57 # figure (1)
58 # plot (x,P/P_0, "or" )
59 # grid
60 # title ('Pressure vs x')
61 # figure (2)
62 # plot (x, U/c_0)
63 # grid
64 # title ('Particle Velocity vs x')
65 # figure (3)
66 # plot (t/t12,T/T_0, "r" )
67 # grid
68 # title ('Temperature vs t')
69 # figure (4)

```



```

70 # plot (x,rho/rho_0, "+bk" )
71 # grid
72 # title ('Density vs x')
73 # figure (5)
74 # plot (x,lambda, "+bk" )
75 # grid
76 # title ('lambda vs x')
77 # figure (6)
78 # plot (t/t12,lambda, "+bk" )
79 # grid
80 # title ('lambda vs t')
81 # figure (7)
82 # plot (x/x12,T/T_0, "r" )
83 # grid
84 # title ('Temperature vs x')
85 # figure (8)
86 # plot (t/t12, rate)
87 # title ('Rate vs time')
88
89 # fid_p = fopen("p", "r+t")
90 # fid_U = fopen("U", "r+t")
91 # fid_Y = fopen("Y", "r+t")
92 # fid_T = fopen("T", "r+t")
93
94 # fskipl (fid_p,22)
95 # fskipl (fid_T,22)
96 # fskipl (fid_Y,22)
97 # fskipl (fid_U,22)
98
99 # m = length(P);
100 # for i = m:-1:1
101     %fputs(fid, '%f \n', P(1:4))
102     # fprintf(fid_p, '%30.15f', P(i)*1000)
103     %fputs(fid, '%f \n', P(1:5))
104     # fskipl (fid_p)
105
106     # fprintf(fid_T, '%30.15f', T(i))
107     # fskipl (fid_T)
108
109     # fprintf(fid_Y, '%30.15f', q*(1-lambda(i)))
110     # fskipl (fid_Y)
111
112     # fprintf(fid_U, '(%30.15f 0 0)', U(i)-U(end))
113     # fskipl (fid_U)
114
115 # endfor
116 #fclose(fid_p)
117 #fclose(fid_U)
118 #fclose(fid_Y)
119 #fclose(fid_T)
120
121
122
123 %open the files for reading only
124 fid_P = fopen('p','rt');
125 fid_U = fopen('U','rt');
126 fid_T = fopen('T','rt');
127 fid_Y = fopen('Y','rt');
128
129 %reads the contents of the files into variables. Those variables are
130 %cell array variables
131 P_file_data = textscan(fid_P, '%s', 'Delimiter', '\n');
132 U_file_data = textscan(fid_U, '%s', 'Delimiter', '\n');
133 T_file_data = textscan(fid_T, '%s', 'Delimiter', '\n');
134 Y_file_data = textscan(fid_Y, '%s', 'Delimiter', '\n');
135
136 %This is to eliminate one layer of the cell array. I'm not 100% sure why it's
137 %needed, but it is.
138 P_file_data = P_file_data{1};
139 U_file_data = U_file_data{1};
140 T_file_data = T_file_data{1};
141 Y_file_data = Y_file_data{1};
142
143

```

```

144 %close the original files
145 fclose(fid_P);
146 fclose(fid_U);
147 fclose(fid_T);
148 fclose(fid_Y);
149
150 %see how many points are in the znd solution
151 m = length(P);
152
153 %replace the values with the znd data in the proper lines
154 for i = m:-1:1
155     P_file_data{21+m-i} = sprintf('%f', P(i)*1000);
156     U_file_data{21+m-i} = sprintf('%f_0_0', U(i)-U(end));
157     T_file_data{21+m-i} = sprintf('%f', T(i));
158     Y_file_data{21+m-i} = sprintf('%f', ((1 - lambda(i))*q));
159 end
160
161 %open files for writing. If you use the same filenames, i.e. p, T, Y, U
162 %the contents of the file will be overwritten
163 fid_P = fopen('p', 'wt');
164 fid_U = fopen('U', 'wt');
165 fid_T = fopen('T', 'wt');
166 fid_Y = fopen('Y', 'wt');
167
168 %write the file data to the file
169 fprintf(fid_P, '%s\n', P_file_data{:});
170 fprintf(fid_U, '%s\n', U_file_data{:});
171 fprintf(fid_T, '%s\n', T_file_data{:});
172 fprintf(fid_Y, '%s\n', Y_file_data{:});
173
174 %close the files that you are outputting to
175 fclose(fid_P);
176 fclose(fid_U);
177 fclose(fid_T);
178 fclose(fid_Y);
179
180
181
182
183 #print -dpng plot1.png
184
185
186
187
188 #print -dpng plot1.png
189
190 %function evaluates CJ properties
191 function [D_cj, P_cj, rho_cj, V_cj, C_cj, M_cj, T_cj, u_cj] = znd_polyfn(P_0, rho_0, T_0, Q, gamma, Rsp, c_0)
192     #M_cj = (2 * q/R*T_0)
193     phi = (2 * Q*((gamma^2)-1)/ gamma)^0.5;
194     #p(x) = m^2 - 1 - phi * m.
195     p = [1, -phi, -1];
196     a = roots(p);
197     M_cj = max(a(1), a(2));
198     #D_cj = M_cj * C_cj;
199     P_cj = (1 + (gamma * (M_cj^2)))/(1 + gamma);
200     P_cj = P_cj * P_0;
201     V_cj = (1 + (gamma * (M_cj^2)))/(M_cj^2 * (1 + gamma));
202     V_cj = V_cj/rho_0;
203     T_cj = (P_cj * V_cj)/Rsp;
204     rho_cj = 1/V_cj;
205     C_cj = sqrt(gamma * Rsp*1000 * T_cj);
206     D_cj = M_cj * c_0;
207     u_cj = D_cj*(1-rho_0/rho_cj);
208
209 %function evaluates reaction progress as a function of space.
210 function [dL, dt] = znd_integrate_x(x,y,P_0, rho_0, T_0, Q, gamma, Rsp, c_0, D, Ea, k)
211 [P, U, c, rho, V, x, M, T] = overdrivenD(P_0, rho_0, T_0, Q, gamma, Rsp, c_0, D, y(1));
212 dL = ((1-y(1)).*k.*e.^(-Ea*T_0./(T)))/(U.-D);
213 dt = 1.0/(U.-D);
214 endfunction
215
216 function [dx, dt] = znd_integrate(lambda,x,P_0, rho_0, T_0, Q, gamma, Rsp, c_0, D, Ea, k)
217 [P, U, c, rho, V, x, M, T] = overdrivenD(P_0, rho_0, T_0, Q, gamma, Rsp, c_0, D, lambda);

```

```

218 dx = (U.-D)/((1-lambda).*k.*e.^(-Ea*T.0./(T)));
219 dt = 1.0/((1-lambda).*k.*e.^(-Ea*T.0./(T)));
220 endfunction
221
222 %function evaluates parameters of an overdriven function.
223 function [P, U, c, rho,V,x,M,T] = overdrivenD(P_0, rho_0, T_0, Q, gamma, Rsp,c0,D,lambda)
224     M = D/c0;
225     x = (((M - M.^-1)^2 - 2.*(gamma.^2 - 1) .* (lambda .* Q / gamma))./((gamma + 1).^2 .* M.^2)).^0.5;
226     V = ((gamma + (M^-2))/(gamma + 1)) - x;
227     rho = V/rho_0;
228     P = (((gamma*M^2) + 1)/(gamma + 1)) + gamma * M^2 * x;
229     P = P * P_0;
230     U = ((M - (M^-1))/(gamma + 1)) + M*x;
231     U = U * c0;
232     rho = 1./V;
233     T = P .* V./Rsp;
234     c = sqrt(gamma.*Rsp.*1090.*T);

```

Listing 2: Shortcuts for ComputeCanada and Linux

```

1 cp -r chebzy@graham.computeCanada.ca:~/scratch/shockTube outputlim %copy files from computeCanada to
  dir
2 scp -r chebzy@graham.computeCanada.ca:~/scratch/detonation_10pts grahamtestcomplete %copy files from
  computeCanada to dir
3 scp -r ./shockTube chebzy@graham.computeCanada.ca:~/shockTubex
4 scp -r ~/grahamtestcomplete/detonation_10pts chebzy@graham.computeCanada.ca:~/ //to send to remote
5 /home/chebzy/scratch/detonation_20pts/system
6 rm -r *-05 remove multiple files ending with -05
7
8 # swapoff -v /swapfile # rm /swapfile %delete swapfile formed when you edit a file and exit without
  saving
9 rm -rf /home/chebzy/grahamtestcomplete/detonation_1pts2d/processor*
10 pwd print working directory
11 #!/bin/bash
12 #SBATCH --time=00:01:00
13 #SBATCH account=def-ckiyand
14 scancel %Use command with job ID to cancel a job
15 #!/bin/bash
16 #SBATCH --time=00:30:00
17 #SBATCH --account=def-ckiyand
18 module purge
19 module load openfoam/5.0
20 cp -r ~/shockTube scratch/
21 cd scratch/shockTube
22 cp -r 0.orig 0
23 /home/chebzy/scratch/detonation_10ptsorre05/system
24 blockMesh
25 setFields
26 rhoCentralFoam
27 postProcess -func sampleDict %post process sampling points for data
28
29 Flow visualization over a server using Paraview %Visualize your simulations
30 1. First, install on your device the same ParaView version as the one available on the cluster you will be
  using; log into Cedar or Graham and start a serial GPU interactive job.
31 salloc --time=1:00:0 --ntasks=1 mem-per-cpu=2024 --gres=gpu:1 --account=
32 make sure the paraview you are running is the same as that in the cluster. Load the variables gradually,
  one at a time.
33
34 salloc --time=0:30:0 --ntasks=16 --cpus-per-task=2 --mem-per-cpu=4024 account=def-ckiyand
35 module load paraview-offscreen/5.3.0
36 srun pvserver mesa
37 ssh chebzy@cedar.computeCanada.ca -L 11111:cdr768:11111
38
39 for effective use of the processor, run your case in the decomposed foam. In paraview. This gives the best
  processing speed as in visualization speed
40 salloc --time=1:30:0 --ntasks=32 --cpus-per-task=3 --mem-per-cpu=3024 --account=def-ckiyand.
41 module load paraview-offscreen/5.3.0
42 srun pvserver --mesa
43
44 cp -r /dir /dir %used this to copy files
45
46 cp -r /home/chebzy/scratch/detonation_80ppts/postProcessing/sampleDict/pp_scriptplot.py scratch/
  case2wallsoolong16/postProcessing/sampleDict

```

```
47
48 module load python                               %load python compute canada
49 python pp-scriptplot.py                          % run python script for making plots of pshock against space and time
50
51 scp took long enough. First case was at 11:46 last was 16:48
52 postProcess -func 'patchintegrate(name=outlet,mdot)' %Evaluate mass flow rate across a surface (outlet in
    this case)
53 touch "${PWD##*/}.foam" % to create .foam file needed for visualization using paraview
54
55
56 When building your case to test various velocity samples, make sure that the inlet and the uniform
    velocity values are the same if not you wil end up having a floating point exception error which will
    make lead to a fork error in computecanada.
57
58 cd globusconnectpersonal-2.3.5/
59 ./globusconnectpersonal
60
61
62 Setting field default values
63 --> FOAM Warning :
64     From function bool setCellFieldType(const Foam::word&, const Foam::fvMesh&, const labelList&, Foam::
        Istream&) [with Type = Foam::Vector<double>; Foam::labelList = Foam::List<int >]
65
66 ControlDict problem ensiure that starttime corresponds to 0 or desired value (n mapped cases)
67
68 Codes are available at Github : chebzy_thesis
```

Bibliography

- [BLWC13] Eric M Braun, Frank K Lu, Donald R Wilson, and José Camberos. Airbreathing rotating detonation wave engine cycle analysis. *Aerospace Science and Technology*, 27(1):201–208, 2013.
- [BM95] Anne Bourlioux and Andrew J. Majda. Theoretical and numerical structure of unstable detonations. *Philosophical Transactions: Physical Sciences and Engineering*, 350(1692):29–68, 1995.
- [BP94] T Bussing and George Pappas. An introduction to pulse detonation engines. In *32nd Aerospace Sciences Meeting and Exhibit*, page 263, 1994.
- [Bus96] Thomas RA Bussing. Rotary valve multiple combustor pulse detonation engine, May 7 1996. US Patent 5,513,489.
- [BZV06] Fedor A Bykovskii, Sergey A Zhdan, and Evgenii F Vedernikov. Continuous spin detonations. *Journal of Propulsion and Power*, 22(6):1204–1216, 2006.
- [Cfd] Cfdsupport. Inletoutlet boundary condition. <https://www.cfdsupport.com/OpenFOAM-Training-by-CFD-Support/node114.html>. Accessed: 2018-11-25.
- [CFM⁺92] G Carrier, F Fendell, R McGregor, S Cook, and M Vazirani. Laser-initiated conical detonation wave for supersonic combustion. *Journal of Propulsion and Power*, 8(2):472–480, 1992.
- [Cha99] David Leonard Chapman. Vi. on the rate of explosion in gases. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 47(284):90–104, 1899.
- [CIR52] Richard Courant, Eugene Isaacson, and Mina Rees. On the solution of nonlinear hyperbolic differential equations by finite differences. *Communications on Pure and Applied Mathematics*, 5(3):243–255, 1952.

- [Dah11] AE Dahoe. Tutorial on deflagagation and detonations. Joint European summer school on fuel cell and hydrogen technology, University of Ulster, United Kingdom, 8 2011.
- [Dör43] Werner Döring. Über den detonationsvorgang in gasen. *Annalen der Physik*, 435(6-7):421–436, 1943.
- [EGL91] Shmuel Eidelman, W Grossmann, and I Lottati. Review of propulsion applications and numerical simulations of the pulsed detonation engine concept. *Journal of Propulsion and Power*, 7(6):857–865, 1991.
- [FD79] Wildon Fickett and William C Davis. *Detonation: theory and experiment*. Dover Publications, New York, 1979.
- [God57] Sergei Konstantinovich Godunov. Difference method of computation of shock waves. *Uspekhi matematicheskikh nauk*, 12(1):176–177, 1957.
- [God59] Sergei Konstantinovich Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik*, 89(3):271–306, 1959.
- [GSMK12] Boris E Gelfand, Mikhail V Silnikov, Sergey P Medvedev, and Sergey V Khomik. *Thermo-gas dynamics of hydrogen combustion and explosion*. Springer Science & Business Media, 2012.
- [HL94] Thomas Y Hou and Philippe G LeFloch. Why nonconservative schemes converge to wrong solutions: error analysis. *Mathematics of computation*, 62(206):497–530, 1994.
- [hys16] hysafe.org2016. Detonations and shockwaves. detonations and shock-waves - fundamentals of hydrogen safety, course materials., 08 2016.
- [Jou05] E Jouguet. Sur la propagation des réactions chimiques dans les gaz. *Journal de Mathématiques Pures et Appliquées*, 1:347–425, 1905.
- [Kai09] K Kailasanath. Research on pulse detonation combustion systems: a status report. In *47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, page 631, 2009.
- [KBS15] Matvey Kraposhin, Arina Bovtrikova, and Sergei Strijhak. Adaptation of kurganov-tadmor numerical scheme for applying in combination with the piso

- method in numerical simulation of flows in a wide range of mach numbers. *Procedia Computer Science*, 66:43–52, 2015.
- [KH13] CB Kiyanda and AJ Higgins. Photographic investigation into the mechanism of combustion in irregular detonation waves. *Shock Waves*, 23(2):115–130, 2013.
- [KNP01] Alexander Kurganov, Sebastian Noelle, and Guergana Petrova. Semidiscrete central-upwind schemes for hyperbolic conservation laws and hamilton–jacobi equations. *SIAM Journal on Scientific Computing*, 23(3):707–740, 2001.
- [KS13] Kailas Kailasanath and DA Schwer. Rotating detonation engine research at nrl. Technical report, 2013.
- [KT00] Alexander Kurganov and Eitan Tadmor. New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations. *Journal of Computational Physics*, 160(1):241–282, 2000.
- [KWG11] Jan Kindracki, Piotr Wolański, and Zbigniew Gut. Experimental research on the rotating detonation in gaseous fuels–oxygen mixtures. *Shock waves*, 21(2):75–84, 2011.
- [LB14] Frank K Lu and Eric M Braun. Rotating detonation wave propulsion: experimental challenges, modeling, and engine concepts. *Journal of Propulsion and Power*, 30(5):1125–1142, 2014.
- [Lee08] John H. S. Lee. *The Detonation Phenomenon*. Cambridge University Press, 2008.
- [MMM98] Franklin Mead, Jr, Leik Myrabo, and Donald Messitt. Flight and ground tests of a laser-booted vehicle. In *34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, page 3735, 1998.
- [MTE17a] LF Gutiérrez Marcantoni, J Tamagno, and S Elaskar. Two-dimensional numerical simulations of detonation cellular structures in h₂o₂ar mixtures with openfoam®. *International Journal of Hydrogen Energy*, 42(41):26102–26113, 2017.
- [MTE17b] LF Gutiérrez Marcantoni, J Tamagno, and Sergio Elaskar. rhocentralrffoam: An openfoam solver for high speed chemically active flows–simulation of planar detonations–. *Computer Physics Communications*, 219:209–222, 2017.
- [Nic57] R. B Morrison JA Nicholls, HR Wilkinson. Intermittent detonation as a thrust-producing mechanism. *Journal of jet propulsion*, 27(5):534–541, 1957.

- [NNL00] H.D. Ng, N. Nikiforakis, and J.H.S Lee. Assessment of high resolution centered scheme for detonation modelling. Technical report, McGill University, Montreal, Canada, 2000.
- [NT90] Haim Nessyahu and Eitan Tadmor. Non-oscillatory central differencing for hyperbolic conservation laws. *Journal of computational physics*, 87(2):408–463, 1990.
- [Ope16] OpenCFD. *OpenFOAM - The Open Source CFD Toolbox - User’s Guide*. OpenCFD Ltd., United Kingdom, 5 edition, 7 April 2016.
- [PFHS15] Daniel E Paxson, Matthew Fotia, John Hoke, and Frederick Schauer. Comparison of numerically simulated and experimentally measured performance of a rotating detonation engine. In *53rd AIAA Aerospace Sciences Meeting*, page 1101, 2015.
- [PJB94] Kumar A. (eds) Powers J.M. Buckmaster J., Jackson T.L. Oblique detonations: Theory and propulsion applications. In *Combustion in High-Speed Flows. ICAS-E/LaRC Interdisciplinary Series in Science and Engineering, vol 1*. Springer, Dordrecht, pages 345–347, 1994.
- [Sha15] Nihar N Shah. Computer modeling of a rotating detonation engine in a rocket configuration. Technical report, Air force Institute of Technology Wright-Patterson AFB OH Graduate School of Engineering and Management, 2015.
- [ST65] KI Shchelkin and Ya K Troshin. Gas dynamics of detonations. *Mono Book Corp., Baltimore*, 10, 1965.
- [Tat15] Christopher Tate. Investigation of pulse detonation engines: the effect of variable blockage ratio on the deflagration to detonation transition. 2015.
- [TFW10] Akihiro Tobita, Toshitaka Fujiwara, and Piotr Wolanski. Detonation engine and flying object provided therewith, August 31 2010. US Patent 7,784,267.
- [Tor] Eleuterio F Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*.
- [TSS⁺14] Colin A Towery, Katherine M Smith, Prateek Shrestha, Peter E Hamlington, and Marthinus Van Schoor. Examination of turbulent flow effects in rotating detonation engines. In *44th AIAA Fluid Dynamics Conference*, page 3031, 2014.
- [VJ14] D Mahaboob Valli and DRTK Jindal. Pulse detonation engine: Parameters affecting performance. *International Journal of Innovative Research in Science, Engineering and Technology*, 3(4), 2014.

- [VL76] B Van Leer. Muscl, a new approach to numerical gas dynamics. In *Computing in Plasma Physics and Astrophysics*, 1976.
- [VM63] BV Voitsekhovskii and VV Mitrofanov. M. e⁻ topchiyaa, structure of the detonation front in gases [ia russian], izd. *Sibirsk. Otd. Akad. Nauk SSSR, Novosibirsk*, 1963.
- [VN42] John Von Neuman. Theory of detonation waves. Technical report, Institute for Advanced Study Princeton NJ, 1 Einstein Drive Princeton, New Jersey 08540 USA, 1942.
- [WTJF98] Henry G Weller, Gavin Tabor, Hrvoje Jasak, and Christer Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in physics*, 12(6):620–631, 1998.
- [Zel40] Ya B Zeldovich. On the theory of detonation propagation in gaseous systems. *Zh. Eksp. Teor. Fiz*, 10(5):542–568, 1940.
- [ZM10] Olindo Zanotti and Gian Mario Manca. A very short introduction to godunov methods. *Albert Einstein Institute, Max-Planck-Institute for Gravitational Physics, Potsdam, Germany*, 2010.
- [ZWW] Rui Zhou, Dan Wu, and Jianping Wang. Progress of continuously rotating detonation engines. *Chinese Journal of Aeronautics*, 29.