**Modeling of the Chamber Flowfield in a
Rod-and-Tube Configuration Solid Rocket Motor**

by

John T. Weaver

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements
for the degree of

Master of Applied Science

Department of Mechanical and Aerospace Engineering
Ottawa-Carleton Institute for
Mechanical and Aerospace Engineering

Carleton University

Ottawa, Ontario

May, 2004

© Copyright 2004, J.T. Weaver

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-93952-9

Our file Notre référence

ISBN: 0-612-93952-9

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this dissertation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de ce manuscrit.

While these forms may be included in the document page count, their removal does not represent any loss of content from the dissertation.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

Currently, DRDC Valcartier is designing a prototype rod-and-tube configuration solid propellant rocket motor that will propel a hypersonic velocity missile. This configuration will incorporate a very low port-to-throat area ratio, which in turn results in very high velocity propellant gas traveling across burning propellant surfaces, particularly near the nozzle end of the rocket. Typically the phenomenon known as erosive burning is present in low port-to-throat ratio motors. While numerical and lumped parameter models are available to design and analyze solid propellant rocket motors and nozzles, many of them provide solutions based on the assumption of quasi-steady flow. Due to the high pressure, high velocity and highly transient nature of the flows expected in the motor under design, it is believed that a CFD simulation will better model the time-dependent phenomena that occur during the functioning of a motor of this type. This simulation couples the fluid dynamics and heat transfer of the gas flowfield within the rocket port to the nozzle and the regression rate of the propellant which is governed by the Lenoir-Robillard erosive burning law. By incorporating regression into the model, the information provided by the resulting time-accurate solution will enable a much improved understanding of the flow phenomena within this rod-and-tube grain motor, which in turn will help in the design of both the motor and the nozzle. The methodology developed within this thesis is widely applicable and supports the use of different erosive burning models, as well as non-steady-state three-dimensional CFD modeling.

Acknowledgements

I would like to thank and acknowledge:

- Dr. J.E.D. Gauthier, my thesis co-supervisor, for his support, understanding, and faith in an old ex-cadet;
- Dr. R.A. Stowe, my thesis co-supervisor, for his support, trust and vision;
- the financial support provided by Defence Research and Development Canada – Valcartier and the Department of Mechanical and Aerospace Engineering at Carleton University;
- my parents for their nurturing of a desire to understand why, and their long distance support;
- Courtney, Ashley and Serena, my children, who have taught me about some of the most important parts of life; and
- Rebecca, my wonderful wife, whose love and support has allowed the pursuit of a dream to become a reality.

Table of Contents

Acceptance Page	ii
Abstract	iii
Acknowledgements	iv
Table of Contents	v
List of Tables	viii
List of Figures	ix
Chapter 1 Introduction	1
1.1 Introduction	1
1.2 Functioning of a Solid Rocket Motor.....	3
1.2.1 GAP-ADN Propellant	5
1.2.2 Combustion Chamber Pressure	6
1.2.3 Initial Propellant Temperature	7
1.2.4 Combustion Gas Temperature	10
1.2.5 Other Physical Factors Affecting Burning Rate	10
1.3 Grain Design Considerations.....	11
1.3.1 Rod-and-Tube Grain.....	12
1.3.2 Plug Nozzle	13
1.4 Solid Propellant Rocket Motor Modeling	15
1.4.1 Characteristics of Solid Rocket Motor Combustion.....	15
1.4.2 Design Tools	16
1.4.2.1 SPP.....	17
1.4.2.2 Introduction to CFD	18
1.4.2.3 Multi-Physics Simulation of Solid Rocket Combustion...	19
1.4.2.4 Use of a Commercial CFD Application.....	20
1.5 Scope of Thesis.....	21
Chapter 2 Erosive Burning	24
2.1 Introduction	24
2.2 Erosive Burning	27
2.2.1 Phenomenological Heat Transfer	28
2.2.2 Integral Boundary Layer Analysis.....	36
2.2.3 Modification of the Propellant Combustion Mechanism	40
2.2.4 Chemically Reacting Boundary Layer Analysis	41
2.2.5 Erosive Burning Summary	46
Chapter 3 CFD Application	53
3.1 Introduction	53
3.1.1 High-Level Solution Process	53
3.2 CFD-ACE+ Solution Procedure.....	55
3.2.1 Discretization	55
3.2.2 Velocity-Pressure Coupling	56
3.2.3 Relaxation	63
3.2.4 Linear Equation Solvers.....	65
3.3 CFD-ACE+ Module Overview	67
3.3.1 Flow Module	67
3.3.1.1 Implementation of Flow Module.....	68

3.3.2	Heat Transfer Module.....	69
3.3.2.1	Implementation of Heat Transfer Module.....	69
3.3.3	Turbulence Module.....	70
3.3.3.1	Treatment of the Burning Grain Surface	71
3.3.3.2	Implementation of Turbulence Module	71
3.3.3.3	Consideration of Surface Mass Injection	83
3.3.4	Chemistry Module.....	86
3.3.4.1	Implementation of Chemistry Module	86
3.3.5	Grid Deformation Module	88
3.3.5.1	Implementation of the Grid Deformation Module.....	89
3.4	CFD Summary.....	89
Chapter 4	<i>Boundary and Initial Conditions</i>	92
4.1	Introduction	92
4.1.1	Instantaneous Chemistry Model	93
4.1.2	Equilibrium Chemistry Model	94
4.1.3	Finite Rate Chemistry Model.....	95
4.1.4	Turbulence and Chemically Reacting Flows	95
4.1.5	Simplifying Assumptions.....	96
4.2	CEA Software Program.....	98
4.2.1	Chemical Equilibrium.....	98
4.2.2	Theoretical Rocket Performance	101
4.2.2.1	Infinite Area Combustor (IAC) Calculations	109
4.2.3	Accuracy of CEA as a Prediction Tool.....	110
4.3	Boundary Conditions Used	111
4.4	Initial Conditions.....	116
4.5	Boundary and Initial Condition Summary.....	119
Chapter 5	<i>Grain Regression Model</i>	120
5.1	Introduction	120
5.1.1	Method Overview	121
5.2	Grid Development	124
5.3	Calculation of Lenoir-Robillard Constants	127
5.4	Grid Regression	132
5.5	Grain Regression Summary	134
Chapter 6	<i>Results and Analysis</i>	136
6.1	Introduction	136
6.2	Analytical Model.....	136
6.2.1	Combustion Chamber Calculations.....	137
6.2.2	Nozzle Flow Calculations.....	139
6.3	Steady-State Results and Analysis	142
6.3.1	Discretization	142
6.3.1.1	Effect of Varying Spatial Differencing	143
6.3.2	Grid Independence	146
6.3.3	Results and Discussion	148
6.3.3.1	Comparison with Reference Solutions	148
6.3.3.2	Validity of High Reynolds Number Turbulence Model... ..	152
6.3.3.3	Use of a Low Reynolds Number Turbulence Model	159

6.4	Non-Steady-State Results and Analysis	161
6.4.1	General	161
6.4.2	Comparison of the Two Burning Laws	164
6.4.2.1	Location of Peak Turbulence Intensity	171
6.4.3	Effect of Initial Conditions	173
6.4.4	Nozzle Inlet Profiles	175
6.5	Discussion	179
6.5.1	General	179
6.5.2	Use of a 1-D Erosive Burning Model	180
6.5.3	Sources of Error	182
6.5.3.1	Choice of Erosive Burning Constants	183
6.5.3.2	Boundary and Initial Conditions	184
6.5.3.3	Mass Flow Rate Update	185
6.5.3.4	Grid Issues	187
6.5.3.5	Effect of Time Step	189
6.5.3.6	Threshold Velocity	191
6.6	Summary of Key Findings	192
Chapter 7	<i>Conclusions and Recommendations</i>	194
7.1	Conclusions	194
7.2	Recommendations for Future Work	197
	<i>Bibliography</i>	201
	<i>Appendix A – Selection of CFD Application</i>	206
	<i>Appendix B – Example CEA Output</i>	270
	<i>Appendix C – User Subroutine for Non-Erosive Burning</i>	277
	<i>Appendix D – User Subroutine for Erosive Burning</i>	291
	<i>Appendix E – Technical Papers Supporting Thesis</i>	306

List of Tables

Table 4-1	Boundary condition values.	114
Table 5-1	Protocols used for non-steady-state runs in terms of time step per iteration and mass flow rate updates.	123
Table 5-2	Burning data summary for various propellant formulations.	131
Table 6-1	Values of key burning rate characteristics which remain constant throughout the calculated burning time.	139
Table 6-2	Relative percentage difference ($\Delta\%$) between UDS and Blended CDS.	145
Table 6-3	Comparison of CFD-ACE+ steady-state results with reference calculations.	150
Table 6-4	Values of key constants and variables used in determining y^+ and V_{inj}^+ .	155
Table 6-5	Turbulent boundary layer characteristics as a function of chamber location.	157
Table 6-6	Comparison of time required for propellant 'burn out' and 'all burnt' to be reached for simple burning and three erosive burning cases.	165
Table 6-7	Variation in total mass flow rates as a function of time for simple and erosive burning cases.	169
Table 6-8	Total mass flux and erosive augmentation ratio compared for simple and erosive burning cases.	170
Table 6-9	Port cross section areas as a function of time and burning law.	171
Table 6-10	Variation in total mass flow rates as a function of time for erosive burning where initial conditions (IC) were simulating STP and 6.9 MPa at 996 K.	175

List of Figures

Figure 1.1	Simplified cross section of a typical tactical solid propellant rocket motor [Sutton and Biblarz (2001)].	3
Figure 1.2	Burning rate characteristics of a double base (DB) propellant as well as three composite modified double base (CMDB) propellants [Sutton and Biblarz (2001)].	7
Figure 1.3	Effect of propellant temperature upon burning time and chamber pressure [Sutton and Biblarz (2001)].	8
Figure 1.4	Simplified diagrams of some grain configurations [Sutton and Biblarz (2001)].	12
Figure 1.5	Simplified diagram of a rod-and-tube configuration solid rocket motor.	13
Figure 1.6	Relation of the thrust coefficient to the Pressure Ratio for 20% Plug Nozzle, Bell Nozzle and Ideal Nozzle [Ito et al. (1999)].	14
Figure 2.1	Illustration of experimental tablet method used by Marklund and Lake to obtain erosive burning data [Marklund and Lake (1960)].	34
Figure 2.2	Comparison of experimental erosive burning rate data versus predicted values [King (1975)].	36
Figure 2.3	Schematic diagram of King's Two-Flame model of erosive burning [King(1977)].	41
Figure 2.4	The physical model considered by Razdan and Kuo in their turbulent boundary layer model [Razdan and Kuo (1978)].	42
Figure 2.5	The effect of the cross-flow velocity on the inner region of the turbulent boundary layer [Razdan and Kuo (1978)].	45
Figure 2.6	Comparison of predicted burning rates to experimental results at various pressures and crossflow velocities [Razdan and Kuo (1982)].	48

Figure 2.7	Erosive burning rate data highlighting the presence of a threshold velocity [Zucrow et al. (1964)].	49
Figure 2.8	Effect of normal burning rate on the erosive burning augmentation factor [Razdan and Kuo (1982)].	50
Figure 2.9	Calculated erosive burning augmentation factor illustrating the effect of the propellant surface roughness (R_h) [Razdan and Kuo (1982)].	52
Figure 3.1	Two grid staggering arrangements used in CFD codes [Ferziger and Peric (2002)].	57
Figure 3.2	Typical non-dimensional velocity profile for a turbulent flow over a flat plate [Hoffmann and Chiang (2000)].	74
Figure 3.3	View of the grid depicting the aft end of the grain and the converging section of the nozzle (2-D axisymmetric representation) at $t=0s$.	82
Figure 4.1	Compressibility of air as a function of pressure [White (1991)].	107
Figure 4.2	Schematic diagram of the rod-and-tube solid rocket motor under consideration.	112
Figure 4.3	Computational domain used for all modeling is shown.	113
Figure 4.4	Typical ignition pressure transient portion of a solid rocket motor pressure time curve [Sutton and Biblarz (2001)].	118
Figure 5.1	View of entire computational domain with cells visible.	125
Figure 5.2	The computational domain representing the chamber and nozzle.	127
Figure 5.3	Pseudocode for the two subroutines which make up the programs used to calculate and execute the grid deformation in accordance with the appropriate burning law.	134

Figure 6.1	From the analytical solution, the crossflow velocities at four locations (normalized length as measured from the head end of the grain) are graphed as a function of burning time.	139
Figure 6.2	Static Pressure, Static Temperature, Crossflow Velocity and Kinetic Energy of Turbulence as a function of normalized length.	146
Figure 6.3	Results of numerical experiment to determine grid independence.	147
Figure 6.4	Comparison of crossflow velocity as a function of normalized chamber length for steady-state simulations.	149
Figure 6.5	Mach contours of flowfield within 2-D axisymmetric steady-state model of rod-and-tube configuration solid rocket motor.	151
Figure 6.6	Injection velocity and V_{inj}^+ as a function of normalized chamber length.	154
Figure 6.7	Turbulence Dissipation values as a function of y^+ value at three locations along chamber length.	157
Figure 6.8	Crossflow velocity profiles at different chamber locations.	158
Figure 6.9	Location of Peak Turbulence Intensity (k) from the propellant surface as a function of crossflow velocity.	159
Figure 6.10	Crossflow velocity in aft section of grain and converging portion of nozzle.	163
Figure 6.11	Comparison of grid in region of aft end of grain and converging portion of nozzle.	163
Figure 6.12	Head end Static Pressure vs time curves comparing the effect of erosive burning for three different values of β against the simple burning law.	166
Figure 6.13	Comparison of grain regression rate versus chamber length as a function of burning law.	167

Figure 6.14	Crossflow velocity as a function of axial position along the grain at four different times during the simulation (0.1 ms, 1 ms, 10 ms and 100 ms) for the simple burning and erosive burning ($\beta = 60$) cases is illustrated.	168
Figure 6.15	Crossflow velocity as a function of time is illustrated.	169
Figure 6.16	The crossflow velocity as a function of normalized length at $t = 10$ ms for the simple burning case as well as the erosive burning case for each of the β values simulated is illustrated.	170
Figure 6.17	Distance between Peak Turbulence Intensity and inlet surface as a function of crossflow velocity measured at the centreline at the same axial location.	172
Figure 6.18	Kinematic viscosity along the chamber from $0.25 \bar{L}$ to $0.99 \bar{L}$ based on the flow conditions at 1.5 ms.	173
Figure 6.19	The effect of differing initial conditions upon the static pressure versus time curves is illustrated.	174
Figure 6.20	The effect of the different initial conditions on crossflow velocity as a function of time is illustrated.	175
Figure 6.21	Crossflow velocity in aft section of grain and converging portion of nozzle.	176
Figure 6.22	Local static pressure values for simple and erosive burning cases measured at nozzle inlet.	177
Figure 6.23	Local crossflow velocity values measured as a function of port height.	177
Figure 6.24	Local velocity values for flow normal to chamber axis, measured as a function of port height.	178
Figure 6.25	Local temperature measured as a function of port height.	178
Figure 6.26	Typical pressure-time curve with and without erosive burning [Sutton and Biblarz (2001)].	179

Figure 6.27	Aft end Static Pressure vs time curves comparing the effect of erosive burning for three different values of β against the simple burning law.	183
Figure 6.28	Static Pressure versus Time curves for non-steady-state simulation using $\beta = 60$ comparing the results achieved when manually updating the mass flow rate every time step and in accordance with the update protocol described in Table 5-1.	186
Figure 6.29	View of grid at interface of two volumes close to aft end of computational domain representing the chamber following grid regression.	189
Figure 6.30	Comparison Static Pressure versus Time curves for the interval 0.011 to 0.021 s using different size time steps.	191

Nomenclature

a	speed of sound, m/s or finite volume coefficient
A	cross sectional area, m^2
A^+	van Driest damping coefficient
A_b	propellant burning surface area, m^2
a_{ij}	stoichiometric coefficients which define the number of kg-atoms i per species j
A_s	Arrhenius frequency factor in propellant surface decomposition, m/s
a_T	empirical constant influenced by the ambient grain temperature
A_{SL}, B_{SL}	Sutherland's Law specific coefficients
B	blowing injection strength parameter, dimensionless
b_k	assigned kg-atoms of element i per kg of total oxidant ($k=1$) or total fuel ($k=2$)
b_k^o	assigned kg-atoms of element i of total reactant
C_{1-4}	constants in turbulence models
c^*	characteristic velocity
C_e	mass flux at cell face
c_f	wall friction coefficient
C_F	thrust coefficient
C_p	heat capacity of combustion gas, $J/(kg \cdot K)$
$C_{p,e}$	equilibrium specific heat at constant pressure, $J/(kg \cdot K)$
$C_{p,f}$	frozen specific heat at constant pressure, $J/(kg \cdot K)$
$C_{p,r}$	reaction specific heat at constant pressure, $J/(kg \cdot K)$
C_{ps}	heat capacity of solid propellant, $J/(kg \cdot K)$
C_μ	molecular viscosity constant
d	mean particle diameter of AP, m
D	diffusion coefficient of gaseous fuel pockets, m^2/s
D_h	chamber bore diameter, m
d_p	pressure coefficient
e	Internal energy, J
E	Law of the wall empirical constant
E_{as}	Activation energy in the propellant surface decomposition, $kcal/mole \cdot K$
F	thrust, N
g	Gibbs energy for mixture (intensive property), J/kg
G	mass velocity of combustion gas, $kg/m^2 s$
G	Gibbs energy for mixture (extensive property), J/kg

g_o	gravitational constant, m/s^2
h	heat transfer coefficient or specific enthalpy of mixture,
h_o	heat transfer coefficient with zero transpiration or specific enthalpy of total reactants
H	molar enthalpy of gases, $kcal/kg$
I	relaxation factor
I_{sp}	specific impulse, s
I_{tot}	total impulse, Ns
I_{turb}	turbulence intensity
k	turbulence kinetic energy, m^2/s^2 or proportionality constant
k_{1-2}	empirical proportionality constants
k_f	rate coefficient of chemical reaction (forward)
k_T	thermal conductivity of the material or species, $W/m \cdot K$
K_n	ratio of the burning surface area to the nozzle throat area
K	turbulent kinetic energy, m^2/s^2
L	length, m
L_f	flame thickness, m
ℓ	number of chemical elements which make up the mixture
$\dot{m}$	mass flow rate, kg/s
M	Mach Number
M_p	characteristic mass of gaseous fuel pockets, kg
MW	molecular weight, $mole^{-1}$
n	exponent in linear burning rate law or turbulent boundary layer velocity profile
N	Number of species within system
n_j	kg-moles of species j per kg of mixture
n_m	number of kg-moles of mixture per unit mass
n_x	cell face normal unit vector
NG	number of gases in the mixture
NS	number of species in the mixture
P	local static pressure, N/m^2
P_k	production of turbulence
Pr	Prandtl number based upon molecular properties of gases
Q	mass flow rate of transpiring gas, $kg/m^2 s$
q_w	heat flux from the wall
Q_T	energy required to raise propellant from initial temperature to surface temperature next to the flame zone, $kcal/kg$
R	specific gas constant, $J/(kg \cdot K)$

r_o	simple or linear burning rate
r_b	total burning rate
r_e	erosive burning component
r_1, r_2	rod portion of grain outer radius and tube portion of grain inner radius, m
Re_x	Reynolds number in x direction (parallel to core gas flow)
R_h	Roughness height, m
R_u	universal gas constant, $J/(kg - mole \cdot K)$
s	specific entropy of mixture
Sc	Schmidt number $\equiv (\mu/\rho)$ based upon molecular properties of the flow
S_h	heat source term
S_m	mass correction in the control volume
St_o	Stanton Number for non-blowing conditions
S_U	upstream source term
t	time
T	ambient temperature (of grain), K
T_o	Reference temperature, $273 K$
T_f	flame temperature, K
T_i	initial solid propellant temperature, K
T_s	final temperature at propellant surface, K
u	core gas or crossflow velocity (x direction), m/s
U	centreline crossflow velocity, m/s
u_{tv}	threshold velocity, m/s
u_τ	friction velocity
v	velocity normal to the propellant surface, m/s
V	specific volume, m^3/kg
$\vec{V}$	velocity field
$\dot{V}_b$	volume flow rate, m^3/s
V_e^n	velocity vector at face centre
V_{inj}	surface mass injection velocity, m/s
V_{inj}^+	injection rate
y_T^+	dimensionless thermal sublayer thickness
y_v^+	wall function thickness
Y_k	mass fraction of the kth species
Z	compressibility factor

α	erosive constant, $in^{2.8}/lb_m^{0.8} sec^{0.2}$ or constant used in solution of $k - \omega$ model
α_B	blocking effect constant
β	dimensionless constant (used in Lenoir-Robillard burning law and in solution of $k - \omega$ model)
β_T	dimensionless transpiration parameter
$\beta^*, \sigma, \sigma^*$	constants used in solving turbulent kinetic and specific dissipation rate equations ($k - \omega$ model)
γ	ratio of specific heat at constant pressure to specific heat at constant volume
γ_e	geometrical weighting function at face e
γ_s	isentropic exponent
δ	boundary layer thickness, m
ε	turbulent dissipation rate, m^2/s^3
η_{comb}	combustion efficiency
η_j	chemical potential of species j
κ	von Karman's constant
λ	thermal conductivity, $kJ/m \cdot s \cdot K$
μ	viscosity of combustion gases, $kg/m \cdot s$
μ_{eff}	effective gas viscosity ($\mu + \mu_{turb}$), $kg/m \cdot s$
μ_{turb}	turbulent gas viscosity, $kg/m \cdot s$
ν	kinematic viscosity, m^2/s
ξ	blending factor
π_i	Lagrangian multiplier for element i
π_K	temperature sensitivity of pressure
ρ	density of combustion gas, kg/m^3
ρ_P	density of solid propellant
$\sigma_k, \sigma_\varepsilon, C_{\varepsilon 1}, C_{\varepsilon 2}$	constants used in solving turbulent kinetic and dissipation rate equations ($k - \varepsilon$ model)
σ_P	temperature sensitivity of the burning rate at constant pressure
τ	shear stress
τ_{ij}	viscous stress tensor
τ_w	wall viscous shear stress
ϑ	cell volume
ϕ	a variable quantity (e.g., enthalpy)
ψ	dimensionless blocking effect constant
ω	specific turbulence dissipation rate
$\dot{\omega}_k$	rate change of k th species due to chemical reactions, $kg/m^3 \cdot s$

Γ diffusion coefficient

Subscripts and Superscripts

b	burning propellant
CD	central differencing
e	outside the boundary layer or cell face value
gap	annular gap between rod and tube propellant boundaries
i	single element or i th direction, function, species or variable
inf	infinite (referring to infinite area combustor)
j	single chemical species or j th direction
k	k th direction
mix	system property for multiple species
nb	values from neighboring cells
P	cell centre value
s,w	at the surface or wall
s	isentropic
seg	segment property
t	property measured at throat
$turb$	turbulent
T	isothermal
UP	upwind differencing
x, y, z	direction
1	chamber property
2	nozzle exit plane property
3	ambient property (referring to outside of rocket)
$^{\circ}$	symbol for standard state
$*$	guessed value
$'$	correction value
$+$	dimensionless
$-$	overbar signifies time averaged or normalized property
$()'$	fluctuating quantity in brackets

Chapter 1 *Introduction*

1.1 Introduction

While the Chinese are credited with developing and using solid propellant in rockets over 800 years ago, most of the key developments in rocket propulsion have occurred over the last century. This included the development of the liquid propellant rocket motor, significant refinement of the solid rocket motor, development of the hybrid propellant rocket motor which utilize both liquid and solid propellants, and development of ducted rocket motors which use a solid propellant coupled with air as the oxidizer [Sutton and Biblarz (2001)]. Each of these rocket motor types are based on the high pressure combustion reaction of a number of propellant chemicals, principally a fuel and an oxidizer. The mostly gaseous products of combustion are heated to very high temperatures (typically 2800 to 4400 K), expanded within a nozzle and accelerated to very high velocities [Sutton and Biblarz (2001)].

In accordance with Newton's second law, a propulsive force is generated in reaction to the momentum of the mass discharged from the rocket.

Each of the rocket motor types offers specific advantages which tailors their use to different types of tasks. Liquid propellant motors require a complex propellant feed mechanism but permit repetitive operation during the same flight as well as the use of more energetic fuels and oxidizers. These traits make them very effective for boosting large payloads outside of the Earth's atmosphere. Solid rocket motors are relatively simple to construct, operate, and store very well. While active throttling of the thrust is very difficult, if not impossible to achieve, the thrust of a solid propellant rocket can be tailored through various grain designs that provide boost, sustain and coast phases. For these reasons, solid rocket motors are most popular for smaller applications and generally used within the Earth's atmosphere. Hybrid propellant and ducted rocket motors have not been utilized to the extent of the other two types, but offer significant benefits such as throttability, simplicity (compared to liquid propellant motors) and increased range and specific impulse (compared to solid propellant motors). Specific impulse, I_{sp} , is an important figure of merit for rocket motors and is the instantaneous ratio of thrust, F , produced by the motor to the unit flow weight of the propellant carried by the rocket [Sutton and Biblarz (2001)]:

$$I_{sp} = \frac{I_{tot}}{m_{prop} g_o} = \frac{F}{\dot{m}_{prop} g_o} \quad (1.1)$$

The motor under consideration within this thesis is a solid rocket which is required to operate close to the ground and accelerate to hypersonic velocities (Mach 5 or greater) very quickly. It is anticipated that combustion will be complete in less than one second following ignition of the motor.

1.2 Functioning of a Solid Rocket Motor

Solid propellant motors are typically based on a hollow, generally cylindrical grain, although some motors are designed in an end burner configuration. The grain is composed of all of the chemical elements necessary for combustion. Once ignited, the grain burns on all exposed internal surfaces and regresses until the outer casing is reached.

Figure 1.1: A simplified cross section of a typical tactical solid propellant rocket motor [Sutton and Biblarz (2001)].

Figure 1.1 shows the typical layout of the propellant grain and nozzle. The grain is shaped like a thick walled cylinder. Combustion occurs on the inner longitudinal face of the cylindrical grain, with the centre of the grain (the port) also providing a path for the hot, high pressure products of solid propellant

combustion to travel to a converging, diverging bell-shaped nozzle. This flow of gases within the chamber, parallel to its longitudinal axis, is known as crossflow. During the expansion in the nozzle, the static pressure of the exhaust gases drops substantially, the temperature of the gases drops to a lesser extent and, as can be calculated using the continuity equations, the velocity of the gases increases from a Mach Number of roughly 0.5 at the nozzle inlet to a nozzle exit value with a Mach Number of roughly 2 to 3.5.

Gas generation is governed by the burning rate of the solid propellant as well as its surface area. The combustion of solid propellant is characterized by the fact that the regression of the grain is normal to the burning surface. This rate of regression is referred to as the burning rate and is typically measured in mm/s. In addition to the propellant composition, the burning rate is a function of the following factors according to Sutton and Biblarz (2001):

- Combustion chamber pressure
- Initial temperature of the solid propellant before burning commences
- Combustion gas temperature
- Velocity of the gas flow parallel to the burning surface
- Motor motion (which causes grain stresses due to acceleration and spin)

1.2.1 GAP-ADN Propellant

Solid rocket propellants are divided into two main categories: homogeneous, where the fuel and oxidant are contained within the same propellant molecule; and heterogeneous, where the propellant consists of a mixture of crystals of oxidant mixed in an organic fuel binder. Homogeneous propellants are also referred to as double-base (DB) propellants due to the main ingredients being nitrocellulose and nitroglycerine. Heterogeneous propellants are commonly referred to as composite propellants. A third type of propellant, which is a mixture of the two types, is known as a composite modified double base propellant (CMDB) [Sutton and Biblarz (2001)].

The propellant being used in the motor under test is a composite propellant and includes ammonium dinitramine (ADN) as an oxidizer and glycidal azide polymer (GAP) as a fuel. The use of these ingredients provides two key advantages over the typical propellant used in solid rocket motors, which use ammonium perchlorate (AP) as an oxidizer and hydroxyl terminated polybutadiene (HTPB) as a binder, which are:

- a. Use of ADN as an oxidizer results in a smokeless and relatively nontoxic exhaust [Sutton and Biblarz (2001)], while the exhaust from propellants containing AP include HCl, which is toxic and causes secondary smoke; and

b. GAP has higher enthalpy of formation than does HTPB as well as a higher burning rate [Komai et al. (1999)]. Both of these characteristics are key to meeting the high acceleration requirements of the motor under consideration.

1.2.2 Combustion Chamber Pressure

The largest physical effect on the propellant burning rate is due to chamber pressure. For many solid propellants, over a limited range of pressure, the burning rate can be described as:

$$r_o = a_T P^n \quad (1.2)$$

where r_o is the linear burning rate, a_T is an empirical constant influenced by the ambient grain temperature, P is the chamber absolute pressure and n is a pressure exponent of the burning rate. This pressure exponent is independent of the ambient grain temperature according to Kruse (1992) and must be less than 1 for a solid rocket motor to operate in a stable fashion (if not, the pressure will continue to increase until catastrophic failure occurs). Note that a_T and n are properties of a given propellant composition. When plotted on a log-log graph, the logarithm of the burning rate varies linearly with the logarithm of the pressure, with n as the slope. Typical burning rate versus pressure curves are shown in Figure 1.2.

Figure 1.2: Burning rate characteristics of a DB propellant as well as three CMDB propellants with varying percentages of Ammonium Perchlorate. The relationship between chamber pressure and burning rate is highlighted [Sutton and Biblarz (2001)].

1.2.3 Initial Propellant Temperature

Since temperature and chemical reaction rates are linked, it is not unexpected that the initial ambient temperature of a solid propellant grain (prior to combustion) has an effect upon the burning rate. As can be seen in Figure 1.3, a variation of 20 to 35% in chamber pressure and burn time is typical for composite propellants over the operating range of an air launched missile motor (219 K to 344 K). Note that the total impulse of the motor remains roughly constant throughout this temperature range.

Figure 1.3: The effect of propellant temperature upon burning time and chamber pressure for a particular motor is shown [Sutton and Biblarz (2001)].

From Kruse (1992), the simple burning law is modified to the following form to account for the effect of temperature on burning rate:

$$r_o = e^{(\sigma_p T + \beta)} P^n \quad (1.3)$$

where σ_p is the temperature sensitivity of burning rate at constant pressure and is expressed as

$$\sigma_p = \left(\frac{\partial \ln r_o}{\partial T} \right)_P = \frac{1}{r_o} \left(\frac{\partial r_o}{\partial T} \right)_P \quad (1.4)$$

T is the ambient grain temperature and β is a constant of integration.

Kruse (1992) notes other equations which describe the key relationships between burning, pressure and temperature. Since the chamber pressure will vary due to a change in the burning rate, a second temperature coefficient

known as the temperature sensitivity of pressure, π_K , must also be utilized and is calculated as follows:

$$\pi_K = \left(\frac{\partial \ln P}{\partial T} \right)_{K_n} = \frac{1}{P} \left(\frac{\partial P}{\partial T} \right)_{K_n} \quad (1.5)$$

where K_n is the ratio of the burning surface area to the nozzle throat area. The need for two temperature sensitivity coefficients is due to the methods by which the coefficients are calculated. σ_p is calculated from strand burner test data while π_K is determined during the test firing of actual motors (either small or full scale).

For a given propellant, the equilibrium chamber pressure P_1 can then be calculated using the classic vented vessel equation, $P = r_o c^* \rho_p K_n$, as well as the simple burning law modified to account for the grain temperature (Equation 1.3) to become:

$$P_1 = \left[e^{(\sigma_p T + \beta)} c^* \rho_p K_n \right]^{\frac{1}{1-n}} \quad (1.6)$$

where ρ_p is the solid propellant density and c^* is the characteristic velocity. The expression for c^* was derived from theory relating isentropic flow through a nozzle:

$$c^* = \frac{\sqrt{\gamma R T_1}}{\gamma \sqrt{\left[\frac{2}{\gamma + 1} \right]^{\frac{\gamma + 1}{\gamma - 1}}}} \quad (1.7)$$

Here R is the specific gas constant and T_1 is the chamber gas temperature.

Using Equation 1.6 and the temperature sensitivity of pressure, it is then possible to calculate the motor pressure P_i at any ambient grain temperature T_i using the following:

$$P_i = P_o e^{[\pi_K (T_i - T_o)]} \quad (1.8)$$

1.2.4 Combustion Gas Temperature

As can be seen when considering the expressions for the equilibrium chamber pressure (Equation 1.6) and characteristic velocity (Equation 1.7) together, changes in the chamber gas temperature will result in a corresponding change to the chamber pressure.

1.2.5 Other Physical Factors Affecting Burning Rate

As for the remaining physical factors which affect the propellant burning rate, the effect of the gas velocity parallel to the burning surface gas is discussed in detail in Chapter 2. The motion of motor (acceleration and spin) may increase the burning rate in all solid propellants, and in particular, those solid propellants which are aluminized. Given that the propellant under

consideration does not incorporate aluminum, this factor will not be addressed further.

1.3 Grain Design Considerations

While the formulation of the propellant certainly has a significant impact on the propellant performance characteristics, the geometry of the grain also plays a key role. This is because the solid propellant burns on all exposed surfaces once ignited, perpendicularly to those surfaces. Grain geometry can be tailored to obtain either a constant burning area throughout the duration of the motor functioning (neutral burn) or a burning surface area which increases or decreases with time and provides a progressive (increasing thrust) or regressive (decreasing thrust) burning rate, respectively. Grain geometry design is a function of the operating requirements for the rocket motor and obtaining the desired thrust-time curve within the space envelope available for the propellant grain. Often, complicated shapes are used for the grain geometry, such as those shown in Figure 1.4.

Figure 1.4: Simplified diagrams of some grain configurations [Sutton and Biblarz (2001)].

1.3.1 Rod-and-Tube Grain

The requirements for the motor of interest are:

- High energy, high efficiency, minimum smoke
- Stable high burning rate, high pressure combustion
- High loading density

Since one of the propellant grain requirements for the motor of interest is to obtain a neutral but high burning rate, the same area of exposed propellant must exist throughout the burn. By using the rod-and-tube grain geometry, the propellant surface area is kept essentially constant throughout the burn since the burning area on the surface of the rod decreases at a similar rate as inner surface area of the tube increases. A simplified example of this grain type is shown in Figure 1.5.

Figure 1.5: Simplified diagram of a rod-and-tube configuration solid rocket motor.

The rod-and-tube configuration necessitates the use of a plug nozzle, rather than the typical deLaval or bell nozzle, to pass the products of combustion. As well, the motor under consideration will incorporate a very low port-to-throat area ratio, which in turn results in very high velocities of the propellant gas traveling across burning propellant surfaces, particularly near the nozzle end of the rocket.

1.3.2 Plug Nozzle

The plug nozzle is a type of aerospike nozzle that has an annular, doughnut shaped chamber, with an annular nozzle slot. For this application, the nozzle directs the combustion gas flow towards the centre of the plug as well as axially so that a single plume is formed. A significant advantage of the aerospike configuration is that a change in ambient pressure during ascending flight is automatically compensated for by a change in the gas pressure distribution on the plug. Inherent in a bell nozzle is the inability to easily compensate for changes in ambient pressure. As a result, the combustion gases will either be overexpanded at low altitudes, which results

in separation of the combustion gas flow from the nozzle wall, or underexpanded at high altitude which results in additional expansion of the combustion gases upon exit from the nozzle which reduces the effective thrust. For the purposes of the motor under consideration, this is not a significant issue as the anticipated flight path does not include a significant change in altitude following launch from the ground so the nozzle can be optimized for operation at low altitude. As Figure 1.6 shows, a bell nozzle which is operating at or close to its design point ambient pressure has the same thrust coefficient as does a plug nozzle. The thrust coefficient is a dimensionless figure of merit which represents the amplification of thrust due to a supersonic nozzle compared to the thrust available if the chamber pressure acted over the throat area only. The key advantage of such a nozzle for the purpose of the motor under consideration is the reduced length required provided that the plug is truncated.

Figure 1.6: Typical relation of the thrust coefficient (C_F) to the Pressure Ratio for a 20% Plug Nozzle, Bell Nozzle and Ideal Nozzle [Ito et al. (1999)].

1.4 Solid Propellant Rocket Motor Modeling

1.4.1 Characteristics of Solid Rocket Motor Combustion

In their simplest form, there are three physical processes that occur during the combustion of a solid rocket motor. The first two are fluid flow and combustion, which were discussed in Section 1.1. The third process is the structural response of the propellant and case. A solid rocket motor is structurally a thin-shelled cylinder which is almost completely filled with propellant, a viscoelastic material. The failure modes of this configuration are numerous, but all have a common effect – increased propellant burning area which results in internal pressure beyond the limits of the case. These three physical processes are coupled during the entire time solid rocket motor combustion is in progress. Of course, there are many other complex phenomena which occur during operation of the motor which have a significant effect upon its performance. These phenomena include:

- a. Erosive burning - the increase in propellant burning rate caused by the high velocity flow of combustion gases over the burning propellant surface. This phenomenon is most prevalent near the nozzle where the mass flow rate is the highest; and
- b. Ignition transient – the ignition of a solid rocket motor involves a comparatively small pyrotechnic device igniting and causing an efflux of hot combustion product gases onto some of the propellant surfaces

which quickly spreads to the entire propellant surface. This is a particularly violent phenomena which involves the chamber pressure going from essentially ambient to tens of MPa in a few milliseconds.

These complex phenomena have been the subjects of considerable investigation and have been summarized in detail [Kuo and Summerfield (1984), DeLuca et al. (1992)]. However, simulation of the three basic physical processes, which includes all of the complex phenomena that occur, is a task that, until recently, was only within the scope of computer applications which relied upon empirical data to produce a realistic simulation such as the design tools discussed in Section 1.4.2.

1.4.2 Design Tools

Although there is no single, well defined methodology used in designing solid rocket motors, it is typical that the propellant type and grain configuration are selected early in the design process. This is necessary in an effort to ensure that the three key requirements met by the propellant are satisfied. These requirements are performance (I_{sp}), a burning rate which matches the required thrust-time curve, and strength (appropriate structural response of the propellant) [Sutton and Biblarz (2001)]. These requirements have typically been solved successfully by assuming a linear burn rate (Equation 1.2), quasi-one-dimensional flow through an ideal rocket and applying empirical corrections to the ideal values (Note: a detailed discussion on the

assumptions made in defining an ideal rocket is found in Section 4.2). One of the more successful computer models employing this approach has been the commercially available Solid Performance Program (SPP) from Software and Engineering Associates, Inc., Carson City, Nevada, USA.

1.4.2.1 SPP

SPP is a standard reference program widely used within industry to predict the delivered performance of solid propellant rocket motors. The application works by separating the analysis into two parts: nozzle performance and motor performance. The maximum (ideal) performance is calculated and then degraded by the application of individual, empirically determined, losses which are known to occur in real motors. This method assumes that most losses can be treated independently from other losses which occur in the motor [Coats et al. (1987)]. These losses include those which result from two-phase flow, finite rate kinetics, boundary layer, combustion efficiency, submergence (the portion of the grain which extends aft of the nozzle) and nozzle erosion. While accurate solutions can and have been obtained using SPP in many cases, this may not be the case if the empirical factors which are used within SPP are not representative of the design being considered. As will be seen in this thesis, these coefficients are dependent upon the type of propellant used and can have a significant effect upon the performance of the rocket motor. In addition, it is not possible to accurately characterize the flow of the products of combustion radially across the port. Knowledge of the

flow in this region is key to providing accurate boundary conditions for nozzle or thrust vector control (TVC) design and modeling. Thus, the use of SPP as a design tool is limited in a number of areas for which a detailed, time-dependent knowledge of the flow characteristics at many locations is required. However, by considering the fluid flow in accordance with proven mathematical models of fundamental physical processes (i.e., Navier-Stokes equations, etc.) it is possible to obtain a solution of the desired detail. This capability can now be achieved through the use of Computational Fluid Dynamics (CFD) and is being pursued in a number of applications.

1.4.2.2 Introduction to CFD

Obtaining solutions to the governing equations of fluid mechanics and heat transfer, which are expressed as partial differential equations (PDEs), is an extremely difficult challenge. It has been simplified somewhat in recent years due to the introduction of modern computing power. Using numerical analysis techniques, it is now possible to solve these coupled, non-linear PDEs in varying geometry and flow conditions. This field is known as CFD and is being used increasingly because of the physical and monetary limitations imposed by experimental solutions.

Due to the introduction of a grid regression capability within CFD applications, it is now possible to accurately describe the boundary conditions and surface shape at the flow inlet (grain surface) on a time-dependent basis. This has

allowed researchers to carry out a complete simulation of a solid rocket motor functioning. One example of this work was carried out recently at the Center for Simulation of Advanced Rockets (CSAR) which is based at the University of Illinois at Urbana-Champaign.

1.4.2.3 Multi-Physics Simulation of Solid Rocket Combustion

CSAR has achieved considerable success in modeling all aspects of solid rocket motor operation. The overall goal of the Center's efforts is to provide detailed numerical simulations of the Space Shuttle Solid Rocket Booster. Using three component codes known as ROCFLO, ROCSOLID and ROCFACE, all of the physical processes attributed to solid rocket motor operation which were described above, have been simulated [Alavilli et al. (2000)].

ROCFLO is a CFD code which utilizes a three-dimensional, structured, finite volume, cell-centered approach. The Navier-Stokes equations are solved on a dynamic grid that features adaptive remeshing. A symmetric Total Variation Diminishing (TVD) scheme is utilized [Yee (1987)] as the spatial discretization scheme, while an explicit multi-stage Runge-Kutta method is used to provide the solutions of the finite difference equations at the next time step. The grid speed is based on structural deformation and combustion induced regression. ROCSOLID is a structural finite element analysis code that utilizes an unstructured mesh. ROCFACE is the interface code that manages the mesh

association and data transfer tasks between other two programs. While the scale of this solution is well beyond the scope of this thesis (ROCFLO subdivides the Space Shuttle's Solid Rocket Booster into 36 million nodes, and the application runs on multiple processors) the approach followed by the Center for Simulation of Advanced Rockets does provide an excellent example as to how a CFD/multi-physics can be used to model solid rocket motor combustion.

1.4.2.4 Use of a Commercial CFD Application

Because of the availability of a grid regression capability within commercial CFD applications, it is now possible to tailor a CFD solution to the challenge of modeling solid rocket combustion without the use of an application such as ROCFLO and its requirement for extensive computer resources. At the commencement of work on this thesis, a commercially available CFD application had not been selected for use in the modeling of solid rocket combustion. Ideally, this application would be able to couple combustion of the solid rocket propellant, the fluid flow of the exhaust gases through the rocket, and the structural reaction of the propellant and casing, i.e., model the complete physics of solid rocket operation. A study recommending the choice of a CFD application based on those ideal criteria was carried out at the commencement of work on the thesis. This study, as presented to DRDC Valcartier, is included at Appendix A to this thesis. The commercial application chosen was CFD-ACE+ as produced by the CFD Research

Corporation based in Huntsville, Alabama, USA. A description of this application, along with a detailed discussion of the various modules incorporated in the solid rocket modeling is found in Chapter 3 of this thesis. The computer used to support the application was a standalone DELL Dual Pentium III Xeon with a 2.2 GHz CPU speed and 1 GB of RAM. The Windows 2000 Operating System was installed on this PC.

1.5 Scope of Thesis

While a great deal of literature is available concerning the modeling of solid rocket motor combustion and flow characteristics within the chamber, there is no literature available concerning the modeling of a solid rocket motor utilizing a rod-and-tube grain. In addition, there is minimal information available concerning this modeling using CFD. Until recently, many modeling efforts did not take into account the transient nature of rocket combustion with boundaries that vary with time and instead carried out the modeling using steady-state analyses. Efforts such as that of CSAR are now directed towards time-dependent analyses, but even the complex ROCFLO code does not take into account erosive burning phenomena. For the motor under consideration, it is anticipated that very high crossflow velocities of the products of combustion will be encountered in the port, thus it is essential that the modeling effort incorporate erosive burning. Thus the purpose of this thesis is expressed in the following statement:

To develop a methodology, based on a commercial CFD application, to predict the flow characteristics within the chamber of a solid rocket motor that addresses all significant physical and chemical phenomena including erosive burning and grain regression. This methodology is to be demonstrated on a two-dimensional simulation of the internal flowfield of a rod-and-tube grain solid rocket motor.

The development of this model can be achieved through the following steps which also provide for a suitable structure as follows:

- a. Chapter 2 describes the erosive burning phenomena in detail, outlining the development of theories related to this phenomena since it was first modeled in 1957. A choice of erosive burning model which will be incorporated into the CFD model is also selected.
- b. Chapter 3 describes the CFD application in detail including a description as to how the governing equations are discretized and solved within the application.
- c. Chapter 4 discusses the development of the boundary and initial conditions used in the CFD model, less the grid deformation feature.

- d. Chapter 5 describes grain regression model used as well as detailing the computational domain on which the model was constructed.
- e. Chapter 6 describes the results of the modeling of the rod-and-tube configuration motor as well as a discussion of the key findings. These results include the results of an analytical model as well as those of the steady-state and non-steady-state CFD model.
- f. Chapter 7 provides conclusions and recommendations for future work.

Chapter 2 *Erosive Burning*

2.1 Introduction

An increase in the propellant burning rate can be caused by the high velocity crossflow of combustion gases over the burning propellant surface. This characteristic of many solid propellant rockets, known as erosive burning, will be analyzed in depth within this thesis; however, an introduction to the phenomena is provided below. Without the presence of the crossflow, the burning rate is equal to the linear burning rate law [Kruse (1992)].

Erosive burning can have a significant effect upon the performance of a solid propellant rocket motor and must be considered in the design of the motor, either by taking measures to avoid the phenomenon or by controlling its effects. Erosive burning most often presents itself either in the port area or in the grain perforations near the nozzle end of the rocket. According to Sutton and Biblarz (2001), rockets with a small port-to-nozzle-throat ratio (less than 4) are very likely to undergo erosive burning. The motor under consideration

incorporates a small port to nozzle ratio (1.72 at ignition), thus consideration of erosive burning in modeling the combustion of the rod-and-tube grain solid propellant motor appears essential.

Very simply, erosive burning occurs when the high velocity of the combustion gases, moving parallel to the burning surface, develop a turbulent boundary layer that penetrates the strongly blowing surface of the burning propellant. The diffusion process which normally controls the burning rate is therefore enhanced by the turbulent mixing. As a result, the burning rate increases, with consequent increases in the mass flow, chamber pressure and thrust during the initial stages of the propellant being burned. Often, erosive burning will increase the flow passage with a corresponding increase in burning area which will then reduce the flow velocity bringing the burning rate back to normal levels; however, by this time sufficient propellant has been burned to cause a premature reduction of thrust at the end of burning. This can cause an early burnout of the grain near the nozzle end and can expose the insulation and aft closure to hot combustion gases for a comparatively large period of time.

Erosive burning does not occur as a function of all combustion gas flow velocities within the rocket port, rather it occurs in only one of three 'regimes' of flow parallel to the burning surface. Initially, at lower mean flow velocities, the burning rate is not affected. At some critical mean flow value, known as

the threshold value, the burning rate begins to increase roughly linearly with the increases in mass flow velocity. Finally, after an upper mean flow level is reached, the burning rate increase is greatly reduced and may even cease as further mean flow velocity increases occur.

For propellants which use Ammonium Perchlorate (AP) as the oxidizer (also known as composite), the following guidelines are applicable according to Kruse (1992):

- a. Low burning rate propellants have a greater tendency to exhibit erosive burning than do high burning rate propellants;
- b. Propellants with large AP particle sizes (e.g., 50 to 200 μm according to Sutton and Biblarz (2001)) have a greater tendency to exhibit erosive burning than do propellants which have small AP particle sizes; and
- c. Propellants operating at low pressures have a greater tendency to exhibit erosive burning than do propellants operating at high pressures.

2.2 Erosive Burning

Due to the requirement to account for erosive burning in designing solid propellant rocket motors, a great deal of research has been carried out over the past 40 years with a goal to accurately model the phenomenon. The models derived involve varying approaches which attempt to balance the conflicting requirements of relative simplicity against accuracy in the largest number of cases. The more complex models have been formulated in recent years due to the availability of increased computing power and better understanding of the interaction of the boundary layer interaction with the propellant flame structure. In a very highly regarded article published in 1984, Kuo was able to classify the various erosive burning models into four basic categories based on how the models accounted for the interaction of the transport properties between the flow of the combustion gas through the port and the flame structure on the burning surface of the propellant. These categories were:

- a. Phenomenological heat transfer;
- b. Integral boundary layer analysis;
- c. Modification of the propellant combustion mechanism; and
- d. Chemically reacting boundary layer analysis.

They are detailed in the following paragraphs.

2.2.1 Phenomenological Heat Transfer

In 1957, a phenomenological heat transfer model which incorporated erosive burning was proposed [Lenoir and Robillard (1957a)]. In their findings, they believed that two mechanisms of gas to solid heat transfer were involved in solid propellant rocket motor combustion:

- a. from the primary burning zone to the solid propellant walls, which is independent of core gas velocity and is a function of pressure only.
- b. from the core of hot combustion gases to the propellant walls, which depends on gas velocity. From this method, the relation

$$r_b = r_o + r_e \quad (2.1)$$

was developed where r_b is the total burning rate normal to the propellant surface, r_o is the pressure dependent burning component and r_e is the erosive burning component.

The main assumption made by Lenoir and Robillard (1957b) in developing their erosive burning formula was that the heat transfer model was not that of a gas flowing down a simple tube, instead, the heat transfer was taking place in a tube under conditions of vigorous transpiration (gas blowing through a porous surface). The result of this transpiration is to reduce the heat transfer

rate and the drag. For conditions of strong transpiration, the propellant gases traveling down the port approach the conditions of being insulated from the propellant walls near the head end of the rocket in particular. Closer to the nozzle end of the rocket, the degree of insulation becomes continuously poorer and drag losses occur against the combustion gas flow. The pressure decrease in combustion gas flow is due principally to the momentum losses caused in accelerating the transpiring gas, and secondarily due to the friction losses.

In developing their model, Lenoir and Robillard related the heat transfer coefficient h under conditions of transpiration to the heat transfer coefficient with zero transpiration, h_o in exponential form as follows:

$$h = h_o e^{\frac{-\beta Q}{G}} \quad (2.2)$$

where Q is the mass flow rate of the transpiring gas and is equal to $r_b \rho$. β is a dimensionless exponential constant, G is the mass velocity of combustion gas ($\text{lb}_m/\text{ft}^2 \cdot \text{sec}$) and ρ is the density of combustion gas. Because many of the semi-empirical formulas which were used in the development of erosive burning formulas were based on measurements and calculations using Imperial units, the use of this system of units is utilized in some cases within this thesis to better follow the research of the referenced authors. The Imperial units used are included within the main body of the thesis.

The erosive burning rate increment, r_e , was then assumed to be proportional to the heat transfer coefficient h . The Chilton-Colburn equation for turbulent flow over a flat plate was used to define h_o [Chapman (1984)]:

$$h_o = 0.0288 G C_p \text{Re}_x^{-0.2} \text{Pr}^{-0.667} \quad (2.3)$$

Recalling that

$$\text{Re}_x = \frac{LV\rho}{\mu}$$

and

$$G = V\rho$$

Equation 2.4 results from a substitution of (2.2) and (2.3) into (2.1) since

$$r_e = k h:$$

$$r = a_T P^n + \left[0.0288 C_p \mu^{0.2} \text{Pr}^{-0.667} \right] \cdot k \left(\frac{G^{0.8}}{L^{0.2} e^{\frac{\beta \cdot r_b \rho_p}{G}}} \right) \quad (2.4)$$

where k is a proportionality factor that relates the erosive burning increment (r_e) to the heat transfer coefficient from the core combustion gases to the propellant wall:

$$k = \frac{1}{\rho_p C_{ps}} \left(\frac{T_f - T_s}{T_s - T_i} \right) \quad (2.5)$$

L is the length from the head of the grain (ft) and ρ_p is the propellant density (lb_m/ft^3). By rearranging Equations 2.4 and 2.5, the Lenoir-Robillard erosive burning law (Equation 2.6) is obtained:

$$r_b = a_T P^n + \frac{\alpha G^{0.8}}{L^{0.2} e^{(\beta \cdot r_b \cdot \rho_p / G)}} \quad (2.6)$$

where α is the erosive constant ($\text{ft}^{2.8}/\text{lb}_m^{0.8} \text{sec}^{0.2}$). This expression for α was derived by considering the energy balance between the heat transfer from the flame (at temperature T_f) to the propellant surface and the heat required to raise the propellant temperature from its initial temperature (T_i) to its final surface temperature (T_s) as follows:

$$\alpha = \frac{0.0288 C_p \mu^{0.2} \text{Pr}^{-0.667} \left(\frac{T_f - T_s}{T_s - T_i} \right)}{\rho_p C_{ps}} \quad (2.7)$$

where C_p is the heat capacity of the combustion gas, C_{ps} is the heat capacity of the solid propellant and μ is the viscosity of the combustion gases. α is calculated using values for the properties of the solid propellant and combustion products at 298K but is relatively insensitive to propellant composition.

Determining the value of β is much more difficult and is typically done using the results of interrupted firings of test motors. By measuring the amount of propellant burned as a function of time and grain position, the value was

determined by trial and error against measured values of head-end and throat static pressure and then validated by matching the computed total mass flow rate out of the rocket to the measured mass flow rate. Studies by Lenoir and Robillard as well as research on other composite propellants showed that the value of β was dependent on the propellant identity [Lawrence et al. (1968)]. Once accurately determined, the values of α and β are independent of grain length, initial port-to-throat ratio and pressure level.

This model seems to be relatively well accepted and is used widely, perhaps due to its simplicity. Many other researchers have further developed the Lenoir-Robillard erosive burning law including Zucrow, whose experimental work identified a threshold velocity [Zucrow et al. (1964)], which was pressure dependent. According to his work, the total burning rate was greater than the linear ($u = 0$) burning rate only above this threshold velocity (u_{iv}), which was defined in the following expression:

$$u_{iv} = \frac{\beta_T \rho_p a_T P^n}{St_o \rho} \quad (2.8)$$

where β_T is the transpiration parameter (typically in the region of 0.2), St_o is the Stanton Number for non-blowing conditions (a non-dimensional representation of heat transfer which equals ratio of $\frac{h}{\rho u C_p}$), and ρ is the gas density.

Although Zucrow found that erosive burning did exist at all gas velocities, there existed two erosive burning regimes which were separated by u_{rv} . At combustion gas velocities (crossflow) less than the threshold velocity, the effect of mass transfer of combustion gases from the burning surface to the core of gases flowing down the port actually reduced the rate of heat conduction to the burning surface from these same hot combustion gases. This is the insulating effect discussed by Lenoir and Robillard. In this regime, the propellant burning rate was actually found to be less than the linear burning rate. This phenomenon is referred to as negative erosion. At u_{rv} , the rate of heat transfer by convection has increased to the point where it balances the reduction in heat transfer due to the mass addition. At this gas velocity, the actual propellant burning rate equals the linear burning rate. Above u_{rv} the convective heat transfer dominates the heat transfer process and the propellant burning rate increases above r_o .

Independent experimental investigation into propellant erosion was carried out shortly following publication of the Lenoir-Robillard erosive burning theory [Marklund and Lake (1960)]. Instead of employing interrupted burning tests as was done by Lenoir and Robillard, Marklund and Lake utilized an apparatus as shown at Figure 2.1 where the effect of erosive burning was measured by placing propellant specimens in the gas stream of a solid propellant gas-producing charge. Specimen configurations included propellant strips and tablets as well as nozzle inserts to investigate propellant

erosion at sonic velocities. Their experimental data had good agreement with the Lenoir-Robillard theory and also established a baseline set of experimental data that has been used by many other researchers investigating erosive burning. They also noted the tendency for the total burning rate to be less than the linear burning rate, which was later explained by Zucrow et al. (1964).

Figure 2.1: Tablet method used by Marklund and Lake. Propellant tablets are cemented on quartz window plugs and placed in gas stream with surfaces parallel to the gas flow. Burning times of tablets determined when light from the propellant gas passes through the quartz windows to a detector [Marklund and Lake (1960)].

Disagreement with the Lenoir-Robillard erosive burning law, using the Marklund and Lake experimental data, was later published [King (1976)]. King argued that Lenoir and Robillard ignored the relationship between the combustion process and the crossflow velocity, instead assuming that erosive

burning was uniquely a function of heat transfer. According to King, that instead of an increased burning rate being caused by erosive burning, at constant pressure, the accepted physical model shows that the propellant flame is being pushed farther from the propellant surface, which decreases the heat feedback rate, hence the propellant burning rate as the crossflow velocity increases. King proposed a different “base” burning rate based on granular diffusion flame theory where the linear burning rate term of Lenoir-Robillard was replaced by the following expression:

$$r_o = \frac{(a_r P^n)^2}{r_b} \quad (2.9)$$

Here, the “base” burning rate is proportional to the combustion pressure and the overall burning rate, rather than being dependent solely upon the combustion pressure. This approach was criticized because of its assumption of additive heat fluxes by conduction from the flame zone burning surface as well as from the combustion (crossflow) gases. Of note is that the results of both Lenoir-Robillard and King produced similar results using the erosive burning data of Marklund and Lake as can be seen in Figure 2.2.

The values in Figure 2.2 approach the burning rate and pressures anticipated for the motor under consideration. They may be of use in determining the values for erosive burning coefficients should a phenomenological heat transfer model be used in this thesis to model erosive burning.

Figure 2.2: Comparison of erosive burning rate data of Marklund and Lake with the predictions of Lenoir and Robillard as well as the Lenoir and Robillard expression as modified by King [King (1975)].

2.2.2 Integral Boundary Layer Analysis

A different interpretation for the erosive burning mechanism was proposed by Lengellé in 1975. He theorized that erosive burning was a function of the turbulence of the combustion gas flow through the port. This model was based on increased heat transfer to the propellant surface due to enhancement of transport properties (diffusion and heat conduction) in the boundary layer caused by turbulent flow. This increased heat flux resulted in an increased burning rate.

Using granular diffusion flame theory, the propellant burning rate is related to the flame thickness as follows:

$$\dot{m}_b Q_T \approx \lambda (T_f - T_s) L_f \quad (2.10)$$

where $\dot{m}_b$ is the propellant burning mass flow rate, Q is the energy required to raise the propellant from its initial temperature to the surface temperature next to the flame zone, λ is the thermal conductivity and L_f is the flame thickness. The flame thickness is a function of the ammonium perchlorate (AP) particle size as shown in the relation:

$$L_f \approx \nu \frac{d^2}{D} \approx \frac{\dot{m}_b M_p^{2/3}}{\rho^{2/3} (\rho D)} \quad (2.11)$$

where ν is the velocity normal to the propellant surface, d is the mean particle diameter of the AP, D is the diffusion coefficient of the gaseous fuel pockets mixing with the oxygen containing gases resulting from the combustion of the AP particles, M_p is the characteristic mass of the gaseous fuel pockets (a function of the AP content and its mean particle size), and ρ is the density of the gaseous fuel pockets.

In the normal, non-erosive combustion regime, the transport coefficients (λ and D) are laminar; however, under erosive conditions, these coefficients are modified by turbulent components depending on the nature of the flow. Based on the assumption that the turbulent Lewis and Schmidt numbers (i.e., the ratios of mass diffusion to thermal diffusion, and viscous diffusion to the mass diffusion) are close to unity, then

$$\lambda \approx C_p \rho D \quad (2.12)$$

and

$$\rho D \approx \mu \left(1 + \frac{\rho \varepsilon}{\mu} \right) \quad (2.13)$$

then

$$\dot{m}_b \approx \left[\frac{C_p (T_f - T_s)}{Q} \right]^{1/2} \left(\frac{\rho}{M_p} \right)^{1/3} (\mu + \rho \varepsilon) \quad (2.14)$$

where μ is the combustion gas viscosity and ε is the turbulent momentum diffusivity. This approach assumes that the flame thickness, which has a height of approximately $10\mu\text{m}$ and is within the laminar sublayer of the turbulent boundary layer, is not affected by the presence of a turbulent flow, which disagrees with the model of King, but is supported by Granular Diffusion Flame theory. By combining Equation 2.14 with a model also developed by Lengellé (using Couette flow and Prandtl's mixing length approximations) for a turbulent boundary layer on a flat plate with injection, the expression for burning mass flow rate under erosive conditions becomes:

$$\dot{m}_b \approx C_p \frac{T_f - T_s}{Q} \left[\frac{\mu}{L_f} + 0.083 \frac{\rho_e u_e}{(R_{ex})^{0.1}} \left(\frac{L_f}{\delta} \right)^n \psi \right] \quad (2.15)$$

where the subscript e applies to the gas properties outside the boundary layer, δ is the boundary layer thickness, n is the exponent in the turbulent boundary layer velocity profile [Lengellé (1975)], and ψ represents the

blocking effect of the gases being injected in the boundary layer and is expressed as:

$$\psi = \frac{\log(1+B)}{B} \left[1 + B \left(\frac{L_f}{\delta} \right)^{\alpha_B} \right] \frac{1}{(2 + \alpha_B)} \quad (2.16)$$

Here B is the injection strength parameter and is equal to $2(\rho v)_w / c_f \rho_e u_e$ and α_B is a second exponent in the turbulent boundary layer velocity profile [Lengellé (1975)]. c_f is a friction coefficient equaling $2\tau_w / \rho_e u_e^2$. Typical values for n are 0.3 to 0.4 while values for α_B is determined from the relation $\phi \approx \eta^{\alpha_B}$ where $\phi = \frac{u}{u_e}$ and $\eta = \frac{y}{\delta}$.

According to Lengellé, the erosive component of propellant combustion was due to the blocking effect as well as the external mass flow rate. This blocking effect caused the appearance of a threshold velocity, below which the normal burning rate occurred (Equation 2.10). The threshold effect was found to be particularly noticeable for propellants with a high normal burning rate, while the erosive effect was predicted to be more noticeable for propellants with a low normal burning rate. Lengellé's erosive burning model compared favorably with the results of Marklund and Lake.

2.2.3 Modification of the Propellant Combustion Mechanism

Using a simplifying assumption that the time for a unit of propellant to be consumed is the linear sum of the chemical reaction time plus the time required for the turbulent transport of heat to the propellant surface, Miller (1966) presented a mathematical model for erosive burning as follows:

$$\frac{P}{r_o \rho_p} = k_1 + \frac{k_2 P L^{0.2} e^{r_o/G}}{G^{0.8}} \quad (2.17)$$

where k_1, k_2 are proportionality constants determined empirically. As Miller himself noted, this model is similar to the Lenoir-Robillard model but is derived from a one-step controlling chemical reaction and the time taken to complete this reaction, as well the time necessary to complete turbulent heat transfer. As this simplifying assumption had no physical basis [Razdan and Kuo (1984)], this model was not well accepted, but later research into this category by King yielded a model based on the assumption that the crossflow of combustion gases bends the diffusion flame, which reduced the overall flame thickness and brought the heat release zone closer to the propellant surface which accelerated the burning rate [King (1977)]. This model is illustrated in Figure 2.3. Similar to Lengellé (Section 2.1.2), King estimated the thickness of the diffusion flame to be approximately 10 μm and to be located within the laminar sublayer of a turbulent boundary layer; however, in King's model, the flame thickness was affected by the crossflow within the laminar sublayer. The model involved the solution of eight simultaneous

equations and gave results similar to the experimental data obtained on a number of propellants characterized in the 1960's.

Figure 2.3: Schematic diagram of King's Two-Flame Model of Erosive Burning [King (1977)]. The height of the fuel-oxidizer mixing zone is reduced in the case with cross-flow, thus increasing heat transfer to the propellant surface with the resulting increase in burning rate.

2.2.4 Chemically Reacting Boundary Layer Analysis

As a logical development to the erosive burning modeling that had gone on previously, Razdan and Kuo (1978) proposed a model to predict erosive burning in the turbulent boundary layer of solid rocket motors which utilize composite propellants containing AP. This model was limited to situations where the propellant is burning under strong convective crossflow (i.e., above a threshold velocity). It has been well documented that a diffusion flame is always associated with the burning of an AP composite propellant. As

originally put forward by King, it is this diffusion flame which interacts with the turbulent crossflow. This, in turn, affects the heat flux to the propellant surface and the burning rate as a result.

Figure 2.4: The physical model considered by Razdan and Kuo in their turbulent boundary layer model [Razdan and Kuo (1978)].

Razdan and Kuo's model was developed in part using the conservation equations for a two-dimensional reactive turbulent compressible flow through a steady-state boundary layer as illustrated in Figure 2.4, with the following assumptions:

- a. flow properties are averaged;
- b. mean flow is two-dimensional;
- c. flow Mach number is of order one;
- d. no reaction generated turbulence;
- e. Lewis number is unity; and
- f. Fick's Law is valid.

A set of two-dimensional boundary layer equations was derived, which incorporated the $k - \varepsilon$ turbulence model:

$$\frac{\partial}{\partial x}(\overline{\rho u}) + \frac{\partial}{\partial y}(\overline{\rho v_o}) = 0 \quad (2.18)$$

$$\text{where } \overline{\rho v_o} \equiv \overline{\rho v} + \overline{\rho' v'}$$

$$\overline{\rho u} \frac{\partial \bar{u}}{\partial x} + \overline{\rho v_o} \frac{\partial \bar{u}}{\partial y} = -\frac{\partial \bar{P}}{\partial x} + \frac{\partial}{\partial y} \left[\mu_{eff} \frac{\partial \bar{u}}{\partial y} - \bar{v} \frac{\mu_t}{C_p \bar{\rho}} \frac{\partial \bar{P}}{\partial y} \right] \quad (2.19)$$

$$\overline{\rho u} \frac{\partial \bar{Y}_k}{\partial x} + \overline{\rho v_o} \frac{\partial \bar{Y}_k}{\partial y} = \frac{\partial}{\partial y} \left[\left(\frac{\mu}{Sc} \right)_{eff} \frac{\partial \bar{Y}_k}{\partial y} \right] + \bar{\omega}_k \quad (2.20)$$

$$\overline{\rho u} \frac{\partial \bar{H}}{\partial x} + \overline{\rho v_o} \frac{\partial \bar{H}}{\partial y} = \frac{\partial}{\partial y} \left[\left(\frac{\mu}{Pr} \right)_{eff} \frac{\partial \bar{H}}{\partial y} + \left\{ \mu_{eff} - \left(\frac{\mu}{Pr} \right)_{eff} \right\} \frac{\partial \bar{u}^2 / 2}{\partial y} \right] \quad (2.21)$$

$$\overline{\rho u} \frac{\partial k}{\partial x} + \overline{\rho v_o} \frac{\partial k}{\partial y} = \frac{\partial}{\partial y} \left[\left(\mu + \frac{\mu_t}{C_1} \right) \frac{\partial k}{\partial y} \right] + \mu_t \left(\frac{\partial \bar{u}}{\partial y} \right)^2 - \bar{\rho} \varepsilon - \bar{v} \frac{\mu_t}{C_p \bar{\rho}} \frac{\partial \bar{\rho}}{\partial x} \frac{\partial \bar{u}}{\partial y} \quad (2.22)$$

$$\begin{aligned} \overline{\rho u} \frac{\partial \varepsilon}{\partial x} + \overline{\rho v_o} \frac{\partial \varepsilon}{\partial y} = & \frac{\partial}{\partial y} \left[\left(\mu + \frac{\mu_t}{C_2} \right) \frac{\partial \varepsilon}{\partial y} \right] + C_3 \left[\mu_t \left(\frac{\partial \bar{u}}{\partial y} \right)^2 - \bar{v} \frac{\mu_t}{C_p \bar{\rho}} \frac{\partial \bar{\rho}}{\partial x} \frac{\partial \bar{u}}{\partial y} \right] \frac{\varepsilon}{k} \\ & - C_4 \bar{\rho} \frac{\varepsilon^2}{k} \end{aligned} \quad (2.23)$$

$$\bar{P} = \frac{R_u \bar{\rho} \bar{T}}{MW} \quad (2.24)$$

where μ_t is the turbulent viscosity, μ_{eff} is the effective gas viscosity ($\mu + \mu_t$), Y_k is the mass fraction of the k th species, Sc is the Schmidt number, $\dot{\omega}_k$ is the rate change of the k th species due to chemical reaction, H is the stagnation enthalpy, k is the turbulent kinetic energy, C_{1-4} are constants used in the turbulence model, and ε is the turbulent dissipation rate.

To calculate $\dot{\omega}_k$ in the previous equations, a single-step forward chemical reaction was assumed. The reasoning for this assumption was the very high rate of chemical kinetics due to the very high pressure and temperature under which combustion takes place. Heat transfer via conduction was assumed to be the dominant mechanism for transferring heat from the gas flow into the solid propellant in a direction that was normal to the burning surface. The burning rate could then be calculated using the Arrhenius law of surface pyrolysis:

$$r_b = A_s \cdot e^{\left(\frac{E_{as}}{R_s T_s}\right)} \quad (2.25)$$

where A_s is the Arrhenius frequency factor in surface decomposition and E_{as} is the activation energy. The link between the burning rate and the gas dynamics was assumed to be the surface temperature T_s . The surface temperature depends upon the heat flux from the gas flow to the solid propellant and the heat flux is evaluated by solving the gas phase conservation equations described previously.

In addition to comparing their results to experimental data which Razdan and Kuo generated using two different types of composite propellant, they also evaluated their method against the Marklund and Lake experimental data and found a very close agreement. Overall, they concluded from the theoretical results obtained that erosive burning increases due to the increase in heat feedback caused by the turbulence enhanced mixing and reaction of the oxidizer and fuel gases. As well, the increase in the gas flow velocity has an effect on the height of the boundary layer (see Figure 2.5) and consequently brings the location of the peak turbulence intensity closer to the propellant surface. Since more heat is released near the propellant surface, the burning rate also increases.

Figure 2.5: The effect of the cross-flow velocity on the inner region of the turbulent boundary layer. The height of the boundary layer is inversely proportional to increases in the cross-flow velocity [Razdan and Kuo (1978)].

2.2.5 Erosive Burning Summary

The aim of erosive burning research is to obtain a model that does not require burning rate versus pressure data that has been corrected for the effects of crossflow, but uses only propellant composition and ingredient size as inputs instead. While achieving this level of universality in an erosive burning model has not yet occurred, the mechanism describing erosive burning appears to be relatively well understood at present.

As can be seen from the review of the different erosive burning theories presented in this chapter, there has been an increase in the complexity of the mathematical models as the interaction between heat flux, turbulent flows and chemical reactions within solid rocket motors has become better understood. The development has also matched the increase in computing power available to obtain solutions through numerical methods. Since the mid-1980's there has been little additional development in the mathematical models describing erosive burning. However, in 1993, Bulgakov and Karpov published their work in further developing the chemically reacting boundary layer model at low Reynolds numbers to predict negative erosive burning.

What has become apparent is that the simple erosive burning models, while not as theoretically sound as those developed at a later date, still provide an accurate estimation of the effects of erosive burning. As an example, work published by Di Giacinto and Serraglia in 2001-2 on the modeling of a

composite propellant solid rocket motor utilizes the Lenoir-Robillard erosive burning model.

In the references reviewed concerning solid rocket motor combustion, the effect of erosive burning from a theoretical and experimental basis has been found to be dependent upon a number of factors. These factors, and their effect upon solid rocket motor combustion, are summarized here:

a. Crossflow Velocity and Pressure

- As the static pressure increases, the effect of erosive burning is reduced as the burn more closely resembles the linear burning law.
- An increased velocity will result in increased erosive burning as can be seen in Figure 2.6.
- Compressibility effects are significant with respect to erosive burning in terms of relating the Mach number to the phenomena [Saderholm (1964)].

b. Threshold velocity

- Zucrow et al. (1964) observed a threshold velocity above which erosive burning occurs, and below which negative erosion can occur. This research was based on lower pressure, lower velocity propellants (2.76 to 4.14 MPa) and is displayed in Figure 2.7 below. Other researchers

noted that the threshold velocity decreased when combustion was at higher pressures and/or temperatures.

- Negative erosion was attributed to the decreased heat transfer from the crossflow to the propellant surface caused by the mass of recently combusted propellant blowing into the boundary layer.
- Saderholm (1964) also observed the existence of a threshold velocity below which no erosive burning occurred as well as a transitional region and upper flow region which had differing rates of erosive burning. This transitional region existed in centreline crossflows where the Mach Number ranged from approximately 0.07 to 0.5).

Figure 2.6: Comparison of predicted burning rates to experimental results at various pressures and crossflow velocities [Razdan and Kuo (1982)].

Figure 2.7: Erosive burning rate data highlighting the presence of a threshold velocity [Zucrow et al. (1964)].

c. Propellant Initial Temperature

- Although a comparatively minor effect, Razdan and Kuo noted that as the propellant initial temperature decreased, the effect of erosive burning increased.

d. Freestream Gas Temperature

- As was shown by King, as well as Marklund and Lake, the effect of erosive burning is independent of gas temperature. This does not agree with the predictions for heat transfer erosive burning models (i.e., Lenoir-Robillard), but does not render these models unusable, but the constants used with these models must be modified as per the work of Lawrence et al (1968).

e. Normal Burn Rate

- A fast burning propellant's burn rate will be less affected by erosive burning. This was highlighted in experimental work carried out by Razdan and Kuo and is highlighted at Figure 2.8.

Figure 2.8: Effect of normal burning rate on the erosive burning augmentation factor. The erosive burning augmentation factor (r_b/r_o) is significantly higher for Propellant (I), which has a normal burning rate which is half that of Propellant (IV) [Razdan and Kuo (1982)].

f. Effect of metal addition

- Experimental work in this area was limited to the addition of aluminum to the composite propellant. It was observed that erosive burning appeared to be independent of the amount of aluminum added.

g. Effect of port size

- Early erosive burning models, which were based upon experimental data drawn from small diameter motors, tended to overpredict the effect of erosive burning on large diameter motors. Indeed, in the Reusable Solid Rocket Motors (RSRM) utilized on the Space Shuttle, erosive burning has not been observed [Wang (2003)]. However, in small diameter motors, both the Lenoir-Robillard and later models such as that of Razdan and Kuo predict an increase in erosive burning as the port size decreases.

h. Propellant characteristics

- It appears that the binder used in the propellant has the effect of increasing erosive burning in motors that operate at lower pressures (less than about 7 MPa). Above that pressure, the effect disappears, although this conclusion is based primarily on the experimental work of Marklund and Lake which incorporated polysulfide epoxy as the binder in their propellant. Double base propellants, which are typically faster burning than composite propellants also exhibit higher rates of erosive burning.
- Both King, and Razdan and Kuo carried out experimental work where the AP diameter in the composite propellant was varied to determine its effect on erosive burning. It was found that increased AP diameter resulted in increased erosive burning. Using the erosive burning

model of Razdan and Kuo, it was theorized that the cause of the increased erosive burning was due to the surface roughness of the propellant since this value increased as the diameter of the AP particles increased. This effect is displayed in Figure 2.9. It is believed that the increased roughness height also increases the turbulence near the propellant surface which enhances the fuel and oxidizer mixing process. This, in turn, increases the rate of reaction and the heat release rate which results in an increased burning rate. This effect lessened at low crossflow velocities due to the emergence of a thicker viscous sublayer.

Figure 2.9: Calculated erosive burning augmentation factor illustrating the effect of the propellant surface roughness (R_h) [Razdan and Kuo (1982)].

Chapter 3 *CFD Application*

3.1 Introduction

Following the selection process described in Appendix A, CFD-ACE+ was chosen as the commercial CFD application that would be used to model the motor under consideration. Chapter 3 describes the CFD application used along with a detailed discussion of the key modules and features incorporated to model regression of the propellant surface due to combustion within a solid rocket motor.

3.1.1 High-Level Solution Process

The high-level solution process incorporating CFD-ACE+ actually comprises four different applications [Anon, (2001)]. They are:

- a. CFD-GEOM – used to generate the grid that provides the basis for the CFD analysis. This application is able to generate multi-block structured, multi-domain unstructured and hybrid grids. During the grid generation process the computational domain is discretized. As well, the boundary conditions and volume conditions (material and source terms) are located on the grid.
- b. CFD-GUI – the Graphical User Interface which allows the user to activate the appropriate modules for the simulation, to define fluid and solid properties, to assign boundary and initial conditions and to set up the solver control parameters.
- c. CFD-ACE+ - generates the CFD solution by discretizing the simulation using the finite volume method and solving the resulting algebraic equations.
- d. CFD-VIEW – the post-processor which enables visualization and analysis of the simulation results.

This chapter will detail the generation of the CFD solution by the CFD-ACE+ application. While the other applications are necessary to obtain and visualize the CFD solution, they are not key to understanding the solution procedure and will not be discussed further.

3.2 CFD-ACE+ Solution Procedure

3.2.1 Discretization

To start the CFD process, the governing equations are discretized to produce a set of algebraic equations that can be manipulated and evaluated through numerical methods. CFD-ACE+ utilizes a finite volume approach in which the governing equations are integrated using numerical methods over each of the computational cells [Anon (2001)]. These governing equations (i.e., Navier-Stokes equations) are expressed in the form of a generalized transport equation:

$$\frac{\partial \rho \phi}{\partial t} + \nabla \cdot (\rho \vec{V} \phi) = \nabla \cdot (\Gamma \nabla \phi) + S_{\phi} \quad (3.1)$$

where the first term is the transient term, the second term is the convection term, the third term is the diffusion term and the last term is the source term. Of the variables, ϕ is a variable quantity (e.g., enthalpy), $\vec{V}$ represents the velocity field which is present at the cell, and Γ is the diffusion coefficient. These terms must be integrated over the finite volume surfaces. Note that at the geometric centre of the cell, the average value of all dependent variables and material properties within the cell are stored.

To integrate the terms which contain the product of several variables and variable gradients (i.e., the convective and diffusion terms), it is necessary to approximate the integrals by interpolation [Ferziger and Peric (2002)]. A number of schemes are available to carry this interpolation out, with varying

degrees of accuracy and stability. Options include first order upwind, central difference, as well as second and third order upwind schemes. Ferziger and Peric note that the first order spatial discretization (otherwise known as the upwind differencing scheme or UDS) unconditionally satisfies the boundedness criterion (will never yield oscillatory solutions) but is numerically diffusive, (i.e., diffusion errors occur in the direction normal to the flow). Furthermore, very fine grids are required to obtain accurate solutions using this interpolation scheme. The central differencing scheme (CDS) or second order spatial discretization results in 2nd order accuracy but oscillatory solutions may occur. This scheme offers a good compromise between accuracy, simplicity and efficiency and was utilized by Alavilli et al. (2000) in creating ROCFLO.

In general, higher order approximation does not guarantee a more accurate solution on any single grid, although they typically converge to an accurate solution with a reduced number of iterations. High accuracy is achieved only when the grid is fine enough to capture all of the essential details of the solution. This grid size is only determined through systematic grid refinement.

3.2.2 Velocity-Pressure Coupling

The momentum equations which make up the Navier-Stokes equations in part include a pressure term. As there is no pressure term in the generic transport Equation 3.1, the momentum equations require additional treatment to be

solved, in addition to solving the convective and diffusion (viscous) terms using the interpolation methods presented briefly in 3.2.1. By using the continuity equation (non-conservative form) and integrating it over a finite volume cell, the cell centre fluid velocity components are solved. A linear interpolation technique which is of the implicit pressure-correction type is then used to determine the cell face velocity components using a staggered grid which couples the velocity and pressure fields as well as aid convergence and minimize oscillations. This technique is utilized within CFD-ACE+ and is described in detail by Ferziger and Peric (2002), Anon (2001) and Patankar (1980). An example of a staggered grid is shown at Figure 3.1.

Figure 3.1: Two grid staggering arrangements used in CFD codes. The partially staggered arrangement is on the left in which the pressure nodes are at the cell centres, while the velocity derivatives are computed at the finite volume corners. In the fully staggered arrangement on the right, the velocity derivatives are computed at the finite volume face centres [Ferziger and Peric (2002)].

By solving for the fluid density at the cell faces using discretization methods previously described, the mass flux is now known. However, the Semi-

Implicit Method for Pressure-Linked Equations Consistent (or SIMPLEC) scheme is then utilized by CFD-ACE+ to solve for the pressure field using the mass flux values and a finite difference form of the momentum equations. The SIMPLEC scheme was proposed by Van Doormaal and Raithby (1984) and is based on the SIMPLE scheme originally proposed by Patankar and others.

The SIMPLEC scheme, as adapted for use in CFD-ACE+ is presented below utilizing the *x-momentum* equation written in finite difference form to solve for the cell centre values [Anon (2001)]:

$$a_p u_p = \left(\sum_{nb} a_{nb} u_{nb} + S_U \right)_p - \left(\sum_e P_e A_e n_{xe} \right)_p \quad (3.2)$$

where a is a finite volume coefficient, S_U is the upstream source term which represents all of the terms that may be explicitly computed with respect to u or other variables at the new time step (like temperature), P represents the pressure value, A is the area of the cell face and n_x is the face normal unit vector. The subscripts nb represent the values from neighboring cells, P represents the cell centre and e represents the faces of the cell.

To solve this equation, the pressure field must be provided; however, as it is not, a guessed value (or latest available pressure in an iterative procedure)

P^* is used to solve for u^* which will satisfy the following equation, but generally will not satisfy continuity:

$$a_p u_p^* = \left(\sum_{nb} a_{nb} u_{nb}^* + S_U \right)_p - \left(\sum_e P_e^* A_e n_{xe} \right)_p \quad (3.3)$$

Intrinsic to the SIMPLEC scheme is to find corrections to u^* and P^* to improve the solution. Let u' and P' stand for the correction values necessary to satisfy

$$u = u^* + u' \quad (3.4a) \quad \text{and} \quad P = P^* + P' \quad (3.4b)$$

By subtracting Equation 3.3 from Equation 3.2 an expression incorporating u'_p is obtained (3.5):

$$a_p u'_p = \left(\sum_{nb} a_{nb} u'_{nb} \right)_p - \left(\sum_e P'_e A_e n_{xe} \right)_p \quad (3.5)$$

Instead of neglecting the $\sum a_{nb} u'_{nb}$ term as is done when utilizing the SIMPLE method [Van Doormaal and Raithby (1984)], the u'_{nb} term is approximated by u'_p which results in the following expression for u'_p :

$$u'_p = \left(\sum_e P'_e A_e n_{xe} \right)_p d_p \quad (3.6)$$

where

$$d_p \equiv \frac{-1}{a_p - \sum a_{nb}} \quad (3.7)$$

Using the following definitions for V_e^n , the face-normal component of the velocity at cell face e and u_e , the velocity vector at the centre of face e is

$$V_e^n = u_e n_x + v_e n_y + w_e n_z \quad (3.8)$$

and

$$u_e = \gamma_e u_p + (1 - \gamma_e) u_E \quad (3.9)$$

where γ_e is the geometrical weighting function at face e and u_E is the x -component of the velocity vector at the upwind cell centre, an expression for u_e' , the velocity correction at the face is obtained as follows:

$$u_e' = \gamma_e u_p' + (1 - \gamma_e) u_E' \quad (3.10)$$

The mass flux at the cell face is defined as

$$C_e = \rho_e V_e^n A_e \quad (3.11)$$

and the continuity equation can be expressed in the following form

$$\frac{\rho \vartheta - \rho^o \vartheta^o}{\Delta t} + \sum_e C_e = \dot{m} \vartheta \quad (3.12)$$

where ϑ represents the cell volume. Similar to Equation 3.3, if u^* and ρ^* are used to evaluate C_e , the resulting C_e^* will not satisfy continuity. Therefore, let C' represent the correction to C so that

$$C_e = C_e^* + C_e' \quad (3.13)$$

and Equation 3.10 can be rewritten as

$$\frac{\rho_p' \vartheta}{\Delta t} + \sum_e C_e' = S_m \quad (3.14)$$

where S_m represents the mass correction in the control volume (i.e., the mass source in the cell) and is equivalent to the following expression:

$$S_m = \left(\frac{\rho_p^o \vartheta^o - \rho_p^* \vartheta}{\Delta t} \right) + \dot{m} \vartheta - \sum_e C_e^* \quad (3.15)$$

Using Equation 3.13 the following expression is derived:

$$C_e' = \left(\rho^* V^n' A \right)_e + \left(\rho' V^{n*} A \right)_e \quad (3.16)$$

Note that the products of the correction terms were neglected in deriving Equation 3.16 as their values approach zero.

To determine the density correction term, ρ' , the following expression is used for compressible flows

$$\rho' = \left(\frac{\partial \rho}{\partial P} \right)^* \cdot P' \quad (3.17)$$

Note that for incompressible flows $\rho' = 0$. Assuming the fluid is an ideal gas, the following equation of state is used to estimate the partial differential of density with respect to pressure:

$$\frac{\partial \rho}{\partial P} = \frac{1}{RT} \quad (3.18)$$

The correction to face-normal velocity is similarly obtained from Equation 3.8

$$V_e^n' = u_e' n_x + v_e' n_y + w_e' n_z \quad (3.19)$$

Substituting the velocity correction formula (Equation 3.6) into Equation 3.14 and incorporating Equations 3.16 and 3.19 a pressure correction equation is obtained:

$$a_p P_p' = \sum_{nb} a_{nb} P_{nb}' + S_m \quad (3.20)$$

Using the above expressions, the steps to the SIMPLEC scheme are summarized as follows:

1. Guess a pressure field P^* value (or use previous iteration value for P^*).
2. Obtain u^* (and v^*, w^*) by solving the discretized momentum Equation 3.3 or equivalent in y and z directions.

3. Evaluate C_e^* using Equation 3.13 and related equations to solve for ρ^* .
4. Evaluate mass source (S_m) using Equation 3.15.
5. Obtain P' value using Equation 3.20.
6. Use the P' value, solve for u' and correct the pressure and velocity field values (Equations 3.4a and 3.4b).
7. Solve the discretized equations for other flow variables.
8. Go back to step 2 and repeat cycle until convergence criteria is achieved.

Other commercial CFD codes such as FLUENT also utilize the SIMPLE scheme (or variations including SIMPLEC or SIMPLER) to solve the Navier-Stokes equations.

3.2.3 Relaxation

The algebraic equations being iteratively solved in generating a solution to the Navier-Stokes equations are non-linear. Often it is desirable to vary the speed of the changes in the variables from iteration to iteration. This process is known as overrelaxation or underrelaxation depending on if the variable changes are accelerated or decelerated, respectively. In particular, underrelaxation is often utilized to avoid divergence in solving strongly non-linear equations [Patankar (1980)].

To explain the concept of relaxation it is necessary to first examine the general transport Equation 3.1 which has been manipulated to the following finite difference form following numerical integration of all terms:

$$a_p \phi_p = \sum_{nb} a_{nb} \phi_{nb} + S_U \quad (3.21)$$

Within CFD-ACE+, two relaxation techniques are used to ensure convergence depending upon the variable. Linear relaxation techniques are used when determining the auxiliary variables: pressure, density, viscosity and temperature. In this case, the general discretization Equation 3.21 is modified so that the updated value of variable ϕ_p depends upon the current value ϕ_p^{**} and a relaxation factor I as follows:

$$\frac{a_p}{I} \phi_p = \sum_{nb} a_{nb} \phi_{nb} + S_U + (1 - I) \frac{a_p}{I} \phi_p^{**} \quad (3.22)$$

Inertial relaxation is used when determining the dependent variables: velocities, pressure correction, enthalpy, turbulence and species concentrations. In this case the discretization equation is replaced by the following relation:

$$a_p (1 + I) \phi_p = \sum_{nb} a_{nb} \phi_{nb} + S_U + a_p I \phi_p^{**} \quad (3.23)$$

By inspection, it can be seen that the effect of underrelaxation increases as the numerical value decreases in the case of linear relaxation, while the converse occurs with inertial relaxation.

3.2.4 Linear Equation Solvers

While pressure-corrector Navier-Stokes equations solvers such as the SIMPLEC scheme are fairly efficient for solving steady-state flow problems, their convergence can be improved significantly for both steady and non-steady-state flows through the incorporation of a multigrid strategy [Ferziger and Peric (2002)]. CFD-ACE+ incorporates the Algebraic Multigrid (AMG) solver [Lonsdale (1993)] with the SIMPLEC scheme to solve the linear equation set. This method is suitable for both structured and unstructured meshes. A summary of the AMG method is as follows:

1. On the fine (original) grid, perform iterations and compute the residual.
2. Perform iterations of the correction equation on the coarse grid. Note that the coarse grid has approximately half of the number of nodes as does the fine grid. The coarse grid is assembled through the use of a coarsening algorithm in which each fine grid node is joined to a neighbour with which it has the closest proximity. This coarsening algorithm is recursive and is repeated until the grid coarsening can be repeated no further. This is the key to the method for, as Ferziger and Peric note, on the coarsest grid, the

number of unknowns has become so small that the equations can be solved exactly with minimal computational effort.

3. Interpolate the corrections to the fine grid and update the fine grid solution.
4. Repeat the entire procedure until the residual is reduced to the desired level.

An important property of the AMG solver is that the number of iterations on the finest grid required to reach a given level of convergence is roughly independent of the number of grid nodes. As a result, the computational time increases only in relation to the number of unknowns and faster convergence can be achieved [Anon (2001)].

CFD-ACE+ also incorporates a Conjugate Gradient Squared plus Preconditioning (CGS+Pre) solver which is a non-stationary iterative method. Based on a personal communication with CFDRC, the AMG solver was utilized in modeling the grain regression due to combustion within the motor under consideration because it is better suited to deal with the large variation in cell size and pressure gradients which exist in the model as will be seen in Chapter 6.

3.3 CFD-ACE+ Module Overview

In undertaking the modeling of the combustion gas flowfield within the solid rocket motor under consideration, the following CFD-ACE+ modules were utilized:

- Flow Module
- Heat Transfer
- Turbulence module
- Chemistry module
- Grid deformation

In this section, a brief description of the theory associated with the module is listed as well as a discussion of each module's implementation with respect to the modeling carried out.

3.3.1 Flow Module

The flow module is the heart of CFD-ACE+ and is used to solve for the flow velocity field using the momentum equations and the pressure field using the SIMPLEC pressure correction algorithm. The theory behind the solution procedure is detailed in Section 3.2 and will not be repeated here.

3.3.1.1 Implementation of Flow Module

Two volume conditions are required to utilize the Flow Module. They are the fluid density and viscosity. In the case of density, the ideal gas law is used to calculate the local solution as follows:

$$\rho = \frac{P \cdot MW}{RT} \quad (3.24)$$

where the molecular weight of the various species which make up the products of combustion are entered using the Chemistry Module. Real gas effects and their possible effect on the solution are discussed in Chapter 4.

Flow viscosity was calculated according to Sutherland's Law which permitted viscosity to vary as a function of temperature as follows:

$$\mu = \frac{A_{SL} T^{3/2}}{B_{SL} + T} \quad (3.25)$$

where A_{SL} and B_{SL} are species specific coefficients. Values for some species were available from the species database included with CFD-ACE+ while others were obtained from Chapman and Cowling (1970). Values are entered into the species database using the Chemistry Module.

Three boundary conditions were used to define the combustion gas inlet into the computational domain, based on the fixed mass flow rate subtype. This subtype best simulates the actual flow phenomena occurring at the burning

propellant surface where the mass flow inlet is a function of burning rate (see Chapter 1) and solid propellant density. The pressure and temperature values along the boundary condition patch are also provided to calculate an initial density value for the flow. The direction of the flow entering the flow field could also be specified. In this case it was perpendicular to the inlet surface which reflects empirical results.

3.3.2 Heat Transfer Module

The conservation of energy equation is derived from the First Law of Thermodynamics and can take many forms. Within CFD-ACE+ the form known as the Total Enthalpy equation is used. This form is fully conservative and is as follows:

$$\begin{aligned} \frac{\partial(\rho h_o)}{\partial t} + \nabla \cdot (\rho \vec{V} h_o) = \nabla \cdot (k_T \nabla T) + \frac{\partial P}{\partial t} + \left[\frac{\partial(u \tau_{xx})}{\partial x} + \frac{\partial(u \tau_{yx})}{\partial y} + \frac{\partial(u \tau_{zx})}{\partial z} \right] \\ + \left[\frac{\partial(v \tau_{xy})}{\partial x} + \frac{\partial(v \tau_{yy})}{\partial y} + \frac{\partial(v \tau_{zy})}{\partial z} \right] + \left[\frac{\partial(w \tau_{xz})}{\partial x} + \frac{\partial(w \tau_{yz})}{\partial y} + \frac{\partial(w \tau_{zz})}{\partial z} \right] + S_h \end{aligned} \quad (3.26)$$

where h_o is the total enthalpy, k_T is the thermal conductivity of the species, τ_{ij} is the viscous stress tensor and S_h is the heat source term.

3.3.2.1 Implementation of Heat Transfer Module

Two volume conditions are required to utilize the Heat Transfer Module. They are specific heat and thermal conductivity. In the case of determining the

specific heats of the flow, the Mix JANNAF Method was selected. Using this option, the specific heat of the flow is calculated on a species mass fraction basis using the JANNAF curvefit coefficients for thermodynamic properties. Many of these curvefit coefficients are held within a CFD-ACE+ database for some of the species found within the products of combustion of the propellant being considered while the remainder of the coefficients were found via an Internet source [Burcat (2002)]. The thermal conductivity is calculated from the Prandtl number provided. For the modeling done, this Prandtl number was taken from the products of combustion output provided from the CEA program (see Appendix B).

There is only one boundary condition with respect to heat transfer and it concerns the wall boundary conditions. In this case, the walls were considered to be adiabatic, and the wall temperatures were calculated based on the flow heat flux.

3.3.3 Turbulence Module

Given the strong influence of turbulence on heat, momentum and mass transfer as well as the high Reynolds number of the flow within a solid rocket motor, this module was the subject of considerable research.

3.3.3.1 Treatment of the Burning Grain Surface

Modeling the flow within a solid propellant rocket motor presents a significant challenge as there exists a very high temperature, turbulent flow reaching velocities approaching sonic in the regions just upstream of the nozzle, as well as a combustion process along a thin layer on the grain surface which involves complex physical and chemical phenomena. Typically, the combustion along the grain surface is the subject of dedicated, detailed experimentation or modeling [e.g., Buckmaster et al. (2002)] and the burning surface is treated as a porous wall with surface mass injection at a given total temperature [Ciucci et al. (1998)].

3.3.3.2 Implementation of Turbulence Module

A number of turbulence modules exist within CFD-ACE+ including:

- a. Standard $k - \varepsilon$ model – based upon the work of Launder and Spalding;
- b. Renormalization Group (RNG) $k - \varepsilon$ model – based upon the work of Yakhot and Orszag;
- c. Kato-Launder $k - \varepsilon$ model – which is optimized for flow around bluff bodies;

- d. Two-Layer $k - \varepsilon$ model – based upon the work of Chen and Patel (where a single PDE model is used within the regions of the boundary layer where viscous effects dominate and two PDEs are solved within the remainder of the boundary layer);
- e. Low Reynolds $k - \varepsilon$ model – based upon the work of Chien;
and
- f. $k - \omega$ model – based upon the work of Wilcox.

The first three models are high Reynolds number turbulence models. This implies that the region between the wall and the fully turbulent part of the flow where viscosity has a significant effect on the flow (i.e., viscous sublayer and buffer layer) is not evaluated using the Reynolds Averaged Navier-Stokes equations, but by using a semi-empirical relationship instead known as a wall function (see Figure 3.2). This relationship uses the law of the wall to determine both the mean velocity of the flow in the streamwise direction at the grid point adjacent to the fully turbulent region, as well as determining the thickness of the viscous sublayer and buffer layer.

The law of the wall is defined as

$$\frac{u}{u_\tau} = \frac{1}{\kappa} \ln \frac{u_\tau y}{\nu} + C \quad (3.27)$$

where κ is the von Karman's constant (≈ 0.41), ν is the kinematic viscosity and u_τ is the friction velocity and is defined as

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} = C_\mu^{1/4} k^{1/2} \quad (3.28)$$

where τ_w is the surface shear stress, C_μ is a molecular viscosity constant equal to 0.09, and k is the kinetic energy of turbulence. The dimensionless velocity u^+ and distance y^+ are defined by the following relations:

$$u^+ \equiv \frac{u}{u_\tau} \quad (3.29) \quad \text{and} \quad y^+ \equiv \frac{u_\tau y}{\nu} \quad (3.30)$$

The wall function thickness (y_v^+) can then be found using the following relations:

$$u^+ = y^+ \quad \text{where} \quad y^+ \leq y_v^+ \quad (3.31a)$$

$$u^+ = \frac{1}{\kappa} \ln(Ey^+) \quad \text{where} \quad y^+ > y_v^+ \quad (3.31b)$$

Note that the two expressions for u^+ are alternate forms of expressing the law of the wall. For purposes of CFD-ACE+ calculations, E has a value of 9.0, the value determined for smooth walls. Within the cells between the wall ($y = 0$) and y_v^+ , the value of k used is the value specified at the wall, while the value of the dissipation of turbulence, ε , is determined by the following relation

$$\varepsilon = \frac{u_\tau^3}{ky} \quad (3.32)$$

Outside of the wall function the values for k and ε are determined through numerical integration of Equations 3.37 and 3.39.

Figure 3.2: Typical non-dimensional velocity profile for a turbulent flow over a flat plate [Hoffmann and Chiang (2000)].

The remaining models are low Reynolds number models which do not incorporate wall functions. These models contain additional terms to account for the dissipation processes which may be non-isotropic occurring within the boundary layer due to molecular viscosity.

Initially, it would appear that the low Reynolds number models would be the preference in achieving a CFD solution as the solution is a result of integrating the two transport equations all the way to the wall; however, these

models require a very fine grid near any walls to ensure accurate calculation of the wall shear stresses. Thus the first grid point away from the wall must be well within the laminar sublayer, with $y^+ \approx 2$. This greatly increases the computational expense of such a solution compared to a solution generated using a high Reynolds number model, where the first grid point away from the wall must be within the fully turbulent layer, ideally with $y^+ \approx 50 - 100$ [Anon (2001)].

A review of literature available indicates that when simulating solid rocket motor combustion, the $k - \varepsilon$ turbulence model is typically used [Razdan and Kuo (1978); Ciucci et al. (1998); and Allavilli et al. (2000)], although Wilcox makes a convincing argument in favour of the $k - \omega$ model using a number of examples of both wall bounded and free shear flow. These examples concentrate on the agreement between theoretical and measured values of skin friction within the boundary layer, and for the large part, do not deal with the performance of the $k - \omega$ model as applied to the fully turbulent region, outer region and wake of the boundary layer. In both cases, the same partial differential equations were derived to define the kinetic energy of turbulence (k) value, i.e.,

$$k = \frac{1}{2} \left[\overline{u'^2} + \overline{v'^2} + \overline{w'^2} \right] \quad (3.33)$$

where the flow velocities are the time averaged values of a fluctuating velocity. For a steady flow, the values of the flow velocities are constant. The PDE dealing with the dissipation of turbulence differs in the two approaches. For the $k - \varepsilon$ approach, the dissipation of turbulence is derived from

$$\varepsilon = \nu_{turb} \left(\overline{\frac{\partial u_i}{\partial x_j}} \right) \left(\overline{\frac{\partial u_i}{\partial x_j}} \right) \quad (3.34)$$

where ν_{turb} is the kinematic eddy viscosity. Note that the turbulent viscosity (μ_{turb}) is related to the kinematic eddy viscosity as follows:

$$\mu_{turb} = \rho \cdot \nu_{turb} \quad (3.35)$$

Using the $k - \omega$ model, the specific dissipation rate (ω) transport equation is derived from the following relation:

$$\nu_{turb} = k/\omega \quad (3.36)$$

The resulting equations (in standard form) describing the two turbulence models in one dimension are:

- a. $k - \varepsilon$ model
 - Turbulent kinetic equation

$$\rho \frac{dk}{dt} = \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_{turb}}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] + P_k - \rho \varepsilon \quad (3.37)$$

where P_k is the production of turbulence term and is defined as

$$P_k \equiv \tau_{ij} (\partial u_i / \partial x_j) \quad (3.38)$$

- Dissipation rate equation

$$\rho \frac{d\varepsilon}{dt} = \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_{turb}}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_j} \right] + c_{\varepsilon 1} P_k \frac{\varepsilon}{k} - c_{\varepsilon 2} \rho \frac{\varepsilon^2}{k} \quad (3.39)$$

Typical constant values are

$$\sigma_k = 1.0 \quad \sigma_\varepsilon = 1.3 \quad c_{\varepsilon 1} = 1.44 \quad c_{\varepsilon 2} = 1.92$$

- b. $k - \omega$ model

- Turbulent kinetic equation

$$\bar{\rho} \frac{dk}{dt} = \frac{\partial}{\partial x_j} \left[(\mu + \sigma^* \mu_{turb}) \frac{\partial k}{\partial x_j} \right] + P_k - \beta^* \rho \omega k \quad (3.40)$$

- Specific dissipation rate equation

$$\bar{\rho} \frac{d\omega}{dt} = \frac{\partial}{\partial x_j} \left[(\mu + \sigma \mu_{turb}) \frac{\partial \omega}{\partial x_j} \right] + \alpha \frac{\omega}{k} P_k - \beta \rho \omega^2 \quad (3.41)$$

where

$$\alpha = \frac{5}{9} \quad \beta = \frac{3}{40} \quad \beta^* = \frac{9}{100} \quad \sigma = \sigma^* = \frac{1}{2}$$

According to Hoffmann and Chiang (2000), the practical difference between the two models is that the $k - \omega$ model performs better than the $k - \varepsilon$ model within the laminar sublayer and the logarithmic region of the boundary layer. However, it has been shown that this model is strongly influenced by the value of ω outside of the boundary layer where the flow Re is much higher. Therefore, Hoffmann and Chiang suggest that the $k - \omega$ model is not ideal for applications that also involved the wake region of the boundary layer. In their opinion, and apparently that of a number of other researchers, the $k - \varepsilon$ model is considerably more accurate in the outer portion and wake regions of the boundary layer, but inferior in the inner region of the boundary layer. Unfortunately, it is not possible within CFD-ACE+ at this time to utilize different turbulence models within the same simulation. Therefore an initial assumption that can be made is that the use of a $k - \varepsilon$ turbulence model for analysis of the solid rocket motor is preferred. From the choice of models of that type available, it would appear that the Low Reynolds $k - \varepsilon$ model is the best choice for use in simulation of the functioning of the motor under consideration. This choice is made on the assumptions that it is preferable to model the entire boundary layer as well as the freestream as accurately as possible, as long as it will not cause excessive computational time to achieve a solution.

It was noted in a study of turbulent recirculating flows incorporating heat transfer [Ciofalo and Collins (1989)] that numerical predictions of that type

often are worse than predictions involving turbulent fluid flow only. At high Reynolds numbers, most of the wall to stream temperature difference is in the near wall region. This is particularly accurate in the case of the simulation being described in this thesis since it involves a very high Reynolds number ($O(10^6)$) as well as the heat source being the grain surface. Thus the requirements for the near wall modeling of the turbulence as well as the heat transfer can be met through the following methods:

- a. Use of a wall function which 'bridges' the conduction-affected layer;
- b. Use of a Low Reynolds number $k - \varepsilon$ model; or
- c. Use of a two layer $k - \varepsilon$ model.

Furthermore, research in this area by Collins, Chieng and Launder (1989) showed that:

- a. Numerical difficulties (i.e., slow convergence) are widely encountered when fine near-wall grids are used;
- b. Actual predictions from the Low Reynolds number models are often in error in the reattachment region following recirculation; and

- c. Two layer models are not easily implemented in general-purpose codes.

In addition, the previously mentioned additional computational expense of a Low Reynolds number or two layer model will play a very large role in the transient CFD modeling which will be carried out in developing this thesis. Based on these findings, the use of a high Reynolds number $k-\varepsilon$ model appears to be the best option, although the use of this type of model brings with it a potential source of inaccuracy due to the use of a wall function. The standard wall function described in Section 3.3.3.2 was derived assuming a high Reynolds number, wall bounded flow which is similar to the modeling being carried out within this thesis; however, results using wall functions give less reliable results when severe pressure gradients and non-equilibrium between the production and dissipation of the turbulent kinetic energy exist [Shyy et al. (1997), Anon (1998)]. A significant source of non-equilibrium is the existence of massive transpiration through the wall which may be a concern considering the transpiration at the propellant surface in a solid rocket motor. This phenomenon will be discussed in additional detail in Section 3.3.3.3.

As the Kato-Launder model is optimized for flow around bluff bodies, it is the least optimal for use in simulating the operation of the motor under consideration. The RNG $k-\varepsilon$ model is derived from the instantaneous

Navier-Stokes equations, using a mathematical technique known as Renormalization Group (RNG) methods. It is similar to the standard $k - \varepsilon$ model, aside from the values of the constants used in the dissipation rate equation as well as incorporating additional terms and functions in the kinetic energy and dissipation transport equations [Anon (1998)]. These slight changes in the values of the constants are due to the inclusion of Coriolis forces in the derivation of the model. These forces play a significant role in recirculating flows. Since some recirculation will be present within the flow, particularly in the region aft of the grain near the converging section of the nozzle (see Figure 3.3), the RNG $k - \varepsilon$ model will be utilized when simulating the flowfield within the motor under consideration, although some comparison with the results from other models will be carried out also.

The heat transfer across the wall function used in all of the high Reynolds number $k - \varepsilon$ models is based on a semi-empirical relation [Anon (1998)]. This relation accounts for heat conduction only within the viscous sublayer, where intermolecular forces dominate heat and momentum transfer [Chapman (1984)]. Within the region of the wall function where turbulence begins to play an effect upon the transfer of heat and momentum, both convection and conduction are accounted for in the relation [Chapman (1984), Anon (1998)]. Heat transfer due to radiation, both within the boundary layer as well as the fully developed flow, is considered negligible within the solid rocket motor [Kubota (1984)]. This assumption is based on the

observation that the vast majority of the thermal energy transferred to the gas flow is a result of the propellant combustion process.

In a similar method to that used in determining ε across the wall function thickness, the temperature across the wall function thickness is determined by non-dimensionalizing the following relation:

$$T^+ \equiv \frac{(T - T_w)\rho C_p u_\tau}{q_w} \quad (3.42)$$

Figure 3.3: View of the grid depicting the aft end of the grain and the converging section of the nozzle (2-D axisymmetric representation) at $t=0$ s. As the flow expands from the port to the converging section of the nozzle, recirculation will occur.

The profiles for T^+ are then determined for the two cases

$$T^+ = \text{Pr } y^+ \quad \text{for} \quad y^+ \leq y_T^+ \quad (3.43a)$$

$$T^+ = \text{Pr}_t (u^+ + P^+) \quad \text{for} \quad y^+ > y_T^+ \quad (3.43b)$$

where

$$P^+ = \frac{\pi/4}{\sin \pi/4} \left(\frac{A^+}{\kappa} \right)^{1/2} \left(\frac{\text{Pr}}{\text{Pr}_t} - 1 \right) \left(\frac{\text{Pr}_t}{\text{Pr}} \right)^{1/4} \quad (3.44)$$

and

A^+ is the van Driest damping coefficient (= 26)

Pr_t is the turbulent Prandtl number

q_w is the heat flux from the wall

y_T^+ is the thermal sublayer thickness

3.3.3.3 Consideration of Surface Mass Injection

Turbulence modeling of solid rocket motor combustion is complicated by the fact that the boundary layer along the grain surface formed by the crossflow of combustion gases is affected by the additional mass of combustion products being injected into the flow from the burning propellant surface. In general, this thickens the boundary layer and makes it more prone to turbulence. For solid rocket motor applications, the significance of this effect has been investigated in detail by Ciucci et al. (1998) who noted the effect of

mass injection on the boundary layer varies with the injection rate. The injection rate, V_{inj}^+ , is a dimensionless parameter and is defined as:

$$V_{inj}^+ = \frac{V_{inj}}{u_\tau} \quad (3.45)$$

where V_{inj} is the injection velocity and u_τ is the friction velocity as previously defined. For weak injection rates where V_{inj}^+ has a value up to $O(1/10)$, the typical structure of the turbulent boundary layer is retained (i.e., viscous sublayer and fully turbulent zone). For strong injection rates where V_{inj}^+ has a value up to $O(1)$, the presence of both the blowing and turbulent fluctuations of the injected fluid may affect the structure of the turbulent boundary layer. Thus, the effect of surface mass injection may or may not have a significant effect upon the turbulent boundary layer which would correspond to the anticipated reliability of the results using a high Reynolds number $k - \varepsilon$ model. The V_{inj}^+ value will be determined and further discussed in Chapter 6.

The suitability of the wall function with respect to heat transfer is also a concern due to the presence of the surface mass injection. This phenomena was investigated by White (1991) who noted that a $Pr_t = 0.9$ appeared to be valid for the cases of blowing surfaces or wall suction as well as surfaces through which no flow occurs. He also noted the similarity in velocity and temperature profiles in flat plate flow with suction and blowing, particularly for

flows where $y^+ \leq 100$. Thus the Reynolds analogy where a similarity between momentum transfer and heat transfer is assumed [Hoffman and Chiang (2000)] holds true for both cases. It is therefore assumed that the use of a wall function should provide a reasonable solution if the momentum transfer, hence the structure of the turbulent boundary layer, is relatively unaffected by the surface mass injection. This corresponds to the case of a weak injection rate. Once again, in the case of a strong injection rate, it is doubtful that the standard wall function would accurately describe the heat transfer from the propellant surface to the outer region of the boundary layer since the turbulent viscosity of the gas crossflow would be significantly increased due to the additional mass injection.

Using a turbulent boundary layer approach Razdan and Kuo (1978) applied the boundary conditions for k and ε near the wall rather than at the wall to avoid the low turbulent Reynolds number region near the wall. Results from this approach were compared in a later paper and were found to have produced an erosive burning model that was generally in agreement with empirical results [Razdan and Kuo (1982)]. The standard wall function previously referred to provides a similar functionality as all cells throughout the wall function thickness have a k value similar to that specified at the wall. However, the dissipation of turbulence value is not constant throughout the wall function thickness, which is not in agreement with the simplification of

Razdan and Kuo. This issue will be discussed further when examining the results of the modeling in Chapter 6.

3.3.4 Chemistry Module

The Chemistry Module in CFD-ACE+ offers two applications of applicability to combustion modeling. They are the simulation of mixing of two or more inert species or mixtures; and the simulation of mixing with gas phase reactions using either the instantaneous, equilibrium or finite-rate mechanisms. Due to the complexity and large number of reactants that are present in solid rocket propellant, including the GAP-ADN propellant proposed for use in the motor under consideration, the gas phase reaction capability of CFD-ACE+ was not utilized. Instead, the CEA output, which defines the products of combustion of the solid propellant under consideration, along with the mole fractions of the products, was used to define the constituents of the inert gas mixture that entered the computational domain at the inlet representing the propellant boundary. A detailed discussion of thermochemistry related to solid propellant rocket modeling is found in Chapter 4.

3.3.4.1 Implementation of Chemistry Module

Five volume condition properties are required to utilize the Chemistry Module. Two of these properties, density and viscosity, are also required to utilize the Flow Module and were discussed in Section 3.3.1.1. Another two of the properties, specific heat and conductivity, are also required to utilize the Heat

Transfer Module and were discussed in Section 3.3.2.1. The final volume condition which must be defined is the mass diffusivity.

It is also necessary to define the solution method within this module as either mixture mass fraction or species mass fraction. Given the assumption of homogeneous composition of the products of combustion made in CEA (see Section 4.2), by default, incorporated within the CFD-ACE+ simulation work, the simpler mixture mass fraction approach was utilized. Based on these limitations, diffusion could be represented by a constant Schmidt number only. The Schmidt number was discussed in Section 2.2.2 as part of Lengellé's development of an integral boundary layer analysis of erosive burning. The assumption made by Lengellé of a Schmidt number of unity [Lengellé (1975)] was also utilized in modeling the motor under consideration.

The boundary condition required to utilize the Chemistry Module is to define the mixture composition of the inert gases entering the computational domain at the inlets. Two mixtures were used in the modeling: the first was a mixture composed of the products of combustion (see Chapter 4) and a second mixture simulating air was used to define the initial condition across the entire computational domain, as well as entering through an inlet simulating the flow of ambient air at standard temperature and pressure entering the region aft of the rocket nozzle.

3.3.5 Grid Deformation Module

The Grid Deformation Module in CFD-ACE+ can be used to simulate fluid (gas or liquid) flow problems where the boundary of the fluid is in motion. It can also be used to force a known, time-dependent deformation upon the edges of the grid. This latter feature was utilized to simulate the regression of the solid propellant boundaries as would occur due to the burning of the propellant. This regression rate was applied to the inlet surfaces which represented the propellant faces using a user-defined subroutine to define the boundary movement.

According to Ferziger and Peric (2002), the chief concern with the use of moving grids is ensuring the conservation of mass particularly if the cell faces are moving in more than one direction. In modeling the motor under consideration, the vast majority of the boundary movement is in either the positive or negative y-direction, although some movement of the boundary simulating the aft end of the grain occurs in the x-direction. This error can be minimized by ensuring the grid movement is relatively simple and that small time steps are used. Following each time step, the grid is automatically remeshed using transfinite interpolation, a method of structured grid generation which determines the interior node distribution based on the motion of the boundary nodes [Anon, (2001)].

3.3.5.1 Implementation of the Grid Deformation Module

Each boundary that will be subject to movement must have that movement described when defining the boundary conditions. Options for grid deformation include translation and/or rotation which can be described either in terms of displacement/velocity or angle/angular velocity. A mathematical description of the deformation rate or displacement must be provided or, for more complicated motion, a *udeform_bc* subroutine is called. The use of a subroutine is necessitated in modeling the grain regression due to the extensive calculations required to determine the boundary deformation.

3.4 CFD Summary

CFD-ACE+ was selected as the commercial CFD application for use in this thesis because of its incorporation of the finite volume method as well as its ability to best model fluid flow and combustion while providing a moving mesh capability to simulate grain regression. The chief advantage of the finite volume method for the application being considered within this thesis is that it permits the use of a staggered grid (i.e., the control volumes on which the governing equations are applied need not correspond with the cells of the grid). The staggered grid approach is used to couple the velocity and pressure fields and has been found to aid convergence as well as minimize numerical oscillations. This coupling is key to the SIMPLEC scheme which is used within CFD-ACE+ (as well as some other CFD applications) to solve the Navier-Stokes equations.

To best model the flow of the products of combustion from the burning grain surface to the port and then through the nozzle, it was necessary to make certain assumptions about the flow. The key assumptions made were:

- a. The density of the products of combustion is determined in accordance with the ideal gas law.
- b. The flow viscosity is determined in accordance with Sutherland's Law.
- c. The specific heat of the flow is calculated on a species mass fraction basis using the JANNAF curvefit coefficients for thermodynamic properties.
- d. The thermal conductivity is calculated from the Prandtl number solved for during CEA program calculations for the propellant under consideration.
- e. The RNG $k-\varepsilon$ turbulence model, a high Reynolds number model, was used for simulating the flow. This model is well proven in similar simulations carried out by other researchers. The ability of the wall function to adequately address the transpiration at the boundary representing the propellant surface is a concern which will be

evaluated within this thesis. However, provided the surface mass injection is considered weak, it has been found that the typical structure of a turbulent boundary layer is retained thus acceptable reliability of the results when using a wall function can be expected.

f. The mass diffusivity is determined from a constant Schmidt number. This assumption in agreement with previous solid rocket motor modeling research.

The grid deformation feature of CFD-ACE+ is to be used to model the grain regression due to burning.

Chapter 4 *Boundary and Initial Conditions*

4.1 Introduction

As with any CFD modeling, establishing suitable boundary conditions are essential in achieving an accurate solution. For a CFD simulation of the flowfield characteristics encountered during solid rocket motor operation, the boundary conditions at the model's inlet are found at the burning propellant surface. Outside of empirical data which is not available for the motor under consideration, suitable boundary conditions must be obtained through modeling.

Modeling of the burning propellant surface is an extremely complex problem in itself due to the heterogeneous nature of the propellant and the complex chemical reactions occurring during combustion as well as the micron-scale resolution required. Other phenomena which have a significant impact upon solid rocket combustion include non-planar regression of the propellant

surface, unsteady heat conduction in the solid propellant, existence of a melt layer as well as the combustion layer, radiation and the influence of the chamber flowfield (shear, acoustics and turbulence) [Jackson and Buckmaster (2002)].

Use of CFD-ACE+ was considered as a possible means to carry out the actual modeling of the burning propellant, which in turn could provide the boundary conditions required to carry out the modeling of the chamber flowfield within the motor under consideration. Basic requirements for the modeling of the burning propellant surface included the ability to utilize a low Reynolds number turbulence model, as well as to model the chemical reactions occurring during the propellant combustion process. Given the existence of the Chemistry module within CFD-ACE+, a further examination of the application's capability was carried out. This module is able to provide the following gas phase reaction models:

- a. Instantaneous chemistry;
- b. Equilibrium; and
- c. Finite Rate reaction.

4.1.1 Instantaneous Chemistry Model

In the instantaneous chemistry model, the stoichiometric constants for both the reactants and products are provided as inputs by the operator. The local composition for each finite volume is determined from the solution of the

transport equations for mixture fraction variables. Then, the change in enthalpy between the products and the reactants is used as a source term in solving the simulation conservation equations for each iteration.

4.1.2 Equilibrium Chemistry Model

Within the equilibrium model, the composition of the reactants and products is determined by minimizing the Gibbs free energy of the system. The overall Gibbs free energy for the system (G_{mix}) is defined as:

$$G_{mix} = \sum_j N_j \left[g_j^o(T) + R_u \ln \left(\frac{P_j}{P^o} \right) \right] \quad (4.1)$$

where N represents the number of species within the system, and $g_j^o(T)$ is the standard state molar Gibbs energy which is a function of temperature. The subscript j refers to the chemical species. The Gibbs function (g) is defined as:

$$g = h - Ts \quad (4.2)$$

where h is the specific enthalpy and s is the specific entropy.

For a constant temperature, constant pressure system to be at chemical equilibrium, the chemically reactive constituents will continue to react until the system reaches chemical equilibrium in accordance with the second law of thermodynamics [Gauthier (2002)]. This model assumes that the chemical reaction proceeds so quickly that chemical equilibrium exists. For the

chemical reactions underway in a solid rocket motor, this is a reasonable assumption as the high temperatures and pressures contribute to the very high reaction rates. The system enthalpy is then determined based on the calculated chemical equilibrium, which is used in determining solutions to the conservation equations for each finite volume in the CFD model.

4.1.3 Finite Rate Chemistry Model

Within the finite rate model, the products of the chemical reaction occur at a rate limited by the Arrhenius form:

$$k_f = A_s \cdot e^{\left(\frac{E_{as}}{R_u T_s}\right)} \quad (4.3)$$

This model is limited to two reactants. Within this chemical kinetics controlled model, the mass fraction of the fuel is calculated by the solution of a transport equation with a source term due to the chemical reaction. The mass fraction for the other species is then calculated from the mixture fractions and the mass fraction of fuel.

4.1.4 Turbulence and Chemically Reacting Flows

The interaction between turbulence and chemically reacting flows can also be modeled within CFD-ACE+ using a Favre averaged probability density function (PDF) approach. The PDF model parameters are related to variables governed by the transport equations to determine the statistical description of the fluid at a single point in space and time. Use of the PDF

approach limits the number of mixtures modeled to two due to a limitation of the CFD-ACE+ application.

4.1.5 Simplifying Assumptions

Based on the limitations of CFD-ACE+ in modeling the interaction between turbulence and a chemically reacting multi-species flow such as that resulting from the combustion of the propellant being considered, a decision was made early to utilize an alternative approach to obtaining the thermodynamic properties of the products of combustion at the burning propellant surface for use as boundary conditions.

This alternative approach was based on utilizing a suitable propellant burning model to which the chamber conditions could be coupled with a propellant grain regression rate. This would allow the surface representing the burning propellant to be used as the inlet boundary conditions, although a method to determine the composition and properties of the products of combustion at the inlet would need to be determined.

The various suitable burning models available were examined in Chapter 2, most of which incorporated the erosive burning phenomena. Although it appears that there is a relationship between cross-flow, turbulence and chemical reaction at the burning propellant surface, the simpler models of erosive burning, such as that of Lenoir-Robillard, have achieved good

agreement with experimental data without considering this interaction. Thus, the Lenoir-Robillard erosive burning model was selected for use in simulating the flowfield within the motor under consideration. This decision was based on the simplifying assumption that an accurate simulation of the propellant burning could be made since well correlated results had been achieved in the past using erosive burning models that did not incorporate turbulence and chemical reaction interaction. Further, the use of the Lenoir-Robillard model, while developed using a typical cylindrical propellant grain with one burning surface, is assumed to be suitable for use within a rod-and-tube configuration grain (with two burning surfaces) due to the fact that the core gas flow in both cases is insulated from the burning propellant surface by a boundary layer (two boundary layers for the rod-and-tube configuration) all believed to have similar structures. It is this core gas flow which is responsible for the increase in heat flux to the propellant surface which affects the overall burning rate, thus it is assumed that the effect on the two burning propellant surfaces in the rod-and-tube configuration motor will be the same as the effect on the single burning propellant surface in a motor with a cylindrical grain.

Based on guidance [Stowe (2002)], the Chemical Equilibrium and Applications (CEA) software application was selected for the purpose of determining equilibrium chemical compositions and thermodynamic properties at the propellant surface. This application was initially released to the public by NASA Lewis Research Centre in 1988 and has been updated continuously

since that time [Gordon and McBride (1994)]. In addition it provides the ability to calculate theoretical rocket performance, thus providing a method of validating the CFD solution produced.

4.2 CEA Software Program

4.2.1 Chemical Equilibrium

Similar to the description of the Equilibrium model within CFD-ACE+, CEA also uses a Gibbs free energy approach to solve for the thermodynamic properties of the products of propellant combustion. Ideal gas law behaviour is assumed using this approach. To determine the point of chemical equilibrium of the products of combustion, the Gibbs free energy equation must be iterated until the Gibbs energy of the mixture is at a minimum. This iterative process is carried out in CEA using a Newton-Raphson solver which has a solution algorithm of general form:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (4.4)$$

Within CEA, the equations necessary to solve for chemical equilibrium take on the following form for solution using the Newton-Raphson method:

$$\Delta \ln n_j - \sum_{i=1}^{\ell} a_{ij} \pi_i - \Delta \ln n_m - \frac{H_j^o}{R_u T} \Delta \ln T = -\frac{\eta_j}{R_u T} \quad (j = 1, \dots, NG) \quad (4.5)$$

$$-\sum_{i=1}^{\ell} a_{ij} \pi_i - \frac{H_j^o}{R_u T} \Delta \ln T = -\frac{\eta_j}{R_u T} \quad (j = NG + 1, \dots, NS) \quad (4.6)$$

$$\sum_{j=1}^{NG} a_{kj} n_j \Delta \ln n_j + \sum_{j=NG+1}^{NS} a_{kj} \Delta n_j = b_k^o - b_k \quad (k=1, \dots, \ell) \quad (4.7)$$

$$\sum_{j=1}^{NG} n_j \Delta \ln n_j - n_m \Delta \ln n_m = n_m - \sum_{j=1}^{NG} n_j \quad (4.8)$$

$$\sum_{j=1}^{NG} \frac{n_j H_j^o}{R_u T} \Delta \ln n_j + \sum_{j=NG+1}^{NS} \frac{H_j^o}{R_u T} \Delta n_j + \left(\sum_{j=1}^{NS} \frac{n_j C_{p,j}^o}{R_u} \right) \Delta \ln T = \frac{h_o - h}{R_u T} \quad (4.9)$$

$$\sum_{j=1}^{NG} \frac{n_j S_j}{R_u} \Delta \ln n_j + \sum_{j=NG+1}^{NS} \frac{S_j^o}{R_u} \Delta n_j + \left(\sum_{j=1}^{NS} \frac{n_j C_{p,j}^o}{R_u} \right) \Delta \ln T = \frac{s_o - s}{R_u} + n_m - \sum_{j=1}^{NG} n_j \quad (4.10)$$

where n_j is the number of kg-moles of species j per kg of mixture, NG is the number of gases in the mixture, ℓ is the number of chemical elements, a_{ij} are the stoichiometric coefficients which define the number of kg-atoms i per species j , π_i is the Lagrangian multiplier for element i , n_m is the number of kg-moles of mixture per unit mass, H_j^o is the molar standard state enthalpy for species j , η_j is the chemical potential of species j , NS refers to the number of species in the mixture, b_k is the number of assigned kg-atoms of element i per kg of total oxidant ($k=1$) or total fuel ($k=2$), b_k^o is the number of assigned kg-atoms of element i of total reactant, $C_{p,j}^o$ is the molar heat capacity at constant pressure for standard state for species or reactant j , h is the specific enthalpy of the mixture, h_o is the specific enthalpy of the total reactants, S_j is the molar entropy of species j , S_j^o is the standard state

entropy for the species j , s is the specific entropy of the mixture, and s_o is the specific entropy of the total reactants.

The chemical potential of species j is written as:

$$\eta_j = \left(\frac{\partial g}{\partial n_j} \right)_{T,P} = \eta_j^o + RT \ln \frac{n_j}{n} + RT \ln P \quad (4.11)$$

where g is the Gibbs free energy. The expressions above which are in suitable form for solution using the Newton-Raphson solver contain the first derivatives of the thermodynamic variables needed to solve for the Gibbs free energy. While a complete derivation of all first derivatives will not be included in this document, an example as to how a derivative is solved for is found below:

As defined in Gordon and McBride (1994):

$$h = \sum_{j=1}^{NS} n_j H_j^o \quad (4.12)$$

where H_j^o is the standard state molar enthalpy for species j at temperature T .

and from Turns (2000):

$$dh = \left(\frac{\partial h}{\partial T} \right)_P dT + \left(\frac{\partial h}{\partial P} \right)_T dP \quad (4.13)$$

$$C_p = \left(\frac{\partial h}{\partial T} \right)_P \quad (4.14)$$

Gordon and McBride assume that the equilibrium value of C_p is the sum of a ‘frozen’ contribution as well as a ‘reaction’ contribution, i.e.,

$$C_{p,e} = C_{p,f} + C_{p,r} \quad (4.15)$$

where, from Equations 4.12 and 4.14

$$C_{p,f} = \sum_{j=1}^{NS} n_j C_{p,f}^o \quad (4.16)$$

and

$$C_{p,r} = \sum_{j=1}^{NG} n_j \frac{H_j^o}{T} \left(\frac{\partial \ln n_j}{\partial \ln T} \right)_p + \sum_{j=NG+1}^{NS} \frac{H_j^o}{T} \left(\frac{\partial n_j}{\partial \ln T} \right)_p \quad (4.17)$$

since n_m is an independent variable, and must be differentiated in conjunction with the following correction variables $\Delta \ln n_j (j = 1..NG)$ and $\Delta n_j (NG + 1..NS)$.

Recall also that $C_p^o = \frac{H^o}{T}$. Thus, for use in CEA

$$(\partial H)_p = C_{p,e} = \sum_{j=1}^{NS} n_j C_{p,f}^o + \sum_{j=1}^{NG} n_j \frac{H_j^o}{T} \left(\frac{\partial \ln n_j}{\partial \ln T} \right)_p + \sum_{j=NG+1}^{NS} \frac{H_j^o}{T} \left(\frac{\partial n_j}{\partial \ln T} \right)_p \quad (4.18)$$

4.2.2 Theoretical Rocket Performance

In an iterative fashion, CEA applies the solution of combustion calculations to either an Infinite Area Combustion Chamber (IAC) or a Finite Area Combustion Chamber (FAC) depending upon user inputs. In both cases, the

assumptions made in calculating flow through a rocket nozzle are one-dimensional forms of the continuity, momentum and energy equations; zero velocity at the combustion chamber inlet; complete, adiabatic combustion; isentropic expansion in the nozzle; homogeneous mixing; ideal gas law; and zero temperature and velocity lags between solid and gaseous species.

According to Sutton and Biblarz (2001), a necessary part of rocket motor prediction is the use of simplifying assumptions since the physical and chemical phenomena are complex, multidimensional, unsteady, non-linear, influenced by many variables, and too difficult to emulate mathematically otherwise. Typically, the actual performance is usually between 1 and 6% below the calculated ideal values, and is accepted practice to use ideal rocket parameters for rocket design. In the case of CEA, the assumptions and their effects are:

- a. One-dimensional continuity, momentum and energy equations
 - The principal result of this assumption is that pressure becomes constant over any given cross section. This greatly simplifies the solution of the conservation equations and is a reasonable assumption provided that:
 - Variation of the cross section of the duct is not excessive.
 - Curvature of the streamlines is not excessive.

- The velocity profile is known not to change appreciably along the duct.

These assumptions hold reasonably true for flow through a rocket nozzle due in part to the choking of the flow in the throat and supersonic flow in the diverging section of the nozzle. The flow is essentially axial in nature aside from a very thin boundary layer within the diverging section of the nozzle.

b. Zero velocity at the combustion chamber inlet – Assuming isentropic steady-state, steady flow, with no heat transfer from the flow to the walls of the nozzle and no work being done on the flow, then the first law of thermodynamics between any two points in the nozzle simplifies to:

$$0 = C_p(T_1 - T_2) - \frac{1}{2}(u_2^2 - u_1^2) \quad (4.19)$$

and since, for an ideal gas

$$C_p = \frac{\gamma R}{\gamma - 1} \quad (4.20)$$

and

$$\frac{T_2}{T_1} = \left(\frac{P_2}{P_1} \right)^{\frac{\gamma-1}{\gamma}} \quad (4.21)$$

then

$$u_2 = \sqrt{\frac{2\gamma}{\gamma-1} \frac{R_u T_1}{MW} \left[1 - \left(\frac{P_2}{P_1} \right)^{\frac{\gamma-1}{\gamma}} \right] + u_1^2} \quad (4.22)$$

As can be seen from the expression above, combustion chamber velocity does influence the value of the velocity of the gas flow at the nozzle exit. However, the chamber velocity is a function of the ratio of cross sectional areas in chamber and at the throat. If $A_1/A_t \geq 4$, then the chamber velocity, u_1 can be neglected. In low port-to-throat area rockets, this assumption is not reasonable, but for an infinite area combustor, it is clearly a reasonable assumption.

c. Complete, adiabatic combustion - According to Sutton and Biblarz (2001), the energy release efficiency (or combustion efficiency) is defined as:

$$\eta_{comb} = \frac{\Delta h_{actual}}{\Delta h_{calc}} \quad (4.23)$$

Because of difficulties in carrying out experimental measurements of combustion temperature and gas composition, combustion efficiency is rarely evaluated in the design of rockets; however, actual measurements on well designed rocket motors have revealed η_{comb} values of 94-99% [Sutton and Biblarz (2001)]. Based on these results, the assumption of complete combustion is reasonable. The adiabatic

assumption is made due to the comparatively minimal amount of heat transferred through the rocket walls compared to the energy that is expelled from the nozzle as combustion gases. This is a function of the smoothness of the nozzle throughout, which has a maximum roughness height of approximately 15 microns [Razdan and Kuo (1978)] and a thin boundary layer (approximately 100 microns thick) which occurs between the flow and the walls of the nozzle.

d. Isentropic expansion in the nozzle – This assumption implies that the gas flow from the nozzle throat to the exit plane is able to occur without the formation of a normal shock. This is typical of a well designed nozzle where the gas flow is able to expand from a relatively high pressure within the nozzle to a pressure approximately equal to the ambient pressure at the exit plane (the back pressure). In this case, weak oblique shocks are formed at the edge of the nozzle exit; however, these losses are insignificant compared to the kinetic energy of the flow exiting the nozzle. It is only when the flow is overexpanded and a normal shock is formed within the diverging section of the nozzle or at the exit plane that the isentropic expansion assumption is not valid [John (1969)].

e. Homogeneous mixing – This assumption is made due to the lack of good data on reaction rates with multiple simultaneous

chemical reactions in a turbulent flow environment. However, this assumption does not preclude CEA from providing solutions based on the shift in product composition from the chamber to the throat and the exit. Use of the shifting equilibrium method usually overestimates performance values by 1-4% above actual performance values according to Sutton and Biblarz (2001).

f. Ideal gas law – All common gases, within the most common range of temperatures encountered with thermodynamics, follow the ideal gas law:

$$P = \rho RT \quad (4.24)$$

The internal energy of ideal gases is a function of temperature only, and, as a result, allows the calculation of C_p and C_v on the basis of temperature alone. An ideal gas also has viscous properties and has a compressibility factor (Z) of unity where

$$Z = \frac{P}{\rho RT} \quad (4.25)$$

According to White (1991), Z remains close to unity even at very high pressures and temperatures. Figure 4.1 below illustrates this property for air.

Figure 4.1: Compressibility of air as a function of pressure. At very high temperatures (reduced temperature (T_r) of 10 or higher), the effect of pressure on compressibility is minimal. Note that the reduced pressure (P_r) is defined as P/P_c where P_c is the critical pressure and similarly, T_r is defined as T/T_c . The critical point is defined as the temperature and pressure where the reduced viscosity is a function of the reduced temperature and pressure (i.e., $\mu_r = T_r = P_r = 1$). For air, T_c is 132K and P_c is 3.69MPa [White (1991)].

At even higher temperatures the effects of dissociation of the multiatomic gases into monatomic gases becomes considerable and the ideal gas law no longer provides sufficient accuracy. However, for the range of temperatures (2800 – 4400K) and pressures (0.1 – 30 MPa) that are encountered when the products of solid rocket motor combustion flow through a nozzle, it is regarded as a reasonable assumption. Typically, this assumption results in an error of 0.2 to 0.7% [Sutton and Biblarz (2001)].

g. Zero velocity and temperature lags between solid and gaseous phases - The flow of combustion products in a solid propellant rocket motor is actually a multiphase flow as the products of combustion

include solid particles that must be accelerated by the gas flow. These particles are typically composed of elements of the metal fuel, such as aluminum, that is often utilized in high performance rocket motors. According to Sutton and Biblarz (2001), if these particles are very small (i.e., a diameter of 0.005 mm or less), their effect on the flow is negligible as an energy balance is maintained between the particulate and the gases by virtue of the kinetic energy given up by the gas flow to accelerate the solid particles and the thermal energy gained by the gas flow from the very hot particles. Implicit in this assumption is that the particles have zero volume. Larger particles do not transfer as much heat to the gas flow due to a reduced surface area to mass ratio, thus reducing the overall kinetic energy of the flow of combustion products through the nozzle. The size and quantity of particles in the flow are dependent upon the propellant and the geometry of the specific rocket motor and in the worst cases can reduce the specific impulse of the motor by 10-20%. This reduction in specific impulse is also due to slag formation, where the particles build up in the nozzle (throat and diverging sections), which reduces the flow rate. It also increases the surface roughness of the walls which results in additional wall friction due to the break up of the viscous sublayer.

4.2.2.1 Infinite Area Combustor (IAC) Calculations

To perform chemical equilibrium calculations for a given solid fuel mixture, it is only necessary to specify the fuel composition and enthalpy as well as the static pressure within the combustion chamber and the ratio between the combustion pressure and ambient pressure. Given the assumptions above, it is then possible to iteratively calculate the static pressure at the throat, where the Mach Number is unity, using the following relation [Sutton and Biblarz (2001)] and modified slightly by Gordon and McBride:

$$\frac{P_{\text{inf}}}{P_t} = \left(\frac{\gamma_s + 1}{2} \right)^{\frac{\gamma_s}{\gamma_s + 1}} \quad (4.26)$$

where the subscripts _{inf} and _t refer to the combustor and throat conditions, and γ_s is the isentropic exponent which is calculated using:

$$\gamma_s = - \frac{\gamma}{\left(\frac{\partial \ln V}{\partial \ln P} \right)_T} \quad (4.27)$$

Equation 4.27 was developed by Gordon and McBride based on the relation defining the velocity of sound a , i.e.,

$$a^2 = \left(\frac{\partial P}{\partial \rho} \right)_s = \frac{P}{\rho} \left(\frac{\partial \ln P}{\partial \ln \rho} \right)_s = - \frac{P}{\rho} \left(\frac{\partial \ln P}{\partial \ln V} \right)_s \quad (4.28)$$

and using the thermodynamic derivative (Bridgman) tables of Lewis and Randall (1961). The speed of sound is then defined as:

$$a = \sqrt{n_m RT \gamma_s} \quad (4.29)$$

Once Equation 4.26 is solved, the static pressure is known in the combustion chamber, the throat and at the exit. Using the assumptions described above, the thermodynamic properties of the flow can then be determined using the ideal gas law and nozzle flow theory, following which the chemical equilibrium can be determined using Newton-Raphson solver previously described. An example of the CEA output is attached as Appendix B.

4.2.3 Accuracy of CEA as a Prediction Tool

Based on the assumptions made in CEA and considering the geometry of the motor under consideration, it appears that the prediction of CEA will be higher than the performance anticipated using the CFD application. This is due to two key assumptions made in CEA that are not included in the CFD model:

- a. The motor under consideration has a low port-to-throat area ratio (1.72 at ignition). As a result, the velocity in the combustion chamber will not be negligible and will result in a loss of static pressure in the chamber which will have a negative impact on the flow velocity at the exit.
- b. Boundary layer effects will not be ignored in the CFD model.

4.3 Boundary Conditions Used

Typically two boundary conditions are required at the inlet [Hoffmann and Chiang (2000)]. Often the pressure and at least one of the velocity components of the fluid flow are the two boundary conditions utilized. Within the CFD-ACE+ Flow Module, it is necessary to provide the velocity, density and temperature for each cell face making up the boundary surface(s). These boundary conditions can be provided by one of four different methods:

- a. Fixed Velocity;
- b. Fixed Mass Flow Rate;
- c. Fixed Total Pressure; or
- d. Fixed Pressure.

Note that the pressure at the boundary surface is required by CFD-ACE+ in all cases as a boundary condition so that the density of the inlet flow may be calculated by the application. In the case of the latter two boundary condition methods, both the velocity and density are calculated by the application. To carry out the flowfield simulation of the motor under consideration, it was recommended that one of the first two methods of specifying boundary conditions be used [Slaby and Radhakrishnan (2002)] due to the extreme gradient between inlet and outlet boundary conditions.

While it is possible to obtain the pressure value (as well as the temperature and equilibrium chemical compositions) within an Infinite Area Combustor

from CEA results for use as a boundary condition for the case being modeled, that application does not provide an indication as to a velocity component of the products of combustion as they leave the propellant surface. Some empirical results of the gas flow velocity are available in literature [King (1977)]; however, these results refer only to a single unknown rate of grain regression. A relation between the gas flow velocity normal to the propellant surface as a function of grain regression rate was not available. Nevertheless, what was readily available was the mass flow rate based on the rate of grain regression according to the following relation since the burning rate of the propellant could be calculated using the Lenoir-Robillard erosive burning law:

$$\dot{m} = r_b \cdot \rho_p \cdot 2\pi(r_1 + r_2) \cdot L_{seg} \quad (4.30)$$

where r_1, r_2 refer to the rod outer and the tube inner radii from the axial centre of the rocket, and L_{seg} is the axial length of the grain segment for which the mass flow rate is being determined (see Figure 4.2). Thus, at the inlet, the fixed mass flow rate boundary condition type was utilized.

Figure 4.2: Schematic diagram of the rod-and-tube solid rocket motor under consideration. Dimensions r_1, r_2 and L_{seg} are indicated.

A second inlet was also incorporated to simulate ambient air moving into the region of the computational domain external to the rocket chamber and nozzle. This inlet was of the fixed velocity type. Before settling upon the dimensions of the ambient air region of the computational domain, the use of which was recommended by the application support staff [Slaby (2002)], a significant amount of testing was carried out which is discussed in Chapter 5.

Figure 4.3: Computational domain used for all modeling is shown. Yellow shaded area in schematic view of rod-and-tube motor indicates the portion of the chamber and nozzle that is included in computational domain (axisymmetric layout of motor exploited in assembling domain. Numbers in blue signify boundary conditions utilized: 1 – propellant grain inlet; 2 – ambient inlet; 3 – ambient outlet. Note that head end of chamber and nozzle walls are of boundary condition type ‘wall’. An example of the u -velocity flow contours are shown within this view of the computational domain.

The key boundary conditions used in all non-steady-state modeling are illustrated in Figure 4.3. Note that the stagnation temperature and pressure

values for the propellant grain inlets are taken from the CEA output (Appendix B) solutions for an Infinite Area Combustor (IAC) for the GAP-ADN propellant under consideration. Mass flow rate values were determined as a function of burning rate and chamber area in accordance with Equation 4.30. Ambient inlet and outlet conditions are air at standard temperature and pressure. Table 4.1 highlights the specific values used for the boundary conditions.

Table 4-1: Boundary condition values.

Location	Stagnation Temperature [K]	Stagnation Pressure [Pa]	Mass (Velocity) Flow Rate	Turbulence kinetic energy (k) and dissipation rate (ϵ)
Propellant Grain (Inlet)	3225	20685000	Combustion products determined in accordance with Equation 4.30 [kg/s/m]	$1488m^2/s^2$ and $1377000m^2/s^3$ respectively
Farfield (Ambient Inlet)	300	101325	Air (2-species model) at 1 m/s	$1488m^2/s^2$ and $1377000m^2/s^3$ respectively
Ambient Outlet	300	101325	Air (2-species model) at 1 m/s	$0.1m^2/s^2$ and $10m^2/s^3$ respectively

The turbulence kinetic energy value was calculated using the following formula [Anon (2001)]:

$$k = \frac{1}{2} \left[\overline{u'^2} + \overline{v'^2} + \overline{w'^2} \right] \quad (4.31)$$

where $(u', v', w') = I_{turb}(\bar{u}, \bar{v}, \bar{w})$ and I_{turb} is the turbulence intensity. The values for the velocity vectors were based upon the peak flow values calculated using the analytical model. A turbulence intensity of 10%, which is considered high [Anon (2001)], was used due to the high Re and turbulence typical of gas flows within solid rocket motors.

The turbulence dissipation value (ε) was calculated from the following relation [Anon (2001)]:

$$\varepsilon = \frac{C_{\mu}^{0.75} k^{1.5}}{\kappa \cdot L} \quad (4.32)$$

where $C_{\mu} = 0.09$, $\kappa = 0.41$ and L is the chamber hydraulic diameter (characteristic length).

While the use of a single value for the turbulent kinetic energy and dissipation at all inlets within the chamber is not ideal, the values were based on the flow velocities determined using an analytical solution (see Chapter 6) which were found to be representative of the flow velocities found along the bulk of the chamber length and in the farfield as determined using CFD-ACE+. This may be a possible source of error within the model. These same k and ε values were used at the ambient inlet due to the high velocity flow of the nozzle plume near the lower region of that inlet. It was essential that these values be high enough to ensure that backflow out of the inlet did not occur, which was

confirmed through examination of the mass flux summary during the simulations. Use of the same k and ε values as those within the chamber for the ambient inlet was recommended by the application support personnel [Slaby (2002)] who noted that this would not affect the results obtained within the region of interest (chamber) and would likely speed convergence as well by matching the values at the inlets with the initial conditions used throughout the computational domain.

Outlet boundary conditions were simply a two-species air model at standard temperature and pressure. Default values for the k and ε were used ($0.1\text{m}^2/\text{s}^2$ and $10\text{m}^2/\text{s}^3$ respectively), based on advice from application support personnel [Slaby (2002)], as these values are only significant in this case to prevent inflow from occurring at the outlets. Examination of the mass flow summaries which were produced during the modeling indicated backflow was not occurring.

4.4 Initial Conditions

Two sets of initial conditions were utilized during the non-steady-state modeling. The first set of conditions, which was used for the majority of the non-steady-state simulations, was simply to set the computational domain at standard temperature and pressure (STP), with each of the volumes in the chamber, nozzle and farfield containing a two-species air model. Initial turbulence values used in all volumes were those specified as the boundary

condition values. This approach was recommended by both user manuals consulted during the modeling work [Anon (1998) and Anon (2001)] so that convergence problems were minimized during start-up by ensuring that the initial turbulence values were close to the boundary condition turbulence values.

From Figure 4.4 it can be seen that this first set of boundary conditions is a blend of the conditions described for Phase I and II. While the chamber pressure utilized is the Phase I pressure, it is also assumed that all chamber surfaces have been ignited in the modeling at $t=0$ s, which actually occurs at the completion of Phase II. So the second set of initial conditions attempts to more accurately account for the conditions at the completion of Phase II, and was used in one non-steady-state simulation for comparison purposes with first set of initial conditions. For the second set, the gas inside all volumes remains as a two-species air model, but within the domain representing the chamber, the pressure is at 6.90 MPa (1000 psi) which is a typical, 'rule of thumb' value that would be found within the chamber following the completion of Phase II of the ignition sequence. Using the ideal gas law for that 'rule of thumb' chamber pressure, the initial chamber temperature was determined to be 996K. The remainder of the initial conditions were unchanged from the first set used. In all cases, any other effects of the ignition transient (i.e., acoustic instabilities, pressure oscillations and uneven flame spreading) were not considered.

While it is likely possible to simulate the ignition transient and the resulting flow of the products of combustion of the igniter solid propellant to create the initial pressurization using CFD-ACE+, this was beyond the scope of this work. In particular, the mechanism of flame spreading over the solid propellant and the subsequent ignition of flame is a specialized field of study in itself [Mumar and Kuo (1984)]. It should be noted that the effect of the ignition transient is to lengthen the total burn time of the motor and may cause a pressure peak as the propellant grain begins to burn. Both of these effects can be seen in Figure 4.4.

Figure 4.4: A typical ignition pressure transient portion of a solid rocket motor pressure time curve with the igniter pressure and motor chamber pressure traces shown. Phase I is the ignition time lag which is the period from the moment the igniter receives a pulse until the first portion of the grain surface starts to burn. Phase II is the flame-spreading interval which is the period from the first ignition of the grain surface until the complete grain surface area has been ignited. At the completion of Phase III, equilibrium chamber pressure and flow have been achieved [Sutton and Biblarz (2001)].

4.5 Boundary and Initial Condition Summary

Prior to commencement of the CFD simulation within the chamber of the motor under consideration, it was necessary to obtain a suitable boundary condition at the inlet representing the burning solid propellant. In lieu of modeling the complex, micron-scale combustion of the solid propellant which is beyond the scope of this work, a simplifying assumption was made to use the proven propellant burning model advanced by Lenoir-Robillard to represent the rate of grain regression based on characteristics of the flow within the chamber. Use of the CEA application was made to obtain the equilibrium chemical compositions as well as the thermodynamic characteristics of the propellant surface. The CEA calculations were made assuming an Infinite Area Combustion Chamber. CEA utilizes a Gibbs free energy approach to solve for the thermodynamic characteristics.

The boundary condition of the ambient air inlet as well as the outlet were assumed to be a two-species air model at standard temperature and pressure. Sensitivity analysis of the model to varying initial conditions was incorporated through the use of two separate initial conditions. This was done to assess the possible impact of the ignition transient on the model.

Chapter 5 *Grain Regression Model*

5.1 Introduction

While the ability to carry out CFD simulation of the flow within the chamber of a solid rocket motor exists in a number of locations, the vast majority of these simulations did not account for the regression of the grain that occurs as the solid propellant is consumed. Simulating the grain regression has been the purview of lumped parameter models such as SPP [Coats et al. (1987)] with the notable exception of the work of CSAR, who have incorporated grain regression into a CFD simulation using a supercomputer architecture [Alavilli et al. (2000)]. Carrying out a CFD simulation of the chamber flowfield on a solid propellant rocket motor which incorporates grain regression using a PC-based computer application is an area of research and development which has only very recently begun getting a significant amount of attention ([Favini et al. (2002)] and [Wang (2003)]). It is this area in which the purpose of this thesis was drawn (repeated from Chapter 1):

To develop a methodology, based on a commercial CFD application, to predict the flow characteristics within the chamber of a solid rocket motor that addresses all significant physical and chemical phenomena including erosive burning and grain regression. This methodology is to be demonstrated on a two-dimensional simulation of the internal flowfield of a rod-and-tube grain solid rocket motor.

5.1.1 Method Overview

The method developed in the course of this thesis was to enable the transient simulation of the chamber flowfield within the rod-and-tube configuration solid rocket motor under consideration. This section provides an overview of the method used, along with some of the key thermodynamic considerations and values utilized. It should be considered in context with the method utilized in operating the CFD-ACE+ application which is discussed in detail at Anon. (2001).

A non-reacting inlet mass flow, at a rate calculated in accordance with Equation 4.30, entered the computational domain along the entire length of the grain, perpendicular to the surface of the grid (see Chapter 4). This mass flow rate is based on the burning rate calculated using the Lenoir-Robillard erosive burning law and is detailed in Section 5.3. The products of combustion of the propellant which incorporates GAP-ADN (in thermochemical equilibrium in the chamber) were used to characterize the

gas mixture modeled. Density of the gas mixture was calculated using the ideal gas law, while viscosity was calculated according to Sutherland's Law which permitted viscosity to vary as a function of temperature. Specific heats as a function of temperature were calculated using the JANNAF coefficients for each of the species present in the propellant gas. Mass diffusivity is related to viscosity through a Schmidt number of 1.0 based on the work of Lengellé (1975). The turbulent Prandtl number (Pr_t) relating the turbulent convective heat transfer to the flow is 0.9 based on the work of Razdan and Kuo (1978). According to White (1991), of note is the apparent validity of $Pr_t = 0.9$ for the cases of blowing surfaces or wall suction as well as surfaces through which no flow occurs. Pr for the gas was 0.67 based on output from CEA for the GAP-ADN propellant.

Separate runs incorporating the simple burning law and the erosive burning law were completed. Each of these runs involved approximately 1000 time steps and required a minimum of 7 to 10 days to complete. All variable properties including the location of the boundary surfaces representing the grain (less the mass flow rate as previously mentioned) were recalculated by the application and associated subroutines each time step, which ranged from 0.01 ms to 0.5 ms. A shorter time step was used early in the simulation to prevent divergence of the non-linear equations being solved. To ensure comparability of results, the mass flow rate was updated after the same time intervals had passed for each of the simulations as the CFD software only

permitted mass flow rate updates to be made manually. As well, the same protocol was followed for each of the simulations in increasing the time step. These protocols are found at Table 5-1.

The iterations were continued until all boundary surfaces (see Section 5.2) representing the grain had regressed to a location representing the ‘all burnt’ condition. When the ‘all burnt’ location was reached, the boundary surface representing an inlet was changed to a wall, simulating the cessation of mass flow and deformation. By using a large number of boundary surfaces, it was possible to simulate some parts of the grain reaching ‘all burnt’ while other parts of the grain were still functioning. Note that the *y-value* corresponding to the location of the inlet surfaces was manually checked continuously during the run, and the boundary surfaces were changed from inlet to wall at the iteration immediately following the ‘all burnt’ being reached.

Table 5-1: Protocols used for non-steady-state runs in terms of time step per iteration and mass flow rate updates.

Iteration Number	Time step per iteration [ms]	Mass flow rate updates [cumulative time in ms]
1 - 20	0.01	0.01, 0.05, 0.1, 0.2
21 - 140	0.02	0.8, 1.4, 2.6
141 - 210	0.04	5.4
211 - 280	0.08	11
281 - 380	0.1	16,21
381 onwards	0.5	41, 71, 96, 131,166, 201 and then each 35 ms or until ‘all burnt’ was reached at one or more inlets

5.2 Grid Development

The construction of the computational domain is based upon mapping either a structured or an unstructured grid to the various curves and lines which make up, in the case under consideration, the physical space representing the chamber, nozzle and flowfield. The use of a structured grid necessitates a coordinate transformation between a typically occurring non-rectangular physical domain to a rectangular computational domain, while an unstructured grid is generated directly in the physical domain. An unstructured grid can more easily be made to fit irregularly shaped domains; but is also much more difficult to program [Hoffman and Chiang (2000)]. In the case of the model under consideration, it would clearly be advantageous to model the chamber and nozzle using an unstructured grid. However, due to a constraint within the CFD-ACE+ application, i.e., the inability of the application to utilize the grid deformation module with an unstructured grain, a structured grid was used to discretize the physical domain.

While the physical dimensions of the 2-D axisymmetric chamber and flowfield were given based on the DRDC Valcartier design under consideration, it was necessary to add an ambient region to the domain which provided an area into which the simulated high pressure, high temperature products of combustion exiting the nozzle could expand. This region of the domain occupied a far greater volume than the chamber and nozzle itself and a number of unsuccessful simulations occurred before an optimum size was

reached. The size of the ambient region was determined by trial and error resulting from the balancing of two conflicting requirements:

- a. the need for an ambient region large enough so that the static pressure of the tail end of the plume was equal to the outlet boundary condition pressure (atmospheric); and
- b. the need to keep the size of this ambient region to the smallest acceptable size so as to minimize the number of cells in the overall domain and reduce the overall computational time.

Figure 5.1: View of entire computational domain with cells visible. Plume exiting nozzle can be seen. A much larger volume for the individual cells, particularly in the portion of the ambient region where there was no interaction with the plume, was utilized to minimize the number of cells required in the domain.

Before settling on a final configuration, the grid underwent a number of adaptations to ensure optimum grid orthogonality, volume ratios and

appropriate y^+ value. For the steady-state simulations (see Chapter 6), the resulting number of elements was approximately 52 000 for the simulations utilizing the RNG $k - \varepsilon$ turbulence model.

For the non-steady-state solutions, a coarser mesh was utilized in the farfield region of the grid. As a result, the model was composed of 28800 cells; even with the reduced number of cells it was possible to maintain the y^+ values within a range of 30 to 150 as required for high Reynolds number turbulence models incorporating a wall function. Otherwise the same grid layout was utilized in an iterative process to determine the local burning rate and its effect upon mass flow throughout the remainder of the solid rocket model. During the development of the subroutines (see Appendices C and D) which were to be used to govern the grid deformation, it was determined that it was not possible to calculate the grid deformation at the node level. This finding, which was identified by the author to CFD Research Corporation [Radhakrishnan (2003)], generated the requirement for a software update which, at the time of writing this thesis, had not yet been completed. (Note that the use of user subroutines will be further discussed in Section 5.4). As a result, it was decided to divide the grid portion which modeled the port of the motor into 51 volumes which incorporated 54 surfaces subject to prescribed movement (see Figure 5.2). This was done to calculate the mass flux (G), local burning rate and mass flow rate at many locations along the lengths of the simulated inner and outer grid, recognizing that the burning and mass flow

rates would be identical within each volume. This compromise was felt to provide a reasonable granularity along the length of the grain which could provide an indication of the motion of the grid surface at many locations without requiring excessive computational time had the length of each volume been decreased. The majority of the volumes were equivalent to 0.04 m long with three at 0.022 m long.

Figure 5.2: The computational domain representing the chamber and nozzle. Note that all surfaces which run parallel to the x -axis are constrained to motion with a y component only. The surfaces representing the inner and outer ends of the grain are not parallel to the y -axis, as they would be according to the motor design, due to a grid adaptation required for the grid to successfully deform. These two surfaces are constrained to motion with an x component only.

5.3 Calculation of Lenoir-Robillard Constants

Recall the Lenoir-Robillard erosive burning law (Equation 2.6):

$$r = r_o + r_e = a_T P^n + \frac{\alpha G^{0.8}}{L^{0.2} e^{(\beta \cdot r \rho_p / G)}} \quad (2.6)$$

where r_o is the pressure dependent normal burning rate (simple burning law) and r_e is the erosive burning component. It was from this erosive burning law that the calculated mass flow rate of the products of combustion into the chamber was based on using the CFD application.

Prior to utilizing this law to determine the grid deformation rate to model the grain regression and its effect upon solid rocket motor combustion, it was necessary to determine the values for two propellant specific constants (α, β). The calculation of the first of the free constants for use with the Lenoir-Robillard (1957a) erosive burning relation (α) was carried out using the following relation (Equation 2.7):

$$\begin{aligned} \alpha &= \frac{0.0288 C_p \mu^{0.2} \text{Pr}^{-0.667} \left(\frac{T_f - T_s}{T_s - T_i} \right)}{\rho_p C_{ps}} \\ &= 0.365 \frac{\text{in}^{2.8}}{\text{lbm}^{0.8} \cdot \text{s}^{0.2}} \end{aligned} \quad (2.7)$$

A number of sources were used to obtain the various propellant properties required for the calculation of α . The principal source was the CEA output for GAP-ADN propellant. Other characteristics values used were the specific heat capacity for solid propellant (C_{ps}) [Kubota (1984)], the surface temperature (T_s) [Liau et al. (2000)] and the gas viscosity (μ) which was the

chamber exit value as calculated by CFD-ACE+ for a non-erosive burning case. The values of the properties were:

$$C_p = 1.706 \frac{kJ}{kg \cdot K} = 0.226 \frac{cal}{g \cdot R}$$

$$C_{ps} = 0.19 \frac{cal}{g \cdot R}$$

$$\mu = 3.902 \cdot 10^{-5} \frac{kg}{m \cdot s} = 2.181 \cdot 10^{-6} \frac{lbm}{in \cdot s}$$

$$Pr = 0.67$$

$$\rho_p = 1666 \frac{kg}{m^3} = 0.0602 \frac{lbm}{in^3}$$

Typically the other constant used in Equation 2.6, (β), is calculated based on actual interrupted firing of test motor data [Lawrence et al. (1968)]. As this data is not available for the case being examined, a range of values for β were chosen and evaluated. As can be seen from an analysis of the characteristics of the GAP-ADN, equally valid arguments can be made for assuming a high or low value of β . Note that from the work of Lawrence et al., it was determined that the lower the value of β , the higher the susceptibility of a given propellant to erosive burning.

- a. The initial work of Lenoir and Robillard (1957b), determined that the β value for polysulfide/AP propellants was 53. This propellant

was noted by them as having a relatively thick burning zone resulting in a sensitivity to the chamber pressure.

b. It is well documented [Lawrence et al. (1968), Razdan and Kuo (1982)] that the binder used in the propellant has a significant effect on the erosive burning rate. Since GAP is an energetic binder and has a higher enthalpy of formation and burning rate than HTPB or CTPB [Komai et al. (2000)], it may be assumed that the erosive burning rate for the GAP propellant will be higher than a propellant incorporating HTPB or CTPB as a binder. For a variety of AP/Aluminum propellants which incorporated a number of binders including UTREZ, CTPB and PBAN the lowest value of β obtained was 60 [Lawrence et al. (1968)].

c. Conversely, it was noted by King (1979) that changing the oxidizer/fuel ratio in the propellant to a less fuel rich formulation will result in an increase in the flame temperature and reduced sensitivity to erosive burning. Since the GAP-ADN propellant formulation being tested [Stowe (2003)] is close to stoichiometric, it is equally plausible that the value of β used would be relatively high. From the work of Lawrence et al. (1968), the highest value of β obtained was 125 using an AP/Aluminum fuel with CTPB as the binder and Fe_2O_3 as a burn rate modifier.

Note that the work of Lawrence et al. did not show a significant effect of pressure on the erosive burning parameters over a range of 3.5 to 18 MPa, which does not agree with the work of Lenoir and Robillard (1957a) or Zucrow (1964). Nonetheless, for the purposes of initial modeling of grain regression the range of values obtained by Lenoir and Robillard as well as Lawrence et al. (see Table 5-2), will be considered as the likely range of β values into which the GAP-ADN propellant would also fall. Within this thesis, modeling the chamber flowfield will be done over a range of β values representative of the results obtained in the work of Lawrence et al., and Lenoir and Robillard.

Table 5-2: Burning data summary for various propellant formulations. $\dot{r}_o$ is the normal strand burning rate and n is the linear burning rate exponent. Data taken from Lawrence et al. (1968).

Propellant	$\dot{r}_o$ at 1000 psia [in/s]	n	α	β
16% UTREZ, 68% AP, 16% Al	0.19	0.23	0.24	60
16% UTREZ, 84% AP	0.19	0.23	0.24	65
16% PBAN, 68% AP, 16% Al, Trace Fe_2O_3	0.4	0.30	0.24	70
16% PBAN, 68% AP, 16% Al	0.23	0.30	0.24	60
14% CTPB, 72% AP, 14% Al, Trace Fe_2O_3	0.43	0.30	0.24	125
17% CTPB, 73% AP, 10% Al	0.23	0.30	0.24	85

Since the α value is a function of known properties of the propellant composition, its value of $0.365 \frac{\text{in}^{2.8}}{\text{lbm}^{0.8} \cdot \text{s}^{0.2}}$ is assumed to be valid for all β values to be modeled (see Chapter 2 concerning use of Imperial units). Further, as the value for α is slightly higher for the GAP-ADN propellant than

the value calculated by Lenoir and Robillard for polysulfide propellants (0.365 vice 0.28), it appears that the effect of erosive burning will be significant for this propellant, irregardless of the β value used.

5.4 Grid Regression

The user subroutine functionality of CFD-ACE+ was used to determine the grid deformation which represented the grain regression. Programs which each incorporate two subroutines are attached as Appendices C (simple burning law) and D (Lenoir-Robillard erosive burning law). These programs are written in Fortran 90 and are compiled using a customized compiler that is included with CFD-ACE+.

User subroutines within CFD-ACE+ are used to exchange information with the CFD-ACE+ Solver that cannot be generated using the standard input settings available. Integral to the user subroutines are user access routines which are existing subroutines that can be called via a user subroutine to obtain variable information or pass updated information back to the CFD-ACE+ application (i.e., grid deformation rate).

To calculate and apply the grid deformation rate, it was necessary to utilize two user subroutines which are linked via an output file (PRESS.OUT). A general framework for these two subroutines (udeform_bc and uout which represent the subroutines used to calculate the grid deformation rate and to

measure the cell-centred pressures and locations next to the inlets respectively) is provided within the CFD-ACE+ documentation [Anon (2001)]; however, the specific functions which the subroutines execute must be coded by the user. These functions are highlighted at Figure 5.3.

While the *udeform_bc* subroutine provides an updated grid deformation rate to the CFD-ACE+ Solver every time step, it is not possible to update the mass flow rate at each inlet other than via manual entry to the CFD-ACE+ GUI. This mass flow rate (which is a function of burning rate, solid propellant density and burning surface area) is calculated every time step, but, for reasons of practicality, can only be entered after a number of time steps have passed (see Section 5.1.1). This is a source of error within the simulation and will be discussed further.

Note that the Lenoir-Robillard burning law was used to calculate the propellant burning rate (i.e., the grain regression rate) at all 52 of the boundary surfaces that would be influenced by the crossflow velocity (those with a horizontal orientation in Figure 5.2). The simple burning law was used in all cases to calculate the propellant burning rate for the two (of 54) boundary surfaces that would not have the crossflow velocity passing parallel to the surface (i.e., the propellant aft ends which have a close to vertical orientation).

Figure 5.3: Pseudocode for the two subroutines which make up the programs used to calculate and execute the grid deformation in accordance with the appropriate burning law. The DEFORM.DAT file is an output file which provides the user with updated mass flow rates at each of the inlets for every time step.

5.5 Grain Regression Summary

Building upon the grid deformation capability of CFD-ACE+ (Chapter 3), a grain regression model was designed as part of this thesis. This model linked the propellant burning rate, calculated using the Lenoir-Robillard erosive burning law, to a rate of grid regression for the inlet boundary surfaces

representing the burning propellant surface. The Lenoir-Robillard erosive burning law uses two constants which are based on various propellant and chamber flow characteristics. The first of these constants, α , was calculated using thermodynamic properties available from the CEA application for the GAP-ADN propellant being evaluated, as well as additional references. The second of these constants, known as β , is based upon matching the actual grain profiles from interrupted firing tests to calculated grain profiles. Because this latter data was unavailable, a range of likely β values were selected and will be evaluated within the scope of this thesis.

To utilize the grid deformation module, it was necessary to write a user subroutine to determine both the mass flow rate at the boundary inlets as well as the grain (grid) regression rate each time step. These values were determined based on various chamber flowfield characteristics including the local pressure and chamber cross sectional dimensions as well as the density of the solid propellant.

Chapter 6 *Results and Analysis*

6.1 Introduction

Based on the development of the 2-D axisymmetric CFD model that was described in Chapters 1-5, the results of the steady-state and transient simulations carried out based on the motor under consideration are described in this chapter. This model highlights the effect of erosive burning by comparing transient simulation results using various Lenoir-Robillard erosive burning constants against a transient simulation where the burning of the propellant is modeled using the simple burning rate law.

6.2 Analytical Model

Prior to commencing detailed CFD modeling of the motor under consideration, an analytical solution of the flow characteristics inside the combustion chamber as well as at the nozzle exit plane was carried out.

6.2.1 Combustion Chamber Calculations

Based on work done at DRDC Valcartier, the following characteristics were provided for the GAP-ADN fuel [Stowe (2003)]:

- a. Burning rate for this propellant is 0.0432 m/s at a nominal chamber pressure of 20.685 MPa.
- b. Burn rate exponent is 0.45.
- c. Density of the propellant is 1666 kg/m³.
- d. Density of the products of combustion is 19.64 kg/m³.
- e. Molecular weight of the products of combustion is 26.26.
- f. Combustion chamber stagnation temperature is 3225 K.

Using a timestep of 0.05 s it was then possible to calculate the effect of burning with respect to the locations of the internal propellant surfaces as well as the crossflow velocity. These calculations were based on a steady mass flow rate of combustion products from the propellant surface, which was equal at all locations on the propellant surface. The combustion products were assumed to behave in accordance with the ideal gas law while the effect of erosive burning was not considered in these calculations. The grain was divided into four segments of equal length (0.246 m each) so that the change in crossflow velocities along the length of the grain became apparent. Other dimensions at $t = 0$ s were provided by DRDC Valcartier [Stowe (2002)]. Note

that the mass flow rate of combustion products from a propellant surface is defined by the following relation:

$$\dot{m}_b = r_b \cdot \rho_p \cdot A_b \quad (6.1)$$

where A_b is the burning propellant surface area. The volume flow rate, $\dot{V}_b$, from the propellant surface is:

$$\dot{V}_b = \frac{\dot{m}_b}{\rho} \quad (6.2)$$

The crossflow velocity at the aft end of the segment, u_{seg} , was equal to the volume flow rate from the burning surface area divided by the area of the annular gap between the rod and tube propellant boundaries. The calculated crossflow velocities at the aft end of each segment as a function of time are plotted at Figure 6.1.

$$u_{seg} = \frac{\dot{V}_b}{A_{gap}} \quad (6.3)$$

The gas velocity normal to the burning propellant surface was defined as:

$$v = \frac{\dot{V}_b}{A_b} \quad (6.4)$$

Calculations were carried out until $t = 0.35$ s (grain thickness of 0.015 m divided by the burning rate of 0.0429 m/s). It should be noted that, based on these calculations, the burning surface area, gas flow velocity normal to the propellant surface and the mass flow rate at any axial point remained constant. These values are found in Table 6-1.

Table 6-1: Values of key burning rate characteristics which remain constant throughout the calculated burning time. The neutral burn characteristics of the rod-and-tube grain are highlighted by the fact that the total burning area remains constant in the absence of erosive burning or variations in burn rate with location. The gas velocity in the *y-direction* is calculated at the propellant surface.

Characteristic	Value
Total A_b	0.594 m^2
Total $\dot{m}_b$	42.418 kg/s
v	3.640 m/s

Figure 6.1: From the analytical solution, the crossflow velocities at four locations (normalized length as measured from the head end of the grain) are graphed as a function of burning time. As expected, the crossflow velocities are much higher during the initial time steps due to the reduced port volume that exists at the start of the burn.

6.2.2 Nozzle Flow Calculations

Theoretical nozzle flow is calculated using the assumption of a deLaval (converging/diverging) nozzle being utilized. As the motor under consideration utilizes a plug nozzle, this assumption remains reasonable for the purposes of calculating an analytical solution based on isentropic flow

through an ideal nozzle as a baseline. This assumption is based on the work of Hagemann et al. (1998) although they noted that this nozzle concept induces shocks and expansion waves which likely lower the theoretically achievable performance by several percent.

The conditions at the exit plane (typically given a subscript ₂) are solved for assuming isentropic flow through the nozzle as well as some values calculated in the previous section. Chamber conditions are given the subscript ₁. Pressure at the exit plane is assumed to be standard atmospheric.

Using Equation (4.22), the exit plane velocity is ($\gamma = 1.23$ based on CEA output):

$$\begin{aligned}
 u_2 &= \sqrt{\frac{2\gamma}{\gamma-1} \frac{R_u T_1}{MW} \left[1 - \left(\frac{P_2}{P_1} \right)^{\frac{\gamma-1}{\gamma}} \right] + u_1^2} \\
 &= \sqrt{\frac{2 \cdot 1.23}{0.23} \cdot \frac{8315 \cdot 3224.6}{26.26} \left[1 - \left(\frac{101325}{20865000} \right)^{\frac{1.23-1}{1.23}} \right] + 447.7^2} \\
 &= 2661 \text{ m/s}
 \end{aligned}$$

The effect of the combustion chamber velocity at the nozzle end of the grain (447.7 m/s) is significant. This value is the analytical solution at $t = 0$ s (i.e., instantaneously after ignition). Without this crossflow, the exit velocity would be 2623.3 m/s.

Considering an isentropic expansion of an ideal gas, the exit plane temperature can also be determined using the following relation (Equation 4.21):

$$T_2 = T_1 (P_2/P_1)^{(\gamma-1)/\gamma} = 3224.6 \cdot \left(\frac{101325}{20685000} \right)^{\frac{0.23}{1.23}} = 1193 K$$

The thrust, F , was estimated at 112888 N for the motor under consideration based on the analytical solution using the following relation [Sutton and Biblarz (2001)]:

$$F = \dot{m}_b u_2 + (P_2 - P_3) A_2 \quad (6.5)$$

where u_2 is the velocity of the propellant gases at the nozzle exit plane, and P_2 and P_3 are the local static pressures at the nozzle exit plane and ambient respectively. The first term in the thrust expression is termed the momentum thrust and the second term represents the pressure thrust. Note that for the analytical solution the value of the static pressure at the exit plane is assumed to be standard atmospheric, which is also the value of P_3 . Typically, the value of P_2 is significantly greater than standard atmospheric as the combustion gases have not undergone a complete expansion by the time they pass the nozzle exit plane. Also, this thrust value is termed the optimum thrust value since it assumes that all of the mass flow exiting the nozzle is traveling at a uniform velocity. Turbulence and boundary layer effects, flow divergence, and multiphase flow, to name a few, all cause energy losses in the

combustion products exiting a real nozzle. The area of the exit plane, A_2 , has been calculated from the dimensions provided by DRDC Valcartier.

6.3 Steady-State Results and Analysis

Prior to commencing a transient simulation of the regression of the propellant's burning surface (i.e., grid regression), a number of steady-state solutions modeling the chamber flowfield were produced. This verified the suitability of the numerical methods utilized as well as finalizing the detailed development of the grid. The steady-state flow results were also compared against results obtained using the CEA application for similar conditions within the chamber.

6.3.1 Discretization

Due to the extreme pressure gradients involved in the simulation, the governing equations were discretized using upwind differencing. This method requires a very fine grid (Section 3.2.1) but yields an unconditionally stable solution. A steady-state simulation utilizing the central differencing scheme was run for comparison purposes and is discussed at Section 6.3.1.1. Even utilizing the first order upwind scheme, it was necessary to run a simulation with transient time dependence for 200 iterations (0.01 ms each) before attempting a steady-state solution. This step was necessary so that the perturbation that formed on initiation was allowed to propagate out of the computational domain via the transient terms in the governing equations.

This perturbation (wave) was caused by the severe pressure gradient between the inlet boundary and the domain within the chamber and was a significant source of instability.

One other benefit of running the simulation for a number of transient iterations prior to attempting a steady-state solution is that it was not necessary to vary the volume initial conditions significantly from one volume to the next, particularly in the region of the nozzle. This is sometimes done in an attempt to set the initial conditions at values near to the anticipated solution so as to reduce the iterations necessary to achieve a converged solution. However, varying the initial conditions in each volume introduces step changes in pressure and velocity at the junction of the volumes. This, in turn, creates artificially high velocities early in the simulation which are a source of instability. Instead, by setting the initial conditions at standard temperature and pressure, the flowfield could develop in a manner more closely resembling the actual process whereby the high pressure, high temperature gas would flow to the lower pressure ambient region through the nozzle.

6.3.1.1 Effect of Varying Spatial Differencing

Although the upwind differencing scheme (UDS) was used during the majority of the simulations carried out due to its characteristic stability, this scheme is susceptible to diffusion errors (see Chapter 3) unless a suitably fine grid was utilized. To evaluate this possible source of error, the steady-state simulation

at 130 ms was run using a blended spatial differencing scheme which incorporated both the UDS and the central differencing scheme (CDS). The CDS is regarded as simple, accurate and efficient but is prone to oscillatory solutions as stated in Chapter 3 and further explained below.

The method used to carry out spatial differencing defines how the convective term (i.e., used to determine mass flux) is calculated at the cell face. The CDS essentially determines the solution variable at the cell face based on the previous, current and next cell centre values. This scheme is prone to divergence if the strength of the flow rate at the control volume faces is significantly greater than the diffusion conductance. This tends to occur in high Reynolds number flows [Patankar (1980)] and may result in the divergence of the discretization equations. To compensate for these difficulties, the upwind differencing scheme is used. Here the assumption is made that the cell face solution variable is the average of the current and next upwind (depends upon whether the direction of the flow rate is positive or negative) cell centre. The upwind scheme is much more stable, but as discussed in Chapter 3, may be prone to diffusion error.

CFD-ACE+ addresses the issue of divergence of the central difference scheme and the diffusion error of the upwind scheme by blending both schemes when solving for the solution variable as follows:

$$\phi_e = \xi \phi_e^{UP} + (1 - \xi) \phi_e^{CD} \quad (6.6)$$

where ϕ_e is the solution variable, ξ is a blending factor between 0 and 1, and the superscripts ^{UP} and ^{CD} refer to upwind and central difference, respectively.

A comparison of the results for identical simulations, other than the choice of spatial differencing schemes is shown at Figure 6.2. It was not possible to obtain a converged solution using the CDS alone and a number of blended schemes were utilized unsuccessfully as well. However, for values of $\xi \geq 0.8$ a converged solution was obtained. In Figure 6.2 a blended CDS with $\xi = 0.8$ was utilized for comparison with the upwind solution. The relative percentage difference ($\Delta\%$) between the results achieved was minimal as can be seen in Table 6-2. Note that the values used for comparison were obtained using a 100-point line probe along the centreline of the chamber. The $\Delta\%$ in values between each of the values was determined and then an average $\Delta\%$ was entered into the table, using the UDS value as the reference value.

Table 6-2: Relative percentage difference ($\Delta\%$) between Upwind Differencing Scheme (UDS) and Blended Central Differencing Scheme (CDS).

	P [MPa]	T [K]	u [m/s]	k [m ² /s ²]
$\Delta\%$	1.7%	0.31%	1.39%	0.49%

From these results it would appear that the choice of spatial differencing scheme has a very small effect upon the results obtained. This was consistent with the finding of grid independence discussed in Section 6.3.2. It would therefore appear that the grid being utilized is sufficiently fine to ensure satisfactory results using the UDS. Of more importance is the utilization of a

stable spatial differencing scheme due to the apparent instability of the simulation caused by the extreme pressures, temperatures and velocities occurring. Thus the UDS will be used for the remainder of the modeling carried out.

Figure 6.2: a. Static Pressure; b. Static Temperature; c. Crossflow Velocity; and d. Kinetic Energy of Turbulence as a function of normalized length. These results are taken from a line probe that extends along the centreline of the nozzle from the head end of the simulated chamber to the nozzle exit plane.

6.3.2 Grid Independence

One key property of a numerical solution is the achievement of a converged solution. A numerical solution is regarded as convergent if the solution of the

Figure 6.3: Results of numerical experiment to determine grid independence. Computational domain for the standard density grid was 52356 cells, 209344 cells for the double density grid and 13084 cells for the half density grid. Properties were measured from the chamber head end ($0.05 \bar{L}$) to the chamber aft end ($0.99 \bar{L}$) along the port centreline. Note that $\bar{L}$ is measured from the head end of the chamber.

discretized equations tends to the exact solution of the differential equations as the grid spacing tends to zero [Ferziger and Peric (2002)]. As the exact solution is unknown with respect to the CFD modeling being done within this work, an alternate accepted method of demonstrating grid convergence through numerical experiment involving successively refined grids will be utilized. If the solution converges to a grid independent solution (i.e., the solution obtained is independent of grid density), it is assumed that the numerical method is sound and that the discretization approximations are consistent.

The degree of grid refinement used in practice to check for grid independence is to double the resolution of the grid in each direction and compare the results [Stowe (2003)]. This was carried out for a representative solution, the steady-state solution generated at $t = 130$ ms. As can be seen from the

results of this numerical experiment, found at Figure 6.3, the differences between solutions at double the resolution are insignificant. The CFD solution for a grid of half the density was also determined and found to be consistent with those of the higher density grids. Based on the results of this numerical experiment, it is assumed that the solution on the proposed grid is stable and grid convergent.

6.3.3 Results and Discussion

6.3.3.1 Comparison with Reference Solutions

Using the RNG $k-\varepsilon$ turbulence model, steady-state solutions representing the grain location at 0, 130 and 270 ms during the functioning of the solid rocket motor were completed. These solutions simulate the conditions within the chamber following start-up and assume that the entire propellant surface is burning at a constant rate (i.e., uniform port height, defined as the difference between the inner and outer diameters along the length of the grain). In all cases, due to the high pressure in the motor chamber, a choked flow condition exists at the nozzle throat and the outflow is supersonic (variables are extrapolated). These results were compared both with the values produced using CEA as well as analytical results. Table 6-3 and Figure 6.4 highlight the comparison.

As a result of using the Infinite Area Combustor (IAC) model within CEA, it was not possible to compare any results predicted to occur within the

Figure 6.4: Comparison of crossflow velocity as a function of normalized chamber length for steady-state simulations at 0, 130 and 270 ms using CFD-ACE+ and Analytical solution.

chamber. However, the flow characteristics at the nozzle throat and exit plane were compared and were in close agreement as can be seen in Table 6-3. The slight differences in values between the different CFD-ACE+ results are not surprising due to the absence of symmetry between the inner and outer nozzle shell as can be seen readily in Figure 6.5. This resulted in a slightly oblique wave front where the Mach number of the flow was unity. At the vertical centre of the nozzle along this wave front was considered to be the throat. This location varied in the horizontal direction slightly between the three simulations due to upstream differences in the flow.

Table 6-3: Comparison of CFD-ACE+ steady-state results with reference calculations. $\Delta\%$ refers to relative percentage difference between the CEA values and those calculated using CFD-ACE+.

Properties (at nozzle throat)	CEA values	CFD-ACE+ values (at 0 ms) – Post-ignition transient	CFD-ACE+ values (at 130 ms)	CFD-ACE+ values (at 270 ms)			
		$\Delta\%$	$\Delta\%$	$\Delta\%$			
u [m/s]	1061	1077	1.5	1062	0.09	1058	0.28
ρ [kg/m ³]	12.12	13.01	7.3	12.67	4.5	12.67	4.5
T [K]	3033	3024	0.30	2924	3.6	2921	3.7
P [MPa]	11.90	12.98	9.1	12.14	2.0	12.21	2.6
C_p [kJ/(kg*K)]	1.906	1.897	0.90	1.884	1.2	1.884	1.2

The significant difference in crossflow velocities between that predicted using CFD-ACE+ and those values calculated using an analytical model as shown in Figure 6.4 was not unusual. The calculation method for the crossflow velocity using the analytical method determines the average flow velocity of the flow traveling through the cavity between an inner and outer cylinder. In the case of the steady state simulations, the peak, centerline crossflow velocity is shown which incorporates the downstream effects of the flow passing through a choked nozzle. No downstream effects, i.e., flow through a nozzle, were considered when calculating the crossflow velocity using the analytical procedure. What was significant was the agreement in trends between the two approaches; that is, highest crossflow velocity when the annular gap was at a minimum, while the difference in the values is in agreement with expectations in comparing average flow velocity through a rod-and-tube motor port to peak centerline flow velocity through a rod-and-tube motor port with a choked nozzle downstream.

Figure 6.5: Mach contours of flowfield within 2-D axisymmetric steady-state model of rod-and-tube configuration solid rocket motor at a) $t = 0$ ms, b) $t = 130$ ms and c) $t = 270$ ms.

Using the steady-state results it was possible to calculate the anticipated thrust values of the motor using Equation 6.5, similar to the method used with the analytical solution. Because of the variation in pressure and velocity across the exit plane, three steps were involved in calculating the momentum thrust and the pressure thrust. In the first case, the line probe function within CFD-ACE+ was used to measure the product of the local gas density times

the local velocity squared. These values were then integrated across the height of the nozzle face (2-D) using the trapezoidal rule [Gerald and Wheatley (1999)]. The momentum thrust could then be solved by integrating this value across the entire surface of the exit plane as follows:

$$F_{momentum} = \int_0^{2\pi} \int_{y1}^{y2} 0.5 \cdot (\rho_{gas} \cdot u_2^2) dr d\theta \quad (6.7)$$

where $y1$ and $y2$ are the inner and outer edges of the plug nozzle. The pressure thrust is calculated in a similar method, although it is the average static pressure across the exit plane that is integrated.

Values calculated for the total thrust at 0, 130 and 270 ms were within 2.7% of each other, and were comparable to results obtained by DRDC Valcartier [Farinaccio (2003)]. This is consistent with the nil change in burning surface areas at the different time intervals and is an advantage of the rod-and-tube grain configuration if a neutral burn is desired. As the burning surface area of the outer ‘tube’ increases, the burning surface area of inner ‘rod’ decreases by a similar amount, assuming both surfaces regress at an identical rate.

6.3.3.2 Validity of High Reynolds Number Turbulence Model

The use of a high Re turbulence model in conjunction with a wall function is limited when the flow departs significantly from the ideal conditions assumed where the law of the wall is valid. Examples of conditions where use of wall functions may result in significant error include [Anon (1998)]:

- a. Pervasive near wall effects such as flow through a small gap, highly viscous or a low velocity flow;
- b. Massive transpiration through the wall; and
- c. Severe pressure gradients leading to boundary layer separation.

In the case of solid rocket motor simulation, the second example (b.) is most relevant to the problem of simulating flow within the chamber since the method by which combustion gases are added to the flow can be considered as mass injection through a blowing surface (Chapter 3). As discussed in that section, the rate of mass injection has a significant effect upon whether the turbulent boundary layer will separate from the wall. Should separation occur, the use of a high Re turbulence model with a wall function is not suitable.

Using the steady-state solution at 130 ms, the injection rate (V_{inj}^+) was determined along the length of chamber from the head end to the commencement of nozzle convergence. These results are displayed graphically at Figure 6.6. The values of V_{inj}^+ were determined at two locations: at the inlet surface and at 0.00001 m above the surface. Averaged along the length of the chamber, V_{inj}^+ was 0.169 at the inlet surface and 0.149 at 0.00001 m above the inlet surface. This height above the surface is within the inner region of the turbulent boundary layer [Hoffman and Chiang (2000)] as it corresponds to an average y^+ value along the chamber length of 52.9.

The y^+ value was calculated using Equations 3.28 and 3.30. The values used to determine the kinematic viscosity (ν) as well as u_τ , which is a function of the kinetic energy of turbulence (k), were taken from the steady-state solution at 130 ms. A summary of the key variable values used in these calculations is listed at Table 6-4. Note that the values for ν and V_{inj} were within the range anticipated within a solid rocket motor [Majdalani (1995)].

Figure 6.6: Injection velocity and V_{inj}^+ as a function of normalized chamber length. The consistency of the results is, in part, due to the constant value for k that was utilized along the length of the chamber ($1488 \text{ m}^2/\text{s}^2$). The value of V_{inj}^+ remained $O(1/10)$ even when k was varied from 100 to 5000.

From the calculated values of V_{inj}^+ , it is determined that a weak injection rate exists and thus the typical boundary layer structure is valid (Chapter 3). Therefore the use of a high Reynolds number $k - \varepsilon$ model should provide an adequate representation of the turbulence anticipated in the near wall region.

The effect of varying k was not significant in terms of the strength of the injection rate. As V_{inj}^+ is inversely proportional to k it was found that a weak injection rate still existed even when the kinetic energy of turbulence value was 100. This value is well below those expected during the normal functioning of the motor given the dimensions and thermodynamic properties of the motor under consideration.

Table 6-4: Values of key constants and variables used in determining y^+ and V_{inj}^+ .
 V_{inj} , k , μ and ρ were taken from steady-state solution at 130 ms.

	V_{inj}	k [m ² /s ²]	C_μ^{**}	μ [kg/m*s]	ρ [kg/m ³]	ν [m ² /s]
At inlet surface	3.42 to 3.60	1488	0.09	8.12 x 10 ⁻⁵ to 8.07 x 10 ⁻⁵	23.5 to 19.2	3.45 x 10 ⁻⁶ to 4.18 x 10 ⁻⁶
0.00001 m above surface	2.78 to 3.63	1487 to 1532	0.09	8.12 x 10 ⁻⁵	22.0 to 19.5	3.70 x 10 ⁻⁶ to 4.16 x 10 ⁻⁶

** - Coefficient value from Anon (2001), also used by Wilcox (1998).

Recall Equation 3.32 which describes the turbulence dissipation values across the wall function:

$$\varepsilon = \frac{u_\tau^3}{\kappa y} \quad (3.32)$$

It is clear that ε will vary within the boundary layer thickness. However, Razdan and Kuo (1978) assumed a constant value for ε within the near wall region and achieved agreement with empirical results using their model. Using the steady-state model at 130 ms, the turbulence dissipation values

throughout the boundary layer, (i.e., the portion of the simulated flow at $y^+ \leq 400$) were determined and are displayed in Figure 6.7. However, when compared to the peak turbulence dissipation value across the chamber, the variation of ε is only 4.4%, which is in close agreement with the assumption of Razdan and Kuo to use a constant value for ε across the near wall region. This reinforces the suitability of a high Re turbulence model with a wall function for use in modelling the motor under consideration.

An indication of the development of the flow across the chamber is provided in Figure 6.8. In particular, the appearance of a fully developed, wall bounded flow is clear. Determination as to the distance from the inlet surface where the outer limit of the turbulent boundary layer ($y^+ > 400$) occurred was also checked and found to be in accordance with Prandtl's mixing length hypothesis [Warsi (1993)] as a thickening of the boundary layer occurred within the flow from the head to the aft end due to the increase in shear stress as the flow velocity increased. This is shown in Table 6-5.

Figure 6.7: Turbulence Dissipation values as a function of y^+ value at three locations along chamber length. Variation of the ε value as a function of distance from the inlet surface is obvious, particularly in the values measured at 0.75 of the normalized length of the chamber. It should be noted that the peak turbulence dissipation value across the chamber is $5.8 \times 10^7 \text{ m}^2/\text{s}^3$, thus the variation in ε across the boundary layer is minimal in comparison.

Table 6-5: Turbulent boundary layer characteristics as a function of chamber location. Results are consistent with theory describing wall bounded flow.

	x=0.25 L	x=0.5 L	x=0.75L
y value at height where $y^+ = 400$	0.000067 m	0.000069 m	0.000073 m
Centreline crossflow velocity	116.4 m/s	229.2m/s	307.2m/s
u_τ	21.38 m/s	21.60 m/s	22.17 m/s

Figure 6.8: Crossflow velocity profiles at different chamber locations – (a) $x=0.1 \bar{L}$, (b) $x=0.5 \bar{L}$, (c) $x=0.9 \bar{L}$, where $\bar{L}$ is the normalized chamber length.

In Figure 6.9, it can be seen that the increase of crossflow velocity brings the location of the peak turbulent intensity closer to the propellant surface (inlet). The trend in this finding is consistent with the erosive burning studies of Razdan and Kuo (1978). The location of peak turbulent intensity and its proximity to the propellant surface has a corresponding effect upon the erosive burning rate. The trend noted in Figure 6.9 is based on steady-state data simulating the motor functioning at 130 ms, and will be examined during the non-steady-state modelling.

Figure 6.9: Location of Peak Turbulence Intensity (k) from the propellant surface as a function of crossflow velocity. Data taken from steady-state run at 130 ms along length of chamber.

6.3.3.3 Use of a Low Reynolds Number Turbulence Model

Although the selection criteria used in choosing a turbulence model indicated that the use of a high Re model (RNG $k - \varepsilon$) was the most suitable, a number of steady-state attempts using low Re turbulence models were made as well. This modelling was not successful even though a number of Reynolds Averaged Navier-Stokes turbulence models were tried including: Chien's low Re $k - \varepsilon$ model, Wilcox's $k - \omega$ model and a two layer $k - \varepsilon$ model. It is believed that the difficulties in achieving a converged solution using a low Re model was due to the following reasons:

a. Previously mentioned difficulties encountered in literature when using low Re turbulence models in complex flows. This was explained in Chapter 3.

b. Additional grid development was still required. In using a low Re turbulence model, it is necessary to use a much finer grid spacing than that required for a high Re model. The structured grid used for the trial attempts had 232,000 cells and was found to be quite highly skewed with the aspect ratios of the cells changing rapidly in many parts of the grid. Given the need for a grid fine enough that the values for $y^+ \approx 1$ near the walls, it would have been necessary to utilize a grid of $O(10^6)$ cells. This optimization of the grid for a low Re model of the motor under consideration would have been very expensive computationally and increased the length of time required to do a single run to a number of weeks or months.

c. Additional calibration of the turbulence model to the grid itself was still required, particularly with respect to the boundary conditions at different locations for the turbulence intensity and turbulence dissipation as well as the relaxation values used. These parameters were found to be extremely critical in order to obtain a converged solution.

Finally, use of a low Re turbulence model and the fine grid that it necessitates result in a large computational expense even when properly calibrated. A single steady-state run preceded by a short transient run using the $k - \omega$ turbulence model took approximately five weeks. The results from this run did not agree with values obtained using the high Re turbulence model nor the analytical solution. As well, the residual values did not reach the recommended normalized value of 10^{-3} to 10^{-5} of the initial value during the simulation which further reduced the confidence in the results.

Based on the excessive computational expense and the questionable results achieved using a low Re turbulence model, as well as the suitability of the RNG $k - \varepsilon$ (high Re) turbulence model, no further simulation work was carried out with this type of turbulence model. Unless the computational resources are significantly increased beyond the current level, it is recommended that the use of low Re turbulence models be restricted to modelling simulations of small segments of the flow in regions where viscous effects prevail, such as near the head end of the chamber.

6.4 Non-Steady-State Results and Analysis

6.4.1 General

Five complete non-steady-state cases were simulated, each requiring approximately 10 days of computational time to complete. The cases run were: (a) grain regression in accordance with the simple burning law with the

chamber initially at STP; (b) grain regression in accordance with the Lenoir-Robillard erosive burning law with the chamber initially at STP using three different values for the β constant; and (c) grain regression in accordance with the Lenoir-Robillard erosive burning law with the chamber initially at 6.90 MPa for a single β value.

At Figure 6.10, the crossflow velocity for the erosive burning case ($\beta = 60$) at $t = 10$ ms is illustrated. Along the wall, within the region where the wall function applies a comparatively low speed flow in the x-direction occurs. It is notable that within a height of $10\mu\text{m}$ from the wall, $u = 10.1$ m/s and $v = 4.4$ m/s, where the y-component of the flow velocity is due to the mass flow rate entering the computational domain. These velocities correlate well with the flow profile used by King (1977) in deriving his granular diffusion flame theory in modeling erosive burning. The injection velocity of 4.4 m/s obtained during the non-steady-state modeling was consistent with the value of 3.6 m/s determined during the steady-state simulation (see Figure 6.6). The difference is believed to be due to the higher inlet mass flow rates which were calculated in accordance with the Lenoir-Robillard erosive burning model during the non-steady-state modeling, but were calculated using the simple burning law for the steady-state modeling.

Figure 6.11 provides an example of the effect of the ‘grain’ regression and its effect upon the computational grid.

Figure 6.10: Crossflow velocity in aft section of grain and converging portion of nozzle.

Figure 6.11: Comparison of grid in region of aft end of grain and converging portion of nozzle to demonstrate effect of grid deformation in simulating grain regression. (a) highlights the grid structure for the erosive burning case at $t=0.001$ s while (b) illustrates the grid at $t=0.18$ s.

6.4.2 Comparison of the Two Burning Laws

At Figure 6.12, the characteristic static pressure versus time curves are compared for the cases where erosive burning was incorporated, against the non-steady-state solution which incorporated the simple burning law only. Initial conditions for all cases are STP. The results are consistent with expectations as the increased burning rate due to erosive burning leads to a corresponding increase in propellant gas mass flow rate and causes the pressure within the chamber to rise above that of the simple burning case. In addition, the inverse relationship between the value of the β constant and the increase in burning rate due to erosive burning is also clearly demonstrated. In all cases, the increased mass flow rate also results in a shortened burn time for the same quantity of propellant.

The curve for the erosive burning cases where β had a value of 53 and 60 exhibited a relatively gradual drop off once the initial propellant ‘burn out’ occurred. This is an example of one of the undesirable characteristics of erosive burning where the grain burning (and thrust) gradually drops off, rather than burning consistently and producing the desired thrust until the propellant grain completes in a uniform manner along its length. Within the CFD model, the time required for the propellant to ‘burn out’, i.e., inlet in one or more volumes reaching deformation limits representing the chamber wall and reach ‘all burnt’, i.e., all inlets reaching their respective deformation limits values for each of the four cases is found at Table 6-6. Noteworthy is the

inability of the modeling to reach the ‘all burnt’ condition for the erosive burning cases due to divergence in the solution. The difficulty in reaching convergence is felt to be due, in part, to the differing grain deformation patterns observed between the two different burning laws followed. In the case of the simple burning model, the burning rate is a function of the local static pressure only; thus the highest rate of burning is at the head end of the chamber. When erosive burning is also considered, the burning rate becomes a function of local static pressure and crossflow velocity. In this latter case, the highest rate of burning is at the aft end of the chamber. This effect is illustrated through a comparison of the grain regression rates as a function of chamber location and is displayed in Figure 6.13. Choked flow continued at the nozzle throat for each of the cases until ‘all burnt’ or divergence occurred.

Table 6-6: Comparison of time required for propellant ‘burn out’ and ‘all burnt’ to be reached for simple burning and three erosive burning cases.

	Propellant ‘Burn Out’ Commences [ms]	Propellant ‘All Burnt’ Occurs [ms]
Simple Burning Law	300	331
Erosive Burning, $\beta = 53$	185	324
Erosive Burning, $\beta = 60$	198	294
Erosive Burning, $\beta = 120$	260	308

Figure 6.12: Head end Static Pressure vs time curves comparing the effect of erosive burning for three different values of β against the simple burning law. Head end value was measured using a point probe from a point located at $0.05 \bar{L}$ (normalized length).

An increase in peak crossflow velocity due to erosive burning occurred on only one occasion with for the erosive burning case where $\beta = 53$ as compared to the simple burning case. This increase was only 6.2 m/s (1% greater than the velocity obtained using simple burning law) at $t = 1.5$ ms. As anticipated, this value was obtained early in the simulation, when the port-to-throat ratio was lowest. What was noteworthy was the similarity in crossflow velocities between the simple and erosive burning cases and, along the majority of the chamber length, the higher crossflow velocity within the chamber for the simple burning case. These characteristics of the flow are highlighted in Figure 6.14, 6.15 and 6.16. Later during the simulations ($t = 100$ ms) the velocities had significantly reduced due to the increased port

diameter. Of note, is that during the mid to later stages of the simulated burn, the crossflow velocity for the simple burning case was slightly higher due to the fact that the port diameter was smaller while the mass flux for the two cases was similar. However, pressure, the other parameter which varies due to the presence of erosive burning, is still much higher for the erosive burning cases as was shown in Figure 6.12. This, in turn, has a significant effect upon the instantaneous mass flow rates which are compared in Table 6-7. Figure 6.15 also highlights the relationship between crossflow velocity and density within the chamber. When considering the increased chamber pressure (Figure 6.12), density and mass flow rates due to erosive burning, it is consistent that a reduction in crossflow velocity would occur due to the choked condition at the nozzle and the upstream effects which occur as a result.

Figure 6.13: Comparison of grain regression rate versus chamber length as a function of burning law. $\bar{L} = 0$ signifies the head end, while $\bar{L} = 1$ signifies the aft end. Even at the head end, grain regression rate is greater for erosive burning case ($\beta = 60$) since higher chamber pressures are predicted.

Figure 6.14: Crossflow velocity as a function of axial position along the grain at four different times during the simulation (0.1 ms, 1 ms, 10 ms and 100 ms) for the simple burning and erosive burning ($\beta = 60$) cases is illustrated. Of note is that in almost all circumstances, the crossflow velocity within the simulation incorporating the simple burning law is greater than that of the simulation incorporating erosive burning (for all values of β) with the exception of the crossflow at the aft end of the chamber.

Later during the simulations ($t = 100$ ms) the velocities had significantly reduced due to the increased port diameter. Of note is that during the mid to later stages of the simulated burn, the crossflow velocity for the simple burning case was noticeably higher due to the fact that the port cross section area was smaller while the mass flux for the erosive burning cases had dropped to less than the value obtained using the simple burning law (Table 6-8). The calculated values for the port cross section areas are found in Table 6-9.

Figure 6.15: Crossflow velocity as a function of time is illustrated. The trend whereby the simple burning case has a higher crossflow velocity than that of the erosive burning case is further highlighted. All values were measured at $0.99 \bar{L}$. Peak crossflow velocity occurred at 1.5ms

Table 6-7: Variation in total mass flow rates as a function of time for simple and erosive burning cases.

	t=0.001 s (1 ms)	t=0.01s (10 ms)	t=0.1 s (100 ms)
Simple burning	18.33 kg/s	41.72 kg/s	48.11 kg/s
Erosive burning			
$\beta = 53$	28.11 kg/s	94.43 kg/s	78.56 kg/s
$\beta = 60$	27.37 kg/s	90.16 kg/s	75.95 kg/s
$\beta = 120$	25.02 kg/s	67.83 kg/s	60.54 kg/s

Figure 6.16: (a) The crossflow velocity and (b) density as a function of normalized length at $t = 10$ ms for the simple burning case as well as the erosive burning case for each of the β values simulated is illustrated. (c) and (d) Similar to (a) and (b) at $t = 100$ ms.

Table 6-8: Total mass flux and erosive augmentation ratio compared for simple and erosive burning cases. Erosive augmentation ratio (EAR) is defined as the ratio of the total burning rate to the pressure dependent normal burning rate (r_b/r_o).

	t=1 ms		t=10 ms		t=100 ms	
	Mass flux [$\text{kg/m}^2\text{s}$]	EAR	Mass flux [$\text{kg/m}^2\text{s}$]	EAR	Mass flux [$\text{kg/m}^2\text{s}$]	EAR
Simple burning	3794	N/A	8373	N/A	6495	N/A
Erosive burning						
$\beta = 53$	3972	1.46	9597	1.92	5672	1.41
$\beta = 60$	3941	1.44	9478	1.86	5746	1.38
$\beta = 120$	4046	1.28	8904	1.50	5990	1.19

Table 6-9: Port cross section areas as a function of time and burning law.

	t=1 ms	t=10 ms	t=100 ms
	Port Cross Section Area [m²]	Port Cross Section Area [m²]	Port Cross Section Area [m²]
Simple burning	0.0048	0.0050	0.0074
Erosive burning			
$\beta = 53$	0.0048	0.0051	0.0099
$\beta = 60$	0.0048	0.0051	0.0096
$\beta = 120$	0.0048	0.0051	0.0084

6.4.2.1 Location of Peak Turbulence Intensity

Similar to the findings in Section 6.3.3.2, the results illustrated in Figure 6.17 also show the anticipated trend of the location of the peak turbulence intensity above the propellant surface actually approaching that surface as the crossflow velocity increases. This effect appears to be solely a function of the crossflow velocity and not the burning law, as the peak turbulence intensity is closest to the propellant surface for the simple burning case for the bulk of the flow at the locations examined.

Figure 6.17 also shows that for the majority of the flow, the peak turbulence intensity is closest to the propellant surface for the simple burning case for a given crossflow velocity. This difference in distance from the propellant surface to the peak turbulence intensity is slight and may be due to the difference in flow properties between the two cases. For example, the kinematic viscosity, which is a function of dynamic viscosity and density, is significantly higher along most of the chamber length for the erosive burning

case as shown in Figure 6.18. This is likely due to the increased mass flow rate (due to the increased burning rate) for the erosive burning case, which varies from 1.86 ($t = 10$ ms) to 1.38 ($t = 100$ ms) times the simple burning rate for the erosive burning case when $\beta = 60$.

Figure 6.17: Distance between Peak Turbulence Intensity and inlet surface as a function of crossflow velocity measured at the centreline at the same axial location. Note that measurements were made axially along chamber from $0.25 \bar{L}$ to $0.99 \bar{L}$ based on the flow conditions at 1.5 ms and 15 ms.

Figure 6.18: Kinematic viscosity along the chamber from $0.25 \bar{L}$ to $0.99 \bar{L}$ based on the flow conditions at 1.5 ms.

6.4.3 Effect of Initial Conditions

Two simulations in which the grid deformation was governed by the Lenoir-Robillard erosive burning law ($\beta = 60$) were carried out. The sole difference between these two simulations was the initial conditions used. The rationale for the two sets of initial conditions is described in Chapter 4. As can be seen from Figure 6.19, the pressure peak was significantly higher and earlier as a result of using the 6.9 MPa, 996 K initial conditions, as compared to the initial conditions simulating STP, thus highlighting the requirement to utilize accurate initial conditions in running a simulation of this type. What is significant as well is the similarity in the pressure-time curves after 0.05s and the point at which the grid deformation limit is reached (0.195 s for the STP initial conditions and 0.19 s for the high T and P initial conditions). This may be a function of the propellant combustion gas properties (including the

erosive constants α and β), since the crossflow velocity curves (Figure 6.20) are also so similar.

The instantaneous mass flow rates, which are a function of the burning rates, are compared in Table 6-10. The vast increase in $\dot{m}$ during the early stages of the simulation for the high T, P case over that of the case with the ambient initial conditions is clearly due to the pressure dependent portion of the burning law. This increased $\dot{m}$ would act to erode a greater proportion of the propellant, thus increasing the port area and correspondingly reducing the crossflow velocity for a given mass flow rate which is shown in Figure 6.20. As the simulations progress, the mass flow rates, chamber pressures and crossflow velocities become more similar between the two cases indicating the existence of a quasi-steady-state burning regime.

Figure 6.19: The effect of differing initial conditions upon the static pressure versus time curves is illustrated.

Table 6-10: Variation in total mass flow rates as a function of time for erosive burning where (a) initial conditions (IC) were simulating STP and (b) initial conditions (IC) simulated 6.895 MPa and 996 K.

	t=0.001 s (1 ms)	t=0.01s (10 ms)	t=0.1 s (100 ms)
IC – Case (a)	27.37 kg/s	90.16 kg/s	75.95 kg/s
IC – Case (b)	98.34 kg/s	122.86 kg/s	74.15 kg/s

Figure 6.20: The effect of the different initial conditions on crossflow velocity as a function of time is illustrated. Although significant, the differences in crossflow velocity between the two simulations are minimal. For the case where the IC were 6.895 MPa and 996 K, $u_{\max} = 583.45 \text{ m/s}$, while for the case where the IC were 101325 kPa and 298 K, $u_{\max} = 636.62 \text{ m/s}$. All values were measured at $0.99 \bar{L}$.

6.4.4 Nozzle Inlet Profiles

Often, the nozzle inlet characteristics available to the nozzle designer consist of uniform, one-dimensional properties. While this level of accuracy (typically $\pm 3\%$) is suitable for estimating motor thrust levels, etc., it is not a sufficient level of accuracy for carrying out detailed nozzle design work, for example

[Stowe (2003)]. The characteristic profiles at the nozzle inlet for the motor under consideration were measured using the CFD model. A view of the nozzle inlet profile is shown at Figure 6.21. As can be seen in Figures 6.22 to 6.26, the assumption of uniform properties across the nozzle inlet is a source of potential error, particularly during the early stages of the motor functioning where the crossflow velocity is the highest. Note that a normalized chamber radius of '0' represents the propellant surface of the propellant 'rod' and the value of '1' represents the propellant surface of the propellant 'tube'.

Figure 6.21: Crossflow velocity in aft section of grain and converging portion of nozzle. Nozzle inlet plane is highlighted.

Figure 6.22: (a) Local static pressure values for simple and erosive burning cases measured at nozzle inlet plane. Elapsed burning time is 1.5 ms. (b) Similar to (a) for elapsed burning time of 60 ms.

Figure 6.23: Local crossflow velocity values measured as a function of port height at nozzle inlet plane. Elapsed burning times for time 't1' is 1.5 ms and 60 ms for time 't2'

Figure 6.24: Local velocity values for flow normal to chamber axis, measured as a function of port height at nozzle inlet plane. Elapsed burning times for time 't1' is 1.5 ms and 60 ms for time 't2'

Figure 6.25: Local temperature measured as a function of port height at nozzle inlet plane. Elapsed burning times for time 't1' is 1.5 ms and 60 ms for time 't2'

6.5 Discussion

6.5.1 General

From the results obtained, it appears that erosive burning is a phenomenon which will have a severe impact upon the actual performance of the solid rocket motor under consideration by DRDC Valcartier. While the crossflow velocity is only moderately changed from the simulation incorporating the simple burning law, the peak static chamber pressures predicted when the erosive burning model is incorporated are between 22.3 and 59.9% greater than the case where the simple burning model is used, depending upon the value for the β constant. As well, the trends observed in the pressure time curves (Figure 6.12) appear to match the trends noted in literature, e.g., higher peak static pressure and shorter burning time. A comparison of typical pressure-time curves is shown at Figure 6.26.

Figure 6.26: Typical pressure-time curve with and without erosive burning [Sutton and Biblarz (2001)].

Interim results and discussion concerning the CFD modeling of erosive burning were presented and defended at Appendix E. In the first of these

papers that were presented, the simulations were made prior to the completion of the subroutines used for the results displayed within this document and slight differences in the results are present. These differences are primarily due to an inability to update the grid regression rate at each time step, which were instead updated along with the mass flow rates following a protocol similar to that presented in Chapter 5.

6.5.2 Use of a 1-D Erosive Burning Model

Many of the erosive burning models advanced by researchers such as those discussed in Chapter 2, are based on the assumption of 1-D flow through the chamber. With the advent of CFD on a widespread basis, there has been some use of these 1-D models for two- and three-dimensional simulations. A recent publication by Wang (2003) is an example of this philosophy, who carried out a number of steady-state CFD solutions at equally spaced temporal intervals during the functioning of the solid rocket motor. Wang used a modified version of the Saderholm erosive burning model, which is shown in its original form as Equation 6.8:

$$r = a_T (MP)^n \left(\frac{1.2}{D_h} \right)^{0.2} \quad (6.8)$$

where D_h is the chamber bore diameter of a centrally perforated cylindrical grain and M is the Mach Number of the centreline crossflow. This model also utilizes motor specific upper and lower threshold velocities which divides the propellant burning rate into three regimes: a base burning regime where

the simple burning law applies, a transitional regime and the erosive burning regime. The Saderholm model was modified by deriving a relation where the heat transfer coefficient was shown to be proportional to $du/dy/U$ where U is the centreline velocity of the flow through the chamber. By utilizing the velocity gradient at the propellant surface, Wang hoped to create a model that was effective for all sizes of rocket motors (the Saderholm model was found to be inaccurate for large solid rocket motors) and acknowledged that erosive burning was principally a phenomenon based on heat transfer from the crossflow to the solid propellant combustion zone. His erosive burning model for flow which exceeded a minimum threshold value for $\partial u/\partial y$ was:

$$\frac{r}{a_T P^n} = c_1 + c_2 \left(\frac{\partial u}{\partial y} \frac{1}{U} \right)^{c_3} \quad (6.9)$$

where the constants c_1 to c_3 were determined through curve fitting the empirical burn rate at a number of points along the grain. Using propellants which incorporated polybutadiene acrylonitrile (PBAN) or hydroxyl terminated polybutadiene (HTPB) as binders, Wang was able to obtain good agreement between empirical pressure measurements and the CFD simulations. In particular the pressure calculated using the CFD model were 10 – 20% higher than those calculated by the CFD model when erosive burning was not incorporated. The motors used by Wang for his work were comparatively low pressure with respect to the motor under consideration within this thesis (approximately 7 MPa). However, the effect of erosive burning was certainly

comparable to that obtained within this thesis for the simulation where $\beta = 120$.

It is also worth noting that the 1-D erosive burning models are still being used regularly by researchers modeling solid rocket motor combustion. Besides, Wang, Hasegawa et al. (2003) based their modeling of a solid propellant (aluminum/AP/HTPB) motor on a modified version of Dickinson's erosive burning law which incorporates the Lenoir-Robillard α and β constants. From these research efforts, as well as others, it is apparent that the use of 1-D erosive burning models are still valid for this type of simulation work and are also applicable to a rod-and-tube configuration as discussed in Chapter 4.

6.5.3 Sources of Error

While the trends observed in the results achieved during the non-steady-state modeling are consistent with accepted results and with the analytical model presented earlier in the chapter, it is not possible to validate the results of the simulations against experimental work incorporating similar chamber and nozzle geometry and/or propellant composition. Thus it is possible that the results presented within this document vary considerably from the actual performance that would result. While the basis of the CFD code, the Navier-Stokes equations are obviously considered sound, there are a number of possible sources of error that exist within the model as it currently exists. These sources of error are discussed in the following paragraphs.

6.5.3.1 Choice of Erosive Burning Constants

One of the largest potential sources of error is in the choice of the β erosive burning constant for which suitable published literature concerning GAP-ADN propellant is not available. This is not the case for the α constant which is determined using physical characteristic data which is available from the public domain. A sensitivity analysis where the β constant was varied from 53 to 120 clearly indicated the significant effect that this constant has on the predicted results as shown previously, as well as in Figure 6.27.

Figure 6.27: Aft end Static Pressure vs time curves comparing the effect of erosive burning for three different values of β against the simple burning law. Aft end value was measured using a point probe from a point located at $0.99 \bar{L}$ (normalized length).

It is only possible to obtain an accurate β erosive burning constant through experiment using methods described by Lawrence et al. (1968), Markland and Lake (1960), Hasegawa et al. (2003) since it is propellant dependent.

6.5.3.2 Boundary and Initial Conditions

The boundary conditions used for the inlets into the domain representing the chamber are based on CEA output for an Infinite Area Combustor (IAC). Given that the solved average static pressure and temperature values within the chamber are much higher than the values at the boundaries (and by extension higher than the IAC solution), it may be necessary to optimize the boundary conditions used. Optimized values could be provided by matching the calculated pressure-time curve with empirical data on a trial and error basis.

The effect of the initial T and P values upon the non-steady-state solution are significant as shown in Section 6.4.3. Determination of the optimum set of initial conditions can be made by comparing the simulated pressure-time curves to a set generated experimentally which highlight the pressure at both the head and aft end of the grain. Accurate modeling of the ignition transient to provide more accurate initial conditions for use with the non-steady-state simulation would improve the accuracy of the solution as well.

6.5.3.3 Mass Flow Rate Update

During the non-steady-state simulations, the mass flow rate was updated according to the protocol described at Table 5-1, although an updated value was calculated by the user subroutine for each time step. This inability to update the mass flow rate utilizing the output from a subroutine is a limitation of the software for non-steady-state simulations and has been addressed to the programmers of CFD-ACE+. This error appeared to be particularly significant during the rapid change in mass flow rates due to the rapid pressure rise during the early part of the simulation. Differences of up to 45% between the mass flow rate used and that calculated for a given time step were encountered, although the typical difference was in the order of 15%. The effect of manually updating the mass flow rate each time step was compared to the results generated using the mass flow rate update protocol for a portion of one non-steady-state run as shown in Figure 6.28. At the head end of the chamber, the peak pressure obtained was 11.8% higher and occurred 0.01 s earlier for the case where the mass flow rate was updated every time step. It is believed that by updating the $\dot{m}$ every time step, the overall rate of $\dot{m}$ increase or decrease would be greater, which would likely have a proportional effect on the rate of chamber static pressure increase or decrease as a function of time.

Figure 6.28: Static Pressure versus Time curves for non-steady-state simulation using $\beta = 60$ comparing the results achieved when manually updating the mass flow rate every time step and in accordance with the update protocol described in Table 5-1.

If the huge increase in computational time was not a factor, the mass flow rate would obviously be updated for each time step.

An additional factor to consider when determining the rate of mass flow rate changes is the finite length of time actually required for combustion of the solid propellant to occur as well as the time required for the increased heat flux from the crossflow to affect the burning rate at the propellant surface. Given the extremely small timesteps being used in the early stages of the simulation in particular (0.01 ms), the assumption of instantaneous combustion may be an incorrect one and necessitate the incorporation of a finite lag between changes to the static pressure and crossflow velocity and

their effect upon changes in the mass flow rate at the inlets of the model which represent the burning propellant surfaces.

6.5.3.4 Grid Issues

The use of multiple boundary surfaces to represent the grain surfaces was a simplification required to simulate the varying rates of grain regression along the length of the chamber. As mentioned in Chapter 5, this simplification was necessitated by the inability of the CFD application to regress at different rates at the node level. During the latter portion of a non-steady-state simulation, this resulted in a noticeable delta between the node locations at the interface of two surfaces. As a result of this delta the cells at the interface were highly skewed. Had the delta between the two nodes been too great, formation of a negative volume would occur resulting in an error exit from the simulation. Excessive skewness is a possible source of discretization error and increases the likelihood of the solution diverging. Unfortunately, given the necessity of utilizing multiple boundary surfaces, the occurrence of excessive skewness in some cells is unavoidable as shown in Figure 6.29.

As well, use of the grid deformation module incorporates an automatic remeshing feature to maintain the structured grid within all volumes whose boundary nodes are moving. A standard transfinite interpolation (TFI) scheme is used to determine the interior node distribution based on the motion of the boundary nodes [Anon (2001)].

Figure 6.29: View of grid at interface of two volumes close to aft end of computational domain representing the chamber following grid regression. The cells at the interface are excessively skewed due to the difference in grid regression rates of the two volumes.

In its simplest form, TFI interpolation blends two given curves to create a grid in the region between the curves. For two curves described by $c_i(r_i)$, the linear interpolation formula is

$$x(r_1, r_2) = c_0(r_1)(1 - r_2) + c_1(r_1)r_2 \quad (6.10)$$

where x is the intersection (i.e., node location) between the two curves [Anon (2003)].

The practical implication of this automatic remeshing using transfinite interpolation is that the volume which defines the converging section of the nozzle which is adjacent to the domain representing the aft end of the propellant grain and should be completely motionless, actually is not. The movement of the nozzle wall section was visible when the animation capability of CFD-ACE+ was utilized to view the simulation. This movement,

although slight (representing approximately 0.007 m), would have a small effect on the flow by changing the dimensions of the volume representing the converging section of the nozzle. This movement represented 4.1% of the overall axial length of volume representing the converging section of the nozzle.

6.5.3.5 Effect of Time Step

Because the CFD-ACE+ application uses an implicit pressure-correction type of linear interpolation scheme, there is no stability criterion (i.e., CFL number) that needs to be met in determining the size of the time step. However, it is still very important to set the size of the time step appropriately to carry out a transient simulation. The 'rule of thumb' followed in judging the size of the time step is to observe the number of iterations required to achieve convergence for a given time step. Convergence is assessed by the solution meeting a residual convergence criteria (a default value of 0.0001 was used for all simulations which is equivalent to a four-fold order of magnitude reduction in the residuals of all variables). Through trial and error, a suitable size time step protocol was established (minimum time step size was 0.01 ms, maximum time step size was 0.5 ms) and utilized consistently for each of the simulations carried out. It was necessary to start the non-steady-state simulations with the smallest time step due to the existence of a 'start-up' transient which decays rapidly [Anon (1998)].

As can be seen in Figure 6.30, the size of the time step did have a minimal effect on the results obtained. After 0.01s of simulation with the different time steps, the error in static pressure values obtained at the head and aft ends were 0.07% for the solution using a time step of 0.5 ms and 0.15% for the solution using a time step of 1.0 ms, both with respect to the solutions obtained with the time step of 0.1 ms. The error in the centreline crossflow values at the aft end was also compared and found to be 0.1% and 0.2% for the same cases, respectively. As can be seen in Figure 6.30, the error between the curves for the different time steps did not appear to be increasing, so it is likely that the use of the largest reasonable time step would allow an decrease in overall computing time required to carry out a simulation, providing convergence criteria were met. In the three cases evaluated, 400 iterations were required between each 0.1 ms time step, 900 iterations between each 0.5 ms time and 1200 iterations between each 1.0 ms time step so that convergence criteria were met.

Figure 6.30: Comparison Static Pressure versus Time curves for the interval 0.011 to 0.021 s using different size time steps. Identical initial conditions were used for the comparison of erosive burning law simulation results ($\beta = 60$) and the mass flow rate was updated each time step.

6.5.3.6 Threshold Velocity

In Chapter 2, the existence of a threshold velocity as quantified by Zucrow et al. (1964) was discussed. The threshold velocity, which was pressure dependent and separated burning between an erosive and a 'negative' erosive regime, was not incorporated in the simulations carried out. As the threshold effect was found to be particularly noticeable for propellants with a high normal burning rate, it may be a significant consideration when modeling a propellant such as GAP-ADN. Zucrow's threshold velocity formula (Equation 2.8) includes an empirical constant based on the propellant. Thus, further simulation work should consider the use of a threshold effect, but until experimental results on the GAP-ADN propellant are available, even the

incorporation of this effect may not reduce this source of modeling error due to the reliance on empirical data.

6.6 Summary of Key Findings

Prior to developing the CFD model of the motor under consideration, an analytical model was produced. The results obtained using this model, including crossflow velocity and mass flow rate, were referred to often throughout the course of this thesis. There was consistency between the analytical model results and those generated using the CFD model.

A steady-state CFD model of the motor under consideration was also developed. This model was used to determine the grid layout and density and to evaluate two differencing schemes (UDS and CDS) as well as to evaluate the different turbulence models available within CFD-ACE+. Solutions developed from the steady-state model included thrust, crossflow velocity and injection rate within the chamber as well as flow characteristics at the exterior of the nozzle. These latter results were found to have very good correlation with results obtained using the well accepted CEA application.

A number of non-steady-state simulations were run to evaluate the effect of varying the β erosive burning constant and initial conditions on the results obtained. All results were consistent with expected trends although it was not possible to compare the quantitative results against experimental data due to

the developmental nature of the motor and propellant under consideration. The observed effects of erosive burning within the CFD model included a significant increase in chamber pressure, particularly early in the simulation. A reduction in overall functioning time was also noted due to the increased mass flow rate when erosive burning was present. Given that the motor under consideration is designed to provide a neutral burn, the effects of erosive burning were clearly undesirable as the performance of the grain more closely resembled that of a regressive burning grain where the chamber pressure (and thrust) decayed continually throughout the burn.

Chapter 7 *Conclusions and Recommendations*

7.1 Conclusions

The overall aim of this thesis, to integrate a commercial CFD application which was capable of modeling grain regression with an erosive burning model to predict the time-dependent flow characteristics of a rod-and-tube configuration solid rocket motor, was successfully achieved. Although some sources of error have been identified (Chapter 6), this model appears to be suitable for wider application in addressing all significant physical and chemical phenomena which occur during the functioning of a solid rocket motor, and is applicable to use as part of a two- or three-dimensional simulation of the internal ballistics of solid rocket motors of varying grain configurations and erosive burning laws. Within the published literature, carrying out a complete non-steady-state CFD simulation has been reproduced only using much more powerful computational resources, such as CSAR [Alavilli et al. (2000)].

This thesis has made a substantial contribution in the field of CFD modeling in the following areas:

- a. The development of a methodology in simulating the internal ballistics of a solid rocket motor. While this methodology was demonstrated with the use of a one-dimensional erosive burning model [Lenoir and Robillard (1957a)] which was successfully incorporated into a non-steady-state two-dimensional CFD model, the applicability of the methodology is such that it supports the use of different erosive burning models, as well as non-steady-state three-dimensional CFD modeling.
- b. The inclusion of the erosive burning model in this non-steady-state CFD solution has shown merit due to the significant difference in peak static pressure between the case where erosive burning is incorporated and the case where burning is governed by the simple (or linear) burning law only. This characteristic of erosive burning has been noted in numerous instances within published literature and this trend is clearly reproduced within the non-steady-state modeling carried out within this thesis.
- c. Simulation of the rod-and-tube configuration grain motor which incorporates a plug nozzle is achievable using a two-dimensional CFD

model as the predicted thrust value results are in agreement with an accepted analytical solution for determining the thrust developed in a rocket incorporating a plug nozzle.

d. The crossflow velocities determined for the erosive and non-erosive burning cases were found to be quite similar. The rod-and-tube configuration used in this motor has a relatively small port diameter, particularly in the early stages of functioning, and this, coupled with the choked flow condition at the nozzle, appears to limit the crossflow velocity, measured within the chamber, to a threshold of approximately 620 to 630 m/s ($M \approx 0.57$).

e. The pressure and velocity peaks for both the erosive and simple burning cases are not apparent when solving the flow phenomena using a quasi-steady-state approach. The ability to carry out a non-steady-state simulation from the point where the complete grain surface area has been ignited until the 'all burnt' stage has been reached is invaluable to the rocket motor designer. In addition, the ability to characterize the nozzle inlet boundary conditions, the output from the modeling of the chamber flowfield, is very beneficial particularly if it is intended to incorporate a thrust vector control system within the rocket.

f. The influence of initial conditions upon the solution were examined and found to have a great impact upon the peak pressure and crossflow velocity results. However, the data for these parameters obtained during the mid to later stages of the simulation were similar indicating a consistency in the model as well as the importance of consideration of the ignition transient in the model.

7.2 Recommendations for Future Work

While the results from the modeling carried out thus far are consistent with the trends expected from a solid rocket motor which undergoes some erosive burning, a critical next step is to validate these results against empirical data. Two options are recommended:

- a. To construct and test a developmental motor based on the configuration modeled using the GAP-ADN propellant considered.
- b. To utilize existing combustion data on a known configuration rocket for which the propellant erosive burning constants are known, or can be developed using the test methods of Markland and Lake (1960), Lawrence (1968) or Hasegawa et al. (2003).

Key results anticipated from this work are to confirm the erosive burning constants used, and to verify the method of obtaining the boundary and initial conditions.

Other recommendations involve updating and/or improving the CFD model using the following approaches:

- a. Optimizing the mass flow rate update and time stepping procedures by increasing the frequency of mass flow rate update to each time step, while adjusting the time step and grid to minimize the computational time necessary to achieve a non-steady-state solution.
- b. The use of a 2-D axisymmetric representation of the rocket motor drastically limits modeling options to the simplest grain configurations. To model a more complex grain configuration (e.g., dendrite, dog bone) a 3-D simulation will be necessary. This would require significantly enhanced computing power to carry out a non-steady-state solution for the 3-D case.
- c. Manipulating the erosive burning law to take advantage of the capability within the CFD application to obtain the flow properties (i.e., velocity profile) near or at the propellant surface. For instance, following along the work of Wang (2003), it is possible to solve for the

heat transfer coefficient within the boundary layer by characterizing the flow near the propellant surface, rather than depending on bulk flow characteristics such as the total mass flux past the propellant, which is currently used in deriving the heat transfer coefficient within the Lenoir-Robillard erosive burning law. This may lead to an improved calculation of the burn rate enhancement due to heat transfer into the propellant which is the key erosive burning mechanism.

d. Incorporation of a threshold velocity criteria within the model. As the existence of a threshold velocity with respect to erosive burning is well documented (Chapter 2), the model developed thus far would be improved through the addition of this criteria. Of course, the subroutine developed for use with CFD-ACE+ model also permits the use of other erosive burning laws which may prove to be more accurate for the propellant and/or motor configuration under consideration. Use of the Saderholm burning law, which was found to be reasonably accurate in predicting the performance of smaller solid rocket motors, and which also incorporates threshold and transitional regions of erosive burning is recommended.

e. Better treatment of surface mass injection at the inlet boundary surfaces. The current approach to use a wall function appears reasonable given the magnitude of the injection velocity (considered to

be a weak injection rate based on the work of Ciofalo (1989)); however, the use of a low Re turbulence model or Large Eddy Simulation would facilitate the use of an updated surface mass injection treatment. Significantly enhanced computational resources would also be necessary to pursue this approach.

f. Investigation of the response of the combustion processes to changes in flow parameters. This finite rate response, which was assumed to be instantaneous for purposes of the modeling carried out thus far, may have a significant effect upon the modeling results because of the small time steps that are necessary to achieve a converged solution.

Bibliography

Alavilli, P.V.S., Tafti, D., Najjar, F., "The Development of an Advanced Solid – Rocket Flow Simulation Program ROCFLO", AIAA Paper 2000-0824, 2000.

Anon., "Fluent 5 – User Guide Volume 2", Fluent Incorporated, July 1998.

Anon., "User and Module Manuals for CFD-GEOM, CFD-ACE+ and CFD-VIEW Version 6.6", CFD Research Corporation, November 2001.

Anon., "TFI Mapping: Transfinite Interpolation",
<http://www.llnl.gov/casc/Overture/henshaw/documentation/mappingHTML/node528.html>, October 2003.

Buckmaster, J., Chen, M., Jackson, T.L., Massa, L., "Numerical Modeling of Three-Dimensional Heterogeneous Propellant Combustion", AIAA Paper 2002-0780, 2002.

Bulgakov, V.K., Karpov, A.I., "Numerical Studies of Solid Propellant Erosive Burning", Journal of Propulsion and Power, Vol. 9, No. 6, pp. 812-818, Nov-Dec 1993.

Burcat, A., "Third Millennium Ideal Gas and Condensed Phase Thermochemical Database for Combustion", Technion Aerospace Engineering Report 867, available online at <ftp://ftp.technion.ac.il/pub/supported/aetdd/thermodynamics>, 2002.

Chapman, A., "Heat Transfer – 4th Edition", MacMillan, pp.189-263, 1984.

Chapman, S., Cowling, T.G., "The Mathematical Theory of Non-uniform Gases; An Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases – 3rd Edition", Cambridge University Press, 1970.

Ciofalo, M., Collins, M.W., " $k - \varepsilon$ Predictions of Heat Transfer in Turbulent Recirculating Flows Using an Improved Wall Treatment", Numerical Heat Transfer, Part B, Volume 15, pp. 21-47, 1989.

Ciucci, A., Iaccarino, G., Moser, R., Najjar, F., Durbin, P., "Simulation of Rocket Motor Internal Flows with Turbulent Mass Injection", Center for Turbulence Research Proceedings of the Summer Program 1998, pp. 245 – 266, 1998.

Coats, D.E., Nickerson, G.R., Dang, A.L., Dunn, S.S., Kehtarnavaz, H., "Solid Performance Program (SPP)", AIAA Paper 87-1701, 1987.

DeLuca, L., Price, E.W., Summerfield, M., "Nonsteady Burning and Combustion Stability of Solid Propellants", AIAA Progress in Astronautics and Aeronautics Volume 143, 1992

Farinaccio, R., DRDC Valcartier, Personal communications, 2003.

Favini, B., Di Giacinto, M., Serraglia, F., "Modeling of Flowfield Features During Ignition Transients of Solid Rocket Motor", AIAA Paper 2002-3753, 2002.

Ferziger, J.H., Peric, M., "Computational Methods for Fluid Dynamics – 3rd Edition", Springer, pp. 21-400, 2002.

Gauthier, J.E.D., "Lecture Notes – 88.580 Gas Turbine Combustion", Carleton University, 2002.

Gerald, C.F., Wheatley, P.O., "Applied Numerical Analysis", Sixth Edition, Addison-Wesley, pp. 15-24, 377-379, 1999.

Gordon, S., McBride, B.J., "Computer Program for Calculation of Complex Chemical Compositions and Applications", NASA Reference Publication 1311, 1994.

Hagemann, G., Immich, H., Terhardt, M., "Flow Phenomena in Advanced Rocket Nozzles – The Plug Nozzle", AIAA Paper 98-3522, 1998.

Hasegawa, H., Tokudome, S., Hanzawa, M., Kohno, M., "Erosive Burning of Aluminized Composite Propellants: X-ray Absorption Measurement, Correlation and Application", AIAA Paper 2003-4812, 2003.

Hoffmann, K.A., Chiang, S.T., "Computational Fluid Dynamics, Volumes 1 to 3 - 4th Edition", Engineering Education System, 2000.

Ito, T., Fujii, K., Hayashi, A.K., "Computations of the Axisymmetric Plug Nozzle Flow Fields", AIAA Paper 99-3211, 1999

Jackson, T.L., Buckmaster, J., "Heterogeneous Propellant Combustion", AIAA Journal, Vol. 40, pp. 1122-1130, 2002

John, J.E.A., "Gas Dynamics", Allyn and Bacon, pp. 82-85, 1969.

King, M.K., "A Modification of the Composite Propellant Erosive Burning Model of Lenoir and Robillard", Combustion and Flame, Vol. 24, pp. 365-368, 1975.

King, M.K., "A Model of Erosive Burning of Composite Propellants", AIAA Paper 77-930, 1977.

King, M.K., "Erosive Burning of Experimental Solid Propellants: Experimental and Modelling Studies", AIAA Journal of Spacecraft and Rockets, Vol. 16, May-June 1979, pp 154-162, 1979.

Komai, I., Kobayashi, K., Kato, K., "Burning Rate Characteristics of Glycidyl Azide Polymer (GAP) Fuels and Propellants", AIAA Progress in Astronautics and Aeronautics Vol. 185, pp. 459-464, 2000.

Kruse, R.B., "Fundamentals of Solid Rocket Motors – Lecture Notes", presented at the University of Tennessee Space Institute, pp 41 – 45, November 9-13, 1992.

Kubota, N., "Survey of Rocket Propellants and Their Combustion Characteristics", Fundamentals of Solid Propellant Combustion – Volume 90 of the AIAA Progress in Astronautics and Aeronautics Series, Chapter 1, pp. 1-52, 1984.

Kuo, K., Summerfield, M., "Fundamentals of Solid-Propellant Combustion", AIAA Progress in Astronautics and Aeronautics Volume 90, 1984.

Lawrence, W.J., Matthews, D.R., Deverall, L.I., "The Experimental and Theoretical Comparison of the Erosive Burning Characteristics of Composite Propellants", AIAA Paper 68-531, 1968.

Lengellé, G., "Model Describing the Erosive Combustion and Velocity Response of Composite Propellants", AIAA Journal, Vol. 13, No.3, pp. 315 -322, March, 1975.

Lenoir, J.M. and Robillard, G., "A Mathematical Method to Predict the Effect of Erosive Burning in Solid Propellant Rockets", Sixth Symposium (International) of Combustion, Reinhold, New York, pp. 663-667, 1957a.

Lenoir, J.M. and Robillard, G., "The Development of a New Erosive Burning Law", Jet Propulsion Laboratory Publication No. 91, 1957b.

Lewis, G.N., Randall, M., "Thermodynamics – 2nd Edition", McGraw-Hill, pp. 667-669, 1961.

Liau, Y-C., Yang, V., Thynell, S.T., "Modeling of RDX/GAP Propellant Combustion with Detailed Chemical Kinetics", Solid Propellant Chemistry, Combustion, and Motor Interior Ballistics – Volume 185 of the AIAA Progress in Astronautics and Aeronautics Series, Chapter 2.11, pp. 477-500, 2000.

Lonsdale, R.D., "An Algebraic Multigrid Solver for the Navier-Stokes Equations on Unstructured Meshes", Int. Journal of Numerical Method for Heat and Fluid Flow, Vol. 3, pp. 3-14, 1993.

Marklund, T., and Lake, A., "Experimental Investigation of Propellant Erosion", ARS Journal, pp. 173-178, February 1960.

Majdalani, J. "Improved Flowfield Models in Rocket Motors and the Stokes Layer with Sidewall Injection", Ph.D. dissertation, Dept. of Mechanical Engineering, University of Utah, 1995.

Miller, E., "Erosive Burning of Composite Solid Propellants", Combustion and Flame, Vol. 10, pp. 330-336, 1966.

Mumar, M., Kuo, K.K., "Flame Spreading and Overall Ignition Transient", AIAA Progress in Astronautics and Aeronautics Series, Volume 90, Chapter 6, pp. 305-360, 1984.

Patankar, S.V., "Numerical Heat Transfer and Fluid Flow", Taylor and Francis, pp. 113-134, 1980.

Razdan, M.K., Kuo, K.K., "Erosive Burning Study of Composite Solid Propellants by Turbulent Boundary Layer Approach", AIAA Paper 78-978, 1978.

Razdan, M.K., Kuo, K.K., "Turbulent Flow Analysis of Erosive Burning of Cylindrical Composite Solid Propellants," AIAA Journal, Vol. 20, pp.122-128, 1982.

Saderholm, C.A., "A Characterization of Erosive Burning for Composite H-series Propellant", AIAA Paper 64, AIAA Solid Propellant Rocket Conference, Palo Alta, California, USA, 29-31 January 1964.

Schreier, S., "Compressible Flow", John Wiley and Sons, pp. 41-43, 1982.

Slaby, M., Radhakrishnan, S., CFD Research Corporation, Various Personal Communications, 2002-3.

Stowe, R., DRDC Valcartier, Various personal communications, 2002-3.

Sutton, G.P., Biblarz, O., "Rocket Propulsion Elements – Seventh Edition", John Wiley and Sons Inc., 2001.

Turns, S.R., "An Introduction to Combustion", McGraw-Hill, pp.11, 2000.

VanDoormaal, J.P, Raithby, G.D., "Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows", Numerical Heat Transfer, Vol. 7, pp.147-163, 1984.

Wang, Q., "Development of Erosive Burning Models for CFD Predictions of Solid Rocket Motor Internal Environment", AIAA Paper 2003-4809, 2003.

Warsi, Z.U.A., "Fluid Dynamics – Theoretical and Computational Approaches", CRC Press, pp. 407-567, 1993.

White, F.M., "Viscous Fluid Flow - 2nd Edition", McGraw-Hill Inc., pp. 33-35, 1991.

Wilcox, D.C., "Turbulence Modeling for CFD – 2nd Edition", DCW Industries, pp. 1-26, 103-217, 1998.

Yee, H.C., "Construction of Explicit and Implicit Symmetric TVD Schemes and their Applications", Journal of Computational Physics, Vol. 68, pp. 151-179, 1987.

Zucrow, M.J., Osborn, J.R., and Murphy, J.M., "Propellant Erosive Burning", Chemical Engineering Progress, Vol. 60, No.2, pp. 43-45, Feb 1964.

Appendix A – Selection of CFD Application

RECOMMENDATIONS ON SELECTION
OF A COMPUTATIONAL FLUID DYNAMICS
APPLICATION

by

John Weaver

163446

Table of Contents

Introduction

Selection Methodology

Selection Process

Conclusion

References

ANNEX A – Internet Literature Review

ANNEX B – CFD Application Short List

ANNEX C – CFX Supporting Documentation

ANNEX D – Fluent 6 Supporting Documentation

ANNEX E – CFD-ACE+ Supporting Documentation

ANNEX F – Computer System Requirements

Introduction

As the review of any Journal or publication that deals with Computational Fluid Dynamics (CFD) shows, there are a great number of computer applications, both commercially available and those purpose built for specific areas of research, that deal with modeling fluid flows. A great number of these codes are optimized to deal with turbomachinery or atmospheric movements as examples. Not so many of them can deal with complete simulations of the exhaust flow through a solid rocket motor due, in part, to the difficulties in modeling such complex fundamental processes as flame spreading, ignition transients and erosive burning. Thus, the request by DREV to identify a suitable commercially available computer application to model the internal flow of the motor under consideration¹ was not a straight forward task. In order to make a recommendation, a selection methodology based on modeling method and characteristics of solid rocket motor combustion was constructed. This methodology is first described, then applied to commercially available CFD applications.

Selection Methodology

Numerical Solution Techniques

Solution of fluid flow and heat transfer problems can be carried out by a number of different numerical techniques available. As an example, the panel method is often used to predict characteristics of an airfoil in an incompressible flow. In a general sense, this method utilizes the techniques of the Finite Element Method (FEM). From that perspective, there are three general methods that are used to solve fluid flow/heat transfer problems. They are the Finite Difference Method (FDM), the previously mentioned FEM and the Finite Volume Method (FVM).

The key element of the FDM is to represent the solution region as a number of discrete grid points, and then solve algebraic equations describing the flow field variables at each of the grid points.^{2 3} These grid points must be placed in a regular, structured grid and this is the primary disadvantage of the FDM. Fluid flow around complex geometries, such as an intricately formed turbine blades, do not lend themselves to solution by the FDM.

With the FEM, the solution region (i.e. the domain of an object with known properties) is built up of many small elements, which can be rectilinear or curved, to form a grid. This grid need not be structured; however, the elements must connect with neighbouring elements at nodes which lie on the element boundaries. At the level of each element, an approximating function is defined which describes the behaviour of the element. These elements are then assembled in a matrix, known as a global stiffness matrix, to which is applied boundary conditions. By solving an integral form of the PDE (known as the weak form) which governs the behaviour of the element, both constant and differential type boundary conditions can be applied to the element.⁴ For CFD applications, the FEM holds two key advantages over applications based on the FDM;

¹ DREV Technical Specification, "CFD Investigation of Internal Solid Rocket Motor Flows", dated 1 August 2001.

² V.K. Garg, "Applied Computational Fluid Dynamics", Marcel Dekker, Inc., p.37.

³ S.V. Patankar, "Numerical Heat Transfer and Fluid Flow", Taylor and Francis, p. 4.

⁴ Op Cit, Garg, p. 49.

they are the ability to simulate flow around complex structures and to utilize differential type boundary conditions.

In the FVM, the solution region is also built up of many small 3-D elements; however one or more elements are used to represent small control volumes over which mass, momentum and energy are conserved. In FVM, the term cell is used to define the elements making up the entire domain, or flow field. These cells, which can be of various shapes such as triangular or quadrilateral, are combined into structured or unstructured grids. The conservation laws are applied at discrete points in the cells known as nodes. Typically, the nodes are located at the cell centres, cell vertices or mid sides. However, by using a staggered grid approach, it is not necessary for the control volumes on which the laws of conservation are applied to correspond with the cells of the grid. This is a key difference between the FEM and the FVM and permits greater accuracy in representing the solution over the flow field. This is because the FVM permits the geometric flexibility of FEM (i.e. the ability to construct complex structures) but also allows flexibility in defining a discrete flow field, which is a strength of the FDM.⁵

Based on the characteristics of the three different methods, it is believed by the author that the FVM offers the greatest potential in accurately modeling the internal flow of the motor under consideration, and is a requirement for any of the commercial CFD codes which will be investigated in detail within this report.

Characteristics of Solid Rocket Motor Combustion

In their simplest form, the physical processes that occur during the combustion of a solid rocket motor are as follows:

a. Fluid Flow - The combustion of the solid propellant produces exhaust gases which flow through a nozzle at the rear of the rocket. While flowing through this nozzle, the high temperature, high pressure gases undergo an expansion to supersonic velocity from a sonic velocity at the throat of the nozzle.

b. Combustion – The propellant burn rate is a function of chamber pressure and in an ideal rocket, obeys the relation:

$$r = ap^n$$

where r is the burn rate, p is the chamber pressure, a is the temperature coefficient and n is the burning rate exponent.⁶ Note that a and n are propellant specific.

c. Structural response of the propellant and case – A solid rocket motor is structurally a thin shelled cylinder which is almost completely filled with propellant, a viscoelastic material. The failure modes of this configuration are numerous, but all have a common mechanism – increased propellant burning area which results in internal pressure beyond the limits of the case.

⁵ Ibid, Garg, p. 56-59.

⁶ G.P. Sutton and O. Biblarz, "Rocket Propulsion Elements, 7th Edition", John Wiley and Sons, p.428.

These three physical processes are coupled during the entire time solid rocket motor combustion is in progress. Of course, there are many other complex phenomena which occur during operation of the motor which have a significant effect upon its performance. These phenomena include:

- a. Erosive burning - the increase in propellant burning rate caused by the high velocity flow of combustion gases over the burning propellant surface.⁷ This phenomena is most prevalent near the nozzle.
- b. Ignition transient – the ignition of a solid rocket motor involves a comparatively small pyrotechnic device igniting and causes an efflux of hot gases onto the propellant surfaces. Once the temperature of the propellant is sufficiently high, the main solid rocket propellant begins to burn. This is a particularly violent phenomena which involves the chamber pressure going from essentially ambient to tens of MPa in less than one second, and is often the phenomena that causes many solid rockets to fail.

These complex phenomena have been the subjects of considerable investigation and are discussed in great detail in publications such as the AIAA Progress in Astronautics and Aeronautics Series, Volumes 90 and 143. However, simulation of the three basic physical processes which includes all of the complex phenomena that occur is a task that, until recently, was only within the scope of computer applications which relied upon empirical data to produce a realistic simulation.⁸

Center for Simulation of Advanced Rockets

Considerable success in modeling all aspects of solid rocket motor operation has been achieved recently by the Center for Simulation of Advanced Rockets which is based at the University of Illinois at Urbana-Champaign. The overall goal of the Center's efforts is to provide detailed numerical simulations of the Space Shuttle Solid Rocket Booster.⁹ Using three component codes; ROCFLO, ROC SOLID and ROCFACE, all of the physical processes attributed to solid rocket motor operation which were described above, have been simulated.

ROCFLO is a CFD code which utilizes a three dimensional, structured, finite volume, cell-centered approach. The Navier-Stokes equations are solved on a dynamic grid which features adaptive remeshing. The grid speed is based on structural deformation and combustion induced regression.¹⁰ ROC SOLID is a structural finite element analysis code which utilizes an unstructured mesh. ROCFACE is the interface code which manages the mesh association and data transfer tasks between other two programs.¹¹ While the scale of this solution is well beyond

⁷Ibid. Sutton, p. 433.

⁸ P. Allavilli, J. Buckmaster, T.L. Jackson, M. Short, "Ignition-Transient Modeling for Solid Propellant Rocket Motors", AIAA Paper No. 2000 – 3567, p.1.

⁹Ibid. Allavilli et al, p.1.

¹⁰ P. Allavilli, D. Tafti, F. Najjar, "The Development of an Advanced Solid-Rocket Flow Simulation Program ROCFLO", AIAA Paper No. 2000-0824, p.1-2.

¹¹ I.D. Parsons, P. Allavilli, A. Namazifard, X. Jiao, A. Acharya, "Fluid Structure Interaction through a Non-Material Interface: Simulations of Solid Rocket Motors", May 2000, p. 2.

that described by DREV at Reference 2 (ROCFLO subdivides the Space Shuttle's Solid Rocket Booster into 36 million nodes, and the application runs on multiple processors¹²) the approach followed by the Center for Simulation of Advanced Rockets does provide an excellent reference point to consider in identifying a suitable commercially available computer application to model the internal flow of the motor under consideration. Specifically, an application that is able to couple combustion of the solid rocket propellant, the fluid flow of the exhaust gases through the rocket, and the structural reaction of the propellant and casing, i.e. model the complete physics of solid rocket operation, is the optimum solution.

Thus the criteria for the recommending a commercial CFD application to support work on the motor under consideration are as follows:

- a. The application should utilize the FVM; and
- b. The application should be able to accurately couple and model all three physical processes described above. Use of an adaptive grid (moving mesh) to model the solid/gas interface within the solid rocket motor appears to be a promising technique and should be a feature of the chosen application.

Selection Process

Literature review

An extensive review of technical papers and publications revealed very little in terms of the use of commercial CFD applications in support of research. Many of the articles focused on the complex phenomena being studied and purpose built applications for studying them, but there was virtually no mention of commercial CFD applications being used to support this research. The search for detailed information on commercial CFD applications then was directed at information available on the Internet through which considerable detailed information was collected. Extensive background information on CFD was located at the Desktop Engineering Online (<http://deskeng.com/>) and CFD Review (<http://www.cfdreview.com/>) websites.

From this background it was possible to obtain information on a large number of commercial CFD applications. The listing of applications that were reviewed is found at Annex A to this report. From the results of this review a short list of CFD applications that appeared to be used routinely in support of solid rocket motor simulations was collected. This short list consisted of the following applications:

- Blue Ridge Numerics Inc. (BRNI) – CFDesign
- CFD-ACE+ (&CFD-Fastran) *
- CFX (& CFX-Tascflow) *
- FLUENT *
- Nika – FloWorks
- STAR-CD

* - Chosen as promising applications

¹² Op Cit. Allavilli et al, p.5.

Contact through telephone and e-mail to representatives of each of the companies that represented promising CFD applications was then carried out. Details concerning these contacts are included at Annex B. Three applications appeared promising following the initial correspondence with the short listed companies. These applications were then evaluated based on the selection criteria previously chosen. (Note: Hardware/software requirements to utilize the applications were also considered; however, cost was not. It is believed that the cost to obtain each of these applications is less than \$100,000 which is believed to be within the funding scope of the DREV program).

Evaluation of Promising CFD Applications

The side-by-side format table below compares the capabilities of the three CFD applications which appeared to be likely candidates for use on the program following the literature review. Additional background information on the applications is found in Annexes C, D and E.

Criteria	CFX	FLUENT	CFD-ACE+
Utilize FVM	Yes	Yes	Yes
Model of gas flow field	Yes, using CFX-Tascflow. Various turbulence models as well as heat transfer simulation capability available.	Yes. Various turbulence models as well as heat transfer simulation capability available.	Yes. Various turbulence models as well as heat transfer simulation capability available.
Model combustion at the propellant surface (moving mesh capability desirable)	No, CFX does not currently have surface reaction models, although this will likely occur in future releases of CFX 5. User routines in Fortran are available for moving mesh, but they must be user modified prior to use. ¹³	Yes, simple User Defined Functions are available from Fluent. These UDFs could be modified to account for multiple species. ¹⁴ Fluent has introduced a moving/deforming mesh capability available for beta testing. This feature was designed for in-cylinder (piston engine) flows. Fluent has expressed interest in	Yes, CFD-ACE+ currently has the capability to model surface reactions. The user can specify prescribed wall motion (which will account for the boundary change as the propellant is burning). CFD-ACE+ currently has a sliding mesh capability which is suitable for use with in-cylinder (piston engine) flows and which was released in 2001.

¹³ R. Steed, Personal communication, 3 Dec 2001.

¹⁴ R. LaRoche, Personal communication, 31 Dec 2001.

		providing assistance with this feature in making it more usable for other applications. ¹⁵	
Model structural dynamics of motor and casing	No.	May be possible by coupling Fluent to a structural dynamics code and importing/exporting deformation data utilizing the NASTRAN format. ¹⁴ (Note: NASTRAN is a finite element analysis application which can currently be used as a preprocessor with Fluent).	Yes. Using CFD-FEMSTRESS, an add-on module to CFD-ACE+, steady-state and transient deformations of the walls and propellant grain have been successfully modeled. Note that Dr. Greatrix of Ryerson Polytechnic is studying this particular issue in detail using CFD-ACE+/CFD-FEMSTRESS. ¹⁶
Computer requirements (see Annex F)	PC compatible, requires Network operating on Windows NT 4.0, Service Pack 5.	PC compatible, requires Network operating on Windows NT 4.0, Service Pack 5.	PC compatible, requires Network operating on Windows NT 4.0, Service Pack 3.
Miscellaneous	1. CFX Technical Services has a significant presence in Waterloo, Ontario. This organization has a very good knowledge of CFX-Tascflow and is involved in the development of this software. 2. Training for this software is available in Canada (Waterloo, Ontario).	1. Very open to working with customers (e.g. beta testing/modification to moving mesh capability was encouraged by Fluent personnel). 2. P. Allavilli, one of the authors of the papers dealing with ROCFLO, is a member of the Fluent development team.	1. This is the only application that has been directly referenced in technical papers concerning solid rocket motor combustion. Many examples of solid rocket motor simulations can be found on the CFDCanada (www.cfdcanada.com), and CFD-RC websites. 2. Technical support for the software is available in Canada through CFD Canada,

¹⁵ R. LaRoche, Personal communication, 11 Oct 2001.

¹⁶ V. Kudriavtsev, Personal communication, 7 Dec 2001.

			which has offices in Toronto and Montreal. 3. CFD-ACE+ is utilized by Thiokol, the world's largest producer of high-technology solid rocket motors for space, defence and commercial launch applications. ¹⁶
--	--	--	--

Based on the criteria selected, the CFD applications are ranked as follows:

1. CFD-ACE+, since all criteria can be met. As well, a user defined moving boundary capability exists, as does a sliding mesh capability for piston engine modeling.
2. Fluent 6 – since all criteria, less the structural dynamics of the motor and casing, can be met. As well, a user defined moving boundary capability exists, as does a moving mesh capability (beta) for piston engine modeling.
3. CFX – since the modeling of gas fluid flow criteria can be met.

Conclusion

While none of the commercial CFD applications available fully meet all of the desired capability, it appears that CFD-ACE+ would best meet the requirements of the program being considered. Although additional consultation with CFDRC would be necessary prior to the purchase of the application, it is believed that the CFD-ACE+ with the CFD-FEMSTRESS add-on module would provide an acceptable start point in modeling the solid rocket motor under consideration.

References:

1. V.K. Garg, "Applied Computational Fluid Dynamics", Marcel Dekker Inc., 1998.
2. DREV Technical Specification "CFD Investigation of Solid Rocket Motor Flows", dated 1 August 2001.
3. K.K. Kuo and M. Summerfield, "Fundamentals of Solid Propellant Combustion, Progress in Astronautics and Aeronautics – Volume 90", American Institute of Astronautics and Aeronautics, Inc., 1984.
4. P. Allivilli, J. Buckmaster, T.L. Jackson, M. Short, "Ignition Transient Modeling for Solid Propellant Rocket Motors", AIAA Paper No. 2000-3567, 2000.
5. J.C. French, D. E. Coats, "Automated 3-D Solid Rocket Combustion Stability Analysis", AIAA Paper No. 99-2797, 1999.
6. P. Allivilli, D. Tafti, F. Najjar, "The Development of an Advanced Solid Rocket Flow Simulation Program ROCFLO", AIAA Paper No. 2000-0824, 2000.
7. T.L. Jackson, J. Buckmaster, "Modeling of Heterogeneous Solid Propellant Flames", Presentation from University of Illinois at Urbana-Champaign website dated September 1999, <http://www.csar.uiuc.edu/>
8. M. Ostrander, V. Kudriavstev, "Preliminary Comments on Solid Rocket Motor Simulation", Paper from CFD Canada website, <http://www.cfdcanada.com/>
9. R. Steed (CFX Team Leader), Personal communication, December 2001.
10. R. LaRoche (FLUENT University Liaison), Personal communications, October – December 2001.
11. V. Kudriavstev (CFD Canada and CFDRC Distributor), Personal communication, December 2001.
12. N. Solanki (CFDRC Technical Marketing Engineer), Personal communication, December 2001.
13. D.R. Greatrix, V. Kudriavstev, "Modeling of Structural Vibration for Motor Chamber Internal Flow Studies (Draft)", December 2001.
14. S.V. Patankar, "Numerical Heat Transfer and Fluid Flow", Taylor and Francis, 1980.
15. G.P. Sutton, O. Biblarz, "Rocket Propulsion Elements, 7th Edition", John Wiley and Sons, 2001.

16. I.D. Parsons, P. Allavilli, A. Namazifard, X. Jiao, A. Acharya, "Fluid Structure Interaction through a Non-Material Interface: Simulations of Solid Rocket Motors", Paper taken from University of Illinois at Urbana-Champaign website dated May 2000, <http://www.csar.uiuc.edu/>

ANNEX A

Internet literature review on CFD Applications

<u>CFD Application</u>	<u>Website</u>	<u>Comments</u>
ADINA	www.adina.com	A finite element analysis (FEA) application which specializes in fluid flows with structural interactions. No mention of solid rocket propulsion.
Analytic and Computational Research Inc.	www.acri-us.com/	No mention of solid rocket propulsion, expertise appears to be mainly in ground water transport.
AMTEC-Inca	www.amtec.com	This is the company that produces TECPLOT. CFD product is an add-on to TECPLOT. No mention of solid rocket propulsion.
ANSYS/ICEM CFD	www.ansys.com http://icemcf.com/	ANSYS/Flotran is a value priced FEA application. No mention of solid rocket propulsion. ICEM CFD, a subsidiary of ANSYS Inc., makes a universal pre-processor used by many CFD solvers.
Blue Ridge Numerics Inc. (BRNI) - CFDesign	www.brni.com	Solid rocket propulsion mentioned in website. Short listed.
CFD-ACE+ (&CFD-Fastran)	www.cfdrc.com	Many references to solid rocket propulsion mentioned in website. Short listed.
CFX (& CFX-Tascflow)	www.software.aeat.com/cfx/	Solid rocket propulsion mentioned in website. Encouraging comments from CFX Team Leader. Short listed.
FLUENT	www.fluent.com	Solid rocket propulsion mentioned in website. Encouraging comments from Fluent University Liaison. Short listed.
NGB	www.ngbtech.com	Application designed for industrial type uses (e.g. power plant).
Nika	www.flowworks.com	Solid rocket propulsion mentioned in website. Short listed.
Numeca	www.numeca.com/	Principal uses of application appear to be turbomachinery. No mention of solid rocket propulsion usage.
STAR-CD	www.cd.co.uk/	Solid rocket propulsion mentioned in website. Short listed.

ANNEX B

CFD APPLICATION SHORT LIST

Based on the internet literature search summarized at Annex A, a 'short list' was compiled for all of the companies whose CFD application was mentioned as being used in solid rocket propulsion modeling and simulation. Based on the literature search previously completed, an e-mail was used to contact each of the companies to permit the companies an opportunity to describe their application's suitability for solid rocket propulsion modeling. The text of the e-mail (which was slightly altered for each of the companies) read as follows:

After reviewing some of the material on the CFX website, as well as some other documents, I was hoping I could get some information from you describing, in rough terms, how I could use CFX to do the solid rocket modeling that will be a major part of my thesis. I have not used a CFD application previously; however, I do have a good understanding of the mechanisms that make up solid rocket propellant combustion.

The application that I use for the modeling should have the capability to couple the following:

- model the fluid dynamics of the gas flow field within the core of the rocket and out the nozzle;
- model the combustion at the propellant surface
- model the structural dynamics of the propellant and casing

The most critical time while the solid rocket propellant is burning is shortly after ignition, although there are severe pressure fluctuations occurring throughout the burn. These pressure fluctuations are the result of non-steady burning, propellant response to pressure fluctuations (the burning rate is dependent upon the pressure), and vorticity generation (due to slag ejection, constriction in the nozzle, etc.)

The burning rate mechanism (erosive burning) has been modeled using two different approaches. They are a heat transfer model, where the total burning rate is the result of a normal burn component (pressure dependent) and an erosive burning component (flow velocity dependent). The second model involves an analysis of the Boundary layer, where the mass flow rate is a function of the flame temperature, the propellant surface temperature, the energy required to raise the temperature of the propellant, and the characteristic mass of the Ammonium Perchlorate oxidizer. From my reading, it is easier to model the burning rate using the heat transfer model, although this was prior to large scale use of CFD applications.

It has been suggested that it will be necessary for the CFD application to have a moving mesh capability which conforms to the evolving boundaries of the propellant. This movement would be governed by the coupling of the 3 elements that were mentioned above.

With this in mind, I'd be interested in your thoughts on how CFX (Tascflow) could model this problem. If you have any examples of work of this type being carried out using Tascflow, I'd really appreciate that as well.

Based on this e-mail, a number of responses were received. The responses from CFX, Fluent and CFDRC were quite positive and are compared in the main body of the report. As for the other three applications that were short listed:

- a. STAR-CD: Replied with the latest copy of the CD-Adapco newsletter in electronic format. It contained no information relating to solid rocket propulsion modeling.
- b. BRNI – CF Design: My e-mail was returned as 'undeliverable' even though it was sent twice to the address specified in the BRNI website.

- c. Nika – FloWorks: A German based company who suggested that a CFD application “such as Fluent” might be more suitable for the work I was describing.

ANNEX C

CFX – SUPPORTING DOCUMENTATION

Product Directory

[CFX Home](#)
[Products Home](#)
[CFX-4](#)
[CFX-5](#)
[CFX-TASCflow](#)
[CFX-BladeGen](#)
[CFX-ProMixus](#)
[CFX-Build](#)
[CFX-Radiation](#)
[CFX-Turbogrid](#)
[Contact CFX](#)

Welcome to the product directory where you will find information on all CFX software products from CFX-5, the most comprehensive and flexible CFD code available, to specialised tools such as CFX-ProMixus, a versatile easy-to-use program for mixer design.

You can browse through the catalog list below or select a product from the main menu.

If you want to take a look at how CFX products have been used to solve real-world problems, visit the case studies section of the site. If you require any assistance with product selection, or wish a sales engineer to contact you, please email us at cfx.info@aeat.com

CFX-4

The leading code for multiphase calculations

CFX-4 software is one of the most popular codes available for the prediction of the complex flows encountered in the process and chemical industries. It offers a unique blend of powerful tools including direct CAD access, highly automated geometry creation tools, sliding meshes, and advanced models for turbulence, combustion, radiation, and multiphase flows. These technologies enable you to easily simulate real industrial flow processes, from settling tank sedimentation to bubble-driven flow in mixing vessels, and from jet flame impingement to coal-fired combustion plant.

Applications areas include: , Aerospace, Automotive, Biomedical Engineering, Fire & Safety, Healthcare, HVAC, Marine, Metals, Oil, Gas & Petrochemical, Power Generation, Process

PFDReaction analysis of Reaction zone near blades of pitched blade turbine
Courtesy of Performance Fluid Dynamics (PFD) Limited

[\[More information on CFX-4\]](#)

Platforms: ☐ Unix ☐ NT

CFX-5

Accurate CFD solutions at minimum computational cost

CFX-5 delivers computational fluid dynamics (CFD) solutions with a difference. Ease of use, robustness and accuracy are ensured by CFX-5's impressive combination of:

- The most advanced solver technologies,
- Superb parallel efficiency,
- Excellent pre- and post-processing capabilities, and
- Industry relevant models.

You can create geometries directly within CFX-5's pre-processor, or import them from CAD packages in native format. CFX-5 then builds a complete mesh automatically, using menu-defined modelling options, boundary conditions and fluid properties. Pressure and momentum conservation equations are solved simultaneously in CFX-5, reducing the usual iterative process and CPU time dramatically - the larger the problem, the greater the performance gain.

Applications areas include: , Aerospace, Automotive, Biomedical Engineering, Fire & Safety, Healthcare, HVAC, Marine, Metals, Oil, Gas & Petrochemical, Power Generation, Process

Platforms: ☐ Unix ☐ NT

CFX mesh adaption improves the flow resolution in a slurry valve model. This shows speed contours and velocity vectors.
Courtesy of Alcoa of Australia Ltd.

[\[More Information on CFX-5\]](#)

CFX-BladeGen

The Complete Solution for Blade Design and Evaluation

CFX-BladeGen is a specialized easy-to-use three-dimensional software tool for the rapid design of rotating machinery components. It can be used for axial, mixed-flow and radial blades in applications such as pumps, compressors, fans, blowers, turbines, expanders, turbochargers and inducers.

CFX-BladeGen's Plus module provides a rapid viscous flow analysis capability through an embedded and fully automated computational fluid dynamics (CFD) solver. The turbomachinery-specific quantitative and qualitative results that it produces, permit pre-selection of component designs and reduce the number of iterations needed to develop optimum blade profiles.

Applications areas include: , Aerospace, Automotive, Marine, Power Generation, Process, Turbomachinery

CFX-BladeGen predicted streamlines through the stator stage of an axial turbine.

[\[More Information on CFX-BladeGen\]](#)

Platforms: ☐ NT

CFX-Build

CFX-Build integrates CAD and CFD, so that engineers can study fluid flow processes during CAD-stage product development. CFX-Build creates meshes using features such as:

Direct access to CAD databases from major CAD packages.
Powerful manipulation facilities such as composite trimmed surfaces to produce high quality meshes from CAD data.

Excellent geometry creation facilities.
Highly automated surface and volume meshing ensuring high quality of grids.

Applications areas include: , Aerospace, Automotive, Biomedical Engineering, Fire & Safety, Healthcare, HVAC, Marine, Metals, Oil, Gas & Petrochemical, Power Generation, Process, Turbomachinery

Platforms: ☐ Unix ☐ NT

CFX-Build geometry of a brake fluid reservoir that has been imported from Pro/Engineer.

[\[More Information on CFX-Build\]](#)

CFX-ProMixus

The mixing analysis tool

CFX-ProMixus is an interactive software tool that provides detailed three-dimensional analysis of mixing processes in batch reactors. Starting from an intuitive GUI, the user specifies reactor configurations from lists of tank, baffle and impeller templates. CFX-ProMixus then automatically performs a CFD simulation of the flow within that vessel. The output is presented in a gallery of graphics and tables which illustrate and quantify mixing performance.

Applications areas include: , Pharmaceutical, Process

Platforms: ☐ NT

High shear regions in dual-impeller mixing vessel

[\[More Information on CFX-ProMixus\]](#)

CFX-Radiation

Thermal radiation modelling for complex geometries

CFX-Radiation calculates radiative heat transfer in complex three-dimensional geometries for grey and non-grey systems. As part of the CFX-4 suite of software, CFX-Radiation can be automatically interfaced to CFX-4 to provide coupled flow and radiation simulations. CFX-Radiation has two calculation options: the Monte Carlo method (which simulates the physical interactions between photons and their environment) and the Discrete Transfer method which is a combination of the well-known flux type methods and Monte Carlo.

Note: Requires CFX-4.

Applications areas include: , Aerospace, Automotive, Biomedical Engineering, Fire & Safety, Healthcare, HVAC, Marine, Metals, Oil, Gas & Petrochemicals, Power Generation, Process

Platforms:

Unix

NT

Surface temperature for a simulation of a fire in the HGV shuttle.

[\[More information on CFX-Radiation\]](#)

CFX-TASCflow

The software of choice for mechanical engineering design and analysis.

CFX-TASCflow is the software of choice for mechanical engineering design and analysis. Its unique coupled, algebraic multigrid solver runs in parallel on any number or combination of UNIX or NT processors. Powerful features make it an ideal tool for the demanding turbomachinery sector - the generalized grid interface allows dissimilar, non-matching grids to be connected, while multiple frames of reference and sliding meshes permit time-averaged or full transient rotor/stator interactions to be modeled. Subsonic, transonic and supersonic flows can all be simulated, and advanced models for gaseous combustion, reacting particles, fire simulation and wet steam and refrigerants are available. CFX-TASCflow's advanced visualization and graphics tools display the critical information needed to maximize the understanding of results, and include a suite of functions for the direct interpretation of flows in rotating machinery.

Surface temperature distribution in the first stages of Space Shuttle turbine
Courtesy of Pratt & Whitney, Liquid Space Propulsion

[\[More Information on CFX-TASCflow\]](#)

Applications areas include: , Aerospace, Automotive,
Marine, Power Generation, Turbomachinery

Platforms: ☐ Unix ☐ NT

CFX-TurboGrid

Fast and efficient meshing for turbomachines

CFX-TurboGrid is a powerful and user-friendly
grid generation software tool for
turbomachinery applications. A selection of
pre-defined templates for most turbomachine
types permits the automatic creation of grid
topologies. Suitable templates are provided for
low- and high-solidity axial, radial, and
mixed-flow blades geometries, and for
multi-bladed (split) flow passages and for
blade tip clearance

Applications areas include: , Aerospace, Automotive,
Marine, Power Generation, Turbomachinery

Platforms: ☐ Unix ☐ NT

3D view of hub, shroud, and blade
of a steam stator created with
CFX-TurboGrid

[\[More Information
on CFX-TurboGrid\]](#)

[Community](#) | [News](#) | [Products](#) | [Industry](#) | [Consulting](#) | [Training](#) | [Seminars](#) | [I-Libraries](#)
[CFX Update](#) | [Jobs](#) | [Projects](#) | [Academic Program](#) | [Contact CFX](#)

© AEA Technology, 2001

Page Last Updated: March 19, 2001

[CFX Webmaster](#)

CFX-5 New Version 5.5!

CFX Home
Products Home

CFX-4

CFX-5

Home

What's New

Step-by-Step

Specification

Solutions

News

Info Request

CFX-TASCflow

CFX-ProMikus

CFX-BladeGen

CFX-Build

CFX-Radiation

CFX-Turbogrid

Contact CFX

Precision... Performance..., PROFIT.

CFX-5 is a fluid analysis software tool that combines CAD input, automatic meshing and a fast solution algorithm, improving ease of use and slashing CFD analysis time.

Pressure and momentum conservation equations are solved simultaneously in CFX-5, reducing the usual iterative process and CPU time dramatically - the larger the problem, the greater the performance gain.

With two-phase modelling capabilities and coupled- multigrid implementation, the latest version of CFX-5 is the first choice for applications such as:

- Liquid sprays in gas within asthma drug inhalers for example
- Two separating liquids such as found in oil water separators
- Gas bubbles in liquids as occurs in bubble columns
- Slurry type flows which are encountered in waste water treatment
- Catalyst particles in chemical reactors.

Inside CFX-5...

Pre-Processing

CFX-5 reduces the time required to set up CFD calculations. CFD has traditionally involved setting up the problem geometry and generating a suitable mesh.

For complex geometries, the meshing limitations have meant that users have had to invest a great deal of time in defining their problem. CFX-5 overcomes these difficulties by incorporating direct CAD import, powerful geometry creation tools, and an automatic hybrid element mesh generator with an adaptive mesh refinement

CFX-5 At-a-Glance

CFX-5 Parallel Efficiency

Parallel operation is available on any combination of single- or multiple-CPU or networked UNIX workstations and Windows NT machines, including mixed UNIX/Windows NT clusters. This provides the most powerful method for solving large-scale CFD problems. CFX-5 automatically decomposes the grid into pieces that fit onto the individual processors, and the flow calculation then proceeds in parallel. This allows large problems to be computed faster or cases to be tackled that exceed the capacity of individual computers.

Making the most of your investment

CFX-5 is fully supported by a comprehensive program of training courses. In addition, users can access the CFX community site for online help. For complex problems, additional consultation is available through the CFX

with an adaptive mesh refinement algorithm.

available through the [CFX consulting service](#).

Page last updated 15 January 2001

- [CFX-5 Pre-Processing technology](#)
- [How does adaptive grid generation work?](#) (Adobe Acrobat Document)

Coupled Multigrid Solver

CFX-5 outperforms other CFD applications by ensuring that ALL hydrodynamic equations are solved simultaneously. Our new multiphase capability makes CFX-5 the first choice for simulating complex industrial flows and provides substantial benefits by solving the fluid equations together. This ensures that the strong inter-phase coupling which is usually a major cause of concern in traditional CFD codes is resolved in a robust and efficient manner. The use of coupled algebraic multigrid to solve the coupled linear system gives a scalable algorithm in the sense that CPU time increases linearly with problem size rather than exponentially compared to traditional technology.

- [CFX-5 advanced solver technology](#)
- [Performance of CFX-5 multigrid solver technology](#) (Adobe Acrobat Document)

CFX-5 provides comprehensive post-processing facilities to visualize solutions using these features:

- Isosurfaces colored with any variable;
- Perspective views with hidden-line removal and light shading;
- Velocity vectors on arbitrary planes;
- Three-dimensional streamlines;
- Geometry colored according to any variable;
- Line graphing facility;
- Fully mouse driven, including mouse manipulation of the view

In addition CFX-5 allows animations to be recorded to an

MPEG file.

[Examples of CFX Postprocessing](#)

[CFX Home](#) | [Products Home](#) | [CFX-4](#) | [CFX-5](#) | [CFX-TASCflow](#) | [CFX-BladeGen](#)
[CFX-ProMixus](#) | [CFX-Build](#) | [CFX-Radiation](#) | [CFX-TurboGrid](#)

© AEA Technology, 2001

Page Last Updated: August 30, 2001

[CFX Webmaster](#)

CFX-5 New Version 5.5!

Technical Specifications

[CFX Home](#)

[Products Home](#)

[CFX-4](#)

[CFX-5](#)

[Home](#)

[What's New](#)

[Step-by-Step](#)

[Specification](#)

[Solutions](#)

[News](#)

[Info Request](#)

[CFX-TASCflow](#)

[CFX-ProMixus](#)

[CFX-BladeGen](#)

[CFX-Build](#)

[CFX-Radiation](#)

[CFX-Turbogrid](#)

[Contact CFX](#)

Capabilities

- Solution on any mixture of tetrahedral, hexahedral, prism and/or pyramid elements
- Incompressible / compressible - subsonic, transonic, supersonic
- Steady state / transient
- Laminar / turbulent
- Rotating or stationary frame of reference
- Turbulence models
 - Zero-equation turbulence
 - k-e, RNG k-e
 - Reynolds Stress Transport Model ("Second Moment Closure")
 - Scalable wall functions
- Heat transfer
 - None, isothermal, thermal energy or total energy
 - Viscous heating
 - Conjugate heat transfer
 - Natural convection
- Newtonian and non-Newtonian fluids
- Species transport
- Multi-component fluids
- Coupled Eulerian multi-phase flow model with drag laws:
 - Grace drag factor for bubbles
 - Schiller-Naumann drag model for particle
 - General user defined drag coefficient
- Linear and quadratic resistance models
- User-defined momentum and energy sources
- Solution-based mesh adaption

Pre-processor

- Full 3D CAD geometry creation
- Direct import from third-party CAD packages
- Integrated geometry, meshing and physics definition
- Flexible fluids editor, including common fluids library
- Expression Language for general user-defined functions
 - Fluid properties
 - Equation sources
 - Boundary condition profiles
 - Initial conditions
- Context sensitive on-line user documentation with hyperlinks

Boundary Conditions

- Inlet: Mass flow, velocity, static pressure, total pressure, supersonic
- Outlet: Mass flow, velocity, uniform pressure, average pressure, supersonic, degassing condition
- Opening ("mixed flow"): Velocity, static pressure, total/static pressure
- Wall: No slip, Slip, rough, moving, adiabatic, temperature specified, heat flux, heat transfer coefficient
- Symmetry
- Periodic Pair

- Thin surfaces

Mesh generation

- Fast and flexible unstructured multi-element mesh generation
- Simple control of mesh length scales
- Automated curvature sensitivity
- Wedge element inflation through the boundary layer
- Third-party mesh import

Platform support

- All popular UNIX workstations (Compaq, HP, SUN, SGI, IBM)
- Personal computers running Windows NT.
- Flow solver available for personal computers running Linux (beta port)

Numerics

- Conservative finite-element based control volume method
- Implicit, pressure-based algorithm for all flow speeds, incompressible to compressible
- Advection modelling
 - Upwind differencing scheme
 - 1st-2nd order blend factor
 - High Resolution bounded discretisation

Linear Solver

- Coupled solution of mass and momentum, including two-phase flows
- Algebraic Multigrid
- Incomplete upper/lower factorisation (ILU)-smoother
- Linear performance for even large numbers of nodes

Parallel computing

- Homogenous or heterogeneous networks of UNIX and/or Windows NT platforms
- Automated domain partitioning
- Scalable execution time over a large number of processors, with efficiencies exceeding 90% for large problems
- Scalable memory usage

Post processing

- Software and hardware-accelerated rendering -
- Unstructured region definition
 - Points
 - Polylines
 - Surfaces: slice and sample planes, cylinders, isosurfaces, boundary regions
 - Volumes: plane volume, isovolume, sphere volume
- Graphics objects
 - Shaded surfaces
 - Contours

- Vectors
 - Streamlines
- Legends
- Text annotation
- Mesh quality display
- Image file output to Postscript and bitmap formats
- Quantitative calculations
 - surface area
 - ranges
 - area, mass-flow and volumetric-flow weighted averages
- Local export of solution values
- User controlled units
- Animation
 - position and value sweeping
 - transient results
 - automated MPEG creation
- Results export to third party and custom formats

[CFX Home](#) | [Products Home](#) | [CFX-4](#) | [CFX-5](#) | [CFX-TASCflow](#) | [CFX-BladeGen](#)
[CFX-ProMixus](#) | [CFX-Build](#) | [CFX-Radiation](#) | [CFX-TurboGrid](#)

© AEA Technology, 2001

Page Last Updated: August 30, 2001

[CFX Webmaster](#)

CFX-TASCflow

Version 2.11
Now Available!

CFX Home
Products Home
CFX-4
CFX-5

CFX-TASCflow

Home
What's New
Step-by-Step
Specification
Requirements
CFX-Hexa
News
Info Request

CFX-ProMikus

CFX-BladeGen

CFX-Build

CFX-Radiation

CFX-Turbogrid

Contact CFX

CFX-TASCflow is the software of choice for mechanical engineering design and analysis. Its unique coupled, algebraic multigrid solver runs in parallel on any number or combination of UNIX or NT processors.

Powerful features make it an ideal tool for the demanding turbomachinery sector - the generalized grid interface allows dissimilar, non-matching grids to be connected, while multiple frames of reference and sliding meshes permit time-averaged or full transient rotor/stator interactions to be modeled.

Subsonic, transonic and supersonic flows can all be simulated, and advanced models for gaseous combustion, reacting particles, fire simulation and wet steam and refrigerants are available. CFX-TASCflow's advanced visualization and graphics tools display the critical information needed to maximize the understanding of results, and include a suite of functions for the direct interpretation of flows in rotating machinery.

Other facilities include:

- powerful features for subsonic, transonic, or supersonic flows.
- efficient parallel processing on any combination of single- or multiple-CPU or networked UNIX workstations and NT PC machines, including mixed UNIX/PC clusters.
- generalized grid interface to connect completely dissimilar, non-matching grids, regardless of how they were created.
- advanced models for reacting and combusting flows including gaseous, droplet, and coal combustion, reacting particle tracking, and fire simulation.
- special fluids analyses for wet steam and refrigerants, and other real gases.
- multiple frames of reference to efficiently model flows with rotor/stator interactions.

Complete CFD simulation of Whittle-type turbojet

CFX-TASCflow
At a Glance

The unique design of CFX-TASCflow provides unmatched robustness, accuracy and efficiency for multi-dimensional fluid flow and heat transfer modeling. CFX-TASCflow has a unique coupled, algebraic multigrid solver that provides exceptional robustness and reliability.

Streak lines showing the separation and recirculation off the suction side of the diffuser blades at the 59% flow condition. © Weir Pumps

CFX-TASCflow Advantages

- Advanced Visualization Tools
- Flame Front Model
- Advanced Combustion Models
- Special Fluids and Real Gas Models
- Specialized post- processing for rotating machinery
- Efficient Parallel Processing
- General Grid Interface
- Dispersed Multiphase Flows
- Simulation of Rotor/Stator Interactions

CFX-TASCflow Solutions

- Fluid flow simulation of pneumatic components
- Study of flow through cracks in power plant piping

Created with the help of CFX, a new high power, low speed Francis runner design for Norcan Hydraulic Turbine Inc. consisting of both original and new replacement components.

Courtesy Swiderski Engineering

Making the most of your investment

CFX-TASCflow is fully supported by a comprehensive program of training courses. In addition, users can access the [CFX community site](#) for online help. For complex problems, additional consultation is available through the [CFX consulting service](#).

In association with our working relationships

[CFX Home](#) | [Products Home](#) | [CFX-4](#) | [CFX-5](#) | [CFX-TASCflow](#) | [CFX-BladeGen](#)
[CFX-ProMixus](#) | [CFX-Build](#) | [CFX-Radiation](#) | [CFX-TurboGrid](#)

© AEA Technology, 2001

Page Last Updated: July 11, 2001

[CFX Webmaster](#)

CFX

Computational Fluid Dynamics Software & Services

CFX-TASCflow Version 2.11
Now Available!

CFX Home

Products Home

CFX-4

CFX-5

CFX-TASCflow

Home

What's New

Step-by-Step

Specification

Requirements

CFX-Hexa

News

Info Request

CFX-ProMixus

CFX-BladeGen

CFX-Build

CFX-Radiation

CFX-Turbogrid

Contact CFX

Whats New in CFX-TASCflow Version 2.11

CFX-TASCflow Version 2.11 is intended for general distribution on all computer platforms (SGI, HP, IBM, Sun, DEC Workstations and Personal Computers running Windows NT 4.0, Windows or Linux RedHat 7.0). Major new features include:

New launcher

A new launcher that has customizable pull down menus and icon bar so that users can add 3rd party supporting software easily. Users can directly access CFX-Bladegen (on NT), CFX-TurboGrid, CFX-Hexa as well as CFX-TASCflow from the same launcher providing better TurboMachinery Application work flow. Pre, post and full modes are now available in the TASCflow GUI and can be launched by clicking on the CFX arrow icons and CFX-TASCflow button centered below the directory selection widget, respectively.

New customizable Solver Monitor

A customizable Solver Monitor allows the user to include their own convergence criterion as well as add application specific macro quantities to be monitored during the simulation via user modifiable source code routines. Global flow imbalance summaries for solved equations are available in the solver monitor so that users can easily determine if global mass, momentum, energy and scalar quantity conservation is being met. The usual MAX residuals for all solved equations are also displayed in the solver monitor and plots to view can be selected from a pull down menu.

Improved Transient simulation

Transient simulation performance has been optimized through automatic adjustment of flow code control parameters as well as some discretization improvements. A factor of 2 to 8 times speed-up has been reached over typical 2.10 flow solver settings.

Consolidated flow code parameter GUI with direct access to documentation search function, reset initial and default settings, in-situ find for known parameters and a major category selection panel. Your parameter file will not be re-ordered and parameters will not be deleted by this GUI.

New Flamelet model

Flamelet model has been enhanced to accept user created flamelet libraries using CFX-RIF software from Advanced Combustion. CFX-RIF will be distributed on a separate CD in late August 2001.

Streaklines in the distributor and the runner of a Kaplan turbine.

Courtesy: EDF-CIH, France.

State-of-the-art Turbulence models

Including Large Eddy Simulation and Algebraic Stress Model

New physical monitoring points

Physical monitoring points can be specified using Cartesian coordinates in the PRM file. All solved variables are interpolated to the specified location and information is written to the RSO file for each time step which can be plotted versus time in the post-processor.

Profile boundary conditions

Profile boundary conditions have been generalized to include 1D and 2D Cartesian and cylindrical coordinate specification, as well as some equation support and the feature is accessible from the graphical user interface.

CFX-TASCflow Post-Processing Enhancements

A mirror plane function has been added to the viewer allowing the visualization of the complete geometry when geometrical symmetry has been employed to reduce the required computer resources to perform the simulation.

Streak balls have been added as an optional visualization tool in the post-processor. The balls can be sized and colored by different field variables allowing users to visualize the variation of several quantities simultaneously throughout the flow domain.

5 - 30 times faster post-processing time for flat planes and isotimic surfaces which also speeds up turbo specific post-processing. Approximately 5 times speed-up of the mass averaging post-processing function.

Render vectors, grid, fringes and contours on isotimic surfaces.

[\[Return to CFX-TASCflow home page\]](#)

[CFX Home](#) | [Products Home](#) | [CFX-4](#) | [CFX-5](#) | [CFX-TASCflow](#) | [CFX-BladeGen](#)
[CFX-ProMixus](#) | [CFX-Build](#) | [CFX-Radiation](#) | [CFX-TurboGrid](#)

© AEA Technology, 2001

Page Last Updated: August 17, 2001

[CFX Webmaster](#)

CFX-TASCflow

Version 2.11
Now Available!

Technical Specifications for Version 2.11

CFX Home

Products Home

CFX-4

CFX-5

CFX-TASCflow

Home

What's New

Step-by-Step

Specification

Requirements

CFX-Hexa

News

Info Request

CFX-ProMixus

CFX-BladeGen

CFX-Build

CFX-Radiation

CFX-Turbogrid

Contact CFX

Capabilities:

- Customizable Launcher : menu and icon bar
- Customizable Solver Monitor
- Single or double precision
- Optimized default transient simulation solver settings
- Incompressible
- Compressible - subsonic, transonic, supersonic
- Turbulence models
 - k-Omega, k-epsilon
 - Shear Stress Transport Model
 - Scalable wall functions
 - Two-Layer model
 - Kato-Lauder model
 - Renormalized Group (RNG)
 - Second Moment Closure Model
 - Large Eddy Simulation Model
- Steady/unsteady state
- Stationary and rotating frames of reference
- Simultaneous, multiple rotating frames of reference
 - Frozen Rotor interface
 - Stage interface
 - Transient interface
- Porous media
- Natural convection
- Heat transfer including conjugate heat transfer
- Radiative heat transfer
 - Gibbs radiation model
 - Surface-to-Surface radiation model
 - Finite Volume radiation model
- Viscous heating
- Moving Grid Capability
- Cavitation model
- Real gas models
- Species transport
- Chemical reaction models
 - Eddy dissipation model -

Discretisation:

- Pressure-based formulation for all flow speeds, incompressible to compressible
- Conservative finite element based control volume method
- Advection modelling
 - Upwind differencing scheme (UDS)
 - Mass weighted skew upstream differencing (MWS)
 - Linear profile skew upstream differencing (LPS)
 - Physical advection correction (PAC)

Solver:

- Coupled solution of mass and momentum
- Algebraic multigrid
- Incomplete upper/lower factorization (ILUO) smoother
- Linear performance even for large number of nodes

Processing of results:

- Command language
- Programmable
- Domain-based built-in functions (+, -, *, /; summation, average, trig)
- Variables
- Logic
- Macros
- Quantitative computations, including custom creation of new variables
- Large suite of graphical and quantitative functions for rotating machinery applications: meridional, blade-to-blade, area and mass

- multiple forward/reverse reaction with kinetic control
- Flamelet with pre-integrated library creation using Advanced Combustion RIF software
- Flamefront
- Emission models
 - NOx (thermal, prompt and fuel-derived)
 - Soot
- Lagrangian tracking model
- Reacting particles/droplets
- Multi-component fluids

Build Case:

- Assemble full turbo-machinery cases from a single flow passage with ease

Boundary and Initial Conditions:

- In stationary and rotating frames of reference
- Profiles : 1D , 2D cylindrical or Cartesian component specification
- Auto detect 1-1 periodicity
- Inlet: velocity, mass flow, total pressure, supersonic
- Outlet: local, constant and averaged pressure, mass flow, supersonic
- Walls: slip, no slip, wall function, adiabatic, isothermal, heat flux, and reflecting, sticking and eroding particles
- General and turbo-machinery specific flow field initialization

Grid:

- Non-orthogonal
- Curvilinear
- Multi-block grids
- General Grid Interface
- Degenerate surfaces
- Cyclic and periodic geometries
- Embedded grid refinement
- Import of 3rd party grids
- Block structured hexahedral grids

averages, m'- θ performance analysis

Visualization:

- Turbo-post processing 4 views including blade-to-blade and average quantity plotting
- Key-Frame Animation for MPEG creation
- Object instancing and transformations
- Mirror Plane
- Save and restore GUI state
- Pseudo batch operation of GUI
- X and OpenGL
- Three-dimensional
- Line and shaded contours
- Isosurfaces with rendering
- Vectors
- Clipping Plane
- Streak lines and balls
- Lighting models
- Hidden line
- Multiple viewer ports
- Zoom box, magnifying glass; recursive undo and redo of transformations
- Image file and screen capture output to Postscript, BMP, PNG, PPM, JEG and VRML

Parallel platforms:

- Homogeneous or heterogeneous networks of UNIX and/or NT computers
- Single processor or multi-processor UNIX and/or NT computers

Computer environment:

- CFX-TASCflow is used on all leading computers including DEC, HP, IBM, SGI, SUN, Linux and personal computers running Microsoft Windows

[\[Return to CFX-TASCflow home page\]](#)

[CFX Home](#) | [Products Home](#) | [CFX-4](#) | [CFX-5](#) | [CFX-TASCflow](#) | [CFX-BladeGen](#)
[CFX-ProMixus](#) | [CFX-Build](#) | [CFX-Radiation](#) | [CFX-TurboGrid](#)

© AEA Technology, 2001

Page Last Updated: July 11, 2001

[CFX Webmaster](#)

ANNEX D

FLUENT 6 – SUPPORTING DOCUMENTATION

Fluent : About Us : News : Press Releases : PR28

FLUENT® 6.0 is Released, Fluent Inc.'s Flagship CFD Software's Newest Version is Tuned to the Challenges of the Aerospace Industry

December 17, 2001, Lebanon, New Hampshire...Fluent Inc., world's leading computational fluid dynamics (CFD) software provider, announces the release of FLUENT 6.0. This major new offering from Fluent contains specific features geared to the accurate simulation of common aerospace engineering challenges, such as flow-induced noise, engine cooling, anti-icing, and turbine engine modeling. Major leaps forward in overall ease-of-use put this CFD code in a class by itself in its ability to work efficiently and integrate with the rest of the design and analysis desktop.

This release underwent extensive testing on thorny industrial-strength engineering problems to validate its out-of-the-box applicability to the problems of many industries, including aerospace. "We were able to test the non-reflecting boundary conditions and partially premixed combustion model - both new to FLUENT 6.0. Overall, we felt that this new version of the code will address many issues which are important to turbomachinery design," says Dr. Peter Flohr of ALSTOM Power Ltd., Switzerland. "Additionally, the new features in turbulence modeling are the result of the remarkable technical advances to FLUENT 6.0," says Rolls-Royce, Germany.

Integration Means Lower Cost-of-Ownership

CFD is a useful tool in its own right, but when it is well-integrated with the design process, its ability to save design costs and reduce design time is greatly enhanced. The software allows for nonconformal walls and periodics, which make it possible to easily model zones where heat is transferred between solids and fluids, such as in rocket engines and cooling systems for a variety of devices including rocket engine nozzles and turbine blades. This feature is also applicable to the modeling of anti-icing systems for aircraft, and simplifies the meshing process overall for faster turnaround time. Improvements to FLUENT 6.0's modeling tool, GAMBIT® 2.0, make for robust geometry

About Us

- Company Profile
- Worldwide Offices
- Strategic Alliances
- Jobs
- News

Documents

LES Example: GAFA LM6000 Combustor

The flow field around a spinning football in flight.

Editor's Note

Fluent Inc. grants permission for use of the enclosed graphics and their captions in conjunction with any or all of this press release.

transfer from CAD packages.

(Download)

From UAV's to Gas Turbine Combustion

Improvements in many of the physical models in FLUENT 6.0 enable the user to model certain geometries and phenomena in greater detail and with more accuracy. A feature for modeling flow-induced noise lets engineers consider the effects of airframe noise, like that produced by landing gear. New models for looking at sprays and detailed chemical kinetics allow for more accurate study of nonequilibrium phenomena in gas turbine combustion. Enhancements to turbulence models have applicability to study of the aerodynamics of unmanned aerial vehicles (UAV's) and enable more accurate calculation of the losses in gas turbine engines due to the clearance between the blade and the shroud.

With respect to turbomachinery modeling, FLUENT 6.0 has made a major step forward. Its unstructured technology offers unparalleled geometric fidelity for modeling rotating machinery, and its versatility makes it possible for the engineer to include detailed turbomachinery results in a complete system analysis, which also considers combustion, supersonic flows, and other phenomena. Turbomachinery-specific panels let the user easily calculate key integral quantities and examine meridional, pitchwise and spanwise cuts. A specialized turbomachinery preprocessing tool, G/Turbo™, is also available from Fluent for doing fast, automated turbomachinery model set-up.

Fast

FLUENT's ability to scale well for parallel processing has been head and shoulders above other general purpose CFD software, and this position is further advanced with the new version, where automatic partitioning, dynamic load balancing and other advanced features enable users to easily adapt FLUENT to a wide variety of multiprocessor configurations. FLUENT 6.0 offers a new set of options for postprocessing and visualization, such as automatic generation of animations, and results can be output in VRML format, which enables the user to send three-dimensional images of results across the Web for remote viewing, where the receiver can change views, zoom in/out, and rotate the image.

Pricing and Availability

FLUENT 6.0 is available on UNIX, NT, and Linux platforms. It is provided at no additional cost to all current FLUENT users who have chosen to purchase upgrades along with their FLUENT license. FLUENT license fees are determined by the number of processes

that a user requires. Users may also purchase unlimited usage support.

About Fluent

Fluent is the world's largest provider of computational fluid dynamics (CFD) software and consulting services. Fluent's software is used for simulation, visualization, and prediction of fluid flow, heat and mass transfer, and chemical reactions. It is a vital part of the computer-aided engineering (CAE) process for companies around the world and in almost every manufacturing industry. Using Fluent's software, product development, design and research engineers build virtual prototypes and simulate the performance of proposed and existing designs, which helps them to optimize, troubleshoot, scale-up, and retrofit. Use of CFD software reduces time-to-market by reducing the need for costly physical testing and prototyping.

Fluent's corporate headquarters are located in Lebanon, New Hampshire, USA. Fluent's European headquarters are located in Sheffield, England, with local offices in Belgium, France, Germany, Italy, and Sweden. Fluent's Asia Pacific headquarters are located in Tokyo, Japan. Its CFD software is also available around the world through joint ventures, partnerships, and distributors in Korea, Australia, Brazil, China, Taiwan, the Czech Republic, Middle East, and most European countries.

Fluent is a subsidiary of Aavid Thermal Technologies Inc. Additional information on Fluent's products can be obtained on the World Wide Web at www.fluent.com or by e-mailing info@fluent.com. More information on Aavid Thermal Technologies is available at www.aavid.com.

FLUENT® and GAMBIT® are registered trademarks of Fluent Inc.
G/Turbo™ is a trademark of Fluent Inc.

For more information, contact:

Paul Whalen
Fluent Inc.
10 Cavendish Court
Lebanon, NH 03766
(603) 643-2600 Ext. 666
pww@fluent.com

[Home](#) | [Solutions](#) | [Software](#) | [Consulting](#) | [Support](#) | [About Us](#) | [Worldwide](#) | [Contact](#) | [Search](#)

All Rights Reserved © 2001 – Fluent Inc.

Reacting Flows

Targeted applications include gas turbine combustion, pulverized coal combustion, laminar flames, and surface reactions/stiff chemistry in chemical vapor deposition reactors.

The eddy dissipation concept model, widely known as the EDC model, has been implemented for the modeling of turbulent finite rate chemistry. It is completely different from the eddy dissipation model that has long been available in FLUENT. Based on the assumption of small turbulent structures, or fine scales in the turbulence field, the EDC model allows users to better account for complex, stiff chemistry through the use of a robust algorithm.

*This simulation of a gas turbine combustor shows the LES turbulence model used in conjunction with the (Norbert) Peters Flame Speed Model.
Courtesy of General Electric*

Several other models have been added or improved in FLUENT 6. These include a new partially premixed combustion model and the laminar flamelet model, which has been extended to include non-adiabatic effects such as radiation, discrete phase sources, and enthalpy corrections to temperature and density. The coupled implicit solver has been enhanced to better handle chemical kinetics in laminar flows. All combustion models in FLUENT 6 are now compatible with LES. In addition, chemical mechanisms can now be imported into FLUENT in the widely used CHEMKIN format. The

Software

Preprocessors

— Select —

General-Purpose

FLUENT

Release News

Simulation Examples

Benchmarks

Requirements:

Unix/Linux

Windows

Platforms:

FLUENT 6

FLUENT 4

Application-Focused

— Select —

Educational Software

— Select —

Technology Leadership

University Program

.....

NOx model now allows the use of the two-variable PDF model to account for turbulence-chemistry interaction even if the PDF model was not used for the main combustion calculation. For coal combustion, the chemical percolation devolatilization (CPD) model has been implemented. Predictions of off-gas amounts and composition using this mechanistic approach have correlated well with measured data for a wide variety of coals. In the area of surface reactions, significant improvements in convergence for low-pressure chemical deposition problems and stiff surface chemistry are now in place. Additionally, surface species can be selected as both reactants and products in surface reactions, allowing for bulk etching to be modeled.

[Home](#) | [Solutions](#) | [Software](#) | [Consulting](#) | [Support](#) | [About Us](#) | [Worldwide](#) | [Contact](#) | [Search](#)

All Rights Reserved © 2001 — Fluent Inc.

Moving & Deforming Meshes

Targeted applications include automotive in-cylinder flows and simple moving and deforming mesh problems.

A completely new capability that will be available for beta testing in FLUENT 6.0 is the ability to model moving/deforming meshes for in-cylinder flows. The novel approach taken in FLUENT requires the user to specify only the initial mesh and the boundary condition for the moving wall. Subsequent changes in the mesh are automatically generated by the solver.

3D engine cycle showing exterior mesh motion

Software

Preprocessors

General-Purpose

Release News

Simulation Examples

Benchmarks

Requirements:

Unix/Linux

Windows

Platforms:

FLUENT 6

FLUENT 4

Application-Focused

Educational Software

Technology Leadership

University Program

.....

Fluent : Solutions : Example X4

Missile Systems

Missile aerodynamics and thrust systems can be modeled quickly and accurately using Fluent's full-featured software. With robust numerical methods for transonic flows, and the ability to model flows that include both low-speed and high-speed regions, Fluent's software can predict the performance of thrusters, control surfaces, nozzles, and solid or liquid rocket motors. Plume analysis, including flow details required for infrared signature prediction, can also be modeled.

Mass fraction of exhaust gases and prediction of Mach number during confined launch of a rocket.

Solutions

- What is CFD?
- Industries

University Program

- Other Web Resources

.....

Examples/Articles

Submit Your Application

Search

[Home](#) | [Solutions](#) | [Software](#) | [Consulting](#) | [Support](#) | [About Us](#) | [Worldwide](#) | [Contact](#) | [Search](#)

All Rights Reserved © 2001 — Fluent Inc.

Solutions To Your Toughest Problems

FLUENT's multiphase flow capabilities let you tackle complex flow regimes, including bubbly, droplet, particle-laden, slug, annular, free-surface and granular flows. With a full range of models, including Lagrangian Dispersed Phase, Eulerian-Eulerian, Eulerian-Granular, Volume of Fluid, and Algebraic Slip Mixture, FLUENT provides the best-available closures for risers, fluidized beds, packed beds, bubble columns, and other multiphase processes.

FLUENT's technology ensures a fast and robust solution process. Solver speed is driven by a suite of

efficient solver algorithms, each tuned to specific problem physics: advanced coupled solvers for compressible and incompressible flows, fully-coupled implicit methods for multiphase flows, and robust segregated solvers for your toughest numerical tasks.

Particle trajectories in a typical Stairmand gas cyclone

Mesh quality can be the primary factor in CFD solution accuracy and FLUENT lets you improve your mesh as part of the solution process. Examine your preliminary solution and use FLUENT's tools for solution-based mesh adaptation to get a more accurate result.

Parallel processing is the key to fast turnaround of CFD calculations. FLUENT leads the way in parallel computing, with all the interactive tools you need and high parallel efficiency on dedicated multiprocessors or networks of workstations. On UNIX or NT, our computing technology brings your solution home fast.

Pressure distribution in a water pump

Partitions for parallel computing on a fighter aircraft

ANNEX E

CFD-ACE+ SUPPORTING DOCUMENTATION

CFD-ACE 6.6 (Multiphysics Solver)

New Features

- Fuel Cell Modeling Capability
- Improved Free Surface Flows/VOF for a faster run
- Powerful parallel solver for Windows NT/2000
- Improved solvers for Stress module, including a sticktion model
- Improved User Subroutine processing
- Anisotropic and Piezoelectric materials stress modeling
- Improved porous media modeling
- Boltzman solver for Plasma
- Improved Spray Modeling including Dielectrophoresis

Download

CFD-ACE 6.4

New Features

- Support for Windows 2000
- Reaction Managers with cut and paste capabilities
- Powerful new parallel AMG solver, which runs at twice the speed using only half the memory
- Parallel PLASMA, Electrochemistry, Electric, Magnetic, and Scalar modules
- Solving for AC conduction in E-Physics module
- Combined AC + DC in the frequency and time domain
- Ability to run more than one copy of CFD-GUI on Windows platforms
- CHEMKIN interface now available for unstructured grids
- Support for new Simulation Manager
 - Invocation of CFD-GEOM templates
 - Simultaneous control of both geometric and simulation parameters
 - Ability to run flow solver and access output data after the run
 - Line plotter for parametric and optimization studies
 - One-dimensional optimization
 - Customizable GUI builder

CFD-ACE 6.2

New Features

- PLASMA module with new, improved GUI
- VOF module with new, improved GUI
- Spray model extended for all grid types and solvers
- Enhanced surface chemistry with surface absorbed species
- Modal Analysis added to stress module
- Completely new and revised
- ElectroChemistry Module
- 2 Fluid Model
- Many more new features, please see the Release Notes for details

CFD-ACE 6.0

Completely new GUI to support fully coupled

multi-disciplinary analysis on structured and arbitrary polyhedral unstructured meshes

New Features

- Improved Low and High Density **Plasma Physics** and Plasma Chemistry Module (ICP & CCP)
- **Electroplating Module**
- **Full Cavitation Model**
- **Fluid-Thermal-Structure-Electrophysics Simulations**
- **FEA Stress Solver**
- **Ice Melting Module**
- Parallel Processing
- Native Windows NT Version, also available for Unix and Linux
- Property Manager (more than just a database)
- Reaction Manager (enter, modify, store and reuse surface/gas phase kinetics)
- Model Wizard Setup

More New Features

Brochure (PDF)

Highlights

Application examples

(click on any image to view it in larger size)

Unparalleled Flexibility for Model Setup

- Mesh Complex Multi-component Geometries easily with Polyhedral & Mixed-Element Unstructured Meshes (Static, Sliding or Moving) with **Arbitrary Cell Face Matching at Domain Interfaces**
- Enhanced Solution Fidelity and Speed Through **Directional Mesh Adaption** with User-Defined Criteria

Unparalleled Flexibility for Multi-Physics

- **Non-gray Radiation**
- **Extensive Combustion/Chemical Reaction Models** Using Multi-Step, Finite Rate **Stiff Chemistry**
- **Advanced Surface Chemistry for CVD Applications**
- Interface to **Surface Chemkin III** for Extended Surface Chemistry
- **Multi-Component Diffusion**
- **Basic and Extended Mechanisms for NOx and CO.**
- **Volume of Fluid (VOF) Free Surface Modeling** with

In-cylinder Flow in an Automotive Engine with Moving Valves and Piston

Conservative and implicit Treatment of Arbitrary Cell Face Matching Ensures Highest Accuracy

Sliding Mesh with Arbitrary Cell Face Matching

Mixed Element Mesh with Arbitrary Cell Face Matching

Surface Tension

- **Concentrated and Distributed Flow Resistances**
(Non-Isotropic Porous Media, Perforated Plates)

Instantaneous Surface Pressure Distribution on Gerotor Oil Pump

Solution Adapted Mesh of a Vortex Behind a Baffle

Temperature Distribution in Automotive Cylinder-Block Cooling Jacket

CFD-ACE Features

Version 6.0 Additional New Features

- Simplified File I/O
- Improved Combustion Chemistry
- Advanced Initial Condition/Boundary Conditions Available
- More Efficient FEMSTRESS Solver
- Improved Free Surface Model with Surface Tension
- User Subroutines Available
- Spray Model
- Advanced Volume Conditions
- User Defined Scalar also for unstructured grids
- Easy Access to Boundary/Volume Conditions
- Improved Radiation Inputs
- Improved Reaction Inputs
- Revised Residual/Monitor Plotter
- Advanced Initial Conditions/Boundary Conditions Available

GEOMETRY AND MESH

CHEMICAL REACTION AND

CFD-ACE is fully integrated with **CFD-GEOM** for creation of geometry, mesh, boundary conditions and import from other CAD/CAE systems.

The geometry capabilities include:

- **Arbitrary Polyhedral (Unstructured) Cell Meshes** for Ease and Flexibility in Model Setup.
- **Arbitrary Cell Face Matching at Domain Interfaces** for Added Flexibility and Ease in Model Setup
- **Directional Solutions Adaption** through CFD-Adapter for Enhanced Solution Fidelity and Speed
- **Sliding Mesh Interface with Arbitrary Cell-Face Match**
- Rotating Coordinates, Moving Meshes and Walls
- Internal Solid Objects, Concentrated and Distributed Resistances (Solid and Perforated Plates, Blockages, Porous Media)

FLUID FLOW

- A full range of flow conditions is handled by CFD-ACE, including:
- Incompressible/compressible flows, to supersonic
- Steady/transient
- Laminar flow with a choice of Non-Newtonian Rheology
- Turbulent flow, with a variety of turbulence
- **Volume of Fluid (VOF) Free Surface Modeling with Surface Tension**
- Profile and time varying boundary conditions

HEAT AND MASS TRANSFER

Particular attention has been paid to the energy equation, ensuring fully conservative treatment under a wide range of conditions:

- Natural and forced convection

COMBUSTION

CFDRC has extensive experience in combustion, providing a comprehensive capability in CFD-ACE:

- Combustion models include:
 - Instantaneous reaction
 - Multi-step, finite rate, **stiff** chemistry
 - Equilibrium chemistry
 - PDF models
 - Heterogeneous reactions
- **Advanced Monte-Carlo PDF Model** (add-on option)
- **Basic and extended NO_x and CO emissions mechanisms**
- **Advanced Surface Chemistry** for CVD and PVT processes. Interface to SURFACE CHEMKIN III (add-on option)

NUMERICS

CFD-ACE blends state-of-the-art numerical techniques to maximize solution robustness, accuracy and efficiency.

- Finite volume, pressure based, strongly conservative formulation
- **Fully implicit treatment of Arbitrary Interfaces and Hanging Nodes** (No Interpolation) for Solution Accuracy
- Differencing schemes (Upwind, Central, Second Order Upwind)
- SIMPLEC/PISO algorithms
- Solvers - Conjugate Gradient (CGS) and Algebraic Multi-Grid (AMG)

HARDWARE PLATFORMS

- Windows NT (Intel, Alpha)
- LINUX

- Multi-media, conjugate heat transfer
- **Non-gray Radiation Heat Transfer** (Discrete Ordinate Method)
- **Surface to Surface Radiation**
- Mass transfer with multi-component diffusion, Soret terms, and Stefan-Maxwell option.
- Specified temperature, heat-flux, heat-transfer coefficient and adiabatic thermal boundary conditions.
- **Cavitation**

- Unix Workstations (SGI, IBM, HP, SUN, DEC)

ADD ON MODULES

CFD-ACE is Fully Coupled with:

- **CFD-Plasma** for Plasma Chemistry and Plasma Enhanced Deposition/Etching
- **CFD-FEMSTRESS** for Coupled Fluid-Structural/Thermal Interaction
- **CFD-FILM** for Thin Film Morphology and Surface Profile Simulation
- **CFD-PARTICLE** for Particulate Generation Growth and Transport Simulations
- **CHEMKIN III** for extended chemistry
- **Monte-Carlo PDF Model**

CFD-ACE+

CFD-ACE is available along with CFD-GEOM and CFD-VIEW as the CFD-ACE+ package. CFD-ACE+ is coupled with other CFDRC solvers for multidisciplinary analysis.

Top of the page

CFD-ACE is a trademark of CFD Research Corporation. All rights reserved.

CFD-FASTRAN 4.2

Compressible Flow Solver With Moving Multi-Body Dynamics

NEW New Features in FASTRAN Version 4.2:

- **New Distributed Parallel Solver.** The new parallel solver technology uses sockets/TCP/IP technology for inter-process communication and data exchange. The new approach replaces PVM used in the previous parallel solver.
- **GUI panels for creating fastran.hosts file.** The GUI panel for creating the fastran.hosts file has been modified to suit the new parallel solver. If the user is setting up a parallel job for the first time, a panel opens to create the fastran.hosts file. This file is a list that contains all of the machines on which the user may launch a parallel job. This new feature allows the user to enter a new host and its default value.

CFD-FASTRAN 4.0

New Features in Fastran Version 4.0:

- Distributed Parallel Processing
- Arbitrary Polyhedral Solver (triangle, quad, polygon, tet, hex, prism, and polyhedral)
- Double Fringe Capability added to Chimera algorithm

The complete CFD-FASTRAN system is comprised of:

- **CFD-GEOM**

Interactive 3D Geometry Modeling and Mesh Generation
(Structured, Unstructured, and Mixed-Element Meshes)

- **CFD-FASTRAN**

Compressible Flow Solver With Moving Multi-Body Dynamics

- **CFD-VIEW**

Interactive 3D Graphics, Animation & Flow Visualization
Software

Highlights

CFD-FASTRAN provides highly accurate simulations of compressible flow and associated phenomena. The code incorporates robust higher-order accuracy numerics and shock capturing schemes.

- Structured, Unstructured, and Automated Chimera Grid System.
- Fully-Coupled 6DOF and Prescribed Motion Models
- Time-dependent Firing and Thrust Models
- General Finite Rate Chemistry and Thermal Non-equilibrium
- Inviscid, Viscous, and Advanced Turbulence Models
- Accurate High Order Explicit and Implicit Schemes
- Navier-Stokes or Euler Simulations
- Finite Volume Discretization
- High Accuracy Numerics: Roe and Van Leer Schemes
- Fully-Automated Chimera/Overset grid methodology for multiple moving bodies
- Comprehensive Output: forces, moments, accelerations, velocities and body location/orientation
- Intuitive, Easy-to-Use GUI, fast model setup
- Seamless Integration with CFD-GEOM and

Application examples

(click one of the links below to see FASTRAN examples)

Applied Aerodynamics

Escape Systems

CFD-VIEW

- Closed Loop Control Models, including autopilot, motor ignition/firing and thrust
- New force and moment output model with support for data reduction
- Integration with MDICE (Multi Disciplinary Computing Environment)

Missile Applications

CFD-FASTRAN Features

Geometry and Mesh

CFD-FASTRAN is fully integrated with CFD-GEOM for geometry/grid generation. CFD-GEOM has the ability to import geometry data from other CAD/CAE systems. The geometry/grid capabilities include:

- Extensive Library of NURBS-Based Tools: Comprehensive Geometry Operations
- CAD Geometry Import
- Structured Mesh generation Auto-Mapping of Meshes in NURBS Surface: Interactive Definition and Alteration of Mesh Point Distributions

Numerics

CFD-FASTRAN offers State-Of-The-Art numerical techniques for solution accuracy and efficiency:

- Finite volume, density-based
- Roe or Van Leer Flux Splitting
- High order differencing schemes (up to third order), Min-Mod, Van Leer, and Osher-Chakravathy limiters
- Explicit, point implicit, and fully implicit time integration

- Unstructured and Hybrid Mesh Generation: Automatic NURBS Surface Creation from Automated 2D/3D Advancing Front Method
- Automatic Template Gridding Using Editable Script

Grid Systems

CFD-FASTRAN supports the following grid systems:

- Multi-block, structured grids with blockages
- Chimera/Overset grids with automatic hole cutting
- Unstructured arbitrary polyhedral grids
- Hybrid grid systems with different grid types in different zones or different cell types in the same zone

Fluid Flow

A full range of flow conditions is handled by CFD-FASTRAN, including:

- Compressible Flows with Euler / Navier-Stokes equations
- Steady/Unsteady
- Inviscid/Laminar/Turbulent
- Choice of Turbulence Models including:
 - k- ϵ
 - k- ω
 - Baldwin-Lomax
 - Spalart-Allmaras
- Finite Rate Chemistry
- Thermal Equilibrium or Non-Equilibrium

Moving-Body Technologies

CFD-FASTRAN offers a very efficient, automated environment for simulation of multiple moving bodies, including:

- Highly efficient automated hole cutting and interpolation for overset grids
- 6DOF (Degrees of Freedom) model with multiple moving bodies, point forces, constraints, and thrust profiles

Visualization

CFD-FASTRAN is fully integrated with CFD-VIEW, an interactive 3D graphics, animation, and flow visualization software.

- Display of vector/scalar data using grids, surfaces, cutting planes, contours, vectors, iso-valued surfaces
- Point & line probes, Function calculator
- Flow animation

Future Directions

CFD-FASTRAN will continue providing the most efficient, accurate, and geometrically computational aerodynamic technologies. Future Plans include:

- Unstructured chimera with 6DOF model
- Actuator Disk Model (for rotating blades and engines)
- Surface Reactions

Support

CFDRC has extensive expertise incompressible flow and moving body applications including pilot ejection/escape systems, store separation, missile staging, control fin effectiveness, external aerodynamics, and a broad range of aerospace applications. Our experienced engineers are available to support your usage of CFD-FASTRAN as well as to provide design and engineering services.

- Comprehensive output including monitor functions, forces, moments, accelerations, velocities, and body orientations

Top of the page

CFD-FASTRAN is a trademark of CFD Research Corporation. All rights reserved.

The Air Turbo Rocket

The Air Turbo Rocket (ATR) engine is an air-breathing propulsion system, combining advantages from both rocket and turbojet technologies. CFD Research Corporation (CFDRC) is developing high performance solid and bipropellant ATR propulsion demonstration engines with prototype flight testing as a near term goal.

Select a link to learn more about the Air Turbo Rocket:

- [How it works](#)
- [ATR Performance](#)
- [Demonstrator Engine Development](#)

- [Solid Fuel Propellant Development](#)
- [Turbomachinery Component Development](#)
- [Packaged Throttle Valve Development](#)
- [Combustor and Inlet Design](#)
- [Global Engine and Mission Analysis Software \(GEMA\)](#)
- [Tactical Missile Implementation](#)
- [Launch Vehicle Implementation](#)
- [CFDRC's Industrial Partners](#)
- [References](#)

Turbomachinery Component Development

High performance, lightweight turbomachinery components have been specifically tailored for maximum gas generator effluent power extraction. The turbine and compressor components are undergoing constant refinement through advanced CFD analysis and experimental testing in a customized Test Rig.

Turbine

Compressor

ATR Turbomachinery Test Rig

Packaged Throttle Valve Development

Thrust and power output of the SFATR are controlled through a high-performance throttle valve, which regulates the hot gas flow rate from the gas generator. In a flight configuration installation, the size, actuation power requirements, and response time of the engine throttle valve must be minimized. The "pressure balanced" design shown here optimally minimizes these critical design requirements.

"Pressure-Balanced" Throttle Valve Concept

Combustor and Inlet Design

The overall performance of an ATR is closely coupled to the design of the combustor and inlets. Our advanced 3-D CFD analysis software indicates that a swirl stabilized combustor is the design of choice for the ATR ([Ref. 1](#)). These CFD tools have also been used to properly locate and size the air inlets. ([Ref. 2](#)). A quad-furcated inlet design has shown excellent overall performance characteristics.

ATR Combustor Analysis
Simulation by CFD-ACE

Engine Inlet Analysis
Simulation by **CFD-FASTRAN**

Global Engine and Mission Analysis Software(GEMA)

A comprehensive understanding of missile vehicle/propulsion/mission interactions requires a highly-integrated analysis capability. GEMA software simulates the performance and trajectories of candidate missile propulsion systems, including ATR, SRM, Turbojet, and Ducted Rocket engines. This PC-based software is composed of a Graphical User Interface for data input and a robust 3 degree of freedom trajectory code for flight simulation. GEMA has proven to be an essential tool in trade studies of proposed engine/vehicle configurations and missions.

Preliminary comments on Solid Rocket Motor Simulation

Mark Ostrander, CFD Research Corporation

Vladimir Kudriavtsev, CFD Canada (416-441-0803, vvk@cfdr.com)

The main impression from working with this case was that the nozzle is severely under-expanded. The area ratio of the nozzle is only 1.62:1. This should produce an exit Mach number of approximately 1.85 and an exit pressure ratio (P/P_0) of .17016. That gives an exit pressure of 2.8 MPa, which is still way above ambient. ACE+ gave an exit pressure on the order of 1.9-2.0 MPa, which is consistent with a slight reduction of the area ratio due to the boundary layer. Regardless, the ratio of nozzle exit pressure to ambient causes the plume to expand radially and pass out through both boundaries of the secondary ambient section.

The Mach number indicated reaches its maximum way out in the ambient region. The outer boundary of the ambient region is still quite close to the nozzle exit. Under other circumstances, this could cause some difficulties, but it did not in this case.

Boundary Conditions: INLET BC's on the left and top surfaces of the upstream section (yellow). The specific BC type was "Total Pressure (Normal)" and I imposed the values indicated in the email. For the downstream block, we used the OUTLET BC's on the top and right surfaces (green). The specific BC type was "Static Pressure".

M. Ostrander, V. Kudriavtsev Solid Rocket Motor Simulation Case

Pressure: preference is to set the reference pressure to 0.0 and then use absolute values on the various boundaries. In this case, it was 16.5 MPa and 2900 deg K for the inlet and 100,000 Pa and 300 deg K for the outlets. Equivalent pressure levels could have been set by keeping the reference pressure at 100,000 and setting $P=16.4$ MPa in the chamber and $p=0.0$ at the ambient boundaries. Pressures can be converted back to PSIA in VIEW using the calculator, and it is much simpler to work in absolute values using my convention.

When working with such severe pressure gradients, the initial conditions to the CFD model are always important, as they relate to the relaxation parameters one is forced to use. For these cases, it is preferable to set the initial conditions on a volume-by-volume basis, gradually stepping the velocity up from the transonic section downstream, and the pressure levels down. Even so, from volume to volume, there will be a step change in pressure level, with the tendency to create artificially high velocities at the junctions in the initial steps of the simulation. To counteract that trend, relaxation parameters were increased from default values, pretty far over to the right. They may have been over-compensated, but it doesn't hurt to be safe.

Steady state simulation was performed in 2 ways. The first was increasing the radial extent of the computational domain in the region external to the nozzle exit. I was originally bothered with how the exhaust plume was exiting every point on the top and right outflow boundaries. If the plume was optimally expanded or especially over-expanded, the boundaries should be moved farther away from the nozzle exit. We used two separate gas compositions – one for the combustion products and one for the ambient air. This allows one to easily distinguish the nozzle plume and opens the door to doing chemistry calculations in the combustion chamber. For instance, the difference between frozen and equilibrium flow can be a major performance distinction.

This figure shows the geometry and boundary conditions in the area external to the nozzle exit. There is an additional inlet at the nozzle exit plane and the outer boundary is farther up. The composition upstream of the nozzle throat was a mixture of combustion products, and the composition at the added inlet was a 2-species air model. The second inlet was added because there were some difficulties in trying to specify an "air" composition on the two outflow boundaries without ever having given the simulation a source of that air. The second inlet section is also a bit more realistic than a vertical wall.

M. Ostrander, V. Kudriavtsev Solid Rocket Motor Simulation Case

After the geometry modification, the plume still expands and goes out the top wall, as shown in the following figure. The velocity vectors are colored by the mixture fraction of the combustion gasses, so the magenta areas are SRM effluent and the blue is the air. It can be seen that a large portion of the air is flowing in at the top boundary where the static pressure outflow boundary is specified. It is desirable to arrange the computational domain such that the flow is going OUT at all the static pressure boundaries, but it seemed to be OK in this case.

At least it is evident that the plume is starting to curve over toward the horizontal with the increased geometric extent of the domain. It would be better if the top boundary was such that the plume did not interact with it at all.

Turbulence models. Standard k-e turbulence model was utilized with code defaults. Same boundary values of k and e at all boundaries, even though some were quiescent air and others were combustion products at higher velocities. If one were concerned about the boundary layer in the nozzle, then the grid would have to be refined considerably. The general grid level in these simulations was pretty coarse, but that depends on the purpose of the run, so it could have been perfectly appropriate. Total pressure boundary conditions in the combustion chamber was used.

ANNEX F

COMPUTER SYSTEM REQUIREMENTS

Computational Fluid Dynamics Software & Services

CFX Home

Products Home

CFX-4

CFX-5

CFX-TASCflow

Home

What's New

Step-by-Step

Specification

Requirements

CFX-Hexa

News

Info Request

CFX-ProMixus

CFX-BladeGen

CFX-Build

CFX-Radiation

CFX-TurboGrid

Contact CFX

CFX-TASCflow

Version 2.11
Now Available!

Computer System Requirements for version 2.11

Unix

- Compaq systems running Tru64 UNIX versions 4.0e or 5.1
On Compaq systems there are some combinations of operating systems, graphics cards and graphics software that will prevent CFX-Build and/or CFX-Visualise running.
- Hewlett Packard systems running HP-UX versions 10.20 or 11.0
- IBM systems running AIX version 4.3
- SGI systems running IRIX or IRIX64 version 6.5
- SUN systems running Solaris version 7 or 8
- Cray and NEC systems.

Microsoft Windows

- Windows NT version 4.0 with Service Pack 5

[\[Return to CFX-TASCflow home page\]](#)

FLUENT 6 Windows Hardware Requirements

Platforms: Intel

Systems Requirements

The following is a list of system requirements for Fluent software products:

Hardware

- CPU: Intel Pentium Family of Processors
- Video graphics device with minimum 1024 x 768 resolution with 256 minimum colors but 24-bit recommended
- 128 MB RAM minimum
- Windows compatible 3-button mouse recommended
- CD-ROM. All Fluent products are distributed on CD-ROM
- Ethernet adapter card

Operating System

- Windows NT 4.0 (Service Pack 5 or higher)
- Windows 2000 (Service Pack 5 or higher). Check specific software products on this web site for their current platform availability.
- TCP/IP network protocol

Disk and Memory Requirements

Memory and Swap File Size

A minimum of 128 MB RAM is required for a standard laminar fluid-flow problem. The size of the swap file should not be smaller than the amount of RAM in your system. Larger problems require considerably more memory (RAM plus swap space) and increasing RAM dramatically improves performance.

Disk Space Requirements

Documentation and tutorials are included on a separate

Software

Preprocessors

General-Purpose

Application-Focused

Educational Software

Technology Leadership

University Program

.....

CD. Installation to the hard-disk is optional.

FLUENT 6	Minimum Disk Space Requirement
software	35 MB
documentation/online help	50 MB
tutorials	300 MB

License Server Software

Fluent Inc. products use FLEXlmTM for license management. If you choose to employ the FLEXlmTM floating manager on a Windows network, your network server must be running Microsoft Windows NT or Windows 2000.

Graphics Support

Fluent Inc. supports Professional OpenGL graphics cards.

Development Tools

FLUENT 6 Compiler Requirements

Compiled UDFs for FLUENT 6 require Microsoft Visual C++ 6.0 - Standard Edition - Minimum (Service Release Pack 3 or higher).

[Home](#) | [Solutions](#) | [Software](#) | [Consulting](#) | [Support](#) | [About Us](#) | [Worldwide](#) | [Contact](#) | [Search](#)

All Rights Reserved © 2001 – Fluent Inc.

CFDRC Software Available Platforms

CFD-ACE+

Unix

Sun: SunOS 5.5.1 or later

HP: HP-UX 11.0 or later

SGI: IRIX 6.5 or later

DEC: Digital Unix 4.0D or later

IBM: AIX 4.2 or later

**LINUX: Redhat 5.2 or later
for Intel Processors**

Windows NT

**Windows NT 4.0, SP 3 or above, Windows
2000, and Windows XP for Intel Processors**

CFD-FASTRAN

Unix

Sun: SunOS 5.5.1 or later

HP: HP-UX 10.20 or later

SGI: IRIX 6.2 or later

DEC: Digital Unix 4.0D or later

IBM: AIX 4.2 or later

**LINUX: Redhat 5.2 or later
for Intel Processors**

Windows NT

**Windows NT 4.0, SP 3 or above, Windows
2000, and Windows XP for Intel Processors**

Appendix B – Example CEA Output

**

NASA-GLENN CHEMICAL EQUILIBRIUM PROGRAM CEA, JUNE 13, 2000
BY BONNIE MCBRIDE AND SANFORD GORDON
REFS: NASA RP-1311, PART I, 1994 AND NASA RP-1311, PART II, 1996

**

```

reac
  name AP N 1 H 4 CL 1 O 4 wt%=75.00
        h,cal=-70690.0 t(k)=298.15
  name HTPB C 6.9773 H 11.875 O 0.2188 N 0.0518 wt%=25.00
        h,cal=-32363.0 t(k)=298.15

prob   case=HTPB/AP_Theo_Isp rocket equilibrium p,psia=1000 pi/p=68.0457

outp   transport massf

end

OPTIONS: TP=F HP=F SP=F TV=F UV=F SV=F DETN=F SHOCK=F REFL=F INCD=F
RKT=T FROZ=F EQL=T IONS=F SIUNIT=T DEBUGF=F SHKDBG=F DETDBG=F
TRANSPT=T

TRACE= 0.00E+00 S/R= 0.000000E+00 H/R= 0.000000E+00 U/R= 0.000000E+00

Pc,BAR =      68.947304

Pc/P =      68.0457

SUBSONIC AREA RATIOS =

SUPERSONIC AREA RATIOS =

NFZ= 1 Mdot/Ac= 0.000000E+00 Ac/At= 0.000000E+00

  REACTANT      WT.FRAC  (ENERGY/R),K  TEMP,K  DENSITY
  EXPLODED FORMULA
N: AP           0.750000 -0.355724E+05  298.15  0.0000
    N 1.00000 H 4.00000 CL 1.00000 O 4.00000
N: HTPB         0.250000 -0.162856E+05  298.15  0.0000
    C 6.97730 H 11.87500 O 0.21880 N 0.05180

SPECIES BEING CONSIDERED IN THIS SYSTEM
(CONDENSED PHASE MAY HAVE NAME LISTED SEVERAL TIMES)
LAST thermo.inp UPDATE: 10/22/02
```

g 7/97	*C	g 8/99	CCL	g 8/99	CCL2
n12/93	CCL3	tpis91	CCL4	tpis79	*CH
g 9/99	CHCL	n12/93	CHCL2	g 7/99	CHCL3
g 4/02	CH2	g12/99	CH2CL	tpis91	CH2CL2
g 4/02	CH3	tpis91	CH3CL	g11/00	CH2OH
g 7/00	CH3O	g 8/99	CH4	g 7/00	CH3OH
srd 01	CH3OOH	g 8/99	*CN	g12/99	CNN
tpis79	*CO	tpis91	COCL	tpis91	COCL2
tpis91	COHCL	g 9/99	*CO2	tpis91	COOH
tpis91	*C2	tpis91	C2CL	g 5/02	C2CL2
tpis91	C2CL3	g 5/02	C2CL4	g 5/02	C2CL6
g 6/01	C2H	g 5/02	C2HCL	g 5/02	C2HCL3
g 1/91	C2H2,acetylene	g 5/01	C2H2,vinylidene	tpis91	C2H2CL2
g 4/02	CH2CO,ketene	g 3/02	O(CH)2O	srd 01	HO(CO)2OH
g 7/01	C2H3,vinyl	g 5/02	C2H3CL	srd 01	CH2CL-COOH
g 9/00	CH3CN	g 6/96	CH3CO,acetyl	g 1/00	C2H4
g 8/88	C2H4O,ethylen-o	g 8/88	CH3CHO,ethanal	g 6/00	CH3COOH
srd 01	OHCH2COOH	g 7/00	C2H5	g 7/00	C2H6
g 8/88	CH3N2CH3	g 8/88	C2H5OH	g 7/00	CH3OCH3
srd 01	CH3O2CH3	g 7/00	CCN	tpis91	CNC
srd 01	OCCN	tpis79	C2N2	g 8/00	C2O
tpis79	*C3	n 4/98	C3H3,1-propynl	n 4/98	C3H3,2-propynl
g 2/00	C3H4,allene	g 1/00	C3H4,propyne	g 5/90	C3H4,cyclo-
g 3/01	C3H5,allyl	g 2/00	C3H6,propylene	g 1/00	C3H6,cyclo-
g 6/01	C3H6O,propylox	g 6/97	C3H6O,acetone	g 1/02	C3H6O,propanal
g 7/01	C3H7,n-propyl	g 9/85	C3H7,i-propyl	g 2/00	C3H8
g 2/00	C3H8O,1propanol	g 2/00	C3H8O,2propanol	srd 01	CNCOCN
g 7/88	C3O2	g tpis	*C4	g 7/01	C4H2,butadiyne
g 8/00	C4H4,1,3-cyclo-	n10/92	C4H6,butadiene	n10/93	C4H6,1butyne
n10/93	C4H6,2butyne	g 8/00	C4H6,cyclo-	n 4/88	C4H8,1-butene
n 4/88	C4H8,cis2-buten	n 4/88	C4H8,tr2-butene	n 4/88	C4H8,isobutene
g 8/00	C4H8,cyclo-	g10/00	(CH3COOH)2	n10/84	C4H9,n-butyl
n10/84	C4H9,i-butyl	g 1/93	C4H9,s-butyl	g 1/93	C4H9,t-butyl
g12/00	C4H10,n-butane	g 8/00	C4H10,isobutane	g 6/01	C4N2
g 8/00	*C5	g 5/90	C5H6,1,3cyclo-	g 1/93	C5H8,cyclo-
n 4/87	C5H10,1-pentene	g 2/01	C5H10,cyclo-	n10/84	C5H11,pentyl
g 1/93	C5H11,t-pentyl	n10/85	C5H12,n-pentane	n10/85	C5H12,i-pentane
n10/85	CH3C(CH3)2CH3	g 2/93	C6H2	g11/00	C6H5,phenyl
g 8/00	C6H5O,phenoxy	g 8/00	C6H6	g 8/00	C6H5OH,phenol
g 1/93	C6H10,cyclo-	n 4/87	C6H12,1-hexene	g 6/90	C6H12,cyclo-
n10/83	C6H13,n-hexyl	g 6/01	C6H14,n-hexane	g 7/01	C7H7,benzyl
g 1/93	C7H8	g12/00	C7H8O,cresol-mx	n 4/87	C7H14,1-heptene
n10/83	C7H15,n-heptyl	n10/85	C7H16,n-heptane	n10/85	C7H16,2-methylh
n 4/89	C8H8,styrene	n10/86	C8H10,ethylbenz	n 4/87	C8H16,1-octene
n10/83	C8H17,n-octyl	n 4/85	C8H18,n-octane	n 4/85	C8H18,isooctane
n10/83	C9H19,n-nonyl	g 3/01	C10H8,naphthale	n10/83	C10H21,n-decyl
g 8/00	C12H9,o-bipheny	g 8/00	C12H10,biphenyl	g 7/97	*CL
g 6/95	CLCN	tpis89	CLO	g 7/93	CLO2
tpis89	CL2	tpis89	CL2O	g 6/97	*H
g 6/01	HCN	g 1/01	HCO	tpis89	HCCN
g 6/01	HCCO	tpis89	HCL	g 6/01	HNC
g 7/00	HNCO	g10/01	HNO	tpis89	HNO2
g 5/99	HNO3	g 1/01	HOCL	g 4/02	HO2
tpis78	*H2	g 5/01	HCHO,formaldehy	g 6/01	HCOOH
g 8/89	H2O	g 6/99	H2O2	g 6/01	(HCOOH)2
g 5/97	*N	g 6/01	NCO	g 4/99	*NH

g 3/01	NH2	tpis89	NH3	tpis89	NH2OH
tpis89	*NO	g 4/99	NOCL	g 4/99	NO2
g 4/99	NO2CL	j12/64	NO3	tpis78	*N2
g 6/01	NCN	g 5/99	N2H2	tpis89	NH2NO2
g 4/99	N2H4	g 4/99	N2O	g 4/99	N2O3
tpis89	N2O4	g 4/99	N2O5	tpis89	N3
g 4/99	N3H	g 5/97	*O	g 4/02	*OH
tpis89	*O2	g 8/01	O3	n 4/83	C(gr)
n 4/83	C(gr)	n 4/83	C(gr)	g11/99	H2O(cr)
g 8/01	H2O(L)	g 8/01	H2O(L)	j 9/65	NH4CL(II)
j 9/65	NH4CL(III)	j 9/65	NH4CL(III)		

SPECIES WITH TRANSPORT PROPERTIES

PURE SPECIES

C	CCL4	CHCL3	CH2CL2
CH3CL	CH4		
CH3OH	CO	CO2	C2H2, acetylene
C2H4	C2H6		
C2H5OH	C2N2	CL2	H
HCN	HCL		
H2	H2O	N	NH3
NO	NOCL		
NO2	N2	N2O	N2O4
O	OH		
O2			

BINARY INTERACTIONS

C	O
CH4	O2
CO	CO2
CO	N2
CO	O2
CO2	H2
CO2	H2O
CO2	N2
CO2	O2
H	H2
H	N
H	N2
H	O
H2	H2O
H2	N2
H2	O2
H2O	N2
H2O	O2
N	NO
N	N2
N	O
N	O2
NO	O
N2	O
N2	O2
O	O2

O/F = 0.000000

ENTHALPY (KG-MOL) (K) /KG	EFFECTIVE FUEL h(2) /R -0.26779434E+03	EFFECTIVE OXIDANT h(1) /R 0.00000000E+00	MIXTURE h0/R -0.26779434E+03
KG-FORM.WT./KG	bi(2)	bi(1)	b0i
*N	0.65130901E-02	0.00000000E+00	0.65130901E-02
*H	0.55222513E-01	0.00000000E+00	0.55222513E-01
*CL	0.63835872E-02	0.00000000E+00	0.63835872E-02
*O	0.26081361E-01	0.00000000E+00	0.26081361E-01
*C	0.17443640E-01	0.00000000E+00	0.17443640E-01

POINT	ITN	T C	N	H	CL	O
1	17	1712.614 -6.173	-12.479	-7.920	-22.229	-25.733
Pinf/Pt = 1.818670						
2	4	1504.881 -5.171	-12.574	-8.017	-23.225	-28.016
Pinf/Pt = 1.818365						
2	1	1504.935 -5.171	-12.574	-8.017	-23.225	-28.016
3	7	904.988 -1.026	-13.635	-9.182	-29.104	-40.639
ADD C(gr)						
3	3	915.065 -1.401	-13.650	-9.174	-29.006	-40.193

THEORETICAL ROCKET PERFORMANCE ASSUMING EQUILIBRIUM

COMPOSITION DURING EXPANSION FROM INFINITE AREA COMBUSTOR

Pinj = 1000.0 PSIA
CASE = HTPB/AP_Theo_Is

NAME	REACTANT	WT FRACTION (SEE NOTE)	ENERGY KJ/KG-MOL	TEMP K
NAME	AP	0.7500000	-295766.960	298.150
NAME	HTPB	0.2500000	-135406.792	298.150

O/F= 0.000000 %FUEL= 0.000000 R,EQ.RATIO= 2.135012 PHI,EQ.RATIO= 0.000000

	CHAMBER	THROAT	EXIT
Pinf/P	1.0000	1.8184	68.046
P, BAR	68.947	37.917	1.0132
T, K	1712.61	1504.93	915.07
RHO, KG/CU M	9.4064 0	5.8914 0	2.8176-1
H, KJ/KG	-2226.58	-2637.56	-4339.92
U, KJ/KG	-2959.56	-3281.16	-4699.53
G, KJ/KG	-20495.7	-18691.3	-14101.3

S, KJ/(KG) (K)	10.6674	10.6674	10.6674
M, (1/n)	19.427	19.442	21.157
MW, MOL WT	19.427	19.442	20.572
(dLV/dLP)t	-1.00096	-1.00245	-1.08247
(dLV/dLT)p	1.0073	1.0224	2.4498
Cp, KJ/(KG) (K)	1.9910	2.0374	12.4114
GAMMAS	1.2774	1.2771	1.1205
SON VEL,M/SEC	967.6	906.6	634.8
MACH NUMBER	0.000	1.000	3.239

TRANSPORT PROPERTIES (GASES ONLY)

CONDUCTIVITY IN UNITS OF MILLIWATTS/(CM) (K)

VISC,MILLIPOISE	0.60803	0.55606	0.39263
-----------------	---------	---------	---------

WITH EQUILIBRIUM REACTIONS

Cp, KJ/(KG) (K)	1.9910	2.0374	8.4086
CONDUCTIVITY	2.2440	2.0743	5.1627
PRANDTL NUMBER	0.5395	0.5462	0.6395

WITH FROZEN REACTIONS

Cp, KJ/(KG) (K)	1.9251	1.8828	1.7350
CONDUCTIVITY	2.1908	1.9661	1.2862
PRANDTL NUMBER	0.5343	0.5325	0.5296

PERFORMANCE PARAMETERS

Ae/At	1.0000	9.2207
CSTAR, M/SEC	1290.9	1290.9
CF	0.7023	1.5927
Ivac, M/SEC	1616.5	2230.8
Isp, M/SEC	906.6	2055.9

MASS FRACTIONS

CH4	0.00017	0.00049	0.02319
*CO	0.43861	0.42770	0.24072
*CO2	0.07803	0.09431	0.26675
*CL	0.00001	0.00000	0.00000
HCN	0.00002	0.00001	0.00000
HCL	0.23275	0.23275	0.23275
*H2	0.03530	0.03593	0.03257
H2O	0.12387	0.11756	0.09665
NH3	0.00011	0.00010	0.00005
*N2	0.09113	0.09113	0.09119
C(gr)	0.00000	0.00000	0.01613

* THERMODYNAMIC PROPERTIES FITTED TO 20000.K

PRODUCTS WHICH WERE CONSIDERED BUT WHOSE MASS FRACTIONS
WERE LESS THAN 5.000000E-06 FOR ALL ASSIGNED CONDITIONS

*C	CCL	CCL2	CCL3	CCL4
----	-----	------	------	------

*CH	CHCL	CHCL2	CHCL3	CH2
CH2CL	CH2CL2	CH3	CH3CL	CH2OH
CH3O	CH3OH	CH3OOH	*CN	CNN
COCL	COCL2	COHCL	COOH	*C2
C2CL	C2CL2	C2CL3	C2CL4	C2CL6
C2H	C2HCL	C2HCL3	C2H2, acetylene	
C2H2, vinylidene				
C2H2CL2	CH2CO, ketene	O (CH) 2O	HO (CO) 2OH	C2H3, vinyl
C2H3CL	CH2CL-COOH	CH3CN	CH3CO, acetyl	C2H4
C2H4O, ethylen-o	CH3CHO, ethanal	CH3COOH	OHCH2COOH	C2H5
C2H6	CH3N2CH3	C2H5OH	CH3OCH3	CH3O2CH3
CCN	CNC	OCCN	C2N2	C2O
*C3	C3H3, 1-propynl	C3H3, 2-propynl	C3H4, allene	C3H4, propyne
C3H4, cyclo-	C3H5, allyl	C3H6, propylene	C3H6, cyclo-	
C3H6O, propylox				
C3H6O, acetone	C3H6O, propanal	C3H7, n-propyl	C3H7, i-propyl	C3H8
C3H8O, 1propanol	C3H8O, 2propanol	CNCOCN	C3O2	*C4
C4H2, butadiyne	C4H4, 1, 3-cyclo-	C4H6, butadiene	C4H6, 1butyne	C4H6, 2butyne
C4H6, cyclo-	C4H8, 1-butene	C4H8, cis2-buten	C4H8, tr2-butene	
C4H8, isobutene				
C4H8, cyclo-	(CH3COOH) 2	C4H9, n-butyl	C4H9, i-butyl	C4H9, s-butyl
C4H9, t-butyl	C4H10, n-butane	C4H10, isobutane	C4N2	*C5
C5H6, 1, 3cyclo-	C5H8, cyclo-	C5H10, 1-pentene	C5H10, cyclo-	C5H11, pentyl
C5H11, t-pentyl	C5H12, n-pentane	C5H12, i-pentane	CH3C (CH3) 2CH3	C6H2
C6H5, phenyl	C6H5O, phenoxy	C6H6	C6H5OH, phenol	C6H10, cyclo-
C6H12, 1-hexene	C6H12, cyclo-	C6H13, n-hexyl	C6H14, n-hexane	C7H7, benzyl
C7H8	C7H8O, cresol-mx	C7H14, 1-heptene	C7H15, n-heptyl	C7H16, n-
heptane				
C7H16, 2-methylh	C8H8, styrene	C8H10, ethylbenz	C8H16, 1-octene	C8H17, n-
octyl				
C8H18, n-octane	C8H18, isooctane	C9H19, n-nonyl	C10H8, naphthale	C10H21, n-
decyl				
C12H9, o-bipheny	C12H10, biphenyl	CLCN	CLO	CLO2
CL2	CL2O	*H	HCO	HCCN
HCCO	HNC	HNCO	HNO	HNO2
HNO3	HOCL	HO2	HCHO, formaldehy	HCOOH
H2O2	(HCOOH) 2	*N	NCO	*NH
NH2	NH2OH	*NO	NOCL	NO2
NO2CL	NO3	NCN	N2H2	NH2NO2
N2H4	N2O	N2O3	N2O4	N2O5
N3	N3H	*O	*OH	*O2
O3	H2O (cr)	H2O (L)	NH4CL (II)	NH4CL (III)

NOTE. WEIGHT FRACTION OF FUEL IN TOTAL FUELS AND OF OXIDANT IN TOTAL OXIDANTS

Appendix C – User Subroutine for Non-Erosive Burning

```
!*****
!  SUBROUTINE non_erosive_burning
!
!  User code by John T. Weaver
!*****
      MODULE cfdrc_user
!*****

      IMPLICIT NONE
      INTEGER, PARAMETER :: int_p  = SELECTED_INT_KIND(8)
      INTEGER, PARAMETER :: string_length = 80
      INTEGER, PARAMETER :: real_p  = SELECTED_REAL_KIND(8)
      INTEGER, PARAMETER :: XDIR = 1, YDIR = 2, ZDIR = 3
! Utility parameters.

      REAL(real_p) , PARAMETER :: zero = 0.0d0, one = 1.0d0, two = 2.0d0, &
& three = 3.0d0, four = 4.0d0, pi = 3.1415926535898d0

! Declare global variables

! USER CODE BEGIN

      INTEGER(int_p) :: ind_deform_v, ind_deform_u, N, cell_index, XC_index, &
& C_index, P_index, YC_index
      INTEGER(int_p), DIMENSION(30) :: ind_upper, ind_lower, n_cells_l, n_cells_u, &
& vol_index_l, vol_index_u
      REAL(real_p), DIMENSION(30) :: low_val, up_val
      LOGICAL :: first_iter = .TRUE.

! USER CODE END

      END MODULE cfdrc_user

!*****
      SUBROUTINE uout(iflag)
!DEC$ ATTRIBUTES DLLEXPORT :: uout
!*****
! copyright (c) 1998  cfd research corp.  all rights reserved.
!
! purpose : for customized user output.
!
! iflag:  flag indicating calling location.
!
! This routine is called 5 times at different instances of iterative
! cycle indicated by iflag.
!
! iflag :
!         0 - At the beginning (only for dtf reading calls, At this point
!         most of the other data may not be available.
!         users can get variable indices, and may be
!         external reading of files can be done with
this
!         flag = 0)
!         1 - At the beginning of RUN. (At this poin most of the boundary
```

```

!                                conditions, properties are set.
users
!                                should be able to get cell or
boundary
!                                values for different variables.)
!                                2 - At the beginning of time step (only for transient problems).
!                                3 - At the end of each iteration.
!                                4 - At the end of each time step.(only for transient problems).
!                                5 - At the end of RUN.
!
! One may use get_value_one_cell to get the values. To get the cell
! indexes, user has to supply the x,y,z locations and use the
! get_cell_index(vc_index,x,y,z,ic_global,error).
!-----

! Include required global variables declared in cfdrc_user module.
  USE cfdrc_user, ONLY : int_p, real_p, string_length
  USE cfdrc_user, ONLY : n_cells_l, vol_index_l, N, cell_index, XC_index,
&
&                                & YC_index, P_index, vol_index_u, n_cells_u,low_val,
&
&                                & up_val

!***** DO NOT REMOVE FOLLOWING LINE FOR MS WINDOWS OS *****
  INCLUDE 'cfdrc_include'
!*****

  IMPLICIT NONE

  INTEGER(int_p), INTENT(IN) :: iflag

! Declare required local variables here.
! USER CODE BEGIN

  REAL(real_p) :: xc, yc,pressure_tot, pressure, pressure_avg,pressnoz,&
&                                & low_temp,low_tempold,up_temp,up_tempold
  CHARACTER(len=string_length) :: vol_name, XC_name = 'XC', &
&                                & YC_name = 'YC', P_name = 'P'
  LOGICAL :: error
  INTEGER(int_p) :: ierror
  INTEGER :: NPRIN,COUNT,I,J,M,K

! USER CODE END

!-----

! Program main body.
! USER CODE BEGIN

! Define Constants

  NPRIN=25
  COUNT=1
  pressure_tot = 0.0

! Retrieve the info that is constant throughout the simulation.

```

```

        IF (iflag == 1) THEN
! Supply name of volume from DTF
        vol_name = '1_lower'

! Get index of volume (vol_index_n)
        CALL get_vc_index(vol_name, vol_index_1(1), error)

! Get remainder of volume indices
        vol_name = '2_lower'
        CALL get_vc_index(vol_name, vol_index_1(2), error)
        vol_name = '3_lower'
        CALL get_vc_index(vol_name, vol_index_1(3), error)
        vol_name = '4_lower'
        CALL get_vc_index(vol_name, vol_index_1(4), error)
        vol_name = '5_lower'
        CALL get_vc_index(vol_name, vol_index_1(5), error)
        vol_name = '6_lower'
        CALL get_vc_index(vol_name, vol_index_1(6), error)
        vol_name = '7_lower'
        CALL get_vc_index(vol_name, vol_index_1(7), error)
        vol_name = '8_lower'
        CALL get_vc_index(vol_name, vol_index_1(8), error)
        vol_name = '9_lower'
        CALL get_vc_index(vol_name, vol_index_1(9), error)
        vol_name = '10_lower'
        CALL get_vc_index(vol_name, vol_index_1(10), error)
        vol_name = '11_lower'
        CALL get_vc_index(vol_name, vol_index_1(11), error)
        vol_name = '12_lower'
        CALL get_vc_index(vol_name, vol_index_1(12), error)
        vol_name = '13_lower'
        CALL get_vc_index(vol_name, vol_index_1(13), error)
        vol_name = '14_lower'
        CALL get_vc_index(vol_name, vol_index_1(14), error)
        vol_name = '15_lower'
        CALL get_vc_index(vol_name, vol_index_1(15), error)
        vol_name = '16_lower'
        CALL get_vc_index(vol_name, vol_index_1(16), error)
        vol_name = '17_lower'
        CALL get_vc_index(vol_name, vol_index_1(17), error)
        vol_name = '18_lower'
        CALL get_vc_index(vol_name, vol_index_1(18), error)
        vol_name = '19_lower'
        CALL get_vc_index(vol_name, vol_index_1(19), error)
        vol_name = '20_lower'
        CALL get_vc_index(vol_name, vol_index_1(20), error)
        vol_name = '21_lower'
        CALL get_vc_index(vol_name, vol_index_1(21), error)
        vol_name = '22_lower'
        CALL get_vc_index(vol_name, vol_index_1(22), error)
        vol_name = '23_lower'
        CALL get_vc_index(vol_name, vol_index_1(23), error)
        vol_name = '24_lower'
        CALL get_vc_index(vol_name, vol_index_1(24), error)
        vol_name = '25_lower'
        CALL get_vc_index(vol_name, vol_index_1(25), error)

```

```

vol_name = '1_upper'
CALL get_vc_index(vol_name, vol_index_u(1), error)
vol_name = '2_upper'
CALL get_vc_index(vol_name, vol_index_u(2), error)
vol_name = '3_upper'
CALL get_vc_index(vol_name, vol_index_u(3), error)
vol_name = '4_upper'
CALL get_vc_index(vol_name, vol_index_u(4), error)
vol_name = '5_upper'
CALL get_vc_index(vol_name, vol_index_u(5), error)
vol_name = '6_upper'
CALL get_vc_index(vol_name, vol_index_u(6), error)
vol_name = '7_upper'
CALL get_vc_index(vol_name, vol_index_u(7), error)
vol_name = '8_upper'
CALL get_vc_index(vol_name, vol_index_u(8), error)
vol_name = '9_upper'
CALL get_vc_index(vol_name, vol_index_u(9), error)

vol_name = '10_upper'
CALL get_vc_index(vol_name, vol_index_u(10), error)
vol_name = '11_upper'
CALL get_vc_index(vol_name, vol_index_u(11), error)
vol_name = '12_upper'
CALL get_vc_index(vol_name, vol_index_u(12), error)
vol_name = '13_upper'
CALL get_vc_index(vol_name, vol_index_u(13), error)
vol_name = '14_upper'
CALL get_vc_index(vol_name, vol_index_u(14), error)
vol_name = '15_upper'
CALL get_vc_index(vol_name, vol_index_u(15), error)
vol_name = '16_upper'
CALL get_vc_index(vol_name, vol_index_u(16), error)
vol_name = '17_upper'
CALL get_vc_index(vol_name, vol_index_u(17), error)
vol_name = '18_upper'
CALL get_vc_index(vol_name, vol_index_u(18), error)
vol_name = '19_upper'
CALL get_vc_index(vol_name, vol_index_u(19), error)
vol_name = '20_upper'
CALL get_vc_index(vol_name, vol_index_u(20), error)
vol_name = '21_upper'
CALL get_vc_index(vol_name, vol_index_u(21), error)
vol_name = '22_upper'
CALL get_vc_index(vol_name, vol_index_u(22), error)
vol_name = '23_upper'
CALL get_vc_index(vol_name, vol_index_u(23), error)
vol_name = '24_upper'
CALL get_vc_index(vol_name, vol_index_u(24), error)
vol_name = '25_upper'
CALL get_vc_index(vol_name, vol_index_u(25), error)

! Get number of cells in each volume (n_cells_n)

DO M=1,25

    CALL get_cells_vc(vol_index_l(M), n_cells_l(M), error)

```

```

        CALL get_cells_vc(vol_index_u(M), n_cells_u(M), error)

    ENDDO

! Get coordinate and variable indices.

! Get variable index of x coordinate.
    CALL get_var_index(XC_name, XC_index, error)

! Get variable index of y coordinate.
    CALL get_var_index(YC_name, YC_index, error)

! Get variable index of pressure P.
    CALL get_var_index(P_name, P_index, error)

    ENDIF

! *****
! Obtain output at end of time step.
! *****

    IF (iflag == 4) THEN
        OPEN (UNIT=8, FILE='PRESS.OUT', STATUS='REPLACE', IOSTAT=ierror)

! Calculate output values for all lower volumes.

        DO I = 1,25

! Loop over all cells in volume.

            DO N = 1, n_cells_l(I)

! Get cell index for volume.

                CALL get_cell_index_from_vc(N, vol_index_l(I), cell_index, error)

! Get y coordinate values for all cells in volume.

                CALL get_value_one_cell(YC_index, cell_index, yc, error)

! Find the y axis location of the cell closest to the inlet boundary for all
lower
! (inner) volumes

                low_temp = yc
                IF (low_temp<low_tempold) THEN
                    low_tempold=low_temp
                ENDIF

                low_val(I)=low_tempold

! If cells border with boundary, write (coordinates and) pressure value.

                IF (NPRIN.EQ.1) THEN

! Get P value at cell closest to inlet boundary (25 cells in y direction in
each lower volume)

```

```

        CALL get_value_one_cell(P_index, cell_index, pressure, error)
        CALL get_value_one_cell(YC_index, cell_index, yc, error)
        low_val(I)=yc
100      FORMAT (2X,F14.2)
        NPRIN=25

! The lower bound of the cell centre is y = 0.02605 m at 'all burnt'.

        IF (yc > 0.02605) THEN
            pressure_tot=(pressure_tot+pressure)
        ELSE
            pressure_tot=0.0
        ENDIF
        COUNT=COUNT+1
    ELSE
        NPRIN=NPRIN-1
    ENDIF

ENDDO

! Calculate average pressure for volume at cell centers along boundary.

    pressure_avg = pressure_tot/COUNT

    IF (I.EQ.1) THEN
        pressnoz = pressure_avg
    ENDIF
    WRITE(8,100) pressure_avg

! Reset counters

    COUNT=1
    pressure_tot=0.0

ENDDO

! Values to use in calculating the deformation of the lower axial and lower
end surfaces
! are written to the output file (same values as Lower_1).

    WRITE(8,100) pressnoz
    WRITE(8,100) pressnoz

! Calculate output values for all upper volumes.

    DO J = 1,25

! Loop over all cells in volume.

        DO N = 1, n_cells_u(J)

! Get cell index for volume.

            CALL get_cell_index_from_vc(N, vol_index_u(J), cell_index, error)

! Get y coordinate values for all cells in volume.

```



```

        CALL get_value_one_cell(YC_index, cell_index, yc, error)
        up_temp = yc
        IF (up_temp>up_tempold) THEN
            up_tempold=up_temp
        ENDIF
        up_val(J)=up_tempold

! If cells border with boundary, write pressure value to output file.

        IF (NPRIN.EQ.25) THEN

! Get P value at cell.

            CALL get_value_one_cell(P_index, cell_index, pressure, error)

200        FORMAT (2X,F14.2)
            NPRIN=1

! The upper bound of the cell centre is y = 0.06995 m at 'all burnt'.

            IF (yc < 0.06995) THEN
                pressure_tot=(pressure_tot+pressure)
            ELSE
                pressure_tot=0.0
            ENDIF
            COUNT=COUNT+1
        ELSE
            NPRIN=NPRIN+1
        ENDIF

    ENDDO

! Calculate average pressure for volume at cell centers along boundary.

    pressure_avg = pressure_tot/COUNT

    WRITE(8,200) pressure_avg
    COUNT=1
    pressure_tot=0.0

    IF (J.EQ.1) THEN
        pressnoz = pressure_avg
    ENDIF

    ENDDO

! Values to use in calculating the deformation of the upper axial and upper
end surfaces
! are written to the output file (same values as Upper_1).

    WRITE(8,200) pressnoz
    WRITE(8,200) pressnoz

! Write y values for cell centres closest to inlet boundaries to output file.

```

```

DO K=1,25

300    FORMAT(2X, F11.7)
        WRITE(8,300) low_val(K)
        WRITE(8,300) up_val(K)

    ENDDO

ENDIF

! USER CODE END

RETURN
END SUBROUTINE uout

!*****
!      SUBROUTINE udeform_bc(bc_index, var_index, node_index)
!DEC$ ATTRIBUTES DLLEXPORT :: udeform_bc
!***** DO NOT REMOVE ABOVE LINE FOR MS WINDOWS OS *****
!*****
! copyright (c) 1998 cfd research corp. all rights reserved.
! purpose : For user defined deformation bc.
! Inputs :
!         bc_index - Integer, Boundary Index number
!         var_index - Integer, Variable Index number
!         node_index - Integer, Global Node Index number

! The subroutine is used to specify the boundary deformation motion.
! This subroutine is called for (dx, dy, dz, u, v, w, x, y, z,
! rot_angle(deg), or rot_speed (deg/s)), each boundary node, where user
! has selected user defined evaluation method. At any point of time,
! a node can have two types of motion (translation and rotation).
! In translation, user can specify displacement, velocity or coordinate
! motion. In rotation, user can specify rotation angle in deg or
! rotation speed in deg/s. user can identify this using var_index.
! variable index is available for all these types, refer user manual.
! user can get the no of nodes on bc using get_bc_node_info(),
! get the global indices of nodes on bc using get_global_bc_nodes(),
! get the nodal coordinate values using get_value_one_node(),
! and set the node motion using set_value_one_node()
! For syntax of user access routines see the user manual.
!
!-----
! Include required global variables declared in cfdrc_user module.

    USE cfdrc_user, ONLY : int_p,real_p,string_length,ind_deform_u,pi
    USE cfdrc_user, ONLY : ind_deform_v,ind_upper,ind_lower,first_iter

!***** DO NOT REMOVE FOLLOWING LINE FOR MS WINDOWS OS *****
!      INCLUDE 'cfdrc_include'
!*****

IMPLICIT NONE

! Declaration of arguments of this subroutine

INTEGER(int_p), INTENT(IN) :: bc_index, var_index, node_index

```

```

! Declare required local variables here.

! USER CODE BEGIN

      REAL (real_p) :: time,deform_v,deform_u,mdot_simp,mdot,mdot_tot,&
                        & in_area,out_area,dim_l_tot,sol_rho, &
                        & dim_u_tot,dim_l_ave,dim_u_ave
      REAL (real_p),DIMENSION(100)::pressure,dim_u,dim_l,port_area
      INTEGER (int_p) ::
no_nodes,time_option=2,I,J,ierror,L,LL,N,time_step_no
      LOGICAL :: error
      CHARACTER(len=STRING_LENGTH) :: var_name, wall_name

! USER CODE END

!-----

! Main Program starts here.
! USER CODE BEGIN

! Define constants and some variables being used.
! Solid propellant density

      sol_rho=1665.86

! Initialize variables which are used to measure height of port from
centreline to inlet
! boundary.

      dim_l_ave=0.0
      dim_l_tot=0.0
      dim_u_ave=0.0
      dim_u_tot=0.0

! Open output file which will be used to list calculated mass flow rate for
each volume
! at each timestep.

      OPEN(UNIT=8,FILE='deform.dat',STATUS='REPLACE',IOSTAT=ierror)

      IF (first_iter) THEN

! Get variable indices for deform_v, deform_u.

      var_name = 'DEFORM_V'
      CALL get_var_index(var_name,ind_deform_v,error)
      var_name = 'DEFORM_U'
      CALL get_var_index(var_name,ind_deform_u,error)

! Get BC indices for all deforming boundaries.

      wall_name = 'upper_1'
      CALL get_bc_index(wall_name,ind_upper(1),error)
      wall_name = 'upper_2'
      CALL get_bc_index(wall_name,ind_upper(2),error)
      wall_name = 'upper_3'

```

```

CALL get_bc_index(wall_name,ind_upper(3),error)
wall_name = 'upper_4'
CALL get_bc_index(wall_name,ind_upper(4),error)
wall_name = 'upper_5'
CALL get_bc_index(wall_name,ind_upper(5),error)
wall_name = 'upper_6'
CALL get_bc_index(wall_name,ind_upper(6),error)
wall_name = 'upper_7'
CALL get_bc_index(wall_name,ind_upper(7),error)
wall_name = 'upper_8'
CALL get_bc_index(wall_name,ind_upper(8),error)
wall_name = 'upper_9'
CALL get_bc_index(wall_name,ind_upper(9),error)
wall_name = 'upper_10'
CALL get_bc_index(wall_name,ind_upper(10),error)
wall_name = 'upper_11'
CALL get_bc_index(wall_name,ind_upper(11),error)
wall_name = 'upper_12'
CALL get_bc_index(wall_name,ind_upper(12),error)
wall_name = 'upper_13'
CALL get_bc_index(wall_name,ind_upper(13),error)
wall_name = 'upper_14'
CALL get_bc_index(wall_name,ind_upper(14),error)
wall_name = 'upper_15'
CALL get_bc_index(wall_name,ind_upper(15),error)

wall_name = 'upper_16'
CALL get_bc_index(wall_name,ind_upper(16),error)
wall_name = 'upper_17'
CALL get_bc_index(wall_name,ind_upper(17),error)
wall_name = 'upper_18'
CALL get_bc_index(wall_name,ind_upper(18),error)
wall_name = 'upper_19'
CALL get_bc_index(wall_name,ind_upper(19),error)
wall_name = 'upper_20'
CALL get_bc_index(wall_name,ind_upper(20),error)
wall_name = 'upper_21'
CALL get_bc_index(wall_name,ind_upper(21),error)
wall_name = 'upper_22'
CALL get_bc_index(wall_name,ind_upper(22),error)
wall_name = 'upper_23'
CALL get_bc_index(wall_name,ind_upper(23),error)
wall_name = 'upper_24'
CALL get_bc_index(wall_name,ind_upper(24),error)
wall_name = 'upper_25'
CALL get_bc_index(wall_name,ind_upper(25),error)
wall_name = 'upper_end'
CALL get_bc_index(wall_name,ind_upper(26),error)
wall_name = 'upper_ax'
CALL get_bc_index(wall_name,ind_upper(27),error)
wall_name = 'lower_1'
CALL get_bc_index(wall_name,ind_lower(1),error)
wall_name = 'lower_2'
CALL get_bc_index(wall_name,ind_lower(2),error)
wall_name = 'lower_3'
CALL get_bc_index(wall_name,ind_lower(3),error)
wall_name = 'lower_4'

```

```

CALL get_bc_index(wall_name,ind_lower(4),error)
wall_name = 'lower_5'
CALL get_bc_index(wall_name,ind_lower(5),error)
wall_name = 'lower_6'
CALL get_bc_index(wall_name,ind_lower(6),error)
wall_name = 'lower_7'
CALL get_bc_index(wall_name,ind_lower(7),error)
wall_name = 'lower_8'
CALL get_bc_index(wall_name,ind_lower(8),error)
wall_name = 'lower_9'
CALL get_bc_index(wall_name,ind_lower(9),error)
wall_name = 'lower_10'
CALL get_bc_index(wall_name,ind_lower(10),error)
wall_name = 'lower_11'
CALL get_bc_index(wall_name,ind_lower(11),error)
wall_name = 'lower_12'
CALL get_bc_index(wall_name,ind_lower(12),error)
wall_name = 'lower_13'
CALL get_bc_index(wall_name,ind_lower(13),error)
wall_name = 'lower_14'
CALL get_bc_index(wall_name,ind_lower(14),error)
wall_name = 'lower_15'
CALL get_bc_index(wall_name,ind_lower(15),error)
wall_name = 'lower_16'
CALL get_bc_index(wall_name,ind_lower(16),error)
wall_name = 'lower_17'
CALL get_bc_index(wall_name,ind_lower(17),error)
wall_name = 'lower_18'
CALL get_bc_index(wall_name,ind_lower(18),error)
wall_name = 'lower_19'
CALL get_bc_index(wall_name,ind_lower(19),error)
wall_name = 'lower_20'
CALL get_bc_index(wall_name,ind_lower(20),error)
wall_name = 'lower_21'
CALL get_bc_index(wall_name,ind_lower(21),error)
wall_name = 'lower_22'
CALL get_bc_index(wall_name,ind_lower(22),error)
wall_name = 'lower_23'
CALL get_bc_index(wall_name,ind_lower(23),error)
wall_name = 'lower_24'
CALL get_bc_index(wall_name,ind_lower(24),error)
wall_name = 'lower_25'
CALL get_bc_index(wall_name,ind_lower(25),error)
wall_name = 'lower_end'
CALL get_bc_index(wall_name,ind_lower(26),error)
wall_name = 'lower_ax'
CALL get_bc_index(wall_name,ind_lower(27),error)

```

! Set first iteration to FALSE as indices are only set on first iteration.

```
first_iter = .FALSE.
```

```
ENDIF
```

! Open 'uout' subroutine output file to read cell centre pressures adjacent to inlet in each
! volume.

```

OPEN(UNIT=22, FILE='PRESS.OUT', STATUS='OLD', FORM='FORMATTED')

DO L=1,54

    READ(22,*) pressure(L)

ENDDO

! Read locations of cell centres adjacent to inlet surface, then calculate
upper and lower
! port areas.

DO L=1,25

    READ(22,*) dim_l (L)
    READ(22,*) dim_u (L)
    port_area(L)=pi*(dim_u(L)**2.0-dim_l(L)**2.0)
    dim_l_tot=dim_l_tot+dim_l(L)
    dim_u_tot=dim_u_tot+dim_u(L)

ENDDO

port_area(26)=port_area(25)
dim_l_ave=dim_l_tot/25.0
dim_u_ave=dim_u_tot/25.0

CLOSE(22)

! Using the simple burning law, the grid deformation rate in the y direction
(deform_v) is
! calculated for each lower volume. The mass flow rate (mdot) for each
volume is also
! calculated.

DO J=1,26

    IF (bc_index == ind_lower(J).AND.var_index == ind_deform_v) THEN

        CALL get_time(time, time_step_no, error)
        deform_v=-0.0000219*(pressure(J)**0.45)
        in_area=(2.0*pi*dim_l_ave*0.986)/(2.0*pi)
        mdot_tot=sol_rho*-1*deform_v*in_area
        IF (J==1 .OR. J==25) THEN
            mdot=mdot_tot*(0.022/0.986)
        ELSEIF (J==26) THEN
            mdot=mdot_tot*(0.022/0.986)*0.85
        ELSE
            mdot=mdot_tot*(0.04/0.986)
        ENDIF

300    FORMAT(2X,'lower mass flow rate at t= ',F10.6,' s for ',&
           & I3,' is ',F10.6)
        WRITE(8,300) time, J, mdot
        CALL set_value_one_node(node_index,ind_deform_v,deform_v,error)

    ENDIF

```

```

        ENDDO

! Using the simple burning law, the grid deformation rate in the x direction
! (deform_u) is calculated for the end surface (lower_ax). The mass flow
rate (mdot)
! for each volume is also calculated.

        IF (bc_index == ind_lower(27).AND.var_index == ind_deform_u) THEN

                CALL get_time(time, time_step_no, error)

! At an appropriate iteration, deform_u is set to zero to prevent negative
volume formation.

                deform_u=-0.0000219*(pressure(27)**0.45)
!                deform_u=0.0

! When propellant surface is flush with inner nozzle, there is no more mass
flux component
! from this surface.
                IF (dim_l_ave>0.037) THEN
                        in_area=(pi*((dim_l_ave**2.0)-(0.037**2.0)))/(2.0*pi)
                ELSE
                        in_area=0.0
                ENDIF
                mdot=sol_rho*(-1*deform_u)*in_area
400        FORMAT(2X,'lower mass flow rate at t= ',F10.6,' s for J=27 is
',F10.6)
                WRITE(8,400) time, mdot
                CALL set_value_one_node(node_index,ind_deform_u,deform_u,error)

        ENDIF

! Using the simple burning law, the grid deformation rate in the y direction
(deform_v) is
! calculated for each upper volume. The mass flow rate (mdot) for each
volume is also
! calculated.

        DO I=1,26

                IF (bc_index == ind_upper(I).AND.var_index == ind_deform_v) THEN

                        CALL get_time(time, time_step_no, error)
                        LL=I+27
                        deform_v=0.0000219*(pressure(LL)**0.45)
                        out_area=(2.0*pi*dim_u_ave*0.986)/(2.0*pi)
                        mdot_tot=sol_rho*deform_v*out_area
                        IF (I==1 .OR. I==25) THEN
                                mdot=mdot_tot*(0.022/0.986)
                        ELSEIF (I==26) THEN
                                mdot=mdot_tot*(0.022/0.986)*0.6
                        ELSE
                                mdot=mdot_tot*(0.04/0.986)
                        ENDIF

```

```

500      FORMAT(2X,'upper mass flow rate at t= ',F10.6,' s for ',I3,&
          &' is ',F10.6)
      WRITE(8,500) time, I, mdot

      CALL set_value_one_node(node_index,ind_deform_v,deform_v,error)

      ENDIF

      ENDDO

! Using the simple burning law, the grid deformation rate in the x direction
! (deform_u) is calculated for the end surface (upper_ax). The mass flow
rate (mdot)
! for each volume is also calculated.

      IF (bc_index == ind_upper(27).AND.var_index == ind_deform_u) THEN

          CALL get_time(time, time_step_no, error)

! At an appropriate iteration, deform_u is set to zero to prevent negative
volume formation.

          deform_u=-0.0000219*(pressure(54)**0.45)
!          deform_u=0.0

! When propellant surface is flush with outer portion of nozzle, there is no
more mass
! flux component from this surface.

          IF (dim_u_ave<0.067) THEN
              out_area=(pi*((0.067**2.0)-(dim_u_ave**2.0)))/(2.0*pi)
          ELSE
              out_area=0.0
          ENDIF
          mdot=sol_rho*(-1*deform_u)*out_area
600      FORMAT(2X,'upper mass flow rate at t= ',F10.6,' s for I=27 is
',F10.6)
          WRITE(8,600) time, mdot
          CALL set_value_one_node(node_index,ind_deform_u,deform_u,error)

      ENDIF

! USER CODE END

      RETURN

      END SUBROUTINE udeform_bc

```


Appendix D – User Subroutine for Erosive Burning

```

!*****
!  SUBROUTINE erosive_burning
!
!  by John T. Weaver
!*****
      MODULE cfdrc_user
!*****

      IMPLICIT NONE
      INTEGER, PARAMETER :: int_p  = SELECTED_INT_KIND(8)
      INTEGER, PARAMETER :: string_length = 80
      INTEGER, PARAMETER :: real_p  = SELECTED_REAL_KIND(8)
      INTEGER, PARAMETER :: XDIR = 1, YDIR = 2, ZDIR = 3
! Utility parameters.

      REAL(real_p) , PARAMETER :: zero = 0.0d0, one = 1.0d0, two = 2.0d0, &
& three = 3.0d0, four = 4.0d0, pi = 3.1415926535898d0

! Declare global variables

! USER CODE BEGIN

      INTEGER(int_p) :: ind_deform_v, ind_deform_u, N, cell_index, XC_index, &
& C_index, P_index, YC_index
      INTEGER(int_p), DIMENSION(30) :: ind_upper, ind_lower, n_cells_l, n_cells_u, &
& vol_index_l, vol_index_u
      REAL(real_p), DIMENSION(30) :: low_val, up_val
      LOGICAL :: first_iter = .TRUE.

! USER CODE END

      END MODULE cfdrc_user

!*****
      SUBROUTINE uout(iflag)
!DEC$ ATTRIBUTES DLLEXPORT :: uout
!*****
! copyright (c) 1998 cfd research corp. all rights reserved.
!
! purpose : for customized user output.
!
! iflag: flag indicating calling location.
!
! This routine is called 5 times at different instances of iterative
! cycle indicated by iflag.
!
! iflag :
!         0 - At the beginning (only for dtf reading calls, At this point
!             most of the other data may not be available.
!             users can get variable indices, and may be
!             external reading of files can be done with
this
!             flag = 0)
!         1 - At the beginning of RUN. (At this poin most of the boundary

```

```

!                               conditions, properties are set.
users
!                               should be able to get cell or
boundary
!                               values for different variables.)
!           2 - At the beginning of time step (only for transient problems).
!           3 - At the end of each iteration.
!           4 - At the end of each time step.(only for transient problems).
!           5 - At the end of RUN.
!
! One may use get_value_one_cell to get the values. To get the cell
! indexes, user has to supply the x,y,z locations and use the
! get_cell_index(vc_index,x,y,z,ic_global,error).
!-----

! Include required global variables declared in cfdrc_user module.
  USE cfdrc_user, ONLY : int_p, real_p, string_length
  USE cfdrc_user, ONLY : n_cells_l, vol_index_l, N, cell_index, XC_index,
&
&                               & YC_index, P_index, vol_index_u, n_cells_u,low_val,
&
&                               & up_val

!***** DO NOT REMOVE FOLLOWING LINE FOR MS WINDOWS OS *****
  INCLUDE 'cfdrc_include'
!*****

  IMPLICIT NONE

  INTEGER(int_p), INTENT(IN) :: iflag

! Declare required local variables here.
! USER CODE BEGIN

  REAL(real_p) :: xc, yc,pressure_tot, pressure, pressure_avg,pressnoz,&
& low_temp,low_tempold,up_temp,up_tempold
  CHARACTER(len=string_length) :: vol_name, XC_name = 'XC', &
& YC_name = 'YC', P_name = 'P'
  LOGICAL :: error
  INTEGER(int_p) :: ierror
  INTEGER :: NPRIN,COUNT,I,J,M,K

! USER CODE END

!-----

! Program main body.
! USER CODE BEGIN

! Define Constants

  NPRIN=25
  COUNT=1
  pressure_tot = 0.0

! Retrieve the info that is constant throughout the simulation.

```

```

        IF (iflag == 1) THEN
! Supply name of volume from DTF
        vol_name = '1_lower'

! Get index of volume (vol_index_n)
        CALL get_vc_index(vol_name, vol_index_1(1), error)

! Get remainder of volume indices
        vol_name = '2_lower'
        CALL get_vc_index(vol_name, vol_index_1(2), error)
        vol_name = '3_lower'
        CALL get_vc_index(vol_name, vol_index_1(3), error)
        vol_name = '4_lower'
        CALL get_vc_index(vol_name, vol_index_1(4), error)
        vol_name = '5_lower'
        CALL get_vc_index(vol_name, vol_index_1(5), error)
        vol_name = '6_lower'
        CALL get_vc_index(vol_name, vol_index_1(6), error)
        vol_name = '7_lower'
        CALL get_vc_index(vol_name, vol_index_1(7), error)
        vol_name = '8_lower'
        CALL get_vc_index(vol_name, vol_index_1(8), error)
        vol_name = '9_lower'
        CALL get_vc_index(vol_name, vol_index_1(9), error)
        vol_name = '10_lower'
        CALL get_vc_index(vol_name, vol_index_1(10), error)
        vol_name = '11_lower'
        CALL get_vc_index(vol_name, vol_index_1(11), error)
        vol_name = '12_lower'
        CALL get_vc_index(vol_name, vol_index_1(12), error)
        vol_name = '13_lower'
        CALL get_vc_index(vol_name, vol_index_1(13), error)
        vol_name = '14_lower'
        CALL get_vc_index(vol_name, vol_index_1(14), error)
        vol_name = '15_lower'
        CALL get_vc_index(vol_name, vol_index_1(15), error)
        vol_name = '16_lower'
        CALL get_vc_index(vol_name, vol_index_1(16), error)
        vol_name = '17_lower'
        CALL get_vc_index(vol_name, vol_index_1(17), error)
        vol_name = '18_lower'
        CALL get_vc_index(vol_name, vol_index_1(18), error)
        vol_name = '19_lower'
        CALL get_vc_index(vol_name, vol_index_1(19), error)
        vol_name = '20_lower'
        CALL get_vc_index(vol_name, vol_index_1(20), error)
        vol_name = '21_lower'
        CALL get_vc_index(vol_name, vol_index_1(21), error)
        vol_name = '22_lower'
        CALL get_vc_index(vol_name, vol_index_1(22), error)
        vol_name = '23_lower'
        CALL get_vc_index(vol_name, vol_index_1(23), error)
        vol_name = '24_lower'
        CALL get_vc_index(vol_name, vol_index_1(24), error)
        vol_name = '25_lower'
        CALL get_vc_index(vol_name, vol_index_1(25), error)
        vol_name = '1_upper'

```

```

CALL get_vc_index(vol_name, vol_index_u(1), error)
vol_name = '2_upper'
CALL get_vc_index(vol_name, vol_index_u(2), error)
vol_name = '3_upper'
CALL get_vc_index(vol_name, vol_index_u(3), error)
vol_name = '4_upper'
CALL get_vc_index(vol_name, vol_index_u(4), error)
vol_name = '5_upper'
CALL get_vc_index(vol_name, vol_index_u(5), error)
vol_name = '6_upper'
CALL get_vc_index(vol_name, vol_index_u(6), error)
vol_name = '7_upper'
CALL get_vc_index(vol_name, vol_index_u(7), error)
vol_name = '8_upper'
CALL get_vc_index(vol_name, vol_index_u(8), error)
vol_name = '9_upper'
CALL get_vc_index(vol_name, vol_index_u(9), error)
vol_name = '10_upper'
CALL get_vc_index(vol_name, vol_index_u(10), error)
vol_name = '11_upper'
CALL get_vc_index(vol_name, vol_index_u(11), error)
vol_name = '12_upper'
CALL get_vc_index(vol_name, vol_index_u(12), error)
vol_name = '13_upper'
CALL get_vc_index(vol_name, vol_index_u(13), error)
vol_name = '14_upper'
CALL get_vc_index(vol_name, vol_index_u(14), error)
vol_name = '15_upper'
CALL get_vc_index(vol_name, vol_index_u(15), error)
vol_name = '16_upper'
CALL get_vc_index(vol_name, vol_index_u(16), error)
vol_name = '17_upper'
CALL get_vc_index(vol_name, vol_index_u(17), error)
vol_name = '18_upper'
CALL get_vc_index(vol_name, vol_index_u(18), error)
vol_name = '19_upper'
CALL get_vc_index(vol_name, vol_index_u(19), error)
vol_name = '20_upper'
CALL get_vc_index(vol_name, vol_index_u(20), error)
vol_name = '21_upper'
CALL get_vc_index(vol_name, vol_index_u(21), error)
vol_name = '22_upper'
CALL get_vc_index(vol_name, vol_index_u(22), error)
vol_name = '23_upper'
CALL get_vc_index(vol_name, vol_index_u(23), error)
vol_name = '24_upper'
CALL get_vc_index(vol_name, vol_index_u(24), error)
vol_name = '25_upper'
CALL get_vc_index(vol_name, vol_index_u(25), error)

```

! Get number of cells in each volume (n_cells_n)

```
DO M=1,25
```

```

    CALL get_cells_vc(vol_index_l(M), n_cells_l(M), error)
    CALL get_cells_vc(vol_index_u(M), n_cells_u(M), error)

```

```

        ENDDO

! Get coordinate and variable indices.

! Get variable index of x coordinate.
        CALL get_var_index(XC_name, XC_index, error)

! Get variable index of y coordinate.
        CALL get_var_index(YC_name, YC_index, error)

! Get variable index of pressure P.
        CALL get_var_index(P_name, P_index, error)

        ENDIF
! *****
! Obtain output at end of time step.
! *****

        IF (iflag == 4) THEN
                OPEN (UNIT=8, FILE='PRESS.OUT', STATUS='REPLACE', IOSTAT=ierror)

! Calculate output values for all lower volumes.

                DO I = 1,25

! Loop over all cells in volume.

                        DO N = 1, n_cells_1(I)

! Get cell index for volume.

                                CALL get_cell_index_from_vc(N, vol_index_1(I), cell_index, error)

! Get y coordinate values for all cells in volume.

                                        CALL get_value_one_cell(YC_index, cell_index, yc, error)

! Find the y axis location of the cell closest to the inlet boundary for all
lower
! (inner) volumes

                                low_temp = yc
                                IF (low_temp<low_tempold) THEN
                                        low_tempold=low_temp
                                ENDIF

                                low_val(I)=low_tempold

! If cells border with boundary, write (coordinates and) pressure value.

                                IF (NPRIN.EQ.1) THEN

! Get P value at cell closest to inlet boundary (25 cells in y direction in
each lower volume)

                                        CALL get_value_one_cell(P_index, cell_index, pressure, error)
                                        CALL get_value_one_cell(YC_index, cell_index, yc, error)

```

```

        low_val(I)=yc
100      FORMAT (2X,F14.2)
        NPRIN=25

! The lower bound of the cell centre is y = 0.02605 m at 'all burnt'.

        IF (yc > 0.02605) THEN
            pressure_tot=(pressure_tot+pressure)
        ELSE
            pressure_tot=0.0
        ENDIF
        COUNT=COUNT+1
    ELSE
        NPRIN=NPRIN-1
    ENDIF

ENDDO

! Calculate average pressure for volume at cell centers along boundary.

    pressure_avg = pressure_tot/COUNT

    IF (I.EQ.1) THEN
        pressnoz = pressure_avg
    ENDIF
    WRITE(8,100) pressure_avg

! Reset counters

    COUNT=1
    pressure_tot=0.0

ENDDO

! Values to use in calculating the deformation of the lower axial and lower
end surfaces
! are written to the output file (same values as Lower_1).

    WRITE(8,100) pressnoz
    WRITE(8,100) pressnoz

! Calculate output values for all upper volumes.

    DO J = 1,25

! Loop over all cells in volume.

        DO N = 1, n_cells_u(J)

! Get cell index for volume.

            CALL get_cell_index_from_vc(N, vol_index_u(J), cell_index, error)

! Get y coordinate values for all cells in volume.

            CALL get_value_one_cell(YC_index, cell_index, yc, error)
            up_temp = yc

```

```

        IF (up_temp>up_tempold) THEN
            up_tempold=up_temp
        ENDIF
        up_val(J)=up_tempold

! If cells border with boundary, write pressure value to output file.

        IF (NPRIN.EQ.25) THEN

! Get P value at cell.

            CALL get_value_one_cell(P_index, cell_index, pressure, error)

200         FORMAT (2X,F14.2)
            NPRIN=1

! The upper bound of the cell centre is y = 0.06995 m at 'all burnt'.

            IF (yc < 0.06995) THEN
                pressure_tot=(pressure_tot+pressure)
            ELSE
                pressure_tot=0.0
            ENDIF
            COUNT=COUNT+1
        ELSE
            NPRIN=NPRIN+1
        ENDIF

    ENDDO

! Calculate average pressure for volume at cell centers along boundary.

    pressure_avg = pressure_tot/COUNT

    WRITE(8,200) pressure_avg
    COUNT=1
    pressure_tot=0.0

    IF (J.EQ.1) THEN
        pressnoz = pressure_avg
    ENDIF

    ENDDO

! Values to use in calculating the deformation of the upper axial and upper
end surfaces
! are written to the output file (same values as Upper_1).

    WRITE(8,200) pressnoz
    WRITE(8,200) pressnoz

! Write y values for cell centres closest to inlet boundaries to output file.

    DO K=1,25

```

```

300      FORMAT(2X, F11.7)
        WRITE(8,300) low_val(K)
        WRITE(8,300) up_val(K)

        ENDDO

    ENDIF

! USER CODE END

    RETURN
END SUBROUTINE uout

! *****
      SUBROUTINE udeform_bc(bc_index, var_index, node_index)
!DEC$ ATTRIBUTES DLLEXPORT :: udeform_bc
! ***** DO NOT REMOVE ABOVE LINE FOR MS WINDOWS OS *****
! *****
! copyright (c) 1998 cfd research corp. all rights reserved.
! purpose : For user defined deformation bc.
! Inputs :
!         bc_index - Integer, Boundary Index number
!         var_index - Integer, Variable Index number
!         node_index - Integer, Global Node Index number

! The subroutine is used to specify the boundary deformation motion.
! This subroutine is called for (dx, dy, dz, u, v, w, x, y, z,
! rot_angle(deg), or rot_speed (deg/s)), each boundary node, where user
! has selected user defined evaluation method. At any point of time,
! a node can have two types of motion (translation and rotation).
! In translation, user can specify displacement, velocity or coordinate
! motion. In rotation, user can specify rotation angle in deg or
! rotation speed in deg/s. user can identify this using var_index.
! variable index is available for all these types, refer user manual.
! user can get the no of nodes on bc using get_bc_node_info(),
! get the global indices of nodes on bc using get_global_bc_nodes(),
! get the nodal coordinate values using get_value_one_node(),
! and set the node motion using set_value_one_node()
! For syntax of user access routines see the user manual.
!
! -----
! Include required global variables declared in cfdrc_user module.

      USE cfdrc_user, ONLY : int_p, real_p, string_length, ind_deform_u, pi
      USE cfdrc_user, ONLY : ind_deform_v, ind_upper, ind_lower, first_iter

! ***** DO NOT REMOVE FOLLOWING LINE FOR MS WINDOWS OS *****
      INCLUDE 'cfdrc_include'
! *****

      IMPLICIT NONE

! Declaration of arguments of this subroutine

      INTEGER(int_p), INTENT(IN) :: bc_index, var_index, node_index

! Declare required local variables here.

```



```

! USER CODE BEGIN

      REAL (real_p) :: deform_v,deform_u,sol_rho,time,alpha,beta,&
&
rate_imp,rho_imp,m_flux,mdot_simp,lseg,glterm,tb_imp,&
&
rate,mdot,mdot_tot,in_area,out_area,dim_l_tot,&
& dim_u_tot,dim_l_ave,dim_u_ave
      REAL (real_p),DIMENSION(100)::pressure,dim_u,dim_l,port_area
      INTEGER (int_p) ::
no_nodes,time_option=2,I,J,ierror,L,LL,N,time_step_no
      LOGICAL :: error
      CHARACTER(len=STRING_LENGTH) :: var_name, wall_name

! USER CODE END

!-----

! Main Program starts here.
! USER CODE BEGIN

! Define constants and some variables being used.
! Solid propellant density

      sol_rho=1665.86

! Solid propellant density in Imperial units

      rho_imp=0.0602

! Lenoir-Robillard equation erosive burning constants

      alpha=0.365
      beta=60.0

! Initialize variables which are used to measure height of port from
centreline to inlet
! boundary.

      dim_l_ave=0.0
      dim_l_tot=0.0
      dim_u_ave=0.0
      dim_u_tot=0.0

! Open output file which will be used to list calculated mass flow rate for
each volume
! at each timestep.

      OPEN(UNIT=8,FILE='deform.out',STATUS='REPLACE',IOSTAT=ierror)

      IF (first_iter) THEN

! Get variable indices for deform_v, deform_u.

      var_name = 'DEFORM_V'
      CALL get_var_index(var_name,ind_deform_v,error)

```

```

var_name = 'DEFORM_U'
CALL get_var_index(var_name, ind_deform_u, error)

! Get BC indices for all deforming boundaries.

wall_name = 'upper_1'
CALL get_bc_index(wall_name, ind_upper(1), error)
wall_name = 'upper_2'
CALL get_bc_index(wall_name, ind_upper(2), error)
wall_name = 'upper_3'
CALL get_bc_index(wall_name, ind_upper(3), error)
wall_name = 'upper_4'
CALL get_bc_index(wall_name, ind_upper(4), error)
wall_name = 'upper_5'
CALL get_bc_index(wall_name, ind_upper(5), error)
wall_name = 'upper_6'
CALL get_bc_index(wall_name, ind_upper(6), error)
wall_name = 'upper_7'
CALL get_bc_index(wall_name, ind_upper(7), error)
wall_name = 'upper_8'
CALL get_bc_index(wall_name, ind_upper(8), error)
wall_name = 'upper_9'
CALL get_bc_index(wall_name, ind_upper(9), error)
wall_name = 'upper_10'
CALL get_bc_index(wall_name, ind_upper(10), error)
wall_name = 'upper_11'
CALL get_bc_index(wall_name, ind_upper(11), error)
wall_name = 'upper_12'
CALL get_bc_index(wall_name, ind_upper(12), error)
wall_name = 'upper_13'
CALL get_bc_index(wall_name, ind_upper(13), error)
wall_name = 'upper_14'
CALL get_bc_index(wall_name, ind_upper(14), error)
wall_name = 'upper_15'
CALL get_bc_index(wall_name, ind_upper(15), error)
wall_name = 'upper_16'
CALL get_bc_index(wall_name, ind_upper(16), error)
wall_name = 'upper_17'
CALL get_bc_index(wall_name, ind_upper(17), error)
wall_name = 'upper_18'
CALL get_bc_index(wall_name, ind_upper(18), error)
wall_name = 'upper_19'
CALL get_bc_index(wall_name, ind_upper(19), error)
wall_name = 'upper_20'
CALL get_bc_index(wall_name, ind_upper(20), error)
wall_name = 'upper_21'
CALL get_bc_index(wall_name, ind_upper(21), error)
wall_name = 'upper_22'
CALL get_bc_index(wall_name, ind_upper(22), error)
wall_name = 'upper_23'
CALL get_bc_index(wall_name, ind_upper(23), error)
wall_name = 'upper_24'
CALL get_bc_index(wall_name, ind_upper(24), error)
wall_name = 'upper_25'
CALL get_bc_index(wall_name, ind_upper(25), error)
wall_name = 'upper_end'
CALL get_bc_index(wall_name, ind_upper(26), error)

```

```

wall_name = 'upper_ax'
CALL get_bc_index(wall_name, ind_upper(27), error)
wall_name = 'lower_1'
CALL get_bc_index(wall_name, ind_lower(1), error)
wall_name = 'lower_2'
CALL get_bc_index(wall_name, ind_lower(2), error)
wall_name = 'lower_3'
CALL get_bc_index(wall_name, ind_lower(3), error)
wall_name = 'lower_4'
CALL get_bc_index(wall_name, ind_lower(4), error)
wall_name = 'lower_5'
CALL get_bc_index(wall_name, ind_lower(5), error)
wall_name = 'lower_6'
CALL get_bc_index(wall_name, ind_lower(6), error)
wall_name = 'lower_7'
CALL get_bc_index(wall_name, ind_lower(7), error)
wall_name = 'lower_8'
CALL get_bc_index(wall_name, ind_lower(8), error)
wall_name = 'lower_9'
CALL get_bc_index(wall_name, ind_lower(9), error)
wall_name = 'lower_10'
CALL get_bc_index(wall_name, ind_lower(10), error)
wall_name = 'lower_11'
CALL get_bc_index(wall_name, ind_lower(11), error)
wall_name = 'lower_12'
CALL get_bc_index(wall_name, ind_lower(12), error)
wall_name = 'lower_13'
CALL get_bc_index(wall_name, ind_lower(13), error)
wall_name = 'lower_14'
CALL get_bc_index(wall_name, ind_lower(14), error)
wall_name = 'lower_15'
CALL get_bc_index(wall_name, ind_lower(15), error)
wall_name = 'lower_16'
CALL get_bc_index(wall_name, ind_lower(16), error)
wall_name = 'lower_17'
CALL get_bc_index(wall_name, ind_lower(17), error)
wall_name = 'lower_18'
CALL get_bc_index(wall_name, ind_lower(18), error)
wall_name = 'lower_19'
CALL get_bc_index(wall_name, ind_lower(19), error)
wall_name = 'lower_20'
CALL get_bc_index(wall_name, ind_lower(20), error)
wall_name = 'lower_21'
CALL get_bc_index(wall_name, ind_lower(21), error)
wall_name = 'lower_22'
CALL get_bc_index(wall_name, ind_lower(22), error)
wall_name = 'lower_23'
CALL get_bc_index(wall_name, ind_lower(23), error)
wall_name = 'lower_24'
CALL get_bc_index(wall_name, ind_lower(24), error)
wall_name = 'lower_25'
CALL get_bc_index(wall_name, ind_lower(25), error)
wall_name = 'lower_end'
CALL get_bc_index(wall_name, ind_lower(26), error)
wall_name = 'lower_ax'
CALL get_bc_index(wall_name, ind_lower(27), error)

```

```

! Set first iteration to FALSE as indices are only set on first iteration.

      first_iter = .FALSE.

      ENDIF

! Open 'uout' subroutine output file to read cell centre pressures adjacent
to inlet in each
! volume.

      OPEN(UNIT=22, FILE='PRESS.OUT', STATUS='OLD', FORM='FORMATTED')

      DO L=1,54

        READ(22,*) pressure(L)

      ENDDO

! Read locations of cell centres adjacent to inlet surface, then calculate
upper and lower
! port areas.

      DO L=1,25

        READ(22,*) dim_l (L)
        READ(22,*) dim_u (L)
        port_area(L)=pi*(dim_u(L)**2.0-dim_l(L)**2.0)
        dim_l_tot=dim_l_tot+dim_l(L)
        dim_u_tot=dim_u_tot+dim_u(L)

      ENDDO

      port_area(26)=port_area(25)
      dim_l_ave=dim_l_tot/25.0
      dim_u_ave=dim_u_tot/25.0

      CLOSE(22)

! Using Lenoir-Robillard erosive burning law, the grid deformation rate in
the y direction
! (deform_v) is calculated for each lower volume. The mass flow rate (mdot)
for each volume
! is also calculated.

      DO J=1,26

        IF (bc_index == ind_lower(J).AND.var_index == ind_deform_v) THEN

          CALL get_time(time, time_step_no, error)
          rate=0.0000219*(pressure(J)**0.45)

          IF (J==1) THEN
            lseg=0.964
          ELSEIF (J==25) THEN
            lseg=0.022
          ELSEIF (J==26) THEN
            lseg=0.986
          
```

```

ELSE
  lseg=0.964-(0.022+(J-2)*0.04)
ENDIF
mdot_simp=sol_rho*rate*(2.0*pi*(dim_u_ave+dim_l_ave)*0.986)*lseg

! Change to Imperial units

rate_imp=rate*39.37
m_flux=(mdot_simp/port_area(J))*0.001422334
glterm=(m_flux**0.8)/((lseg*39.37)**0.2)
tb_imp=rate_imp*alpha*glterm*exp(-
1*(beta*rate_imp*rho_imp/m_flux))

! Change to metric units and solve for regression rate and mass flow rate.

deform_v=-1*(rate+(tb_imp/39.37))
in_area=(2.0*pi*dim_l_ave*0.986)/(2.0*pi)
mdot_tot=sol_rho*-1*deform_v*in_area
IF (J==1 .OR. J==25) THEN
  mdot=mdot_tot*(0.022/0.986)
ELSEIF (J==26) THEN
  mdot=mdot_tot*(0.022/0.986)*0.85
ELSE
  mdot=mdot_tot*(0.04/0.986)
ENDIF

300   FORMAT(2X,'lower mass flow rate at t= ',F10.6,' s for ',&
      & I3,' is ',F10.6)
      WRITE(8,300) time, J, mdot
      CALL set_value_one_node(node_index,ind_deform_v,deform_v,error)
ENDIF

ENDDO

! Using the simple burning law, the grid deformation rate in the x direction
! (deform_u) is calculated for the end surface (lower_ax). The mass flow
rate (mdot)
! for each volume is also calculated.

IF (bc_index == ind_lower(27).AND.var_index == ind_deform_u) THEN

  CALL get_time(time, time_step_no, error)

! At an appropriate iteration, deform_u is set to zero to prevent negative
volume formation.

  deform_u=-0.0000219*(pressure(27)**0.45)
!   deform_u=0.0

  in_area=(2.0*pi*dim_l_ave*0.986)/(2.0*pi)
  mdot_tot=sol_rho*(-1*deform_u)*in_area

! When propellant surface is flush with inner nozzle, there is no more mass
flux component
! from this surface.
  IF (dim_l_ave>0.037) THEN

```

```

        in_area=(pi*((dim_l_ave**2.0)-(0.037**2.0))/(2.0*pi)
    ELSE
        in_area=0.0
    ENDIF
    mdot=sol_rho*(-1*deform_u)*in_area
400    FORMAT(2X,'lower mass flow rate at t= ',F10.6,' s for J=27 is
',F10.6)
    WRITE(8,400) time, mdot
    CALL set_value_one_node(node_index,ind_deform_u,deform_u,error)

ENDIF

! Using Lenoir-Robillard erosive burning law, the grid deformation rate in
the y direction
! (deform_v) is calculated for each upper volume. The mass flow rate (mdot)
for each volume
! is also calculated.

DO I=1,26

    IF (bc_index == ind_upper(I).AND.var_index == ind_deform_v) THEN

        CALL get_time(time, time_step_no, error)
        LL=I+27
        rate=0.0000219*(pressure(LL)**0.45)
        IF (I==1) THEN
            lseg=0.964
        ELSEIF (I==25) THEN
            lseg=0.022
        ELSEIF (I==26) THEN
            lseg=0.986
        ELSE
            lseg=0.964-(0.022+(I-2)*0.04)
        ENDIF
        mdot_simp=sol_rho*rate*(2.0*pi*(dim_u_ave+dim_l_ave)*0.986)*lseg

! Change to Imperial units

        rate_imp=rate*39.37
        m_flux=(mdot_simp/port_area(I))*0.001422334
        glterm=(m_flux**0.8)/((lseg*39.37)**0.2)
        tb_imp=rate_imp*alpha*glterm*exp(-
1*(beta*rate_imp*rho_imp/m_flux))

! Change to metric units

        deform_v=(rate+(tb_imp/39.37))
        out_area=(2.0*pi*dim_u_ave*0.986)/(2.0*pi)
        mdot_tot=sol_rho*deform_v*out_area
        IF (I==1 .OR. I==25) THEN
            mdot=mdot_tot*(0.022/0.986)
        ELSEIF (I==26) THEN
            mdot=mdot_tot*(0.022/0.986)*0.6
        ELSE
            mdot=mdot_tot*(0.04/0.986)
        ENDIF
    
```

```

500      FORMAT(2X,'upper mass flow rate at t= ',F10.6,' s for ',I3,&
          &' is ',F10.6)
      WRITE(8,500) time, I, mdot

      CALL set_value_one_node(node_index,ind_deform_v,deform_v,error)

      ENDIF

      ENDDO

! Using the simple burning law, the grid deformation rate in the x direction
! (deform_u) is calculated for the end surface (upper_ax). The mass flow
rate (mdot)
! for each volume is also calculated.

      IF (bc_index == ind_upper(27).AND.var_index == ind_deform_u) THEN

      CALL get_time(time, time_step_no, error)

! At an appropriate iteration, deform_u is set to zero to prevent negative
volume formation.

      deform_u=-0.0000219*(pressure(54)**0.45)
!      deform_u=0.0

! When propellant surface is flush with outer portion of nozzle, there is no
more mass
! flux component from this surface.

      IF (dim_u_ave<0.067) THEN
        out_area=(pi*((0.067**2.0)-(dim_u_ave**2.0)))/(2.0*pi)
      ELSE
        out_area=0.0
      ENDIF
      mdot=sol_rho*(-1*deform_u)*out_area
600      FORMAT(2X,'upper mass flow rate at t= ',F10.6,' s for I=27 is
',F10.6)
      WRITE(8,600) time, mdot
      CALL set_value_one_node(node_index,ind_deform_u,deform_u,error)

      ENDIF

! USER CODE END

      RETURN

      END SUBROUTINE udeform_bc

```

Appendix E – Technical Papers Supporting Thesis

MODELING OF THE CHAMBER FLOWFIELD IN A ROD-AND-TUBE CONFIGURATION SOLID ROCKET MOTOR

J.T. Weaver^{*}, J.E.D. Gauthier[†], Carleton University
R.A. Stowe[‡], Defence Research and Development Canada – Valcartier

NOMENCLATURE

a	- empirical constant influenced by the ambient grain temperature
C_p	- specific heat capacity for propellant combustion gas
C_{ps}	- specific heat capacity for solid propellant
G	- mass flux
K	- turbulent kinetic energy
L	- length from the head of the grain
$\bar{L}$	- normalized length
n	- exponent in simple burning rate law
P	- chamber pressure (static)
Pr	- Prandtl number
Pr_t	- turbulent Prandtl number
r_b	- total burning rate of propellant
T	- temperature
T_f	- flame temperature
T_i	- initial solid propellant temperature
T_s	- burning propellant surface temperature
u	- x-component of gas flow velocity
v	- y-component of gas flow velocity
Y^+	- surface spatial coordinate
α	- erosive constant
β	- erosive constant
ϵ	- turbulent diffusivity
μ	- gas viscosity
ρ	- density of gaseous products of combustion
ρ_p	- density of solid propellant

ABSTRACT

A CFD simulation of the flowfield within a rod-and-tube solid propellant rocket motor has been developed. It couples the fluid dynamics and heat transfer of the gas flowfield within the rocket port to the nozzle and the regression rate of the propellant. The propellant regression is initially described with an empirical power law relationship and is then described with a provision for erosive burning phenomena. The erosive burning model is based on a phenomenological heat transfer approach. By incorporating regression into the model, the information provided by the resulting time-accurate solution will enable a much improved understanding of the flow phenomena within this rod-and-tube grain motor, which in turn will help in the design of both the motor and the nozzle.

INTRODUCTION

Currently, DRDC Valcartier is designing a prototype rod-and-tube configuration solid propellant rocket motor that will propel a hypersonic velocity missile. This configuration will incorporate a very low port-to-throat area ratio, which in turn results in very high velocity propellant gas traveling across burning propellant surfaces, particularly near the nozzle end of the rocket. Typically the phenomenon known as erosive burning is present in low port-to-throat ratio motors. An extensive number of experimental and modeling studies which describe the erosive burning phenomenon have been made, many of which are summarized by Razdan and Kuo¹. Also anticipated during combustion of this solid rocket motor are particularly high chamber pressures, which tend to reduce the effect of erosive burning and, as such, the regression of the propellant grain tends to be more even and more closely follows the simple burning law.

While numerical models are available to design and analyze solid propellant rocket motors and nozzles, many of them provide solutions based on the assumption of quasi-steady flow. Due to the high pressure, high velocity and highly transient nature of

^{*} Graduate Student, Aerospace Engineering

[†] Professor, Aerospace Engineering, Member AIAA

[‡] Scientist, Propulsion Group, Senior Member AIAA

Copyright © 2003 by the Department of National Defence, Canada. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

the flows expected in the motor under design, it is believed that a CFD simulation will better model the time dependent phenomena that occur during the functioning of a motor of this type.

BACKGROUND

Lenoir and Robillard² theorized that the erosive component of the burning rate is proportional to the heat transfer rate from the core of the combustion gas stream to the propellant wall. This relation, although dependent upon two free constants (α , β) for a given propellant, has provided a simple erosive burning model which has been widely used. The total burning rate is expressed as:

$$r_b = aP^n + \frac{\alpha G^{0.8}}{L^{0.2} e^{(\beta r_b \rho_p / G)}}$$

Using this relation in combination with a computer application capable of predicting the internal flowfield properties, it is possible to determine the local burning rates of the propellant. With current commercial CFD applications which permit grid deformation during computation, the non steady-state erosive burning can be displayed as it occurs on a local scale.

METHOD

Using CFD-ACE+, a commercial multi-physics computational application developed by CFD Research Corporation, a 2D axisymmetric representation of the port and nozzle has been generated using a multi-block computational mesh of about 45,000 nodes in a structured grid. Key boundary conditions utilized are as per Table 1. A non-reacting inlet mass flow entered the computational domain along the entire length of the grain, perpendicular to the surface of the grid. The products of combustion of the propellant which incorporates GAP-ADN (in thermochemical equilibrium in the chamber) were used to characterize the gas mixture modeled. Density of the gas mixture was calculated using the Ideal Gas Law, while viscosity was calculated according to Sutherland's Law which permitted viscosity to vary as a function of temperature. Specific heats as a function of temperature were calculated using the JANNAF coefficients for each of the species present in the propellant gas. Mass diffusivity is related to viscosity through a Schmidt number of 0.7078. The turbulent Prandtl number (Pr_t) relating the turbulent convective heat transfer to the flow is 0.9. Both of

these default values are based on the work of Razdan and Kuo³. According to White⁴, of note is the apparent validity of $Pr_t=0.9$ for the cases of blowing surfaces or wall suction as well as surfaces through which no flow occurs. Pr for the gas was 0.67 based on output from CEA for the GAP-ADN propellant.

Turbulence model

A review of literature available indicates that the $k - \epsilon$ turbulence model is typically used in simulating solid rocket motor combustion [e.g., Razdan and Kuo¹; Ciucci et al⁵; and Allavilli et al⁶]. Of the low- and high- Reynolds number models that were available, research by Ciaofolo et al⁷ showed that:

- Numerical difficulties (i.e., slow convergence) are widely encountered when fine near-wall grids are used;
- Actual predictions from the low-Reynolds number are often in error in the reattachment region following recirculation; and
- Two-layer models are not easily implemented in general-purpose codes.

TABLE 1
Boundary Conditions

Location	Stagnation Temperature [K]	Stagnation Pressure [Pa]	Steady-State Mass flow of combustion products per unit length [kg/s/m]
Propellant Grain (Inlet)	3225	20685000	43.11
Farfield	300	101325	N/A

Based on these findings, the use of a high-Reynolds number $k - \epsilon$ model appears to be the best option. It is noted that Razdan and Kuo¹ also chose to apply the boundary conditions to the near wall rather than the surface in their turbulent boundary layer model of erosive burning. The RNG $k - \epsilon$ model is derived from the instantaneous Navier-Stokes equations, using a mathematical technique known as Renormalization Group (RNG) methods. It is similar to the standard $k - \epsilon$ model, aside from the values of the constants used in the dissipation rate equation as well as incorporating additional terms and functions in the kinetic energy

and dissipation transport equations [Anon⁸]. These slight changes in the values of the constants are due to the inclusion of Coriolis forces in the derivation of the model. These forces play a significant role in recirculating flows. Since some recirculation will be

present within the flow, particularly in the region aft of the grain near the converging section of the nozzle, the RNG $k - \epsilon$ model is utilized in conjunction with a wall function.

Figure 1 – X component of flowfield velocity within 2D axisymmetric model of rod-and-tube configuration solid rocket motor.

Figure 2 – Static Pressure distribution at aft end of grain and at nozzle in 2D axisymmetric model of rod-and-tube configuration solid rocket motor.

Steady-State Solutions

Prior to commencing a transient simulation of the regression of the propellant's burning surface (i.e., grid regression), a number of steady-state solutions modeling the chamber flowfield were produced. These solutions simulate the conditions within the chamber following start-up and assume that the entire propellant surface is burning at a constant rate (i.e., constant port diameter along the length of the grain). Figure 1 indicates the flowfield velocity at 130 ms after ignition. Because of the high pressure in the motor chamber, a choked flow condition exists at the nozzle throat and the outflow is supersonic (variables are extrapolated). Static pressure varies by less than 20% along the entire length propellant grain prior to the entering the motor chamber just prior to entering the nozzle as displayed in Figure 2.

The internal ballistic steady state values of density, pressure and temperature (stagnation and static) are relatively consistent with the output from ballistic calculation data taken from the Solid Performance Program (SPP), a reference computer program widely used to predict the delivered performance of solid rocket motors[Coats et al⁹]. Although the loss mechanisms which are considered within SPP such as nozzle throat erosion, slag formation and two phase flow were not incorporated within the CFD model which forms the subject of this paper, these losses typically would account for less than 6% of the performance difference between the calculated ideal value and the actual measured value according to Sutton and Biblarz¹⁰. The flow characteristics at the nozzle throat and exit plane were also compared to data taken from the CEA application¹¹ and were in close agreement as can be seen in Table 2.

The recirculation of the modeled flow between the grain and the nozzle throat is highlighted at Figure 3. It was possible to successfully incorporate the mass flow from the propellant face at the aft end of the grain for the case where the aft end of the grain is not inhibited. While the contribution to the mass flow is minimal compared to the remainder of the grain, its incorporation makes a significant difference in the velocity of the

flow parallel to the diverging wall of the nozzle.

TABLE 2
Comparison of CFD-ACE+ Steady State
Results with Reference Calculations

Propertie s (at nozzle throat)	CEA value s	CFD- ACE+ values (at 0 ms) – Post- ignition transie nt	CFD - ACE + value s (at 130 ms)	CFD- ACE + value s (at 270 ms) – 'All burn t'
u [m/s]	1060. 6	1076.6	1061. 6	1057. 5
ρ [kg/m ³]	12.12	13.01	12.67	12.67
T [K]	3033. 4	3024.3	2924. 3	2921. 2
P [MPa]	11.90	12.98	12.14	12.21
C_p [kJ/(kg*K)]	1.906	1.897	1.884	1.884

It was also possible to calculate the mass flow rate of the flowfield through the chamber as a function of distance from the head end of the rocket. This relation, as displayed in Figure 4, is consistent with the anticipated increase in mass flow rate as a function of length down the chamber. The agreement of the CFD-ACE+ results along the length of the grain against those from SPP, CEA and related literature is encouraging and indicative that the values predicted by the CFD code are reliable and should provide a realistic non steady-state solution when the grid regression subroutine is added.

Mass flow from propellant wall

Mass flow from propellant wall inhibited

Figure 3 – The difference in flow velocity is significant due to the influence of the mass flow inlet from the propellant grain perpendicular to the longitudinal axis of the rocket. At the location highlighted in each diagram, the flow velocity is 77.0 m/s in the case where mass is flowing from the perpendicular wall (Figure 3a) and 92.6 m/s when it is not (Figure 3b).

Figure 4 – Mass flow rate as a function of distance from the head end of the grain.

Non Steady-State Solutions

For the non steady-state solutions, a coarser mesh was utilized in the farfield region of the grid. As a result, the model was composed of 28800 cells; even with the reduced number of cells it was possible to maintain the Y^+ values within a range of 30 to 150 as required for high Reynolds number turbulence models incorporating a wall function. Otherwise the same grid layout was utilized in an iterative process to determine the local burning rate and its effect upon mass flow throughout the remainder of the solid rocket model. In order to calculate the local burning rate, the grid portion which modeled

the port of the motor was divided into 24 volumes. At the completion of time steps which varied from 5ms to 100ms (shorter time steps were used near the start of the simulation due to the rapid change in flow characteristics), the average values of the properties in each volume needed to calculate the burning rate were input into a separate computer application. The pertinent results (mass flow rate per grain area in each volume and burning rate) were then input into CFD-ACE+ as inlet boundary conditions in addition to those mentioned in Table 1. The application was then restarted and run until the next time step was reached. The grid deformation capability of CFD-ACE+ was key to this iterative procedure, as the grid representing the grain boundaries were able to move at the rate determined by the iterative evaluation of the appropriate burning rate equations. Figure 5 highlights the deformation of the grid in the region where the aft end of the grain was modeled. In this region deformation in both the X and Y axes was incorporated. Separate runs incorporating the simple burning law and the erosive burning law were completed.

Erosive Burning Constants

The calculation of the first of the free constants for use with the Lenoir-Robillard² erosive burning relation (α) was carried out using the following relation:

$$\alpha = \frac{0.0288 C_p \mu^{0.2} \text{Pr}^{-0.667} \left(\frac{T_f - T_s}{T_s - T_i} \right)}{\rho_p C_{ps}} = 0.365 \frac{\text{in}^{2.8}}{\text{lbm}^{0.8} \cdot \text{s}^{0.2}}$$

Values for all of the propellant properties were taken from the CEA output for GAP-ADN propellant, less the value for the specific heat capacity for solid propellant (C_{ps}) [Kubota¹²], the surface temperature (T_s) [Liau et al¹³] and the gas viscosity (μ) which was the chamber exit value as calculated using CFD-ACE+ for a non-erosive burning case.

Typically the other constant used in the equation, (β), is calculated based on actual interrupted firing of test motor data [Lawrence et al¹⁴]. As this data is not available in the case being examined, a value

Figure 5 – Comparison of grid in region of aft end of grain and converging portion of nozzle. Figure 5a highlights the grid structure for the erosive burning case at $t=0.001$ s while Figure 5b illustrates the grid at $t=0.18$ s.

for β was chosen based on the similarity of properties of the GAP-ADN propellant with the properties of propellants for which β was available. It is well documented [Lawrence et al¹⁴, Razdan and Kuo¹] that the binder used in the propellant has a significant effect on the erosive burning rate. Since GAP is an energetic binder and has a higher enthalpy of formation and burning rate than HTPB or CTPB [Komai et al¹⁵], it was assumed that the erosive burning rate for the GAP propellant will be higher than a propellant incorporating HTPB or CTPB as a binder. According to the work of Lawrence et al¹⁴ for a variety of AP/Aluminum propellants which incorporated a number of binders including UTREZ, CTPB and PBAN the lowest value of β obtained was 60. From their work, it was determined that the lower the value of β , the higher the susceptibility of a given propellant to erosive burning. Thus for the case of a propellant incorporating GAP as an energetic binder, a relatively low value for β was chosen.

As the value for α is slightly higher for the GAP-ADN propellant than the value calculated by Lenoir and Robillard for polysulfide propellants (0.365 vice 0.28), it appears that the effect of erosive burning will be significant for this propellant, particularly

as it is coupled with a β value that is susceptible to erosive burning.

Results

At Figure 6, the characteristic static pressure vs. time curves are compared for the cases where erosive burning was incorporated, against the non-steady solution which incorporated the simple burning law only. Results are consistent with expectations as the increased burning rate, hence increased propellant gas mass flow rate, causes the pressure within the chamber to rise above that of the simple burning case. The increased mass flow rate also results in a shortened burn time for the same quantity of propellant.

A comparison of the grain regression rates is highlighted at Figure 7. Values for an analytical calculation of the erosive and simple burning rates are also included as a baseline comparison. This analytical solution used a much larger time step (0.05s) than that of the non-steady CFD solution (0.01ms to 0.5ms), but provided a useful comparison. Again, the CFD generated results were consistent with those of the analytical solution. A significant difference in the burning pattern was noted between the simple and erosive burning results which was anticipated. The erosive burning rate was highest at the aft end of the grain, as

that is the region of highest crossflow velocity and mass flux. For the simple burning rate, which is dependent upon static pressure only, the rate of regression was highest at the head end of the chamber, which is the location of highest pressure as shown in Figure 6.

Figure 6 – Static Pressure vs time curves which incorporated the erosive burning and simple burning laws. Head end value was from a point which is $0.04 \bar{L}$ (normalized length) and aft end value was taken at a point at $0.99 \bar{L}$. Choked flow continued at the nozzle throat for each of the cases until the time shown.

Figure 7 – Grain regression rate as a function of burning model.

The peak increase in crossflow velocity due to erosive burning was determined to be approximately 45 m/s (8% greater than velocity obtained using simple burning law). This value was obtained early on in the simulation, when the port to throat ratio was lowest ($t=0.01s$). Later during the simulations ($t=0.1s$) the velocities had significantly reduced due to the increased port diameter. Of note, is that during the mid to later stages of the simulated burn, the crossflow velocity for the simple burning case was slightly higher due to the fact that the port

diameter was smaller while the mass flux for the two cases was similar (Figure 8).

Figure 8 –Port center line crossflow velocity with respect to grain location. Peak velocity in this figure is at $0.99 \bar{L}$.

At Figure 9, the crossflow velocity for the erosive burning case at $t=0.01s$ is illustrated. Along the wall, a comparatively low speed flow in the x-direction occurs. It is notable that within a height of $10\mu m$ from the wall, $u=10.1$ m/s and $v=4.4$ m/s, where the y-component of the flow velocity is due to the mass flow rate entering the computational domain. These velocities correlate well with the flow profile used by King¹⁶ in deriving his granular diffusion flame theory in modeling erosive burning.

Figure 9 – Crossflow velocity in aft section of grain and converging portion of nozzle.

The profile of the crossflow velocity across the port was also compared as shown in Figure 10. Boundary layer effects in all of the cases are considerable. The off-center peak to each of the profiles is due to the differing mass flow rates of the combustion products entering the computational domain. The mass flow rate per radian from the outer grain (tube) is greater than that of the inner grain (rod) due to its larger inlet surface area per radian. The

burning rate (grid regression) was apparently unaffected by this difference in mass flow rates.

Figure 10 – Crossflow velocity profile at aft end of grain.

DISCUSSION

The factors that influence erosive burning are many. According to Razdan and Kuo¹, they include, but are not limited to:

- Crossflow Velocity and Pressure
- Normal Burn Rate
- Port size
- Propellant characteristics
- Propellant Initial Temperature
- Threshold velocity

Use of the Lenoir-Robillard² erosive burning model in conjunction with the capabilities of CFD takes into account the first four of these elements, although detailed propellant characteristics (i.e., α and β values) are only available through experimental work. For this model, an assumed value for the β constant was used and may be a significant source of error. It is possible to investigate the effects of propellant initial temperature using this modeling technique. As well, a threshold velocity below which negative erosive burning occurs can be incorporated into the model, but experimental work would be required to determine this threshold velocity.

CONCLUSIONS

Use of a non steady-state CFD solution in determining the internal ballistic characteristics of a rod-and-tube configuration solid propellant rocket motor has shown merit. This is due to the significant peak values in internal pressure and flow velocity that are predicted to occur as well as the difference in burning rates as a function of grain location

which is dependent upon the burning law used. The pressure and velocity peaks are not apparent when solving the flow phenomena using a quasi-steady state approach. Further studies incorporating a developmental rod-and-tube configuration rocket motor compared against CFD results are required for validation purposes and to determine the erosive burning constants. Future development work for the non steady-state model includes updating the iterative procedure so that local pressure at all inlet nodes can be used in calculating the burning rate, and the updated grid regression rates can be applied to each node in the inlet independently and with a smaller time step.

ACKNOWLEDGEMENTS

The authors wish to thank Matt Slaby and Sekhar Radhakrishnan of CFDRC for their contributions in developing the CFD model used. As well, the developers of CEA¹¹ are also thanked for making available their application on a cost-free basis.

REFERENCES

1. Razdan, M.K., and Kuo, K.K., "Erosive Burning of Solid Propellants", Fundamentals of Solid Propellant Combustion – Volume 90 of the AIAA Progress in Astronautics and Aeronautics Series, Chapter 10, pp.515-598, 1984.
2. Lenoir, J.M., and Robillard, G., "A Mathematical Method to Predict the Effects of Erosive Burning in Solid-Propellant Rockets," 6th Symposium (International) on Combustion, Reinhold, New York, pp. 663-667, 1957.
3. Razdan, M.K., and Kuo, K.K., "Erosive Burning Study of Composite Solid Propellants by Turbulent Boundary Layer Approach", AIAA Paper 78-978, 1978.
4. White, F.M., "Viscous Fluid Flow", 2nd Edition, McGraw-Hill, pp.481-494, 1991.
5. Ciucci, A., Iaccarino, G., Moser, R., Najjar, F., Durbin, P., "Simulation of rocket motor internal flows with turbulent mass injection", Center for Turbulence Research Proceedings of the Summer Program 1998, pp. 245 – 266, 1998.
6. Alavilli, P.V.S., Tafti, D., Najjar, F., "The Development of an Advanced Solid – Rocket Flow Simulation Program ROCFLO", AIAA Paper 2000-0824, 2000.
7. Ciofalo, M., Collins, M.W., " $k - \epsilon$ Predictions of Heat Transfer in Turbulent

- Recirculating Flows Using an Improved Wall Treatment”, Numerical Heat Transfer, Part B, Volume 15, pp. 21-47, 1989.
8. Anon., “Fluent 5 – User Guide Volume 2”, Fluent Incorporated, July 1998.
 9. Coats, D.E., Nickerson, G.R., Dang, A.L., Dunn, S.S., Kehtarnavas, H., “Solid Performance Program (SPP)”, AIAA Paper 87-1701, 1987.
 10. Sutton, G.P., Biblarz, O., “Rocket Propulsion Elements”, Seventh Edition”, John Wiley and Sons Inc., pp. 46-101.
 11. Gordon, S., McBride, B.J., “Computer Program for Calculation of Complex Chemical Compositions and Applications”, NASA Reference Publication 1311, 1994.
 12. Kubota, N., “Survey of Rocket Propellants and Their Combustion Characteristics”, Fundamentals of Solid Propellant Combustion – Volume 90 of the AIAA Progress in Astronautics and Aeronautics Series, Chapter 1, pp. 1-52, 1984.
 13. Liao, Y-C., Yang, V., Thynell, S.T., “Modeling of RDX/GAP Propellant Combustion with Detailed Chemical Kinetics”, Solid Propellant Chemistry, Combustion, and Motor Interior Ballistics – Volume 185 of the AIAA Progress in Astronautics and Aeronautics Series, Chapter 2.11, pp. 477-500, 2000.
 14. Lawrence, W.J., Matthews, D.R., Deverall, L.I., “The Experimental and Theoretical Comparison of the Erosive Burning Characteristics of Composite Propellants”, AIAA Paper 68-531, 1968.
 15. Komai, I., Kobayashi, K., Kato, K., “Burning Rate Characteristics of Glycidyl Azide Polymer (GAP) Fuels and Propellants”, Volume 185 of the AIAA Progress in Astronautics and Aeronautics, Chapter 2.9, pp. 455-464, 2000.
 16. King, M.K., “A Model of the Erosive Burning of Composite Propellants”, AIAA Paper 77-930, 1977.

Transient Simulation of Chamber Flowfield in a Rod-and-Tube Configuration Solid Rocket Motor

J.T. Weaver¹, R.A. Stowe², J.E.D. Gauthier¹ and R. Farinaccio²

¹ Carleton University, Ottawa, ON, K1S 5B6, Canada

² Defence R&D Canada - Valcartier, Val-Bélair, QC, G3J 1X5, Canada

Email: jrweaver@storm.ca

NOMENCLATURE

A_T	- empirical constant influenced by the ambient grain temperature
C_p	- specific heat capacity for propellant combustion gas
C_{ps}	- specific heat capacity for solid propellant
G	- mass flux
k	- turbulent kinetic energy
L	- length from the head of the grain
L_{seg}	- axial length of grain segment
$\bar{L}$	- normalized length
$\dot{m}$	- mass flow rate
n	- exponent in simple burning rate law
P	- chamber pressure (static)
Pr	- Prandtl number
Pr_t	- turbulent Prandtl number
r	- total burning rate of propellant
r_o	- simple burning rate
r_e	- erosive burning component of burning rate
r_1	- propellant rod outer radius
r_2	- propellant tube inner radius
t	- time
T	- temperature
T_f	- flame temperature
T_i	- initial solid propellant temperature
T_s	- burning propellant surface temperature
u	- x-component of gas flow velocity
u_τ	- friction velocity
v	- y-component of gas flow velocity
V_{inj}	- surface mass injection velocity
V_{inj}^+	- injection rate
Y^+	- surface spatial coordinate
α	- erosive constant
β	- erosive constant
ε	- turbulent diffusivity

μ	- gas viscosity
ρ	- density of gaseous products of combustion
ρ_p	- density of solid propellant

ABSTRACT

Currently, DRDC Valcartier of the Canadian Department of National Defence is designing a prototype rod-and-tube configuration solid propellant rocket motor that will propel a hypersonic velocity missile. This configuration will incorporate a very low port-to-throat area ratio, which in turn results in very high velocity propellant gas traveling across burning propellant surfaces, particularly near the nozzle end of the rocket. This causes an augmentation in the propellant burning rate. While numerical and lumped parameter models are available to design and analyze solid propellant rocket motors and nozzles, many of them provide solutions based on the assumption of quasi-steady flow. Due to the high pressure, high velocity and highly transient nature of the flows expected in the motor under design, it is believed that a CFD simulation will better model the time-dependent phenomena that occur during the functioning of a motor of this type. This simulation couples the fluid dynamics and heat transfer of the gas flowfield within the rocket port to the nozzle and the regression rate of the propellant. By incorporating the regression of the propellant surfaces into the model, the information provided by the resulting time-accurate solution will enable a much improved understanding of the flow phenomena within this rod-and-tube grain motor and a better prediction of the internal ballistics of the motor, which in turn will help in the design of both the motor and the nozzle.

1. METHOD

Using CFD-ACE+, a commercial multi-physics computational application developed by CFD Research Corporation, a 2D axisymmetric representation of the port and nozzle has been generated using a multi-block computational mesh in a structured grid. A non-reacting inlet mass flow entered the computational domain along the entire length of the grain (L_{seg}), perpendicular to the surface of the grid (Figure 1). The mass flow rate is a function of the burning rate as follows:

$$\dot{m} = r \cdot \rho_p \cdot 2\pi(r_1 + r_2) \cdot L_{seg}$$

where the burning rate r is calculated in accordance with the 1-D Lenoir-Robillard erosive burning law [1]:

$$r = r_o + r_e = a_T P^n + \frac{\alpha G^{0.8}}{L^{0.2} e^{(\beta \cdot r \rho_p / G)}}$$

The Lenoir-Robillard erosive burning law incorporates two free constants (α, β) which are a function of the solid propellant used. The first constant is determined using the following relation:

$$\alpha = \frac{0.0288 C_p \mu^{0.2} \text{Pr}^{-0.667} \left(\frac{T_f - T_s}{T_s - T_i} \right)}{\rho_p C_{ps}}$$

while β is a function of experimental results, such as interrupted firing tests. As experimental results were not available for this propellant and configuration, a range of β values from 53 to 120 were examined based on empirical results obtained from [1] and [2].

The propellant selected for development of the motor under consideration incorporates GAP-ADN as the fuel and oxidizer components. The products of combustion of this propellant (in thermochemical equilibrium in the chamber) as determined with the NASA Chemical Equilibrium and Analysis (CEA) application [3] were used to characterize the gas mixture modeled using a pressure of 20.7 MPa and 3225 K at the boundaries representing the propellant surface. Turbulence was modeled using the Renormalization Group (RNG) $k - \epsilon$ model. Density of the gas mixture was calculated using the ideal gas law, while viscosity was calculated according to Sutherland's Law which permitted viscosity to vary as a function of temperature. Specific heats as a function of temperature were calculated using the JANNAF coefficients for each of the species present in the propellant gas. A Schmidt

number of 1.0 [4] relates mass diffusivity to viscosity. The turbulent Prandtl number (Pr_t) relating the turbulent convective heat transfer to the flow is 0.9 [5] while the Pr for the gas was 0.67 based on output from CEA for the GAP-ADN propellant.

1.1 Steady-State

Prior to commencing a transient simulation of the regression of the propellant's burning surface (i.e., grid regression), a number of steady-state solutions modeling the chamber flowfield were produced. The purpose of this modeling was to validate the spatial differencing scheme used (Upwind Differencing Scheme or UDS), the existence of a grid-independent solution as well as finalizing the detailed development of the grid. Using the RNG $k - \epsilon$ turbulence model, steady-state solutions representing the grain location at 0, 130 and 270 ms during the functioning of the solid rocket motor were completed. These grain locations were calculated assuming uniform grain regression at a rate determined using the simple burning law (r_o). The steady-state solutions simulate the conditions within the chamber following start-up and assume that the entire propellant surface is burning at a constant rate (i.e., uniform port diameter along the length of the grain). In all cases, due to the high pressure in the motor chamber, a choked flow condition exists at the nozzle throat and the outflow is supersonic.

1.2 Non Steady-State

Following the successful completion of the steady-state simulations, separate non steady-state runs incorporating the simple burning law and the erosive burning law with the different β values were completed. Each of these runs involved approximately 1000 time steps and required a minimum of 7 to 10 days to complete using a Dual Pentium III Xeon PC with a 2.2 GHz CPU speed and 1 GB of RAM. All variable properties including the location of the boundary surfaces representing the grain were recalculated by CFD-ACE+ each time step, which ranged from 0.01 ms to 0.5 ms. A shorter time step was used early in the simulation to prevent divergence of the non linear equations being solved.

The initial condition utilized for the simulations was a two-specie air model at standard temperature and pressure. The ignition transient process was not modeled.

Figure 1: Schematic view of rod-and-tube configuration solid rocket motor. Highlighted area (yellow) corresponds to the region of interest represented within the computational domain.

Rather, a simultaneous ignition of all surfaces representing the propellant at $t = 0$ s was assumed.

Figure 2: Pseudocode representing udeform_bc user access routine.

In order to model the chamber flowfield while incorporating the effect of the grain regression, the grid deformation functionality of CFD-ACE+ was utilized. Because of the dependence of the grid deformation rate upon flowfield characteristics, it was necessary to develop a user subroutine to determine this rate each time step for use by the

CFD-ACE+ Solver. This user subroutine is itself composed of two subroutines which were written in Fortran 90 and are linked via an output file (PRESS.OUT). An overview of the structure of the two subroutines (UDEFORM_BC and UOUT), which were used to calculate the grid deformation rate and to measure the cell-centred pressures and locations next to the inlets respectively, is found in Figures 2 and 3.

Figure 3: Pseudocode representing uout user access routine.

2. RESULTS

Although the aim of producing this CFD model was to highlight the effect of erosive burning by comparing transient simulation results using various Lenoir-Robillard erosive burning constants against a transient simulation where the burning of the propellant is modeled using the simple burning rate law, some notable results were also achieved during the steady-state modeling. These included a verification that the solution obtained was grid independent, a validation of the use of a high Reynolds number turbulence model, and a comparison with reference steady-state results generated using the CEA application.

2.1 Steady-State Results

2.1.1 Grid Independence

A numerical solution is regarded as convergent if the solution of the discretized equations tends to the exact solution of the differential equations as the grid spacing tends to zero [6]. As the exact solution is unknown with respect to the CFD modeling being done for the motor under consideration, an alternate accepted method of demonstrating grid convergence through numerical experiment involving successively refined grids was utilized. If the solution converges to a grid independent solution (i.e., the solution obtained is independent of grid density), it is assumed that the numerical method is sound and that the discretization approximations are consistent. The degree of grid refinement used in practice to check for grid independence is to double the resolution of the grid in each direction and compare the results.

Within the steady-state CFD model, the computational domain for the standard density grid was 52356 cells and 209344 cells for the double density grid. The comparison was carried out for a representative solution, the steady-state solution generated at $t = 130$ ms. As can be seen from the results of this numerical experiment, found at Table 1, the differences between solutions at double the resolution are insignificant. Based on the results of this numerical experiment, it is assumed that the solution is grid-independent as well as stable and convergent.

2.1.2 Validity of High Re Turbulence Model

The use of a high Re turbulence model in conjunction with a wall function is limited when the flow departs significantly from the ideal conditions assumed where the law of the wall is valid. Examples of conditions where use of wall functions may result in significant error include [7]:

- Pervasive near wall effects such as flow through a small gap, highly viscous or a low velocity flow;
- Massive transpiration through the wall; and
- Severe pressure gradients leading to boundary layer separation.

In the case of solid rocket motor simulation, the case of massive transpiration through the wall is of concern since the method by which combustion gases are added to the flow can be considered as mass injection through a blowing surface. As the significance of this effect is dependent upon the rate of mass injection [8] in determining whether

	ρ [kg/m ³]	u [m/s]	T [K]	P [MPa]	C_p [kJ/(kg*K)]
Head End					
Standard Density Grid	24.9	18.9	3225	26.1	1.904
Double Density Grid	25.2	18.6	3225	26.6	1.904
Aft End					
Standard Density Grid	20.4	364.5	3192	21.4	1.902
Double Density Grid	20.9	362.0	3193	21.8	1.902
Mid-Nozzle					
Standard Density Grid	1.53	2212.0	1908	1.0	1.770
Double Density Grid	1.48	2258.4	1847	0.9	1.759

Table 1: Results of numerical experiment to determine grid independence. Properties were measured at the centre line of the chamber head end ($0.05 \bar{L}$), chamber aft end ($0.99 \bar{L}$) and midway along the diverging portion of the nozzle. Note that $\bar{L}$ is measured from the head end of the chamber.

the turbulent boundary layer will separate from the wall, this characteristic was investigated using the steady-state model. Should separation occur, the use of a high Re turbulence model with a wall function is not suitable.

The injection rate, V_{inj}^+ , is a dimensionless parameter and is defined as:

$$V_{inj}^+ = \frac{V_{inj}}{u_\tau}$$

For weak injection rates where V_{inj}^+ has a value up to $O(1/10)$, the typical structure of the turbulent boundary layer is retained (i.e., viscous sublayer and fully turbulent zone). For strong injection rates where V_{inj}^+ has a value up to $O(1)$, the presence of both the blowing and turbulent fluctuations of the injected fluid may affect the structure of the turbulent boundary layer.

Using the steady-state solution at 130 ms, the injection velocity (V_{inj}) and injection rate (V_{inj}^+) were determined along the length of chamber from the head end to the commencement of nozzle convergence. These results are displayed at Figure 4. From the calculated values of V_{inj}^+ , it is determined that a weak injection rate exists and thus the typical boundary layer structure is valid. Therefore the use of a high Reynolds number $k-\varepsilon$ model should provide an adequate representation of the turbulence anticipated in the near wall region.

Figure 4: V_{inj} and V_{inj}^+ as a function of $\bar{L}$.

In addition, the injection velocity profile values correlated well with published flow profile data [9] used previous modeling of erosive burning. The

published injection velocity value of 4.4 m/s was consistent with the values determined during the steady-state simulation.

2.1.3 Comparison with Reference Values

The flow characteristics at the nozzle throat and exit plane were compared to data taken from Infinite Area Combustor (IAC) calculations using the CEA application and were found to be in close agreement as seen in Table 2. The slight dissimilarities in the results at the three instances evaluated are believed to be due to the different nozzle inlet profiles that would result from propellant regression.

Properties (at nozzle throat)	CEA values	CFD- ACE+ values (at 0 ms)	CFD- ACE+ values (at 130 ms)	CFD- ACE+ values (at 270 ms) – 'All burnt'
u [m/s]	1060.6	1076.6	1061.6	1057.5
ρ [kg/m ³]	12.12	13.01	12.67	12.67
T [K]	3033.4	3024.3	2924.3	2921.2
P [MPa]	11.90	12.98	12.14	12.21
C_p [kJ/(kg*K)]	1.906	1.897	1.884	1.884

Table 2: Comparison of CFD-ACE+ Steady-State results with reference calculations from the CEA application.

2.2 Non Steady-State Results

2.2.1 Effect on Characteristic Pressure-Time Curves

For the non steady-state or transient simulation, regression of the grid was governed initially using the simple propellant burning law and then compared to the regression governed by the Lenoir-Robillard erosive burning law using three different β values: 53, 60 and 120. Results were consistent with expectations as the increased burning rate, hence increased propellant gas mass flow rate, causes the pressure within the chamber to rise above that of the simple burning case. At Figure 5, the characteristic static pressure vs. time curves were compared at two locations in the chamber for the cases where erosive burning was

	t=1 ms		t=10 ms		t=100 ms	
	Mass flux [kg/m^2s]	EAR	Mass flux [kg/m^2s]	EAR	Mass flux [kg/m^2s]	EAR
Simple burning	3794	N/A	8373	N/A	6495	N/A
Erosive burning						
$\beta = 53$	3972	1.46	9597	1.92	5672	1.41
$\beta = 60$	3941	1.44	9478	1.86	5746	1.38
$\beta = 120$	4046	1.28	8904	1.50	5990	1.19

Table 3: Total mass flux and erosive augmentation ratio compared for simple and erosive burning cases. Erosive Augmentation Ratio (EAR) is defined as the ratio of the total burning rate to the pressure dependent normal burning rate (r/r_o).

Figure 5: Static Pressure vs time curves which incorporated the erosive burning and simple burning laws. The head end value (5a) was measured at the radial centre of the port at $0.05 \bar{L}$ while the aft end value (5b) was at $0.99 \bar{L}$.

incorporated, against the non steady-state solution which incorporated the simple burning law only. The increased mass flow rate resulted in a shortened burn time for the same quantity of propellant. The increase in burning rate due to erosive burning is highlighted in Table 3 where the calculated mass flux and Erosive Augmentation Ratio (EAR) are listed.

For the non steady-state solutions, a coarser mesh was utilized in the farfield region of the grid. As a result, the model was composed of 28800 cells; even with the reduced number of cells it was possible to maintain the Y^+ values within a range of 30 to 150 as required for high Reynolds number turbulence models incorporating a wall function [10].

2.2.2 Crossflow Velocity

The peak increase in crossflow velocity due to erosive burning cases was found to be only 6.2 m/s (1% greater than velocity obtained using simple burning law) at $t=1.5$ ms. This was for the erosive burning case where $\beta = 53$. As anticipated, this value was obtained early in the simulation, when the port-to-throat ratio was lowest. However, for the vast majority of the simulation, the crossflow velocity is highest when the simple burning law is used due to a reduced chamber cross sectional area as a result of a lower rate of grain regression. A comparison of the crossflow velocity profiles for each of the three erosive burning cases and the simple burning law profile is found at Figure 6.

Figure 6: Crossflow velocity as a function of time is illustrated. All values were measured at $0.99 \bar{L}$.

3. DISCUSSION

From the results obtained, it appears that erosive burning is a phenomenon which will have a severe impact upon the actual performance of the solid rocket motor under consideration. While the crossflow velocity is only moderately changed from the simulation incorporating the simple burning law, the peak static chamber pressures predicted when the erosive burning model is incorporated are between 22.3% and 59.9% greater than the case where the simple burning model is used, depending upon the value for the β constant. As well, the trends observed in the pressure time curves (Figure 5) appear to match the trends noted in literature, e.g., higher peak static pressure and shorter burning time.

3.1 Erosive Burning

The occurrence of erosive burning has been noted in numerous instances within published literature and this trend is clearly reproduced by the non steady-state modeling carried out. The factors that influence erosive burning are many. They include [11], but are not limited to :

- Crossflow Velocity and Pressure
- Normal Burn Rate
- Port size
- Propellant characteristics
- Propellant Initial Temperature
- Threshold velocity

Use of the Lenoir-Robillard erosive burning model [1] in conjunction with the capabilities of CFD takes into account the first four of these elements, although detailed propellant characteristics (i.e., β value in particular) are only available through experimental work. For this model, an assumed range of values for the β constant was used and may be a significant source of error. As well, a threshold velocity below which no erosive burning occurs can be incorporated into the model, but experimental work would be required to determine this threshold velocity.

3.2 Nozzle Inlet Profiles

Often, the nozzle inlet characteristics available to the nozzle designer consist of uniform (1-D) properties. While this level of accuracy (typically $\pm 3\%$) is suitable for estimating motor thrust levels, etc., it is not a sufficient level of accuracy for carrying out detailed nozzle design work, for example. The characteristic profiles at the nozzle inlet for the motor under consideration were measured using the CFD model. A view of nozzle inlet profile is shown at Figure 7. As can be seen in Figures 8 - 11, the assumption of uniform properties across the nozzle inlet is a source of potential error, particularly during the early stages of the motor functioning where the crossflow velocity is the highest. Note that a normalized chamber radius of '0' represents the propellant surface of the propellant 'rod' and the value of '1' represents the propellant surface of the propellant 'tube'.

Figure 7: Crossflow velocity in aft section of grain and converging portion of nozzle. Nozzle inlet plane is highlighted.

Figure 8: Local static pressure values for simple and erosive burning cases measured at nozzle inlet. Elapsed burning time is 1.5 ms.

Figure 9: Local static pressure values for simple and erosive burning cases measured at nozzle inlet. Elapsed burning time is 60 ms.

Figure 10: Local crossflow velocity values measured as a function of port height. Elapsed burning times for time 't1' is 1.5 ms and 60 ms for time 't2'.

Figure 11: Local temperature measured as a function of port height. Elapsed burning times for time 't1' is 1.5 ms and 60 ms for time 't2'.

4. CONCLUSION

Use of a transient CFD solution incorporating a 1-D erosive burning law to determine the internal ballistic characteristics of a rod-and-tube configuration solid propellant rocket motor has shown merit. This is due to the significant peak values in internal pressure and flow velocity that are predicted to occur as

well as the difference in burning rates as a function of grain axial location which is dependent upon the burning law used. The pressure and velocity peaks for both the erosive and simple burning cases are not apparent when solving the flow phenomena using a quasi-steady state approach. The ability to carry out a transient simulation from the point where the complete grain surface area has been ignited until the 'all burnt' stage has been reached is invaluable to the rocket motor designer.

5. REFERENCES

- [1]. Lenoir, J.M., and Robillard, G., "A Mathematical Method to Predict the Effects of Erosive Burning in Solid-Propellant Rockets," 6th Symposium (International) on Combustion, Reinhold, New York, pp. 663-667, 1957.
- [2]. Lawrence, W.J., Matthews, D.R., Deverall, L.I., "The Experimental and Theoretical Comparison of the Erosive Burning Characteristics of Composite Propellants", AIAA Paper 68-531, 1968.
- [3]. Gordon, S., McBride, B.J., "Computer Program for Calculation of Complex Chemical Compositions and Applications", NASA Reference Publication 1311, 1994.
- [4]. Lengellé, G., "Model Describing the Erosive Combustion and Velocity Response of Composite Propellants", AIAA Journal, Vol. 13, No.3, pp. 315 -322, March, 1975.
- [5]. Razdan, M.K., and Kuo, K.K., "Erosive Burning Study of Composite Solid Propellants by Turbulent Boundary Layer Approach", AIAA Paper 78-978, 1978.
- [6]. Ferziger, J.H., Peric, M., "Computational Methods for Fluid Dynamics - 3rd Edition", Springer, pp. 21-400, 2002.
- [7]. Anon., "Fluent 5 - User Guide Volume 2", Fluent Incorporated, July 1998.
- [8]. Ciucci, A., Iaccarino, G., Moser, R., Najjar, F., Durbin, P., "Simulation of rocket motor internal flows with turbulent mass injection", Center for Turbulence Research Proceedings of the Summer Program 1998, pp. 245 - 266, 1998.
- [9]. King, M.K., "A Model of the Erosive Burning of Composite Propellants", AIAA Paper 77-930, 1977.
- [10]. Anon., "User and Module Manuals for CFD-GEOM, CFD-ACE+ and CFD-VIEW Version 2003", CFD Research Corporation, 2003.

[11]. Razdan, M.K., and Kuo, K.K., "Erosive Burning of Solid Propellants", Fundamentals of Solid Propellant Combustion – Volume 90 of the AIAA Progress in Astronautics and Aeronautics Series, Chapter 10, pp.515-598, 1984.