Development of a One and Two-Dimensional Model for Calculating Pore Pressure in an Ablating Thermal Sacrificial Liner

Keegan P. Delaney

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
In
Mechanical Engineering

Danesh K. Tafti – Chair
Mark R Paul
Saad A Ragab

May 2, 2007

Development of a One and Two-Dimensional Model For Calculating Pore Pressure in an Ablating Thermal Sacrificial Liner

Keegan P. Delaney

ABSTRACT

Understanding the behavior of charring or decomposing materials exposed to high temperature environments is an essential aspect in rocket design. In particular, the tip of re-entry vehicles and sacrificial rocket nozzle liners are both exposed to extremely high temperatures. This thesis is specifically concerned with better understanding the reaction of sacrificial rocket nozzle liners to these high temperature environments. The sacrificial liners are designed to shield the rocket nozzle from the thermal and chemical effects of the heated exhaust gas that flows through the nozzle. However, in the design process space and weight of the rocket are at a premium. The sacrificial liners need to be designed to be as light and thin as possible, while properly shielding the nozzle from the heated exhaust gases.

The sacrificial liner material is initially impermeable in its virgin state; however, as the liner is exposed to the heated exhaust gases, it chars and the liner material begins to decompose. The decomposition of the liner by heating in the absence of oxygen is known as pyrolysis. At high temperatures, the virgin material will decompose into a solid material (charred liner) and a vapor (pyrolysis gas). The pyrolysis process leads to the flow of pyrolysis gases throughout the porous charred liner. As a result, significant pressures can build within the liner. If the pressures within the liner are high enough, mechanically weak portions of the liner may fracture and break off. Fracturing of the liner could expose the nozzle to the heated exhaust gases, thus jeopardizing the structural integrity of the nozzle. Therefore, it is important to understand the pressure distribution within the sacrificial liners that occurs as a result of the pyrolysis process.

This work describes the code *PorePress*, which solves for steady state and transient pressure distributions in 1- and 2-D axisymmetric geometries that represent sacrificial liners. The *PorePress* code is essentially a 1- and 2-dimensional differential equation solver for mixed, unstructured geometries. Specifically, the code is used for solving a coupled form of the Ideal Gas Law, Conservation of Mass, and Conservation of Momentum Equations, which describe the flow and resulting pressures within liner geometries. The code centers around using Taylor Series expansions to approximate derivatives needed to solve the appropriate differential equations. The derivative approximation process used in *PorePress* is grid transparent, meaning the same method can be used for any combination of quadrilateral (4-sided) or triangular (3-sided) elements in a mesh, without any changes to the code.

Stability issues arise in both the 1- and 2-D *PorePress* solution processes, as a result of the non-linear nature of the coupled equations, high spatial gradients, and large variations in material properties. In the 1-D case stabilization techniques such as:

upwinding, dynamic differencing, under-relaxation, and preconditioning are applied. Meanwhile, in the 2-D case, stabilization techniques such as: inverse weighting and QR factorization of the coefficient matrix, under-relaxation, and preconditioning are applied.

The steady state and transient solution processes for both the 1- and 2-D pore pressure solution processes used in *PorePress* are covered in this thesis, as well as discussion of the resulting pressure distributions. Certain sacrificial liner design considerations that arise as a result of *PorePress* models for sample liner burns are also covered.

Acknowledgements

This paper concludes nearly two years of hard work, and will remain a work that I will be very proud of for years to come. Words cannot describe the gains I have achieved from working on this project. Throughout this project I have learned how to think for myself, and that no task is truly impossible. Throughout the two years there have been ups and downs working on this project; however, I have truly enjoyed everything (including the struggles).

This work bears my name; however, many people have contributed in their own way. First, I would like to thank my advisor Dr. Danesh Tafti. I truly am grateful for all the time and effort that he has put into guiding me in this work. I have been around the graduate school long enough to know that not every student is lucky enough to have an advisor who will help them through hard times as much as Dr. Tafti. Also, after all of his work with graduate students, he has become a master motivator in his own way.

I would also like to thank my sponsoring company Aerojet. Without them I would never have had the chance to work on this project. In particular, I would like to thank my contact at the company: Kent Hennessey. I appreciate all of the time he has taken to teach me about the finer details of the project, and the time to provide me with necessary resources.

I would like to thank my committee members, Drs. Paul and Ragab, for taking the time to review my work and provide suggestions.

Also, this work would not be complete without thanking my HPCFD lab mates: Aroon Viswanathan, Ali Rozati, Evan Sewall, Shi-Ming Li, Mohammad Elyyan, Pradeep Gopalkrishnan, Anant Shah, and Sai Shrinivas Sreedharan . You have all been a true joy to work with, and I could not think of a better group of people to work alongside.

I would like to thank my lovely fiance Whitney Thompson. Words could not express all of the happiness and joy you have brought into my life. Finally, I would like to acknowledge my family for always being there.

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

$\mathbf{A}$	coefficient matrix
$\mathbf{A_{wgt}}$	**A** matrix after inverse weighting
$\mathbf{b}$	difference vector
C	inertial coefficient
CN	condition number
K	permeability
$\dot{m}_{O-G}$	out-gassing rate
M	molecular weight of the pyrolysis gas
P	pyrolysis gas pressure
$\mathbf{Q}$	orthogonal matrix from QR factorization
r	radial distance to a computational cell
R	universal gas constant
$\mathbf{R}$	upper triangular matrix from QR factorization
t	time
T	temperature of the pyrolysis gas
v	pyrolysis gas velocity
V	volume of a computational cell
$\bar{\mathbf{x}}$	linear least-squares approximation of vector of unknowns
$\mathbf{x}$	vector of unknowns
z	distance in the stream-wise direction in cylindrical coordinates
ε_{avg}	average L2 norm of the error vector
λ	eigenvalue of $\mathbf{A^T A}$ matrix
ϕ	porosity of the material in the computational cell
μ	dynamic viscosity of the pyrolysis gas
ρ_g	pyrolysis gas density
ψ	sample variable

subscripts:

i	identifies an element in a mesh
r	radial direction in cylindrical coordinates
z	stream-wise distance in cylindrical coordinates

superscripts:

k	iteration counter
n	time counter

CHAPTER 1. INTRODUCTION

The scope of this thesis is to analyze and describe the theory behind the computer code, *PorePress*, which predicts the pressure response inside a rocket nozzle sacrificial liner exposed to a high temperature environment. Essentially, *PorePress* is a one and two-dimensional porous flow pressure distribution solver, which will be used in concert with previously developed heat transfer and chemistry solvers to describe the behavior of thermally decomposing rocket nozzle liners.

Behavior of charring or decomposing materials exposed to high temperature environments is of particular interest to the combustion, ceramic and aerospace fields. A lot of common combustion processes require careful attention to detail throughout the burning evolution, such as wood and rocket fuel combustion [Moyer and Rindal, 1965]. Ceramic materials are used in such fields as dentistry and ballistics. In both cases the ceramic material must be formed to certain specifications, which are highly dependant on the reactions of the ceramic material to high temperature environments. In the aerospace field, the two most important applications are heat shield materials for the tip of reentry vehicles and sacrificial liners for rocket nozzles.

This work is mainly concerned with the decomposition of rocket nozzle liners as a result of high temperature environments. During the operation of a general rocket, the rocket propellant burns and produces combustion gases. The gases are then accelerated through the rocket nozzle. The combustion gases are at extremely high temperatures, and are subject to interaction with the actual nozzle material itself. Therefore, to prevent

distortion or damage to the actual rocket nozzle, a sacrificial liner is used as a heat shield. Weight of the rocket is at a premium in the rocket design, so ideal sacrificial liners are designed to be as thin as possible while still protecting the nozzle from extreme temperatures. If the sacrificial liners are somehow compromised during a flight the results could be catastrophic. These are some of the reasons why it is very important to understand all of the processes in a rocket nozzle sacrificial liner during the duration of a flight.

In aerospace applications the heat shield or liner material used most commonly is a form of reinforced carbon-phenolic. Carbon-phenolic typically consists of a base of graphitic fibers which are strongly temperature resistant, and a resin of organic molecules containing carbon, hydrogen, and oxygen. [Ahn et al., 1998] At high temperatures the virgin material (reinforced carbon-phenolic) will begin to decompose to solid matter (carbonaceous char or resin) and a vapor (pyrolysis gas). The decomposition of the solid organic carbon-phenolic by heating in the absence of oxygen is known as pyrolysis [Moyer and Rindal, 1965]. A simple schematic of the pyrolysis process can be seen in Figure 1 below.

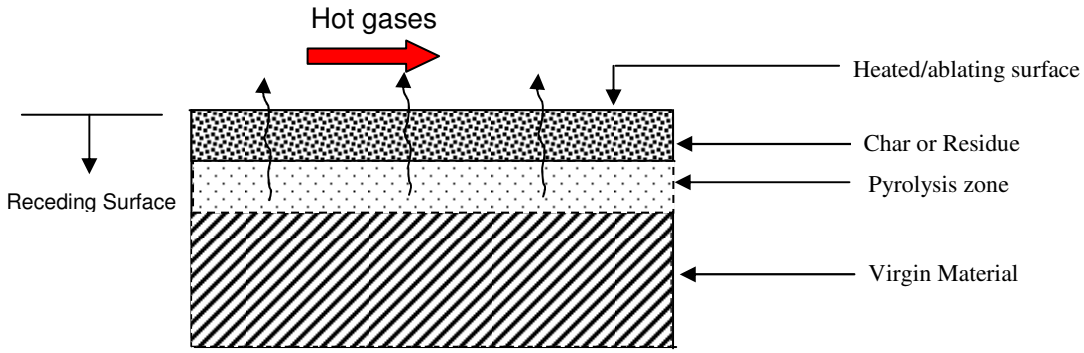

Figure 1: Schematic of generalized pyrolysis process

Once the virgin material begins to decompose under high temperatures, there are many risk factors that could ultimately cause failure of the sacrificial liner. One such risk factor is unwanted chemical reactions within the liner. Mixing between the pyrolysis gas and the carbonaceous char, and mixing between different carbonaceous char constituents could lead to a change in the physical properties of the liner. The sacrificial liners are initially designed to meet very strict thermal and physical requirements, thus unexpected changes to the composition of the liner could lead to unpredictable behavior of the nozzle liner when subjected to high temperatures.

The decomposition of the virgin material leads to the flow of pyrolysis gases throughout the resulting porous carbonaceous char. As the pyrolysis gases flow throughout the porous char, they may react with the char. The result of the gas-char reactions may further erode the char layer, or deposit additional residue. The process where additional residue is added to the pre-existing layer of char is known as coking. Coking is dangerous because it could potentially collapse the porous sacrificial liner by adding weight in a structurally weak portion of the porous medium or distort the thermal properties of the sacrificial liner. It is also possible for different forms of the carbonaceous char to react chemically with each other, forming a different solid material with new thermal and mechanical properties that could react unfavorably to the pyrolysis environment [Moyer and Rindal, 1965].

Significant accumulation of pressure inside the sacrificial liner is another significant risk to the integrity of the sacrificial liner. The main goal of this work is to predict the pressure distribution inside the liner as a result of the pyrolysis process. Once the virgin material pyrolyzes the resulting gas tends to move towards the original heated surface through the porous carbonaceous char. The movement of the gas through the porous medium results in a build up of pressure in the sacrificial liner. The pressure could potentially build up to a significant enough level to cause the sacrificial liner to fracture and break off in certain areas of high pressure, potentially resulting in failure of the liner. It is the potential failure of the sacrificial liner that motivates the need to better understand the pressure distribution inside the liners during the pyrolysis process.

The remainder of this thesis will describe the code *PorePress*, which solves for the pressure distribution inside a sacrificial nozzle liner. The code solves for steady state and transient pressure distributions in nozzle liners modeled by 1- and 2-D axisymmetric geometries. In the 1-D case the *PorePress* code inputs the geometry in the form of unstructured, predetermined gas paths, while in the 2-D case the code solves for the pressure distribution over mixed, unstructured meshes. Thus, there are different solution procedures for the 1- and 2-D cases. Chapters 2 and 3 of this thesis will cover background, solution methodology, verification, and results for both the 1- and 2-D cases, respectively. Chapter 4 will draw conclusions from the work done by the *PorePress* code for both the 1- and 2-D cases. Several appendices have also been added to this thesis. Appendix A displays material property equations specific to this research, and plots of each of these properties. Appendix B contains supplemental figures, which will give the

reader a better understanding of the 1- and 2-D results. Appendix C contains dimensional analysis results used as validation for assumptions made with the 1- and 2-D Momentum Equations. Finally, Appendices D and E are User's Manuals for the 1- and 2-D *PorePress* code. The User's Manuals will provide detail on each of the routines used in the pore pressure solution process and a defined list of variables used in the routines.

CHAPTER 2. 1-D AXISYMMETRIC POROUS FLOW MODEL

The one-dimensional *PorePress* code solves for the pressure distribution by breaking up the rocket nozzle liner, such as on the left of Figure 2.1 below, into one-dimensional paths that are perpendicular to the heated ablating surface. It is assumed that the pyrolyzed gas only flows along these 1-D axisymmetric paths in a 2-D liner. Figure 2.1 below shows a sketch that visually illustrates boundary conditions and gas flow for a typical one-dimensional path and a corresponding computational cell.

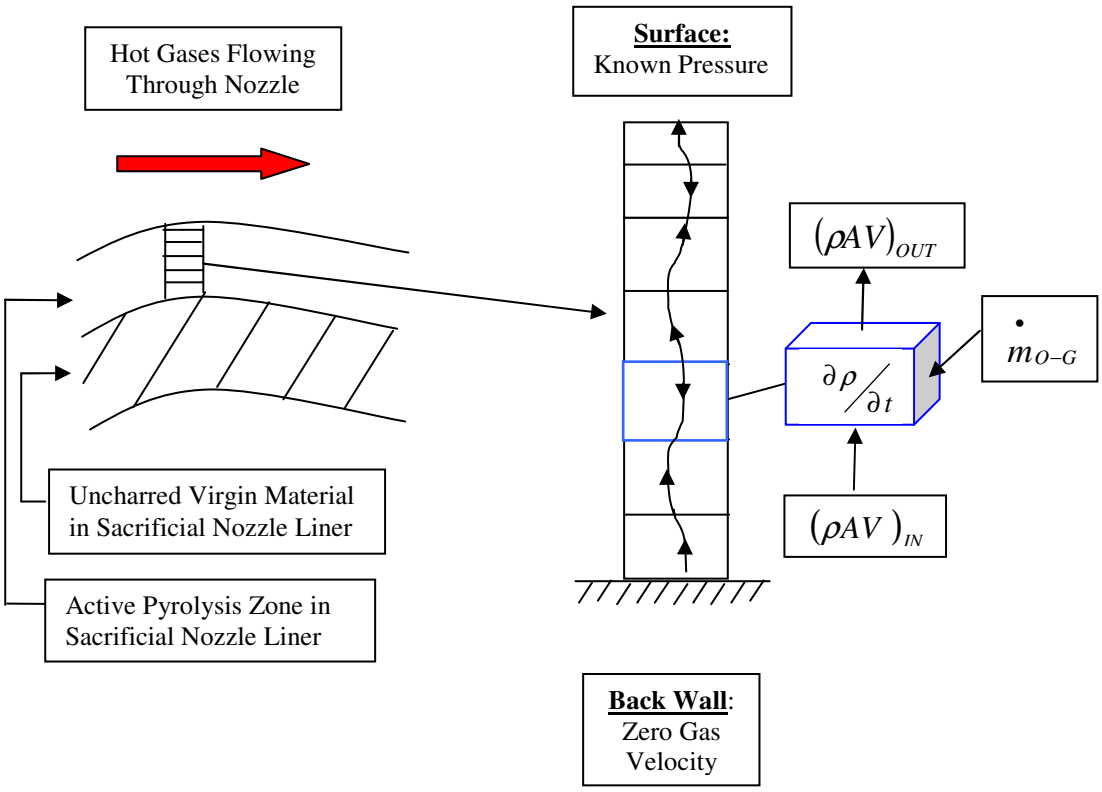

Figure 2.1: Sketch of a path in a nozzle liner (left), pyrolysis gas flow through a typical path (center) and through a computational cell (right)

In Figure 2.1 above, the back wall represents a boundary between the active pyrolysis zone and the inactive virgin material. The active zone consists of previously virgin material that has been charred due to the hot nozzle gases which are interacting with the surface, and has pyrolysis gas flowing through it as a result. In the active zone the pyrolysis gas is free to flow either towards or away from the surface. Once the pyrolysis gas reaches the surface it is assumed that the gas is ejected into the nozzle gas path. It is also assumed that the solid material (carbonaceous char or resin) and the pyrolysis gas are in thermodynamic equilibrium in each cell, which means they both have the same temperature and pressure in each computational cell.

The *PorePress* program has the option for the user to select either the steady state or transient pressure solution. The steady state solver is more stable and computationally cheaper, as it does not integrate the pressure and temperature in time; however, the transient solver yields more accurate results.

2.1 1-D Axisymmetric Porous Flow Model Equations

The *PorePress* program uses the Ideal Gas Law, Conservation of Mass, and Conservation of Momentum equations in a 1-D axisymmetric, coupled manner to solve for pore pressure. Material property equations specific to this work such as: degree of char, porosity, viscosity, permeability, and molecular weight can be seen in Appendix A.

2.1.1 Ideal Gas Law

The pyrolysis gas is assumed to behave as an ideal gas, as described in the equation below:

$$\rho_g = \frac{PM}{RT} \tag{2.1}$$

where P is the pyrolysis gas pressure $\left[\frac{lbf}{ft^2}\right]$, M is the molecular weight of the pyrolysis gas $\left[\frac{lbm}{lbmol}\right]$, R is the universal gas constant $\left[\frac{{}^{lbf}/_{ft^2}\, ft^3}{lbmol - R}\right]$, and T is the temperature of the pyrolysis gas [R]. The resulting pyrolysis gas density, ρ_g, is in $\left[\frac{lbm}{ft^3}\right]$. The pyrolysis gas and solid carbonaceous char material are assumed to be at thermodynamic equilibrium; therefore, the temperature used in the ideal gas law equation is input as the temperature of the solid char.

2.1.2 1-D Axisymmetric Conservation of Mass

The 1-D conservation of mass equation used in *PorePress* is implemented by balancing the rate of production of pyrolyzed gas in a computational cell with the mass flux both entering and leaving the computational cell. As the material in the cells is charred it produces a pyrolyzed gas which adds mass to the path, in the form of out-gassing. Assuming constant porosity inside an element, results in the following form of the 1-D axisymmetric continuity equation for porous media:

$$\phi \frac{\partial \rho_g}{\partial t} + \frac{1}{r} \frac{\partial (r \rho_g v)}{\partial r} = \frac{\dot{m}_{O-G}}{V} \tag{2.2}$$

where $\dot{m}_{O-G}$ is the out-gassing rate $\left[\frac{lbm}{s}\right]$, V is the volume of the computational cell [ft^3], ϕ is the porosity of the material in the computational cell, ρ_g is the gas density $\left[\frac{lbm}{ft^3}\right]$, v *is* the gas velocity either entering or leaving the cell $\left[\frac{ft}{s}\right]$, and the radial ($r$) direction runs along the length of the path. For the applications of *PorePress*, the porosity ranges from 8-26% open volume, and the out-gassing rate typically varies from 0.0- 10^{-6} $^{lbm}/_{s}$. Plots of all the material properties used in *PorePress* can be seen in Appendix A.

2.1.3 1-D Axisymmetric Conservation of Momentum

Lastly, the 1-D axisymmetric Conservation of Momentum equation is used in the Darcy form to describe the pyrolysis gas flow through a porous medium. The initial form of the one-dimensional momentum equation [Nield and Bejan, 1999] can be seen below in Equation 2.3.

$$\frac{1}{\phi} \frac{\partial (\rho v)}{\partial t} + \frac{1}{\phi^2} \nabla \cdot (\rho v v) = -\nabla P - \frac{\mu}{K} v - \frac{C \rho}{2} |v| v \tag{2.3}$$

where μ is the dynamic viscosity of the pyrolysis gas $\left[\frac{lbf-s}{ft^2}\right]$, and K is the permeability of the medium [ft^2], and C is an inertial coefficient [ft^{-1}]. For the applications of *PorePress*, the permeability ranges from 10^{-9}-10^{-14} ft^2 and the dynamic viscosity ranges from 10^{-6} to 10^{-7} $\frac{lbf-s}{ft^2}$. Plots of all the material properties used in *PorePress* can be seen in Appendix A.

One can usually drop the transient term in the momentum equation when the kinematic viscosity of the fluid (μ/ρ_g) is large in comparison to the K/t_o ratio, where t_o is the characteristic time of the process. [Nield and Bejan, 1999]. Inertial effects are also considered to be negligible for this study, where the pyrolysis gas velocity is relatively low. The resulting dimensional analysis used for the Darcy assumption can be seen in Appendix C. The one-dimensional axisymmetric Momentum Equation used in *PorePress* is as follows:

$$-\frac{\partial P}{\partial r} = \frac{\mu}{K} v \tag{2.4}$$

where $\partial P/\partial r$ is the pressure gradient across a given cell in the radial direction.

2.2 1-D Axisymmetric Porous Flow Model Solution Process

The 1-D *PorePress* code receives predetermined pyrolysis gas paths similar to the sketch from Figure 2.1 above. The predetermined gas paths are broken up into multiple

computational cells similar to the cell in Figure 2.1. Each computational cell is input with given thermal and material properties. Each 1-D path has a back wall where the pyrolysis gas velocity is zero and an open surface with a known surface pressure. The pyrolysis gas is free to move in either direction in the path; however, the pyrolysis gas is eventually ejected from the surface into the nozzle hot gas path. Two solution methods are available in the 1-D *PorePress* code: steady state and transient. The steady state method is more stable and computationally cheaper, as it does not integrate the pressure and temperature in time, but the transient solution is more accurate. The *PorePress* code gives the user the option to select steady state, transient, or a combination of the two methods as desired. Sections 2.2.1 and 2.2.2 below will further distinguish the differences between the two solution methods.

For the 1-D case the Ideal Gas Law, Conservation of Mass, and Conservation of Momentum equations are combined and discretized to obtain a pressure solution. A figure showing the combination of the three equations can be seen below in Figure 2.2.

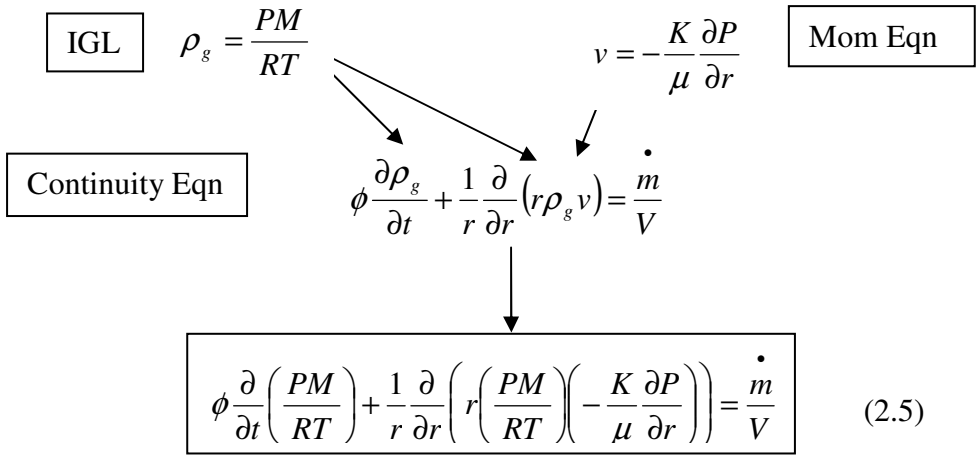

Figure 2.2: 1-D equation combination process and resulting final equation

To obtain the differential equation (Equation 2.5) used to solve the pressure distribution, the Ideal Gas Law in terms of gas density and the momentum equation in terms of gas velocity are substituted into the Continuity Equation. Equation 2.5 can then be expanded out using the chain rule to obtain the final non-linear partial differential equation seen below in Equation 2.6.

$$\phi\frac{\partial P}{\partial t}-\phi\frac{P}{T}\frac{\partial T}{\partial t}-\frac{K}{\mu}P\frac{\partial^2 P}{\partial r^2}-\frac{\partial P}{\partial r}\left(P\frac{\partial}{\partial r}\left(\frac{K}{\mu}\right)-\frac{K}{\mu}\left(\frac{P}{T}\frac{\partial T}{\partial r}-\frac{\partial P}{\partial r}-\frac{1}{r}P\right)\right)=\frac{RT}{M}\frac{\dot{m}_{OG}}{V} \quad (2.6)$$

2.2.1 1-D Steady State Solution

The 1-D steady state pressure solution is achieved by discretizing Equation 2.6 above without the transient terms. The steady state differential equation and the discretized form can be seen below in Equations 2.7 and 2.8 respectively, where the superscript *k* represents an iteration counter.

$$\boxed{-\frac{K}{\mu}P\frac{\partial^2 P}{\partial r^2}-\frac{\partial P}{\partial r}\left(P\frac{\partial}{\partial r}\left(\frac{K}{\mu}\right)-\frac{K}{\mu}\frac{P}{T}\frac{\partial T}{\partial r}+\frac{K}{\mu}\frac{\partial P}{\partial r}+\frac{K}{\mu}\frac{1}{r}P\right)=\frac{RT}{M}\frac{\dot{m}_{OG}}{V}} \quad (2.7)$$

$$
\begin{aligned}
&-\frac{K}{\mu}\bigg|_i P_i^k\left(\frac{P_{i+1}^k-2P_i^{k+1}+P_{i-1}^{k+1}}{\Delta r^2}\right)-P_i^k\left(\frac{P_i^k-P_{i-1}^{k+1}}{\Delta r}\right)\frac{1}{\Delta r}\left(\frac{K}{\mu}\bigg|_{i+1}-\frac{K}{\mu}\bigg|_i\right)\\
&\quad+\frac{K}{\mu}\bigg|_i\frac{P_i^k}{T_i}\left(\frac{P_i^k-P_{i-1}^{k+1}}{\Delta r}\right)\left(\frac{T_i^{n+1}-T_{i-1}^{n+1}}{\Delta r}\right)-\frac{1}{r_i}\frac{K}{\mu}\bigg|_i P_i^k\left(\frac{P_i^k-P_{i-1}^{k+1}}{\Delta r}\right)\\
&\quad-\frac{RT_i}{M}\frac{\dot{m}_{O-G}}{V}-\frac{K}{\mu}\bigg|_i^{n+1}\left(\frac{P_{i/i+1}^{k+1}-P_{i-1/i}^{k+1}}{\Delta r}\right)\left(\frac{P_{i/i+1}^{k+1}-P_{i-1/i}^{k+1}}{\Delta r}\right)=0
\end{aligned}
\tag{2.8}
$$

One important note from Equation 2.8 is that all of the first derivatives are discretized by a first order accurate upwinding scheme (element *i* minus element *i*-1) as opposed to second order accurate central differences (element *i*+1 minus element *i*-1). The upwinding is applied to add stability by smoothing out the large gradients that occur over the paths. More detail on the large gradients can be seen below in the 1-D Results Section 2.4.

The last term in the left hand side of Equation 2.8 represents a dynamic difference term, which is used to add stability to the non-linear equation, by means of increasing diagonal dominance in the coefficient matrix. The pressure derivatives of this term are discretized with a first order forward $\{(P_{i+1}^k-P_i^{k+1})(P_{i+1}^k-P_i^k)\}$ or backward $\{(P_i^{k+1}-P_{i-1}^{k+1})(P_i^k-P_{i-1}^{k+1})\}$ difference depending on the pressure distribution around a current element, *i*. Equation 2.8 is then solved for in terms of the pressure at the next iteration, P_i^{k+1}, to result in the final discretized form of the equation:

$$P_i^{k+1} = \left[\begin{array}{l} P_i^k \left(P_{i+1}^k + P_{i-1}^{k+1} \right) + P_i^k \left(P_i^k - P_{i-1}^{k+1} \right) \dfrac{\mu}{K}\Big|_i \left(\dfrac{K}{\mu}\Big|_{i+1} - \dfrac{K}{\mu}\Big|_i \right) \\ -\dfrac{P_i^k}{T_i} \left(P_i^k - P_{i-1}^{k+1} \right) \left(T_i^{n+1} - T_{i-1}^{n+1} \right) + \dfrac{\Delta r}{r_i} P_i^k \left(P_{i+1}^k - P_i^k \right) + \Delta r^2 \dfrac{RT_i}{M} \dfrac{\dot{m}_{O-G}}{V} \\ \pm P_{i/i+1}^{k+1} \left(P_{i/i+1}^{k+1} - P_{i-1/i}^{k+1} \right) \end{array} \right] \div \left[\begin{array}{l} 2P_i^k \pm \left(P_{i/i+1}^{k+1} - P_{i-1/i}^{k+1} \right) \end{array} \right] \qquad (2.9)$$

Upon discretization, Equation 2.9 is solved iteratively over each individual computational cell until the average L1 norm of the residual vector converges to standards set by the user in the main program *PorePress*. The Ideal Gas Law and Conservation of Momentum equations can be used with the converged pressure solution to find the pyrolysis gas density and velocity respectively, by means of Equations 2.1 and 2.4 above, respectively.

2.2.2 1-D Transient Solution

The 1-D transient solution process is similar to the aforementioned 1-D steady state process, but with a different form of the Conservation of Mass Equation. In the transient solution process the entire Equation 2.6, from above, is discretized into the following implicit form:

$$\phi\frac{P_i^{k+1}-P_i^n}{\Delta t}-\phi\frac{P_i^k}{T_i^{n+1}}\left(\frac{T_i^{n+1}-T_i^n}{\Delta t}\right)-\left.\frac{K}{\mu}\right|_i P_i^k\left(\frac{P_{i+1}^k-2P_i^{k+1}+P_{i-1}^{k+1}}{\Delta r^2}\right)$$

$$-P_i^k\left(\frac{P_i^k-P_{i-1}^{k+1}}{\Delta r}\right)\frac{1}{\Delta r}\left(\left.\frac{K}{\mu}\right|_{i+1}-\left.\frac{K}{\mu}\right|_i\right)+\left.\frac{K}{\mu}\right|_i\frac{P_i^k}{T_i^{n+1}}\left(\frac{P_i^k-P_{i-1}^{k+1}}{\Delta r}\right)\left(\frac{T_i^{n+1}-T_{i-1}^{n+1}}{\Delta r}\right) \qquad (2.10)$$

$$-\frac{1}{r_i}\left.\frac{K}{\mu}\right|_i P_i^k\left(\frac{P_{i+1}^k-P_i^k}{\Delta r}\right)-\frac{RT_i}{M}\frac{\dot{m}_{O-G}}{V}-\left.\frac{K}{\mu}\right|_i^{n+1}\left(\frac{P_{i/i+1}^{k+1}-P_{i-1/i}^{k+1}}{\Delta r}\right)\left(\frac{P_{i/i+1}^{k+1}-P_{i-1/i}^{k+1}}{\Delta r}\right)=0$$

where the superscript n represents the time step (n is the previous time step and n+1 is the current time step), and the superscript k represents the iteration number (k is the previous iteration and k+1 is the current iteration). As the iteration number k tends to infinity, it approaches n+1 which is the converged solution for the time step. Equation 2.10 is then solved for in terms of the pressure at the current time step, and next iteration, P_i^{k+1}, to result in the final discretized form of the equation:

$$P_i^{k+1}=\left[\ \phi\frac{\Delta r^2}{\Delta t}\left.\frac{\mu}{K}\right|_i P_i^n+\phi\frac{\Delta r^2}{\Delta t}\left.\frac{\mu}{K}\right|_i P_i^k\left(\frac{T_i^{n+1}-T_i^n}{\Delta t}\right)+P_i^k\left(P_{i+1}^k+P_{i-1}^{k+1}\right)\right.$$

$$+P_i^k\left(P_i^k-P_{i-1}^{k+1}\right)\left.\frac{\mu}{K}\right|_i\left(\left.\frac{K}{\mu}\right|_{i+1}-\left.\frac{K}{\mu}\right|_i\right)-\frac{P_i^k}{T_i^{n+1}}\left(P_i^k-P_{i-1}^{k+1}\right)\left(T_i^{n+1}-T_{i-1}^{n+1}\right)$$

$$\left.+\frac{\Delta r}{r}P_i^k\left(P_{i+1}^k-P_i^k\right)+\frac{\Delta r^2}{\Delta t}\left.\frac{\mu}{K}\right|_i\frac{RT_i}{M}\frac{\dot{m}_{O-G}}{V}\pm P_{i/i+1}^{k+1}\left(P_{i/i+1}^{k+1}-P_{i-1/i}^{k+1}\right)\ \right]$$

$$\div\left[\ \phi\frac{\Delta r^2}{\Delta t}\left.\frac{\mu}{K}\right|_i+2P_i^k\pm\left(P_{i/i+1}^{k+1}-P_{i-1/i}^{k+1}\right)\ \right] \qquad (2.11)$$

Equation 2.11 is iterated until the average L1 norm of the residual vector reaches a suitable value as defined by the user in the main program *PorePress*.

Over the course of the transient solution the mesh surface is constantly burning back due to temperature effects from the heated nozzle exhaust gases. As the surface elements are continuously heated, they lose density which can lead to the element fracturing from the rest of the liner and getting swept off in the stream of hot gases. Besides the surface effects, as time progresses the temperature increases in the interior as a result of conduction and the pyrolysis zone moves deeper into the virgin material. As far as the mesh is concerned, these physical processes result in previously interior elements becoming surface elements, previously inactive elements becoming active, and previously active elements no longer existing. The transient *PorePress* code accounts for the dynamic nature of the mesh by reading in different meshes at each time step, and reevaluating boundaries, active zones and boundary conditions.

2.3 1-D Axisymmetric Porous Flow Model Verification

The following sample problem from the appendix of the CMA 90S manual [Ross and Stroebel, 1990] was used to test the accuracy of the 1-D steady state and transient *PorePress* code solutions.

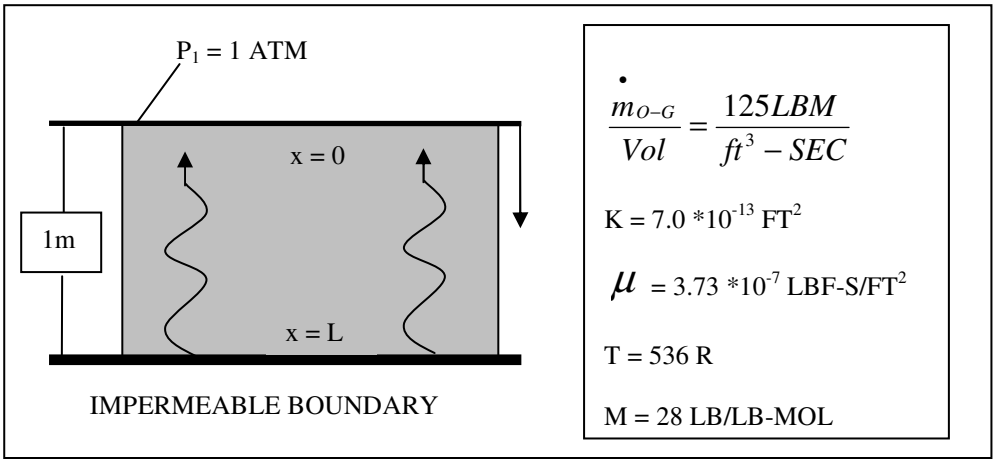

Figure 2.3: Sample problem with constant out-gassing over a 1m long block

The verification problem in Figure 2.3 is a one meter long block with specified material properties. This problem is similar to 1-D applications of *PorePress*, because it has a specified pressure on one surface, an impenetrable back wall on the opposite surface, and a specified out-gassing rate in the material. The boundary conditions for this problem are: the gas velocity is zero at the impermeable boundary (v(x=L) = 0) and specified pressure at the opposite boundary (P(x=0) = P_1).

2.3.1 Steady State Solution Verification

The steady state analytical pressure solution for the problem represented in Figure 2.3 above is:

$$P^2 = \frac{2\mu RT}{KM}\frac{\dot{m}_{O-G}}{V}\left(Lx - \frac{x^2}{2}\right) + P_1^2 \qquad (2.12)$$

where $\dot{m}_{O-G}/V$ is the out-gassing rate divided by the cell volume (or the specified decomposition rate), and the other variables are the same as previously noted. Once the pressure distribution throughout the path is known, the steady state analytical velocity solution for this problem is:

$$v = -\frac{K}{\mu}\frac{\partial P}{\partial x} \tag{2.13}$$

A comparison of the steady state analytical solution and the steady state solution from *PorePress* for both pyrolysis gas velocity and pressure can be seen below in Figures 2.4 and 2.5.

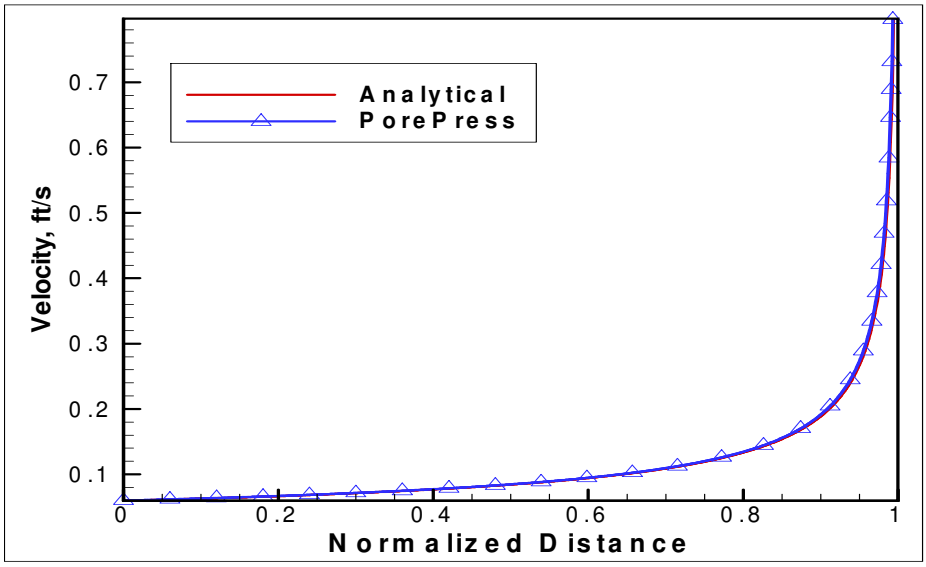

Figure2.4: Verification problem: analytical and calculated pyrolysis gas velocity solution comparison

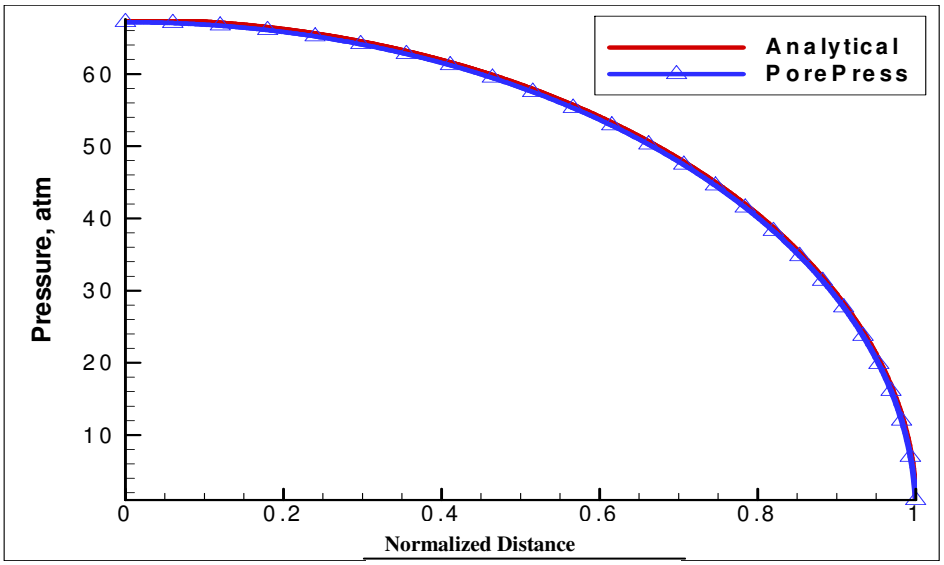

Figure 2.5: Verification problem with out-gassing rate of 125 lbm/s: analytical and calculated pyrolysis gas pressure solution comparison

2.3.2 Transient Solution Verification

Unlike the 1-D steady state case, there is no available analytical transient solution for the problem in Figure 2.3 above. Therefore, the *PorePress* transient solution was compared to the *FLUENT* transient solution for the sample problem in Figure 2.3 above. Comparing *PorePress* and *FLUENT* solutions also serves as validation for using the Darcy form of the Momentum Equation (2.4) in *PorePress*, as *FLUENT* solves the full form of the 1-D Momentum Equation (2.3).

The solution was tested at two volumetric out-gassing rates: 10 lbm/ft^3-s and 125 lbm/ft^3-s, and two constant porosities: 0.10 (10% open volume) and 0.20 (20% open volume). Both the out-gassing rates and porosities represent common values found in the

input files for the intended applications of *PorePress*. Sample comparisons between the *PorePress* (PP) solution and the *FLUENT* (FL) solution can be seen below for two different out-gassing values in Figures 2.6 and 2.7.

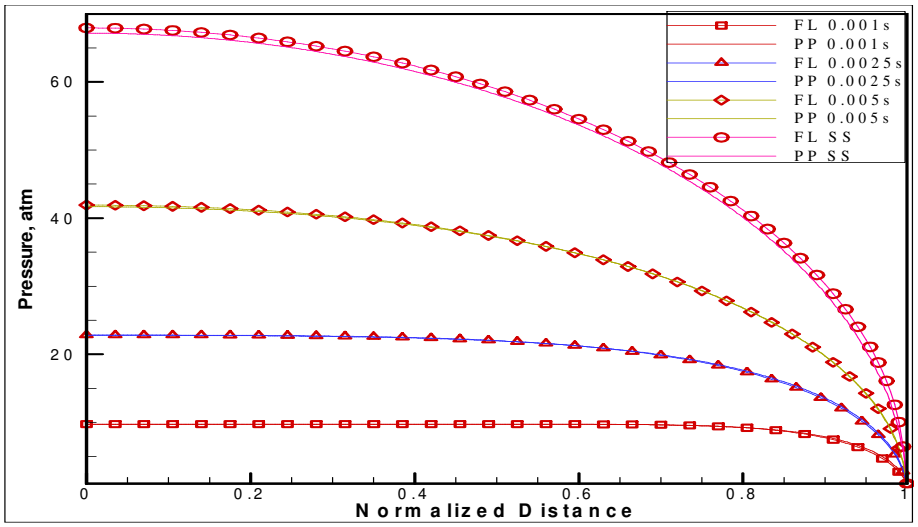

Figure 2.6: Plot of transient *PorePress* and *FLUENT* solutions at various time intervals with an out-gassing rate of 125 lbm/s and a porosity of 0.20

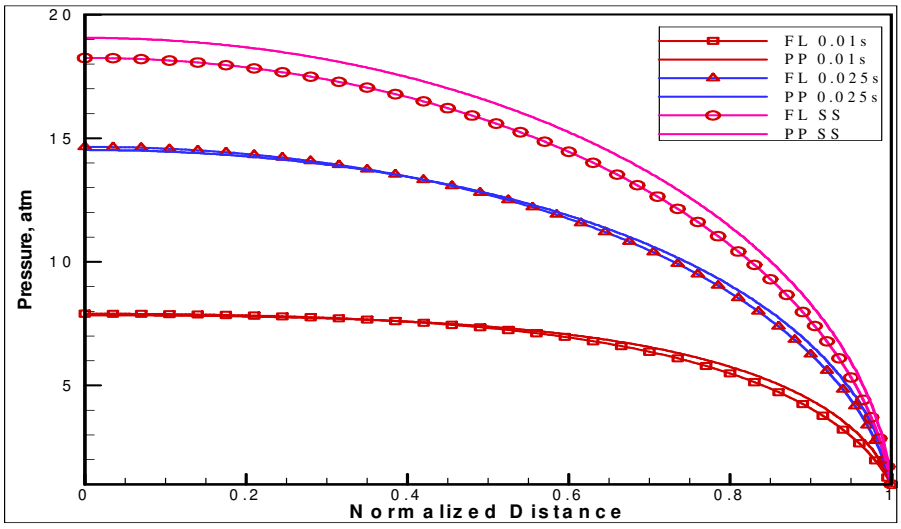

Figure 2.7: Plot of transient *PorePress* and *FLUENT* solutions at various time intervals with an out-gassing rate of 10 lbm/s and a porosity of 0.20

The plot in Figure 2.6 represents the transient solutions for a volumetric out-gassing rate of 125 lbm/ft^3-s with a porosity of 0.20, while the plot in Figure 2.7 has an out-gassing rate of 10 lbm/ft^3-s with a porosity of 0.20. Other plots of transient solutions for other combinations of volumetric out-gassing rates (10 lbm/ft^3-s and 125 lbm/ft^3-s) at a porosity of 0.10 can be seen in Appendix B.

Slight discrepencies between the *PorePress* transient solution and the *FLUENT* transient solution can be attributed to the fact that *FLUENT* does not assume Darcy's form of the momentum equation, and solves the full form of the porous momentum equation. However, the differences are minimal, and serve as validation that for flows typical of applications for *PorePress* the Darcy's form of the momentum equation is suitable.

2.4 1-D Axisymmetric Porous Flow Model Results

A sample geometry representing a path inside a rocket nozzle liner, and an FMAP file representing the time evolution of the path during a burn were provided by *Aerojet*. The path consists of 500 equally spaced 0.002 inch elements spanning a one inch section of carbon-phenolic liner in a nozzle that is 14 inches in diameter. Data from the FMAP file was provided for 100 seconds of burn time (starting from zero and in 0.5 second time increments), with the nozzle initially consisting of all virgin material at 70^{o}F and 1 atm. The surface pressure for the paths provided remains constant for all time steps at 67.9 atm

(998 psi). A schematic of the original path and some of its properties can be seen below in Figure 2.8.

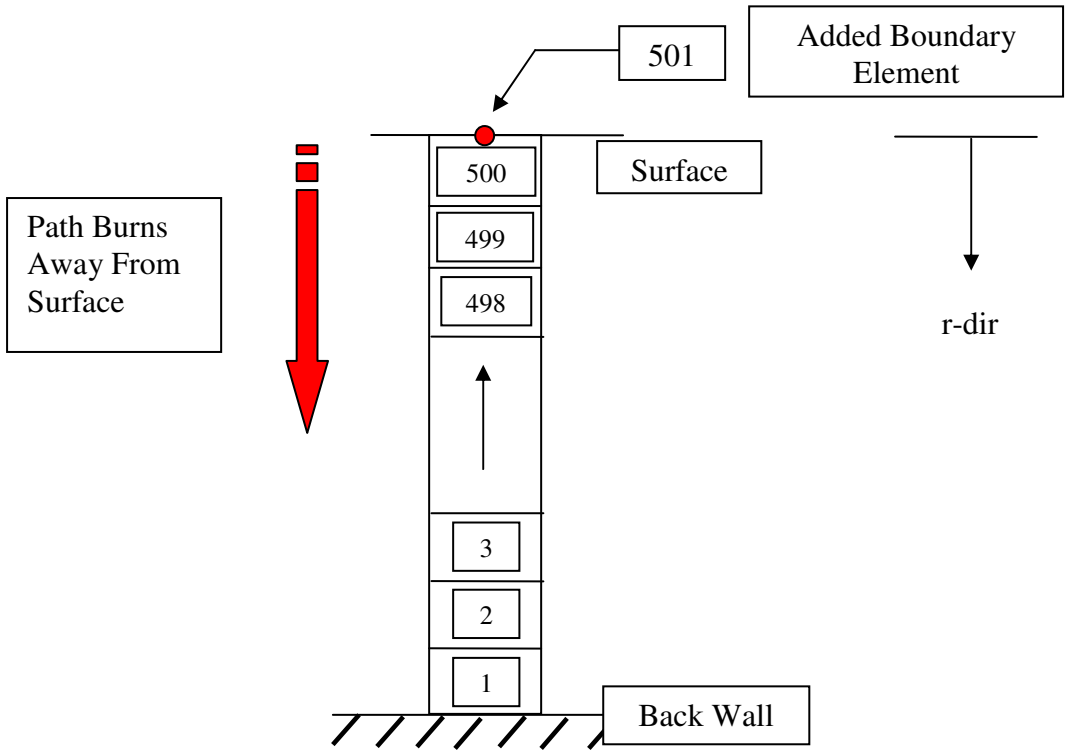

Figure 2.8: Schematic of initial path (time = 0.0 seconds) used in *PorePress*

After 100 seconds of burning the original 500 element mesh reduces to 251 elements, as the other 249 elements charred and got swept away through the burning process. At the first time step (0.5 seconds), the temperature effects only propagate 20 elements in from the original surface element (element 500). At the final time step (100 seconds) the 249th element is now the surface element and the temperature effects have propagated all the way to the back wall (element 1). Figure 2.9 below displays the boundary elements, and path lengths over the course of the burn time covered in the FMAP file, where the initial surface is at a radial distance of 7 inches.

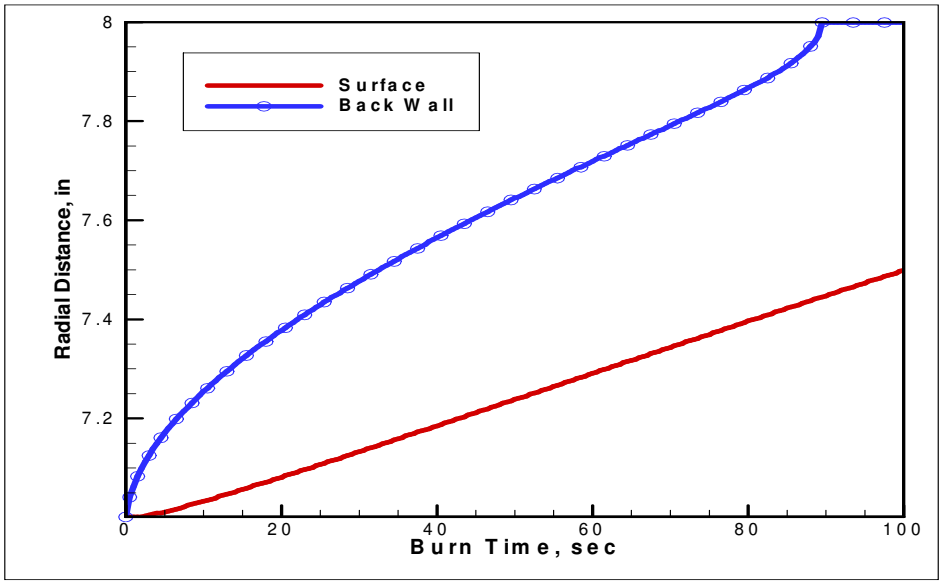

Figure 2.9: Plot of path characteristics throughout charring process used in *PorePress*

One interesting note from Figure 2.9 is that the surface decays almost linearly with time. Meanwhile, after the initial transients die out (approximately 10 seconds into the burn) the temperature effects propagate into the material linearly, but at a slope about twice as steep as the surface decay. This means that the paths continuously get larger, until the temperature effects reach the end of the sacrificial liner (element 1), which occurs about 90 seconds into the burn.

The path used in this study of 500 elements spans only an inch, and from Figure 2.9 above the active portion of the mesh is about half of an inch at most. However, over the course of these relatively small paths, the physical properties change at extremely high rates. In particular, the permeability (K) changes 5 orders of magnitude and the temperature ranges from 70^{o} F to 4500^{o} F over the course of the active paths, which are

less than an inch. Numerical instabilities arise as a result of these sharp gradients, and certain techniques, such as first order upwinding on first derivatives, described in Section 2.2.1 above are employed to add stability by effectively smoothing out the effects of the steep gradients. Also, the porosity over the paths ranges from 8-26% open volume, which means that there is not much space for the pyrolysis gas to flow through.

The *PorePress* code allows the user to select the steady state or transient solver throughout the entire burn, or some combination of the two solvers. Further detail of the actual 1-D *PorePress* code can be seen in the User's Manual Section of the Appendix D. Figures 2.10 and 2.11 below represent steady state and transient pressure solutions from *PorePress* at different burn times from the FMAP file provided. The steady state solution is calculated at a given time step, whereas the transient solution starts at t=0 and integrates the pressure in time.

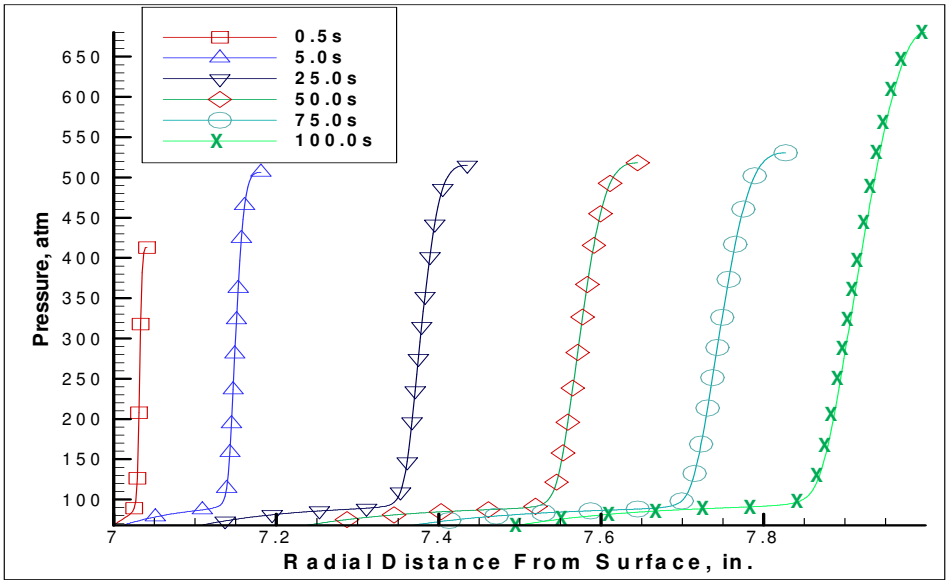

Figure 2.10: Steady State *PorePress* pressure solution at 6 burn times, spanning 100 seconds

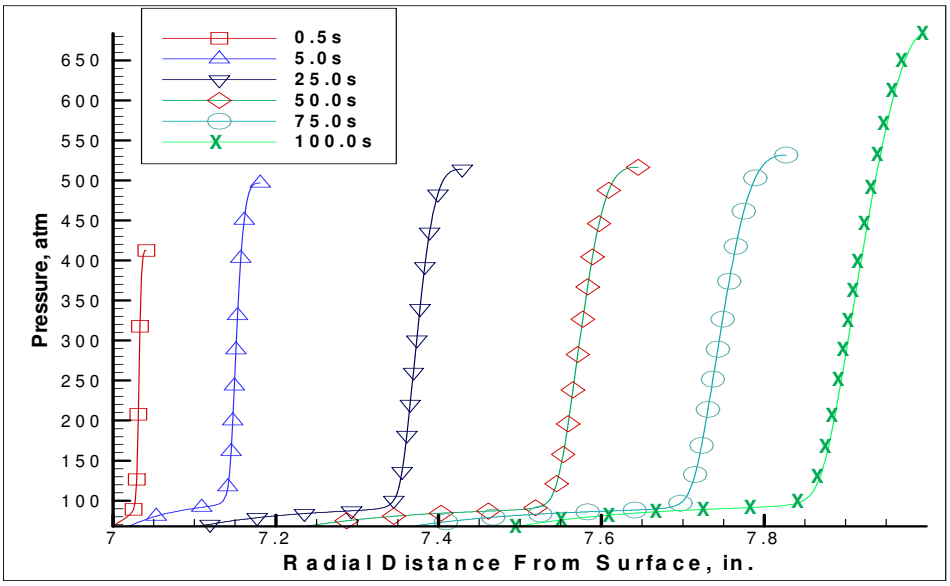

Figure 2.11: Transient *PorePress* pressure solution at 6 burn times, spanning 100 seconds

From Figures 2.10 and 2.11 it can be seen that the pressure distributions in the sacrificial liner display relatively the same curved profile at different burn times. Once the initial transients decay (approximately 10 seconds into the burn), the maximum path pressure remains relatively constant at about 450-520 atmospheres (depending on steady or transient model). However, once the temperature effects reach the back end of the sacrificial liner (89.5 seconds into the burn) and the path can no longer extend, the maximum pressure begins to build. From a design standpoint, this would suggest that as long as the sacrificial liner is thick enough so that the temperature effects do not reach the back wall, the maximum pressure and the pressure distribution curve in the liner will remain relatively constant. The difference between the steady state and transient solutions at certain burn times can be seen in more depth in the plots below in Figure 2.12.

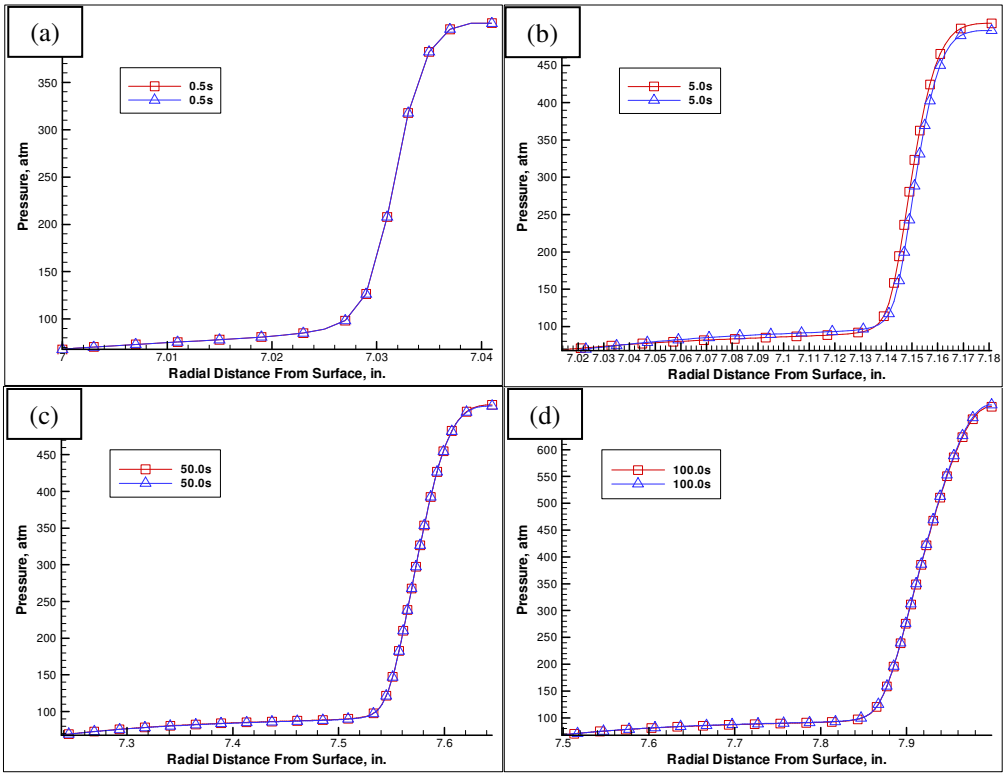

Figure 2.12: Plot of steady state (red line w/ squares) and transient (blue line w/ triangles) pressure solutions at (a) 0.5 s (b) 5.0 s (c) 50.0 s (d) 100.0 s burn times

Typically, the steady state solution over-predicts the transient solution by 1-2% from the time that the initial transients decay to the time that the temperature effects reach the end of the nozzle. Also, the pressure distribution for all time steps reflects the out-gassing distributions over the path. The pressure always peaks in areas of higher out-gassing rates. For the sample case provided the out-gassing rates were highest towards the back wall. A plot of the out-gassing rates for the path at 50.0s can be seen below.

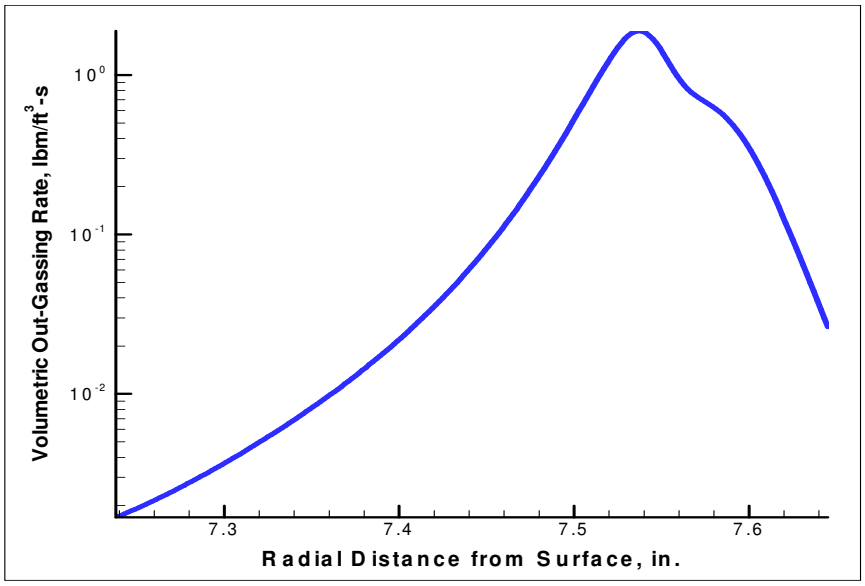

Figure 2.13: Plot of out-gassing rate distribution over 1-D path at a time of 50.0s

Finally, an interesting note from all of the pressure distribution plots is the very high spatial pressure gradients. Over the course of the fraction of an inch paths, the pressure rises as much as 500 atmospheres (~50,600 kPa or 7300 psi). Other physical properties such as temperature, density, out-gassing, permeability, and porosity also display high spatial gradients across the paths. The large spatial gradients, as well as the non-linearity of the original equation (Equation 2.6) contribute greatly to the unstable nature of the overall problem.

CHAPTER 3. 2-D AXISYMMETRIC POROUS FLOW MODEL

The 2-D pore pressure model must use a different solution process from the 1-D case, because the 2-D *PorePress* code does not receive pre-defined gas paths as in the 1-D case. In the 2-D model, the *PorePress* code solves for the pressure distribution over imported meshes that represent a rocket nozzle liner. Just as in the 1-D case the user may select to solve the pressure distribution in steady state, transient or a combination of the two.

The 2-D meshes are mixed, meaning the mesh uses both triangular (3-sided) and quadrilateral (4-sided) elements. The 2-D meshes are also unstructured, meaning they have no set order (i.e. i,j,k ordering), and are instead described by connectivity relationships from cell to cell. Figure 3.1 below shows some of the differences between unmixed-structured meshes and mixed-unstructured meshes.

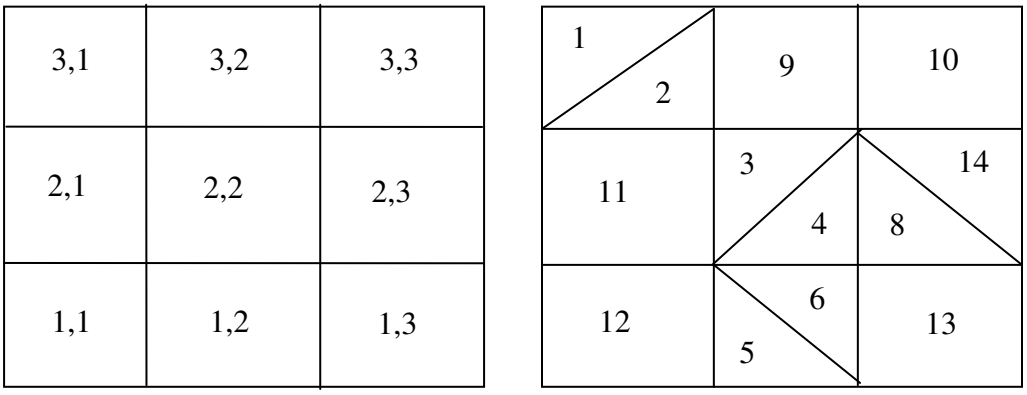

Figure 3.1: Sample unmixed-structured mesh (left) and mixed-unstructured mesh (right)

The left of Figure 3.1 represents a structured mesh that has a set ordered arrangement of elements. For example, to the right of element (1,1) is element (1,2), above element (1,1) is element (2,1), and so on. Meanwhile in unstructured meshes, moving from element to element is done by using connectivity relationships. In the sample mesh on the right of Figure 3.1, it would be established that to the right of element 12 is element 5, above element 12 is element 11, and so on.

As in the 1-D case, the 2-D nozzle liner meshes have a surface exposed to the flow of the heated exhaust gases, and an effective back wall. The back wall may either represent a depth of the liner to which the temperature effects have not reached, or the end of the liner and the beginning of the actual nozzle. Figure 3.2 below represents the physical model and the boundary conditions for a typical section of a nozzle liner.

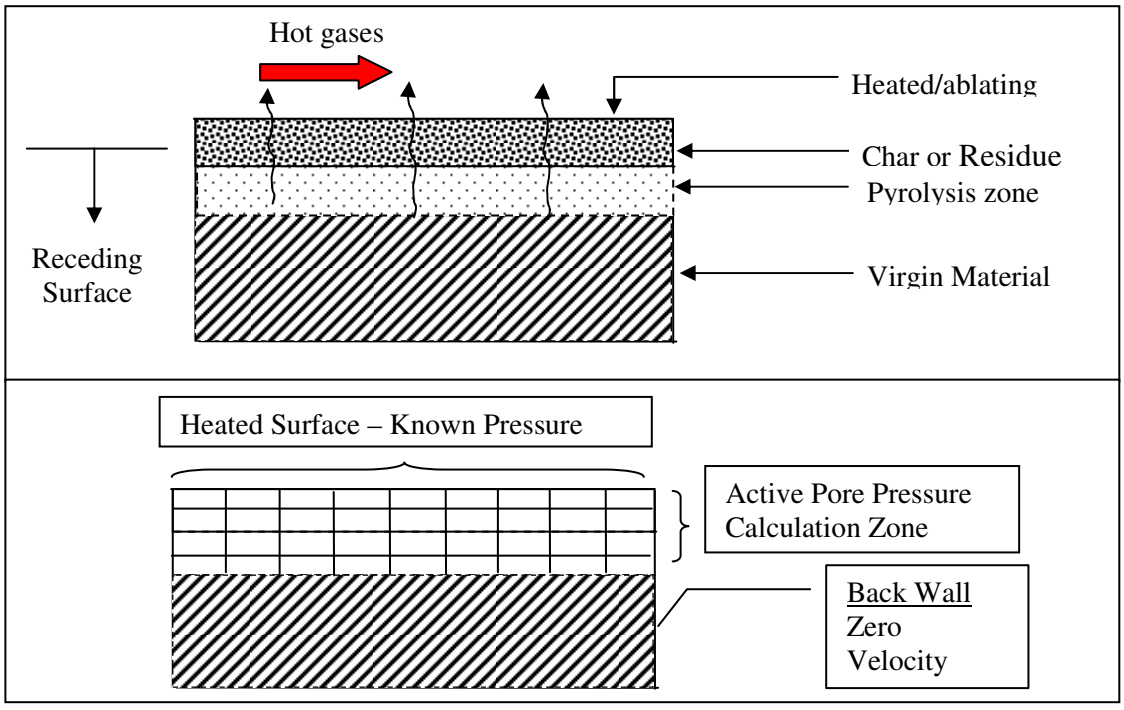

Figure 3.2: A physical section of rocket nozzle liner (top) and the mathematical model for the same section (bottom)

The active pore pressure calculation zone of Figure 3.2 represents the computational cells where the pyrolysis gas flows. As in the 1-D case the computational cells may be subject to out-gassing.

In the 2-D case the geometries are relatively complex and the computational cells rarely line up with the principal axes; therefore, conventional derivative approximation methods will not yield sufficiently accurate solutions. The 2-D form of *PorePress* uses a linear least-squares gradient reconstruction method based on 2nd order Taylor Series expansions to approximate first and second derivative operators.

3.1 2-D Axisymmetric Porous Flow Model Equations

The derivative approximations are used with coupled forms of the Ideal Gas Law, 2-D Continuity, and 2-D Conservation of Momentum equations to yield the final pressure solution in the rocket nozzle liner.

3.1.1 Ideal Gas Law

The pyrolysis gas is assumed to behave as an ideal gas, as described in the equation below:

$$\rho_g = \frac{PM}{RT} \tag{3.1}$$

The variables and assumptions for the Ideal Gas Law are the same as in the 1-D case (Section 2.1.1)

3.1.2 2-D Axisymmetric Conservation of Mass

The form of the 2-D Axisymmetric Continuity Equation used in *PorePress* is as follows:

$$\phi \frac{\partial \rho_g}{\partial t} + \frac{1}{r} \frac{\partial \left(r \rho_g v_r \right)}{\partial r} + \frac{\partial \left(\rho v_z \right)}{\partial z} = \frac{\dot{m}_{O-G}}{V} \tag{3.2}$$

where the *r* and *z* subscripts refer to the *r* and *z*- directions in cylindrical coordinates. The variables and assumptions for the 2-D Conservation of Mass are the same as in the 1-D case (Section 2.1.2)

3.1.3 2-D Axisymmetric Conservation of Momentum

The form of the 2-D Axisymmetric Conservation of Momentum equation used in *PorePress* is as follows:

$$-\frac{\partial P}{\partial r} = \frac{\mu}{K_r} v_r \tag{3.3}$$

$$-\frac{\partial P}{\partial z} = \frac{\mu}{K_z} v_z \tag{3.4}$$

where the *r* and *z* subscripts refer to the *r* and *z*- directions in cylindrical coordinates. The variables and assumptions for the 2-D Conservation of Mass are the same as in the 1-D case (Section 2.1.3).

3.2 2-D Axisymmetric Porous Flow Model Solution Process

Due to the nature of the predetermined paths provided in the 1-D case, the derivatives could be approximated with simple finite differences, such as the following 1st order upwind pressure derivative in the r-dir:

$$\left.\frac{\partial P}{\partial r}\right|_i = \frac{1}{\Delta r} P_i - \frac{1}{\Delta r} P_{i-1} \tag{3.5}$$

where the stencil for the current element *i*, contains only one other element (*i*-1), and is first order accurate. However, in the 2-D case there are no predetermined gas paths. and the mesh does not typically line up with the cylindrical axes. To maintain the same order of accuracy derivative approximations for the 2-D case, *PorePress* uses a linear least-squares gradient reconstruction method with eight point stencils. For this work the linear least-squares reconstruction method is used to approximate first and second derivatives on a mixed, unstructured grid by using Taylor Series approximations. For example, the 2-D linear least-squares gradient reconstruction method will result in the following form of a 1st order pressure derivative in the r-dir:

$$\left.\frac{\partial P}{\partial r}\right|_0 = d_1 P_1 + d_2 P_2 + d_3 P_3 + d_4 P_4 + d_5 P_5 + d_6 P_6 + d_7 P_7 + d_8 P_8 + d_0 P_0 \tag{3.6}$$

where the d_i are coefficients from the linear least-squares gradient reconstruction. A similar approximation is used to obtain the other 1st and 2nd order derivatives.

The derivative approximations are substituted into an expanded form of the following coupled Ideal Gas Law, 2-D Axisymmetric Continuity, and 2-D Axisymmetric Conservation of Momentum Equation:

$$\frac{\phi}{T}\frac{\partial P}{\partial t} - \phi\frac{P}{T^2}\frac{\partial T}{\partial t} - \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{P}{T}\left(\frac{K_r}{\mu}\frac{\partial P}{\partial r}\right)\right) - \frac{\partial}{\partial z}\left(\frac{P}{T}\left(\frac{K_z}{\mu}\frac{\partial P}{\partial z}\right)\right) = \frac{R}{M}\frac{\dot{m}_{OG}}{V} \quad (3.7)$$

where all of the variables are the same as the 1-D case, except they now apply to both the radial (r) and stream-wise (z) directions.

3.2.1 Constructing the System of Equations

The linear least-squares gradient reconstruction process begins by setting up a system of equations for each active element based on 2nd order Taylor series expansions. A second order Taylor Series expansion in cylindrical coordinates for element *0*, surrounded by element *i* is

$$\psi_i - \psi_0 = \Delta r_{0i}\frac{\partial \psi}{\partial r} + \Delta z_{0i}\frac{\partial \psi}{\partial z} + \frac{1}{2}\Delta r_{0i}^2\frac{\partial^2 \psi}{\partial r^2} + \Delta r_{0i}\Delta z_{0i}\frac{\partial^2 \psi}{\partial r \partial z} + \frac{1}{2}\Delta z_{0i}^2\frac{\partial^2 \psi}{\partial z^2} + \cdots \quad (3.8)$$

where ψ is any variable (pressure, temperature, velocity, etc.), Δr_{0i} is the distance in the r-direction from the centroid of element *0* to element *i*, (r_i - r_o), Δz_{0i} is the distance in the z-direction from the centroid of element *0* to element *i*, (z_i - z_o).

The Taylor Series Equation 3.8 must be expanded for at least eight surrounding elements for each active element, in order to achieve approximations for the five (two 1st order and three 2nd order) derivatives in Equation 3.11 below. More on how the surrounding eight elements are selected for each element can be seen below in Section 3.2.4. The result is eight separate equations of the form of Equation 3.8 that can be assembled into a system of linear equations. A general system of linear equations can be expressed in matrix form:

$$\mathbf{Ax} = \mathbf{b} \tag{3.9}$$

where **A** is a coefficient matrix, **x** is a vector of unknowns and **b** is a vector of constants. The solution for this system can be described by:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \tag{3.10}$$

Equation 3.10 can be expressed as:

$$\begin{bmatrix} \Delta z_{01} & \Delta r_{01} & 0.5\Delta z_{01}^2 & \Delta z_{01}\Delta r_{01} & 0.5\Delta r_{01}^2 \\ \Delta z_{02} & \Delta r_{02} & 0.5\Delta z_{02}^2 & \Delta z_{02}\Delta r_{02} & 0.5\Delta r_{02}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Delta z_{0n} & \Delta r_{0n} & 0.5\Delta z_{0n}^2 & \Delta z_{0n}\Delta r_{0n} & 0.5\Delta r_{0n}^2 \end{bmatrix} \begin{bmatrix} \frac{\partial \psi_0}{\partial z} \\ \frac{\partial \psi_0}{\partial r} \\ \frac{\partial^2 \psi_0}{\partial z^2} \\ \frac{\partial^2 \psi_0}{\partial z \partial r} \\ \frac{\partial^2 \psi_0}{\partial r^2} \end{bmatrix} = \begin{bmatrix} \psi_1 - \psi_0 \\ \psi_2 - \psi_0 \\ \vdots \\ \psi_n - \psi_0 \end{bmatrix} \tag{3.11}$$

where $n = 8$, the coefficient matrix **A** consists of centroid to centroid differences between the central element and a specified neighboring element, the unknown vector **x** consists of the derivative approximations, and the **b** vector consists of scalar differences from element to element.

3.2.2 Solving for an Over-Determined System

When a system of linear equations contains more linearly independent equations than unknowns, the result is an over-determined system. Specifically, the system from Equation 3.11, with n=8, would be considered over-determined, as the coefficient matrix, **A**, and scalar difference vector, **b**, have more rows than number of unknowns. In this case there exists more than one solution vector, **x**, that will satisfy the system of equations. The linear least squares technique is designed to make the magnitude of the residual (**Ax-b**) as small as possible. The resulting linear least squares equation is:

$$\mathbf{A}^T \mathbf{A} \bar{\mathbf{x}} = \mathbf{A}^T \mathbf{b} \tag{3.12}$$

and the resulting linear least-squares solution is:

$$\bar{\mathbf{x}} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b} \tag{3.13}$$

where $\bar{\mathbf{x}}$ is the linear least-squares approximation solution that corresponds to the smallest magnitude of the residual (**Ax-b**).

3.2.3 Condition Number and Inverse Weighting

Condition numbers (*CN*) for the coefficient matrices ($\mathbf{A}^T\mathbf{A}$) of each element are calculated in *PorePress* by the following equation:

$$CN = ABS\left(\frac{\lambda_{\max}}{\lambda_{\min}}\right) \tag{3.14}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are the maximum and minimum eigenvalues of the $\mathbf{A}^T\mathbf{A}$ matrix. Elements with relatively high condition numbers (representing near-singular $\mathbf{A}^T\mathbf{A}$ matrices) will lead to instabilities later on in the solution process. Specifically, singular or near-singular matrices present future problems when $\mathbf{R}^{-1}$ (from the QR factorization) is calculated (by attempting to take the inverse of a singular or near-singular matrix).

In general, the mixed, unstructured meshes used in applications for *PorePress* yield coefficient matrices with relatively high condition numbers. One technique used in

PorePress to lower condition numbers and add stability is an inverse weighting scheme [Haselbacher and Vasilyev, 2003]. Each row in the coefficient matrix (which corresponds to a neighboring element in its stencil) is divided by the following weight:

$$\sqrt{\Delta r_{0i}^2 + \Delta z_{0i}^2} \tag{3.15}$$

where Δr_{0i} and Δz_{0i} are the centroid to centroid differences for the central element and the i^{th} element of its 8 point stencil. Therefore, the new coefficient matrix, $\mathbf{A}_{\text{wgt}}$, after the inverse weighting is applied would look like the following:

$$A_{wgt} = \begin{bmatrix} \frac{1}{\sqrt{\Delta r_{01}^2 + \Delta z_{01}^2}} \begin{pmatrix} \Delta z_{01} & \Delta r_{01} & 0.5\Delta z_{01}^2 & \Delta z_{01}\Delta r_{01} & 0.5\Delta r_{01}^2 \end{pmatrix} \\ \frac{1}{\sqrt{\Delta r_{02}^2 + \Delta z_{02}^2}} \begin{pmatrix} \Delta z_{02} & \Delta r_{02} & 0.5\Delta z_{02}^2 & \Delta z_{02}\Delta r_{02} & 0.5\Delta r_{02}^2 \end{pmatrix} \\ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ \frac{1}{\sqrt{\Delta r_{08}^2 + \Delta z_{08}^2}} \begin{pmatrix} \Delta z_{08} & \Delta r_{08} & 0.5\Delta z_{08}^2 & \Delta z_{08}\Delta r_{08} & 0.5\Delta r_{08}^2 \end{pmatrix} \end{bmatrix} \tag{3.16}$$

In the future, $\mathbf{A}_{\mathbf{wgt}}$ will simply be referred to as **A**, and it will be assumed that **b** is also divided by the corresponding weights. Essentially, each row in the system represents a linear equation pertaining to an element in the stencil; therefore the coefficient matrix (as well as the right hand side vector, **b**, is divided by the weight corresponding to the distances to stencil elements. The inverse weighting system lowers element condition

numbers by minimizing the contributions to the derivative functions of the elements farthest away from the central element, and accentuates the contributions from nearby elements.

3.2.4 Condition Number and Stencil Quality

The condition number of an element is also a good approximation of the quality of an element's 8 point stencil. For example, bunching of elements in a stencil in one direction from the central element or poor computational cell aspect ratios will yield higher condition numbers than elements with favorable (evenly distributed) stencils. Figure 3.3 below represents both some favorable and unfavorable stencils.

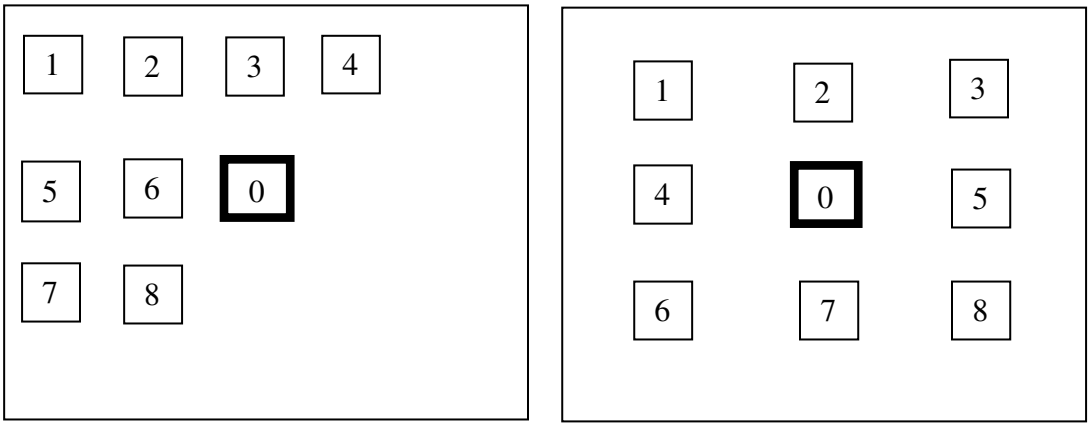

Figure 3.3: Examples of an unfavorable (left) and favorable (right) stencils for element 0

The stencil on the left hand side of Figure 3.3 will yield a relatively high condition number due to the poor spread of the stencil. Conversely, the right hand side of Figure 3.3 represents an ideal spread of elements around the central element. Therefore, in order

to get good derivative approximations for each element, it is imperative to get the best possible spread of neighboring elements for the stencil of each active element.

PorePress uses a priority ranking system when creating 8 point stencils over the mixed, unstructured mesh to get the most favorable 8 point stencils for each element. The stencil collection process begins by gathering elements in the neighborhood of the current element through connectivity relationships. Starting at the central element, the routine gathers all elements within four connectivity sweeps of the current element. For instance, the first sweep gathers all elements that are immediately connected to the central element, the second sweep gathers all elements connected to the elements from the first sweep, and so on until the fourth sweep. All the while, each neighboring element gathered from the sweeps is ranked in a priority system. An example of the sweeps and ranking system for sample element 41 in an 81 element mesh can be seen below in Figure 3.4.

1	2	3	4	**5 (6)**	6	7	8	9
10	11	12	**13 (5)**	**14 (4)**	**15 (5)**	16	17	18
19	20	**21 (5)**	**22 (3)**	**23 (2)**	**24 (3)**	**25 (5)**	26	27
28	**29 (5)**	**30 (3)**	**31 (1)**	**32 (1)**	**33 (1)**	**34 (3)**	**35 (5)**	36
37 (6)	**38 (4)**	**39 (2)**	**40 (1)**	**41 (0)**	**42 (1)**	**43 (2)**	**44 (4)**	**45 (6)**
46	**47 (5)**	**48 (3)**	**49 (1)**	**50 (1)**	**51 (1)**	**52 (3)**	**53 (5)**	54
55	56	**57 (5)**	**58 (3)**	**59 (2)**	**60 (3)**	**61 (5)**	62	63
64	65	66	**67 (5)**	**68 (4)**	**69 (5)**	70	71	72
73	74	75	76	**77 (6)**	78	79	80	81

Figure 3.4: Sample 81 element mesh representing stencil sweep process for sample element 41, and the corresponding priority ranks of each neighboring element in parenthesis

The stencil ranking system prioritizes elements immediately connected to the central element (32,40,42, and 50 in sample mesh above) in the first sweep by giving them a rank of 1. On the second sweep, elements that are neighboring more than one element from the previous sweep (elements 39,33,49, and 51 from sample mesh above) are also ranked with a 1. All other elements from the second sweep (elements 23,39,43, and 59) are ranked with a 2. A similar process is carried out for the rest of the neighboring element sweeps. Within the priority ranking groups (1through 6), elements are also sorted by radius from the central element. The end result is a ranked list of all the neighbors encountered in the 4 sweeps, sorted by priority number first and radial distances within the priority numbers. The first 8 elements in the ranked list are used for the 8 point stencil (elements 31,32,33,40,42,49,50, and 51 from the mesh above). The priority system used in selecting the 8 point stencils for each element is specifically designed to give a spread of elements in a stencil which will provide the lowest condition numbers and most stability in the solution process.

Another factor that will lead to high condition numbers is the aspect ratios of the computational cells that represent each element. Figures 3.5 and 3.6 below visually illustrate how poor aspect ratios for quadrilateral cells can lead to an unfavorable spread of elements.

1	2	3
4	0	5
6	7	8

Figure 3.5: An example of poor aspect ratios on a stencil

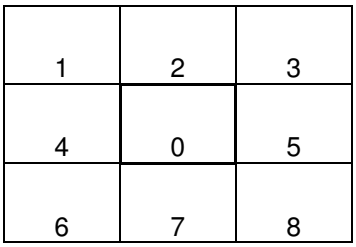

Figure 3.6: An example of ideal aspect ratios when making up a stencil

In Figure 3.5 above, the high aspect ratios creates a bunching effect of elements in the vertical direction, while the elements are spread far out in the horizontal direction. This will lead to disproportionate contributions for the horizontal and vertical directions on the coefficient matrix, thus a poor condition number. Ideally, the quadrilateral cells will remain as close to a square shape (triangular cells as close to an equilateral shape) as the mesh will allow as seen in Figure 3.6 above. The aspect ratios of the individual cells are effects of the original mesh, which is beyond the control of *PorePress*, but should be monitored closely in the mesh generation process.

The *PorePress* code was designed to be used for several different unstructured meshes; therefore, there is not a standard condition number for an element that will cause instabilities. The *PorePress* code will break down if a single element has an extremely high condition number, or if a group of elements have relatively high condition numbers. The effect of poor conditioning propagates throughout areas of the mesh. In certain areas such as edges and corners poor condition numbers are unavoidable, making it important that interior elements near the edges and corners have favorable aspect ratios and stencil shapes to absorb the nearby instabilities. There aren't exact quantitative numbers that can

be put on stability, thus for effective use of the *PorePress* code it is important to monitor the condition numbers of the active cells in the mesh.

3.2.5 QR Factorization of the Coefficient Matrix

In the linear least-squares solution process, significant numerical errors commonly arise when calculating $(\mathbf{A}^T\mathbf{A})^{-1}$ from Equation 3.13 [Strang, 1993]. Therefore, methods such as the QR factorization that circumvent directly calculating $(\mathbf{A}^T\mathbf{A})^{-1}$ are preferable. The QR factorization is a linear algebra matrix factorization technique commonly used in linear least squares problems [Strang, 1993].

There are 3 widely used orthogonalization methods used to compute the **Q** and **R** matrices for the factorization of **A**: Givens Rotations, Householder reflections, and Gram-Schmidt process. Givens Rotations are typically used when the coefficient (**A**) matrix contains a relatively low number of non-zero off diagonal elements, which is not the case for a typical coefficient (**A**) matrix derived from the 2^{nd} order Taylor Series approximation. The Householder reflection technique is considered to be slightly more numerically stable than the Gram-Schmidt process, but computationally more expensive. Additionally, slight changes can be made to the Gram-Schmidt algorithm to add numerical stability, this is referred to as the modified Gram-Schmidt process (see Appendix E). For the scope of this work the modified Gram-Schmidt process has proven to be numerically stable, thus is used as the orthogonalization method.

The QR decomposition of a real, square matrix **A** is given by

$$\mathbf{A} = \mathbf{QR} \tag{3.17}$$

where **Q** is an orthogonal matrix (meaning $\mathbf{Q}^T\mathbf{Q}=\mathbf{I}$) and **R** is an upper triangular matrix. The QR decomposition requires that the original matrix **A** is non-singular with linearly independent columns. The end result of the QR decomposition process is a more numerically stable method for solving a linear system of equations.

3.2.6 Solving for the Derivative Approximations

The QR matrix factorization can be substituted for the original **A** matrix and used to solve linear least square system from Equation 3.12 ($\mathbf{A}^T\mathbf{A}\,\bar{\mathbf{x}}=\mathbf{A}^T\mathbf{b}$).

$$\mathbf{QR}\bar{\mathbf{x}} = \mathbf{A}^T\mathbf{b} \tag{3.18}$$

Then take advantage of the orthogonal identity ($\mathbf{Q}^T=\mathbf{Q}^{-1}$), and divide through by $\mathbf{R}^T$.

$$\bar{\mathbf{x}} = \left(\mathbf{R}^{-1}\mathbf{Q}^T\right)\left(\mathbf{A}^T\mathbf{b}\right) \tag{3.19}$$

Solving Equation 3.19 for the vector of approximated unknowns, $\bar{\mathbf{x}}$, yields the final solution:

$$\bar{\mathbf{x}} = \left[\frac{\partial}{\partial z} \quad \frac{\partial}{\partial r} \quad \frac{\partial^2}{\partial z^2} \quad \frac{\partial}{\partial z \partial r} \quad \frac{\partial^2}{\partial r^2}\right]^T = \mathbf{R}^{-1}\mathbf{Q}^T\mathbf{A}^T\mathbf{b} \tag{3.20}$$

where in *PorePress* the resulting $\bar{\mathbf{x}}$ vector is the unknown first and second order derivative linear least-squares approximations, **R** and **Q** are the respective upper triangular and orthogonal matrix factorizations of the original coefficient matrix based on centroid to centroid distances, and **b** is the original scalar difference vector.

The result of Equation 3.20 will be 5 sets of 9 coefficients (the 8 elements in the stencil and the central element) that can be used to approximate any first and second order derivative for any variable on the desired mesh. For example, first order derivatives for a variable, ψ, in the *r*- and *z*- directions for any active element will have the following form:

$$\left.\frac{\partial \psi}{\partial r}\right|_0 = c_1\psi_1 + c_2\psi_2 + c_3\psi_3 + c_4\psi_4 + c_5\psi_5 + c_6\psi_6 + c_7\psi_7 + c_8\psi_8 + c_0\psi_0 \qquad (3.21)$$

$$\left.\frac{\partial \psi}{\partial z}\right|_0 = d_1\psi_1 + d_2\psi_2 + d_3\psi_3 + d_4\psi_4 + d_5\psi_5 + d_6\psi_6 + d_7\psi_7 + d_8\psi_8 + d_0\psi_0 \qquad (3.22)$$

where ψ can be the pressure (*P*), temperature (*T*), or any material property term (K/μ), and the coefficients, c_i and d_i, represent the Taylor Series approximation coefficients. The subscripts *1* through *8* represent each of the 8 neighboring elements contained in the stencil of the current element (represented with the subscript *0*).

3.2.7 2-D Steady State Solution

The steady state expanded form of the original coupled Ideal Gas Law, 2-D Continuity, and 2-D Momentum Equation 3.7 above is:

$$-\frac{PK_r}{\mu}\left(\frac{\partial^2 P}{\partial r^2}+\frac{\mu}{K_r}\frac{\partial P}{\partial r}\frac{\partial}{\partial r}\left(\frac{K_r}{\mu}\right)+\frac{1}{P}\frac{\partial P}{\partial r}\frac{\partial P}{\partial r}-\frac{1}{T}\frac{\partial T}{\partial r}\frac{\partial P}{\partial r}+\frac{1}{r}\frac{\partial P}{\partial r}\right) \tag{3.23}$$

$$-\frac{PK_z}{\mu}\left(\frac{\partial^2 P}{\partial z^2}+\frac{\mu}{K_z}\frac{\partial P}{\partial z}\frac{\partial}{\partial z}\left(\frac{K_z}{\mu}\right)+\frac{1}{P}\frac{\partial P}{\partial z}\frac{\partial P}{\partial z}-\frac{1}{T}\frac{\partial T}{\partial z}\frac{\partial P}{\partial z}\right)=\frac{RT}{M}\frac{\dot{m}_{OG}}{V}$$

Each of the derivative terms in Equation 3.23 can be represented at each active element by functions similar to the sample derivatives of Equations 3.21 and 3.22 above. The derivative functions are dependant on pressure, temperature, and material property values from the surrounding elements in the 8 point stencil and coefficients from the linear least-squares gradient reconstruction.

In the steady state solution process, all the first order derivatives in Equation 3.23 are substituted for by the full approximation function, an example of which can be seen above in Equations 3.21 and 3.22. The two second order pressure derivative terms are split up in the following manner:

$$-\frac{PK_r}{\mu}\frac{\partial^2 P}{\partial r^2}\bigg|_0=-\frac{P_0^k K_r}{\mu}\left[f_1P_1+f_2P_2+\cdots+f_7P_7+f_8P_8\right]-\frac{P_0^k K_r}{\mu}\left[f_0P_0^{k+1}\right] \tag{3.24}$$

$$-\frac{PK_z}{\mu}\frac{\partial^2 P}{\partial z^2}\bigg|_0 = -\frac{P_0^k K_z}{\mu}\left[g_1 P_1 + g_2 P_2 + \cdots + g_7 P_7 + g_8 P_8\right] - \frac{P_0^k K_z}{\mu}\left[g_0 P_0^{k+1}\right] \quad (3.25)$$

where f_i and g_i are the derivative approximation coefficients for the 2nd order r- and z-derivatives, respectively. From the two terms in Equations 3.24 and 3.25, the pressure at the next iteration (P_0^{k+1}) is isolated and eventually solved for. The final form of the 2-D steady state equation is:

$$P_0^{k+1} = \left[\frac{P_0^k K_r}{\mu}\left(\left[f_1 P_1 + \cdots + f_8 P_8\right] + \frac{\mu}{K_r}\frac{\partial P}{\partial r}\frac{\partial}{\partial r}\left(\frac{K_r}{\mu}\right) + \frac{1}{P_0^k}\frac{\partial P}{\partial r}\frac{\partial P}{\partial r} - \frac{1}{T}\frac{\partial T}{\partial r}\frac{\partial P}{\partial r} + \frac{1}{r}\frac{\partial P}{\partial r}\right)\right.$$

$$\left. + \frac{P_0^k K_z}{\mu}\left(\left[g_1 P_1 + \cdots + g_8 P_8\right] + \frac{\mu}{K_z}\frac{\partial P}{\partial z}\frac{\partial}{\partial z}\left(\frac{K_z 1}{\mu}\right) + \frac{1}{P_0^k}\frac{\partial P}{\partial z}\frac{\partial P}{\partial z} - \frac{1}{T}\frac{\partial T}{\partial z}\frac{\partial P}{\partial z}\right) + \frac{RT}{M}\frac{\dot{m}_{OG}}{V}\right]$$

$$\div\left[-\frac{P_0^k K_r}{\mu} f_0 - \frac{P_0^k K_z}{\mu} g_0\right] \quad (3.26)$$

where all of the first order derivatives are substituted by derivative functions (similar to Equations 3.21 and 3.22 above). Equation 3.26 is iterated over each active element in the mesh until the average L1 norm of the residual vector (**Ax-b**) converges to a value selected by the user. More details on the steady state solution process and the corresponding computational steps involved in the process can be seen in Appendix E.

3.2.8 2-D Transient Solution

The transient expanded form of the original coupled Ideal Gas Law, 2-D Continuity, and 2-D Momentum Equation 3.7 above is:

$$\phi\frac{\partial P}{\partial t}-\phi\frac{P}{T}\frac{\partial T}{\partial t}-\frac{PK_r}{\mu}\left(\frac{\partial^2 P}{\partial r^2}+\frac{\mu}{K_r}\frac{\partial P}{\partial r}\frac{\partial}{\partial r}\left(\frac{K_r}{\mu}\right)+\frac{1}{P}\frac{\partial P}{\partial r}\frac{\partial P}{\partial r}-\frac{1}{T}\frac{\partial T}{\partial r}\frac{\partial P}{\partial r}+\frac{1}{r}\frac{\partial P}{\partial r}\right)$$

$$-\frac{PK_z}{\mu}\left(\frac{\partial^2 P}{\partial z^2}+\frac{\mu}{K_z}\frac{\partial P}{\partial z}\frac{\partial}{\partial z}\left(\frac{K_z}{\mu}\right)+\frac{1}{P}\frac{\partial P}{\partial z}\frac{\partial P}{\partial z}-\frac{1}{T}\frac{\partial T}{\partial z}\frac{\partial P}{\partial z}\right)=\frac{RT}{M}\frac{\dot{m}_{OG}}{V} \qquad (3.27)$$

where all the terms in Equation 3.27 are solved in the same way as in the 2-D steady state solution process above with the exception of the two transient terms:

$$\phi\left.\frac{\partial P}{\partial t}\right|_0=\phi\frac{P_0^{k+1}-P_0^n}{\Delta t}=\frac{\phi}{\Delta t}P_0^{k+1}-\frac{\phi}{\Delta t}P_0^n \qquad (3.28)$$

$$\phi\left.\frac{P}{T}\frac{\partial T}{\partial t}\right|_0=\phi\frac{P_0^k}{T_0^{n+1}}\frac{T_0^{n+1}-T_0^n}{\Delta t} \qquad (3.29)$$

where just as in the 1-D case, as the iteration counter *k* approaches infinity, the pressure solution (P_0^{k+1}) approaches the solution at the next time step (P_0^{n+1}). Again, P_0^{k+1} is isolated and solved for, yielding the final form:

$$P_0^{k+1} = \left[\phi P_0^k + \phi \frac{P_0^k}{T_0^{n+1}} \left(T_0^{n+1} - T_0^n \right) \right.$$

$$+ \Delta t \frac{P_0^k K_r}{\mu} \left(\left[f_1 P_1 + \cdots + f_8 P_8 \right] + \frac{\mu}{K_r} \frac{\partial P}{\partial r} \frac{\partial}{\partial r} \left(\frac{K_r}{\mu} \right) + \frac{1}{P_0^k} \frac{\partial P}{\partial r} \frac{\partial P}{\partial r} - \frac{1}{T} \frac{\partial T}{\partial r} \frac{\partial P}{\partial r} + \frac{1}{r} \frac{\partial P}{\partial r} \right)$$

$$\left. + \Delta t \frac{P_0^k K_z}{\mu} \left(\left[g_1 P_1 + \cdots + g_8 P_8 \right] + \frac{\mu}{K_z} \frac{\partial P}{\partial z} \frac{\partial}{\partial z} \left(\frac{K_z}{\mu} \right) + \frac{1}{P_0^k} \frac{\partial P}{\partial z} \frac{\partial P}{\partial z} - \frac{1}{T} \frac{\partial T}{\partial z} \frac{\partial P}{\partial z} \right) + \Delta t \frac{RT}{M} \frac{\dot{m}_{OG}}{V} \right]$$

$$\div \left[\phi - \frac{P_0^k K_r}{\mu} f_0 - \frac{P_0^k K_z}{\mu} g_0 \right] \tag{3.30}$$

where all of the first order derivatives are solved by derivative functions (similar to Equations 3.21 and 3.22 above). Equation 3.30 is iterated over each active element in the mesh until the average L1 norm of the residual vector (**Ax-b**) converges to a value selected by the user. One important note is the transient solution process requires a time history of the pressure and temperature; whether it be initial conditions, or from the converged solution at the previous time step. More details on the transient solution process and the corresponding computational steps involved in the process can be seen in Appendix E.

3.3 2-D Axisymmetric Porous Flow Model Verification

A sample Poisson Equation with a known solution is used to verify the linear least-squares gradient reconstruction method based on 2nd order Taylor Series expansions used in *PorePress* to approximate first and second derivative operators. The sample

Poisson Equation can be seen below in Equation 3.31, where the source term is represented in Equation 3.32, and the solution to the sample Poisson Equation is represented by Equation 3.33 below.

$$\nabla^2 \psi = \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = S(x, y) \tag{3.31}$$

$$S(x, y) = 50{,}000\left\{\exp\left[-50\left\{(1-z)^2 + y^2\right\}\right]\right\}\left\{100\left((1-x)^2 + y^2\right) - 2\right\} \tag{3.32}$$

$$\psi(x, y) = 500\left\{\exp(-50((1-x)^2 + y^2))\right\} + 100(1-y)x \tag{3.33}$$

The two second derivatives in the sample Poisson Equation are approximated by *PorePress* over a sample mixed, unstructured mesh created in *Gridgen*, and compared to the exact solution. The mesh used in the validation (seen below in Figure 3.7) is a 2,620 element square mesh with the sides 40 computational cells long. The sample mixed, unstructured mesh contains a cluster of triangular elements in the center, surrounded by quadrilateral elements.

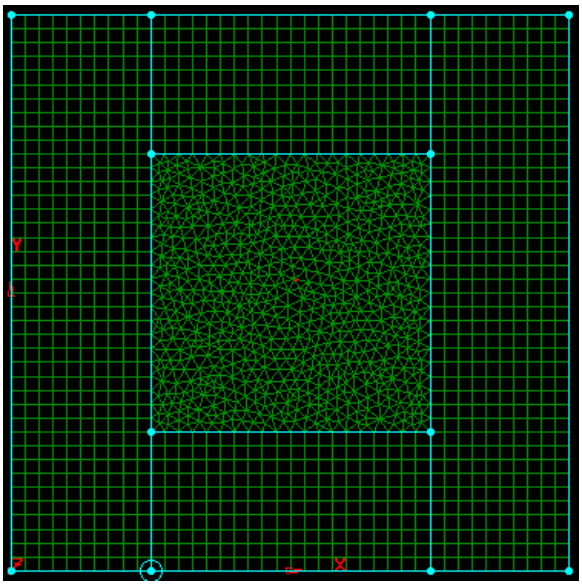

Figure 3.7: Sample mixed, unstructured mesh used in PorePress linear least-squares gradient reconstruction derivative approximation validation

The sample Poisson Equation 3.31 was iterated with the derivative approximations from *PorePress* until the average L_2 norm of the pressure converged to less than 10^{-6}. Upon convergence, the average L_2 norm of the error (ε_{avg} in Equation 3.34) was 0.124%. Contour plots of the exact solution and the *PorePress* solution over the sample domain can be seen below in Figure 3.8.

$$\varepsilon_{avg} = \left[\frac{1}{2620} \sum_{i=1}^{2620} \left(\psi_{Pore\,\mathrm{Pr}\,ess}(i) - \psi_{exact}(i) \right)^2 \right]^{1/2} \qquad (3.34)$$

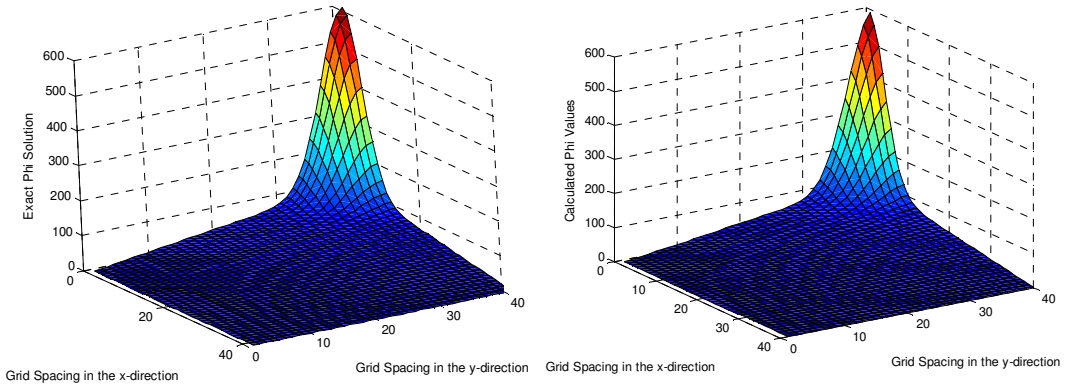

Figure 3.8: Contour plots over the square 40 by 40 mesh for the exact solution (left) and the PorePress solution (right)

3.4 2-D Axisymmetric Porous Flow Model Results

A sample geometry representing a rocket nozzle liner, and an FMAP file representing the time evolution of the liner during a burn were provided by *Aerojet*. The nozzle geometry consists of 24,300 elements spanning a 7 inch long fictitious carbon-phenolic sacrificial liner segment which is approximately 0.75 inches thick. Data from the FMAP file was provided for 0.02 seconds of burn time (starting 5 seconds into the burn and in 0.001s increments). A schematic of the liner section and some of its properties can be seen below in Figure 3.9.

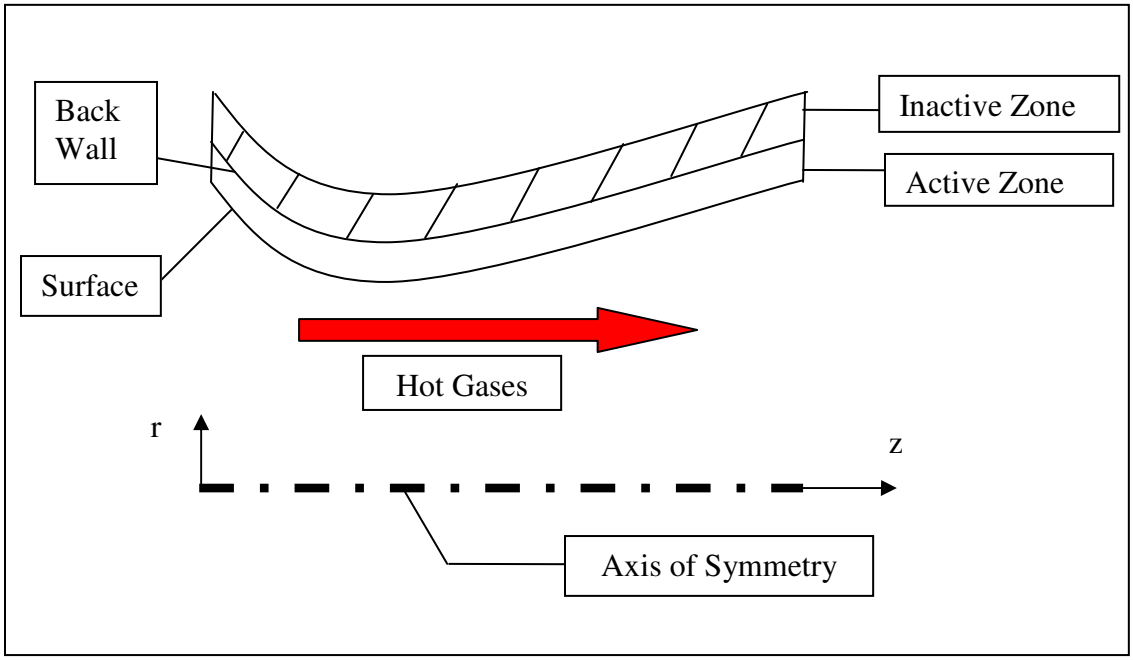

Figure 3.9: Schematic of 2-D axisymmetric geometry representing a sacrificial nozzle liner

As can be seen in Figure 3.9 above, the geometry is modeled in cylindrical coordinates, and is considered to be axisymmetric. Just as in the 1-D case, the dynamic nature of the mesh is accounted for by reading in new FMAP files for each time step.

The boundary conditions used for the 2-D case are: known pressure at the surface of the liner and zero pressure gradient at the back wall and sides of the liner. The surface pressures from the provided FMAP file are constant throughout the given burn time, and vary spatially from 52.2 atm (764 psi) at the upstream end (left side in Figure 3.9 above) of the nozzle liner to 1 atm (14.7) at the downstream end (right side in Figure 3.9 above) of the nozzle.

Over the extent of these 0.75 inch thick nozzle liner segments, the physical properties change at very high rates. In particular, the permeability changes 5 orders of magnitude and the temperature ranges from 70^{o} F to 4500^{o} F over the extent of the active portion of the liner. Also, the porosity over the paths ranges from 8-26% open volume, which means that there is not much space for the pyrolysis gas to flow through. These high spatial gradients can lead to numerical instabilities, thus techniques such as inverse weighting and under-relaxation are applied in the 2-D *PorePress* code.

Solutions for the given FMAP file were run for two different permeability conditions: 1-D permeability ($K_r = K_z = K$) and 2-D permeability ($K_r = 2K_z = 2K$) . The variation of permeability (K) with char density is given in Appendix A. In some cases the permeability can vary greatly in the r- and z- directions; however, the solver used in *PorePress* could not achieve a converged solution if the permeabilities differed by more than a factor of two in the r- and z- directions. It is the authors' belief that in future work a more sophisticated solver or a lower time step should be used to find solutions when the permeability varies by more than a factor of two from one direction to the other.

The *PorePress* code allows the user to select the steady state or transient solver throughout the entire burn, or some combination of the two solvers. The steady state solution is calculated at a given time step, whereas the transient solution starts at a given time (t=0 or a time step with known pressure history) and integrates the pressure in time.

Further detail of the actual 2-D *PorePress* code can be seen in the User's Manual Section of Appendix E.

One important note is that the general distribution of properties (permeability, temperature, pressure, etc.) does not change significantly from the first FMAP file provided (5.0 s) to the final (5.019 s) FMAP file. Therefore, all plots for this sample geometry are shown at the 20th time step (5.019 s), unless otherwise noted. The contour plots in Figure 3.10 below represent steady state and transient *PorePress* pressure solutions for the throat section of the nozzle from the FMAP file provided.

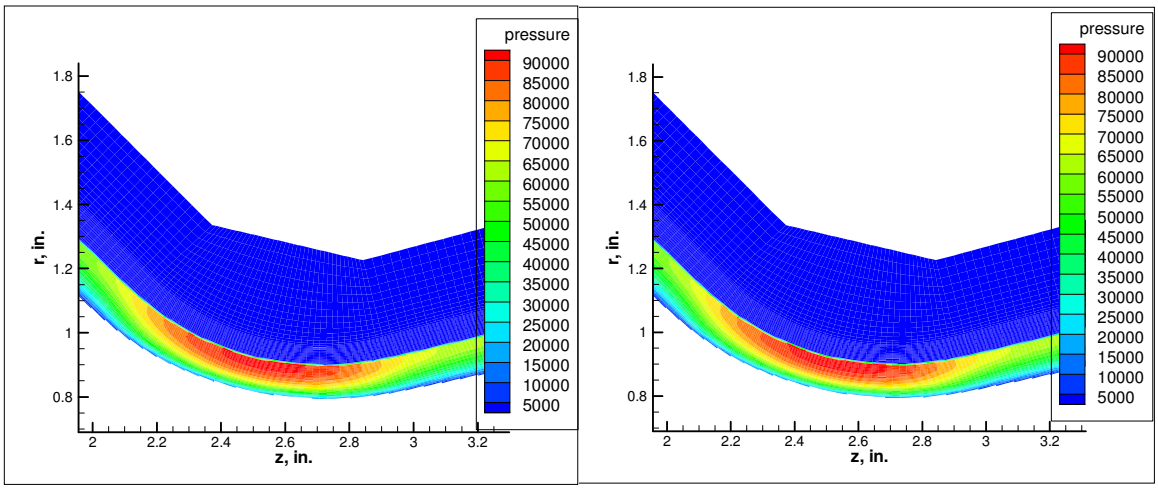

Figure 3.10: Steady state (left) and transient (right) pressure [atm] contour plots at the throat using 1-D permeability

Contour plots for other sections of the nozzle and for the 2-D permeability case can be seen in Appendix B. From Figure 3.10 it can be seen that the pressure ranges from 5,000 atm to 92,000 atm in the nozzle liner. It is conjectured that in reality the pressure in the

liner is relieved by different thermochemical and physical mechanisms, which are outside the scope of the current work. Also, the pressures were calculated using permeabilties that have high uncertainty in their values. The general pressure contour distribution pattern (highest pressure in throat where out-gassing is high) is not affected by permeability values; however, the magnitude of the pressure values are highly sensitive to the permeability. For example, the above geometry with a constant permeability of 10^{-9} ft^2 (the upper bound) results in a distribution with maximum pressure of 700 atm. However, the same geometry with a constant permeability of 10^{-14} ft^2 (the lower bound) results in a distribution with a maximum pressure of 110,000 atm. Therefore, in analyzing the FMAP file provided, it is more important to focus on trends and distributions, rather than the magnitude of the pressure values.

From Figure 3.10, the highest pressures occur in the throat of the nozzle, toward the back wall, despite the fact that the surface pressure is highest at the upstream end of the nozzle. This can be attributed to the fact that the highest volumetric out-gassing rates ($\dot{(\frac{m}{V})}$) occur toward the back wall in the throat of the nozzle. A plot of volumetric out-gassing rates in the throat of the nozzle can be seen below.

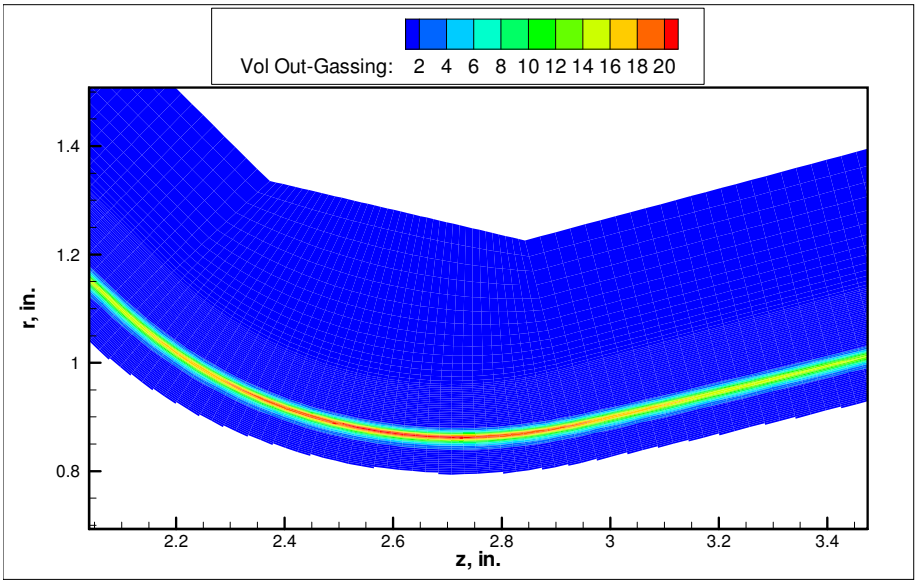

Figure 3.11: Contour plot of volumetric out-gassing rates [$\frac{lbm}{ft^3 s}$]

The pressure distributions are relatively similar for both of the permeability cases; however, the flow of gas is changed significantly. In the case of 1-D permeability ($K_r = K_z$), the pyrolysis gas tends to flow away from the high pressure throat region of the nozzle in all directions. However, as one would expect, the pyrolysis gas flow is directed vertically to the surface more in the 2-D permeability case ($K_r = 2K_z$). Figures 3.12 and 3.13 below represent pressure contour plots of the upstream and throat sections of the nozzle with pyrolysis gas velocity vectors. One interesting note is that the direction of the pyrolysis gas flow seems to be relatively unchanged by the permeability conditions in the high pressure throat region of the nozzle, as seen in Figure 3.13. However, in the low pressure regions the pyrolysis gas flow is affected greatly by the permeability changes, as seen in Figure 3.13 below.

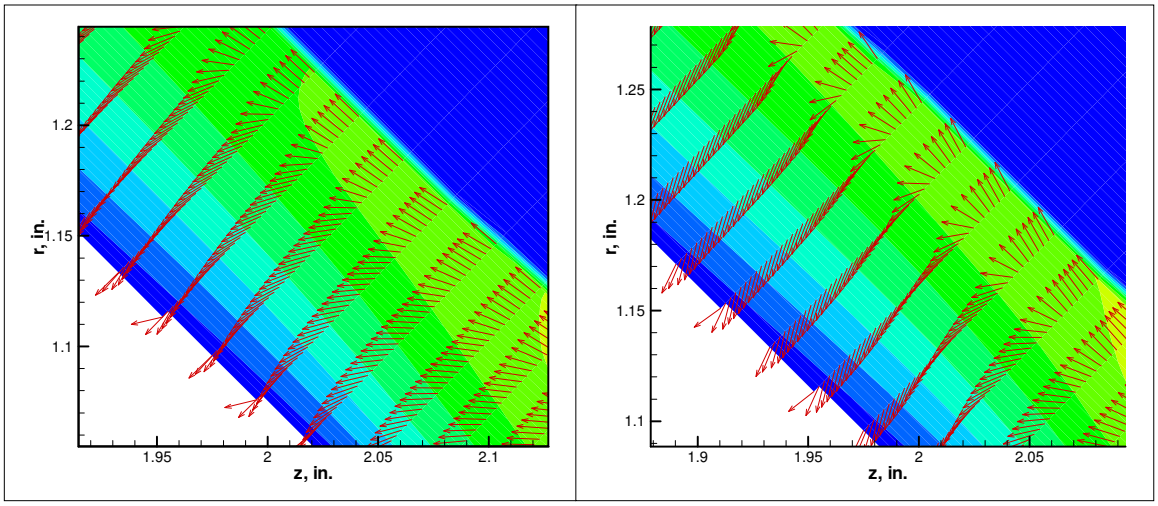

Figure 3.12: Pressure contour plots with pyrolysis gas velocity vectors for 1-D permeability ($K_r = K_z$) [left] and 2-D permeability ($K_r = 2K_z$) [right] in the upstream section of the nozzle. Velocity vectors on surface corners are a result of post-processing interpolation error, where a vector is automatically applied to non-existent surface.

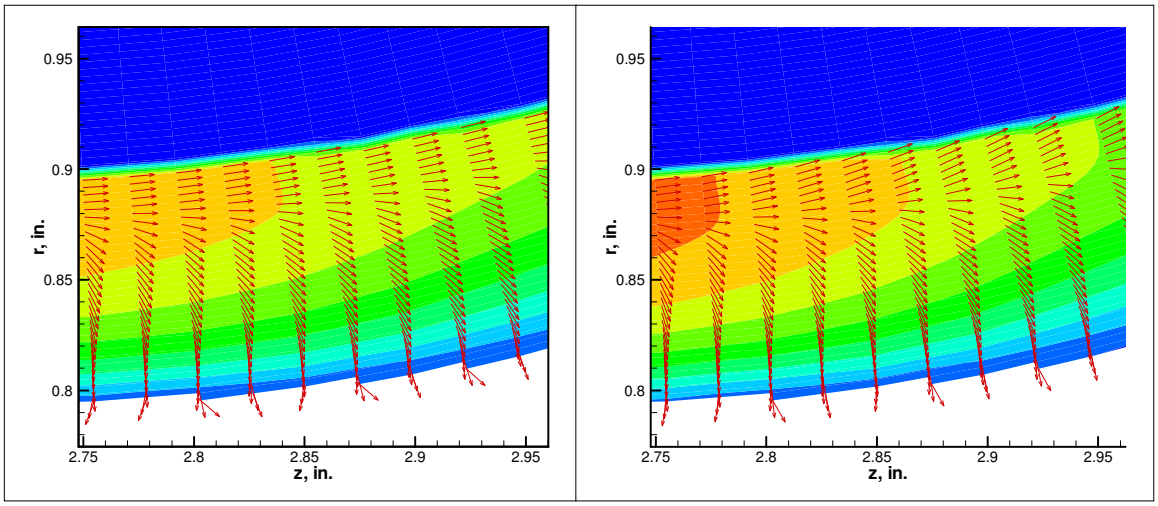

Figure 3.13: Pressure contour plots with pyrolysis gas velocity vectors for 1-D permeability ($K_r = K_z$) [left] and 2-D permeability ($K_r = 2K_z$) [right] in the throat of the nozzle. Velocity vectors on surface corners are a result of post-processing interpolation error, where a vector is automatically applied to non-existent surface.

The plots in Figures 3.12 and 3.13 represent the steady state solution with velocity vectors plotted not to scale, to display the flow direction. A plot of Figures 3.12 and 3.13 with properly scaled velocity vectors scaled with magnitude can be seen in Appendix B.

Finally, a section from the high pressure area of the throat in the 2-D mesh was selected to represent a path, similar to those used in the 1-D *PorePress* code. The section of the 2-D mesh used to simulate a path can be seen below in Figure 3.14.

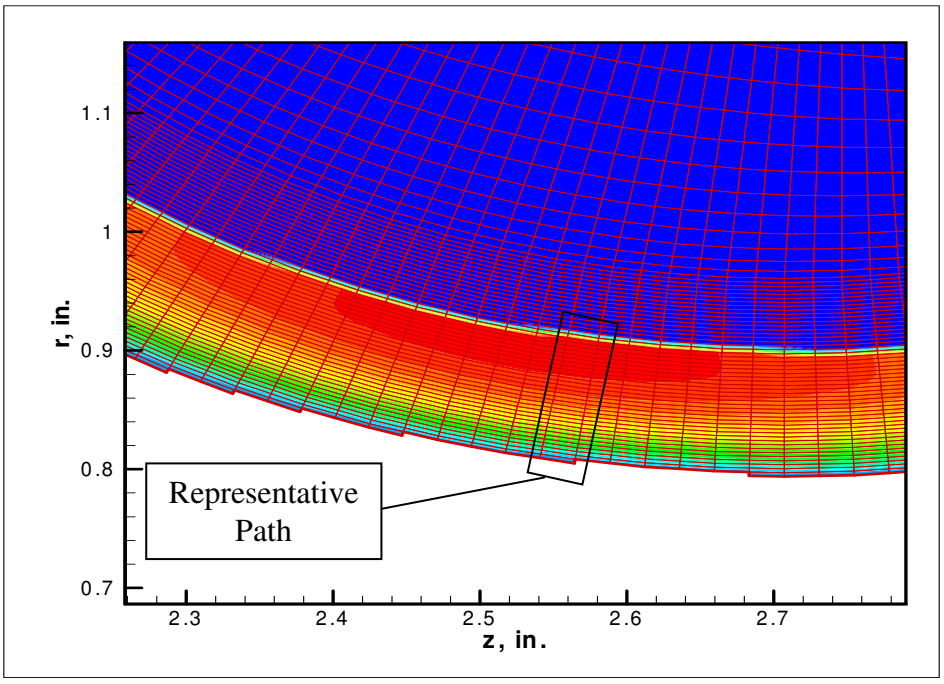

Figure 3.14: Plot of mesh used in *PorePress* over a pressure contour plot in the high pressure throat region. The representative path is outlined (blue section is inactive).

The steady state and transient pressure distributions for the sample path at the first and last time step of the provided FMAP file can be seen below in Figure 3.15. The plots

represent pressure in each of the computational cells in the sample path and their respective distance from the liner surface. The permeability was assumed to be 1-D ($K_r = K_z$) in Figure 3.15 below. A plot similar to Figure 3.15 for the 2-D permeability case (which represents the same characteristics as the 1-D ($K_r = K_z$) case) can be seen in Appendix B.

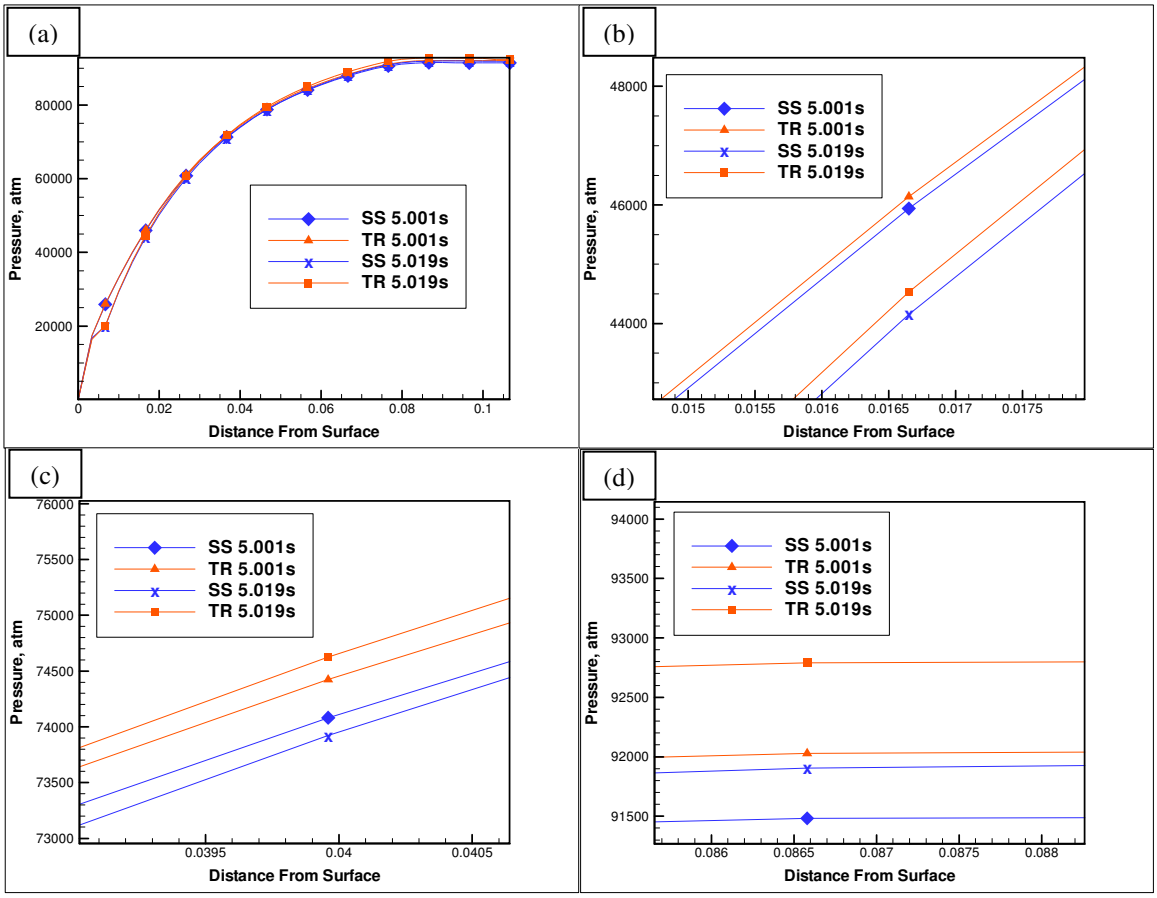

Figure 3.15: Plots of pressure distribution for a path in the nozzle throat [in.], with permeability varying in one direction only for (a) total path (b) section near surface (b) section in middle of path (d) section at end of path

CHAPTER 4. SUMMARY, CONCLUSIONS, AND RECOMENDATIONS

4.1 Summary

4.1.1 1-D *PorePress* Code Summary

The *PorePress* code can be used to find steady state and transient pore pressure distributions on mixed, unstructured 1- and 2-D meshes. In the 1-D case the code solves for the pressure distribution over predetermined gas paths, which simplifies the solution process. Derivatives in a non-linear coupled form of the Ideal Gas Law, Conservation of Mass, and Conservation of Momentum are approximated by 1^{st} and 2^{nd} order backward/forward, and central differences, respectively. The non-linear equation with the appropriate derivative approximations is then solved iteratively. The 1-D solver can be used to solve the steady form of the coupled equation, or can implicitly solve the unsteady form of the coupled equation.

The non-linear nature of the coupled equation and high spatial gradients cause numerical stability problems with the 1-D *PorePress* code; therefore, certain stabilization techniques are applied:

- upwinding
- dynamic differencing
- under-relaxation
- preconditioning

The dynamic difference technique is used on terms that included the squared spatial gradient terms (i.e. $\frac{\partial P}{\partial r}\frac{\partial P}{\partial r}$). Throughout the solution process, the squared spatial gradient terms tend to go unstable before other terms in the non-linear pressure solution equation. The dynamic difference technique applies either first order forward or backwards differences to the spatial pressure gradients, depending on the pressure gradient at an individual cell. The dynamic difference adds stability by increasing the diagonal dominance of the pressure coefficient matrix. Also, first order upwinding is used to discretize all other first derivatives. The upwinding effectively helps to smooth out high spatial gradients that occur throughout the mesh. Under-relaxation is also applied as a means of easing in the effects of high spatial gradients into the solution process. Finally, the solution from a simplified Poisson form of the full coupled Ideal Gas Law, Conservation of Mass, and Conservation of Momentum Equations is used as a preconditioner for the full solution.

4.1.2 2-D *PorePress* Code Summary

The 2-D case does not have predetermined gas paths like the 1-D case, thus a more complex solution procedure is involved. Derivatives for the 2-D case are approximated using a linear least-squares gradient reconstruction technique based on 2nd order Taylor Series expansions. In order to obtain derivative approximations for all 1st and 2nd order derivatives, an eight point stencil is necessary. A system of equations is constructed for each element based on 2nd order Taylor Series expansions for each of the eight stencil elements about the central element. The result is an over-determined system

that is solved by the linear least-squares approximation technique. The solution process then requires inversion of the linear least-squares coefficient matrix to obtain the derivative approximations. However, the linear-least squares coefficient matrix is very sensitive to numerical round off errors during the inversion process. Thus, a QR factorization is performed on the coefficient matrix for each element.

Finally, each element has derivative approximations in the form of functions based on properties of the eight stencil elements. These derivative approximations are used in a 2-D coupled form of the Ideal Gas Law, Conservation of Mass, and Conservation of Momentum Equations. The coupled equation, with the appropriate derivative approximations, is then solved iteratively. The 2-D solver can be used to solve the steady form of the coupled equation, or can implicitly solve the unsteady form of the coupled equation.

As in the 1-D case, stability techniques are applied; however, in the 2-D case stability issues are far more complex than simply smoothing out high spatial gradients. The following stabilization techniques are used in the 2-D *PorePress* code:

- stencil mapping methods
- inverse weighting on coefficient matrix
- QR factorization of coefficient matrix
- under-relaxation
- preconditioning

The 2-D derivative approximations are another source of possible numerical instabilities. The accuracy of the derivative approximation process is highly dependant on the quality of the eight point stencils for each element. A specific mapping method is used to create the best possible spread for each element's eight point stencil. Also, an inverse weighting scheme is applied to accentuate the stencil contributions of elements closest to the current element. The *PorePress* code calculates the condition number for each element as a means of determining the relative quality of a stencil spread. Elements with relatively high condition numbers represent singular or near-singular coefficient matrices, which will cause stability problems when the QR factorization is taken in the 2-D solution process. Under-relaxation and preconditioning are used in the same manner as in the 1-D case.

4.2 Conclusions

The 1- and 2-D *PorePress* code was used with sample geometries representing a nozzle liner at the beginning (sample 1-D mesh) and in the midst of (sample 2-D mesh) a burn. In both the 1- and 2-D cases, for the conditions used in this study, the pressure distribution is driven mainly by the volumetric out-gassing rates and permeability.

In the 1-D case, the pressure is highest towards the back wall, where volumetric out-gassing is highest. In the 2-D case, higher pressures are found near the back wall of the liner in the throat of the nozzle, where the volumetric out-gassing is highest. Also, the magnitude and distribution of the pressure inside the liner was found to be highly

dependant on the permeability. In particular, for the 2-D case, the permeability values for this study spanned five orders of magnitude. The magnitude of the pressure values in the liner varied 3 orders of magnitude, increasing as the permeability decreased. The permeability distribution ($K_r=K_z$ or $K_r=2K_z$) also influenced the direction of the gas flow throughout the nozzle, especially in the relatively low pressure areas.

4.3 Recommendations

Ultimately, the accuracy of the results from *PorePress* used with the sample geometries provided is hard to quantify. The pressure values from sample geometries were very high; however, all of the processes that occur during the decomposition of the liner and the true liner material properties were not well known. There are thermochemical and physical processes that affect the pressure distribution that are considered beyond the scope of this work. Thus, in future work *PorePress* should be used in conjunction with models that accurately depict all of the processes involved in the thermal decomposition of the liner.

Physical properties such as molecular weight, dynamic viscosity, permeability, and porosity contained large uncertainties. In particular, the 1- and 2-D models proved to be very sensitive to the permeability. In future research, the results of *PorePress* should be further analyzed with more accurate characterizations of physical properties, such as permeability. Depending on the variation of the permeability, it may also be necessary to implement a more sophisticated solver than the non-linear, iterative solver used in *PorePress*.

An interesting area of further research would be to take a 2-D mesh and compare the 2-D *PorePress* results to 1-D *PorePress* results when the 2-D mesh is approximated as a series of 1-D paths. The 2-D solution process is much more complex than the 1-D solution process, and many new numerical difficulties arise as a result. Thus, if a 2-D mesh could accurately be modeled as a collection of 1-D paths, a lot of computational time could be saved and several numerical difficulties could be avoided.

REFERENCES

1. Ahn, H.-K., Park, C., Sawada, K., "Dynamics of Pyrolysis Gas in Charring Materials Ablation." AIAA 36th Aerospace Sciences Meeting Exhibit, Reno, Nevada, January 12-15 1998, AIAA A98-16086.

2. Haselbacher, A., Vasilyev, V., "Commutative Discrete Filtering on Unstructured Grids Based on Least-Squares Techniques." *Journal of Computational Physics* **187**, 197-211, 2003.

3. Moyer, C., Rindal, R. "An Analysis of the Coupled Chemically Reacting Boundary Layer and Charring Ablator." NASA Contracted Report: ITEK Corporation, Vidya Division Paolo Alto California, NASA CR-1061, July 1965.

4. Nield, D., Bejan, A. "Convection in Porous Media.", Springer Publishing Co., New York, New York, 1999.

5. Ross, R., Strobel, F. "CMA 90S Input Guide and User's Manual." Acurex Corporation-Aerotherm Division, Huntsville, Alabama, December 1990.

6. Strang, G., "Introduction to Linear Algebra", Wellesley-Cambridge Press Pub. Co., Wellesley, Massachusetts, 1993.

APPENDIX A. MATERIAL PROPERTIES

A.1 Degree of Char

As the virgin material decomposes, the degree of charring is used to describe the amount of decomposition to the original virgin material resulting from the pyrolysis process. The degree of charring formula can be seen below in Equation A.1:

$$DC = 1 - (0.0553 * \rho - 4.0487) \tag{A.1}$$

where the solid char density (ρ) is in *lbm/ft*^{3}, and *DC* is the degree of char material. Equation A.1 shows that the decomposition of the carbon-phenolic in the pyrolysis process is related directly to the density of the solid material that remains. The relationship shown in Equation A.1 was obtained through experimental results for carbon-phenolic properties performed by Aerojet. A plot of degree of charring versus density can be seen below in Figure A.1.

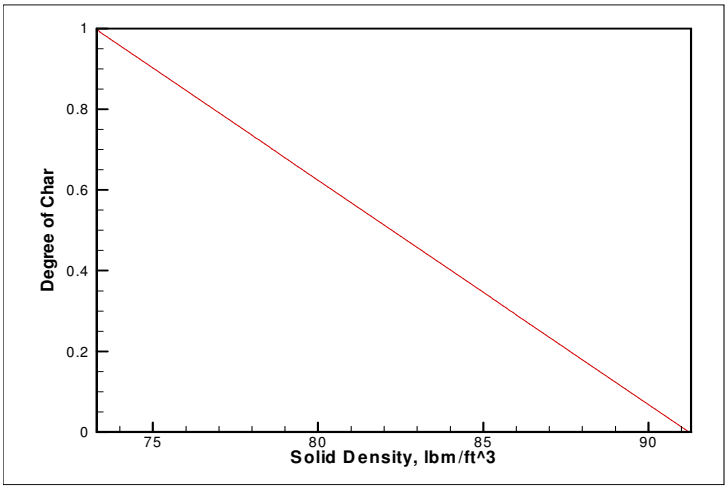

Figure A.1 Plot of Degree of Char vs Solid Density used in *PorePress*

A.2 Porosity

Porosity, ϕ, is defined as the ratio of void volume (V_{open}) to the total volume (V_{tot}) of a medium.

$$\phi = \frac{V_{open}}{V_{tot}} \tag{A.2.1}$$

For this study, the impermeable virgin material is thermally decomposed by pyrolysis into carbonaceous char or residue which forms the porous matrix, and then the pyrolysis gas moves through the void created by the decomposition. It is assumed that all of the void space is connected. The effective result of having isolated voids would create coking in the model, which is beyond the scope of this work. It is also assumed that the porosity is uniform in each computational cell.

For this particular work the porosity is defined as

$$\phi = \frac{\rho}{\rho_{vir}} + 0.05 * \frac{\rho}{\rho_{H2O}} \tag{A.2.2}$$

where ρ is the current density of the carbon-phenolic, ρ_{vir} is the virgin density of the carbon phenolic, and ρ_{H2O} is the density of water (all densities in lbm/ft^3). The second term on the right side of Equation A.2.2 is due to the fact that the carbon-phenolic

samples are known to soak up 5% water weight by volume before displaying porous characteristics. A plot of porosity versus density can be seen below in Figure A.2.

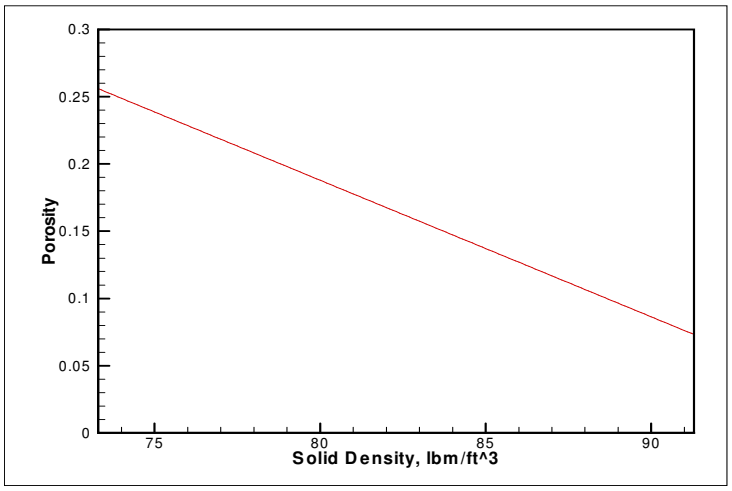

Figure A.2 Plot of Porosity vs Solid Density used in *PorePress*

A.3 Dynamic Viscosity

The dynamic viscosity of the pyrolysis gas is related directly to the temperature of the gas by:

$$\mu = \left(-1.11*10^{-14}\right)*T^2 + \left(3.07*10^{-10}\right)*T + 1.92*10^{-7} \qquad \text{(A.3)}$$

where T is the absolute temperature in degrees Rankine, and the resulting dynamic viscosity, μ, is in *lbf-s/ft*^{2}. Equation A.3 is based on a curve-fit from experimental carbon-phenolic data provided by Aerojet. A plot of the dynamic viscosity curve can be seen below in Figure A.3.

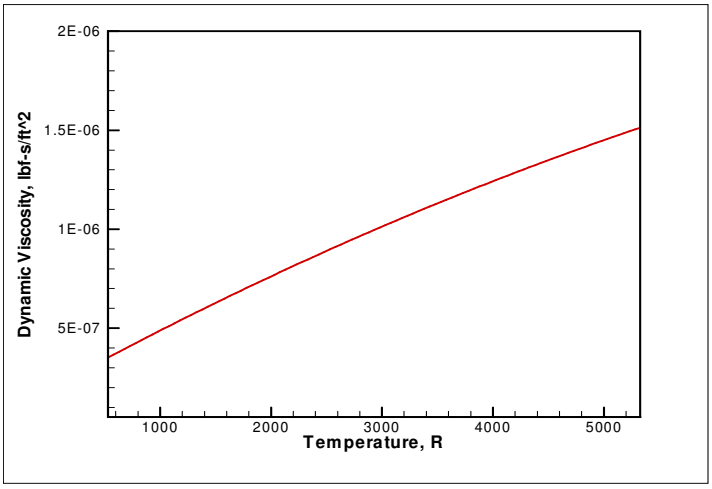

Figure A.3 Plot of Dynamic Viscosity vs Temperature used in *PorePress*

A.4 Permeability

The permeability of the porous carbonaceous char that remains from decomposition, through which the pyrolysis gas percolates is related directly to degree of charred material by:

$$K = \frac{1}{10\text{^}(-5 * \mathrm{DC} + 16.968)} \tag{A.4}$$

where the resulting permeability, K, is in ft^2. Equation A.4 is based on experimental carbon-phenolic data provided by Aerojet. A plot of the permeability based on the degree of charring can be seen below in Figure A.4.

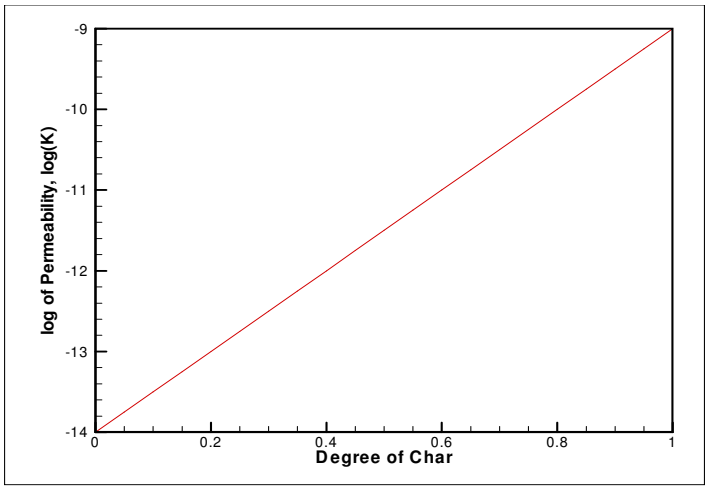

Figure A.4 Plot of the log of Permeability vs Degree of Char used in *PorePress*

A.5 Molecular Weight

The molecular weight of the pyrolysis gas is related to the temperature of the gas by:

$$MW = 8.05 * 10^{-7} * T^2 - 6.60 * 10^{-3} * T + 21.87 \qquad \text{(A.5)}$$

where the temperature T is in degrees Rankine, and the resulting molecular weight (MW) is in *lbm/lbmol*. Equation A.5 is based on experimental carbon-phenolic data provided by Aerojet. A plot of the molecular weight based on the absolute temperature can be seen below in Figure A.5.

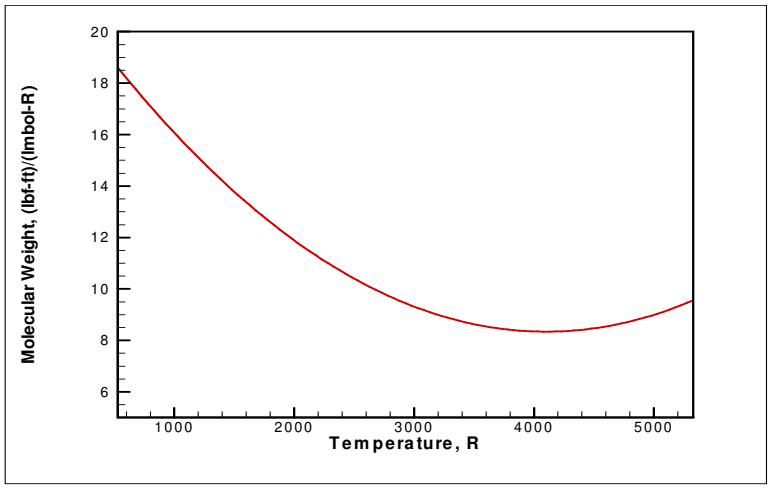

Figure A.5 Plot of Molecular Weight vs Temperature used in *PorePress*

APPENDIX B. SUPPLEMENTAL FIGURES

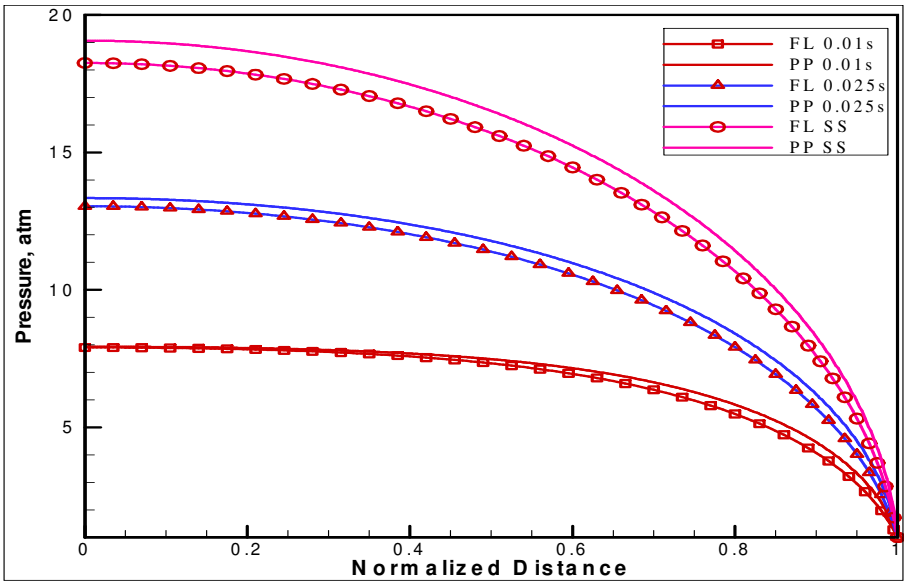

Figure B.1 Plot of transient *PorePress* and *FLUENT* solutions at various time intervals with an out-gassing rate of 10 lbm/s and a porosity of 0.10

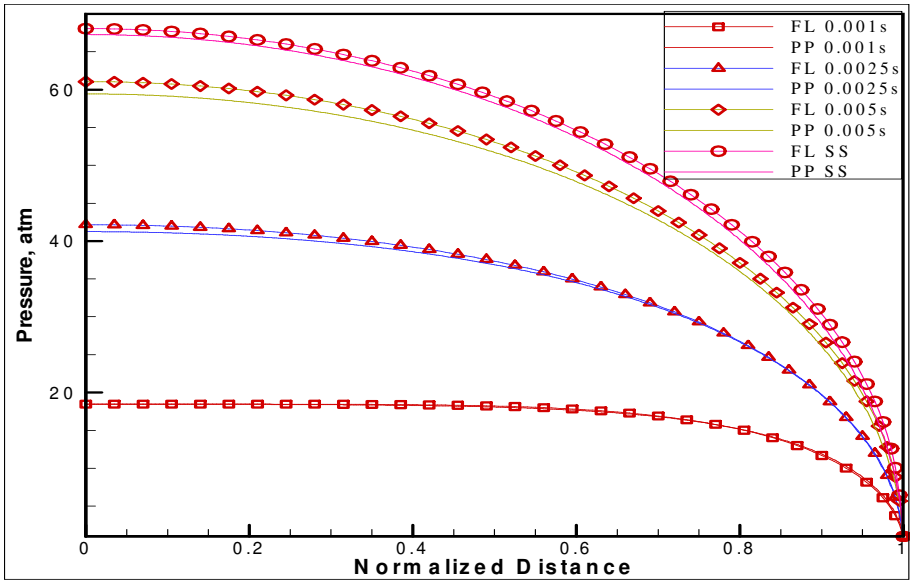

Figure B.2 Plot of transient *PorePress* and *FLUENT* solutions at various time intervals with an out-gassing rate of 125 lbm/s and a porosity of 0.10

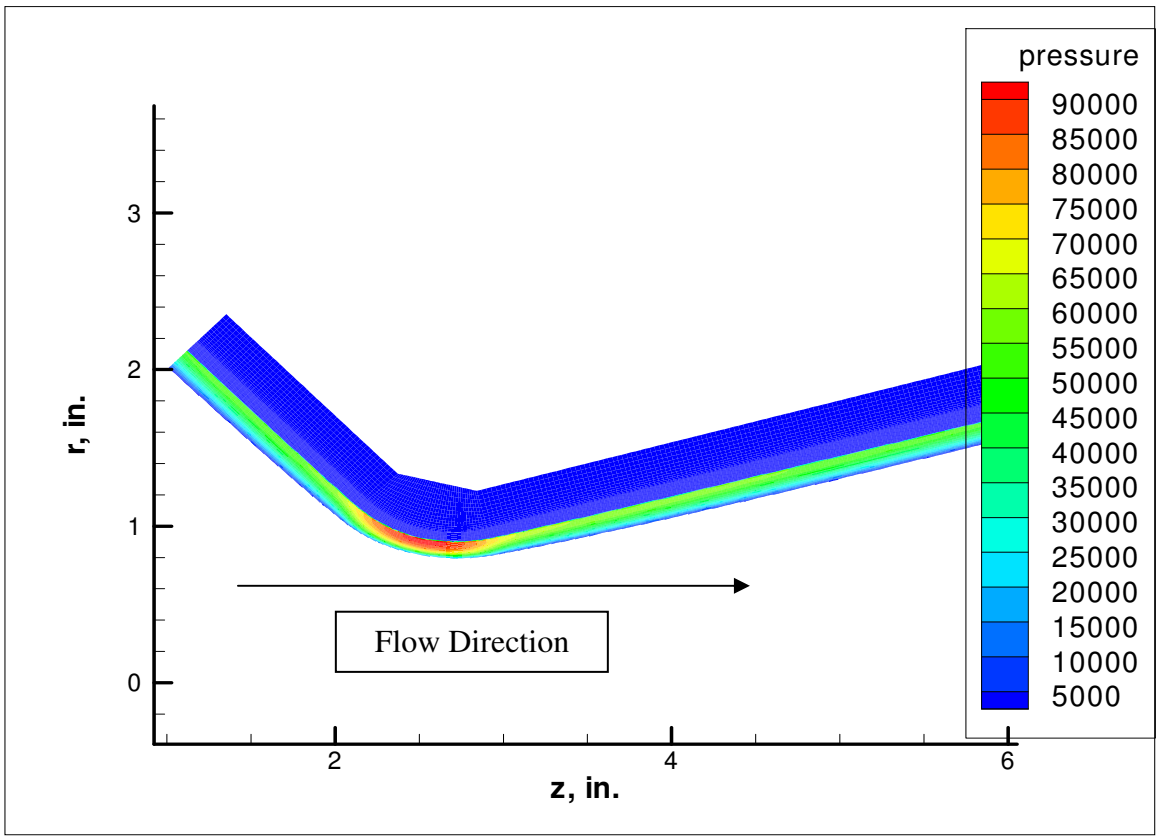

Figure B.3 Larger pressure contour plot for steady state solution with uniform permeabilities (transient plot and variable permeability display same characteristics

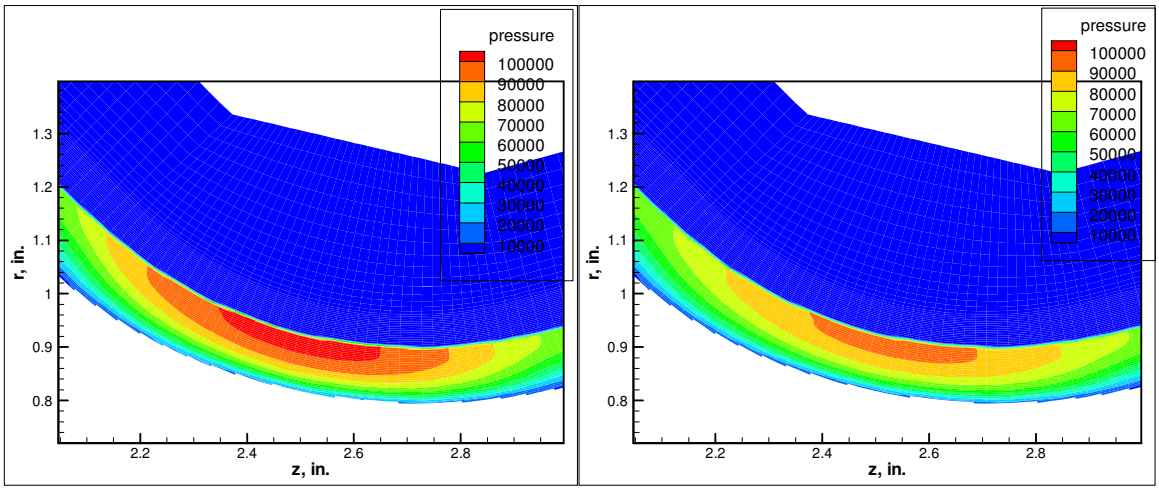

Figure B.4 Steady state (left) and transient (right) pressure [atm] contour plots at the throat for the permeability case $K_r = 2K_z = 2K$

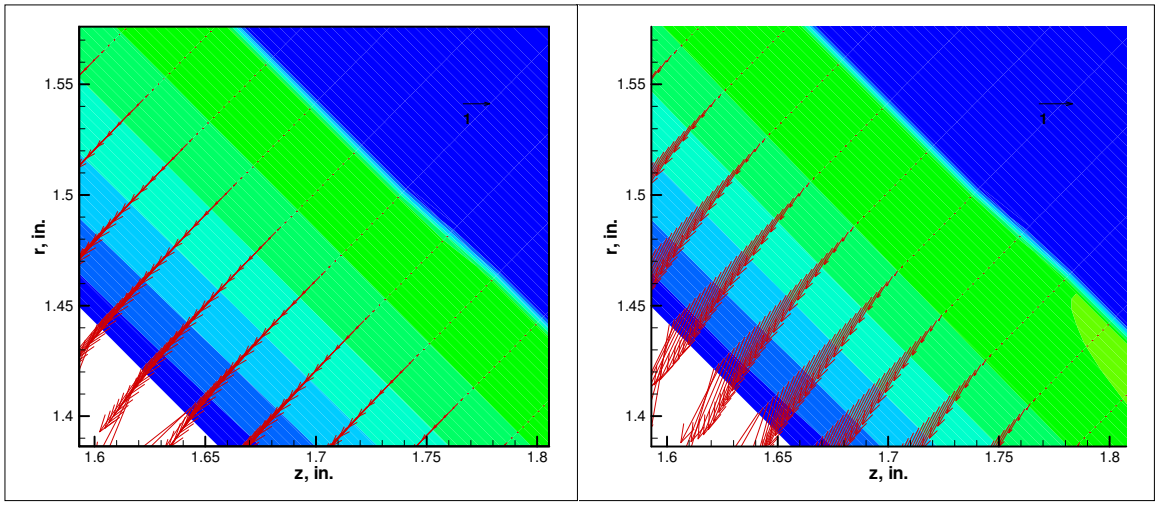

Figure B.5 Pressure contour plots with pyrolysis gas velocity vectors for 1-D permeability (left) and 2-D permeability (right) in the upwind section of the nozzle. Reference vector represents 1 *ft/s*. Velocity vectors on surface corners are a result of post-processing interpolation error, where a vector is automatically applied to non-existent surface.

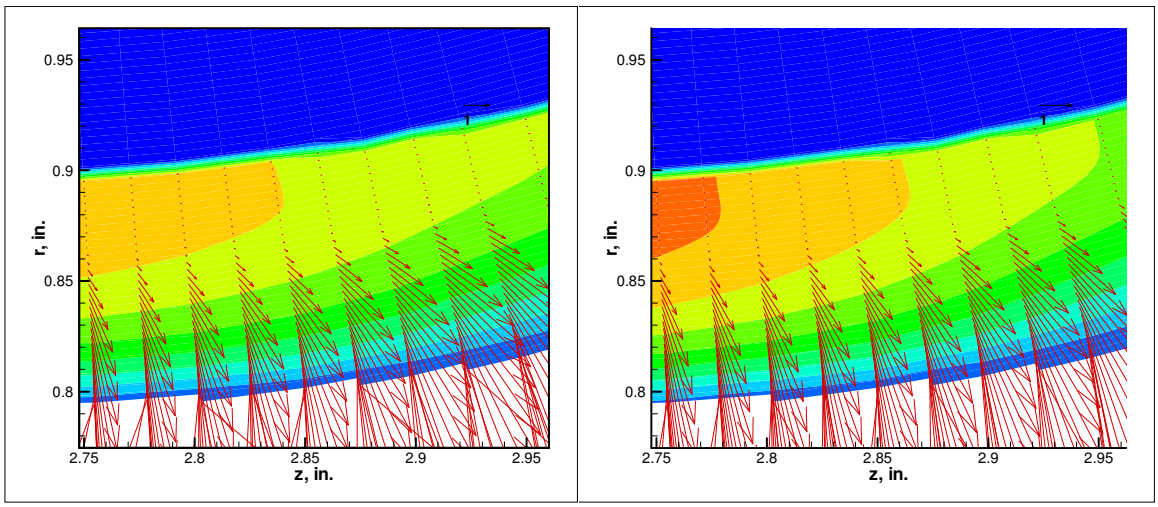

Figure B.6 Pressure contour plots with pyrolysis gas velocity vectors for 1-D permeability (left) and 2-D permeability (right) in the throat section of the nozzle. Reference vector represents 1 *ft/s*. Velocity vectors on surface corners are a result of post-processing interpolation error, where a vector is automatically applied to non-existent surface.

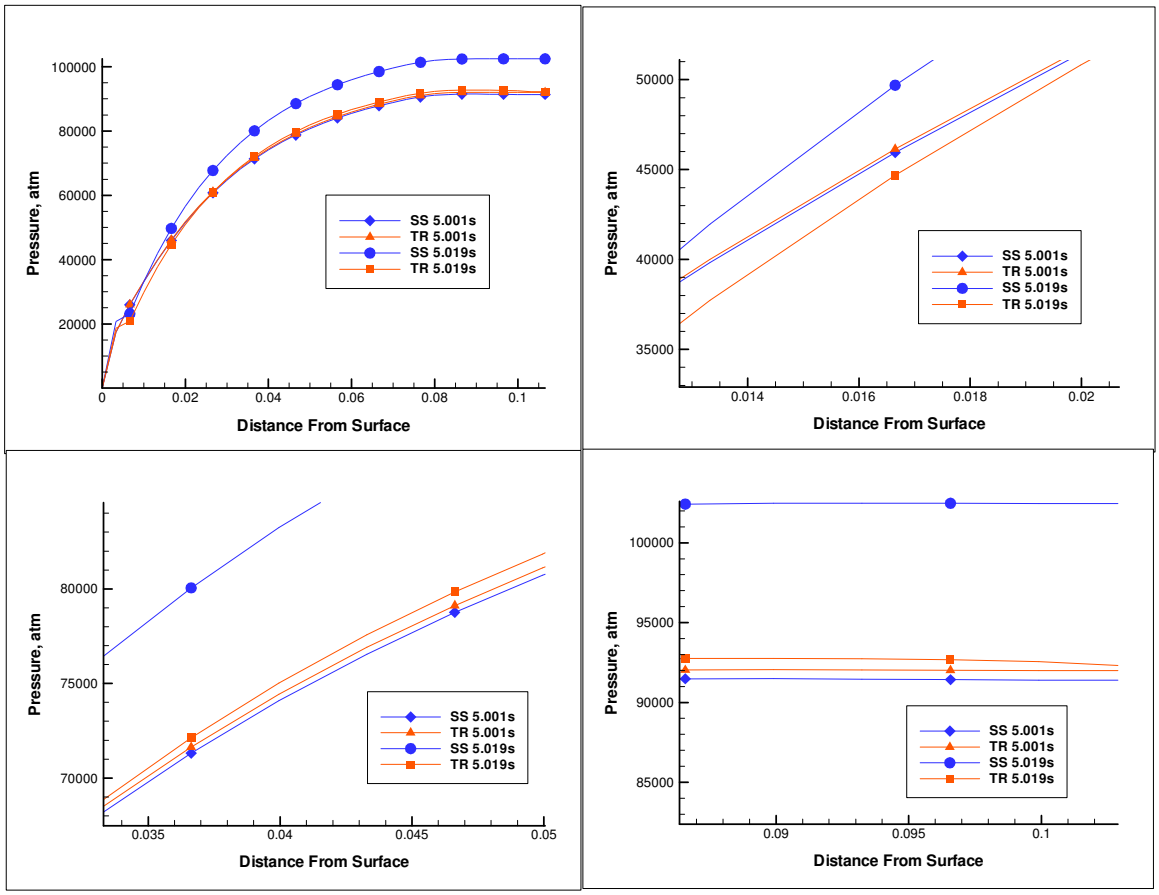

Figure B.7 Plots of pressure distribution for a path in the nozzle throat [in.], with permeability varying in one direction only for (a) total path (b) section near surface (b) section in middle of path (d) section at end of path

APPENDIX C. DIMMENSIONAL ANALYSIS

Original Equation

$$\frac{1}{\phi}\rho\frac{\partial \mathbf{v}}{\partial t}+\frac{1}{\phi^2}\rho(\mathbf{v}\cdot\nabla)\mathbf{v}=-\nabla P-\frac{\mu}{k}\mathbf{v}$$

Non-Dimensional Variables

$$\mathbf{x}^*=\frac{\mathbf{x}}{\sqrt{k}} \qquad P^*=\frac{P}{\mu V_c/\sqrt{k}} \qquad \mathbf{v}^*=\frac{\mathbf{v}}{\sqrt{k}/T_0} \qquad t^*=\frac{t}{T_0}$$

where k is the permeability of the medium, T_o is a characteristic flow time, and V_c is a characteristic velocity of the porous flow of the form $\sqrt{k}/T_0$,.

Non-Dimensional Equation

$$\frac{\partial \mathbf{v}^*}{\partial t^*}+\frac{1}{\phi}(\mathbf{v}^*\cdot\nabla^*)\mathbf{v}^*=-\tfrac{1}{2}\phi\frac{T_0\nu}{k}\nabla^*P^*-\phi\frac{T_0\nu}{k}\mathbf{v}^* \qquad \text{(C)}$$

$O(10^0)$	$O(10^1)$	$O(10^{17})$	$O(10^{16})$

where,

$\nu = \dfrac{\mu}{\rho} = O(10^{-4})$

$k = O(10^{-14})$

$T_0 = O(10^0)$

$\phi = O(10^{-1})$

From Equation C above, it can be seen that the pressure gradient and viscous effects terms on the right hand side are significantly larger than the terms on the left hand side for flows *PorePress* is designed for. Therefore, as discussed in Nield and Bejan, the transient and inertial terms in the momentum equation can be neglected for laminar flows through porous media where the kinematic viscosity is significantly larger than the permeability.

APPENDIX D. 1-D *PorePress* USER'S MANUAL

MAIN PROGRAM *POREPRESS*

All of the 1-D pressure calculation subroutines are called from the main program *PorePress*. The geometry file is only read in once, at the beginning of the main program, as the overall mesh should not change in the middle of the pressure solution process. Next, the main program starts iterating in time and subsequently reads in the FMAP file for the current time step, which gives active element information. After the active portion of the mesh is set, zero volume boundary elements are added to enforce the specified surface pressure boundary conditions from the FMAP file for the current time step. After the active and boundary elements for the current time step are identified, material properties are calculated for each of these elements.

Before calculating the final solution, the under-relaxation factor and preconditioning options need to be set. The under-relaxation factor (OMEGA) controls what fraction of the new calculated pressure at the current iteration is used. OMEGA is used to smooth out high gradients, and is an essential part in stabilization. Also, when a good initial guess is not available for a time step (typically the first time step), preconditioning is available. If the IFLAG is set at 0, a simplified form of the full equation will be used as a preconditioner to the full form of the solution. Otherwise, if IFLAG equals 1, then the full form of the equation is solved. Finally, the solver (steady state or transient) is iterated until convergence. Upon convergence of the solver, the velocity and pressure solutions are output for each element, and post processing output

options are available in the code. A schematic of the one-dimensional pressure solution process can be seen below in Figure D.1.

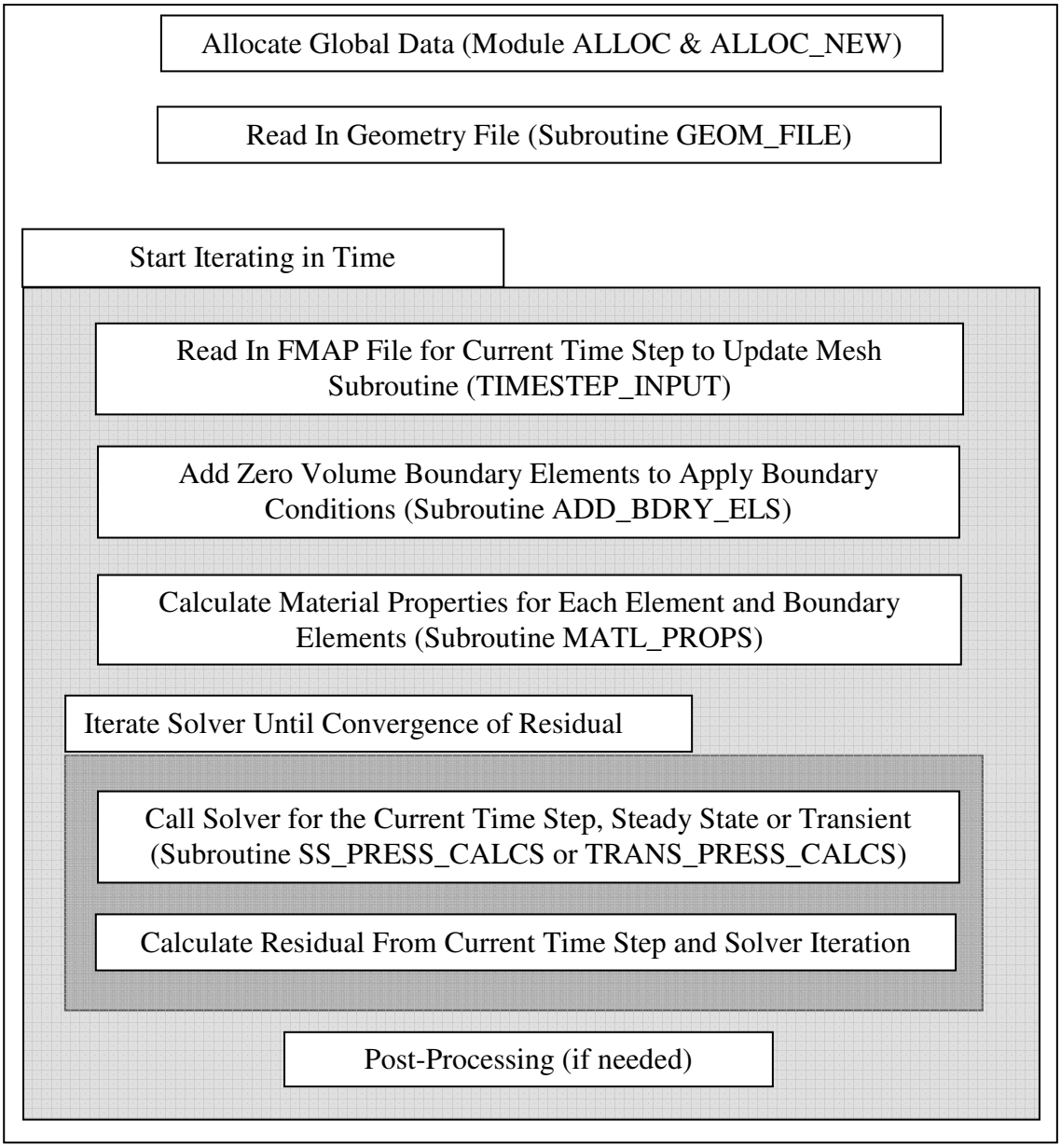

Figure D.1: Schematic of 1-D pressure solution process

Main Program *PorePress* Variables:

Variable	Data	Units
ITIMEIN	Iterate in time	--
TAPREV(el#)	Temperature from previous time step	R
ICT	Number of active elements in path	--
PRESSURE(el#)	Pressure for a given element, pressure solution is sent to the main program	$\frac{lbf}{ft^2}$
DELTAT	time step	s
DELTAR(el#)	Change in radial distance	ft
RADIUS(el#)	Radial distance to the centroid of an element	ft
RESSUM	Sum of residuals (b-Ax) for a path	$\frac{lbf}{ft^2}$
RESAVG	Average residual (b-AX) of a path, for an iteration	$\frac{lbf}{ft^2}$
RESOLD	Average residual (b-AX) of a path, for previous iteration	$\frac{lbf}{ft^2}$
RHOG(el#)	Gas density in an element	$\frac{lbm}{ft^3}$
VEL(el#)	Gas velocity in an element	ft/s
IFLAG	Flag used for preconditioning. If IFLAG=0, then a simplified form of the solution is solved and can be used as a preconditioner to the full solution. IFLAG=1 is the full form of the solution.	--
OMEGA	pressure relaxation factor	--

PREVPRESS(el#)	Pressure from previous iteration	$\frac{lbf}{ft^2}$
PRESS_OLD(el#)	Converged pressure from previous time step	$\frac{lbf}{ft^2}$
IBACKWALL	Back wall element for current time step	--
ILIVE	Number of active elements in the FMAP file	--

SUBROUTINE *GEOM_FILE*

The geometry file consists of basic information about each computational cell in the mesh. These are basic properties that describe the physical mesh that is input into *PorePress*. Each geometry file contains element paths in the form of lists of elements which constitute a path.

Subroutine *Geom_File* Variables:

Variable	Data	Units
NSURP	Number of paths given in geometry file	--
MATL(el#)	Number corresponding to the type of material in a computational cell	--
VOL(el#)	Volume of a computational cell	ft^3
AR(face#,el#)	Surface area of an element face	ft^2
PL(face#,el#)	Distance from the centroid of an element to an element face (at a 90^0 angle)	ft
KCON(face#,el#)	Neighboring element number	--
KPATHI(path#)	The first element (element closest to the surface) in a list of all the elements in a path	--
KPATHF(path#)	The last element (element farthest from the surface)	--

	in a list of all the elements in a path	
IGASPATH(path#)	The number of elements in a path	--
NOFACES(el#)	Total number of faces on an element	--

SUBROUTINE *TIMESTEP_INPUT*

The subroutine TIMESTEP_INPUT imports the FMAP file, which contains individual element information for each time step, for all of the interior elements in the mesh as well as the surface elements. Each time step contains two segments of information for a path: a list of surface elements and their properties followed by a list of interior elements and their properties (both active and inactive).

Before reading in the cell properties for the current time step, the temperatures for each element from the previous time step are stored in the array TAPREV for use in the discretization of the time derivative of temperature in the transient solution process. The elements and their properties are then read in for the current time step, replacing the same array values from the previous time step. The interior elements are all defined by a PFLAG value of 0 or 1 which identifies inactive and active elements, respectively. The number of active elements in a path are recorded as the integer value ILIVE. The array LIVEL contains the actual element numbers corresponding to the ILIVE numbers. An example of how this process works for a sample path can be seen below in FigureD.2.

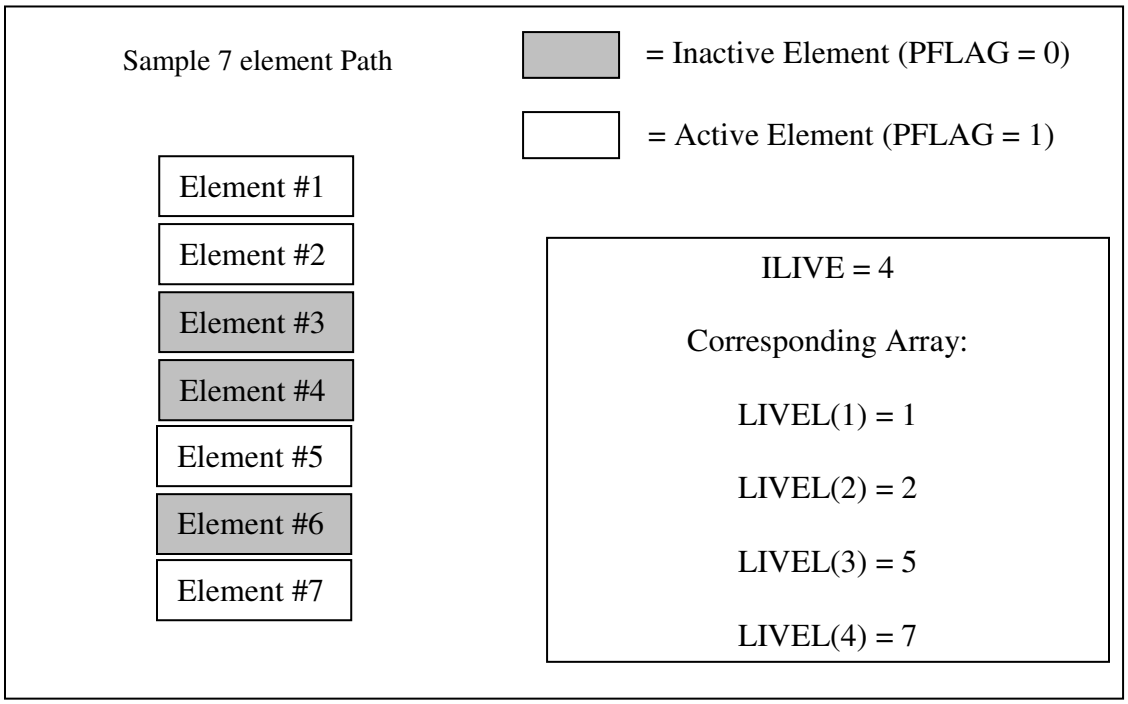

Figure D.2: Schematic showing how ILIVE and LIVEL correspond to active and inactive elements in a sample path

Subroutine *Timestep_Input* Variables:

Variable	Data	Units
TIME	Current time for the provided data	s
NUMPOINTS	Total number of elements (both surface and interior) in the FMAP file at the current time step	--
SURFOREL	Character value (S or E) representing surface or interior	--
NROW(el#)	Row in the mesh that contains the element	--
NBLOCK(el#)	Block in the mesh that contains the element	--
SURFX/SURFY(el#)	X and Y coordinates of a surface element	*ft*
TS(el#)	Surface temperature	*F*
PRESSURF(el#)	Surface pressure	$\frac{lbf}{ft^2}$

XREF/YREF(el#)	Coordinates for a surface element	ft
TA(el#)	Temperature of an element. Corresponds to both the char and the gas in a cell	R
RON(el#)	Density of the solid char in a computational cell	$\frac{lbm}{ft^3}$
OGRATE(el#)	Out-Gassing rate for an element (mass production)	$\frac{lbm}{s}$
PFLAG(el#)	Flag which tells if an element is active or dead (100% charred)	--
LIVEL	Array containing active element numbers	--
ILIVE	Number of active elements in the FMAP file	--
ISURFCOUNT	Number of surface elements at current time step	--
SURFACEFLAG(el#)	Tells which face of an element is a surface(1,2,3 or 4)	--
TSNEW(el#)	Surface temperature in Rankine, TS+459.67	R
TAPREV(el#)	temperature from previous time step	R
KBREAK	Number that represents break b/t surface elements and interior elements	--

SUBROUTINE *ADD_BDRY_ELS*

This subroutine is responsible for adding in fictitious zero volume boundary elements that are used to enforce the specified surface pressure boundary conditions. This routine takes the surface elements given from the FMAP file and adds them to the surface of the element which borders the surface. Figure D.3 below illustrates the process of adding zero volume boundary elements to a path of active interior elements.

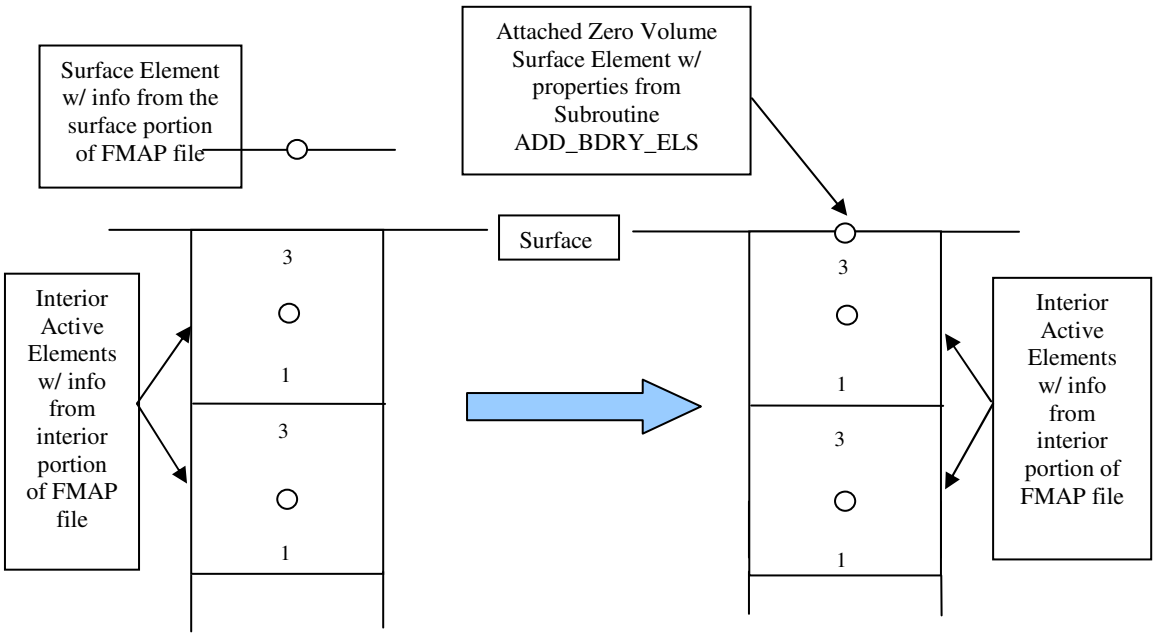

Figure D.3: Schematic illustrating the process of adding zero volume surface elements. The numbers inside each element represent face numbers used to establish connectivity.

The zero volume elements are added at the coordinates of SURFX and SURFY which were provided in the surface portion of the FMAP file. The new elements are given element numbers starting after the highest element number in the FMAP file (ILIVE + 1), and the list of total elements is also increased for each added element (the new total of elements is ILIVE + IEXTRACOUNT). The temperature and pressure from the surface portion of the FMAP file are also applied to the added element. The rest of the surface element material and physical properties are assumed to be the same as the interior element it is being added to. Finally, the connectivity relationships between the new surface element and the interior element it is being added onto are established by setting:

kcon(1,surface element) = interior element

kcon(3,interior element) = surface element

At the end of subroutine ADD_BDRY_ELS, zero volume surface elements have been fixed to the surface of the given interior element, added to the list of active elements (with pressure fixed at the given surface pressure), and given the necessary properties to maintain finite difference relationships from cell to cell.

Subroutine *Add_Bdry_Els* Variables:

Variable	Data	Units
ILIVE	Number of active elements in the FMAP file	--
LIVEL	Array containing active element numbers	--
PFLAG(el#)	Flag which tells if an element is active or dead (100% charred)	--
SURFACEFLAG(el#)	which face of an element is a surface(1,2,3 or 4)	--
PRESSURF(el#)	Surface pressure	$\frac{lbf}{ft^2}$
TSNEW(el#)	Surface temperature in Rankine, TS+459.67	*R*
SURFX/SURFY(el#)	X and Y coordinates of a surface element	*ft*
IEXTRACOUNT	Number of added zero volume surface elements	--
AR(face#,el#)	Surface area of an element face, added new values	ft^2
KCON(face#,el#)	Neighboring element number, added new values	--
XREF/YREF(el#)	Coordinates for a surface element, added new values	*ft*
PRESS(el#)	Pressure values for surface elements are set to honor BC	$\frac{lbf}{ft^2}$
TA(el#)	Temperature of an element. Corresponds to both the	R

	element and the gas in a cell, added new values	
RON(el#)	Density of the solid material in a computational cell, added new values	$\frac{lbm}{ft^3}$
LIVEL	Array containing active element numbers, and added boundary elements	--

SUBROUTINE *MATL_PROPS*

This subroutine simply calculates material properties for each computational cell based on density and temperature. One important note is that material properties need to be calculated for boundary elements as well as interior elements. Boundary element material properties are necessary to evaluate finite differences near the boundary.

Subroutine *Matl_Props* Variables:

Variable	Data	Units
LIVEL	Array containing active element numbers, and added boundary elements	--
ILIVE	Number of active elements in the FMAP file	--
PFLAG(el#)	Flag which tells if an element is active or dead (100% charred)	--
IEXTRACOUNT	Number of added zero volume surface elements	--
TA(el#)	Temperature of an element. Corresponds to both the element and the gas in a cell, added new values	R
RON(el#)	Density of the solid material in a computational cell	$\frac{lbm}{ft^3}$

DVIRG(el#)	Percent of virgin material remaining. 1 = virgin material, and 0 = fully charred material	--
DCHAR(el#)	Percent of charred material (= 1 – DVIRG)	--
POROS(el#)	Porosity of the remaining solid material. 0 = fully solid, and 1 = nothing remaining, or open air	--
ALPHA(el#)	Inverse of the permeability of the material in the r and z directions respectively	ft^{-2}
MU(el#)	Dynamic viscosity of the pyrolysis gas	$\frac{lbf-s}{ft^2}$
MWGT(el#)	Molecular weight of the pyrolysis gas	$\frac{lb}{lb-mol}$

SUBROUTINE *SS_PRESS_CALCS*

The steady state pressure solver routine (SS_PRESS_CALCS) iterates Equation D.2 below, which is the discretized form of Equation D.1, until the average pressure residual reaches a limit set by the user.

$$-\frac{K}{\mu}P\frac{\partial^2 P}{\partial r^2}-\frac{\partial P}{\partial r}\left(P\frac{\partial}{\partial r}\left(\frac{K}{\mu}\right)-\frac{K}{\mu}\left(\frac{P}{T}\frac{\partial T}{\partial r}-\frac{\partial P}{\partial r}-\frac{1}{r}P\right)\right)=\frac{RT}{M}\frac{\dot{m}_{OG}}{V} \qquad \text{(D.1)}$$

$$P_i^{k+1}=\left[\begin{array}{l} P_i^k\left(P_{i+1}^k+P_{i-1}^{k+1}\right)+P_i^k\left(P_i^k-P_{i-1}^{k+1}\right)\frac{\mu}{K}\Big|_i\left(\frac{K}{\mu}\Big|_{i+1}-\frac{K}{\mu}\Big|_i\right) \\ -\frac{P_i^k}{T_i}\left(P_i^k-P_{i-1}^{k+1}\right)\left(T_i^{n+1}-T_{i-1}^{n+1}\right)+\frac{\Delta r}{r_i}P_i^k\left(P_{i+1}^k-P_i^k\right)+\Delta r^2\frac{RT_i}{M}\frac{\dot{m}_{O-G}}{V} \\ \pm P_{i/i+1}^{k+1}\left(P_{i/i+1}^{k+1}-P_{i-1/i}^{k+1}\right)\end{array}\right] \div \left[\begin{array}{l} 2P_i^k\pm\left(P_{i/i+1}^{k+1}-P_{i-1/i}^{k+1}\right)\end{array}\right] \qquad \text{(D.2)}$$

In the subroutine, Equation D.2 is broken up into the six numerator terms labeled RNUMER1 through RNUMER6, and the two denominator terms are entered as RDENOM1 and RDENOM2. The pressure for each element is calculated as the sum of the RNUMER terms divided by the sum of the RDENOM terms.

The second numerator and second denominator terms in Equation D.2 above are dynamic difference terms, as described in Chapter 2. In the solver there are *if* statements for each element that apply either a forward or a backward difference for these terms based on the pressure gradient surrounding the element at the current iteration. The dynamic difference is used to increase numerical stability of the solver by adding diagonal dominance to the pressure coefficient matrix.

In conclusion, at a given time step, the steady state solver will solve Equation D.2 for each element using a dynamic difference on specific terms to add stability. Once the pressure from Equation D.2 is calculated for each element, the pressure solution is sent back to the main program. The main program will then determine if the average L1 norm of the residual vector (**Ax-b**) is too high (resulting in another iteration of the steady state solver) or if the pressure residual is sufficiently low, the program will end for the current time step.

Subroutine *SS_Press_Calcs* Variables:

Variable	Data	Units
LIVEL	Array containing active element numbers, and added	--

	boundary elements	
DELTAR(el#)	Change in radial distance	ft
ILIVE	Number of active elements in the FMAP file	--
PFLAG(el#)	Flag which tells if an element is active or dead (100% charred)	--
IEXTRACOUNT	Number of added zero volume surface elements	--
ALPHA(el#)	Inverse of the permeability of the material in the r and z directions respectively	ft^{-2}
MU(el#)	Dynamic viscosity of the pyrolysis gas	$\frac{lbf-s}{ft^2}$
MWGT(el#)	Molecular weight of the pyrolysis gas	$\frac{lb}{lb-mol}$
TA(el#)	Temperature of an element. Corresponds to both the element and the gas in a cell, added new values	R
RUGC	Universal gas constant = 1545.12	$\frac{\frac{lbf}{ft^2}\,ft^3}{lbmol-R}$
KCON(face#,el#)	Neighboring element number, added new values	--
OGRATE(el#)	Out-Gassing rate for an element (mass production)	$\frac{lbm}{s}$
RNUMER*i*	6 numerator terms in pressure calculation	$\frac{lbf}{ft^2}$
RDENOM*i*	2 denominator terms in pressure calculation	$\frac{lbf}{ft^2}$
PRESSURE(el#)	Pressure for a given element, pressure solution is sent to the main program	$\frac{lbf}{ft^2}$
IBACKWALL	Number of the element where the zero pressure gradient boundary condition is to be applied	--
RPRESSNEW	Temporary full pressure value at new iteration	$\frac{lbf}{ft^2}$

PRESS_OLD(el#)	Pressure from previous time step	$\frac{lbf}{ft^2}$
IFLAG	Flag used for preconditioning. If IFLAG=0, then a simplified form of the solution is solved and can be used as a preconditioner to the full solution. IFLAG=1 is the full form of the solution.	--
OMEGA	pressure relaxation factor	--

SUBROUTINE *TRANS_PRESS_CALCS*

The transient pressure solver routine (TRANS_PRESS_CALCS) iterates Equation D.4 below, which is the discretized form of Equation D.3, until the average pressure residual reaches a limit set by the user.

$$\phi\frac{\partial P}{\partial t}-\phi\frac{P}{T}\frac{\partial T}{\partial t}-\frac{K}{\mu}P\frac{\partial^2 P}{\partial r^2}-\frac{\partial P}{\partial r}\left(P\frac{\partial}{\partial r}\left(\frac{K}{\mu}\right)-\frac{K}{\mu}\left(\frac{P}{T}\frac{\partial T}{\partial r}-\frac{\partial P}{\partial r}-\frac{1}{r}P\right)\right)=\frac{RT}{M}\frac{\dot{m}_{OG}}{V} \tag{D.3}$$

$$\begin{aligned}P_i^{k+1}=\Bigg[\;&\phi\frac{\Delta r^2}{\Delta t}\left.\frac{\mu}{K}\right|_i P_i^n+\phi\frac{\Delta r^2}{\Delta t}\left.\frac{\mu}{K}\right|_i P_i^k\left(\frac{T_i^{n+1}-T_i^n}{\Delta t}\right)+P_i^k\left(P_{i+1}^k+P_{i-1}^{k+1}\right)\\&+P_i^k\left(P_i^k-P_{i-1}^{k+1}\right)\left.\frac{\mu}{K}\right|_i\left(\left.\frac{K}{\mu}\right|_{i+1}-\left.\frac{K}{\mu}\right|_i\right)-\frac{P_i^k}{T_i^{n+1}}\left(P_i^k-P_{i-1}^{k+1}\right)\left(T_i^{n+1}-T_{i-1}^{n+1}\right)\\&+\frac{\Delta r}{r}P_i^k\left(P_{i+1}^k-P_i^k\right)+\frac{\Delta r^2}{\Delta t}\left.\frac{\mu}{K}\right|_i\frac{RT_i}{M}\frac{\dot{m}_{O-G}}{V}\pm P_{i/i+1}^{k+1}\left(P_{i/i+1}^{k+1}-P_{i-1/i}^{k+1}\right)\;\Bigg]\\\div\;\Bigg[\;&\phi\frac{\Delta r^2}{\Delta t}\left.\frac{\mu}{K}\right|_i+2P_i^k\pm\left(P_{i/i+1}^{k+1}-P_{i-1/i}^{k+1}\right)\;\Bigg]\end{aligned} \tag{D.4}$$

The transient solver structure is the same as the steady state structure which is described in depth above. The only difference is the addition of the two transient terms from Equation D.4 which result in two additional RNUMER terms and one additional RDENOM term.

One important note for the transient solver is that the size of the time step can affect the stability of the problem. For the given FMAP files, the largest time step used by *PorePress* was 0.5 seconds. The code has not been used with any FMAP files with larger time steps. Thus, it is important for the time step size to be monitored closely for future uses if a larger step size is desired.

Subroutine *Trans_Press_Calcs* Variables:

Variable	Data	Units
DELTAT	time step	s
DELTAR(el#)	Change in radial distance	ft
LIVEL	Array containing active element numbers, and added boundary elements	--
ILIVE	Number of active elements in the FMAP file	--
PFLAG(el#)	Flag which tells if an element is active or dead (100% charred)	--
IEXTRACOUNT	Number of added zero volume surface elements	--
PREVPRESS(el#)	Pressure for a given element, pressure solution from a previous time step is input into the routine	$\frac{lbf}{ft^2}$

ALPHA(el#)	Inverse of the permeability of the material in the r and z directions respectively	ft^{-2}
MU(el#)	Dynamic viscosity of the pyrolysis gas	$\frac{lbf-s}{ft^2}$
MWGT(el#)	Molecular weight of the pyrolysis gas	$\frac{lb}{lb-mol}$
TA(el#)	Temperature of an element. Corresponds to both the element and the gas in a cell, added new values	R
TAPREV(el#)	Temperature of an element from previous time step. Corresponds to both the element and the gas in a cell, added new values	R
RUGC	Universal gas constant = 1545.12	$\frac{\frac{lbf}{ft^2}\,ft^3}{lbmol-R}$
KCON(face#,el#)	Neighboring element number, added new values	--
OGRATE(el#)	Out-Gassing rate for an element (mass production)	$\frac{lbm}{s}$
RNUMER*i*	8 numerator terms in pressure calculation	$\frac{lbf}{ft^2}$
RDENOM*i*	3 denominator terms in pressure calculation	$\frac{lbf}{ft^2}$
PRESSURE(el#)	Pressure for a given element, pressure solution is sent to the main program	$\frac{lbf}{ft^2}$
IBACKWALL	Number of the element where the zero pressure gradient boundary condition is to be applied	--
RPRESSNEW	Temporary full pressure value at new iteration	$\frac{lbf}{ft^2}$
PRESS_OLD(el#)	Pressure from previous time step	$\frac{lbf}{ft^2}$

IFLAG	Flag used for preconditioning. If IFLAG=0, then a simplified form of the solution is solved and can be used as a preconditioner to the full solution. IFLAG=1 is the full form of the solution.	--
OMEGA	pressure relaxation factor	--

APPENDIX E. 2-D *PorePress* USER'S MANUAL

MAIN PROGRAM *POREPRESS*

All of the main subroutines used in the 2-D pressure solution process are called from the *PorePress* main program. Initially, all of the data arrays are allocated in the ALLOC and ALLOC_NEW modules, which are used in all of the subroutines called from the main program. The program begins iterating in time before it inputs the Geometry file, unlike the 1-D case. This is done to effectively zero out all of the artificial zero volume boundary elements and forced connectivity relationships that were added in the previous time step. The FMAP file is then read in with all of the mesh properties for the current time step.

Once all of the active and inactive elements are defined, the subroutine *STENCIL* adds in the zero volume boundary elements which will enforce the known surface pressure and zero pressure gradient boundary conditions. The new added boundary elements are marked inactive as far as pressure calculations are concerned, but available to be used in other active element's 8 point stencil. Once the 8 point stencils are calculated, subroutine *QR* sets up the linear least-squares system of equations based on a 2^{nd} order Taylor Series expansion, and then computes the QR factorization of the coefficient matrices. The final step of subroutine *QR* is to calculate the derivative approximation coefficients based on the factorized coefficient matrices.

The next step in the solution process is to gather the material properties for all of the active and boundary elements. The main program then sets the initial conditions and boundary conditions before iterating the pressure solver (steady state or transient). The pressure solver is iterated until the average residual converges to constraints set by the user. Upon convergence, the pressure and velocity distributions are returned and made available for post-processing if needed. A schematic of the 2-D pressure solution process can be seen below in Figure E.1.

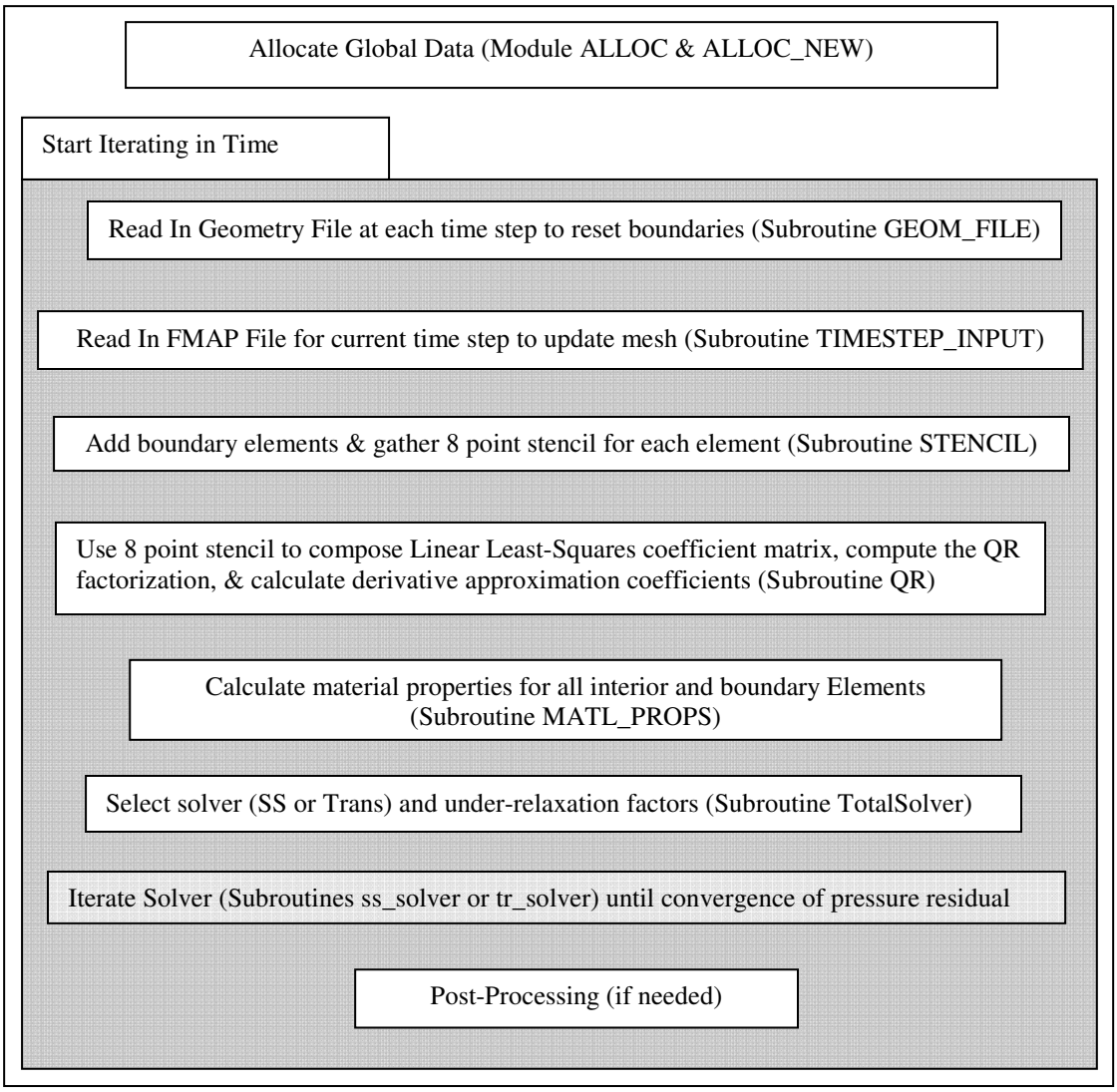

Figure E.1. Schematic of One-Dimensional Pressure Solution Process

SUBROUTINE *GEOM_INPUT*

Initially, the 2-D *PorePress* code receives mixed unstructured meshes with established connectivity from cell to cell and defined geometric cell properties, this is identically similar to the geometry file from the 1-D case. However, in the 1-D case predetermined paths of gas flow were used in the solution process, but in the 2-D case these paths are ignored and the gas is allowed to flow through any active region in the mesh. The geometry file consists of cell properties such as: cell type (quad or tri), cell connectivity relationships, cell volume, cell face area, and cell vertex locations. These are basic properties that describe the physical mesh that is input into *PorePress*.

Subroutine *GEOM_INPUT* variables:

Variable	Data	Units
NSURP	Number of paths given in geometry file	--
MATL(el#)	Number corresponding to the type of material in a particular computational cell	--
VOL(el#)	Volume of an entire computational cell	ft^3
AR(face#,el#)	Surface area of an element face	ft^2
PL(face#,el#)	Distance from the centroid of an element to an element face (at a 90^0 angle)	*ft*
KCON(face#,el#)	Neighboring element number	--
KF(face#,el#)	Face number which neighboring element is connected to the current element by	--
KPATHI(path#)	The first element (element closest to the surface) in a list of all the elements in a path	--

KPATHF(path#)	The last element (element farthest from the surface) in a list of all the elements in a path	--
IGASPATH(path#)	The number of elements in a path	--
NOFACES(el#)	Total number of faces on an element	--

SUBROUTINE *FMAP_INPUT*

The FMAP file contains information about the mesh that changes in time. Each time step has a corresponding FMAP file which contains updated material and cell properties. The mesh needs to get updated during the pyrolysis process when the surface elements will burn off, and elements that were previously considered interior elements are now surface elements. Also, as the temperature effects propagate inside the liner, certain elements that were once considered inactive become active. Each FMAP file lists the current active surface and interior elements at an individual time step. The FMAP file also updates individual cell material properties such as: out-gassing rate, temperature, density and surface pressures.

In the FMAP subroutine the number of active elements in the mesh (ALIVE), and an array corresponding to only active elements (LIVEL) are defined and set. This is done to prevent wasted computational time looping through thousands of inactive elements in future routines. An example of how ALIVE and LIVEL work can be seen below.

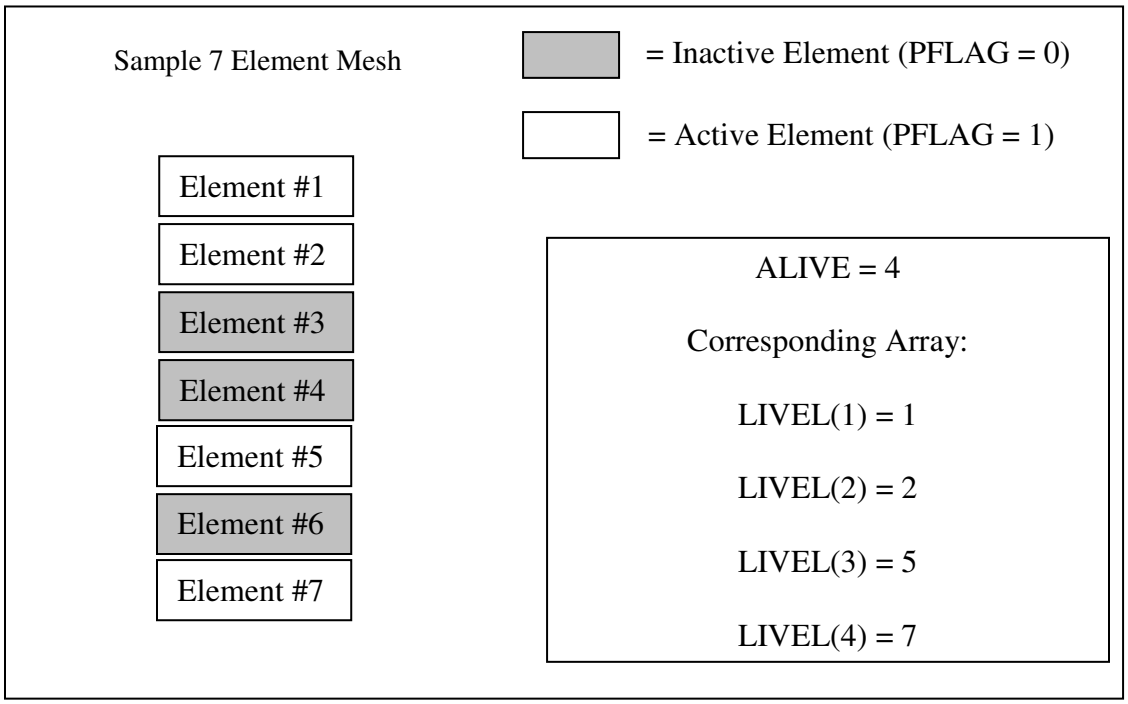

Figure E.2: Schematic showing how ALIVE and LIVEL correspond to active and inactive elements in a sample path

Subroutine *FMAP_INPUT* variables:

Variable	Data	Units
TIME	Current time for the provided data	s
POINTS	Total number of points in the FMAP file	--
NROW(el#)	Row in the mesh that contains the element	--
NBLOCK(el#)	Block in the mesh that contains the element	--
XREF/YREF(el#)	Coordinates for an element's centroid	*in*
TA(el#)	Temperature of an element. Corresponds to both the element and the gas in a cell	R

RON(el#)	Density of the solid material in a computational cell	$\frac{lbm}{ft^3}$
OGRATE(el#)	Out-Gassing rate for an element (mass production)	$\frac{lbm}{s}$
PFLAG(el#)	Flag which tells if an element is active or dead (100% charred)	--
PRESSURF(el#)	Surface pressure	$\frac{lbf}{ft^2}$
TSNEW(el#)	Surface temperature	F
TAOLD(el#)	Temperature of element from previous time step	R
LIVEL(#)	Array which contains only active element numbers	--
DELTAT	Time step for current FMAP file	s
ALIVE	Total number of active elements in FMAP file	--

SUBROUTINE *STENCIL*

The subroutine *STENCIL* is responsible for setting up the boundary conditions over the mesh and constructing an eight point stencil for each element. The first step in *STENCIL* is to mark all active boundary elements (surfaces, back wall and sides) in the mesh for the current time step. Then the active boundary elements are marked further into two categories based on the boundary condition to be applied: surface pressure elements at which a known pressure acts as the boundary condition and zero pressure gradient surface elements at which a zero pressure gradient is assumed. All surface elements that are not input from the FMAP file with a defined surface pressure are designated zero pressure gradient surface elements. These elements are typically located on the sides of

nozzle liner, and the interior barriers between active and inactive elements (referred to as the back wall). Specified pressure surface elements are typically located on the surface that is exposed to the heated flow.

Surface elements are added in the form of new zero volume boundary elements on the border of the actual FMAP element. The numbering of the new zero volume surface elements begins at one plus the highest elements number in the entire mesh (ALIVE+1). An example of how surface elements are added into the mesh can be seen below in Figure E.3.

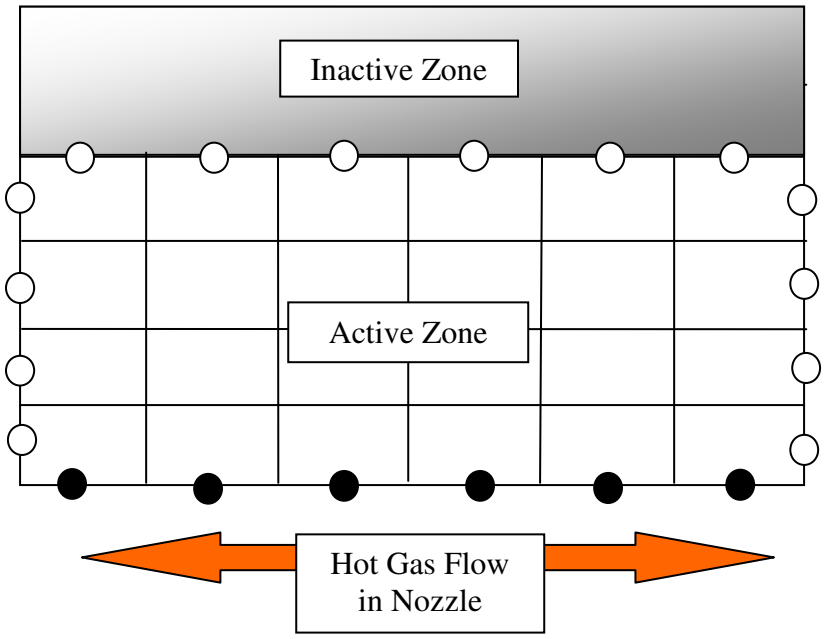

Figure E.3: Sample mesh and added zero volume boundary elements

In Figure E.3, the filled circles represent zero volume surface elements that have a specified surface pressure, and the solid white circles represent zero pressure gradient surface elements. The zero volume surface elements have their own pressure and

temperature based on the surface element information given in the FMAP file. All of the other surface element material properties (permeability, porosity, etc.) are assumed to be equal to the interior element it is attached to. The final step in the process of adding zero volume surface elements is to mark the elements as inactive as far as pressure calculations are concerned, but available to be included in future eight point stencil neighborhoods.

The main point of the *STENCIL* routine is to now assemble eight point stencils for each active element using specific neighboring elements to minimize the condition number of the coefficient matrix used later in the code. The stencil collection process is described in detail in the Condition Number and Stencil Quality Section 3.2.4 of this report.

Once all the neighboring elements have been gathered for the current element, the next step is to sift through the neighboring elements to create an eight point stencil which will produce a favorable condition number. First, the subroutine *STENCIL* removes the current element from the list of potential stencil neighbors. Next, *STENCIL* eliminates all repeated elements encountered in the search and stores repeated elements only once. Finally, the routine creates an ordered list which is ordered by priority sweep number first and then centroid-to-centroid distance from the current element to the listed element. The final step in the stencil process is to store the top eight elements in the list based on priority sweep number with centroid to centroid distances being used as the tie-breaker. A

sample list up to the second priority sweep level is provided below in Figure E.4 to illustrate the process.

Sample Element Number	Sweep Level Number	Radial Distance
121	1	0.01
125	1	0.01
126	1	0.01
123	1	0.01
135	2	0.025
201	2	0.025
525	2	0.025
526	2	0.025
202	2	0.03
524	2	0.03
120	2	0.03

Figure E.4: Top of sorted list for sample mesh, which is sorted first by level number (middle column), and then by centroid to centroid radius distance (right column)

In the case of the above sample mesh elements 121, 125, 126, 123, 135, 201, 525, and 526 would be stored as the eight point stencil for the sample element in the *stelem* array for later use in the code. This process is carried out for all active elements in the mesh, so that eventually every active element has an array of stored eight point stencil neighboring elements.

The distribution of the elements that are selected for the stencils play a significant role in the overall stability of the *PorePress* 2-D code. An ideal stencil would have an

equal spread of elements in each direction around the central element. If a stencil contains too many elements bunched together on one side of the central element or a side with no stencil points, the effect will be near-singular coefficient matrices. If an element has an extremely poor stencil or if too many elements in an area of the mesh have poorly defined stencils, *PorePress* will not be able to obtain a converged solution. More detail on the significance of stencils and stability can be seen in the Condition Number and Stencil Quality Section 3.2.4 of this report.

Subroutine *STENCIl* variables:

Variable	Data	Units
PFLAG(el#)	Flag which tells if an element is active or dead (100% charred)	--
LIVEL(#)	Array which contains only active element numbers	--
ALIVE	Total number of active elements in FMAP file	--
KCON(face#,el#)	Neighboring element number	--
KF(face#,el#)	Face number which neighboring element is connected to the current element by	--
EX/EY(node#,el#)	X/Y coordinates of a node number for an element	*in*
XREF/YREF(el#)	Coordinates for an element's centroid	*in*
SURFX/Y(el#)	Coordinates for a surface element	*in*
NOFACES(el#)	Total number of faces on an element	--
MATL(el#)	Material number of an element	--
GRADFLAG(el#)	Flag which tells when to apply zero pressure gradient boundary condition	--

TA(el#)	Temperature of an element. Corresponds to both the element and the gas in a cell	R
RON(el#)	Density of the solid material in a computational cell	$\frac{lbm}{ft^3}$
PRESSURF(el#)	Surface pressure	$\frac{lbf}{ft^2}$
SURFACEFLAG(el#)	Tells which face of an element is connected to a surface. Used to add fictitious boundary elements	--
CONFLAG(el#)	Flags an element that is available to be used in future connectivity lists (includes interior and boundary elements)	--
ALIVEP	Total number of active interior elements and fictitious zero volume boundary elements	--
BDRY_ORIGIN(el#)	Original surface element number that a fictitious zero volume boundary element has been connected to	--
RADI/2/3/4(el#)	Temporary arrays containing radii from centroid of current element to neighboring element	*in*
ELIST/2/3/4(el#)	Temporary arrays of element numbers for elements surrounding the current element	--
NEIGHFLAG(el#)	Flag which prioritizes neighboring elements based on proximity by means connectivity to current element	--
ICOUNT/2/3/4	Total number of neighboring connections recognized in levels 1,2,3, and 4 of connectivity	--
REPEAT(el#)	Temporary flag which tells when an element has already been added to a connectivity list (used to remove repeats of the same element in a connectivity list)	--
STELEM(#,el#)	Matrix which stores the element numbers of the 8 neighboring elements selected from the routine to use in the Taylor Series approximation. STELEM(9,el#) is the actual element number of the current element.	--

SUBROUTINE *QR*

Once the eight point stencil for each element has been stored (in *stelem* from subroutine *Stencil*), the next step is to perform the QR decomposition by means of the Modified Gram-Schmidt Process and use the results to calculate the derivative approximation coefficients. The first step in *QR* is to set up the coefficient (***A***) matrix for each active element, which will eventually be decomposed. The coefficient matrices will have the form of the matrix below in E.1, which is based on a second order Taylor Series approximation.

$$A = \begin{bmatrix} \Delta z_{01} & \Delta r_{01} & 0.5\Delta z_{01}^2 & \Delta z_{01}\Delta r_{01} & 0.5\Delta r_{01}^2 \\ \Delta z_{02} & \Delta r_{02} & 0.5\Delta z_{02}^2 & \Delta z_{02}\Delta r_{02} & 0.5\Delta r_{02}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Delta z_{0n} & \Delta r_{0n} & 0.5\Delta z_{0n}^2 & \Delta z_{0n}\Delta r_{0n} & 0.5\Delta r_{0n}^2 \end{bmatrix} \qquad \text{(E.1)}$$

Matrix E.1 above is assembled using centroid to centroid differences in the r- and z-directions respectively. Each row in the coefficient matrix corresponds to centroid differences between the current element and one of the eight elements in the stencil.

In order to add stability to the solution process, matrix E.1 above is multiplied by an inverse weight, resulting in the $\mathbf{A}_{wgt}$ matrix below. For more detail on the inverse weighting scheme see Section 3.2.3 in the main part of this thesis.

$$A_{wgt} = \begin{bmatrix} \frac{1}{\sqrt{\Delta r_{01}^2 + \Delta z_{01}^2}} \begin{pmatrix} \Delta z_{01} & \Delta r_{01} & 0.5\Delta z_{01}^2 & \Delta z_{01}\Delta r_{01} & 0.5\Delta r_{01}^2 \end{pmatrix} \\ \frac{1}{\sqrt{\Delta r_{02}^2 + \Delta z_{02}^2}} \begin{pmatrix} \Delta z_{02} & \Delta r_{02} & 0.5\Delta z_{02}^2 & \Delta z_{02}\Delta r_{02} & 0.5\Delta r_{02}^2 \end{pmatrix} \\ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ \frac{1}{\sqrt{\Delta r_{08}^2 + \Delta z_{08}^2}} \begin{pmatrix} \Delta z_{08} & \Delta r_{08} & 0.5\Delta z_{08}^2 & \Delta z_{08}\Delta r_{08} & 0.5\Delta r_{08}^2 \end{pmatrix} \end{bmatrix} \tag{E.2}$$

from here on out in the Appendices the $\mathbf{A}_{wgt}$ matrix will simply be referred to as the **A** matrix and it will be assumed that the **b** vector is also multiplied by the corresponding inverse weights.

Next, the linear least-squares technique is employed in the code to yield a best fit solution to the over-determined system. This technique involves multiplying both sides of the linear system by the transpose of the coefficient **A** matrix. The *QR* routine then computes the QR decomposition of the $\mathbf{A}^T\mathbf{A}$ matrix by means of the Modified Gram-Schmidt Process.

For an *n* x *n* matrix $\mathbf{A}^T\mathbf{A}$ with *n* linearly independent column vectors: $\mathbf{a}_1, \mathbf{a}_2, \ldots \mathbf{a}_n$ the first step in the modified Gram-Schmidt process is to calculate the norm of each of the column vectors, and then normalize each column vector. The following modified

Gram-Schmidt algorithm begins at the k=1 column vector and then cycles through each on:

$$r_{kk} = \left(\sum_{i=1}^{n} a_{ik}^2 \right)^{1/2} \qquad (i=1\ldots,n) \qquad \text{(E.3)}$$

$$q_{ik} = \frac{a_{ik}}{r_{kk}} \qquad (i=1\ldots,n) \qquad \text{(E.4)}$$

where i represents the individual elements in the k^{th} column vector. Equation E.3 begins to form the diagonals of the upper triangular **R** matrix with the magnitude of each of the column vectors. Equation E.4 begins to form the orthogonal matrix **Q** with a normalization process. Next, the projection of each $\mathbf{a}_j$ (j=k+1,…n) onto $\mathbf{q}_k$ is calculated, and the projections are subtracted off in the original matrix of column vectors for the next iteration.

$$r_{kj} = \sum_{i=1}^{n} q_{ik} a_{ij} \qquad (i=1\ldots,n) \qquad \text{(E.5)}$$

$$a_{ij} = a_{ij} - q_{ik} r_{kj} \qquad (i=1\ldots,n) \qquad \text{(E.6)}$$

Increase k to k+1 and return to Equation E.3 [Strang, 1993] Once Equations E.3-E.6 have gone through all n column vectors, the result is the upper triangular matrix **R** and orthogonal matrix **Q** from the QR factorization of the coefficient matrix **A**.

The final matrix manipulation step in the QR decomposition process is to calculate the inverse of the **R** matrix. A simple back substitution can be used to calculate $\mathbf{R}^{-1}$, due to its upper triangular form.

Finally, the *QR* routine calculates the coefficients that make up the first and second order derivatives based on a second order Taylor Series expansion. The coefficients that represent the derivative approximations are as follows:

$$\left[\frac{\partial}{\partial z} \quad \frac{\partial}{\partial r} \quad \frac{\partial^2}{\partial z^2} \quad \frac{\partial}{\partial z \partial r} \quad \frac{\partial^2}{\partial r^2}\right]^T = \left(\mathbf{R}^{-1}\mathbf{Q}^T\right)\mathbf{A}^T\mathbf{b} \tag{E.7}$$

where $\mathbf{A}^T$ is the transpose of the original coefficient matrix after the inverse weights have been applied. In *QR*, the derivative approximations found on the left hand side of Equation E.7 are merely a collection of coefficients. An example of what the 1st derivatives in the r- and z-direction for some variable, ϕ, at an active elements (*0*) in the mesh can be seen below:

$$\left.\frac{\partial \phi}{\partial z}\right|_0 = c_1\phi_1 + c_2\phi_2 + c_3\phi_3 + c_4\phi_4 + c_5\phi_5 + c_6\phi_6 + c_7\phi_7 + c_8\phi_8 + c_0\phi_0 \tag{E.8}$$

$$\left.\frac{\partial \phi}{\partial r}\right|_0 = d_1\phi_1 + d_2\phi_2 + d_3\phi_3 + d_4\phi_4 + d_5\phi_5 + d_6\phi_6 + d_7\phi_7 + d_8\phi_8 + d_0\phi_0 \tag{E.9}$$

where the coefficients ,c_i and d_i, represent the Taylor Series approximation coefficients. The subscripts *1* through *8* represent each of the 8 neighboring elements contained in the stencil of the current element (represented with the subscript *0*). Specifically, the coefficients in the r- and z-direction for the first order derivative approximation would be obtained by using the following equations in conjunction with each of the current element's 8 neighboring elements ($i=1 \rightarrow 8$) from the stencil:

$$c_i = (R^{-1}Q^T)_{1,1}\Delta z_{0i} + (R^{-1}Q^T)_{1,2}\Delta r_{0i} + (R^{-1}Q^T)_{1,3}\frac{1}{2}\Delta z_{0i}^2 + (R^{-1}Q^T)_{1,4}\Delta z_{0i}\Delta r_{0i} + (R^{-1}Q^T)_{1,5}\frac{1}{2}\Delta r_{0i} \quad \text{(E.10)}$$

$$c_0 = -\sum_{i=1}^{8} c_i \quad \text{(E.11)}$$

$$d_i = (R^{-1}Q^T)_{2,1}\Delta z_{0i} + (R^{-1}Q^T)_{2,2}\Delta r_{0i} + (R^{-1}Q^T)_{2,3}\frac{1}{2}\Delta z_{0i}^2 + (R^{-1}Q^T)_{2,4}\Delta z_{0i}\Delta r_{0i} + (R^{-1}Q^T)_{2,5}\frac{1}{2}\Delta r_{0i} \quad \text{(E.12)}$$

$$d_0 = -\sum_{i=1}^{8} d_i \quad \text{(E.13)}$$

The end result of the routine *QR* is five arrays of nine coefficients for each active element in the mesh that represent derivative approximations. Each of the five coefficient

arrays for each element correspond to the five derivative approximations represented in the left hand side of Equation E.7 above.

At the end of the *QR* routine is an option to calculate the condition number for each $\mathbf{A}^T\mathbf{A}$ matrix, which yields a better understanding of the stability of the matrices undergoing the QR decomposition.. Poorly conditioned matrices will result in a singular or near-singular **R** matrix. The solution process requires the calculation of $\mathbf{R}^{-1}$, which will contain copious amounts of errors if **R** is singular or near singular. Typically, the matrices will be well conditioned as long as the 8 neighboring elements in the stencil are not mostly clustered in one area. In other words, the stability of the derivative approximation depends greatly on the spread of elements around the current element in the eight point stencil. More detail on condition numbers and their role in stability of the *PorePress* code can be seen in Sections 3.2.3 and 3.2.4 in the main part of the thesis

Subroutine *QR* variables:

Variable	Data	Units
PFLAG(el#)	Flag which tells if an element is active or dead (100% charred)	--
LIVEL(#)	Array which contains only active element numbers	--
ALIVE	Total number of active elements in FMAP file	--
ALIVEP	Total number of active interior elements and fictitious zero volume boundary elements	--
DELTAX/Y0(#)	Distance in the x (or z)/ y(or r) direction from centroid to centroid for current element to neighboring element	*in*

XREF/YREF(el#)	Coordinates for an element's centroid	*in*
Aoo(#,#)	Original 8 by 5 coefficient matrix for an element based on Taylor Series expansion at current element	--
INV_WGT(ROW#)	Inverse weighting value (based on radii) divided by each row in Aoo to add stability	--
aT(#,#)	Transpose of Aoo, used for linear least-squares procedure	--
A(#,#)	5 by 5 matrix for current element which is A^TA	--
Ao(#,#)	Original A^TA matrix before alterations are made	--
R(#,#)	5 by 5 upper triangular matrix resulting from the QR decomposition	--
Q(#,#)	5 by 5 orthogonal matrix resulting from the QR decomposition	--
IDENT(#,#)	5 by 5 identity matrix used to take inverse of R	--
Rinv(#,#)	5 by 5 matrix which is the inverse of R	--
SUMS(#,#)	5 by 5 temporary matrix used in taking the inverse of R	--
Qt(#,#)	5 by 5 transpose of the Q matrix	--
RiQt(#,#)	5 by 5 matrix which represents the factor of $R^{-1}*Q^T$	--
CX(#,el#)	Matrix containing coefficients used to approximate the first order derivative in the x-direction (or z-dir)	--
CY(#,el#)	Matrix containing coefficients used to approximate the first order derivative in the y-direction (or r-dir)	--
CX2(#,el#)	Matrix containing coefficients used to approximate the second order derivative in the x-direction (or z-dir)	--
CY2(#,el#)	Matrix containing coefficients used to approximate the second order derivative in the y-direction (or r-dir).	--
Atemp(#,#)	Temporary matrix representing Ao	--
EIGS(#)	5 eigenvalues corresponding to each Ao matrix	--

COND_NO(el#)	Condition number for each element's coefficient matrix	--

SUBROUTINE *MATL_PROPS*:

Subroutine *MATL_PROPS* uses the individual cell data from the FMAP file and the information from the added boundary elements to determine physical properties of the pyrolysis gas and the carbonaceous char. One important note is to loop through ALIVEP, not just ALIVE, and get material properties for active interior elements and added zero volume boundary elements. The material properties for the added boundary elements are needed for material derivative terms later on.

Subroutine *MATL_PROPS* variables:

Variable	Data	Units
PFLAG(el#)	Flag which tells if an element is active or dead (100% charred)	--
LIVEL(#)	Array which contains only active element numbers	--
ALIVE	Total number of active elements in FMAP file	--
ALIVEP	Total number of active interior elements and fictitious zero volume boundary elements	--
DVIRG(el#)	Percent of virgin material remaining. 1 = virgin material, and 0 = fully charred material	--
POROS(el#)	Porosity of the remaining solid material. 0 = fully solid, and 1 = nothing remaining, or open air	--

ALPHAR/Z(el#)	Inverse of the permeability of the material in the r and z directions respectively	ft^{-2}
MU(el#)	Dynamic viscosity of the pyrolysis gas	$\frac{lbf-s}{ft^2}$
MWGT(el#)	Molecular weight of the pyrolysis gas	$\frac{lb}{lb-mol}$

SUBROUTINE *TOTALSOLVER*

This routine controls the solvers (steady state and transient) and their individual properties that may add stability to the solvers. For the case of *PorePress*, steep spatial and time gradients are typically the source for instabilities. Therefore, in the iterative solution procedure it is often necessary to stabilize the solution by using an under-relaxation factor OMEGA(ω), which controls the permissible change in pressure from one iteration to the next as:

$$P_0^{k+1} = P_0^k * \omega + P_0^{k+1} * (1-\omega) \tag{E.14}$$

Another technique used in *TotalSolver* is to solve a subset of the full equation to get a good initial guess for the pressure distribution. This is typically done by solving a non-linear form of the full equation (Equation E.15 below) and then use that solution as an initial guess to either the steady state or transient form of the full solution (Equation E.16 below).

$$-\frac{P}{T}\frac{K_r}{\mu}\frac{\partial^2 P}{\partial r^2}-\frac{P}{T}\frac{K_z}{\mu}\frac{\partial^2 P}{\partial z^2}=\frac{R}{M}\frac{\dot{m}_{OG}}{V} \qquad \text{(E.15)}$$

$$\phi\frac{\partial P}{\partial t}-\phi\frac{P}{T}\frac{\partial T}{\partial t}-\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{P}{T}\left(\frac{K_r}{\mu}\frac{\partial P}{\partial r}\right)\right)-\frac{\partial}{\partial z}\left(\frac{P}{T}\left(\frac{K_z}{\mu}\frac{\partial P}{\partial z}\right)\right)=\frac{R}{M}\frac{\dot{m}_{OG}}{V} \qquad \text{(E.16)}$$

The relaxation techniques are typically used only in the steady state case when a good initial guess of the pressure is not available. However, during time integration, the best initial guess for a time step is typically the solution for the previous time step. Therefore, the solution relaxation technique is only used when an appropriate initial guess for a solver is not available.

Finally, the permeability and out-gassing rates can be relaxed in *TotalSolver* by introducing these values gradually into the solution process. The solution with the minimized values can be passed as an initial approximation for the full solution with the full permeability or out-gassing values. Also, the routine exports pyrolysis gas velocity and pressure solutions for desired time steps for post-processing if necessary.

Subroutine *TotalSolver* variables:

Variable	Data	Units
ITERO	Solver iteration counter	--

tempOGrate	Temporary form of out-gassing used for relaxation. i.e. Solution with tempOGrate=OGRATE/100.0 is sent as initial guess to full equation with tempOGrate=OGRATE	$\frac{lbm}{s}$
kssflag	equal to 0 for Poisson form of equation equal to 1 for full form of equation	--
mm	number of iterations for solver	--

SUBROUTINE *SS_SOLVER*

The steady state pressure solver routine (SS_SOLVER) iterates Equation E.18 below, which is the discretized form of Equation E.17, until the average pressure residual reaches a limit set by the user.

$$-\frac{PK_r}{\mu}\left(\frac{\partial^2 P}{\partial r^2}+\frac{\mu}{K_r}\frac{\partial P}{\partial r}\frac{\partial}{\partial r}\left(\frac{K_r}{\mu}\right)+\frac{1}{P}\frac{\partial P}{\partial r}\frac{\partial P}{\partial r}-\frac{1}{T}\frac{\partial T}{\partial r}\frac{\partial P}{\partial r}+\frac{1}{r}\frac{\partial P}{\partial r}\right)$$

$$-\frac{PK_z}{\mu}\left(\frac{\partial^2 P}{\partial z^2}+\frac{\mu}{K_z}\frac{\partial P}{\partial z}\frac{\partial}{\partial z}\left(\frac{K_z}{\mu}\right)+\frac{1}{P}\frac{\partial P}{\partial z}\frac{\partial P}{\partial z}-\frac{1}{T}\frac{\partial T}{\partial z}\frac{\partial P}{\partial z}\right)=\frac{RT}{M}\frac{\dot{m}_{OG}}{V} \qquad \text{(E.17)}$$

$$P_0^{k+1} = \left[\frac{P_0^k K_r}{\mu} \left(\left[f_1 P_1 + \cdots + f_8 P_8 \right] + \frac{\mu}{K_r} \frac{\partial P}{\partial r} \frac{\partial}{\partial r} \left(\frac{K_r}{\mu} \right) + \frac{1}{P_0^k} \frac{\partial P}{\partial r} \frac{\partial P}{\partial r} - \frac{1}{T} \frac{\partial T}{\partial r} \frac{\partial P}{\partial r} + \frac{1}{r} \frac{\partial P}{\partial r} \right) \right.$$

$$\left. + \frac{P_0^k K_z}{\mu} \left(\left[g_1 P_1 + \cdots + g_8 P_8 \right] + \frac{\mu}{K_z} \frac{\partial P}{\partial z} \frac{\partial}{\partial z} \left(\frac{K_z}{\mu} \right) + \frac{1}{P_0^k} \frac{\partial P}{\partial z} \frac{\partial P}{\partial z} - \frac{1}{T} \frac{\partial T}{\partial z} \frac{\partial P}{\partial z} \right) + \frac{RT}{M} \frac{\dot{m}_{OG}}{V} \right]$$

$$\div \left[- \frac{P_0^k K_r}{\mu} f_0 - \frac{P_0^k K_z}{\mu} g_0 \right] \qquad \text{(E.18)}$$

Further detail on the discretization and notation of the derivatives in E.18 above can be seen in the 2-D Steady State Solution Section 3.2.7. In the subroutine, Equation E.18 is broken up into the ten numerator terms labeled RNUMER1 through RNUMER10, and the two denominator terms are labeled as RDENOM1 and RDENOM2. The pressure for each element is calculated as the sum of the RNUMER terms divided by the sum of the RDENOM terms.

Once the pressure values for each active element have been calculated for an iteration, the average L1 norm of the residual vector (**Ax-b**) is calculated. Finally, the subroutine calculates the *r*- and *z*- velocities from the equations below.

$$- \frac{\partial P}{\partial r} = \frac{\mu}{K_r} v_r \qquad \text{(E.19)}$$

$$-\frac{\partial P}{\partial z} = \frac{\mu}{K_z} v_z \qquad \text{(E.20)}$$

Subroutine *SS_Solver* variables:

Variable	Data	Units
PFLAG(el#)	Flag which tells if an element is active or dead (100% charred)	--
LIVEL(#)	Array which contains only active element numbers	--
ALIVE	Total number of active elements in FMAP file	--
ALIVEP	Total number of active interior elements and fictitious zero volume boundary elements	--
PRESS(el#)	Pressure for a given element, pressure from a previous iteration or time step is input into the routine	$\frac{lbf}{ft^2}$
ALPHAR/Z(el#)	Inverse of the permeability of the material in the r and z directions respectively	ft^{-2}
MU(el#)	Dynamic viscosity of the pyrolysis gas	$\frac{lbf-s}{ft^2}$
MWGT(el#)	Molecular weight of the pyrolysis gas	$\frac{lb}{lb-mol}$
TA(el#)	Temperature of an element. Corresponds to both the element and the gas in a cell, added new values	R
RUGC	Universal gas constant = 1545.12	$\frac{\frac{lbf}{ft^2} ft^3}{lbmol-R}$
KCON(face#,el#)	Neighboring element number, added new values	--

OGRATE(el#)	Out-Gassing rate for an element (mass production)	$\frac{lbm}{s}$
dPdz/dPdr	First pressure derivatives	$\frac{lbf}{ft^3}$
d2Pdz2/d2Pdr2	Second pressure derivatives	$\frac{lbf}{ft^4}$
d2Pdz21t8/d2Pdr21t8	Sum of pressures times 2nd derivative coefficient for the eight neighboring stencil elements only (no contribution from central element)	$\frac{lbf}{ft^4}$
d2Pdz29/d2Pdr29	Central element's pressure times its derivative coefficient	$\frac{lbf}{ft^4}$
dTdz/dTdr	First temperature derivatives	*R/ft*
dmpdz/dmpdr	First material property (1/(alpha * mu)) derivative	$\frac{ft^3}{lbf - s}$
RNUMER*i*	10 numerator terms in pressure calculation	$\frac{lbf}{ft^2}$
RDENOM*i*	2 denominator terms in pressure calculation	$\frac{lbf}{ft^2}$
PRESSURE_NEW	sum of RNUMER terms divided by RDENOM terms	$\frac{lbf}{ft^2}$
PRESSURE_OLD(el#)	pressure from previous iteration	$\frac{lbf}{ft^2}$
OMEGA	pressure relaxation factor	--
PRESSURE(el#)	Pressure for a given element, pressure solution is sent to the main program (relaxed value)	$\frac{lbf}{ft^2}$
RES_SUM	Sum of pressure residuals for all active elements	$\frac{lbf}{ft^2}$
ICOUNT	Number of active elements	

RES_AVG	RES_SUM divided by ICOUNT	$\frac{lbf}{ft^2}$
RVEL/ZVEL(el#)	r and z velocities	*ft/s*

SUBROUTINE *TR_SOLVER*

The transient pressure solver routine (TR_SOLVER) iterates Equation E.22 below, which is the discretized form of Equation E.21, until the average pressure residual reaches a limit set by the user.

$$\phi\frac{\partial P}{\partial t}-\phi\frac{P}{T}\frac{\partial T}{\partial t}-\frac{PK_r}{\mu}\left(\frac{\partial^2 P}{\partial r^2}+\frac{\mu}{K_r}\frac{\partial P}{\partial r}\frac{\partial}{\partial r}\left(\frac{K_r}{\mu}\right)+\frac{1}{P}\frac{\partial P}{\partial r}\frac{\partial P}{\partial r}-\frac{1}{T}\frac{\partial T}{\partial r}\frac{\partial P}{\partial r}+\frac{1}{r}\frac{\partial P}{\partial r}\right)$$

$$-\frac{PK_z}{\mu}\left(\frac{\partial^2 P}{\partial z^2}+\frac{\mu}{K_z}\frac{\partial P}{\partial z}\frac{\partial}{\partial z}\left(\frac{K_z}{\mu}\right)+\frac{1}{P}\frac{\partial P}{\partial z}\frac{\partial P}{\partial z}-\frac{1}{T}\frac{\partial T}{\partial z}\frac{\partial P}{\partial z}\right)=\frac{RT}{M}\frac{\dot{m}_{OG}}{V} \qquad \text{(E.21)}$$

$$P_0^{k+1}=\left[\phi P_0^k+\phi\frac{P_0^k}{T_i^{n+1}}\left(T_i^{n+1}-T_i^n\right)\right.$$

$$+\Delta t\frac{P_0^k K_r}{\mu}\left(\left[f_1P_1+\cdots+f_8P_8\right]+\frac{\mu}{K_r}\frac{\partial P}{\partial r}\frac{\partial}{\partial r}\left(\frac{K_r}{\mu}\right)+\frac{1}{P_0^k}\frac{\partial P}{\partial r}\frac{\partial P}{\partial r}-\frac{1}{T}\frac{\partial T}{\partial r}\frac{\partial P}{\partial r}+\frac{1}{r}\frac{\partial P}{\partial r}\right)$$

$$\left.+\Delta t\frac{P_0^k K_z}{\mu}\left(\left[g_1P_1+\cdots+g_8P_8\right]+\frac{\mu}{K_z}\frac{\partial P}{\partial z}\frac{\partial}{\partial z}\left(\frac{K_z}{\mu}\right)+\frac{1}{P_0^k}\frac{\partial P}{\partial z}\frac{\partial P}{\partial z}-\frac{1}{T}\frac{\partial T}{\partial z}\frac{\partial P}{\partial z}\right)+\Delta t\frac{RT}{M}\frac{\dot{m}_{OG}}{V}\right]$$

$$\div\left[\phi-\frac{P_0^k K_r}{\mu}f_0-\frac{P_0^k K_z}{\mu}g_0\right] \qquad \text{(E.22)}$$

The transient solver structure is the same as the steady state structure which is described in depth above. The only difference is the addition of the two transient terms from Equation E.22, which result in two additional RNUMER terms and one additional RDENOM term.

One important note for the transient solver is that the size of the time step can affect the stability of the problem. For the given FMAP files, the largest time step used by *PorePress* was 0.001 seconds. The code has not been used with any FMAP files with larger time steps. Thus, it is important for the time step size to be monitored closely for future uses if a larger step size is desired.

Subroutine *TR_Solver* variables:

Variable	Data	Units
PFLAG(el#)	Flag which tells if an element is active or dead (100% charred)	--
LIVEL(#)	Array which contains only active element numbers	--
ALIVE	Total number of active elements in FMAP file	--
ALIVEP	Total number of active interior elements and fictitious zero volume boundary elements	--
PREV_PRESS(el#)	Pressure for a given element, pressure from a previous time step is input into the routine	$\frac{lbf}{ft^2}$
DELTAT	Time step	s

ALPHAR/Z(el#)	Inverse of the permeability of the material in the r and z directions respectively	ft^{-2}
MU(el#)	Dynamic viscosity of the pyrolysis gas	$\frac{lbf-s}{ft^2}$
MWGT(el#)	Molecular weight of the pyrolysis gas	$\frac{lb}{lb-mol}$
TA(el#)	Temperature of an element. Corresponds to both the element and the gas in a cell, added new values	R
POROS(el#)	Porosity	--
RUGC	Universal gas constant = 1545.12	$\frac{\frac{lbf}{ft^2}\,ft^3}{lbmol-R}$
KCON(face#,el#)	Neighboring element number, added new values	--
OGRATE(el#)	Out-Gassing rate for an element (mass production)	$\frac{lbm}{s}$
dPdz/dPdr	First pressure derivatives	$\frac{lbf}{ft^3}$
d2Pdz2/d2Pdr2	Second pressure derivatives	$\frac{lbf}{ft^4}$
d2Pdz21t8/d2Pdr21t8	Sum of pressures times 2nd derivative coefficient for the eight neighboring stencil elements only (no contribution from central element)	$\frac{lbf}{ft^4}$
d2Pdz29/d2Pdr29	Central element's pressure times its derivative coefficient	$\frac{lbf}{ft^4}$
dTdz/dTdr	First temperature derivatives	*R/ft*
dmpdz/dmpdr	First material property (1/(alpha * mu)) derivative	$\frac{ft^3}{lbf-s}$

RNUMER*i*	10 numerator terms in pressure calculation	$\frac{lbf}{ft^2}$
RDENOM*i*	2 denominator terms in pressure calculation	$\frac{lbf}{ft^2}$
PRESSURE_NEW	sum of RNUMER terms divided by RDENOM terms	$\frac{lbf}{ft^2}$
PRESSURE_OLD(el#)	pressure from previous iteration	$\frac{lbf}{ft^2}$
OMEGA	pressure relaxation factor	--
PRESSURE(el#)	Pressure for a given element, pressure solution is sent to the main program (relaxed value)	$\frac{lbf}{ft^2}$
RES_SUM	Sum of pressure residuals for all active elements	$\frac{lbf}{ft^2}$
ICOUNT	Number of active elements	
RES_AVG	RES_SUM divided by ICOUNT	$\frac{lbf}{ft^2}$
RVEL/ZVEL(el#)	r and z velocities	*ft/s*

VITA

Keegan Delaney was born in Lynchburg, Va on August 27, 1981. He received his Bachelor's in Mechanical Engineering from Virginia Tech in May of 2005, and then joined the Masters program at Virginia Tech in August of 2005.