MODELING THREE-DIMENSIONAL DEFORMATION MECHANISMS IN CANDU REACTOR FUEL ELEMENTS

MODÉLISATION EN TROIS DIMENSIONS DES MÉCANISMES DE DÉFORMATION DES ÉLÉMENTS DU COMBUSTIBLE DES RÉACTEURS CANDU

A Thesis Submitted

to the Division of Graduate Studies of the Royal Military College of Canada

By

Kyle Allan Lawrence Gamble, B.Eng, EIT

In Partial Fulfillment of the Requirements for the Degree of

Master of Applied Science in Nuclear Engineering

May 2014

ROYAL MILITARY COLLEGE OF CANADA
COLLÈGE MILITAIRE ROYAL DU CANADA

DIVISION OF GRADUATE STUDIES AND RESEARCH
DIVISION DES ÉTUDES SUPÉRIEURES ET DE LA RECHERCHE

This is to certify that the thesis prepared by / Ceci certifie que la thèse rédigée par

KYLE ALLAN LAWRENCE GAMBLE

entitled / intitulée

MODELING THREE-DIMENSIONAL DEFORMATION MECHANISMS IN CANDU REACTOR FUEL ELEMENTS/ MODÉLISATION EN TROIS DIMENSIONS DES MÉCANISMES DE DÉFORMATION DES ÉLÉMENTS DU COMBUSTIBLE DES RÉACTEURS CANDU

complies with the Royal Military College of Canada regulations and that it meets the accepted standards of the Graduate School with respect to quality, and, in the case of a doctoral thesis, originality, / satisfait aux règlements du Collège militaire royal du Canada et qu'elle respecte les normes acceptées par la Faculté des études supérieures quant à la qualité et, dans le cas d'une thèse de doctorat, l'originalité,

for the degree of / pour le diplôme de

MASTER OF APPLIED SCIENCE IN NUCLEAR ENGINEERING / MAÎTRISE DU SCIENCE EN GÉNIE NUCLÉAIRE

Signed by the final examining committee: /
Signé par les membres du comité examinateur de la soutenance de thèse

Capt. Konstantyn Grygoryev, Chair / Président

Dr. Kelly Conlon, External Examiner / Examinateur externe

Dr. Paul K. Chan, Main Supervisor / Directeur de thèse principal

Approved by the Head of Department : /
Approuvé par le Directeur du Département : **Maj. Paul Hungler** Date: **May 20, 2014**

To the Librarian: This thesis is not to be regarded as classified. /
Au Bibliothécaire : Cette thèse n'est pas considérée comme à publication restreinte.

Dr. Paul K. Chan
Main Supervisor / Directeur de thèse principal

ACKNOWLEDGEMENTS

First and foremost I extend my appreciation to my supervisors Dr. Paul K. Chan and Dr. Anthony F. Williams for providing me the opportunity to pursue this research endeavour. Their guidance and knowledge were critical to ensure the success of this work. My thanks are also extended to the Multiphysics Object-Oriented Simulation Environment (MOOSE) developers at the Idaho National Laboratory (INL), in particular Derek Gaston and Cody Permann for providing me access to the computational framework and allowing me to use INL's computational resources. In addition, I am grateful to the BISON developers at INL, in particular Benjamin Spencer and Jason Hales who contributed greatly in the debugging of my model and invited me down to INL to present a seminar on this research. I graciously extend my thanks to my family. Without their support this endeavour would not have been possible.

Moreover I would like to thank Atomic Energy of Canada Limited (AECL) for providing me with the opportunity to be attached at the Chalk River Laboratories (CRL) for a full year to complete this research. I am appreciative of the guidance from my colleagues at AECL-CRL, in particular Markus Piro, Jeff Baschuk, and Brian Leitch. Figure 2.2 in this work is reproduced with permission from Brian. I also want to thank my fellow graduate students at RMC that made the first eight months of this program unforgettable.

I would like to acknowledge the Natural Sciences and Engineering Research Council of Canada (NSERC) under CRDPJ# 415247 for funding this work. Last but not least I would like to thank Dr. Hugues Bonin for taking the time to translate both the thesis title and abstract into French.

ABSTRACT

Gamble, Kyle Allan Lawrence, M.A.Sc. (Nuclear Engineering). Royal Military College of Canada, May 2014. Modeling Three-Dimensional Deformation Mechanisms in CANDU Reactor Fuel Elements. Thesis Supervisors: Dr. Paul K. Chan and Dr. Anthony F. Williams.

The horizontal orientation of nuclear fuel elements in CANDU reactors presents some unique challenges when it comes to modeling their performance and behaviour under reactor operating conditions. The elements can undergo two different three-dimensional deformation mechanisms during operation: bow and sag. Bowing is the deflection of the fuel elements due to temperature gradients developed across the diameter, and sagging is the deformation at high temperatures due to the self weight of the element. Quantifying these deflections will help determine if element-to-element or element-to-pressure tube contact may occur during reactor operation, possibly causing sheath failure or pressure tube rupture under certain accident conditions.

This thesis assesses the feasibility, for CANDU reactor fuel elements, of using the state-of-the-art MOOSE computational framework developed at the Idaho National Laboratory for modeling these three-dimensional deformation mechanisms. This work is the first use of the MOOSE framework in Canada. An application is developed upon the MOOSE framework called HORSE (HORizontal nuclear fuel Simulation Environment) for modeling CANDU fuel. Using HORSE, models are developed that examine the contact capabilities within the framework in two and three dimensions. In addition models are constructed that predict fuel performance, the flexural rigidity of an element, and the thermal bowing behaviour in three dimensions. Assessment and comparison of results against industry codes, numerical models, and analytical solutions are presented. The capabilities of the framework for predicting flexural rigidity and thermal bowing

are also discussed. These comparisons demonstrate the potential of the MOOSE framework for developing these complex models. Some additional development of the contact algorithms and inclusion of more nuclear phenomena is required to improve the accuracy for horizontal fuel.

Keywords: Finite Element Analysis, Multiphysics, Thermo-mechanics, Deformation, MOOSE, BISON

RÉSUMÉ

Gamble, Kyle Allan Lawrence, M. Sc. A. (Génie nucléaire). Collège militaire royal du Canada, mai 2014. Modélisation en Trois Dimensions des Mécanismes de Déformation des Éléments du Combustible des Réacteurs CANDU. Directeurs de thèse: Dr. Paul K. Chan et Dr. Anthony F. Williams.

L'orientation horizontale des éléments de combustible nucléaire dans les réacteurs CANDU implique certains défis uniques quant à la modélisation de leur performance et de leur comportement dans les conditions d'exploitation du réacteur. Les éléments sont soumis à deux différents mécanismes de déformation durant l'exploitation: le fléchissement et l'affaissement. Le fléchissement est la déviation des éléments de combustible due aux gradients de température qui se produisent le long du diamètre, et l'affaissement est la déformation à hautes températures causée par le poids même de l'élément. La quantification de ces déflexions devrait aider à déterminer si le contact élément-à-élément ou élément-à-tube de force peut se produire durant l'exploitation du réacteur, et devenir une cause possible de la défaillance des gaines de combustible ou de la rupture du tube de force selon certaines conditions d'accident.

Cette thèse évalue la possibilité d'utiliser pour les éléments de combustible du réacteur nucléaire CANDU le logiciel MOOSE développé au *Idaho National Laboratory* et présentement utilisé pour la modélisation en trois dimensions de ces mécanismes de déformation. Cette recherche est la première utilisation de la structure du logiciel MOOSE au Canada. On a développé une application dans la structure de MOOSE appelée HORSE ("*HORizontal nuclear fuel Simulation Environment*") pour la modélisation du combustible CANDU. À l'aide de HORSE, on a développé des modèles qui examinent les possibilités de contact dans la structure en deux et trois

dimensions. De plus, des modèles ont été construits pour prédire la performance du combustible, la rigidité d'un élément de combustible quant à la flexion, et le comportement en fléchissement thermique en trois dimensions. On présente l'évaluation et la comparaison des résultats de ce modèle avec ceux des codes industriels, des modèles numériques et de solutions analytiques. Les capacités de la structure pour la prédiction de la rigidité d'un élément de combustible quant à la flexion et du fléchissement thermique font aussi l'objet d'une discussion. Ces comparaisons démontrent le potentiel de la structure de MOOSE pour le développement de ces modèles complexes. Quelques additions au développement d'algorithmes pour traiter le contact et l'inclusion de phénomènes nucléaires additionnels sont nécessaires pour améliorer la précision des résultats pour l'élément de combustible horizontal.

Mots-clefs: Analyse par éléments finis, Multiphysique, Thermomécanique, Déformation, MOOSE, BISON

Table of Contents

List of Figures

List of Tables

List of Symbols and Acronyms

List of Acronyms

AECL	Atomic Energy of Canada Limited
AGR	Advanced Gas-cooled Reactor
AMP	Advanced MultiPhysics
BACO	BArra COmbustible
BC	Boundary Condition
BWR	Boiling Water Reactor
CANDU	CANada Deuterium Uranium
CANLUB	CANdu LUBricant
CNSC	Canadian Nuclear Safety Commission
CRL	Chalk River Laboratories
CTF	Curvature Transfer Factor
DOE	Department of Energy
dof	degrees of freedom
ECC	Emergency Core Coolant
ELESTRES	ELEment Simulation and sTRESses
ELK	Extended Library of Kernels
ELOCA	Element Loss of Coolant Analysis
EPRI	Electric Power Research Institute
FAST	Fuel and Sheath modelling Tool
FEA	Finite Element Analysis
FEM	Finite Element Method
FES	Fuel Element Simulator
FORCE	Fuel Operational peRformance Computations in an Element
GFP	Gaseous Fission Products
GMRES	Generalized Minimal RESidual
HORSE	HORizontal nuclear fuel Simulation Environment
HPC	High Performance Computing
csimsoft	Computational Simulation Software
INL	Idaho National Laboratory
IST	Industry Standard Toolset
ITU	European Institute for Transuranium Elements
JAEA	Japan Atomic Energy Agency
JAERI	Japan Atomic Energy Research Institute
JFNK	Jacobian Free Newton-Krylov
LBLOCA	Large Break Loss of Coolant Accident
LOCA	Loss of Coolant Accident
LWR	Light Water Reactor

MATPRO	MATerial PROperties library
MOOSE	Multiphysics Object-Oriented Simulation Environment
MOX	Mixed OXide
NNL	National Nuclear Laboratory
NEAMS	Nuclear Energy Advanced Modeling and Simulation
NOC	Normal Operating Conditions
PETSc	Portable Extensible Toolkit for Scientific Computation
PDE	Partial Differential Equation
PHTS	Primary Heat Transport System
PHWR	Pressurized Heavy Water Reactor
PIE	Post Irradiation Examination
PNNL	Pacific Northwest National Laboratory
PWR	Pressurized Water Reactor
REF	Rigidity Enhancement Factor
RMCC	Royal Military College of Canada
SCC	Stress Corrosion Cracking
SCWR	Super Critical Water-cooled Reactor
SFP	Solid Fission Products
TRISO	TRIstructural ISOtropic
USNRC	United States Nuclear Regulatory Commission

List of Symbols

A	Area; Constant in λ_0
a_0	Constant in h_s
B	Constant in λ_0
Bu	Fuel Burnup
b	Constant in ε
C	Constant in h_f; Constant in λ_0; Fuel Chamfer
C_p	Specific Heat Capacity
D	Constant in λ_0; Percent of Theoretical Density
d	Dish; Grain Size
E	Young's Modulus
E_d	Constant in α of UO_2; Constant in C_p of UO_2
EI	Flexural Rigidity
$\vec{F}$	Arbitrary Function
F	Fraction of Initial Porosity Removed; Creep Rate Coefficient
$\mathbf{f}$	Body Force per Unit Mass
$\mathbf{f}_c$	Contact Force
f	Forcing Function; Fraction of code that is Parallelizable
f_{mag}	Proportionality Coefficient

g	Gap Distance, Jump Distance, Gravity
H	Meyer Hardness
H_r	Hole Radius
h	Heat Transfer Coefficient
I_n	n^{th} Order Modified Bessel Function of the 1^{st} Kind
$\mathbf{J}$	Jacobian Matrix
K	Strength Coefficient
K_1	Constant in α of UO_2; Constant in C_p of UO_2
K_2	Constant in α of UO_2; Constant in C_p of UO_2
K_3	Constant in α of UO_2; Constant in C_p of UO_2
K_{diag}^{-1}	Diagonalized Stiffness Matrix of Slave Nodes
K_n	n^{th} Order Modified Bessel Function of the 2^{nd} Kind
k	Thermal Conductivity; Boltzmann's Constant
k_p	Penalty Factor
L	Fuel Stack Length; Length between Supports
M	Atomic Weight
M_u	Initial Mass of Uranium
N	Number of Unknowns
$\mathbf{n}$, $\vec{n}$	Normal Vector
n	Number of Moles
$\mathbf{P}$	Preconditioning Matrix
P	Pressure; Number of Processors; Applied Load at Element Midplane
P_{lin}	Linear Power Rating
P_r	Pellet Radius
p	Porosity
Q	Activation Energy divided by the Universal Gas Constant
$Q_{fission}$	Heat Generation due to Fission
$\mathbf{R}$	Residual Vector
R	Universal Gas Constant; Surface Roughness; RMS of Surface Roughnesses
$\mathbf{r}_0$	Initial Linear Residual
$\mathbf{r}_m$	Residual Vector of Master Surface
$\mathbf{r}_s$	Residual Vector of Slave Surface
r	Radial Location
S	Speedup
s	Slip Distance
T	Temperature
TM	Tangent Modulus
T_1	Wall-clock Time; Temperature at Top of Element for Thermal Bowing
T_2	Temperature at Bottom of Element for Thermal Bowing

t	Time
u	Field Variable
$\mathbf{u}$	Displacement Vector
u_h	Discretized Field Variable
$\mathbf{u}_n$	State Vector
V	Volume
$\mathbf{v}$	Krylov Vector
$\mathbf{w}$	Vector used in Preconditioning Calculation
$\vec{w}_{qp}$	Weighting Factor Associated with the Quadrature Point
$\mathbf{x}$	Slave Node Position
$\mathbf{x}_p$	Projected Location of the Slave Node on the Master Surface
$\mathbf{x}_{p_{t-1}}$	Projected Location of the Slave Node from Previous Timestep
$\vec{x}_{qp}$	Position of the Quadrature Point
y	Mole Fraction; Vertical Deflection at Element Midplane

List of Symbols (Greek)

α	Coefficient of Thermal Expansion
β	Fuel Burnup in Atom Percent
β_{flux}	Flux Depression Parameter in $Q_{fission}$
ε	Small Perturbation; Emissivity; Strain
$\dot{\varepsilon}_{gb}$	Creep Rate due to Grain Boundary Sliding (Diffusional Creep)
θ	Constant in C_p of UO_2
κ	Thermal Conductivity Contributions
κ_{flux}	Flux Depression Parameter in $Q_{fission}$
λ	Lamé's First Parameter
λ_0	Unirradiated Thermal Conductivity of UO_2
λ_{flux}	Flux Depression Parameter in $Q_{fission}$
μ	Coefficient of Friction; Lamé's Second Parameter
ν	Poisson's Ratio
ρ	Density
$\boldsymbol{\sigma}$	Cauchy's Stress Tensor
σ	Stefan-Boltzmann Constant; Stress
σ_p	Pore Shape Factor
ϕ	Shape Function
ψ	Test Function
Ω	Domain of Interest

List of Superscripts and Subscripts

0	Initial; At 273 K

$1d$	Dissolved Fission Products
$1p$	Precipitated Fission Products
$2p$	Pores and Fission Gas Bubbles
$3x$	Deviation from Stoichiometry
$4r$	Radiation Damage
a	Axial
$axial$	Axial Stress
c	Critical; Coolant
$copy$	Copy of Value from Previous Timestep
$creep$	Due to Creep
d	Depth
$dens$	Densification
el	Elastic
f	Fuel
GFP	Due to Gaseous Fission Products
g	Gap
gas	Fill Gas and or Fission Gas in Gap
$hoop$	Hoop Stress
i	Column Index in a Matrix; Node Number; Shape Function Number
j	Sum Index; Row Index in a Matrix; Strain Component of Interest
k	Krylov Iteration
m	Strain Rate Sensitivity Exponent
N	Normal Direction
n	Iteration Counter; Strain Hardening Exponent
qp	Quadrature Point
R	Reduced
r	Radial; Radiative Conductance
$radial$	Radial Stress
SFP	Due to Solid Fission Products
s	Sheath; Solid-to-solid
$surf$	Surface
T	Tangential Direction
th	Theoretical; Thermal Expansion
U	Uranium
UO_2	Uranium Dioxide
UT	Ultimate Tensile
VM	Von Mises
w	Width
y	Yielding

Chapter 1 - Introduction

Nuclear energy is a safe and reliable source of electricity that produces zero greenhouse gases. With the ever increasing concerns in regards to fossil fuels and climate change, a clean reliable alternative is required. This need for emission free energy has brought nuclear energy back to the forefront of many governmental agendas. Worldwide, approximately 160 nuclear reactors have been planned with a net generating capacity of 177 000 MWe, with an additional 320 reactors being proposed [1]. These new builds will significantly decrease the dependence upon fossil fuels such as natural gas, coal and oil, thereby reducing greenhouse gas emissions.

In Ontario, nuclear energy provides a reliable source of base load electricity. Almost fifty percent of all electricity in the province is generated by the nuclear reactors at the Bruce, Darlington and Pickering power stations. The reactors at these power plants are of the CANada Deuterium Uranium (CANDU) variety [2].

A CANDU reactor core contains between 380 and 480 fuel channels and within these channels is a pressure tube containing 12-13 half-metre long bundles. A schematic of a CANDU reactor core, including the primary heat transport system (PHTS) and fuelling machines is presented in Figure 1.1. The horizontal orientation of the reactor is to facilitate online refuelling for continuous operation of the reactor. In contrast, the fuel in light water reactors (LWR) are arranged vertically, utilize enriched instead of natural uranium and must be powered down to refuel. The fuel channels of the reactor are housed in a low-pressure vessel called the calandria, which contains the heavy water neutron moderator. The use of heavy water produces greater neutron efficiency than light water reactor systems. Heavy water is also used as the primary

coolant, which flows through the pressure tubes and sub-channels of the fuel bundles to remove heat. The heated water then flows to the steam generators to produce steam in a separate supply of water. This steam is then used to turn a turbine to generate electricity, similar to a conventional fossil fuel power plant. By examining Figure 1.1 it can be seen that two steam generators and fuelling machines are used. This is because fresh bundles are pushed into the reactor from one of the fuelling machines, and ejecting older bundles into the fuelling machine on the other side. Neighbouring fuel channels are cooled from and refuelled from opposite directions (checkerboard pattern). This, combined with the online refuelling provides operators with increased control over the flux profile and reactor environment compared to batch refuelled reactors.

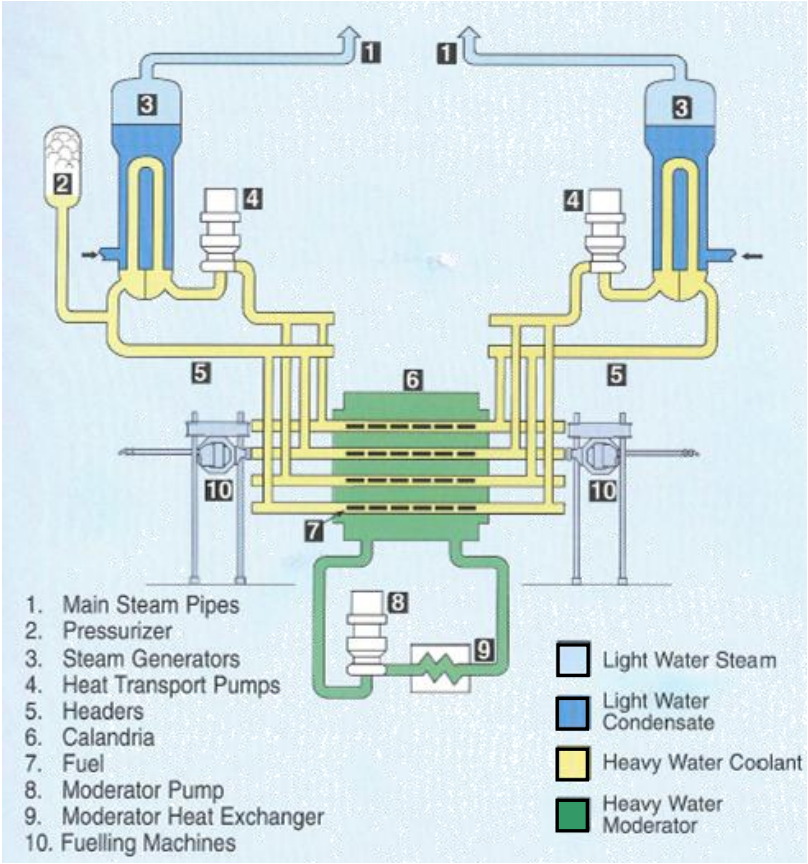


Figure 1.1: Schematic of a CANDU reactor core including the primary heat transport system and fuelling machines. Reproduced from **[3]**.

Each bundle is composed of either 28 or 37 fuel elements (depending on the type of CANDU reactor) assembled in concentric rings. The outside of the element, called the sheath, separates the fuel from the coolant and provides structural support. Typically the sheaths are made of a zirconium alloy (Zircaloy-4), that has a nominal thickness of 0.4 mm. The use of heavy water as the moderator and coolant allows the use of natural uranium containing 0.72 weight percent uranium-235 in the form of sintered uranium dioxide (UO_2) pellets contained within the sheath. Each fuel element within a bundle contains approximately 30 pellets. Pellets are nominally 16 mm in length and 6 mm in radius, although there is some variability between fuel designs. The small, approximately 45 μm gap between the pellets and inner surface of the sheath is initially filled with inert helium gas. To mitigate stress corrosion cracking (SCC) of the sheath due to fission product iodine, a unique graphite coating called CANLUB (CANdu LUBricant) is applied to the inside surface of the sheath during manufacturing. A schematic containing an end view representation of the inside of a pressure tube and an isometric view of a 37-element fuel bundle is shown in Figure 1.2. The inter element spacers and end plates provide the bundle with some structural rigidity and keeps the bundle intact. The bearing pads on the outer ring of elements prevent the bundle from coming in contact with the pressure tube. During some extreme transients in which coolant flow is constricted between the outer elements and the pressure tube, dry out will occur and element to pressure tube contact is possible.

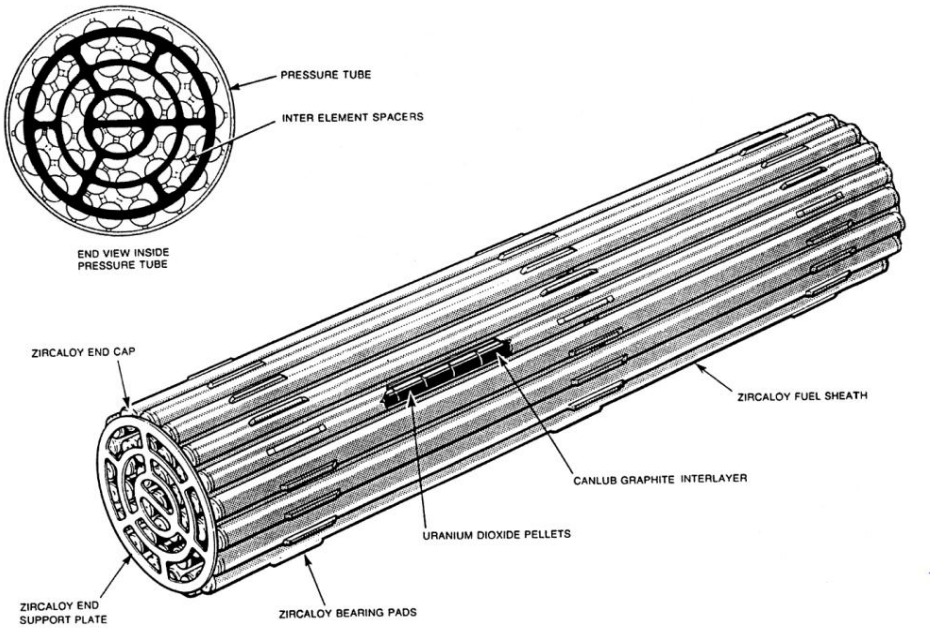


Figure 1.2: Schematic view of a CANDU 37-element fuel bundle. Reproduced from **[4]**.

Under normal operating conditions (NOC) the nuclear fuel bundles are within a very complex and ever changing environment. The neutron flux within the reactor core causes fission within the fuel generating heat and large temperature gradients. The fuel sheath also experiences forces due to hydraulic drag and the high pressures within the core. The fission process generates fission products, including krypton and xenon gases that mix with the helium in the vacant region of the element which degrade the heat transfer across the gap. Permanent (plastic) deformation of the element occurs after long irradiation times. Therefore the fuel and sheathing materials have constantly changing physical and chemical properties. Since the fuel is the first barrier to the release of radioactive material, the ability to understand and predict this complex behaviour is essential for safe operation.

There are two methods of gaining an understanding of the phenomena occurring in the reactor environment, experiments and numerical computer models. Experiments are useful because they measure directly the evolution of a particular quantity of interest under specific conditions. However, irradiated fuel experiments are expensive, take a long time to complete, and do not provide predictive capabilities. Contrarily, computer models can provide an in-depth look into the behaviour of the fuel as well as predict what will occur in the future. Computer models do require some assumptions to be made to ensure convergence of the simulations. Models and codes to be used for safety analyses must be validated against existing experimental data. Once validated these models can only be used within the domain of validation meaning the code cannot be extrapolated outside of the boundaries for which it was validated. Moreover, if additional phenomena, material properties or different fuel types are added to the code, it must be revalidated before being used for safety analysis. Models can also be developed solely for research and development purposes, with much lower validation requirements. These models can be benchmarked against existing experiments, validated codes, or existing research models as available. Research models can be used to investigate different fuel types, or to examine phenomena not captured in the currently validated codes (e.g., 3D deformation). The rules regarding extrapolation out of the validated domain and revalidation when additional phenomena are added are less rigorous for purely research and development models. Simulations from research models benchmarked against existing experiments may be used to provide insight for experimentalists to assist in determining which trials should be run in an experimental environment to reduce the cost of running all trials of the experiment. Current computer models and codes have their strengths and limitations in what they can predict and model. This work is concerned with developing a thermo-mechanical fuel model using a new multiphysics platform

developed at the Idaho National Laboratory (INL) entitled MOOSE (Multiphysics Object Oriented Simulation Environment) to predict 3D fuel performance and to investigate fuel element deformation at high temperatures.

Chapters 2 and 3 outline the current status of industry nuclear fuel performance and deformation codes, nuclear models developed using commercially available multiphysics suites, and the goals of the current work. Chapter 4 goes into detail on how the MOOSE platform works, and its unique features. Chapter 5 provides details about the physical phenomena and material properties incorporated into the model developed here called HORSE (HORizontal nuclear fuel Simulation Environment. Chapter 6 investigates the key components required for each of the analyses completed in this work. Chapter 7 provides a comparison against existing codes, models and analytical solutions of the results obtained during this study. Chapters 8 and 9 provides a summary and concluding remarks on the significance of this thesis work, as well as recommendations to expand this work into a more detailed and fully mechanistic model.

Chapter 2 - State of the Art

Nuclear fuel modeling is an extensive field of study with a wide variety of researchers, engineers and graduate students developing and perfecting models as specific as a correlation for fission gas release to beam models of full 37-element fuel bundles. The nuclear fuel environment is a complex and fully coupled system. Structural mechanics influences thermal behaviour, and thermal behaviour affects structural mechanics. The neutron environment has an effect on the mechanical, thermal, and chemical properties of the fuel and sheath. There is a strong cyclic coupling between the fuel and sheathing materials with all other phenomena occurring in a fuel element. Figure 2.1 shows the mutual interaction between many of the material properties and nuclear phenomena occurring within a fuel element.

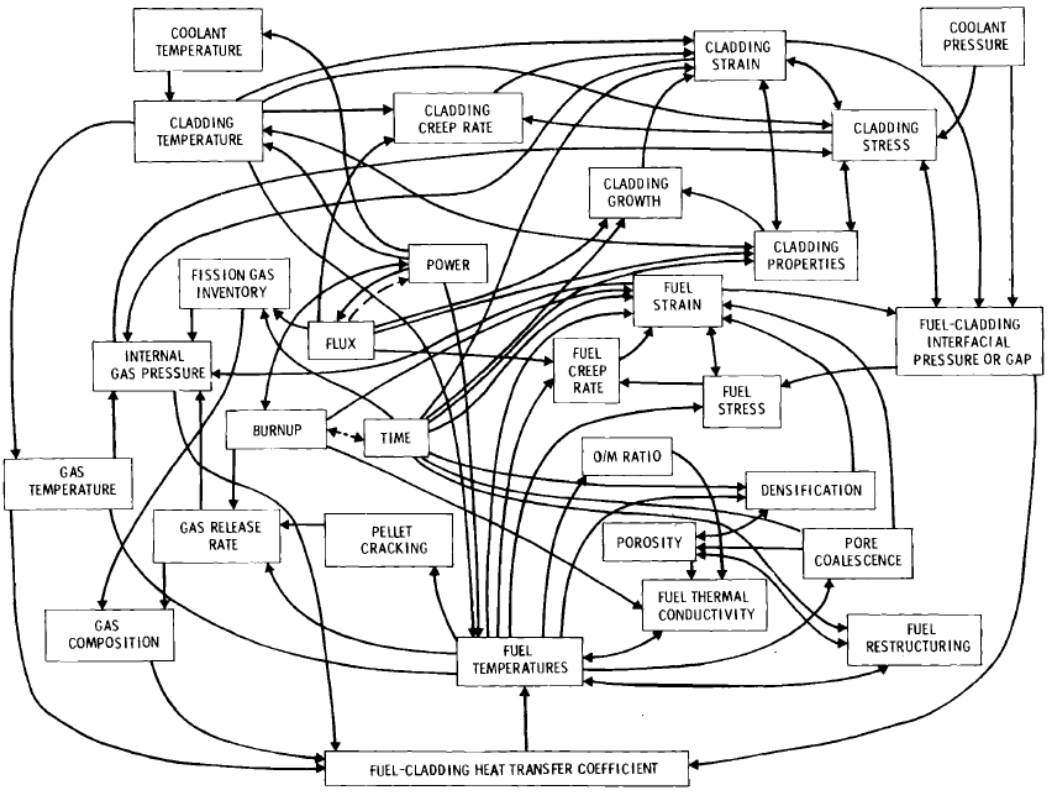


Figure 2.1: Fuel element parameters and how they influence one another. Cladding is an interchangeable name for sheath. Reproduced from **[5]**.

In thermal mechanical analyses, one of the most important parameters is the fuel-to-sheath heat transfer coefficient. As illustrated in Figure 2.1 it can be seen that this parameter is directly influenced by four parameters: 1) the gas temperature within the gap region, 2) the composition of the gas mixture, 3) the pressure of the gas inside the gap and 4) the interfacial pressure between the sheath and fuel if the gap is closed or the gap length if the gap is open. By further examination of Figure 2.1 it can be seen that the interfacial pressure or gap width, and gas temperature, composition, and pressure themselves depend on a variety of material properties and phenomena. Since the heat transfer coefficient of the gap determines the heat flux from the fuel to the sheath and subsequently to the coolant, accurate modeling of the fuel-to-sheath gap heat transfer coefficient is essential for realistic fuel performance results. The multiphysics phenomena in the reactor environment occur over distances of inter-atomic spacing to meters, and over times ranging from nanoseconds to years [6]. This large variation in space and time scales adds to the complexity and need for an all-encompassing fuel performance and 3D deformation model.

There are a variety of codes that have been developed for nuclear fuel modeling by industry, research laboratories and graduate students. Of interest for this work are fuel performance codes, and element or bundle deformation models. The details of many existing codes and models are presented in subsections 2.1 and 2.2. These subsections are further divided into industry codes and multiphysics models.

2.1 Fuel Performance Codes

Fuel performance codes provide information about the fuel behaviour under normal operating and or transient conditions. These types of codes tend to encompass a broad range of the phenomena present in Figure 2.1. The geometric dimension of fuel performance codes are typically quasi-two-dimensional[1] (1.5D) or two-dimensional.

2.1.1 Industry Codes

National laboratories in many countries have developed their own fuel performance codes that are used for licensing of reactors by their respective nuclear regulators. Industry partners within these countries make use of the codes for their own analysis purposes. A brief overview of the most well-known fuel performance codes is presented.

The first fuel performance code for CANDU fuel was developed by the Atomic Energy of Canada Chalk River Laboratories (AECL-CRL) called ELESIM. The ELESIM code models the midplane of a pellet in a one dimensional axisymmetric manner for normal operating conditions. The subroutines are written in the Fortran programming language, and are empirically rather than mechanistically based. Models are included for fuel-to-sheath heat transfer, fuel thermomechanics, and fission product release. The constituent equations are solved iteratively at predetermined points during the power history using a finite difference method [7].

Building upon the ELESIM framework, the ELEment Simulation and sTRESes (ELESTRES) code was developed. ELESTRES replaces the one-dimensional stress-independent fuel

[1] A geometry is divided into axial and or radial slices that are analyzed individually before being coupled together.

expansion model of ELESIM with a two-dimensional axisymmetric finite element stress analysis of the fuel pellet. The thermal component of the analysis is still solved using a simple finite difference method. ELESTRES is limited to normal operating conditions and therefore an additional code for transient analyses needed to be developed [8]. This code, called Element Loss of Coolant Analysis (ELOCA), uses the data for the physical condition of the fuel at the start of the transient from ELESIM or ELESTRES [9]. Both ELESTRES and ELOCA have been classified as Industry Standard Toolsets (IST), validated against experiments and used to design, build and license reactors in Canada.

An analogous pair of codes developed by the Pacific Northwest National Laboratory (PNNL) for the United States Nuclear Regulatory Commission (USNRC) is FRAPCON and FRAPTRAN. The latest version of the normal operating conditions code FRAPCON-3 models the steady-state behaviour of a single fuel rod in a Pressurized Water Reactor (PWR) or Boiling Water Reactor (BWR) assembly. The material properties modeled include fuel thermal conductivity and expansion, clad thermal conductivity and expansion, fission gas release, clad corrosion, and cladding plasticity among others [10]. Recently, the material models have been updated to account for behavioural changes at extended burnups [11]. FRAPTRAN is used for transient and accident simulations. Both codes are written in Fortran.

A code developed by the European Institute for Transuranium Elements (ITU) called TRANSURANUS is used for the thermal and mechanical modeling of nuclear fuel rods. TRANSURANUS is the most well-known and comprehensive fuel performance code used in Europe. The code is capable of handling different fuel rod designs and can model normal, off-

normal and accident conditions over timescales ranging from milliseconds to years. To avoid large computational times with modeling a full fuel rod, a quasi-two-dimensional model is used. Using this model, the fuel rod is divided into axial slices that are analyzed individually for each timestep using a finite difference technique. Once each slice has been analyzed they are coupled together. Comprehensive models are available for a variety of fuel materials, Zircaloy and steel cladding, as well as a range of different coolants. TRANSURANUS is written in Fortran 77 and is available for use by anyone who is interested [12].

In the United Kingdom, a standalone fuel performance code called ENIGMA owned by the National Nuclear Laboratory (NNL) is used. This code written in Fortran is capable of calculating the thermo-mechanical behaviour of a LWR or advanced gas-cooled reactor (AGR) fuel rod. Different fuel and cladding types can be modeled. ENIGMA uses a quasi-two-dimensional fuel rod representation by dividing the rod into a series of axial zones which are further divided into equal thickness radial annuli. Detailed models are implemented for the fuel pellets, cladding materials and coolant subchannels. ENIGMA has recently been extended to perform a whole core and dry storage analyses. Therefore, ENIGMA can now perform fuel performance calculations during irradiation, pond cooling, drying, and dry storage [13].

Argentina, a country with a CANDU reactor at their Embalse nuclear facility, has developed a thermo-mechanical fuel performance code. The BArra COmbustible (BACO) code was originally designed to model non CANDU Pressurized Heavy Water Reactor (PHWR) fuel rods under irradiation. However, the code structure can be used to model almost any type of cylindrical fuel rod. The BACO code is written in Fortran 77 and contains material models for

the UO_2 fuel and Zircaloy cladding. Cylindrical symmetry is assumed and a finite difference scheme is used to solve the system of equations. It is believed that the BACO code was the first code that contained a quasi-two-dimensional treatment of fuel cracking [14].

The last fuel performance code to be discussed was developed by the Japanese Atomic Energy Research Institute (JAERI) and is currently owned by the Japan Atomic Energy Agency (JAEA) for the modeling of light water reactors, called FEMAXI. FEMAXI is capable of modeling both NOC and transient conditions. In the code a single fuel rod and the surrounding coolant are modeled. The solution is determined using a coupled one-dimensional thermal analysis and a two-dimensional finite element analysis. An axisymmetric representation is used and models include thermal and mechanical properties of the fuel and cladding, fission gas release, and gap conductance. FEMAXI has been used for experimental interpretation, safety analyses and model verification [15].

The codes presented in this section are just a small portion of the industry codes in use worldwide. The list is vast and includes other codes that are notable exceptions to the rule of separate codes for steady-state and transients, including the 2D FALCON code developed by Electric Power Research Institute (EPRI) [16], and the 3D codes developed in France TOUTATIS [17] and ALCYONE [18]. All of the codes explained in this section are used for licensing of reactors in their respective countries but contain approximations that do not capture all of the phenomena occurring in the fuel element or rod. The use of a 1.5D or 2D approximation greatly simplifies the model but comes at a risk of not capturing axial behaviour along the fuel rod, or local affects due to appendages or deformation due to high temperatures

and large temperature gradients (i.e., bow and sag). In addition, the development of the codes are focused on one particular geometry and do not allow for different fuel geometries such as annular, plate or spherical.

2.1.2 Multiphysics Models

Multiphysics software allows for the solving of complex fully coupled partial differential equations for a wide variety of engineering and science disciplines. For nuclear engineering, multiphysics software enables the coupled modeling of heat transfer, solid mechanics, diffusion, fission gas production and release, arbitrary forms of material properties, and fluid dynamics. Although industry toolsets have captured most of the key phenomena, a major difference in the multiphysics approach is allowing the user the customization of the geometry, material properties, and physics to be included in the simulation. In addition, the outputs of multiphysics programs allow for very detailed data analysis for the presentation of results. Moreover, multiphysics programs allow the geometries to be increased to two and three dimensions which allow investigation into phenomena that could not be observed using the quasi-2D or 1D codes. There have been many multiphysics models developed for nuclear fuel performance in LWR fuel rods, and CANDU reactor fuel elements and bundles. Many of these models have been developed at national labs throughout the United States, AECL-CRL, and the Royal Military College of Canada (RMCC).

One commercial finite element environment used for multiphysics applications in the nuclear industry is Comsol Multiphysics™. To use Comsol one must purchase a variety of modules for the physics that needs to be included into the model including heat transfer and solid mechanics, among others. Each module has associated annual maintenance fees, which can become costly

for commercial enterprises. The license and maintenance fees are less for academic institutions. There is also no access to the source code by the users. One of the first attempts to apply Comsol's capabilities to modeling CANDU fuel was done by Morgan [19]. This model was a 2D representation of half of a fuel pellet with azimuthal symmetry, and included capabilities of modeling the thermal response of the fuel, fuel thermal expansion, fuel densification and swelling, fission gas release, and elastic and plastic deformation of the fuel and sheath under normal operating conditions. The 2D representation of the fuel and cladding allows for the modeling of axial heat transfer variations within the fuel that cannot be captured in the ELESTRES and ELOCA codes. The Morgan model produced results of fuel temperature, fission gas pressure and release, interfacial pressure, and sheath strain that are in good agreement with simulations completed using the ELESTRES code.

Building upon the work done by Morgan, Prudil [20] developed the Fuel and Sheath modeling Tool (FAST) in Comsol for analysis of CANDU fuel under normal operating and transient conditions. FAST contains many nuclear phenomena important to fuel performance including heat generation and transport, elastic strain, pellet relocation, contact, grain growth, among others. The differential equations are solved on a 2D axisymmetric representation of a fuel element and sheath. The results obtained by Prudil were in good agreement with experimental data provided by AECL Chalk River Laboratories and the ELESTRES and ELESIM codes [21]. In addition to the commercial software Comsol, two important Department of Energy (DOE) code development endeavours have applications in nuclear engineering and fuel performance modeling: Advanced MultiPhysics (AMP), and BISON. BISON is a nuclear fuel performance code for a variety of fuel types including LWR fuel rods, TRIstructural ISOtropic (TRISO) fuel

particles, and plate and annular fuels built upon the MOOSE framework developed at the Idaho National Laboratory (INL). MOOSE and AMP were competing codes for advanced multiphysics modeling both funded by the Nuclear Energy Advanced Modeling and Simulation (NEAMS) program. AMP was a three-dimensional finite element multiphysics tool for modeling the behaviour of nuclear fuel during normal operation and anticipated operational transients of LWR fuel rods. AMP did not contain some important phenomena including chemistry, mechanical contact and fracture, neutronics and mesoscale physics [22]. Ultimately, the NEAMS program decided to cancel the funding to the AMP project and solely fund the MOOSE endeavour. The details of the MOOSE framework and its associated applications, including BISON, are provided in Chapter 4 as it is the code used to develop the models analyzed in this work. The exploration of 3D capabilities should allow for more accurate fuel performance modeling and will allow for the quantification of fuel element and bundle deformation as presented in Section 2.2.

Table 2.1 summarizes the industry and multiphysics nuclear fuel performance codes discussed previously, including the type of fuel that can be modeled, the dimensionality and modeling capabilities (normal operating and or transient conditions) of the code, the numerical method used, and the primary method of implementation of the code. This table is similar to that of Shaheen [23], with some omissions, additions, and modifications.

Table 2.1: Comparison of industry and multiphysics nuclear fuel performance codes

Code	Fuel Type	Dimensionality	Modeling Capabilities	Numerical Method	Implementation
ELESIM	CANDU	1D	NOC	Finite Difference	FORTRAN
ELESTRES	CANDU	1.5D	NOC	Mixed Finite Difference/ Finite Element	FORTRAN
ELOCA	CANDU	1.5D	Transients	Finite Difference	FORTRAN
FRAPCON	LWR	1.5D	NOC	Mixed Finite Difference/ Finite Element	FORTRAN
FRAPTRAN	LWR	1.5D	Transients	Finite Difference	FORTRAN
TRANSURANUS	LWR	1.5D	NOC, transients	Finite Difference	FORTRAN
ENIGMA	LWR AGR	1.5D	NOC, transients	Finite Difference	FORTRAN
BACO	PHWR	2D	NOC, transients	Finite Difference	Standalone Code
FEMAXI	LWR BWR	2D	NOC, transients	Finite Element	Standalone Code
Morgan Model	CANDU	2D	NOC	Finite Element	COMSOL Multiphysics
Prudil Model	CANDU	2D	NOC, transients	Finite Element	COMSOL Multiphysics
AMP	LWR	3D	NOC, transients	Finite Element	Standalone code
BISON	LWR TRISO	1D 2D 3D	NOC, transients	Finite Element	MOOSE (Object-Oriented C++)

2.2 Fuel Element and Bundle Deformation Models

The horizontal nature of CANDU reactors generate some interesting phenomena that needs to be investigated to ensure safe operation. Fuel elements in a fuel bundle can mechanically deform in two ways: bow and sag. Bowing occurs due to temperature gradients across the fuel element (e.g., dryout) which can lead to deformation in either the upwards or downwards directions. Bowing can occur very rapidly during transients as the temperature gradients can change drastically in a short period of time. Sagging is the deformation of the fuel element due to the weight of pellets and sheath under gravity. The sagging rate is accelerated at high temperatures as the sheath undergoes plastic deformation, including creep. Determining the amount of deformation of outer ring elements in a fuel bundle is of great interest to the CANDU industry because element-to-pressure tube contact may increase heat transfer to the pressure tube and restrict coolant flow, which could lead to pressure tube rupture, resulting in a Loss of Coolant Accident (LOCA). Deformation of a fuel bundle is also of interest as it can lead to subchannel constrictions and flow bypass resulting in reduced cooling and higher fuel temperatures. In this case Emergency Core Coolant (ECC) injection would be less efficient. Figure 2.2 illustrates the possibility of pressure tube rupture and flow bypass under hypothetical accident conditions.

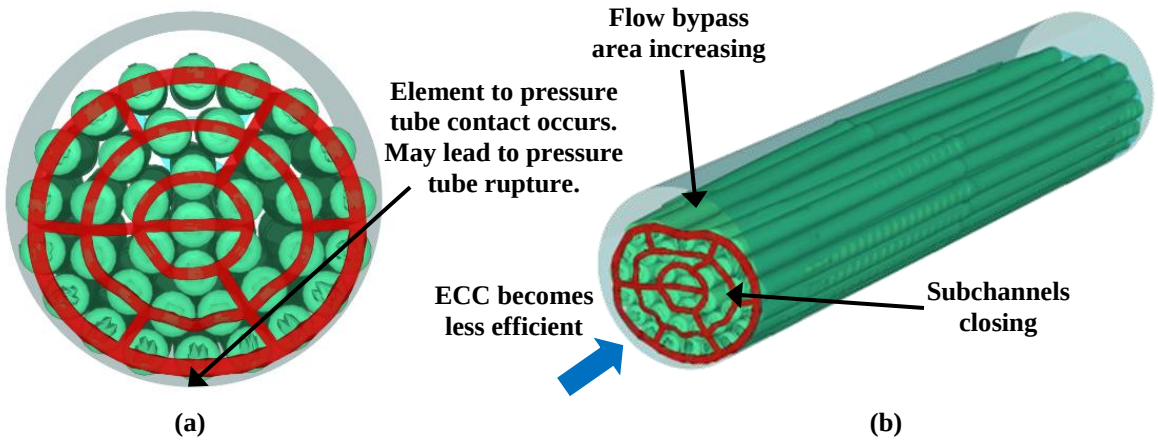


Figure 2.2: Illustrations of a 37-element bundle deformation leading to (a) potential pressure tube rupture and (b) increased flow bypass area due to subchannel closure causing reduced ECC efficiency under hypothetical accident conditions.

There are codes and models that attempt to predict the thermo-mechanical deformation of fuel elements and bundles in CANDU reactors. The capabilities of one industry code and three multiphysics models are presented in the subsequent subsections.

2.2.1 Industry Codes

The IST code used for nuclear fuel element and bundle bowing for CANDU fuel was developed at AECL Sheridan Park (now CANDU Energy Inc.), and is called the BOW code. In the BOW code the fuel elements are represented as composite beams. The BOW code considers many physical phenomena including: circumferential and axial variations in the temperature, pellet cracking, grip and slip between the pellets and the sheath, hydraulic drag due to the coolant, restraints from the endplates, nearby elements, and the pressure tube, gravity, neutron flux gradients, and temperature dependent material properties of the pellet and the sheath. The code is based upon the fundamental principles of solid mechanics. BOW represents each element by one-dimensional beam finite elements that can deform in three-dimensional space. These beam

elements can be assembled for modeling a full CANDU bundle in three-dimensions with end plates. The code is validated against known analytical solutions and post-irradiation examinations (PIEs) [24].

The results generated via the BOW code are accurate to closed-formed solutions and a few experiments, but there are some issues associated with the user input into the BOW code. To account for the curvature of the individual pellets within the sheath a Curvature Transfer Factor (CTF) is introduced, which describes the extent to which a pellet can transfer its curvature to the sheath. The CTF varies between zero, for no transfer of the pellet curvature to the sheath, and one for full transfer of the pellet curvature to the sheath. To account for the contribution of the pellets resistance to bending of the sheath another factor varying between zero and one, called the Rigidity Enhancement Factor (REF), is introduced. A value of zero means the pellets provide zero resistance to bending whereas a value of one indicates the pellets provide a large amount of resistance to bending [24]. Since the CTF and REF are inputs into the BOW code, the user can tune these values until they match the experiment of interest. This provides a very accurate code for matching existing experiments, but may cause difficulties for predicting element behaviour a priori. Thus, more mechanistic models are required that can predict the bowing of fuel elements without the help of tuneable factors.

2.2.2 Multiphysics Models

Many models have been developed for fuel element or fuel bundle deformation using commercial software platforms including ANSYS, Abaqus and Comsol. The ANSYS suite of software provides capabilities for multiphysics coupling between solid mechanics, heat transfer, fluid dynamics, and electromagnetic analyses using the finite element method. Abaqus is

another finite element based code that has the ability to couple structural, thermal, electrical, piezoelectric, pore pressure, acoustics and fluid dynamics [25]. Even though each commercial multiphysics package has similar capabilities, each one is more sophisticated and robust at modeling certain physics than others. For example, ANSYS and Abaqus have more robust mechanical contact algorithms then Comsol, whereas Comsol has additional multiphysics capabilities that are not available in the other software packages, including the ability to implement custom partial differential equations.

A model developed using Comsol for CANDU bundle deformation analyses by Bell [26] couples a 2D simulation of the heat conduction over a cross section of a fuel pin (including the pellets and sheath) to the beam element representation of the solid mechanics and deformation. Similarly to the BOW code, the beam elements are one-dimensional in nature but can deform and displace in three spatial dimensions. Beam elements were also used for the webbing of the endplates which provided a boundary condition for the fuel elements by assuming the endcap-to-endplate weld to be a continuous solid. The model contains temperature dependent material properties of the fuel and sheath, heat generation taking into account radial flux depression, and burnup. The code still requires a 'gripping factor' (like the CTF in BOW) to transfer the deformation behaviour of the pellets to the sheath, and does not take into account element-to-element or element-to-pressure tube contact. The bundle model was benchmarked against the same analytical formulations as BOW, an out-reactor fuel bundle deformation experiment and a dryout patch simulation completed with the BOW code, all with satisfactory results.

In order to remove the tuneable factors 3D simulations are required which account for the pellet-to-pellet and pellet-to-sheath contact. Two attempts have been made to model CANDU fuel element deformation in three-dimensions: a model by Williams [27] using ANSYS Mechanical and a model by McCluskey [28] using Abaqus/CAE. In ANSYS Mechanical, thermal and solid mechanics processes can be coupled. The purpose of the Williams model is to model fuel element bowing under temperature gradients due to off-normal conditions such as dryout and fuel element sagging under elevated temperatures during accidents such as a Large Break Loss of Coolant Accident (LBLOCA). These analyses can help determine whether cooling is sufficient after bundle subchannel closure during the transient, in addition to the thermal and mechanical loading on the pressure tube introduced by element-to-pressure tube contact. Williams' model uses two planes of symmetry to model a quarter of a full 31 pellet fuel element in three-dimensions using hexahedral finite elements. The model contains algorithms for mechanical and thermal contact between each individual pellet and the exterior surface of the pellets and the interior surface of the sheath. The model contains temperature dependent material properties of the fuel and sheath, sheath creep, uniform volumetric heat generation, coolant temperature, pressure, and heat transfer coefficient, and internal gas pressure. The initial version of the model did not contain an initial diametrical gap between the fuel and the sheath. Recent discussions with Williams have indicated that the latest version of the model has rectified the oscillation issues associated with the initial gap. The model was compared against out of reactor experiments within a fuel element simulator (FES) with excellent agreement [27].

Although the ANSYS model was able to accurately predict the sagging of fuel element simulator experiments, convergence issues were encounter during cooling cycles as rigid body motion of

the pellets would occur when the sheath had detached from the fuel pellets. In order to try and rectify this issue McCluskey examined the feasibility of using Abaqus to model a full fuel element to examine the contact capabilities. McCluskey's model included temperature dependent material properties and a heat generation term taking into account radial flux depression. However, the model did not include sheath creep which was determined to be an integral factor in predicting the correct amount of sag by Williams. McCluskey's model contained 11 fuel pellets assuming half symmetry along the axial direction and the linear powers that could be simulated ranged from 20-30 kW m^{-1}. At powers above 30 kW m^{-1} convergence issues were encountered. These issues may be related to the axial constraints placed upon the pellets. The model did produce acceptable results when comparing fuel performance parameters, such as sheath surface temperatures and fuel centerline temperatures, and diametrical gap sizes against ELESTRES simulations for fresh fuel. Excellent results were obtained when comparing the deflection of the element under simple beam loading against analytical calculations. An attempt to quantify the value of REF in the BOW code as a function of the linear element rating was also reported by McCluskey [28]. Table 2.1 summarizes the different fuel element and bundle deformation models described above.

Table 2.2: Summary of CANDU fuel element and bundle deformation models

Code	Finite Element Dimensionality	Spatial Dimensionality	Modeling Capabilities	Numerical Method	Implementation
BOW	1D	3D	Element, Bundle	Finite Element	FORTRAN
Bell Model	1D	3D	Bundle	Finite Element	COMSOL Multiphysics
Williams Model	3D	3D	Element	Finite Element	ANSYS Mechanical
McCluskey Model	3D	3D	Partial Element	Finite Element	Abaqus/CAE

Although the models discussed in this section provide satisfactory results for certain conditions and fuel performance parameters, there are significant drawbacks to using commercial codes for fuel deformation analyses. The first issue is the cost. The costs of ANSYS, Comsol and Abaqus licenses are very high for a single license. As the model being simulated increases in size and complexity, additional computational resources are required and High Performance Computing (HPC) licenses are required. For example, the cost of ANSYS's HPC license continually increases as additional processors are desired for simulations. In addition, there is yearly maintenance costs associated with commercial codes in addition to the initial cost of the license. Moreover, ANSYS and Abaqus contain limited multiphysics capabilities, and the addition of material properties are typically in the form of look-up tables or strict equation formats. The ability to add custom equations is very difficult and troublesome. In addition, important nuclear phenomena such as fission gas production and release, fuel swelling and densification, grain boundary evolution, and neutronics need to be coupled via external codes. This coupling is difficult and support is limited. While Comsol has very good multiphysics capabilities and has the ability to add any custom equations and models desired, the scalability and high performance capabilities are poor up to version 4.2b as per discussion with users of Comsol at both RMCC and AECL-CRL. Although the purpose of the models discussed in this section were for fuel element or bundle deformation, they should still accurately predict fuel performance parameters such as temperatures, and stresses and strains, which requires the inclusion of additional physics phenomena. Therefore, a modeling package is required that is scalable, is customizable for nuclear applications, and is robust to model all of the contacts between the individual pellets and between the pellets and the sheath, all at a minimal financial and computational cost.

Chapter 3 - Goals of Research

As explained in Chapter 2, codes used in industry for evaluating fuel performance and deformation are primarily one-dimensional, or quasi-two-dimensional in nature. In Canada and the United States, for fuel performance, there is typically a separate code for normal operating and transient conditions. Recent work (Williams [27], McCluskey [28]) have begun investigating the possibility of developing an all-encompassing three-dimensional model that is able to predict fuel performance parameters such as centerline temperature, sheath hoop strain, and pellet hourglassing, as well as estimate the sagging under high temperatures and bowing due to thermal gradients. However, these models contain limited amounts of phenomena that contribute to the fuel performance parameters that ultimately affect the deflection under sagging and bowing conditions. Moreover, the cost of licenses for the commercial codes limits the capabilities of running larger and more detailed models. Therefore, a new approach to modeling coupled fuel performance and 3D deformation of a fuel element under both normal operating and transient conditions is desirable.

This work introduces the MOOSE computational framework and outlines its discretization and solver processes, as well as its contact capabilities. Furthermore the HORizontal nuclear fuel Simulation Environment (HORSE) built upon the MOOSE framework is presented. HORSE provides all of the physics and material properties required for the analyses completed in this work and was developed as part of this work. It contains models for heat generation and transport, momentum conservation, temperature and burnup dependent material properties, fuel swelling and densification, sheath creep, sheath plasticity, pellet-to-pellet and pellet-to-sheath

contact, and gas volume, pressure and composition. The HORSE simulation environment is used to develop models that:

1) Assess the contact capabilities and limitations of the contact algorithms in the MOOSE framework. The pellet-to-sheath interaction greatly effects heat transfer across the gap and the flexural rigidity of the sheath. Moreover the pellet-to-pellet interaction influences the local sheath deformation at the pellet ridges as well as contributing to the flexural rigidity of the fuel element. Therefore an understanding of how contact works in MOOSE is required.

2) Determine the contribution of the fuel pellets to the flexural rigidity of the sheath under increasing loads at the element midplane, and compare against analytical solutions and Williams' model. By varying the load applied and predicting the element deflection in the downward direction, the effective flexural rigidity of the element can be determined.

3) Examine the thermal bowing of a full fuel element under predefined temperature gradients and compare against analytical solutions.

4) Predict important fuel performance parameters of a full fuel element including the inside sheath temperature, fuel surface temperature, fuel-to-sheath heat transfer coefficient, centerline temperature, sheath hoop strain and stress at the pellet ridge, and compare these against the ELESTRES-IST.

The overall goal of this thesis is to assess the capabilities of modeling 3D deformation mechanisms in a CANDU fuel element using the MOOSE computational framework and provide a preliminary analysis of the results. These results are then compared against analytical solutions, numerical models, and ELESTRES-IST.

Chapter 4 - The Computational Framework

The central goal of MOOSE is to provide a robust and efficient solver of boundary value problems in engineering and science involving complex fully coupled partial differential equations. In a boundary value problem the dependent variables must satisfy the governing differential equations everywhere within a known domain of independent variables. Specific conditions, also known as boundary conditions, must be satisfied by the dependent variables on the boundary of the domain of interest. Depending on the physical situation being examined the dependent (field) variables could include displacement, temperature, and heat flux, among others [29]. Since the domains of interest in many engineering applications are complex, such as combustion engines or nuclear fuel, an analytical solution cannot be found. Therefore, a computational technique known as the Finite Element Method (FEM) is often used. The application of the FEM to engineering problems is referred to as Finite Element Analysis (FEA). This technique is employed by MOOSE and it uses a numerical method known as the Jacobian Free Newton-Krylov (JFNK) method to solve the system of equations generated by the FEM approximation.

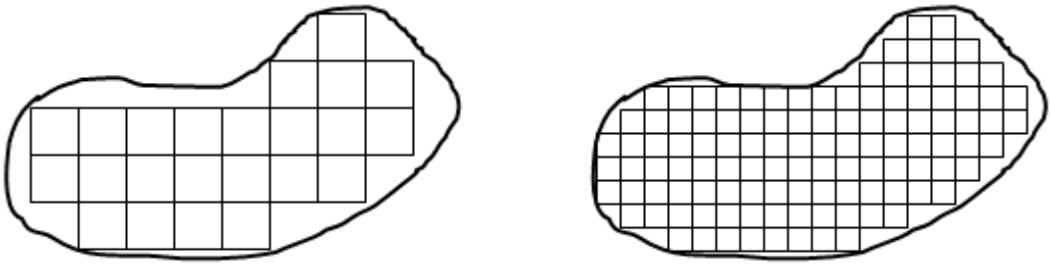
The finite element method is a numerical method used to convert systems of differential equations and boundary conditions on an arbitrary geometry into a system of equations which can be solved yielding an approximate solution to the partial differential equations. The geometry of interest is represented by a finite element mesh that consists of nodes. The groups of nearby nodes combine to form finite elements. Depending on the geometry being modeled the finite elements can be one-dimensional (bars, beams), two-dimensional (quadrilaterals, shells), three-dimensional (hexahedrals, tetrahedrals, shells), or any combination of these elements. The

finite element that is chosen for a particular analysis is one of the parameters that determines the stiffness matrix and the translational and rotational degrees of freedom (dof) that must be solved. Additional parameters that affect the stiffness matrix are the geometry and material properties. Beams, 2D and 3D shells have both translational and rotational dof, whereas bars, quadrilaterals, hexahedrals, and tetrahedrals only have translational dof. Although the field variables are only solved at nodes, they are interpolated between the nodes by polynomial functions known as shape functions. The form of the shape functions depend upon the type and order of finite element used.

The FEM is always an approximation of the real life system as simplifications and assumptions need to be made in order to numerically solve the problem. A sufficient number of boundary conditions and constraints are required to solve a problem using the FEM. The system under consideration must be sufficiently constrained such that there are no rigid body modes. Rigid body motion occurs when the distance between two given nodes on a body remain the same distance from one another regardless of external forces. In other words the body does not deform due to external forces but rather translates through space because it is unconstrained. An understanding of the affects the boundary conditions and constraints have on the final results is required to interpret whether the solutions obtained are realistic or behaving as expected.

One way to increase the accuracy of the analysis is by increasing the number of nodes in the finite element mesh. This reduces the discretization error. There are two ways to do this, increasing the number of finite elements in the mesh, or increasing the order of the finite elements in the mesh. Increasing the order of the finite elements adds additional nodes to the

elements and increases the order of the polynomial of the shape functions. Typically second order elements, containing quadratic shape functions, are used for complex geometries (e.g., circles, cylinders) as they are more capable of capturing the curved geometry. As the order or number of elements in the mesh increases, the computational requirements also increase. The art of FEM becomes the optimization of the mesh such that acceptable results are obtained in an acceptable amount of computational time. The MOOSE framework currently supports finite elements with only translational degrees of freedom (e.g., hexahedrals, quadrilaterals, tetrahedrons) of varying orders. Figure 4.1 shows how a complex geometry can be more accurately represented by a finite element mesh by increasing the element density. Note that this illustration assumes uniform element size. Increasing the order would allow the edges of the elements in the figure to be curved such that even more of the domain of interest could be captured.



(a) **(b)**

Figure 4.1: Illustration showing that (a) a coarser finite element mesh captures less of the geometry than (b) a denser mesh.

The model created in this work is developed upon the Mulitphysics Object-Oriented Simulation Environment computational framework developed at the INL. The MOOSE framework is developed in the C++ programming language and is object-oriented in nature, which strongly depends upon the programming concepts of class templates and inheritance. A class is a user-

defined type that contains its own variables and functions that can only be used within that particular class. Class templates are recipes for generating a family of classes using one or more parameters. Class templates contain two components, a unique name, and a set of parameters. The arguments provided for each parameter in the template determine the definition of the class that is generated [30]. The concept of inheritance is essential to reduce the reproduction of identical code to different files within the MOOSE framework. Inheritance in C++ is the ability to define one type in terms of another, where the class being inherited from is known as the base class. For example, in MOOSE the Material base class provides access to important parameters in the finite element mesh including the quadrature point and element numbers, the Cartesian coordinates of the quadrature points and many others. When a user wants to create their own material property, for example thermal conductivity, one would generate their own ThermalConductivity class that inherits all the members (functions and variables) of the Material base class because material properties are defined at the quadrature points on the mesh. The user then can add functions and variables that are unique to this newly defined class. It can be seen that inheritance is powerful in a framework like MOOSE because there could be many different types of materials and physics that could be inherited from their respective base class, thereby reducing the duplication of code in multiple instances.

It is important to note that the MOOSE framework itself does not contain any specific physics or material properties. In general, MOOSE only contains the computational framework required for extracting the finite element mesh from the mesh file, ensuring boundary conditions, material properties and loads are applied, and performing the numerical calculations. MOOSE also contains most of the base classes required to develop any model. Many capabilities of MOOSE

are provided by the Portable Extensible Toolkit for Scientific Computation (PETSc) data structures, Trilinos software project, and libMesh library. The libMesh library is a finite element framework for solving highly coupled nonlinear partial differential equations on both serial and parallel computer environments. Depending on the architecture and operating system of the computer system being used, the linear systems of equations within libMesh are solved using either PETSc or the Trilinos Project [31]. PETSc is a highly scalable suite of routines used to solve scientific applications of partial differential equations in parallel [32]. The Trilinos Project is an endeavour to develop package based algorithms in an object-oriented framework for solving complex multiphysics problems in science and engineering [33]. Most MOOSE users utilize the PETSc formulation with libMesh. This compilation of libraries was used in all analyses presented in this work.

As stated previously, MOOSE does not contain any physics of its own. Therefore, science and engineering applications that are built upon the MOOSE framework introduce the physics used in simulations. For consistency, each and every MOOSE based application is given an animal name to distinguish itself as a member of the 'herd'. A variety of applications have already been developed such as the Extended Library of Kernels (ELK), MARMOT, FOX, BISON, and many others. Figure 4.2 illustrates a snapshot of the hierarchy of some of the MOOSE based applications. As one moves up the hierarchy the particular application becomes more specialized. ELK is a general library of common physics that can be applied to a wide range of engineering and science fields, including solid mechanics, heat transfer, fluid dynamics (Navier-Stokes), and diffusion. FOX is a set of routines for nuclear fuel performance analysis of LWRs. The algorithms, including material properties in FOX are generally for uranium dioxide and

mixed oxide (MOX) fuels, and Zircaloy-4 and stainless steel cladding. BISON is the main fuel performance code developed at INL that is applicable to light water reactor fuel rods, TRISO fuel particles, metallic rod, and plate fuel geometries, and a variety of material types including uranium-silicides.

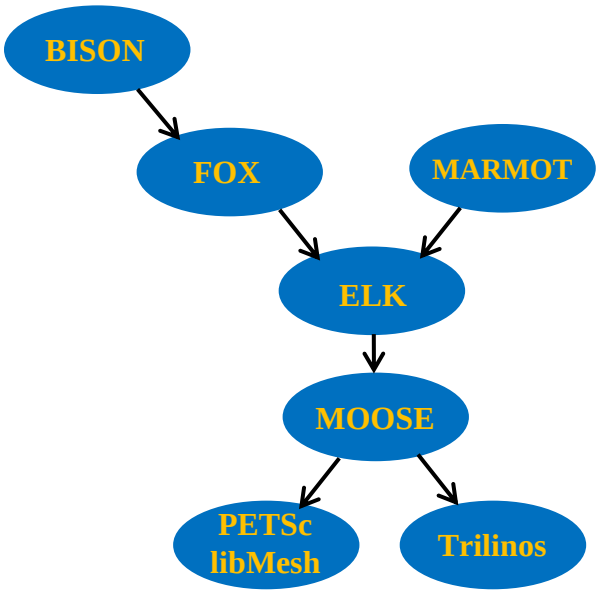


Figure 4.2: A subset of the hierarchy of the MOOSE Framework

The algorithms within FOX and BISON are export controlled by the United States Department of Energy (DOE), but the source code is accessible once clearance has been granted. MARMOT is a different branch off of ELK that focuses on mesoscale irradiation induced microstructure evolution. The phase field analysis predicts the effect of radiation damage on void nucleation and growth, bubble growth, grain boundary migration, and gas diffusion and segregation, as well as calculating the effect of microstructure evolution on bulk properties, such as thermal conductivity and porosity. Some analyses have been completed that couple MARMOT to BISON such that the effect of microstructure evolution on the bulk properties is fed into the material properties used on the continuum scale [34]. The HORSE framework developed in this

work is developed upon ELK. It is important to note there are many other MOOSE based applications built upon the framework that have nothing to do with nuclear engineering. The applications presented here are of the most interest to the work undertaken in this study.

4.1 Finite Elements in MOOSE

In the introduction of this chapter, the finite element method was introduced. It was mentioned that shape functions are used to determine the value of variables in-between nodes of a finite element. In MOOSE, shape functions are also used to discretize what is known as the weak form of a partial differential equation (PDE). The steps of converting a strong form of a PDE to its weak form is illustrated using the divergence of the gradient of a field variable as is found in the heat conduction equation. This example is adapted from a similar example in the MOOSE training manual [35]:

$$-\nabla \cdot k\nabla u = f \tag{4.1}$$

where the k variable is independent of the field variable (in this example), u is the field variable and f is a forcing function. The first step is to rearrange terms so that zero is on the right hand side of the equal sign resulting in

$$-\nabla \cdot k\nabla u - f = 0 \tag{4.2}$$

The next step is to multiply the entire equation by a 'test' function (ψ) to obtain

$$-\psi(\nabla \cdot k\nabla u) - \psi f = 0 \tag{4.3}$$

Then the entire equation is integrated over the domain of interest (Ω) yielding

$$-\int_{\Omega} \psi(\nabla \cdot k\nabla u)\, dV - \int_{\Omega} \psi f\, dV = 0 \tag{4.4}$$

By examining the leftmost term in equation (4.4) one can see that it is a volume integral of the divergence of a quantity. Recalling the divergence theorem one can convert the volume integral into a surface integral to make the discretization process simpler. In general the divergence theorem states

$$\int_{\Omega} \nabla \cdot \vec{F}\, dV = \int_{\partial\Omega} \vec{F} \cdot \vec{n}\, ds \tag{4.5}$$

where $\vec{F}$ is an arbitrary function of a field variable and $\vec{n}$ is a vector normal to the surface being integrated around. Assuming $\vec{F} = -k\nabla u$ the divergence theorem gives

$$-\int_{\Omega} \psi(\nabla \cdot k\nabla u)\, dV = \int_{\Omega} \nabla\psi \cdot k\nabla u\, dV - \int_{\partial\Omega} \psi(k\nabla u \cdot \vec{n})\, ds \tag{4.6}$$

Substituting equation (4.6) into equation (4.4) produces the following

$$\underbrace{\int_{\Omega} \nabla\psi \cdot k\nabla u\, dV}_{Kernel} - \underbrace{\int_{\partial\Omega} \psi(k\nabla u \cdot \vec{n}) ds}_{BC} - \underbrace{\int_{\Omega} \psi f dV}_{Kernel} = 0 \tag{4.7}$$

which now needs to be discretized such that it can be solved on a finite element mesh. Equation (4.7) indicates that the first and third terms are kernels and the second term is a boundary condition. In MOOSE a kernel is a type of physics or more generically a term in the weak form of a PDE that applies to the entire domain of interest and a boundary condition is something that applies to the edge of the domain. As a developer, coding the weak form as given in equation (4.7) is sufficient to run simulations in MOOSE as the discretization is done internally. However, it is important to take this example one step further and carry out the discretization to gain a more in depth understanding of how MOOSE works. Discretization is achieved using shape functions (ϕ) such that

$$u \approx u_h = \sum_{j=1}^{M} u_j \phi_j \tag{4.8}$$

and

$$\nabla u \approx \nabla u_h = \sum_{j=1}^{M} u_j \nabla \phi_j \tag{4.9}$$

where u_h is the discretized field variable. In the Galerkin finite element method used in MOOSE the same shape functions are used as the test functions implying that $\psi_j = \phi_j$ converting the weak form into

$$\int_{\Omega} \nabla \psi_j \cdot k \nabla u_h \, dV - \int_{\partial \Omega} \psi_j (k \nabla u_h \cdot \vec{n}) \, ds - \int_{\Omega} \psi_j f \, dV = 0 \tag{4.10}$$

where the left-hand side of the equation is referred to as the j[th] component of the residual vector and defined as $R_j(u_h)$. The integrals in equation (4.10) are then solved numerically using Gaussian Quadrature [35]. Gaussian quadrature approximates continuous integrals by discrete sums:

$$\int f(\vec{x}) = \sum_{qp} f(\vec{x}_{qp}) w_{qp} \tag{4.11}$$

where $\vec{x}_{qp}$ is the position of the quadrature point and w_{qp} is the weighting factor associated with the quadrature point. The values of the weighting factors are determined by the order of quadrature used. In MOOSE the default order is automatically chosen depending on the finite elements used. The user has the option of choosing a different quadrature order. The sampling of the field variable u for quadrature yields

$$u(\vec{x}_{qp}) \approx u_h(\vec{x}_{qp}) = \sum_{qp} u_j \phi_j(\vec{x}_{qp}) \tag{4.12}$$

and

$$\nabla u(\vec{x}_{qp}) \approx \nabla u_h(\vec{x}_{qp}) = \sum_{qp} u_j \nabla \phi_j(\vec{x}_{qp}) \tag{4.13}$$

which can then be substituted into the left-hand side of equation (4.10) to obtain the form of the j[th] component of the residual vector in the simulation:

$$\begin{aligned} R_j(u_h) = & \sum_{qp} w_{qp} \nabla \psi_j(\vec{x}_{qp}) \cdot k(\vec{x}_{qp}) \nabla u_h(\vec{x}_{qp}) \\ & - \sum_{qp_{face}} w_{qp} \psi_j\left(\vec{x}_{qp_{face}}\right) k\left(\vec{x}_{qp_{face}}\right) \nabla u_h\left(\vec{x}_{qp_{face}}\right) \cdot \vec{n}\left(\vec{x}_{qp_{face}}\right) \\ & - \sum_{qp} w_{qp} \nabla \psi_j(\vec{x}_{qp}) f(\vec{x}_{qp}) \end{aligned} \tag{4.14}$$

It is this final form of the residual vector that is used internally within MOOSE, with all material properties being evaluated at the quadrature points.

4.2 Jacobian Free Newton-Krylov Methods

The MOOSE computational framework is believed to be more efficient and robust for some problems then current numerical solvers, such as Newton-Raphson used by commercial codes (e.g., ANSYS) because it uses a Jacobian Free Newton-Krylov (JFNK) numerical method. The exercise of deriving the discretized weak form in Section 4.1 was necessary to define the residual vector such that one can understand how the JFNK methods introduced in this section minimize the residual vector and solve the system of nonlinear equations in an efficient manner.

JFNK methods are combinations of Newton-type methods for linearly convergent solutions of nonlinear systems of equations and Krylov subspace algorithms for solving the Newton

correction equations. The connection between the Newton and Krylov methods is the Jacobian-vector product that can be approximated such that the analytical determination of the Jacobian is not required. However, to obtain a convergent solution approximations of the Jacobian may be required, which is known as preconditioning [36]. The following subsections introduce a JFNK method by explaining Newton's method, a Krylov method, and preconditioning.

4.2.1 Newton's Method

Newton's method is a root finding method of solving nonlinear equations. For a nonlinear equation of a single variable Newton's method yields

$$x_{n+1} = x_n + \delta x_{n+1} \tag{4.15}$$

where x_n is the current value of the variable, x_{n+1} is the next iteration of the variable and

$$\delta x_{n+1} = -\frac{f(x_n)}{f'(x_n)} \tag{4.16}$$

For a single variable it is quite simple to solve this equation, however computational constraints can become a factor when solving a system of nonlinear equations as one obtains

$$\mathbf{J}(\mathbf{u}_n)\delta\mathbf{u}_{n+1} = -\mathbf{R}(\mathbf{u}_n) \tag{4.17}$$

where **R** is the residual vector given by the weak form of each nonlinear equation and **J** is the Jacobian matrix with its elements given by

$$J_{ij} = \frac{\partial R_i(\mathbf{u})}{\partial u_j} \tag{4.18}$$

It can be seen from equation (4.18) that the Jacobian matrix is a complex object to find. The elements of the Jacobian are found by taking the partial derivative of each nonlinear equation in the system with respect to each variable being solved for in the system. Analytical derivative calculations can be difficult and error prone. Therefore, a method that eliminates the need to

solve for the Jacobian explicitly is desired. Equation (4.17) is essentially a system of linear equations that need to be solved to obtain $\delta\mathbf{u}_{n+1}$ which is then used to solve the nonlinear system as given by equation (4.15) but in vector form. This system of linear equations is solved using a Krylov solver.

4.2.2 Krylov's Method

MOOSE employs the Generalized Minimized Residual (GMRES) iterative Krylov solver. In this method the representation of the solution to the linear system of equations is given by

$$\delta\mathbf{u}_{n+1}^{k} = a_0\mathbf{r}_0 + a_1\mathbf{J}\mathbf{r}_0 + a_2\mathbf{J}^2\mathbf{r}_0 + \ ... + a_k\mathbf{J}^k\mathbf{r}_0 \tag{4.19}$$

where the scalars a_i minimize the residual, *k* is the Krylov iteration number, and $\mathbf{r}_0$ is the initial linear residual vector defined as

$$\mathbf{r}_0 = -R(u_h) - J(u_h)\delta u_0 \tag{4.20}$$

given an initial guess δu_0. Using equation (4.19), the Krylov method is iterated until the right-hand side of equation (4.17) is within some specified tolerance of the initial residual. By examining equation (4.19) it can be seen that for every Krylov iteration an additional term is added to the right hand side of the equation. Once the Krylov solution is within the specified tolerance the nonlinear step is said to have converged. Equation (4.19) also illustrates that only the vector-matrix product of the Jacobian and the Krylov vector is required which can be approximated by a finite difference of the form [36]

$$\mathbf{J}\mathbf{v} \approx \frac{\mathbf{R}(\mathbf{u} + \varepsilon\mathbf{v}) - \mathbf{R}(\mathbf{u})}{\varepsilon} \tag{4.21}$$

where,

$$\varepsilon = \frac{1}{N\|\mathbf{v}\|_2}\sum_{i=1}^{N} b|u_i| + b \tag{4.22}$$

which represents a small perturbation, b = 1.0 x 10^{-8}, N is the number of unknowns, and $\mathbf{v}$ is a Krylov vector (i.e., $\mathbf{v} \in (\mathbf{r}_0, \mathbf{J}\mathbf{r}_0, \mathbf{J}^2\mathbf{r}_0, \ldots)$). Note that equation (4.22) is one possibility to use for ε. This method produces the average ε that one would obtain if each element of the Jacobian was computed as

$$J_{ij} = \frac{R_i(\mathbf{u} + \varepsilon_j \mathbf{e}_j) - R_i}{\varepsilon_j} \tag{4.23}$$

where $\varepsilon_j = bu_j + b$. The vector $\mathbf{e}_j$ contains zeros in all locations except the jth location where the value is one. The advantage of using this approximation is that a large amount of computational time is saved as analytical or computational derivatives are not required to compute **J** and no memory space is needed to store the potentially large **J** matrix. However, since the GMRES solver stores all of the previous Krylov vectors in memory, it is necessary to minimize the number of Krylov iterations required to solve the linear system of equations [37].

4.2.3 Preconditioning

Minimization of the Krylov iterations is accomplished by using what is known as right preconditioning. For each Krylov iteration preconditioning is achieved by solving

$$\mathbf{J}(\mathbf{u}_n^k)\mathbf{P}^{-1}\mathbf{P}\delta\mathbf{u}_{n+1}^k = -\mathbf{R}(\mathbf{u}_n^k) \tag{4.24}$$

where **P** represents the preconditioning matrix and $\mathbf{P}^{-1}$ the inverse of the preconditioning matrix. Right preconditioning is achieved through a two-step process. The first step is to solve

$$\mathbf{J}(\mathbf{u}_n^k)\mathbf{P}^{-1}\mathbf{w} = -\mathbf{R}(\mathbf{u}_n^k) \tag{4.25}$$

for **w**. By examining equation (4.24) it can be seen that

$$\mathbf{w} = \mathbf{P}\delta\mathbf{u}_{n+1}^{k} \tag{4.26}$$

The second step is to then solve

$$\delta\mathbf{u}_{n+1}^{k} = \mathbf{P}^{-1}\mathbf{w} \tag{4.27}$$

for $\delta\mathbf{u}_{n+1}^{k}$,which yields the right preconditioned form of equation (4.21) given by [36]

$$\mathbf{J}\mathbf{P}^{-1}\mathbf{v} \approx \frac{\mathbf{R}(\mathbf{u} + \varepsilon\mathbf{P}^{-1}\mathbf{v}) - \mathbf{R}(\mathbf{u})}{\varepsilon} \tag{4.28}$$

By choosing the correct preconditioner $\mathbf{P}^{-1}$ the number of Krylov iterations in the GMRES method can be minimized. The difficulty that arises is to choose the correct type of preconditioner for the nonlinear system under consideration. There are many options available such as Newton-Krylov-Schwarz, Multigrid and physics-based preconditioners [36]. A common physics based preconditioner is operator splitting. Operator splitting refers to the splitting of the solution process based on different types of physics [38]. The types of preconditioning offered in MOOSE are included in the PETSc library. The details of the different preconditioners can be found in the PETSc User Manual [39]. The MOOSE developers advise new users to use the default block diagonal preconditioner until convergence issues are encountered [35].

4.3 Contact in MOOSE

In finite element analyses there are three types of structural nonlinearities that can be encountered: geometric (large deformation), material, and contact. Geometric nonlinearity is when the equilibrium equations used in the FEM must be written in terms of the deformed geometry as the displacements and strains are very large. Material nonlinearity occurs when the material properties of the components are dependent upon the stress or strain within the material. This is particularly important within the sheath in CANDU reactors as over time and at high

temperatures the sheath can permanently deform due to plastic behaviour and creep. Contact nonlinearity occurs when there is a sudden change in stiffness of the system as occurs when two or more bodies touch [40]. In the model developed in this work, contact is present between individual pellets and the pellets and the sheath. Therefore an understanding of how contact is employed and enforced in MOOSE is required. This understanding is of key interest in this thesis because it is an ongoing challenge in modeling bundle deformation.

The contact algorithms used by MOOSE based applications are housed in ELK. ELK employs a node-to-face routine to enforce contact. In MOOSE simulations, the contact algorithms for mechanical and thermal contact are declared separately in the input file but solved simultaneously. In node-to-face contact, a contact pair is defined that consists of a master and slave surface. The nodes on the slave surface are allowed to penetrate the faces (element faces in 3D) of the master surface, but penetration must be reduced as much as possible to obtain the most accurate solution. To ensure reduced penetration when using node-to-face contact, the slave surface must have a finer mesh than the master surface.

In nuclear fuel simulations, mechanical contact occurs between each individual pellet and its neighbouring pellets, as well as the exterior surface of all the pellets with the interior surface of the sheath. The contact models and enforcement methods within ELK attempt to satisfy three requirements of mechanical contact [41]:

$$g \leq 0 \tag{4.29}$$

$$f_c \geq 0 \tag{4.30}$$

$$f_c g = 0 \tag{4.31}$$

where g is the gap distance between the master surface and the slave nodes, and f_c is the contact force that is opposing penetration of the slave nodes. Ideally one of these values must be zero at all times. In pellet-to-pellet contact the choice of the master and slave surfaces is irrelevant as the surfaces coming into contact have the exact same mesh. For pellet-to-sheath contact the interior surface of the sheath is classified as the master and the exterior surface of the fuel pellets are defined as the slave. There are three different methods of modeling mechanical contact in ELK: glued, frictionless, and coulomb frictional. Glued contact enables two bodies to come into contact, but once in contact, these bodies are unable to come out of contact. This enforcement is numerically robust but it does not allow tangential sliding between surfaces, which can affect the stress fields obtained.

Frictionless contact allows bodies to come into and out of contact, and permits tangential movement. However, there is no resistance to movement in the tangential direction, which is an issue when simulating fuel in a horizontal fuel configuration. For a vertically oriented fuel design, as is the case in LWRs, gravitational forces acting on the pellets would force all the pellets into contact and effectively allowing them to be meshed together (assuming no eccentricity of the fuel stack). Frictionless contact is fairly robust if the bodies can be numerically constrained without significantly affecting the physical results. Generally speaking, under-constraining the system allows for rigid body motion, which is unsolvable for time-independent models using an implicit solution method. This phenomenon affects all implicit

finite element codes. Implicit solution methods, as employed in MOOSE, obtain a solution by solving an equation involving the current state of the system and the subsequent one. This can be seen in equation (4.15).

There are two methods of enforcing frictionless and glued contact: penalty and kinematic. Each method has its own way of determining the contact force and residual of the slave and master nodes based upon the geometric search completed for each node. Figure 4.3 describes how the geometric search is completed and defines the important parameters required for the contact enforcement methods. In Figure 4.3 $\boldsymbol{x}$ is the current location of the slave node, $\boldsymbol{x_p}$ is the projected location of the slave node on the master surface, $\boldsymbol{x_{p_{t-1}}}$ is the projected location of the slave node from the previous iteration, $\boldsymbol{g}$ is the gap distance, and $\boldsymbol{s}$ is the slip distance.

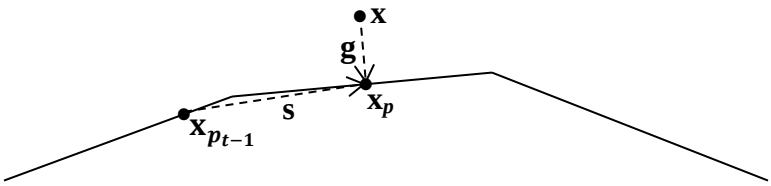


Figure 4.3: Illustration of the geometric search algorithm used for both thermal and mechanical contact.

Using the parameters defined in Figure 4.3 the equations for mechanical contact enforcement, including the contact force, residual of the slave node, and residual of the master node can be defined. Table 4.1 lists the mechanical contact enforcement equations for both penalty and kinematic enforcement of the glued and frictionless contact models. Bolded parameters indicate a vector quantity. The table is reproduced from [42].

Table 4.1: Equations for penalty and kinematic contact enforcement of the glued and frictionless mechanical contact models.

	Penalty	Kinematic
Glued	$\mathbf{f}_c = k_p(\mathbf{x} - \mathbf{x}_{p_{t-1}})$ $\mathbf{r}_{s+1} = \mathbf{r}_s + \mathbf{f}_c$ $\mathbf{r}_{m+1} = \mathbf{r}_m - \phi_i \mathbf{f}_c$	$\mathbf{f}_c = -\mathbf{r}_{s_{copy}}$ $\mathbf{r}_{s+1} = \mathbf{r}_s + \mathbf{f}_c + k_p(\mathbf{x} - \mathbf{x}_{p_{t-1}})$ $\mathbf{r}_{m+1} = \mathbf{r}_m - \phi_i \mathbf{f}_c$
Frictionless	$\mathbf{f}_c = \mathbf{n} \cdot \mathbf{n} \cdot k_p(\mathbf{x} - \mathbf{x}_p)$ $\mathbf{r}_{s+1} = \mathbf{r}_s + \mathbf{f}_c$ $\mathbf{r}_{m+1} = \mathbf{r}_m - \phi_i \mathbf{f}_c$	$\mathbf{f}_c = -\mathbf{n} \cdot \mathbf{n} \cdot \mathbf{r}_{s_{copy}}$ $\mathbf{r}_{s+1} = \mathbf{r}_s + \mathbf{f}_c + \mathbf{n} \cdot \mathbf{n} \cdot k_p(\mathbf{x} - \mathbf{x}_{p_{t-1}})$ $\mathbf{r}_{m+1} = \mathbf{r}_m - \phi_i \mathbf{f}_c$

Where in the equations in Table 4.1, $\mathbf{f}_c$ is the contact force, k_p is the penalty factor, $\mathbf{n}$ is the vector representing the outward normal from the master surface, ϕ_i is the ith shape function connected to the node, the subscripts s and m denote slave and master respectively, $\mathbf{r}$ is the residual vector, and $\mathbf{r}_{s_{copy}}$ is a copy of the residual at the slave node from the previous timestep. From the table it can be seen that in penalty contact enforcement the penalty factor behaves as a spring constant for an artificial spring that is attached between the master and slave surfaces. Thus, for penalty enforcement, there has to be some small finite interpenetration between the master and slave surfaces or else the contact force and consequentially the contact pressure would be zero, which is unphysical when two bodies are in contact. A model used to analyze the glued, frictionless and frictional contact models with both penalty and kinematic enforcement is introduced in Section 6.1 and the results and discussion are provided in Section 7.1. Kinematic enforcement still uses a penalty factor but it is only used to penalize the error in the solution. In this case, zero penetration is enforced resulting in the term containing the penalty factor dropping out of the slave residual calculation causing the slave residual to approach zero yielding the maximum contact force (and pressure) that would be calculated for that node.

Coulomb friction is an attempt in ELK to implement frictional contact. In frictional contact nodes of the slave surface can come into contact with the master surface and translate tangentially along the master surface with some resistance due to the coefficient of friction. The coefficient of friction causes the node to slip across the surface. Similarly to glued and frictionless contact, there are two methods of enforcing contact, penalty and kinematic. However, in the case of frictional contact the implementation of these enforcements is different. The penalty method is analogous to elastic-perfect plasticity where the normal force is used to predict the tangential force using the penalty stiffness. Some penetration is required to calculate the normal force in the same manner as for frictionless contact. This force is then multiplied by the coefficient of friction to obtain an estimate of the tangential force. If the tangential force exceeds the frictional capacity of the surface, the tangential force is limited to the frictional capacity. The tangential displacement is then composed of a penalty and a slip component.
A different approach to calculating slip is used in the kinematic enforcement algorithm. In this case two nested solves are used, the glued solve, and the slip update solve. In the inner solve (glued solve) sliding constraints are treated as glued contact with kinematic enforcement. Once convergence is achieved, a slip calculation is performed. Nodes that have tangential forces greater than the frictional capacity given by the coefficient of friction multiplied by the normal force undergo slip. The slip calculation is repeated until all nodes have their tangential force equal to the frictional capacity at which point the inner glued contact loop is entered for the next timestep. The slip of the slave nodes is calculated by

$$\mathbf{s}_i = \frac{\mathbf{f}_{cT}}{|\mathbf{f}_{cT}|}(|\mathbf{f}_{cT}| - \mu f_{cN}) \cdot K_{diag}^{-1} \tag{4.32}$$

where $\mathbf{f}_{cT}$ is the contact force in the tangential direction, μ is the coefficient of friction, f_{cN} is the contact force in the normal direction, and K_{diag}^{-1} is the diagonalized stiffness matrix of the slave

nodes used as a regularization parameter [42]. The friction coefficient is specified by the user. The displacement variables are then updated with the slip as follows:

$$\mathbf{u}_{i+1} = \mathbf{u}_i + \mathbf{s}_i \tag{4.33}$$

Coulomb friction is a new development and is currently not as robust as required for the analyses completed in this work. The results obtained using penalty enforcement are less accurate at low penalty factors but the computational requirements are orders of magnitude less than kinematic enforcement [42]. In addition penalty enforcement is generally more stable as enforcing zero interpenetration, as is the case with kinematic enforcement, can lead to 'chatter' or oscillatory solutions. An investigation of the penalty enforcement of frictional contact is also completed as part of the contact analysis introduced in Section 6.1 and discussed in Section 7.1.2.

Chapter 5 - HORSE Development

Nuclear fuel modeling is a complex endeavour that requires the numerical solution to governing equations given detailed models of the material properties and relevant phenomena. In this section, the governing equations, the material models of the fuel, sheath, fuel-to-sheath gap and coolant, and the deformation mechanisms of the fuel and sheath are presented. The governing equations in HORSE are fully-coupled PDEs for energy and momentum conservation. Additional nuclear phenomena that are included in HORSE are fuel swelling and densification, sheath plasticity, sheath creep, burnup, and volumetric heat generation accounting for radial flux depression.

5.1 Heat Generation and Transport

The primary goal of nuclear energy production in Canada is to generate electricity primarily through the fission process of uranium atoms present in the fuel used in CANDU reactors. The fission process occurs when a uranium atom absorbs a thermal (low kinetic energy) neutron. This absorption of the neutron causes the fissionable uranium atom to split into two smaller nuclei known as fission fragments. There are always two fission fragments produced, typically one heavy and one light [43]. Momentum conservation requires that the majority of the binding energy of the original nucleus is passed to the heavier atom. This kinetic energy of the fission fragments is deposited within the fuel when the fragments come to rest within the fuel. The heat generated by the fission process is then passed across the fuel-to-sheath gap via conduction, convection and radiation processes before conducting through the sheath, after which point it is finally transferred to the coolant via convection. The coolant then passes through the primary

heat transport system to the steam generators to produce electricity. The energy balance of the heat transfer from the fuel to the coolant is governed by the heat conduction equation

$$\rho C_p \frac{\partial T}{\partial t} + \nabla \cdot \mathbf{q} - Q_{fission} = 0 \tag{5.1}$$

where ρ, T, and C_p are the density of the material, temperature, and specific heat (at constant pressure) of the material, respectively. The heat flux is calculated by

$$\mathbf{q} = -k\nabla T \tag{5.2}$$

where k is the thermal conductivity of the material. The $Q_{fission}$ term is the volumetric heat generation due to fission in units of W m^{-3}. Due to the deposition of heat in the fuel relatively close to the site of fission, a distribution of the heat generation can be determined based upon the neutron flux [21]. The fission products formed within the fuel during irradiation have higher neutron absorption cross-sections. In addition, plutonium is formed in the rim region of the fuel due to a neutron capture process of uranium-238. This leads to radial flux depression across the fuel. The equation used for the neutron flux profile within the fuel element is an analytical solution to the neutron diffusion equation for a cylindrical geometry containing a central hole [44] and the flux depression parameters were developed using the HAMMER neutron transport code [45]. It is given by

$$\phi = \left(I_0(\kappa_{flux} r) + \frac{I_1(\kappa_{flux} H_r)}{K_1(\kappa_{flux} H_r)} K_0(\kappa_{flux} r) + \beta_{flux} e^{\lambda_{flux}(r - P_r)} \right) \tag{5.3}$$

where I_n is the nth order modified Bessel function of the first kind, K_n is the nth order modified Bessel function of the second kind, H_r and P_r are the hole and pellet radius of the fuel respectively, r is the radial location within the pellet, and κ_{flux}, β_{flux}, and λ_{flux} are flux depression parameters. If there is no central hole, the central hole radius is set to zero and the second term in equation (5.3) is zero. The flux depression parameters are determined using a

linear interpolation scheme from a table of values determined using the HAMMER code as a function of the initial fuel pellet radius, UO_2 enrichment of the fuel, and the average fuel burnup of the element. The table used for interpolation of these flux parameters is included in Appendix A. It is important to note that this table of flux depression parameters are legacy values used in the ELESTRES-IST, and the values presented are unique to the thermal neutron spectrum of a heavy water moderated CANDU reactor. An updated table for CANDU reactors or neutron spectrums experienced in other reactors can be determined using a neutron physics code (e.g., WIMS-AECL, WIMS9, DRAGON). In this work the fuel does not contain a central hole and the flux equation simplifies to

$$\phi = \left(I_0\left(\kappa_{flux} r\right) + \beta_{flux} e^{\lambda_{flux}(r - P_r)}\right) \tag{5.4}$$

To obtain a correlation for the volumetric heat generation, equation (5.4) must be normalized to the required linear power of the fuel element, by multiplying by a proportionality coefficient denoted f_{mag}. This representation of the volumetric heat generation has been previously used in a variety of other fuel modeling codes including the IST codes ELESTRES and ELOCA, FAST the fuel performance model developed by Prudil [20], and the Fuel Operational peRformance Computations in an Element (FORCE) model created by Shaheen [23]. The proportionality coefficient is calculated from the linear power as a function of the axial length of a portion of the fuel element, and by equating to the required heat generation as follows

$$P_{lin} = \frac{2\pi \int_A Q_{fission} r dA}{L} \tag{5.5}$$

where L represents the length of the fuel stack within the fuel element. Equation (5.5) can be solved for exactly but due the chamfer and dishes on the fuel pellets and the axial gap between the pellets the computations become difficult and intensive [21]. Therefore, if one assumes that

the fuel pellets are right cylinders and there is no axial gap present the linear power can be simplified to

$$P_{lin} = \frac{2\pi L \int_r \ Q_{fission} r dr}{L} = 2\pi \int_0^{P_r} Q_{fission} r dr \tag{5.6}$$

This approximation does introduce an error but the volume of the dishes and chamfers on a single fuel pellet is approximately 1% of the total volume which indicates that the assumption made here is acceptable. By substituting equation (5.4) into equation (5.6) and rearranging, the following is obtained for the proportionally coefficient

$$f_{mag} = \frac{P_{lin}}{2\pi \int_0^{P_r} Q_{fission} r dr} \tag{5.7}$$

The radial flux integral is provided by

$$\int_0^{P_r} Q_{fission} r dr = \int_0^{P_r} I_0(\kappa_{flux} r) r dr + \int_0^{P_r} \beta_{flux} e^{\lambda_{flux}(r-P_r)} r dr \tag{5.8}$$

where

$$\int_0^{P_r} I_0(\kappa_{flux} r) r dr = \frac{P_r}{\kappa_{flux}} I_1(\kappa_{flux} P_r) \tag{5.9}$$

$$\int_0^{P_r} \beta_{flux} e^{\lambda_{flux}(r-P_r)} r dr = \beta_{flux} \left(\frac{P_r}{\lambda_{flux}} + \frac{1}{\lambda_{flux}^2} \left(e^{-P_r \lambda_{flux}} - 1 \right) \right) \tag{5.10}$$

Multiplying equation (5.4) by the calculated proportionality coefficient yields the representation of the volumetric heat generation due to fission incorporated into this analysis:

$$Q_{fission} = \frac{P_{lin}}{2\pi} \left[\frac{\left(I_0(\kappa_{flux} r) + \beta_{flux} e^{\lambda_{flux}(r-P_r)} \right)}{\frac{P_r}{\kappa_{flux}} I_1(\kappa_{flux} P_r) + \beta_{flux} \left(\frac{P_r}{\lambda_{flux}} + \frac{1}{\lambda_{flux}^2} \left(e^{-P_r \lambda_{flux}} - 1 \right) \right)} \right] \tag{5.11}$$

Equation (5.11) is solved for the as fabricated fuel radius. Therefore, to take into account the change in volume due to thermal expansion, fuel swelling and densification, the volumetric heat generation equation is multiplied by the ratio of the local fuel density to the initial density. This is a simpler method than trying to determine the pellet radius as a function of time because the pellet centerline location changes due to sagging and thermal bowing. In reality, the overall mass of the fuel within the element changes due to the fission process, and production of fission fragments. However, determining this change in mass is difficult and cumbersome, and the magnitude is small. Therefore, using the ratio of densities instead of the volume is acceptable because for the analyses completed in this work the mass can be assumed to be essentially constant for the duration of irradiation.

5.2 Momentum Conservation

When an external load is applied to a structure, the structure responds in such a way as to approach an equilibrium position. In MOOSE based applications momentum conservation is determined assuming a static equilibrium is achieved at each time increment via Cauchy's equation [46]

$$\nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{f} = 0 \tag{5.12}$$

where **f** is the body force per unit mass (e.g., gravity) and **σ** is the Cauchy stress tensor given by

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \tag{5.13}$$

The two primary variables in a nuclear fuel analysis are the temperature and displacement vector **u**. The stresses in the Cauchy tensor are connected to **u** via the strain using constitutive relationships. These relationships are introduced in Section 5.5.1.

5.3 Burnup

Fuel burnup is an important parameter in nuclear fuel modeling as it affects the heat generation given in equation (5.11) as well as many of the fuel material properties and nuclear phenomena discussed in subsequent sections in this chapter. Burnup is a measure of the amount of initial fuel atoms that have been consumed by the fission process. There are two typical methods of reporting the burnup, 1) as an atom percent (i.e., percent of initial fissionable atoms consumed), and 2) a measure of the total energy released per heavy atom in the fuel. For CANDU reactors, the burnup is typically measured in MWh kgU^{-1}. The relationship between atom percent is: 1 % burnup ≈ 225 MWh kgU^{-1} [21]. In this work two measures of burnup are required, the local burnup within the fuel and the average burnup for the entire fuel element. The burnups are functions of irradiation time and are calculated as a rate and then integrated over time. The rate of fuel burnup for stoichiometric UO_2 at a particular location within the fuel, in units of MWh $kgU^{-1}\,s^{-1}$, is related to the heat generation rate per fission by

$$\frac{dBu}{dt} = \frac{Q_{fission}}{3.6 \times 10^9 \rho_U} = \frac{Q_{fission}}{3.6 \times 10^9 T_f \left(\frac{238 \rho_{UO_2}}{238+32}\right)} \tag{5.14}$$

where $Q_{fission}$ is given by equation (5.11) in units of W m^{-3}, ρ_{UO_2} is the density of the fuel in kg m^{-3}, ρ_U is the density of the uranium in kg m^{-3} and T_f is the ratio of thermal power to the total fission power. The default value for T_f is taken as 0.925 as used in ELESTRES. Integrating equation (5.14) over the volume of the fuel provides the average element burnup rate which simplifies to

$$\frac{dBu_{avg}}{dt} = \frac{P_{lin} L}{(3.6 \times 10^9) T_f M_U} \tag{5.15}$$

where M_U is the initial mass of uranium in an element in kg, and L is the length of the fuel within the fuel element. A derivation of these equations is given in Appendix B. This method of determining burnup was used by [21].

5.4 Thermo-physical Properties

The thermo-physical properties of materials are the properties that govern heat conduction and mechanical behaviour. For solids, the six important thermo-physical properties include: thermal conductivity, thermal expansion, specific heat capacity, density, modulus of elasticity, and Poisson's ratio. In this section, the first four are defined. The modulus of elasticity and Poisson's ratio are defined in the deformation section (Section 5.5). For fluids (i.e., the coolant) the three most important properties for the analyses completed in this work are the heat transfer coefficient, pressure and temperature.

5.4.1 Properties of Uranium Dioxide Fuel

Accurately modeling the material properties of the uranium dioxide fuel is imperative to obtain useful and interesting results. The most important fuel material property in obtaining accurate centerline temperatures is the thermal conductivity. Over the years a variety of correlations have been suggested for the thermal conductivity of both unirradiated and irradiated uranium dioxide. Fink [47] suggests that the thermal conductivity correlation containing a phonon lattice term, and the polaron ambipolar contribution proposed by Ronchi et al. [48] be used. This correlation applies only for 95% dense UO_2 and is given by

$$k = \frac{100}{7.5408 + 17.692t + 3.6142t^2} + \frac{6400}{t^{5/2}} \exp\left(\frac{-16.35}{t}\right) \tag{5.16}$$

where $t = T/1000$, and T is the temperature in K. However this correlation does not contain corrections for irradiation damage and the build up of fission products within the fuel. To take these effects into account Lucuta et al. [49] proposed the following representation for the thermal conductivity of uranium dioxide:

$$k = \kappa_{1d}\kappa_{1p}\kappa_{2p}\kappa_{3x}\kappa_{4r}\lambda_0 \tag{5.17}$$

where λ_0 is the thermal conductivity of unirradiated UO_2, κ_{1d} is the contribution due to dissolved fission products, κ_{1p} is the contribution due to precipitated fission products, κ_{2p} accounts for pores and fission-gas bubbles, κ_{3x} accounts for the deviation from stoichiometry, and κ_{4r} is the contribution due to irradiation damage. Lucuta recommends that the term for unirradiated UO_2 be of the form

$$\lambda_0 = \frac{1}{A + BT} + \frac{C}{T^2}\exp\left(-\frac{D}{T}\right) \tag{5.18}$$

where the coefficients A, B, C, and D are given in Table 5.1 alongside the modified values used in HORSE. The values chosen for HORSE are the ones used by ELOCA and ELESTRES. The difference in the numbers between HORSE and Lucuta arise from the data that was used for the fit of equation (5.18).

Table 5.1: Parameters required for the unirradiated thermal conductivity

Constant	**Lucuta et al.** [49]	**HORSE**
A (m W^{-1})	0.0375	0.030771
B (m K W^{-1})	2.165×10^{-4}	2.25×10^{-4}
C (W K m^{-1})	4.715×10^{9}	9.28×10^{9}
D (K)	16361	18295.09

The correlations recommended by Lucuta and implemented into HORSE for dissolved and precipitated solid fission products are

$$\kappa_{1d} = \left(\frac{1.09}{\beta^{3.265}} + \frac{0.0643}{\sqrt{\beta}}\sqrt{T}\right)\arctan\left(\frac{1}{\left(\frac{1.09}{\beta^{3.265}}\right) + \left(\frac{0.0643}{\sqrt{\beta}}\right)\sqrt{T}}\right) \tag{5.19}$$

and

$$\kappa_{1p} = 1 + \frac{0.019\beta}{(3 - 0.019\beta)}\frac{1}{1 + \exp\left(-\left(\frac{T - 1200}{100}\right)\right)} \tag{5.20}$$

respectively, where β is the local burnup of the fuel in atom percent. It can easily be seen that the dissolved fission products have a negative impact on the thermal conductivity, whereas the solid precipitated fission products increase the thermal conductivity. This is primarily due to the fact that the solid fission products are mostly metals that have higher individual thermal conductivities than the UO_2 ceramic [19]. Lucuta suggests using the modified Maxwell factor to account for the effect of pores and fission-gas bubbles:

$$\kappa_{2p} = \frac{1 - p}{1 + (\sigma_p - 1)p} \tag{5.21}$$

where p is the volume fraction of pores and bubbles as given by (5.28) and σ_p is the pore shape factor, equal to 1.5 for spherical bubbles. However, in reality the bubbles in the grains are spherical but the bubbles on the grain boundaries are lenticular which is not captured by equation (5.21). Therefore HORSE uses a modified version of the expression suggested by Loeb [50]:

$$\kappa_{2p} = 1.0 - (2.05 - 5.0 \times 10^{-4}T)p \tag{5.22}$$

For models that do not take into account deviation of stoichiometry (i.e., the sheath is assumed to remain intact):

$$\kappa_{3x} = 1. \tag{5.23}$$

If the sheath was to fail, oxidation of the fuel would occur causing the fuel to deviate from stoichiometric UO_2. The effects of radiation damage are characterized by

$$\kappa_{4r} = 1 - \frac{0.2}{1 + \exp\left(\frac{T - 900}{80}\right)} \tag{5.24}$$

It can be seen that the effects due to irradiation damage significantly reduce the thermal conductivity at low temperatures. Another important material property that is required to accurately determine the mechanical and thermal behaviour of the fuel is the density. The density of the fuel is strongly affected by the thermal expansion. The thermal strain of uranium dioxide is taken to be isotropic and given by

$$\varepsilon_{th} = \frac{\Delta L}{L_0} = \frac{L_T}{L_0} - 1 = \alpha \Delta T \tag{5.25}$$

where ΔL is the change in length due to thermal expansion, L_0 is the length at approximately 273 K, L_T is the length at the current temperature, α is the thermal expansion coefficient, and ΔT is the difference in temperature between the current temperature and 273 K.

The modified version of the thermal expansion coefficient given in the MATerial PROperties (MATPRO) library [51] used in this work is

$$\alpha = \frac{K_1 + \frac{K_3 E_D}{kT^2} \exp\left(-\frac{E_D}{kT}\right)}{1 + K_1 T - K_2 + K_3 \exp\left(-\frac{E_D}{kT}\right)} \tag{5.26}$$

where the coefficients K_1, K_2, K_3 and E_D are tabulated in Table 5.2, and $k = 1.38 \times 10^{-23}$ J K^{-1} is Boltzmann's constant.

Table 5.2: Coefficients for the Thermal Expansion of UO_2

Coefficient	Value
K_1 (K^{-1})	1.0×10^{-5}
K_2	3.0×10^{-3}
K_3	4.0×10^{-2}
E_D (J)	6.9×10^{-20}

The change in theoretical density of the fuel due to thermal expansion governed by mass conservation is

$$\rho_{th} = \rho_{th}(273)\left(\frac{L_0}{L_T}\right)^3 \tag{5.27}$$

where $\rho_{th}(273) = 10963$ kg m^{-3} is the theoretical density of UO_2 at 273 K. The porosity of the fuel is defined as

$$p = 1 - \frac{\rho}{\rho_{th}} \tag{5.28}$$

where ρ_{th} is determined from equation (5.27). During irradiation, the fuel's porosity will change. Earlier in the lifetime of the fuel, due to the high temperatures and pressures in the reactor environment the pores within the fuel will shrink or be destroyed and thus through an annealing process the fuel's density increases and some porosity is removed [19]. This process is known as densification. The fraction of initial porosity removed from the fuel due to densification is given by Hastings [52]

$$F = 0.6 - \exp\left(-0.506 - 8.67\times10^{-10}T^3\left(1-\exp(-2.867\times10^{-2}Bu)\right)\right) \tag{5.29}$$

where T is the temperature in Kelvin and Bu is the local burnup in MWh kgU^{-1}. It can be seen that the fraction of initial porosity removed by the densification process will never be more than 60%. As irradiation continues, fission products begin to form and the fuel begins to swell due to the presence of these solid and gaseous products. Solid fission products cause swelling as the

overall space required by two fission fragments is larger than that of a single uranium atom in the UO_2 matrix. The volumetric strain produced by solid fission products suggested by Olander is

$$\frac{\Delta V_{SFP}}{V_0} = 0.0032\frac{Bu}{225} \tag{5.30}$$

where Bu is the average burnup of the fuel in MWh kgU^{-1} [53].

Gaseous fission product swelling occurs due to the formation of fission gas bubbles that migrate to the grain boundaries within the fuel. A modified version of the correlation suggested by MATPRO [51] for the volumetric strain rate due to gaseous fission products is given by:

$$\frac{d(\Delta V_{GFP}/V_0)}{dt} = 8.716(10^{-36})(2800 - T)^{11.73}e^{[-0.0162(2800-T)-7.92344(10^{-7})\rho Bu]}\rho\frac{dBu}{dt} \tag{5.31}$$

where Bu is the local burnup of the fuel in MWh kgU^{-1} as used by Morgan [19] for temperatures below 2800 K. Above 2800 K the swelling is zero as it is assumed all the gas causing the swelling has been released to the fuel-to-sheath gap and plenum regions. The conversion from the standard MATPRO correlation to the one given in equation (5.31) is given in Appendix B. Using the swelling and densification equations, the updated value of the porosity is [19]

$$p = p_0\left(1 - F - \frac{\Delta V_{SFP}}{V_0}\right) + \frac{\Delta V_{GFP}}{V_0} \tag{5.32}$$

where p_0 is the initial porosity of the fuel. The unirradiated density of UO_2 used in CANDU reactors is typically around 10650 kg m^{-3} at 300 K [19] yielding a value for the initial porosity of 0.0286. The density of the fuel can then be determined taking into account fission product swelling and densification processes via

$$\rho = \rho_{th}(1 - p) \tag{5.33}$$

where ρ_{th} is given by equation (5.27). The above method for determining the fuel density was presented and used by Morgan [19]. The specific heat used in this model was suggested by Kerrisk and Clifton [54] in units of cal mol^{-1}:

$$C_p = \frac{K_1 \theta^2 e^{\theta/T}}{T^2 (e^{\theta/T} - 1)^2} + 2K_2 T + \frac{K_3 E_D}{RT^2} e^{\frac{-E_D}{RT}} \tag{5.34}$$

where T is the temperature in K, and the constants K_1, K_2, K_3, θ and E_D are defined in Table 5.3. The value calculated by equation (5.34) is multiplied by 15.496 to convert to SI units of J kg^{-1}.

Table 5.3: Constants used in the calculation of the specific heat capacity of UO_2

Constant	Value
R (cal mol^{-1} K^{-1})	1.987
θ (K)	535.285
E_D (cal mol^{-1})	37694.6
K_1 (cal mol^{-1} K^{-1})	19.145
K_2 (cal mol^{-1} K^{-2})	7.8473×10^{-4}
K_3 (cal mol^{-1})	5.6437×10^{6}

5.4.2 Properties of the Zircaloy-4 Sheath

Zircaloy-4 is used for the cladding material in CANDU reactor fuel elements. The correlation used for the thermal conductivity of Zircaloy is given by MATPRO [51]

$$k = 7.51 + 2.09 \times 10^{-2} T - 1.45 \times 10^{-5} T^2 + 7.67 \times 10^{-9} T^3 \tag{5.35}$$

where T is the temperature in K. Due to the manufacturing process of Zircaloy-4 (e.g., cold working), the crystal structure of the Zircaloy becomes oriented differently in different coordinate directions producing an anisotropic material. This results in the sheath expanding differently in the axial and radial directions. Using the empirical correlations suggested by MATPRO-9 [55], the thermal expansion in the axial direction is

$$\frac{\Delta L}{L_0} = -2.506 \times 10^{-5} + 4.441 \times 10^{-6}(T - 273)$$

$$\frac{\Delta L}{L_0} = 0.0120387 - 1.06387 \times 10^{-5}(T - 273) \tag{5.36}$$

$$\frac{\Delta L}{L_0} = -8.3 \times 10^{-3} - 9.7 \times 10^{-6}(T - 273)$$

for the alpha (300 K < $T \leq$ 1073 K), transition (1073 K < $T \leq$ 1273 K), and beta (1273 K < $T \leq$ 2098 K) phases, respectively, where ΔL is the change in the length due to thermal expansion, and L_0 is the length at 300 K. Similarly, thermal expansion in the radial direction is determined by

$$\frac{\Delta D}{D_0} = -2.373 \times 10^{-4} + 6.721 \times 10^{-6}(T - 273)$$

$$\frac{\Delta D}{D_0} = 0.0140975 - 1.11975 \times 10^{-6}(T - 273) \tag{5.37}$$

$$\frac{\Delta D}{D_0} = -6.8 \times 10^{-3} + 9.7 \times 10^{-6}(T - 273)$$

for the alpha, transition and beta phases, respectively, where ΔD is the change in diameter due to thermal expansion, and D_0 is the diameter at 300 K.

The thermal expansion of the sheath affects its density over time. In ELK the generic density function determines the density change due to thermal expansion processes given an initial density of the material. The initial density of the sheath supplied to this function is considered constant at its published value of 6.551×10^3 kg m^{-3} [56]. The specific heat capacity of the sheath in units of J kg^{-1} K^{-1} is

$$C_p = \frac{6.55 \times 10^6 (1.1061 \times 10^{-4} T + 0.2575)}{\rho}$$

$$C_p = \frac{2.3118 \times 10^6}{\rho} \tag{5.38}$$

for temperatures less than 1115 K, and for temperatures greater than or equal to 1115 K respectively. This correlation produces results similar to the correlation used in MATPRO-9 [55] and is equivalent to what is used in ELOCA.

5.4.3 Properties of the Fuel-to-Sheath Gap

Correctly determining the properties of the fuel-to-sheath gap is one of the most important aspects of nuclear fuel modeling as explained Chapter 2. An incorrect heat transfer coefficient across the gap will yield incorrect fuel temperatures. Improper values for the gas temperature and pressure will result in incorrect values for the interfacial pressure between the outside surface of the fuel pellets and the inside surface of the sheath for contact analyses. Furthermore, the heat transfer coefficient depends upon this interfacial pressure. To model the fuel-to-sheath gap HORSE employs a modified form of the Ross and Stoute [57] correlation that includes radiation heat transfer. This model has been used by a variety of previous models and codes including those developed by Morgan [19], Shaheen [23], ELESTRES and ELOCA. The heat transfer correlation of the gap is a sum of three separate components,

$$h_{gap} = h_s + h_f + h_r \tag{5.39}$$

where h_s is the solid-to-solid conductance between the fuel and sheath due to contact, h_f is the gas conductance due to the fill gas/fission gas mixture in the gap and plenum regions, and h_r is radiative conductance due to radiation heat transfer between the fuel and the sheath.

5.4.3.1 Solid-to-Solid Conductance

The solid-to-solid conductance component is provided by

$$h_s = \frac{k}{a_0\sqrt{R}}\frac{\sqrt{P}}{H} \tag{5.40}$$

where P is the interfacial pressure between the exterior of the pellets and the interior of the sheath in MPa, k is the harmonic mean of the fuel and sheath conductivities in kW m^{-1} K^{-1}, R is the root-mean squared surface roughness in m, $H = 4.4 \times (297.39 - 0.2733\overline{T_s})$ is the Meyer hardness of the sheath as a function of the average sheath temperature in MPa, and a_0 is a constant equal to 8.6×10^{-3} $m^{0.5}$ $MPa^{-0.5}$. The harmonic mean of the conductivities is calculated via

$$k = \frac{2k_f k_s}{k_f + k_s} \tag{5.41}$$

and the root-mean squared surface roughness is given by

$$R = \sqrt{\frac{R_f^2 + R_s^2}{2}} \tag{5.42}$$

where k_f and k_s are the thermal conductivities of the fuel and sheath, respectively, and $R_f = R_s = 1 \times 10^{-6}$ m are the assumed surface roughnesses of the fuel and sheath. The solid-to-solid conductance term is zero when the pellets are not in contact with the sheath.

5.4.3.2 Gas Conductance

The gas conductance component of the fuel-to-sheath gap is the most difficult to calculate as it depends upon the gas composition inside the gap. Usually this is coupled to a fission gas release model to have a constantly updating gas mixture in the gap during irradiation. HORSE does not contain a fission gas production and release model, however it does allow for the input of the amount of fission gas added at each timestep. This capability allows the fission gas release data from, say, ELESTRES to be added as an input to HORSE for a more accurate comparison

between the two codes. The mole fraction y_i seen in some of the following equations is calculated for the three gases, helium, argon, and fission gas, such that the sum of all the mole fractions is equal to one at all timesteps within a simulation. The thermal conductance of the gas mixture is

$$h_f = \frac{k_g}{C\left(R_f + R_s\right) + g_w + g} \tag{5.43}$$

where $k_g$is the conductivity of the gas mixture in kW K^{-1}, C is a unitless constant, g_w is the gap width in m, and g is the temperature jump distance. The thermal conductivity of the gas can be represented as a function of temperature and pressure as

$$k(T,P) = X(T)Y(P,T) \tag{5.44}$$

The primary temperature variation $X(T)$ can be obtained by using a power law developed by Von Ubisch et al. [58] for each individual gas before computing the overall thermal conductivity [59]:

$$X(T) = \frac{\sum_i y_i X_i(T)(M_i)^{1/3}}{\sum_i y_i (M_i)^{1/3}} \tag{5.45}$$

where $X_i(T) = k_0 T_i^s$, and y_i and $M_i$are the mole fraction and atomic weight of the i^{th} gas. A correction factor introduced due to the high pressure environment within the pressure tube was proposed by Lenoir [60]:

$$Y(P,T) = 1 + (0.51T_R^{-2.26})\left(P_R^{1+2.5T_R^{-2.26}}\right) \tag{5.46}$$

where T_R and P_R are the reduced temperature and pressure, respectively. These terms are determined by

$$T_R = \frac{T_g}{\sum_i y_i T_{ci}} \tag{5.47}$$

$$P_R = \frac{P_g}{\sum_i y_i P_{ci}} \tag{5.48}$$

where P_g and T_g are the pressure and temperature of the gas within the fuel-to-sheath gap. The constant C in equation (5.43) was measured by Ross and Stoute and it was found to vary depending on whether interfacial pressure was high or low (1.5 - 2.5). Since, the irradiation environment is at high pressure, C is taken as 1.5. The jump distance in equation (5.43) is given by

$$g = \frac{1}{\sum_i \left(\frac{y_i}{g_{0i} \left(\frac{T_g}{273.15} \right)^{s+1/2}} \right)} \times \frac{0.101325}{P_g} \tag{5.49}$$

where g_{0i} is the temperature jump distance of gas i at standard temperature and pressure. The data required to solve the equations needed to determine the gas conductance component of the fuel-to-sheath gap heat transfer coefficient is provided in Table 5.4.

Table 5.4: Gas properties for gas conductance of fill gas and fission gas mixtures

Gas	M (g)	k_0 (W m^{-1} K^{-1})	s	T_c (K)	P_c (MPa)	g_0 (μm)
Helium	4.00	2.504×10^{-3}	0.721	5.2	0.229	5.2
Argon	39.95	2.217×10^{-4}	0.772	151.0	4.86	0.57
Fission Gas	127.0	4.627×10^{-5}	0.856	281.0	5.86	0.26

5.4.3.3 Radiation Conductance

The radiative heat transfer coefficient is given by the standard relationship for two infinitely long concentric cylinders:

$$h_r = \sigma \frac{1}{\left(\frac{1}{\varepsilon_s} + \frac{1}{\varepsilon_f} - 1 \right)} \left(T_f^2 + T_s^2 \right) \left(T_f + T_s \right) \tag{5.50}$$

where T_f and T_s are the fuel and sheath temperatures, respectively, σ is the Stefan-Boltzmann constant, $\varepsilon_s = 0.325$, is the emissivity of the intact fuel sheath, and ε_f is the emissivity of the fuel given by

$$\varepsilon_f = 0.7856 + 1.5263 \times 10^{-5} T_{fsurf} \tag{5.51}$$

where T_{fsurf} is the temperature of the fuel surface, as suggested by MATPRO [51].

5.4.3.4 Gas Temperature and Pressure

The gas temperature and pressure is required in some of the equations used to determine the gas conductance component of the heat transfer coefficient. The gas temperature is taken as the average of the temperature between the two bounding surfaces (i.e., the inside of the sheath and outside of the fuel). The volume is calculated as the free volume within the element including the dishes on the pellets, the plenum region, and the gap between the exterior surface of the pellets and the interior surface of the sheath when the gap is open. The number of moles of gas within the free volume changes during the simulation. Both the temperature and volume vary over time and once these parameters have been determined the internal pressure within the gap is found using the ideal gas law

$$P_g = \frac{n R_{gas} T_g}{V_g} \tag{5.52}$$

where n is the number of moles of gas in the element, V_g is the volume of the cavity within the fuel element and R_{gas} is the ideal gas constant.

5.4.4 Properties of the Coolant

The thermalhydraulic behaviour in nuclear fuel sub-channels is a difficult and complex field of study and is beyond the scope of this work. However, certain assumptions of the coolant conditions are required for boundary condition purposes to facilitate convergent solutions. To sufficiently define the simulations in this work the coolant pressure, temperature and heat transfer coefficient are required. The coolant pressure is selected to be 10 MPa, which is the pressure within the pressure tubes in a CANDU reactor, the bulk temperature of the coolant is taken as 583 K, and the heat transfer coefficient is chosen to be 50 kW m^{-2} K^{-1}, which is a nominal value for heavy water. These coolant parameters are supplied by the user and can be modified.

5.5 Deformation

Accurately determining the deformation of the fuel and sheath provides a complete understanding of the mechanical state of the fuel. The deformation is presented in terms of the stresses, strains and displacements. The mechanical behaviour of the fuel and sheath takes into account elastic and plastic effects. In the models presented in this work, the deformation takes into account the external loading on the sheath (due to the coolant and constraints), the thermal expansion of the fuel and the sheath, plasticity of the sheath, creep of the sheath, and densification and fission product swelling of the fuel. The displacement state vector $\mathbf{u}$ is customarily defined as $\mathbf{u} = \{u, v, w\}$ to avoid confusion with the coordinate directions in the Cartesian system.

5.5.1 Fuel Deformation

Stresses and strains are calculated in a tensor format. The stress tensor being solved in HORSE is given by the Cauchy stress tensor in equation (5.13) where σ_{xx}, σ_{yy}, and σ_{zz} are the normal stresses and σ_{xy}, σ_{xz}, σ_{yz}, σ_{yx}, σ_{zx}, and σ_{zy} are the shear stresses. The first index in the subscript indicates that the stress acts on a plane normal to that axis, with the second index indicating the direction in which the stress acts. By definition

$$\sigma_{xy} = \sigma_{yx};\ \sigma_{xz} = \sigma_{zx};\ \sigma_{yz} = \sigma_{zy} \tag{5.53}$$

which indicates that there are only six unique elastic stresses. The normal stresses are also known as the principal stresses. The stresses are determined from the strains (ε_{ij}) within the material through constitutive relations, where i and j are defined as the row and column indices in the stress tensor. In HORSE the total strain within a material is defined as the summation of the individual components of strain. The fuel experiences strains from elastic behaviour (ε_{el}), thermal expansion (ε_{th}), gaseous fission product swelling (ε_{GFP}), solid fission product swelling (ε_{SFP}), and densification (ε_{dens}):

$$\varepsilon_f = \varepsilon_{el} + \varepsilon_{th} + \varepsilon_{GFP} + \varepsilon_{SFP} + \varepsilon_{dens} \tag{5.54}$$

The elastic component of the total strain is determined using Hooke's Law. Hooke's Law can be expressed in a symmetric tensor for isotropic materials as follows

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{bmatrix} = \frac{E}{(1+\upsilon)(1-2\upsilon)} \begin{bmatrix} 1-\upsilon & \upsilon & \upsilon & 0 & 0 & 0 \\ \upsilon & 1-\upsilon & \upsilon & 0 & 0 & 0 \\ \upsilon & \upsilon & 1-\upsilon & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\upsilon}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\upsilon}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\upsilon}{2} \end{bmatrix} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ 2\varepsilon_{xy} \\ 2\varepsilon_{yz} \\ 2\varepsilon_{xz} \end{bmatrix} \tag{5.55}$$

where E and υ are the Young's modulus and Poisson's ratio of the material, respectively. The Young's modulus of the fuel depends upon the temperature and percentage of theoretical density of stoichiometric UO_2 as suggested by MATPRO [51] and the Poisson's ratio of the fuel is given by MATPRO [51], which was used by Morgan [19] and Bell et al. [26]

$$E_f = 2.334 \times 10^{11}\big(1 - 2.752(1 - D)\big)(1 - 1.0915 \times 10^{-4}T) \tag{5.56}$$

$$\upsilon = 0.316 \tag{5.57}$$

where D is the percentage of theoretical density. Equation (5.55) can be simplified by introducing Lamé's first (λ) and second (μ) parameters,

$$\begin{bmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \sigma_{zz}\\ \sigma_{xy}\\ \sigma_{yz}\\ \sigma_{xz}\end{bmatrix} = \begin{bmatrix}2\mu+\lambda & \lambda & \lambda & 0 & 0 & 0\\ \lambda & 2\mu+\lambda & \lambda & 0 & 0 & 0\\ \lambda & \lambda & 2\mu+\lambda & 0 & 0 & 0\\ 0 & 0 & 0 & \mu & 0 & 0\\ 0 & 0 & 0 & 0 & \mu & 0\\ 0 & 0 & 0 & 0 & 0 & \mu\end{bmatrix}\begin{bmatrix}\varepsilon_{xx}\\ \varepsilon_{yy}\\ \varepsilon_{zz}\\ 2\varepsilon_{xy}\\ 2\varepsilon_{yz}\\ 2\varepsilon_{xz}\end{bmatrix} \tag{5.58}$$

where Lamé's parameters are defined by

$$\mu = G = \frac{E}{2(1+\upsilon)} \tag{5.59}$$

$$\lambda = \frac{\upsilon E}{(1+\upsilon)(1-2\upsilon)} \tag{5.60}$$

and G is the shear modulus. By expanding equation (5.58) the individual stress components are calculated via

$$\begin{aligned}
\sigma_{xx} &= (2\mu + \lambda)\varepsilon_{xx} + \lambda(\varepsilon_{yy} + \varepsilon_{zz}) \\
\sigma_{yy} &= (2\mu + \lambda)\varepsilon_{yy} + \lambda(\varepsilon_{xx} + \varepsilon_{zz}) \\
\sigma_{zz} &= (2\mu + \lambda)\varepsilon_{zz} + \lambda(\varepsilon_{xx} + \varepsilon_{yy}) \\
\sigma_{xy} &= \sigma_{yx} = 2\mu\varepsilon_{xy} \\
\sigma_{yz} &= \sigma_{zy} = 2\mu\varepsilon_{yz} \\
\sigma_{xz} &= \sigma_{zx} = 2\mu\varepsilon_{xz}
\end{aligned} \tag{5.61}$$

where the individual elastic strain components can be determined from the displacements of the material by the kinematic equations [19]:

$$\begin{aligned}
\frac{\partial u}{\partial x} = \varepsilon_{xx};\ \frac{\partial v}{\partial y} = \varepsilon_{yy};\ \frac{\partial w}{\partial z} = \varepsilon_{zz} \\
\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = \varepsilon_{xy};\ \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} = \varepsilon_{yz};\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} = \varepsilon_{zx}
\end{aligned} \tag{5.62}$$

Once the elastic stresses are determined, they are used to satisfy Cauchy's equation for static equilibrium as given by equation (5.13). By substituting the stresses in Cauchy's equation, momentum conservation becomes

$$\begin{aligned}
(2\mu + \lambda)\frac{\partial \varepsilon_{xx}}{\partial x} + \lambda\left(\frac{\partial \varepsilon_{yy}}{\partial x} + \frac{\partial \varepsilon_{zz}}{\partial x}\right) + 2\mu\left(\frac{\partial \varepsilon_{xy}}{\partial y} + \frac{\partial \varepsilon_{xz}}{\partial z}\right) + \rho f_x = 0 \\
(2\mu + \lambda)\frac{\partial \varepsilon_{yy}}{\partial y} + \lambda\left(\frac{\partial \varepsilon_{xx}}{\partial y} + \frac{\partial \varepsilon_{zz}}{\partial y}\right) + 2\mu\left(\frac{\partial \varepsilon_{xy}}{\partial x} + \frac{\partial \varepsilon_{yz}}{\partial z}\right) + \rho f_y = 0 \\
(2\mu + \lambda)\frac{\partial \varepsilon_{zz}}{\partial z} + \lambda\left(\frac{\partial \varepsilon_{xx}}{\partial z} + \frac{\partial \varepsilon_{yy}}{\partial z}\right) + 2\mu\left(\frac{\partial \varepsilon_{xz}}{\partial x} + \frac{\partial \varepsilon_{yz}}{\partial y}\right) + \rho f_z = 0
\end{aligned} \tag{5.63}$$

where f_i is the body force in that coordinate direction. Finally, by using the kinematic relations given by equation (5.62), the equation for static equilibrium in terms of the displacement vector **u** becomes

$$(2\mu+\lambda)\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 v}{\partial x\partial y}+\frac{\partial^2 w}{\partial x\partial z}\right)+2\mu\left(\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right)+\rho f_x=0$$

$$(2\mu+\lambda)\left(\frac{\partial^2 v}{\partial v^2}+\frac{\partial^2 u}{\partial y\partial x}+\frac{\partial^2 w}{\partial y\partial z}\right)+2\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial z^2}\right)+\rho f_y=0 \tag{5.64}$$

$$(2\mu+\lambda)\left(\frac{\partial^2 w}{\partial z^2}+\frac{\partial^2 u}{\partial z\partial x}+\frac{\partial^2 y}{\partial z\partial y}\right)+2\mu\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 x}{\partial y^2}\right)+\rho f_z=0$$

The component of the total strain due to thermal expansion is given by equation (5.25) and the strain components due to gaseous and solid fission product swelling are related to the volume change given by equations (5.30) and (5.31), respectively. Since the fuel is isotropic the linear strain due to fuel swelling is

$$\varepsilon_j \cong \frac{1}{3}\frac{\Delta V_j}{V_0} \tag{5.65}$$

where j denotes the strain component of interest. The volumetric change of the fuel due to densification is given by Hastings [52] as

$$\frac{\Delta V_{dens}}{V_0}=\frac{1-p_0}{1-p_0(1-F)}-1 \tag{5.66}$$

where p_0 is the initial porosity of the fuel and the strain is subsequently determined by equation (5.65) and F is given by equation (5.29).

5.5.2 Sheath Deformation

Similarly to the fuel, the total strain in the sheath is determined by the sum of the individual components. The sheath contains strain contributions due to elastic behaviour (ε_{el}), thermal expansion (ε_{th}), plasticity (ε_{plas}), and creep (ε_{creep}):

$$\varepsilon_{sheath}=\varepsilon_{el}+\varepsilon_{th}+\varepsilon_{plas}+\varepsilon_{creep} \tag{5.67}$$

The elastic strain component is computed in the same manner as for uranium dioxide where the Young's modulus and Poisson's ratio of Zircaloy-4 are adopted from MATPRO-9 [55] and Bell et al. [26], respectively. The Young's modulus is defined as

$$E_s = 1.128 \times 10^{11} - 5.99 \times 10^{7} T$$
$$E_s = 1.005 \times 10^{11} - 4.725 \times 10^{7} T \quad (5.68)$$
$$E_s = 3.30 \times 10^{8}$$

for (T < 1335 K), (1135 K ≤ T < 2120 K), and (T ≥ 2120 K), respectively. The effect of sheath melting on the modulus of elasticity is not taken into account, as the sheath temperature is kept below the melting temperature for all simulations completed in this work. The Poisson's ratio used is assumed to be isotropic for simplicity[2]:

$$\upsilon = 0.3 \quad (5.69)$$

Due to the cylindrical nature and thin wall of the sheath it is common to express the stresses and strains within the sheath in terms of the cylindrical stresses: axial, radial, and hoop. An effective stress known as the Von Mises stress is useful for determining when the sheath begins yielding due to plasticity and creep. For illustration purposes the derivation of these representations will be done using the Cauchy stress tensor. To determine the corresponding axial, radial, hoop, and Von Mises strains one needs to simply substitute the equivalent element of the strain tensor into the equations. Based upon Von Mises' observations it was determined that even if none of the normal or shear stresses exceeded the yield stress of a material, the material could still yield. Therefore, Von Mises developed an equivalent stress that combines the normal and shear stresses called the Von Mises stress. It is the Von Mises stress that is compared against the yield stress of

[2] It is known that the sheathing is anisotropic due to the manufacturing process but is taken as isotropic for simplicity.

a material to determine if yielding occurs. The formula for the Von Mises stress in terms of the components of the Cauchy stress tensor is

$$\sigma_{VM} = \left(0.5\left[\left(\sigma_{xx} - \sigma_{yy}\right)^2 + \left(\sigma_{yy} - \sigma_{zz}\right)^2 + \left(\sigma_{xx} - \sigma_{zz}\right)^2 + 6\left(\sigma_{xy}{}^2 + \sigma_{xz}{}^2 + \sigma_{yz}{}^2\right)\right]\right)^{\frac{1}{2}} \tag{5.70}$$

To obtain the hoop, radial and axial stresses a coordinate transformation from Cartesian to cylindrical coordinates is required. This coordinate transformation is achieved by defining the axial axis (in this case the z-direction) and determining the required hoop (azimuthal) and radial directions for any point defined by (x, y, z). Therefore, for an arbitrary point in the mesh the hoop, radial and axial stresses are calculated from the Cauchy stress tensor components via

$$\sigma_{hoop} = \left(\frac{y^2}{x^2 + y^2}\right)^2 \sigma_{xx} + \left(\frac{x^2}{x^2 + y^2}\right)^2 \sigma_{yy} + 2\left(\frac{y^2}{x^2 + y^2}\right)\left(\frac{x^2}{x^2 + y^2}\right)\sigma_{xy} \tag{5.71}$$

$$\sigma_{radial} = \left(\frac{x^2}{x^2 + y^2}\right)^2 \sigma_{xx} + \left(\frac{y^2}{x^2 + y^2}\right)^2 \sigma_{yy} + 2\left(\frac{y^2}{x^2 + y^2}\right)\left(\frac{x^2}{x^2 + y^2}\right)\sigma_{xy} \tag{5.72}$$

$$\sigma_{axial} = \sigma_{zz} \tag{5.73}$$

respectively. Due to the anisotropic nature of Zircaloy the thermal expansion strain contains two components, diametrical and axial. The strain components depend upon the temperature of the sheath and are given by equations (5.36) and (5.37). The plasticity strain component is calculated using a bilinear plasticity model derived by Williams at AECL-CRL from the MATPRO [51] library. To generate the bilinear family of curves used in this model, the Young's modulus, yield stress and tangent modulus of the sheath is required. The tangent modulus defines the slope of the plastic strain hardening curve that is followed once the yield stress is exceeded. The yield stress is determined by

$$\sigma_y = \left(\frac{K}{E^n}\right)^{\left(\frac{1}{1-n}\right)} \tag{5.74}$$

where K is the strength coefficient, n is the strain hardening exponent, and E is calculated by equation (5.68). The tangent modulus is defined as the gradient between the yield stress and the ultimate tensile strength of the sheath:

$$TM = \frac{\sigma_{UT} - \sigma_y}{\varepsilon_{UT} - \varepsilon_y} \tag{5.75}$$

Therefore the ultimate tensile stress (σ_{UT}), strain at yielding (ε_y), and strain at tensile failure (ε_{UT}) are required:

$$\sigma_{UT} = K\left(\frac{n}{1+m}\right)^n \tag{5.76}$$

$$\varepsilon_y = \left(\frac{K}{E}\right)^{\left(\frac{1}{1-n}\right)} \tag{5.77}$$

$$\varepsilon_{UT} = \frac{n}{1+m} \tag{5.78}$$

where m is the strain rate sensitivity exponent. Figure 5.1 presents a plot of the bilinear plasticity curve for a variety of selected temperatures. The bilinear curve can be determined for temperatures ranging from room temperature to sheath melting because E, K, n, and m are functions of temperature which allows the yield stress and tangent modulus to be determined for any temperature. Appendix C contains the data points used to generate this model.

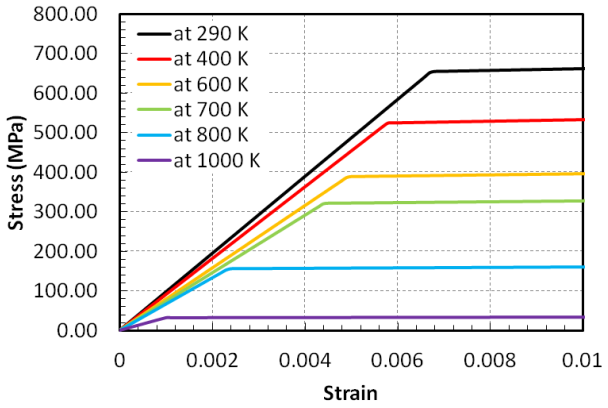


Figure 5.1: Example set of curves used for the bilinear plasticity model of Zircaloy-4.

Material creep occurs due to microstructural changes at high and low temperatures. Sills and Holt [61] developed a high temperature creep model that takes into account three creep mechanisms: diffusional creep, dislocation creep and transition strain. All three creep mechanisms are employed in the ELESTRES and ELOCA ISTs. In this work only diffusional creep is modeled because of its contribution to fuel element bowing and sagging. Diffusional creep examines the effect of grain boundary sliding on the sheath deformation. The strain rate due to diffusional creep at temperatures above 923 K is given by

$$\dot{\varepsilon}_{gb} = F\left(\frac{\sigma_a}{d}\right)^m e^{-Q/T} \tag{5.79}$$

where σ_a is the applied stress in MPa, d is the grain size in μm (3 for α-phase, 100 for β-phase), m is an exponent (2.0 for α-phase, 1.9 for β-phase), Q is the activation energy divided by the universal gas constant in K^{-1} (9431 for α-phase, 6039 for β-phase), and F is the creep rate coefficient defined by

$$F = \frac{6.34 \times 10^6}{G^2} \tag{5.80}$$

where G is the shear modulus. For the transition phase, d, m, and Q are linearly interpolated between the α-phase and β-phase values. The creep strain component of the total strain can be determined by integrating equation (5.79). For temperatures less than 923 K the creep rate due to diffusional creep is given by

$$\dot{\varepsilon}_{gb} = \frac{1.41 \times 10^{10}}{Td^2}{\sigma_a}^2 e^{-29500/T} \tag{5.81}$$

Chapter 6 - HORSE Implementation

The material property correlations and nuclear phenomena equations are a portion of what is required to produce results for analysis and comparison against other codes, models, and experiments. To solve these equations, the geometry of interest, the finite element mesh used to represent that geometry, and the input file defining the simulation and solver parameters are required. Three different geometric models were created and analyzed in this work: 2D and 3D contact analysis models and a full fuel element containing natural uranium dioxide fuel (like in CANDU fuel bundles). The full fuel element is used for fuel performance, flexural rigidity and thermal bowing analyses. This section outlines the details of the geometries, meshes, boundary conditions, constraints, solver options and the input file setup used in this thesis.

Geometries and meshes are generated in a third party commercial program called Trelis provided by csimsoft (Computational Simulation Software). The cost of the Trelis license is inexpensive as opposed to the commercial codes ANSYS and Comsol. For U.S. DOE employees Trelis can be obtained for free under the name Cubit from Sandia Laboratories. Trelis version 14.0.3 FEA was used in this work. The geometry and mesh created is output in an ExodusII file format (.e extension), and is referenced in the MOOSE input file. A sample Trelis journal file used to generate the full fuel element geometry and mesh is provided in Appendix D. When the HORSE application is compiled and generated, it inherits all the data types and methods in the applications lower in the hierarchy (i.e., MOOSE and ELK). Thus, when the input file is passed into HORSE, it is the MOOSE component that reads the mesh file and all associated materials, boundary conditions, contact definitions, and solver settings that are defined. An in-depth look into blocks that make up an input file is provided in Section 6.4. A sample HORSE input file is

provided in Appendix E. Once the simulation is complete, an output file containing the solutions of all the nodal and elemental variables at each time step as well as the postprocessor values requested by the user is produced. Examples of postprocessor values are gas pressure, average sheath surface temperature, and time step size. The output file is then opened in an open source postprocessing program called ParaView. ParaView has extensive data analysis and visualization capabilities, as well as providing the option to output data into comma separated value files (.csv extension) for analysis in more familiar programs like Microsoft Excel. Figure 6.1 presents a flowchart that illustrates the solution process.

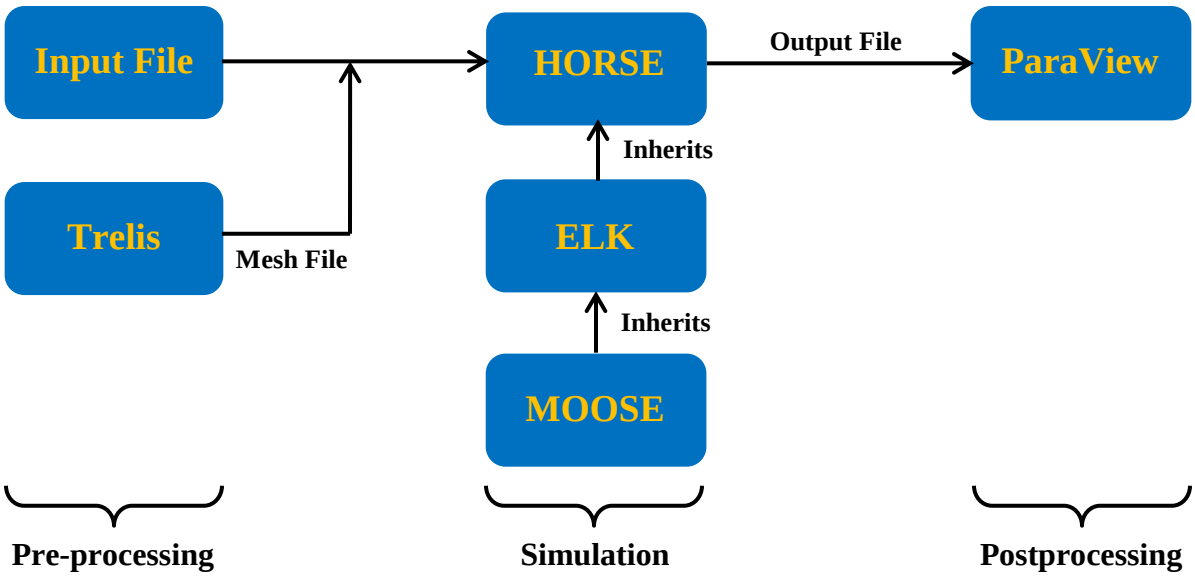


Figure 6.1: Flowchart outlining the process of mesh generation to input file creation, simulation and postprocessing

6.1 Contact Analysis

To accurately model 3D deformation of horizontal fuel elements using finite element analysis, contact is of great importance. Contact exists between individual pellets as well as the exterior of the pellets with the interior of the sheath. Since the MOOSE framework is still in its infancy, the behaviour and capabilities of the contact algorithms are not well understood. At the time of writing, this work is the first and only known use of the MOOSE framework in Canada. To gain an understanding of the behaviour of the frictionless, glued, and coulomb friction models available in MOOSE, two simplified models were developed. One model is two-dimensional and the other is three-dimensional to examine the dimensional variability in the contact models. The models are considered to be simplified as they contain constant sheath and fuel material properties, and constant uniform heat generation. In the three-dimensional case the fuel does not contain dishes or chamfers. These simplifications are made to isolate the convergence behaviour of the contact models from any convergence difficulties introduced by the nonlinearities associated with the material models, heat generation and flux depression, fuel features, and nuclear phenomena (e.g., fuel swelling).

The two-dimensional analysis considers a radial cross-section of a fuel pellet, fuel-to-sheath gap and sheath at one particular axial location. The assumption in a 2D analysis like this is that the fuel and sheath are infinitely long (i.e., plane strain approximation). Figure 6.2 (a) and (b) present the geometry and finite element mesh used in the 2D analysis, respectively. The pellet and sheath nominal dimensions are indicated. The mesh used contained 1360 elements, which yields 1481 nodes in the 1^{st} order case and 4321 nodes in the 2^{nd} order case. The coordinate directions are included.

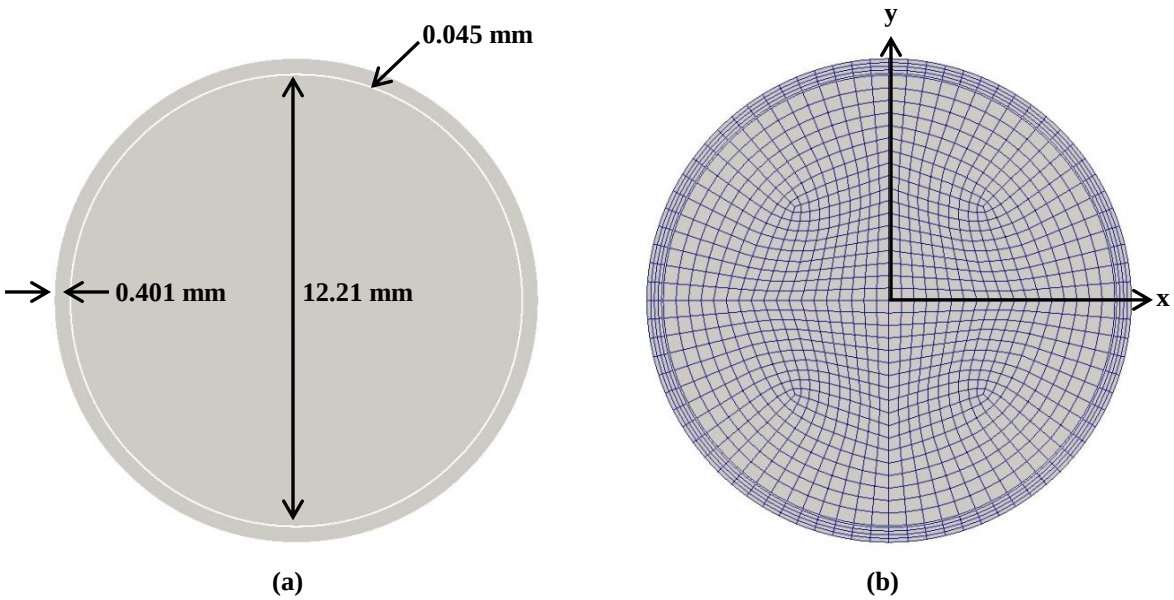


Figure 6.2: (a) Geometry and (b) finite element mesh used for the 2D analysis of the contact algorithms.

In Trelis, boundary conditions and constraints are applied to what are called nodesets and sidesets. Nodesets apply to vertices or nodes, whereas sidesets apply to surfaces and curves. If possible vertices should be chosen for nodesets. This is because if the mesh density changes, the node numbering would change causing nodesets defined by the nodes to be different. Therefore, the nodesets may need to be redefined to correctly define the desired boundary condition or constraint. Material properties and kernels (e.g., heat generation) are applied to blocks. In this contact analysis two blocks are defined: the sheath and the fuel. In order to constrain the system sufficiently, nodes on the x-axis are fixed such that they can only move in the horizontal direction and nodes on the y-axis are fixed such that they can only move in the vertical direction. These conditions actually create two lines of reflectional symmetry indicating that a quarter model could have been used. However, for this two-dimensional analysis the difference in required computing resources is inconsequential between the quarter and full models. Therefore the full model was run. The above constraints prevent unconstrained translation and rotation.

The initial temperature conditions and the coolant properties used as boundary conditions are summarized in Table 6.1. The coolant properties are the same as presented in Section 0but are reproduced here for ease of reference alongside the initial temperature condition of the system.

Table 6.1: Initial system temperature and coolant properties used in contact analysis models

Property	Value
Initial System Temperature (K)	300
Coolant Pressure (MPa)	10
Bulk Coolant Temperature (K)	583
Coolant Heat Transfer Coefficient (W m^{-2} K^{-1})	50 000

As explained earlier, the material properties and heat generation used in the contact analyses are simplified from those introduced in Section 5.4 to isolate the convergence of the contact algorithms from the complications that may arise from the additional nonlinearities introduced by temperature and burnup dependent material properties and non-uniform heat generation. The properties used in the 2D contact analysis are constant and are summarized in Table 6.2. The linear power rating given in the table is converted into a volumetric heat generation, in units of W m^{-3}.

Table 6.2: Material properties and linear power used in the 2D and 3D contact analyses

Property	Fuel	Sheath
Density (kg m^{-3})	10650.0	6551.0
Young's Modulus (GPa)	180	70
Poisson's Ratio	0.3	0.3
Coefficient of Thermal Expansion (K^{-1})	13.9×10^{-6}	5.7×10^{-6}
Thermal Conductivity (W m^{-1} K^{-1})	3	16
Specific Heat (J m^{-3} K^{-1})	320	330
Linear Power (kW m^{-1})	45	-

In the 3D contact analysis the same material properties, initial conditions and coolant conditions are used as presented in tables Table 6.1 and Table 6.2. The main differences between the 2D and 3D analyses are the geometry and constraint conditions. The fuel and sheath dimensions remain the same as in 2D, except they are extended in the axial direction with a finite distance. In three-dimensions three different mesh densities were used to analyze the effect of mesh density on the contact behaviour. As in the 2D case, first and second order meshes are investigated. Figure 6.3 displays the axial dimensions of the fuel and sheath, and the three meshes used for the mesh density analysis, which are denoted as original (b), half (c), and quarter mesh (d) respectively. The original, half, and quarter meshes contained 4128, 2499, and 1537 hexahedral elements. This produces first order meshes containing 5443, 3456, and 2212 nodes, and second order meshes of 20358, 12783, and 8103 nodes.

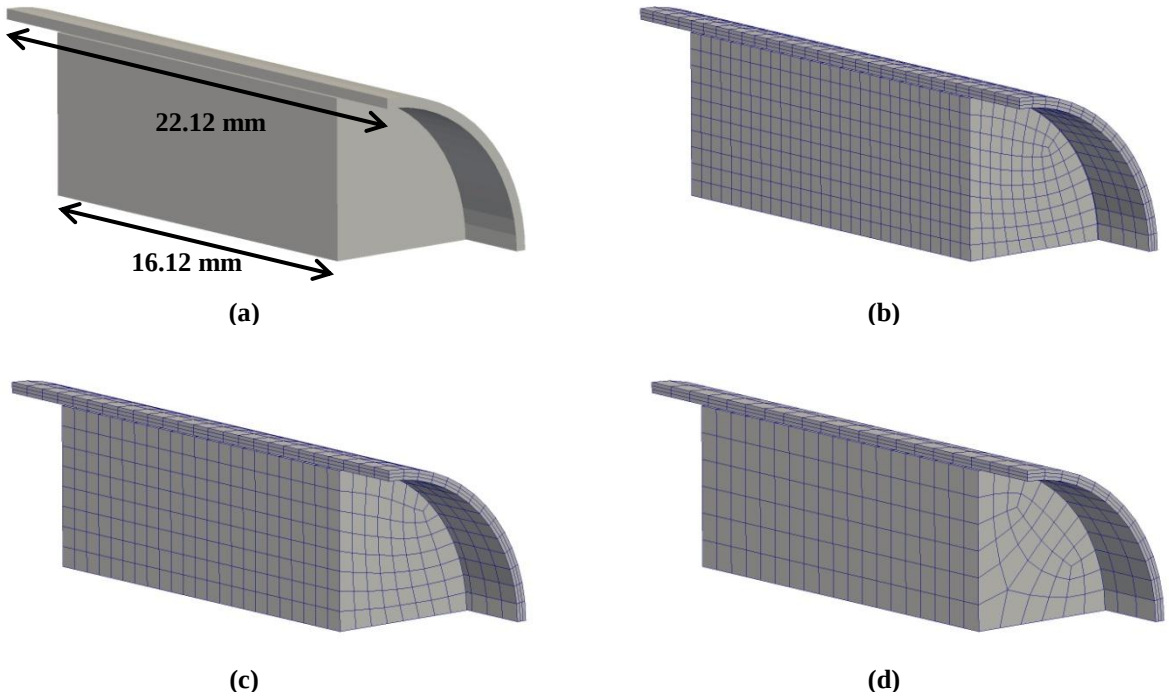


Figure 6.3: Illustrations of the (a) geometry, (b) original mesh, (c) half mesh and (d) quarter mesh used in the 3D contact analysis.

To obtain convergent solutions both the pellet and sheath must be constrained against rigid body motion. The best way to illustrate the boundary conditions of the 3D contact analysis is through Figure 6.4. All nodes on the y-z plane (magenta) are fixed such that they cannot move normal to the plane, whereas all nodes on the x-z plane (red) are fixed such that they cannot move normal to the plane. The front of the sheath, and corner node of the fuel pellet, denoted by the colour yellow, are fixed such that they cannot move in the axial (z) direction. These constraints are sufficient to prevent rigid body rotation and translation of the fuel and sheath system.

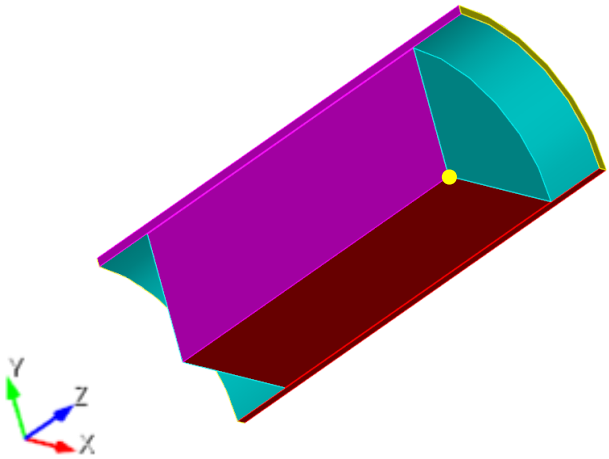


Figure 6.4: Illustration of the boundary conditions used in the 3D analysis to eliminate rigid body translation and rotation of the system. Translation in y-z plane (magenta) and translation in x-z plane (red) are allowed. No axial translation allowed (yellow).

6.2 Full Fuel Element

Upon completion of the contact analysis the model was incrementally increased from a single fuel pellet up to a full fuel element containing thirty-one pellets. In each increment two fuel pellets were added. Once there are multiple pellets the ability to obtain convergent solutions becomes significantly more difficult due to the presence of multiple contact pairs (i.e., pellet-to-pellet and pellet-to-sheath). To be able to predict fuel performance as well as fuel element

deformation in a reasonable amount of time, symmetry can be used. For this model a quarter of a full fuel element is modeled by taking two planes of symmetry, one axially along the element (i.e., the y-z plane), and one at the midplane of the element (i.e., the x-y plane). Figure 6.5 shows a diagram of the quarter element indicating the orientation on the Cartesian axis as well as the planes of symmetry. The numbers on the pellets indicate the pellet's 'number' used for identification purposes and to keep track of the pellet-to-pellet contact pairs.

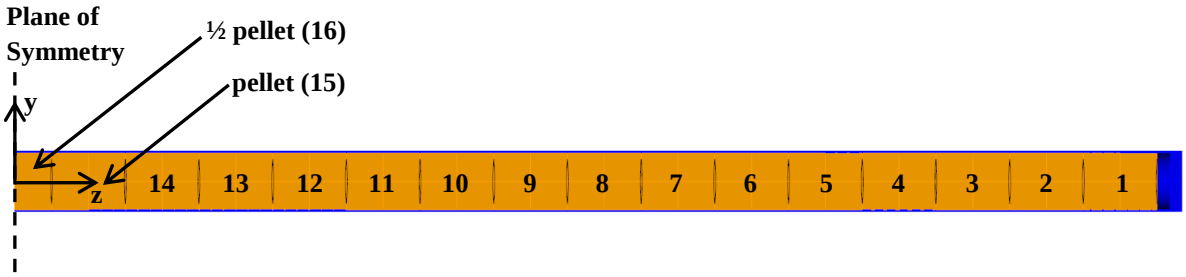


Figure 6.5: Fuel element orientation, planes of symmetry and pellet identification

In contrast to the 3D contact analysis model that had no end caps, the full fuel element model contains simplified representations of the fuel element end caps, which are considered to be right cylinders that are permanently welded to the sheath. For simplicity there is no axial gap in between the individual pellets, which helps with solution stability as the pellets already start in contact. In reality there could be varying axial gaps between different pellets, however because the fuel element is completely sealed at manufacture, the actual axial location of the pellets within the sheath is unknown. Regardless of the initial distribution of gaps any axial gaps between the pellets are consumed by thermal expansion during operation. Since a glued contact constraint is required to prevent rigid body motion, there is assumed to be no axial gap between individual pellets. This is a more accurate approximation then using glued contact with axial gaps between the pellets because the glued contact would prevent the axial gaps between the

pellets from being consumed. This would lead to an unrealistic representation of the fuel element. A fuel element containing axial gaps between pellets while using glued contact would be less stiff than it should be because the pellets contribution to stiffening the sheath would be reduced. The bearing pads and inter-element spacers are not included in this analysis.

To obtain accurate results of stresses and strains within the fuel and sheath, the fuel pellets are modeled including chamfers, lands and dishes for the full fuel element analyses. The radial gap, and fuel diameter remain the same as in the 3D contact analysis, but the pellet length has been reduced to 15.5 mm to more accurately resemble the full length of an element containing 31 fuel pellets (i.e., ~0.5 m long). During manufacturing, the dishes are formed by pressing using a double acting die-set. Dies are shaped to ensure all features of the pellets are reproduced. To recreate the dish in a FEA model, a sphere can be generated which has a radius that can be determined by the dish width and depth. Figure 6.6 illustrates the dish geometry and its relation to the sphere radius. To generate the dish in Trelis, a sphere with the required radius is subtracted from both ends of the fuel pellet (i.e., a dish on each end).

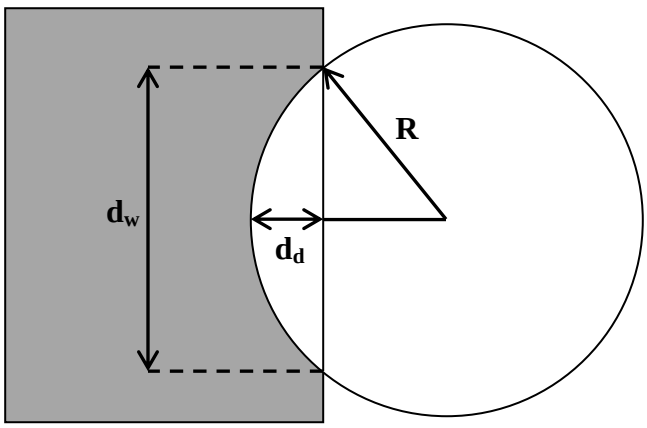


Figure 6.6: Determining the sphere radius required to generate the desired dish geometry.

Where $\mathbf{d_w}$ is the dish width, $\mathbf{d_d}$ is the dish depth and **R** is the radius of the source sphere. Using the Pythagorean Theorem for right triangles the calculation of the source sphere radius is given by

$$R = \frac{d_w^2}{8d_d} + \frac{d_d}{2} \tag{6.1}$$

There are two fuel chamfers applied to both ends of the fuel pellet, an axial and a radial chamfer. The land is defined as the distance between the end of the chamfer and the beginning of the dish. Figure 6.7 shows the definition of the chamfers and the land.

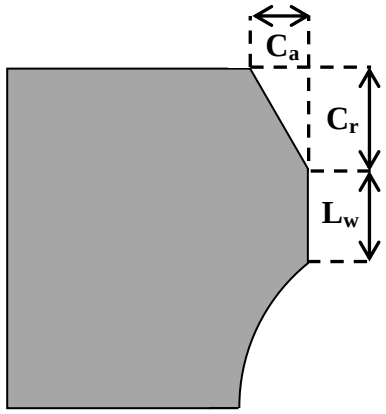


Figure 6.7: Schematic of the chamfer and land fuel features (not to scale).

Where $\mathbf{C_a}$ is the axial chamfer, $\mathbf{C_r}$ is the radial chamfer, and $\mathbf{L_w}$ is the land width. The sheath inner and outer diameters, and thickness also remain the same as in the contact analysis. The fuel element features, including element axial gap, end cap thickness, radial and axial chamfers, dish width and depth, land width and element length are presented in Table 6.3. These nominal values were adapted from references [19, 62].

Table 6.3: Fuel feature dimensions used for the full fuel element model

Fuel Element Feature	Length (mm)
Element Length including End Caps	487.6
End Cap Thickness	2.55
Axial Gap	2
Axial Chamfer	0.066
Radial Chamfer	0.63
Dish Depth	0.2
Dish Width	9.768
Land Width	0.591

The mesh used to represent the quarter model of a full fuel element contained a total of 122012 HEX20 finite elements with a total of 577047 nodes. HEX20's are second order hexahedral elements with quadratic shape functions to more accurately capture the cylindrical shape of the fuel pellets and sheath as well as the curvature of the pellet dish and the chamfer. Ideally for node-to-face contact algorithms, the mesh on the master surface is the same or coarser then the slave surface. Therefore for pellet-to-sheath contact the pellet has a slightly finer mesh axially than the sheath. In the azimuthal direction, the mesh is identical, which is sufficient. For pellet-to-pellet contact, the mesh on the master and slave surfaces is the same because each pellet has the identical mesh. The mesh for pellets one through fifteen in Figure 6.5 is shown in Figure 6.8. Pellets one through fifteen contain 5320 elements and 24143 nodes, whereas the sixteenth pellet contains 2660 elements and 12383 nodes as it represents a quarter of a full fuel pellet.

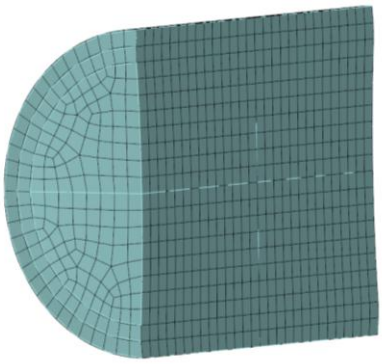


Figure 6.8: Finite element mesh used for the pellets in the full fuel element model.

To accurately determine the stresses and strains within the sheath a minimum of three first order finite elements are required through the thickness. With second order finite elements, one element would be sufficient, but in practice three second order (quadratic) elements are typically used as per discussion with colleagues at AECL. Care must be taken to ensure that the aspect ratio is low enough for these elements while still minimizing the total number of elements. The aspect ratio is the ratio of the longest side to the shortest side of the finite element, and ideally would be as close to one as possible. As the density of the sheath mesh increases, the density for the pellets must also go up to keep the sheath slightly denser for contact purposes. The sheath contains 39552 elements and 202519 nodes.

The primary goal of this work is to assess the capabilities of the MOOSE computational framework for modeling various phenomena of interest that occur in CANDU fuel in three-dimensions, such as sagging, flexural rigidity, thermal bowing, and fuel performance. Therefore, the full fuel element described in this section is used to analyze the flexural rigidity, thermal bowing, and 3D fuel performance capabilities of the framework. Each analysis requires different boundary conditions and constraints, which are discussed next.

6.2.1 Fuel Performance

As with the contact analysis, the model must be constrained against rigid body modes. The constraining of the sheath is assisted by the planes of symmetry introduced for computational efficiency. The nodes on the axial (y-z plane) and midplane (x-y plane) planes of symmetry are fixed such that they cannot move in the direction normal to their respective plane. These constraints eliminate translation in the axial and horizontal directions, and rotation about the midplane and the fuel element's central axis. To prevent rigid body translation in the vertical direction the central node on the outside of the fuel element end cap is fixed from moving vertically. This approximates the restraint of the fuel bundle endplate on the fuel element. Even with all the constraints introduced, pellets one through fifteen in Figure 6.5 are free to move axially. Even though the pellets have no axial gap between them, frictionless contact is used for pellet-to-pellet contact to allow tangential sliding between the pellet surfaces. Since the pellets are not bonded to one another the pellets would be permitted to slide axially within the sheath. Ideally, frictional contact with a high enough coefficient of friction would prevent the unconstrained axial sliding because pellet bottoming causes the pellets to be initially in contact with the bottom of the interior surface of the sheath. However, the frictional contact algorithms in MOOSE are not yet robust enough for 3D modeling of the magnitude analyzed here. Therefore to sufficiently constrain the pellets within the sheath, a glued contact model with penalty enforcement is used for pellet-to-sheath contact. The details of this contact algorithm were provided in Section 4.3. Glued contact forces the two contacting bodies to stay in contact once they have come in contact, and does not allow for tangential sliding. In reality the pellets do slide tangentially along the inside of the sheath, but due to the rapid closing of the diametrical gap under normal operating conditions, and the high pressures experienced on the sheath, the gap

will not reopen and axially sliding of the pellets is minimal. There are implications associated with this approximation that need to be investigated, such as the axial gap between the end pellet and the end cap not closing. However, a full investigation must wait until there is a suitable tool to complete the analysis (i.e., frictional contact). For the analyses completed in this work, the glued contact approximation was deemed acceptable.

The temperature boundary conditions are provided by the coolant properties and initial conditions in Table 6.1. To examine 3D fuel performance the fuel element model is subject to a typical power profile of an ELESTRES test case for two separate cases, normal operating and overpower. Results are compared to ensure that the fuel performance of the full fuel element model in HORSE is behaving in an expected fashion. Since there is no fission gas release model currently in HORSE (there is a model in BISON for LWR fuel rods), the gas composition and amount of fission gas released to the gap is added as an input into the calculation of the gap parameters. This allows for comparison against ELESTRES with a fuel element of appreciable burnup as the addition of fission gas degrades the heat transfer across the fuel-to-sheath gap.

6.2.2 Flexural Rigidity

For the flexural rigidity analysis the temperature and most of the mechanical boundary conditions are identical to the fuel performance analysis. The only difference applies to the vertical translation constraint of the sheath. In this analysis a single node is prevented from translation in the vertical direction on the bottom of the sheath along the y-z plane at the outside surface of the end cap. This causes the fuel element to behave like a simply supported beam.

The flexural rigidity analysis examines the stiffness enhancement of the sheath due to the pellets that are within. To examine the flexural rigidity, simulations are run with constant linear power ratings of 30, 45, and 60 kW m^{-1} until steady-state is reached to assess the effect of low, normal, and high operating powers on the results. Once the element reaches steady-state, a nodal force should be applied to the node at the intersection of the planes of symmetry at the top of the sheath to determine the response of the fuel pin; however MOOSE does not currently contain a nodal force boundary condition. Therefore, an external pressure is ramped from 0 to 130 MPa on the finite element at the top of the sheath at the intersection of the two planes of symmetry. The pressure is ramped at a rate of 0.325 MPa s^{-1} resulting in the total pressure of 130 MPa being applied after 400 s. When applying a pressure boundary condition, the pressure is always applied perpendicularly to the surface even as the element deflects, whereas a force would always be applied in the same direction. For this analysis because the area where the pressure is applied is very small (0.384853 mm^2), and only the vertical component of the pressure is used, it is an acceptable substitute to a nodal force. When applying a pressure, the total pressure applied needs to be adjusted such that the desired force is applied to the midplane of a full fuel element. The range of pressures applied yields loads ranging from 0 to approximately 200 N on the entire fuel element. By quantifying the midplane deflection of the element due to a particular load, the flexural rigidity of the fuel element can be determined via [63]

$$y = \frac{-PL^3}{48EI} \tag{6.2}$$

where y is the vertical midplane deflection, P is the applied load at the midplane, L is the length between the supports, and EI is the flexural rigidity. Plots at various linear powers would show how the fuel pellets add to the flexural rigidity for both elastic and elastic-plastic models of the sheath. Simulations of an empty sheath containing no pellets, and a discrete stack of UO_2 fuel

(i.e., nodes on the lands are merged) without a sheath are completed for verification purposes against analytical solutions to ensure the material properties are coded correctly. Based upon the results that will be presented in Section 7.3 additional simulations were completed that ramped the applied load to 3200 N over 1100 s after steady-state for all three linear powers using an elastic model for the sheath.

6.2.3 Thermal Bowing

The thermal and mechanical boundary conditions of the thermal bowing analysis are identical to that of the flexural rigidity analysis up to steady-state. Upon reaching steady-state a thermal gradient is applied to the fuel element such that it bows. The gradient is applied by varying the bulk coolant temperature for the convective boundary condition on the outside of the sheath. The bottom of the sheath (y = -6.551 mm) is kept at the initial bulk coolant temperature of 583 K and the top of the sheath (y = 6.551 mm) is forced to a higher temperature denoted by T_1. The bulk coolant temperature is linearly interpolated between these two points. Simulations are completed where T_1 is equal to 683, 783, and 883 K respectively for a constant linear power of 45 kW m^{-1}. The equation used to apply the varying bulk coolant temperature is given by

$$T(y) = \left(\frac{T_1 - 583}{0.013102}\right) y + \frac{1}{2}(T_1 - 583) + 583 \tag{6.3}$$

This analysis can provide insight into the bowing behaviour of an element during transient conditions when a thermal gradient forms across the diameter of the fuel element. An estimate of the bowing at the fuel element midplane can be determined as a function of the thermal gradient applied and the coefficient of thermal expansion of the material via

$$y_{max} = \frac{-\alpha L^2}{8d}(T_2 - T_1) \tag{6.4}$$

where L is the length between the supports of the beam, α is the instantaneous coefficient of thermal expansion of the material, d is the diameter of the beam, T_2 is the temperature at the bottom of the beam, T_1 is the temperature at the top of the beam. It is important to note that simulations are completed for an empty sheath and discrete stack of fuel (nodes on the lands merged) which are compared against analytical calculations. In the case of the discrete fuel stack the varying convective boundary condition is applied to the outside surface of the fuel.

6.3 Solver Options

In order to run a simulation in HORSE solver settings are required. These parameters include the type of executioner, initial timestep size, minimum and maximum allowable timestep size, simulation start and end time, type of preconditioning, linear and nonlinear tolerances, maximum number of linear and nonlinear iterations, and other PETSc options. In MOOSE there are a variety of executioner methods available which determine the behaviour of the solver, including steady-state, transient, adaptive transient, and transient with an adaptive timestepper. The steady-state executioner uses the supplied end time and attempts to solve the solution completely from the start time to the end time in one timestep where only the initial and final solutions are output. A transient executioner starts the solution from the start time and advances by the provided initial timestep size. If the solution converges the timestep remains constant until the end time is reached. If at any point the solution fails to converge (the maximum number of nonlinear iterations is reached) the timestep size is reduced by one half. The reduced timestep size becomes the timestep size for all subsequent timesteps. The adaptive transient executioner is similar to the transient executioner except that if the solution converges within a specified number of nonlinear iterations the solver doubles the timestep increment. This allows the

executioner to reduce the timestep if the solver is struggling to converge and increase the timestep if the solver is converging rapidly. The transient executioner with an adaptive timestepper contains a subtle difference to the adaptive transient executioner. Within the input file a timestepper sub-block is defined that specifies the adaptive timestepping parameters. Within this block, there is an option to force the timestepper to take timesteps at particular points as governed by a function within the simulation. This is useful when a particular power history is to be followed, and it is desired that a solution be output at each point defined in the power history file. At times between those specified in the power history file, the solver behaves as the regular adaptive transient executioner.

Another important solver option is the linear tolerance, and relative and absolute tolerances for the nonlinear iterations that determine convergence. In MOOSE there are two ways of determining when a solution has converged, the initial residual and the reference residual. When using the initial residual as the convergence criteria the relative and absolute tolerances of the nonlinear iterations need to be tighter than in the case of the reference residual. This is because the residual is mathematically constructed based upon the system being analysed and the goal is to minimize it as much as possible. Setting the tolerances too tight may lead to divergence. However, when using the reference residual option, the solution is compared against the reaction forces due to the loads applied on the system. This allows for looser tolerances and faster convergence of solutions with limited loss in accuracy in most cases. It also allows some problems that would not converge using the initial residual criteria to converge. In the reference residual method, convergence of each of the four state variables, x, y, and z displacement and temperature must be converged before the timestep is considered converged. Therefore, the

reference residual convergence criterion was used for all analyses completed in this work. The other PETSc options defined in the executioner block in the sample input file, as indicated in Appendix E, were recommended by the BISON developers. The description and behaviour of the multitude of PETSc options available and their uses is beyond the scope of this work, and the reader is encouraged to read the PETSc User's Manual [32] for further information. For all analyses completed here the default preconditioned JFNK method is used. Table 6.4 summarizes the executioner options used for the various analyses completed in this work including the 2D and 3D contact models, the flexural rigidity, thermal bowing, and fuel performance models, and the fuel element simulator model. Table 6.5 provides the linear and nonlinear tolerances used for all analyses completed.

Table 6.4: Executioner parameters used for each model.

	2D Contact	3D Contact	Flexural Rigidity	Thermal Bowing	Fuel Performance
Start Time (s)	0	0	0	0	0
End Time (s)	40	40	700 1400*	1300	2.8152×10^{7} (N) 2.5097×10^{7} (O)
Initial Timestep (s)	1	1	3	3	20
Maximum Timestep (s)	1	10	50	50	10^{6}
Minimum Timestep (s)	0.01	0.1	1	1	1
Maximum Linear Iterations	100	100	100	100	100
Maximum Nonlinear Iterations	25	25	25	25	25
Executioner	Transient	Adaptive Transient	Transient with Timestepper	Transient With Timestepper	Transient With Timestepper

*****1400 s was used for the analyses when the load was ramped to 3200 N.
(N) Indicates the end time for the nominal case.
(O) Indicates the end time for the overpower case.

Table 6.5: Linear and nonlinear tolerances used for all models.

Tolerance	Value
Linear	1×10^{-2}
Nonlinear Absolute	1×10^{-4}
Nonlinear Relative	1×10^{-3}

6.4 The MOOSE/HORSE Input File

In order to run a simulation in a MOOSE based application an input file is required. The input file is divided into blocks which are used to specify different components of the simulation. An explanation of all the blocks contained within the input file used for the fuel performance models are included in this section to provide the reader with some additional insight into the workings of the MOOSE framework and its associated applications. An example input file is included in Appendix E.

The Global Parameters (GlobalParams) block is utilized to specify global variables available for use by other blocks within the input file. Usually the order of the finite elements and the family (e.g., Lagrange, Monomial) of the state variables are specified here. Throughout the input file a Lagrange family indicates a nodal variable, whereas a monomial family indicates an elemental variable. The Problem block is present to define the parameters of the reference residual convergence criterion presented in Section 6.3. If convergence against the initial residual is desired, the problem block may be removed and the tolerances in the Executioner block adjusted accordingly. The Problem block is also used if a two-dimensional axisymmetric (r-z) simulation or kinematic enforcement of frictional contact is desired. The Mesh block provides the full path to the mesh file to be used in the analysis. If the file is in the same directory as the input file the name of the mesh file is sufficient. The displacement directions that the mesh can undergo during the simulation are also included here, which is required for contact.

The Variables block specifies the state variables to be solved for in the simulation. In the models produced in this work, the state variables are always x, y, and z displacements and temperature.

Initial conditions of any variables are specified within the Variables block. The Auxiliary Variables (AuxVariables) block defines the variables that are calculated based upon the state variables or are for analysis purposes (e.g., stresses, strains, burnup). In most cases auxiliary variables are elemental variables. There is the option to output elemental variables as nodal variables for smoother contour plots. The Functions block is where functions required in the simulation are defined. The most common function in nuclear fuel performance analyses is the power history. Piecewise linear functions defined in this block can be functions of time or position.

The Solid Mechanics (SolidMechanics) block is used to reduce the number of kernels to be defined in the input file. This block generates the required kernels for each displacement direction required in the simulation, rather than requiring the user to list them separately in the Kernels block. The Kernels block defines the components of the PDEs that the variables in the Variables block must satisfy. Each variable defined in the Variables block must have at least one kernel associated with it. The Auxiliary Kernels (AuxKernels) block defines the calculations to be completed on the auxiliary variables and how often the calculation is completed (e.g., beginning of timestep, residual). The Auxiliary Boundary Conditions (AuxBCs) block is used to determine the value of a particular auxiliary variable on a specified surface or surfaces. This saves computational time when it is known that a variable will only have a finite value on a surface. Every auxiliary variable must have an auxiliary kernel or boundary condition defined, with the exemption of the saved variables which are used for reference residual convergence.

The Contact block is where the mechanical contact parameters and the contact surfaces are defined for each contact pair. The Nodal Normals (NodalNormals) block contains a list of the master surfaces in the analysis to perform a smoothing algorithm to make contact more robust and stable. The Thermal Contact (ThermalContact) block is where the all of the thermal contact parameters for each contact pair is defined. Although mechanical and thermal contact is defined separately in the input file, they are solved simultaneously.

The boundary conditions and materials used in the simulation are defined in the Boundary Conditions (BCs) and Materials blocks in the input file. The sub-blocks define the type of boundary condition or material and which sideset, block (fuel or sheath), or nodeset to which it is applied. Depending on the type of BC or material a variety of other parameters must be defined for the simulation to execute.

Dampers can be used to help slow the system down to avoid rapid changes in conditions within a single iteration. Dampers can be used to limit the increment that a particular state variable can take within a single nonlinear iteration. These are defined in the Dampers block. The Executioner block contains all the information required for the solver options discussed in Section 6.3. Postprocessors are calculations that are sent to the output file for data analysis or useful simulation information (e.g., timestep size, number of nonlinear iterations). Sometimes postprocessors are coupled back to boundary conditions or materials as they are needed. For example, a postprocessor can be defined that calculates the average surface temperature of the fuel. This is coupled back to the gap conductance model as it is required to determine the emissivity of the fuel as given by Equation (5.51). Finally, the Output block lists a variety of

output options. It is in this block where the output file name is defined, what type of files are to be output for further analysis (e.g., ExodusII), among other options. The output block has recently been phased out for a new Outputs block.

Table 6.6 summarizes the input file blocks used in this work, including their names, descriptions, and examples of their use.

Table 6.6: Summary of the input file blocks

Block Name	Description	Example Parameters/Use
GlobalParams	Defines global variables available for use by other blocks.	Order of finite elements, family of state variables
Problem	Defines a specific type of problem to be used for the simulation.	Reference residual, kinematic frictional contact, axisymmetric
Mesh	Supplies the mesh to be used in the analysis.	Mesh file
Variables	Defines the state variables to be solved.	Displacement, temperature, initial conditions
AuxVariables	Defines variables used for analyses purposes.	Stresses, strains, burnup
SolidMechanics	Defines the solid mechanics kernels for every direction in the simulation.	Eliminates the need to explicitly define the displacement kernels
Kernels	Defines the terms that the variables are to satisfy on the entire domain of interest.	Heat conduction, heat generation
AuxKernels	Defines the calculations to be completed on the AuxVariables and how frequent the calculation is made.	Stresses, strains, burnup, reference residual variables
AuxBCs	Defines calculations to be completed on the AuxVariables that only occur on boundaries.	Gap conductance
Contact	Defines the mechanical contact pairs.	Pellet-to-pellet, pellet-to-sheath
NodalNormals	Defines the list of master surfaces used for smoothing algorithms.	Used when contact is between curved surfaces.
ThermalContact	Defines the thermal contact pairs.	Pellet-to-pellet, pellet-to-sheath
BCs	Defines the boundary conditions the state variables much satisfy.	Coolant temperature, displacement constraints
Dampers	Defines the maximum increment a state variable takes in nonlinear iteration.	Limit the temperature increase or displacement
Executioner	Defines the solver parameters used in the simulation.	Timestepper, tolerances, PETSc options
Postprocessors	Defines the postprocessors to be used in materials or postprocessing.	Gas temperature, number of iterations, plenum pressure
Output	Defines the output parameters for the simulation.	ExodusII, terminal (screen), initial condition

Chapter 7 - Results and Discussion

Results from computer models should be benchmarked against existing experiments, results from validated codes, similar computational models or analytical solutions. In this chapter a detailed analysis of the contact algorithms within MOOSE is provided. A comparison of the flexural rigidity, thermal bowing, and fuel performance models against Industry Standard Toolsets, existing commercial software models and or analytical solutions are also discussed in this chapter.

7.1 Contact Analysis

To analyze the way contact behaves within the MOOSE framework, constant material and gap heat transfer properties were used to isolate the contact behaviour from nonlinearities that would be introduced from these sources. The material properties, geometry and mesh were discussed in Section 6.1. As previously discussed MOOSE separates the declaration of mechanical and thermal contact in the input file but solves them simultaneously. Table 7.1 lists the contact settings used for the contact analysis to simplify the behaviour. The penalty factor used for mechanical contact is varied to examine its effect on the contact pressure and penetration between the pellets and sheath. The tangential tolerance is used to extend the master surfaces by a small amount at planes of symmetry to avoid slave nodes from not having a contact surface in front of it. The specified gap conductivity is the thermal conductivity of helium gas, which is chosen as the fill gas for this test. Since a thermal conductivity is given (as per helium) the gap heat transfer coefficient is determined by dividing the thermal conductivity by the gap width. These parameters are used for both the 2D and 3D contact analyses for consistency.

Table 7.1: Mechanical and thermal contact parameters used in the contact analysis.

Parameter	Mechanical Contact	Thermal Contact
Penalty Factor	Varies	-
Tangential Tolerance (m)	10^{-4}	-
Gap Conductivity (W m^{-1} K^{-1})	-	0.15

Both 2D and 3D analyses were completed to examine dimensionality effects on the contact algorithms. The analyses examine both kinematic and penalty enforcement of frictionless and glued contact models as well as penalty enforcement of frictional contact. The results of both the 2D and 3D analyses are presented in the following two subsections.

7.1.1 Two-Dimensions

The contact force produced when two bodies come into contact depends upon the penalty factor chosen, the loads applied, and the calculated interpenetration. The resulting pressure at a node can be determined via the nodal area associated with that node. The contact pressure is one characteristic value for a specific combination of loads, material properties, and geometry that should be constant if no penetration is observed. When using the penalty enforcement with frictional and frictionless contact some amount of penetration is required to produce a contact force. Therefore it is expected that as the penetration asymptotically approaches zero, the contact pressure approaches its constant value. To avoid the choice of penalty factor from affecting the results, a penalty factor that produces results on the asymptotic plateau should be used.

Figure 7.1 shows (a) the contact pressure and (b) penetration as a function of the penalty factor for the 2D frictionless case. Simulations using kinematic contact enforcement on the first order mesh led to oscillatory situations and no convergent results were obtained. Based upon the

results from the second order mesh it appears that the kinematic enforcement produces the maximum value for contact pressure and penetration that would be observed when the pellets come into contact with the sheath as the results are constant regardless of the penalty factor used. This is expected based upon the equations introduced in Table 4.1, which indicates that kinematic enforcement forces penetration to zero. In addition the kinematic results appear to be the bounding case as to which the penalty enforcement results approach asymptotically. The contact pressure obtained through kinematic enforcement is 44.8 MPa and the penetration is much less than the surface roughness of the materials. For the kinematic case convergence was not achieved until a penalty factor of 1×10^{7}. This indicates that when using kinematic contact enforcement higher penalty factors are required than with penalty contact enforcement. However, because the results are constant regardless of the penalty factor chosen, it is best to choose the lowest penalty factor that obtains convergence because it will result in the fastest simulation time.

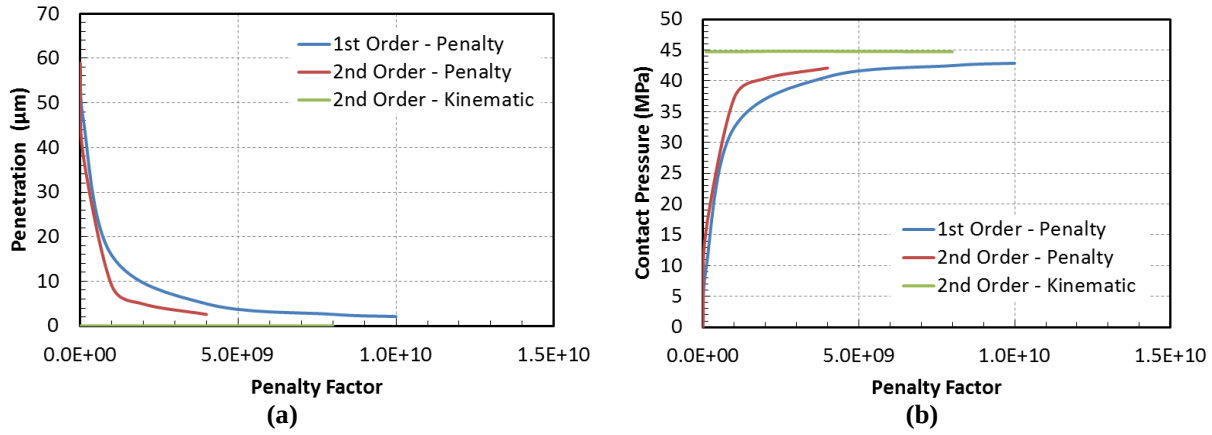


Figure 7.1: Frictionless contact model results for (a) penetration and (b) contact pressure as a function of the penalty factor in two-dimensions.

Another important observation from the 2D frictionless analysis is that the second order finite elements approach the asymptotic values quicker than the first order elements but the highest

penalty factor for which converged solutions could be obtained is lower. Figure 7.2 presents the contact pressure results for both penalty and kinematic contact enforcements, for first and second order meshes for a glued contact model in 2D. Penetration results are not presented because the glued contact model enforces zero penetration as explained in Section 4.3. Similar to the frictionless case the kinematic enforcement appears to provide a limiting value that the contact pressure should approach. This value is 44.75 MPa. The same trends are observed with first order and second order penalty enforcement as with the frictionless contact model. The results between frictionless and glued contact models are very similar for the 2D geometry examined in this analysis.

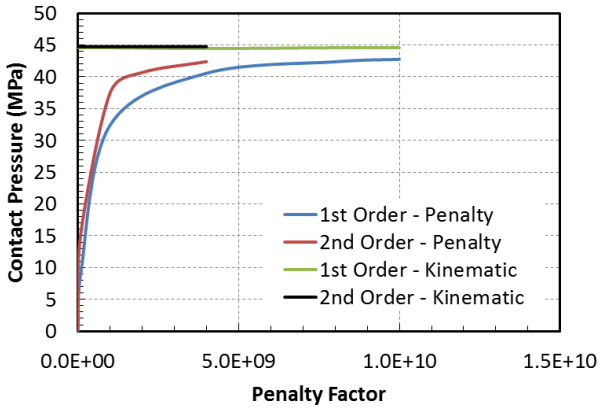


Figure 7.2: Contact pressure as a function of penalty factor for the 2D glued contact model.

Due to the way that the 2D geometry is constructed the results from penalty enforcement of frictional contact will be identical to the results for penalty enforcement of frictionless contact as there is no possibility of tangential sliding of the exterior pellet surface across the interior surface of the sheath. If an azimuthal slice of the pellet was taken (i.e., axisymmetric) as used in the FAST code [21], frictional contact results would differ from the frictionless results as tangential sliding would occur as the pellet expands axially. However, the geometry used in this analysis

was chosen because it approximately represents the transverse slice at the midplane of a fuel pellet. This is important when comparing the results obtained from the 3D contact analysis.

7.1.2 Three-Dimensions

The 2D contact analysis provided some basic insights into how the contact algorithms behave in MOOSE. The results observed were as expected based on an intuitive understanding of how the penalty and kinematic contact models are constructed as summarized in Table 4.1. However, the 3D contact analysis provides additional insight that must be understood before analyzing the fuel performance, flexural rigidity, and thermal bowing models.

Figure 7.3 (a)-(f) presents some of the results obtained for the frictionless and glued contact analyses in 3D. All of the results focus on penalty enforcement of contact because it provides the most interesting results. Kinematic results produce the asymptotic limits as experienced in the 2D case. In three-dimensions, the contact pressure and penetration results are different depending upon the axial location along the pellet. The pellet is divided into two broad areas, the midplane, and the ridge. This phenomenon is known as pellet hourglassing caused by the radial temperature distribution within the fuel pellet and is not observed in the 2D case. It is encouraging to observe pellet hourglassing when using constant material properties and uniform heat generation. The contact pressure results for the original mesh are presented in Figure 7.3 (a). It can be seen that first order finite elements produce higher contact pressures then second order elements. Moreover, the contact pressure at the pellet midplane is 10-13 times less than at the pellet ridge depending upon the mesh used as described below. In addition, the contact pressure at the pellet midplane is slightly lower than that observed in the 2D case and this is because pellet hourglassing causes the pellet ridges to penetrate further into the sheath which

causes less penetration at the pellet midplane. This observation is illustrated in Figure 7.3 (d) which shows the penetration results for frictionless contact. Figure 7.3 (b) shows the variation in contact pressure at the pellet midplane for the second order original, half and quarter meshes. As the mesh density increases the profile of the contact pressure follows a very similar curve indicating that the solutions have converged on the mesh. However, when the mesh sensitivity is completed at the pellet ridge the contact pressure solutions depend strongly on the mesh density. To further investigate, meshes containing approximately double and quadruple the amount of nodes of the original mesh were analyzed at the pellet ridge. The results of this study are presented in Figure 7.3 (c). These results indicate that the solution does not converge on the mesh at the pellet ridge. The quadruple mesh density simulations encountered convergence issues at the higher penalty factors and the computational requirements continually increased. This is because abrupt (sharp) edges do not have an associated area meaning the contact pressure at these points should approach infinity. As the mesh density increases, the nodal areas associated with the nodes decrease towards zero causing the contact pressure to continually increase. Figure 7.3 (e) presents the glued contact results for the original mesh. Similarly to the frictionless case the first order results are higher than the second order results. By comparing Figure 7.3 (e) and Figure 7.3 (a) it can be seen that the glued contact pressures are slightly higher than the frictionless. This is further shown in Figure 7.3 (f) which highlights contact pressure results at the pellet ridge of the original mesh. The contact pressure being higher for the glued model compared to frictionless is expected because the contact pressure is calculated based upon the distance between the slave node and the project location of the slave node on the master surface from the previous iteration as illustrated in Table 4.1. It can also be seen that the magnitude of the penalty factor required to reach the asymptotic plateau is much lower in 3D.

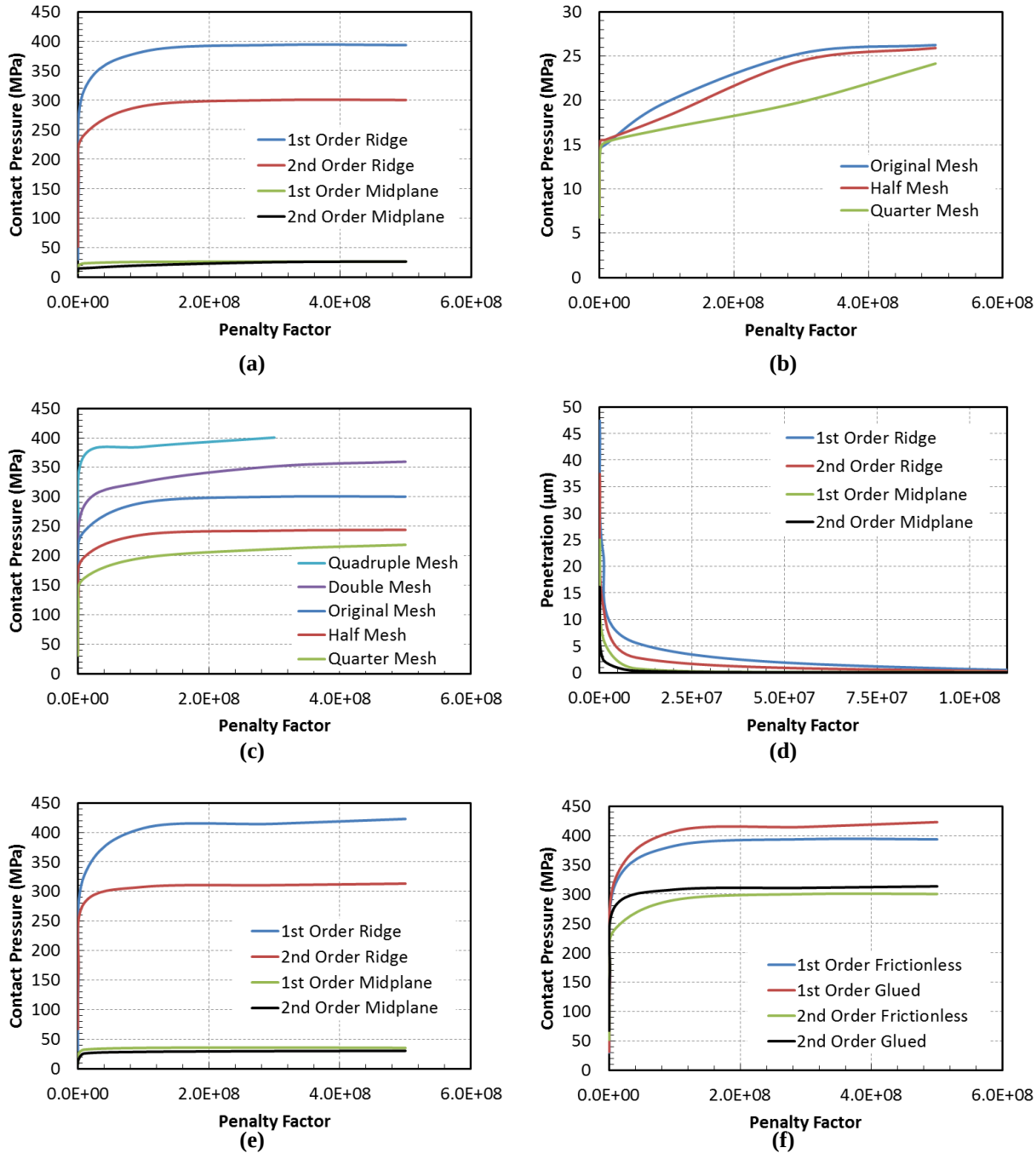


Figure 7.3: Various results from the 3D frictionless and glued contact analyses, including (a) frictionless contact pressure at the pellet ridge and midplane for the 1st and 2nd order original mesh, (b) mesh sensitivity of frictionless contact at the pellet midplane, (c) mesh sensitivity of frictionless contact at the pellet ridge, (d) frictionless penetration at the pellet ridge and midplane for the 1st and 2nd order mesh, (e) glued contact pressure at the pellet ridge and midplane for the 1st and 2nd order original mesh, (f) comparison of the glued and frictionless contact pressures at the pellet ridge for the 1st and 2nd order original mesh.

The final study in the contact analysis was to examine the penalty enforcement of frictional contact. Select results from the frictional contact analyse are presented in Figure 7.4 (a)-(d). Figure 7.4 (a) shows the contact pressure as a function of penalty factor for the original mesh with a coefficient of friction of 0.6 used between the pellet and sheath. As expected the contact pressure is higher at the pellet ridge than at the midplane. However, convergence difficulties were encountered at penalty factors greater than 2×10^6, and the contact pressure at the ridge has not yet approached an asymptotic constant pressure as expected. Figure 7.4 (b) plots the mesh sensitivity of the contact pressure at the pellet ridge for μ=0.6. When the mesh density increases the contact pressure increases further indicating the solutions have not converged on the mesh. Moreover convergence at higher penalty factors is obtained for lower density meshes but not higher density meshes. This is because denser meshes have more nodes slipping as a result of friction. Figure 7.4 (c) presents results that are consistent with the glued and frictionless analysis. This figure shows a comparison of the contact pressure at the pellet ridge for the second order original mesh for coefficients of friction equal to 0.2 and 0.6.

At the pellet midplane the results are essentially identical, however, at the pellet ridge the contact pressures becomes higher for the higher coefficient of friction. This is expected because the contact pressure from glued contact was higher than frictionless contact. As the coefficient of friction increases towards one it approaches a glued contact state, whereas as the coefficient decreases towards zero it approaches a frictionless state. The results are consistent with this observation. Finally the penetration results of the original mesh are shown in Figure 7.4 (d). As expected the penetration is lower at the midplane than at the ridge. However, the observed penetration is much higher than the surface roughness of the fuel and the sheath at the ridge.

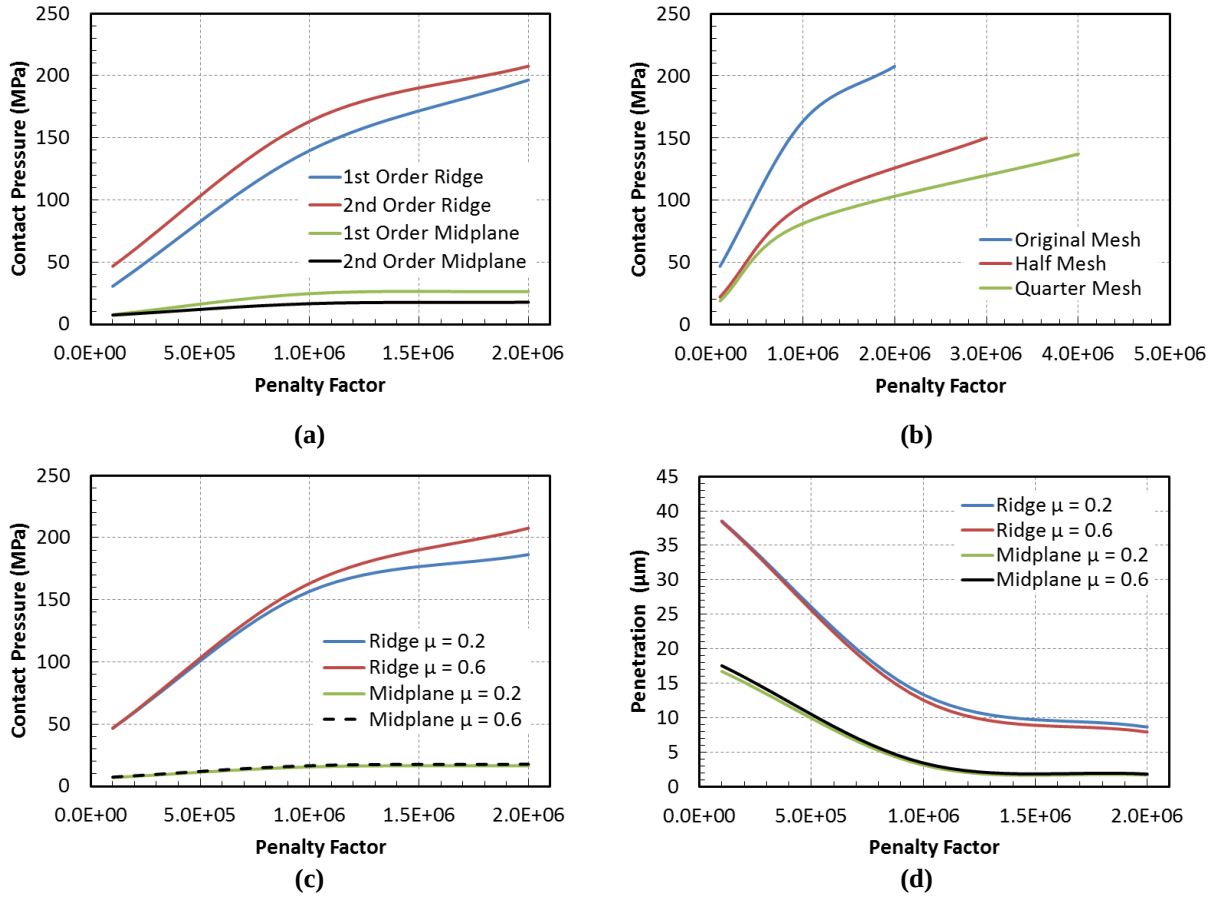


Figure 7.4: Results of 3D frictional contact analysis, including, (a) contact pressure at the pellet ridge and midplane for the 1st and 2nd order original mesh with μ=0.6, (b) contact pressure at the pellet ridge for the 2nd order original, half, and quarter meshes with μ=0.6, (c) comparison of the contact pressure at the pellet ridge for the 2nd order original mesh using μ=0.2 and μ=0.6, (d) comparison of the penetration at the pellet ridge for the 2nd order original mesh using μ=0.2 and μ=0.6.

Attempts were made to use penalty enforcement of frictional contact for the models analyzed in the subsequent sections; however, the models encountered negative Jacobian errors which stopped the simulations and led to divergence. A negative Jacobian in a finite element is when the node numbering of the element becomes reversed (also known as an inverted mesh element). This occurs when an element inverts on itself resulting in a negative volume. The results obtained for the contact pressure and penetration as a function of the penalty factor from the frictionless and glued contact analyses are as expected. Ideally for penalty enforcement a

penalty factor should be chosen such that the contact pressure observed is on the asymptotic plateau as observed by the frictionless and glued models.

In the full fuel element analyses, a penalty factor of 1×10^{7}, which is just before the asymptotic plateau, was used for both pellet-to-sheath and pellet-to-pellet contact to balance ease of convergence and computational time. Pellet-to-sheath contact uses a glued contact model to prevent rigid body motion of the fuel pellets with penalty enforcement and pellet-to-pellet contact uses a frictionless model with kinematic enforcement to allow tangential sliding between the pellets.

7.2 Fuel Performance

The study of fuel performance examines the temperature profile, stress and strain states within the pellet and cladding during irradiation, and other parameters including the volume of fission gas released. In many cases fuel performance analyses can be completed using 2D axisymmetric simulations as in FAST [20] or BISON [46], or quasi-2D like ELESTRES and ELOCA. Two-dimensional axisymmetric simulations reduce computational times and constraining the system against rigid body motion becomes easier. In addition, frictionless or frictional contact can be used instead of glued contact as the pellets can be constrained separately from the sheath. However, depending upon the level of detail desired and the situation under investigation 3D fuel performance analyses may still be required. For example, if one of the fuel pellets in the fuel stack has a missing surface (i.e., one of the pellets is not perfectly cylindrical) a 2D axisymmetric analysis cannot be used. This scenario has been simulated with the BISON code for a fuel rodlet (shortened fuel stack) for an LWR. Although the fuel performance analyses completed in this work could be analysed with a 2D model, a 3D model provides a way of

analyzing the behaviour of the burnup dependence of the material properties, the fission gas input mechanisms employed in HORSE, and further determining if the contact algorithms are behaving as expected in three-dimensions. Assessing the feasibility of using MOOSE for 3D fuel performance in CANDU reactor fuel elements is an objective of this thesis.

For the fuel performance analysis two test cases were completed, nominal and overpower. Both of these power histories are given as linear power as a function of burnup and are presented in Figure 7.5. The nominal power history is a representation of a typical operating profile for a CANDU reactor under normal operating conditions. The overpower case examines a power history at higher than normal operating powers resulting in higher fission gas releases, temperatures and strains. ELESTRES operates such that each timestep is treated as a steady-state calculation and thus the power history is a series of step functions. In HORSE the transition between one linear power to another is instantaneous because a piecewise constant function is used as the timestep limiting function in the transient executioner with an adaptive timestepper. This eliminates the need to ramp the power over short time periods which would require the use of a short timestep and a transient solution technique, which in turn significantly increases the computation time.

Six parameters are analyzed to determine the capability of the HORSE full fuel element to predict fuel parameters. These parameters include the heat transfer coefficient across the gap as a function of time, the sheath inside, fuel surface and fuel centerline temperatures as a function of burnup (time), and the hoop strain and stress at the pellet ridge over time. These results are presented in Figure 7.6 (a)-(f) respectively.

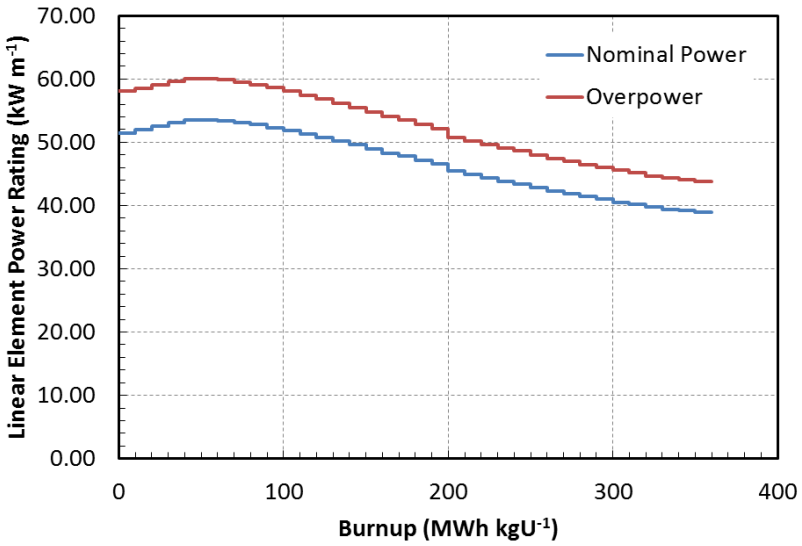


Figure 7.5: The power history profile for nominal and overpower situations used for the ELESTRES fuel performance comparisons.

By examining Figure 7.6 (a) it can be seen that the heat transfer coefficient degrades as a function of burnup in the ELESTRES simulations with the overpower case degrading the most. This degradation is due to the production of fission gases that are released to the fuel-to-sheath gap. In addition as the power supplied to the fuel decreases at higher burnups as shown in Figure 7.5 the contact pressure between the fuel and sheath decreases further reducing the fuel-to-sheath heat transfer coefficient. In the HORSE simulations it is observed that the heat transfer coefficient is lower than the ELESTRES simulations at low burnups but rapidly increases to a constant value for the rest of the simulation. The heat transfer coefficient remains relatively constant in the HORSE simulations due to the glued contact constraint. Once the fuel and sheath come in contact they remain in contact and the contact pressure remains essentially constant. This causes the solid-to-solid component of the heat transfer coefficient to be overestimated at higher burnups. Moreover, the axial gap within the element is not consumed because glued contact prevents the pellets from expanding axially to consume the gap causing the internal gas

pressure to be underestimated as the internal volume is too high. The lower internal gas pressure degrades the gas conductance term of the heat transfer coefficient through the jump distance and thus under predicting the internal gas pressure results in more efficient heat transfer.

The effects of the constant heat transfer coefficient for the duration of irradiation can directly be seen in Figure 7.6 (b)-(d). It is observed that the temperature of the inside surface of the sheath is slightly underpredicted by HORSE in Figure 7.6 (b). The same qualitative trends are observed between HORSE and ELESTRES. The largest deviation is 0.39% at a burnup of 0.0027 MWh kgU^{-1} for the nominal case, and 0.48% at a burnup of 0.0029 MWh kgU^{-1} for the overpower case.

For the fuel surface temperature in plot (c) the temperature predicted by HORSE is significantly underestimated with a maximum deviation of 15.4% at a burnup of 360 MWh kgU^{-1} for the nominal simulation and 22.1% at a burnup of 208.3 MWh kgU^{-1} for overpower. This severe under prediction of the fuel surface is a direct result of the constant heat transfer coefficient across the pellet-to-sheath gap once the pellet gap has closed due to glued contact. In the ELESTRES simulations as the heat transfer coefficient degrades the fuel surface temperature increases, whereas the HORSE simulations remain relatively constant only decreasing by 29.1 K from burnups of 10 to 360 MWh kgU^{-1}. Due to glued contact the fuel surface and inside sheath surface temperatures begin to approach the same value in the HORSE simulations.

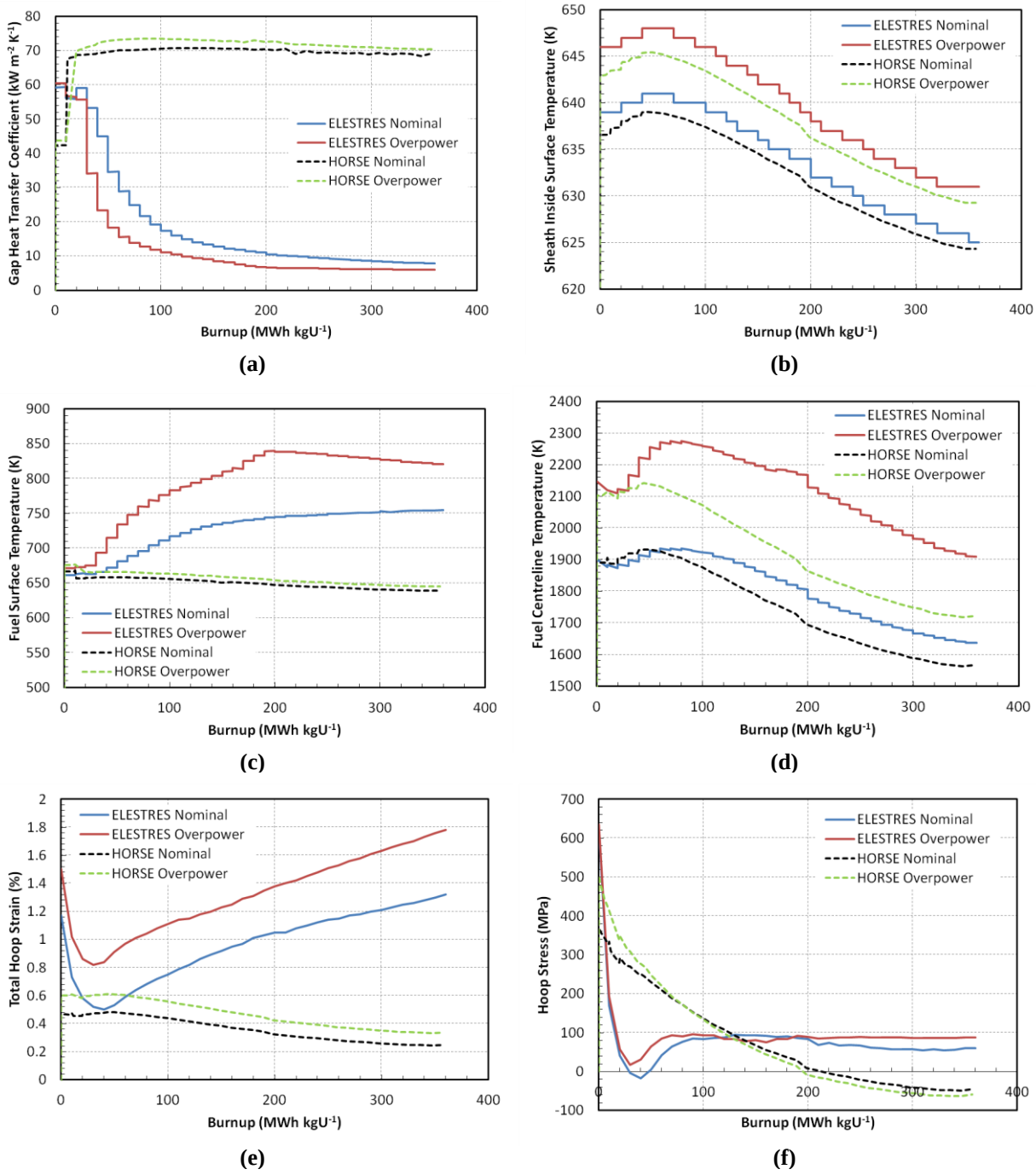


Figure 7.6: Comparison of ELESTRES and HORSE results of (a) heat transfer coefficient across the gap, (b) inside sheath surface temperature, (c) fuel surface temperature, (d) fuel centerline temperature, (e) sheath hoop strain at the pellet ridge, and (f) sheath hoop stress at the pellet ridge for both nominal and overpower cases.

The lower fuel surface temperatures in the HORSE simulations directly influence the fuel centerline temperature. Therefore, the fuel centerline temperatures in Figure 7.6 (d) are also underpredicted. Interestingly, even with the assumptions made in HORSE, including glued contact and the lack of fission gas production and release, the maximum difference between the centerline temperatures are 77.15 K (4.69%) and 262.83 K (12.3%) for the nominal and overpower cases respectively. The nominal value is well within the experimental uncertainty of the thermal conductivity correlations and fuel measurements, and are consistent with the findings of McCluskey [28], but the overpower results is slightly out of this range and is directly a result of the requirement of a glued contact constraint. Another contribution to the deviation in centerline temperature could be the use of slightly different correlations of thermal conductivity between ELESTRES and HORSE. In ELESTRES at temperatures below 727 K the thermal conductivity of UO_2 is taken equal to the value at 727 K whereas HORSE uses the irradiation damage correlation given by equation (5.24).

The last fuel performance parameters of interest examine the mechanical response of the fuel pellets and sheath. The hoop strain and stress within the sheath at the pellet ridge (pellet-to-pellet interface) are presented in Figure 7.6 (e) and (f). From the very beginning of irradiation the hoop strain is under predicted. This result highlights the requirement of a fuel cracking model in 3D fuel element simulations. Fuel cracking releases the built up stresses and allows the fuel pellets to further expand in the radial direction which in turn causes more strain on the sheath. In a 3D model without pellet cracking the Poisson effect causes the pellet to expand axially more than would normally be observed. However, in a 2D model without pellet cracking the pellet will continue to expand in the radial direction without releasing the built up stresses

resulting in correctly predicted strains in the sheath but over predicted stresses within the fuel pellet. If in reality the fuel pellets did not crack, the 3D and 2D model would produce the same results as the problem would be truly axisymmetric. Since the pellets do crack in reality, the 2D model with pellet cracking will produce more accurate stress and strain results in the sheath than a 3D model without pellet cracking. The most accurate model would be a 3D model that includes pellet cracking. In addition, ELESTRES contains three creep mechanisms whereas HORSE only contains one. Moreover, the internal gas pressure is higher in the ELESTRES simulations due to the detailed evolution of fission gas production and release. At higher burnups even as the power decreases the sheath continues to plastically deform resulting in ever growing strains. Contrarily, the HORSE simulations begin to remove some of the plastic strain initially introduced due to thermal expansion. This affect is likely due to glued contact. As the fuel cools down the pellets begin to contract and impose strains in the opposite direction on the sheath. In reality the fuel could come out of contact leaving the sheath permanently deformed. However due to glued contact the fuel begins to pull the sheath with it, introducing negative strains resulting in a recovery of the deformation in the sheath as the power decreases over time. This recovery is further seen in the hoop stress in the sheath as it becomes compressive in the HORSE simulations at a burnup of approximately 200 MWh kgU^{-1}.

The overprediction of the fuel-to-sheath gap heat transfer coefficient results in under predicting the inside sheath, fuel surface and fuel centerline temperatures. The under prediction of the hoop stress and strain within the sheath due to glued contact indicate that further investigation is required into frictional contact algorithms to remove the glued contact constraint. In addition a fuel cracking model and fission gas production and release should be added to further improve

the fuel performance capabilities of the HORSE framework. All of these are proposed future work for the continued development of HORSE.

7.3 Flexural Rigidity

The flexural rigidity (also called stiffness) of a beam is a measure of its resistance against bending. A comparison of an empty sheath's flexural rigidity to that of a fuel element can determine the enhancement against bending that the pellets provide to the sheath. This can provide insight into the loading required to cause element-to-pressure tube contact or element-to-element contact within the fuel bundle. Results are calculated for three linear power ratings 30, 45, and 60 kW m^{-1} to assess the effects of low, nominal and overpower scenarios on the flexural rigidity of the fuel element for elastic and elastic-plastic models of the sheath.

To gain an understanding of the bounding limits for the flexural rigidity of a fuel element, simulations of an empty sheath made of Zircaloy-4 as well as a discrete stack of UO_2 fuel were completed. A discrete fuel stack is one where the pellets are permanently attached to one another at the lands (i.e., the two distinct surfaces representing the lands on adjacent pellets become one common surface with shared nodes). An analytical solution can be found for these two cases using the Young's modulus for the constant temperature the systems were set to (583 K) and determining the second moment of inertia from the dimensions of the cross-sectional area of the fuel sheath or fuel stack. The results of these simulations and the analytical calculations are presented in Table 7.2. In addition the flexural rigidity results obtained by Williams' ANSYS model using frictional contact or bonded (glued) contact between the pellets and the sheath are provided. [27] In Williams' simulations the same loading conditions were applied to the

element. The flexural rigidity is determined from the slope of a plot of deflection as a function of applied load as given by equation (6.2). The results for the empty sheath and discrete fuel stack obtained by HORSE are within acceptable agreement of the analytical solutions, which also indicates that the material property correlations are correctly input into the HORSE environment. The larger difference for the discrete fuel stack is misleading because the analytical solution assumes a solid cylindrical stack of fuel, whereas the calculated value is for a fuel stack containing dishes and chamfers. The presence of the dishes and chamfers produce weak spots at certain axial locations along the fuel stack because the second moment of inertia is smaller at the axial locations containing chamfers and dishes. It is expected that a fuel sheath containing fuel pellets would have a flexural rigidity that lies somewhere between the discrete fuel stack and an empty sheath, that depends upon the linear power applied to the fuel. This is confirmed by Williams' ANSYS results for both frictional and bonded contact between the pellets and the sheath. As expected pellets bonded to the sheath produce a flexural rigidity closer to that of a solid fuel stack as the pellets are unable to slide upon the inside surface of the sheath, whereas frictional contact allows tangential sliding and the flexural rigidity is approximately twice that of an empty sheath.

Table 7.2: Flexural rigidity results of HORSE simulations against analytical solutions, and ANSYS simulation results

Model	**HORSE Flexural Rigidity ($N\ m^2$)**	**Analytical/ANSYS Flexural Rigidity ($N\ m^2$)**
Empty Sheath	24.15	25.15 (Analytical)
Discrete UO_2 Stack	191.1	219.2 (Analytical)
Full Element Pellet-to-sheath Bonded	$>10^5$	149.5 (ANSYS)
Full Element Pellet-to-sheath Frictional	No Convergence	50.6 (ANSYS)

Once it was confirmed that HORSE was behaving correctly for the individual parts of a fuel element the next step was to combine them and run a full fuel element. It can be seen in the third row of Table 7.2 that the flexural rigidity reported for the HORSE full fuel element with the glued contact constraint between the pellets and the sheath is many orders of magnitude greater than predicted by ANSYS. This conclusion is drawn from the analysis of the full fuel element whose results are presented in Figure 7.7.

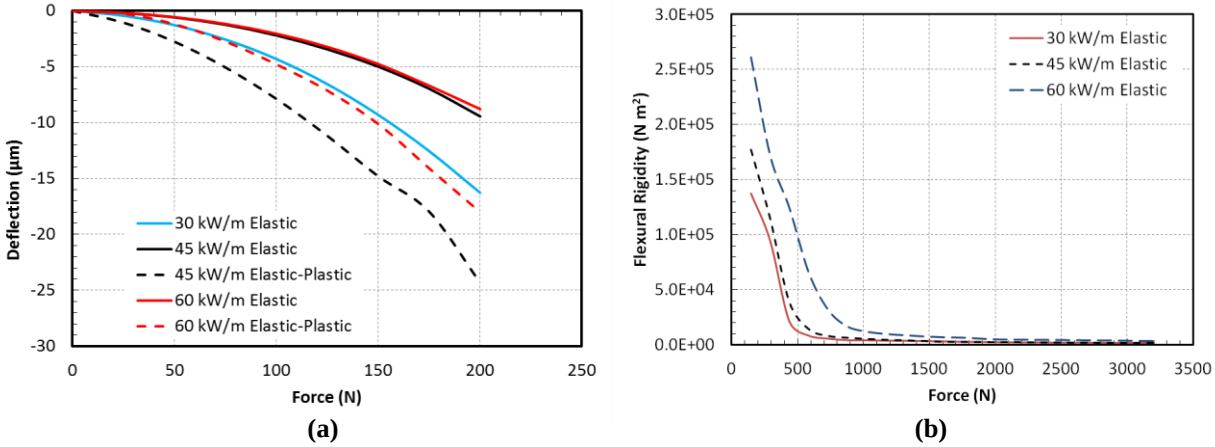


Figure 7.7: Results of the flexural rigidity analysis showing (a) deflection as a function of the applied force, and (b) the flexural rigidity as a function of force.

Figure 7.7(a) shows the maximum deflection at the element midplane as a function of the applied load. A significant issue with these results is that the slopes of the lines are nonlinear indicating that the flexural rigidity depends upon the stress state of the system which is contrary to the analytical solution given by equation (6.2). It is true that in reality some dependence on the stress state will be observed as the sheath grips the pellets, but it is not expected to be this large. The ANSYS simulations showed an approximately linear behaviour over this range of conditions. The order of the curves in the figure are as expected because the higher linear power cases deflect less as stress stiffening occurs as the contact pressure increases. In addition the models containing the bilinear plasticity model in the sheath deflect the most as they are softer

than a purely elastic material. However a deflection on the order of microns is too small, especially when the flexural rigidity is determined and compared against Williams' ANSYS solutions. The flexural rigidity can be determined based upon the analytical equation given the maximum deflection and applied load for each data point in the HORSE simulation. To examine the behaviour of the flexural rigidity a simulation was completed that ramped up to a very large load of 3200 N using the elastic sheath model. The results are presented in Figure 7.7(b). It is observed that the flexural rigidity is greater than 10^5 N m^2 for loads that could be experienced by the element in reactor. This is orders of magnitude larger than the results produced by ANSYS. As the load is increased the flexural rigidity quickly drops to a constant value but this value is still too large.

These results indicate that for contact problems, once external loads are applied that induce lateral deflections the contact algorithms appear to prevent these deflections. In order to determine if it is the glued contact between the pellets and the sheath or the frictionless contact condition between the individual pellets that is causing these results, one final simulation was completed. In this simulation a discrete fuel stack was placed within the empty sheath and the diametrical gap was removed. Since both components behave correctly on their own it is safe to assume that if the results obtained are similar to Figure 7.7 that something in the glued contact algorithm is preventing lateral deflections. If the simulation behaves correctly than it follows that the additional rigidity must be due to the frictionless contact between the pellets. Upon completion of the simulation it was observed that the results were similar to those presented in Figure 7.7. This implies that once lateral deflections are introduced glued contact seems to behave incorrectly. Delving into the details of what could be causing this phenomenon in the

code is beyond the scope of this work. The problem has been presented to the BISON development team for investigation.

7.4 Thermal Bowing

Based upon the results of the flexural rigidity analysis further investigation into the behaviour of the glued contact algorithm within MOOSE is required. The results of the flexural rigidity analysis seem to indicate that there is an issue with the glued contact algorithm once lateral deflections are introduced. A thermal bowing analysis was completed to assess the deflection of the full fuel element geometry against diametrical temperature gradients across the fuel element. This analysis can help determine if anomalous results due to glued contact apply to both mechanically induced loads as well as thermally induced loads. In the thermal bowing analysis three different temperature gradients were applied across the diameter of the element, 100, 200 and 300 K. These temperature gradients are induced by varying the bulk coolant temperature on the outside surface of the sheath after steady-state has been reached at 300 s. In the discrete UO_2 fuel stack this varying convective boundary condition was applied to the exterior surface of the fuel. The change from constant temperature to the temperature gradient occurs over one second. The simulation is continued with the temperature gradient applied for 1000 s to eliminate any transient effects. It is expected that larger temperature gradients will produce larger lateral deflections.

Similar to the flexural rigidity analysis, simulations were completed for a discrete fuel stack and empty sheath to confirm that these individual components behave as expected by comparing against analytical calculations. The maximum deflection due to thermal bowing with a uniform temperature gradient across the diameter (or thickness) of a beam along its entire length is

characterised by equation (6.4). This analytical equation depends upon the axial thermal expansion coefficient of the beam material and is assumed constant for the temperature ranges across the beam. Table 7.3 summarizes the results of the analytical solutions for a hollow sheath and a discrete fuel stack with the nodes on the land surfaces merged. It can be seen that HORSE produces results that are within a maximum relative error of 2.6 % of the analytical value for both cases. As the temperature gradient increases the relative error continually decreases because the absolute difference between the analytical solutions and the HORSE simulations are approximately constant but the maximum bow increases resulting in a lower percentage deviation. The discrepancy likely arises from the choice of axial coefficient of thermal expansion for the analytical calculations as Zircaloy and UO_2 both have coefficients of thermal expansion that vary as a function of temperature as given by equations (5.36) and (5.26) respectively. For the analytical solutions the coefficient of thermal expansion used is at the average of the two extremes of the temperature gradient applied. For example for the 100 K temperature gradient the temperature used for the coefficient of thermal expansion was 633 K.

Table 7.3: Comparison of the maximum deflection due to thermal bowing of the analytically expected values and the HORSE simulations for both the empty sheath and discrete fuel stack.

Temperature Gradient Diameter (K)	**Maximum Bow (mm)**			
	HORSE Empty Sheath	**Analytical Sheath**	**HORSE Discrete UO_2 Stack**	**Analytical Discrete UO_2 Stack**
100	1.040	1.068 (2.6%)	2.376	2.398 (0.92%)
200	2.080	2.120 (1.9%)	4.806	4.840 (0.70%)
300	3.120	3.160 (1.3%)	7.314	7.343 (0.40%)

The thermal bowing analysis on the individual components illustrates that as with the flexural rigidity analysis the material properties are coded correctly and the boundary conditions for the applied load appear to be applied as expected. Knowing this, the next step was to run a simulation for the full fuel element and compare these results against what was observed for the individual components. Figure 7.8 illustrates this comparison. It is obvious from the analytical equation given by equation (6.4) that the profile is expected to be parabolic with the maximum deflection at the midplane ($z = 0$). Figure 7.8 (a) presents the thermal bowing profile for the top surface of the sheath, and it is observed that the displacement profile is consistent with the expected analytical solution. Moreover the magnitude of the deflection is on the order of several millimeters. However, when the temperature gradient is applied to the full fuel element containing individual pellets within the sheath an unexplainable displacement profile is observed as shown in Figure 7.8 (b). At lower temperature gradients the displacement profile is not near parabolic in nature. As the temperature gradient approaches 300 K the profile begins to approach a parabolic shape. These results are not as expected. The composite beam containing individual fuel pellets within the sheath was expected to have maximum deflections due to bowing lying between the empty sheath and discrete fuel stack simulations. Thermal bowing profiles within these two bounding cases were observed by Yang [64] using constant material properties (e.g., coefficient of thermal expansion, Young's modulus, Poisson's Ratio) and frictional contact in ANSYS for varying temperature gradients.

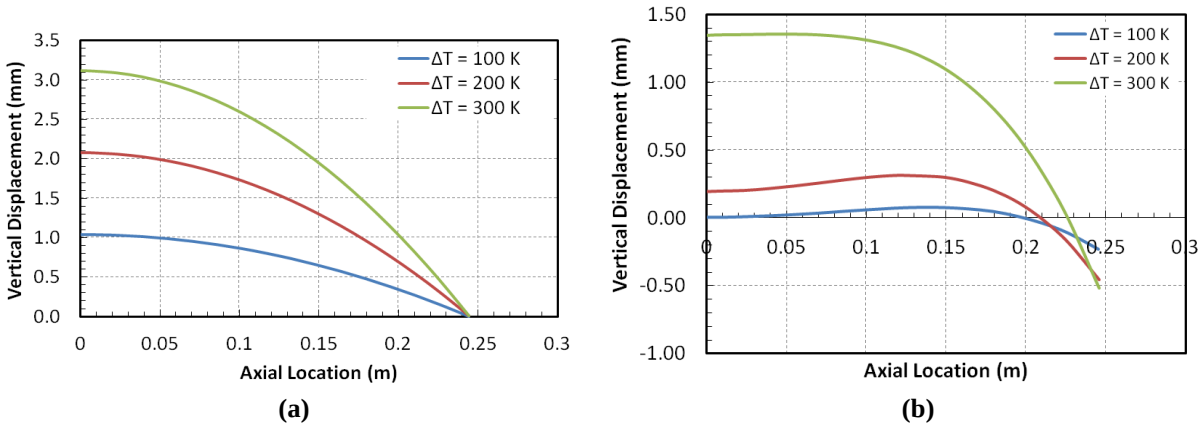


Figure 7.8: Illustration of thermal bowing profiles extracted from the top surface of the sheathing for different diametrical temperature gradients for (a) empty sheath and (b) the full fuel element including pellet-to-pellet and pellet-to-sheath contact.

Similarly to the flexural rigidity analysis the results of the thermal bowing analysis of the full fuel element seem to indicate that there is something occurring in the contact calculations either between the pellets and the sheath or at the pellet-to-pellet interfaces. Once again to test whether it is the glued contact constraint or the frictionless contact a full fuel element was constructed containing the discrete (merged node) fuel stack inside a sheath with no diametrical gap. This ensures the gap is closed from the beginning of the simulation. The results of this analysis produced less thermal bowing and very similar profiles to the lower temperature gradients observed in Figure 7.8(b). These results in addition to the flexural rigidity analysis results strongly indicate that there is something behaving incorrectly with the glued contact algorithms within the MOOSE framework once lateral deflections are introduced.

7.5 Optimization and Scalability

In computer science, there are two techniques used to obtain results more efficiently and quickly: serial and parallel optimization. Serial optimization primarily refers to how a program is coded and how memory is accessed. An example of serial optimization is how multi-dimensional

arrays are stored within memory. This storage is largely language dependent, for example Fortran accesses memory column-wise, whereas C accesses memory row-wise. This means if a program is coded in C and arrays are stored column-wise, it will have a large negative impact on overall performance [65, 66].

For computational analyses, efficiency and optimization is calculated based upon the wall-clock time. Wall-clock time is defined as the length of time measured by an individual if he or she was to measure the time on a clock hanging upon the wall. Parallel optimization is where a user uses more than one processor to run a simulation as was done for every simulation completed in this work. When running problems in parallel the wall clock time should decrease but the total CPU time of all processors will most likely increase [65]. Theoretically the shortest execution time that can be achieved is inversely proportional to the number of processors P given by the ratio

$$\frac{T_1}{P} \tag{7.1}$$

where T_1 is the wall-clock time for running the program on one processor. Therefore, running a parallelizable process on two processors should theoretically half the execution time required. Another way to measure the effectiveness of parallelization is through a quantity known as the parallel speedup defined by

$$S = \frac{T_1}{T_P} \tag{7.2}$$

where T_P designates the wall-clock time on P processors. Almost every program has portions suitable for parallelization and other portions that are not. This means that part of the program is still run in serial. As the number of processors increases the length of time for the parallel portions of the code to execute is decreased, but the amount of time for the serial portion of code

remains constant. This obstacle produces an upper limit on the expected speedup. This upper limit can be determined by Amdahl's law [66]:

$$S = \frac{1}{(f/P + (1-f))} \tag{7.3}$$

where f is the fraction of the code that is parallelizable. In the ideal case when there is no serial region in the program, $f = 1$, and the speed up is equal to the number of processors [67]. The efficiency of parallelization, E, is thus

$$E = \frac{S}{P} \tag{7.4}$$

An illustration of Amdahl's law is provided in Figure 7.9 for a range of processors and fractions of parallelizable code. The upper limit of expected speedup is clearly visible indicating that there is a point of diminishing returns when it comes to parallelization.

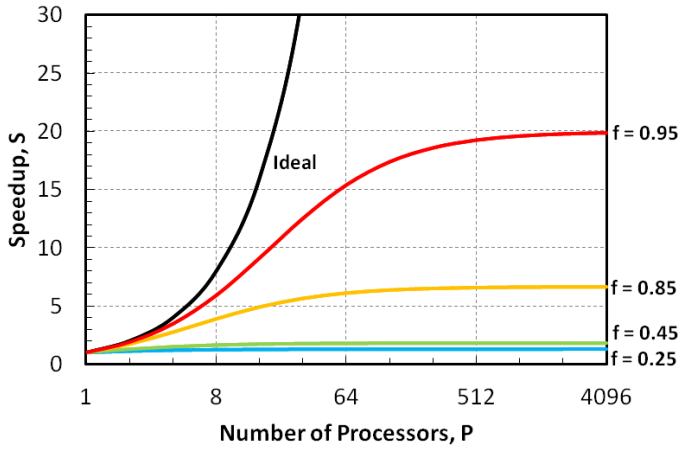


Figure 7.9: Illustrative representation of Amdahl's law showing the upper limit of speedup for a number of processors at a given percentage of parallelizable code. Adapted from **[66]**.

The parallelization of a simulation in MOOSE depends upon the components included. With the addition of uncontrolled deterministic processes such as mechanical contact, the communication

between processors can cause the scalability of the code to become random (e.g., 16 processors solving more efficiently than 32 processes). In addition the use of adaptive timestepping could cause simulations using different numbers of processors to take different timesteps resulting in a different path to the solution, which could lead to differences in the expected wall clock time. The simpler the problem to be scaled the higher an increase in speed that will be achieved. To illustrate the scalability of a HORSE mechanical contact problem a short 1.5 pellet fuel element (3 fuel pellets with symmetry) is run for a simulation time of 700 s with a constant volumetric heat generation and elastic material properties for the sheath. The results of speed up and efficiency of parallelization are presented in Figure 7.10.

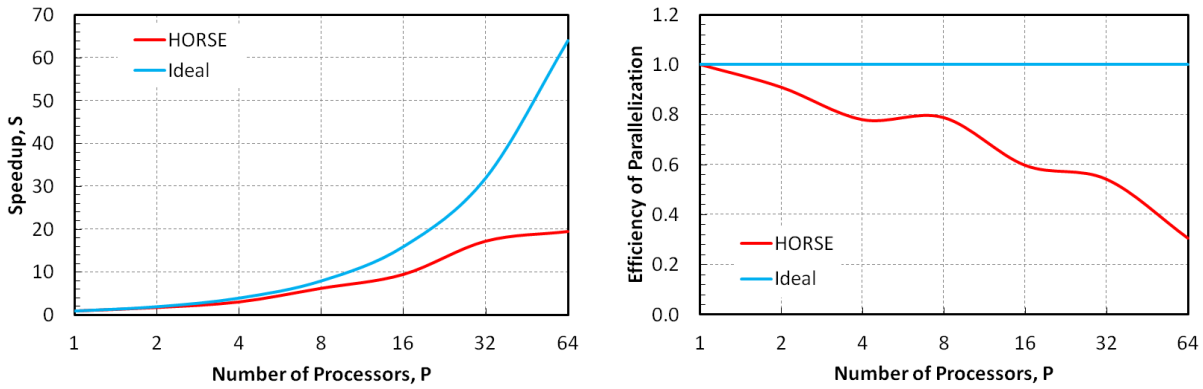


Figure 7.10: The scalability of HORSE for a simple 3 pellet glued contact model with elastic material behaviour of the sheath showing (a) speedup and (b) efficiency of parallelization.

It can be seen that as the number of processors used increases there is diminishing returns in the amount of speedup obtain as governed by Amdahl's law. This causes the efficiency of parallelization to decrease as the number of processors increase. As the deviation from the ideal case increases the efficiency decreases. Once the plateau of speedup is reached the efficiency of parallelization will approach zero. Table 7.4 presents the fraction of the code that is parallelizable for each processor. Some variation is observed due to communication between

processors during parallel computation. By taking the average of the values, an effective value of parallelizable code of 0.943 (94.3%) is obtained.

Table 7.4: Results of the HORSE scalability study

Number of Processors	Wall Clock Time (s)	Speedup	Efficiency	Fraction of Parallelizable Code
1	49279.6228	1.000	1	N/A
2	27101.8	1.818	0.909	0.900
4	15806.9401	3.118	0.779	0.906
8	7822.244	6.300	0.787	0.961
16	5160.8726	9.549	0.597	0.955
32	2846.3622	17.313	0.541	0.973
64	2519.9036	19.556	0.306	0.964

Chapter 8 - Summary and Conclusions

The purpose of this thesis was to assess the capabilities of the state of the art MOOSE computational framework developed at the Idaho National Laboratory in modeling 3D deformation mechanisms of CANDU reactor fuel elements. The reasons to undertake this endeavour include the following: licenses and modules for commercial codes such as ANSYS, Abaqus and Comsol are extremely costly and some commercial codes are incapable of adding advanced material models or nuclear phenomena (e.g., ANSYS). The details of the mathematical algorithms used within MOOSE based applications and the models used within HORSE were presented. The conclusions of the analyses completed in this work are summarized below:

1) The 2D and 3D contact analyses provided preliminary insight into the behaviour of the penalty and kinematic enforcement methods of the glued and frictionless contact models as well as penalty enforcement of coulomb frictional contact. The contact algorithms behave as expected with respect to contact pressure and penetration as a function of the penalty factor. Frictional contact results indicate that the penetration is too high for penalty factors in which convergence was achieved. It may be possible to improve the convergence of the frictional contact model via modification to the algorithm or by adjusting the solver settings.

2) When comparing to ELESTRES for fuel performance parameters the glued contact constraint leads to over predictions of the contact pressure between the pellets and sheath at higher burnups, resulting in lower than expected fuel temperatures. This is because glued contact causes the pellets and sheath to remain in contact for the duration of the simulation once contact has been initiated which overestimates the fuel-to-sheath heat transfer coefficient. At higher burnups as the amount of fission gas released to the plenum/gap regions the higher the

gas pressure becomes and the contact pressure between the pellet and sheath should decrease which would significantly lower the pellet-to-sheath heat transfer coefficient but glued contact prevents the contact pressure from decreasing. Overall these results are as expected due to the glued contact constraint.

3) HORSE predicted the flexural rigidity and maximum deflection due to thermal bowing for the individual components of a fuel element (i.e., empty sheath and discrete fuel stack) that agreed reasonably well with analytical solutions. The discrepancies are attributed to the slight deviation in geometry analysed for flexural rigidity and the difficulty in choosing a correct single value for the coefficient of thermal expansion for the analytical calculations of thermal bowing. The results of full fuel element with individual pellets within the sheath produce anomalous results. It is the hypothesis that something within the glued contact algorithms currently employed in MOOSE is preventing lateral deflection due to external mechanical and thermal loading. Further investigation beyond the scope of this work is required. The results have already been presented to the BISON team and they are currently investigating.

In conclusion, this thesis was concerned with examining the capabilities of the MOOSE computational framework in modeling 3D deformation mechanisms present in horizontally oriented nuclear fuel rods. At the time of writing this is the only known work in Canada that uses the MOOSE framework and is believed to be the first attempt at modeling many multi-body contacts. The contact algorithms behave as expected for isolated contact analyses and fuel performance simulations but seem to breakdown when lateral deflections are introduced. The MOOSE framework is a very powerful modeling tool and further investigation is required to examine the intricacies and complexity of the code to be able to make the best use of this tool for model development of interest to the Canadian nuclear industry.

Chapter 9 - Recommendations

The work completed in this thesis is a preliminary analysis of the capabilities of the MOOSE framework for modeling nuclear fuel of interest to Canadians. A list of recommendations is provided below on how to either extend the HORSE framework for CANDU fuel or to use the MOOSE framework for modeling different fuel types of interest to the Canadian nuclear industry.

1) The first recommendation to extend the capabilities of HORSE is to further investigate the contact algorithms available within MOOSE. Limited sensitivity analyses were completed on the penalty enforcement of frictional contact and further analysis is required. Perhaps changing the mesh density may lead to more robust simulations. Investigation into the kinematic enforcement of frictional contact may lead to more stable convergence. Kinematic enforcement was not analyzed in this work because convergence could be orders of magnitude more computationally expensive than penalty enforcement due to the many slip iterations that are required.

2) Additional analysis of the executioner block is suggested. This block defines the preconditioning and PETSc solver options to be used for the simulation. The amount of options in the PETSc's user's manual is vast and an investigation of the variety of options available is outside the scope of this work. Different combinations of PETSc options and preconditioning could lead to more efficient and robust solutions. Simulations that diverged may begin to converge as effective preconditioning is required for the JFNK method. A discussion with a PETSc developer indicated that a preconditioner specifically designed for improving convergence of contact simulations is currently under development.

3) The addition of more nuclear phenomena and material properties such as fission gas production and release, pellet cracking, and fuel creep would lead to more realistic fuel performance behaviour leading to an all-encompassing fuel performance and 3D deformation tool. Additional sheath creep mechanisms including dislocation, transition, thermal, irradiation induced and primary creep should be added to the HORSE framework.

4) Once a detailed model of a full fuel element is obtained including frictional contact, the next logical step is to begin extending the model to examine deformation of an entire CANDU fuel bundle which is of interest to the Canadian Nuclear Safety Commission (CNSC), Canada's nuclear regulator. To avoid large computational requirements from full fuel bundle simulations including all pellet-to-sheath, pellet-to-pellet, and element-to-element contacts, investigation into developing a lookup table may be warranted. The values within the table could correspond to the effective material properties for a composite beam consisting of the fuel pellets and sheath. Then the behaviour of a full fuel bundle can be run using beam elements by running a single fuel element in full detail.

5) The application of the MOOSE code does not have to be restricted to further developing the HORSE application. Instead of focusing on urania fuel, it is of interest to develop computational models for thoria-based fuels for Canada's contribution to the Generation IV nuclear reactor program. Canada's contribution to this program is the Super Critical Water-cooled Reactor (SCWR) which has a proposed vertical orientation for the fuel. The vertical orientation will remove the rigid body motion issues encountered that currently requires a glued contact constraint. In addition, BISON is typically used for 2D axisymmetric simulations of LWR fuel rods and thus simply adding the material properties and behaviours for thoria based fuels would allow simulation of proposed SCWR fuel designs.

References

[1] World Nuclear Association, March 2013. [Online]. Available: http://world-nuclear.org/info/Current-and-Future-Generation/Plans-For-New-Reactors-Worldwide/.

[2] CANDU Energy Inc., 2011. [Online]. Available: http://www.candu.com/en/home/candureactors/default.aspx.

[3] International Atomic Energy Agency, "Assessment and management of ageing of major nuclear power plant components important to safety: CANDU pressure tubes," Vienna, 1998.

[4] S. Xu, D. C. Wen and S. D. Yu, "Mixed linear complementary formulation of lateral contacts among a system of parallel beams," *Communications in Nonlinear Science and Numerical Simulation,* vol. 13, pp. 2308-2319, 2008.

[5] K. Lassmann, "The Structure of Fuel Element Codes," *Nuclear Engineering and Design,* vol. 57, no. 1, pp. 17-39, April 1980.

[6] R. Williamson, S. Novascone, J. Hales, B. Spencer, G. Pastore and D. Perez, *BISON Workshop,* December 2012.

[7] M. J. F. Notley, "ELESIM: A computer code for predicting the performance of nuclear fuel elements," *Journal of Nuclear Technology,* vol. 44, pp. 445-450, August 1979.

[8] H. H. Wong, E. Alp, W. R. Clendening, M. Tayal and L. R. Jones, "ELESTRES: A finite element fuel model for normal operating conditions," *Journal of Nuclear Technology,* vol. 57, pp. 203-212, May 1982.

[9] A. F. Williams, "The Eloca Fuel Modelling Code: Past, Present And Future," in *9th International CNS Conference on CANDU Fuel*, Belleville, 2005.

[10] K. J. Geelhood, W. G. Luscher, C. E. Beyer, D. J. Senor, M. E. Cunningham, D. D. Lanning and H. E. Adkins, "Predictive Bias and Sensitivity in NRC Fuel Peformance Codes," Richland, WA, 2009.

[11] H. S. Aybar and P. Ortego, "A review of nuclear fuel performance codes," *Progress in Nuclear Energy,* vol. 46, no. 2, pp. 127-141, 2005.

[12] K. Lassmann, "TRANSURANUS: a fuel rod analysis code ready for use," *Journal of Nuclear Materials,* vol. 188, pp. 295-302, 1992.

[13] G. Rossiter, "Development of the ENIGMA Fuel Performance Code for whole Core Analysis and Dry Storage Assessments," *Nuclear Engineering and Technology,* vol. 43, no. 6, pp. 489-498, December 2011.

[14] A. C. Marino, E. J. Savino and S. Harriague, "BACO (BArra COmbustible) code version 2.20: a thermo-mechanical description of a nuclear fuel rod," *Journal of Nuclear Materials,* vol. 229, pp. 155-168, December 1996.

[15] S. Hanawa, J. Ohgiyanagi and M. Suzuki, "Verification of FEMAXI-7 code by using irradiation test in Halden reactor for He-pressurization effect on FGR of BWR fuels under power transient," *Journal of Nuclear Science and Technology,* vol. 49, no. 5, pp. 516-525, May 2012.

[16] W. Lyon, R. Montgomery, A. Zangari, D. Sunderland, Y. Rashid and R. Dunham, "Fuel Performance Analysis Capability in FALCON," Palo Alto, 2002.

[17] F. Bentejac and N. Hourdequin, "TOUTATIS: an application of the CAST3M finite element code for PCI three-dimensional modeling," in *Pellet-Clad Interaction in Water Reactor Fuels*, Aix-en-Provence, 2004.

[18] J. Sercombe, B. Michel, G. Thouvenin, B. Petiprez, R. Chatelet, D. Leboulch and C. Nonon, "Multi-dimensional modeling of PCMI during base irradiation and ramp testing with ALCYONE V1.1," in *Top Fuel*, Paris, 2009.

[19] D. Morgan, "A Thermomechanical Model of CANDU Fuel," Kingston, 2007.

[20] A. Prudil, B. J. Lewis, P. K. Chan and J. J. Baschuk, "FAST: A combined NOC and Transient fuel Model for CANDU Fuel," in *12th International Conference on CANDU Fuel*, Kingston, 2013.

[21] A. A. Prudil, "FAST: A Fuel And Sheath Modeling Tool for CANDU Reactor Fuel," Kingston, 2013.

[22] K. T. Clarno, B. Philip, W. K. Cochran, R. S. Sampath, S. Allu, P. Barai, S. Simunovic, M. Berrill, L. J. Ott, S. Pannala, G. A. Dilts, B. Mihaila, G. Yesilyurt, J. H. Lee and J. E. Banfield, "The AMP (Advanced MultiPhysics) Nuclear Fuel Performance code," *Journal of Nuclear Engineering and Design,* vol. 252, pp. 108-120, July 2012.

[23] K. Shaheen, "A Mechanistic Code for Intact and Defective Fuel Element Performance," Kingston, 2011.

[24] M. Tayal, "Modelling the Bending/Bowing of Composite Beams such as Nuclear Fuel: The

BOW Code," *Nuclear Engineering and Design,* vol. 116, pp. 149-159, 1989.

[25] Dassault Systemes, 2013. [Online]. Available: http://www.3ds.com/products-services/simulia/portfolio/abaqus/abaqus-portfolio/abaqus-multiphysics/?xtmc=abaqus_multiphysics&xtcr=1.

[26] J. S. Bell and B. J. Lewis, "CANDU fuel bundle deformation modelling with COMSOL multiphysics," *Journal of Nuclear Engineering and Design,* vol. 250, pp. 134-141, May 2012.

[27] A. F. Williams and S. Yatabe, *An ANSYS based 3-Dimensional Model of a CANDU fuel pin,* 2014.

[28] M. R. McCluskey, "A Three Dimensional Finite Element Analysis of a CANDU Fuel Element," Kingston, 2012.

[29] D. V. Hutton, "Basic Concepts of the Finite Element Method," in *Fundamentals of Finite Element Analysis*, New York, McGraw-Hill, 2004, pp. 1-18.

[30] "Chapter 18: Class Templates," in *Beginning C++: The Complete Lanugage*, Birgmingham, Wrox Press Ltd., 1998, pp. 707-753.

[31] libMesh Developers, May 2013. [Online]. Available: http://libmesh.sourceforge.net/.

[32] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. Curfman McInnes, B. Smith and H. Zhang, PETSc Users Manual Revision 3.3, Chicago: Computer Science Division, Argonne National Laboratory, 2012.

[33] Sandia National Laboratory, 2013. [Online]. Available: http://trilinos.sandia.gov/.

[34] N. Stricker, September 2011. [Online]. Available: https://inlportal.inl.gov/portal/server.pt?open=514&objID=1269&mode=2&featurestory=DA_582160.

[35] D. Gaston, C. Permann, D. Andrs, J. Peterson and J. Miller, "MOOSE Training Workshop," January 2013. [Online]. Available: http://www.mooseframework.org/static/media/uploads/docs/main.pdf.

[36] D. A. Knoll and D. E. Keyes, "Jacobian-free Newton-Krylov methods: a survey of approaches and applications," *Journal of Computational Physics,* vol. 193, pp. 357-397, 2004.

[37] V. Mousseau, "Implicitly balanced solution of the two-phase flow equations coupled to nonlinear heat conduction," *Journal of Computational Physics,* vol. 200, pp. 104-132, 2004.

[38] V. A. Mousseau, D. A. Knoll and W. J. Rider, "Physics-Based Preconditioning and the Newton-Krylov Method for Non-equilibrium Radiation Diffusion," *Journal of Computational Physics,* vol. 160, pp. 743-765, 2000.

[39] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. Curfman McInnes, B. Smith and H. Zhang, "4.4 Preconditioners," in *PETSc Users Manual Revision 3.3,* Chicago, Computer Science Division, Argonne National Laboratory, 2012, pp. 78-85.

[40] R. D. Cook., D. Malkus, M. Plesha and R. Witt, Concepts and Applications of Finite Element Analysis, John Wiley and Sons, 2002.

[41] J. D. Hales, D. M. Novascone, G. Pastore, D. M. Perez, B. W. Spencer and R. L. Williamson, BISON Theory Manual, Idaho Falls: Fuels Modeling & Simulation Department, 2013.

[42] B. Spencer, J. Hales, S. Novascone, D. Perez and R. Williamson, "Frictional Contact in Nuclear Fuel Simulations," in *USNCCM12,* Raleigh, 2013.

[43] J. R. Lamarsh and A. J. Baratta, "Interaction of Radiation with Matter," in *Introduction to Nuclear Engineering,* Upper Saddle River, Prentice Hall, 2001, pp. 52-116.

[44] J. A. L. Robertson, "$\int k d\theta$ in Fuel Irradiations," Chalk River, 1959.

[45] J. E. Suich and H. C. Honeck, "The HAMMER System: Heterogeneous Analysis by Multi-Group methods of Exponentials and Reactors," 1967.

[46] R. L. Williamson, J. D. Hales, S. R. Novascone, M. R. Tonks, D. R. Gaston, C. J. Permann, D. Andrs and R. C. Martineau, "Multidimensional multiphysics simulation of nuclear fuel behavior," *Journal of Nuclear Materials,* vol. 423, pp. 149-163, 2012.

[47] J. K. Fink, "Thermophysical properties of uranium dioxide," *Journal of Nuclear Materials,* vol. 279, pp. 1-18, November 1999.

[48] C. Ronchi, G. J. Hyland, M. Sheindlin and M. Musella, "Thermal Conductivity of Uranium Dioxide up to 2900 K from Simultaneous Measurement of the Heat Capacity and Thermal Diffusivity," *Journal of Applied Physics,* vol. 85, no. 2, pp. 776-789, 1999.

[49] P. G. Lucuta, H. Matzke and I. J. Hastings, "A pragmatic approach to modelling thermal conductivity of irradiated UO2 fuel: review and recommendations," *Journal of Nuclear Materials,* vol. 232, pp. 166-180, May 1996.

[50] A. L. Loeb, "Thermal Conductivity: A Theory of Thermal Conductivity of Porous Materials," *Journal of the American Ceramic Society,* vol. 37, pp. 96-99, 1954.

[51] Idaho National Engineering and Environmental Laboratory, "SCDAP/RELAP5/MOD3.2 Code Manual Volume IV: MATPRO - A Library of Materials Properties for Light-Water-Reactor Accident Analysis," Idaho Falls, 1997.

[52] I. J. Hastings and L. E. Evans, "Densification Algorithm for Irradiated UO2 Fuel," *Journal of the American Ceramic Society,* vol. 62, pp. 217-218, 1979.

[53] D. Olander, Fundamental Aspects of Nuclear Reactor Fuel Elements, US Department of Energy, 1976.

[54] J. F. Kerrisk and D. G. Clifton, "Smoothed values of the enthalpy and heat capacity of UO2," *Journal of Nuclear Technology,* vol. 16, 1972.

[55] Idaho National Laboratory, "MATPRO - Version 09, A Handbook of Materials Properties for Use in the Analysis of Light Water Reactor Fuel Rod Behaviour," 1976.

[56] J. A. Walsworth and H. E. Sills, "High Temperature Transient Fuel Performance Modelling, ELOCA.Mk4 (User's Manual, Program Description and Validation)," 1984.

[57] A. M. Ross and R. L. Stoute, "Heat Transfer Coefficient Between UO2 and Zircaloy-2," 1962.

[58] H. Von Ubisch, S. Hall and R. Srivastav, "Thermal Conductivities of Mixtures of Fission Product Gases with Helium and Argon," in *International Conference on Peaceful Uses of Atomic Energy,* Geneva, 1958.

[59] L. Friend and S. B. Adler, Transport Properties of Gases, Evanston: Northwestern University Press, 1958.

[60] J. M. Lenoir and e. al., "Measurement and Correlation of Thermal Conductivities of Gases at High Pressures," *Chemical Engineering Progress,* vol. 49, no. 539, 1953.

[61] H. E. Sills and R. A. Holt, "Predicting High-Temperature Transient Deformation from Microstructural Models," in *Zirconium in the Nuclear Industry (Fourth Conference),* 1979.

[62] T. A. Cunning, "A Statistical Approach to Nuclear Fuel Design and Performance," Kingston, 2013.

[63] W. C. Young, Roark's Formulas for Stress and Strain, 6th Edition ed., New York: McGraw-Hill Professional, 1989.

[64] H. Yang, "Thermally Induced Deformations in Nuclear Fuel Elements," Toronto, 2004.

[65] R. van der Pas, *Sun Application Tuning Seminar - Application Performance Optimization on Sun Systems,* 2005.

[66] M. H. A. Piro, "Computation of Thermodynamic Equilibria Pertinent to Nuclear Materials in Multi-Physics Codes," Kingston, 2011.

[67] B. Chapman, G. Jost and R. van der Pas, "2.6 Performance Considerations," in *Using OpenMP: portable shared memory parallel programming*, Cambridge, The MIT Press, 2008, pp. 33-34.

Appendices

Appendix A – Flux Depression Parameters

Table A.1 lists the flux depression parameters used in the radial flux depression and heat generation correlation given by equation (5.11). These values depend upon the initial pellet radius, the enrichment of the fuel, and the average fuel burnup. The values listed are for fuel with 0.71% enrichment and a nominal initial pellet radius of 6.075 mm. Linear interpolation is used to determine the values of κ_{flux}, β_{flux}, and λ_{flux} when the average burnup is between the tabulated values. These values are specific to the neutron spectrum of a CANDU reactor.

Table A.1: Flux depression parameters used in heat generation term of the heat conduction equation

Average Burnup (MWh kgU^{-1})	Kappa (m^{-1})	Beta	Lambda (m^{-1})
0	89.94	0	100
48	98.13	0.3731	8035.63
96	103.93	0.6247	8391.87
144	108.1	0.8058	8514.39
192	111.9	0.9394	8552.43
240	115.56	1.0341	8532.3
288	119.14	1.1049	8524.06
336	122.17	1.1523	8506.05
384	124.47	1.1831	8481.78
432	126.23	1.2038	8469.16
480	127.5	1.2173	8461.99
528	128.23	1.2226	8445.24
576	128.75	1.2281	8445.15
624	128.94	1.2286	8434.82
672	129.05	1.2291	8437.16
720	129.05	1.228	8439.19
768	128.88	1.2243	8431.15
816	128.73	1.2209	8424.98
864	128.55	1.2169	8422.47
912	128.34	1.2141	8419.65
960	128.1	1.2081	8411.84

Appendix B – Burnup Derivation and Gaseous Fuel Swelling Conversion

This appendix includes the derivation of the local and average burnup equations, (5.14) and (5.15), respectively, as well as the conversion of the gaseous fission product swelling correlation given in MATPRO [51] to the version used in this work given by equation (5.31).

Burnup Derivation

It is known that the volumetric heat generation within the fuel is given by $Q_{fission}$ in equation (5.11) and is in units of W m^{-3}. The generated heat is produced from the uranium within the fuel. Thus the amount of heat production per kilogram of uranium can be determined by

$$\frac{Q_{fission}}{\rho_U} \tag{B.1}$$

where ρ_U is the density of uranium in kg m^{-3}. The standard unit of burnup used in the CANDU industry is MWh kgU^{-1}. To obtain this unit equation (B.1) can be modified by dividing by 3600 seconds and converting $Q_{fission}$ from W to MW as given by:

$$\frac{Q_{fission}}{3600 \times 10^6 T_f \rho_U} \tag{B.2}$$

Equation (B.2) has units of MWh kgU^{-1} s^{-1} which is equivalent to burnup per unit time. One must also take into account the ratio of the thermal power to the fuel to the fission power which is denoted by T_f (default is 0.925 used by ELESTRES). Therefore, equation (B.2) becomes

$$\frac{dBu}{dt} = \frac{Q_{fission}}{3600 \times 10^6 T_f \rho_U} \tag{B.3}$$

which is equivalent to equation (5.14). The last step is to replace the density of uranium to a term that contains the density of uranium dioxide. This calculation is simply finding the

percentage of the density of uranium dioxide that is taken by the uranium atoms in the fuel and is completed by using the molar masses of uranium and oxygen, which yields

$$\rho_U = \left(\frac{238\rho_{UO_2}}{238+32}\right) \tag{B.4}$$

Substituting equation (B.4) into equation (B.3) yields the equation used in the source code for local burnup rate. The equation is integrated over time to obtain the local burnup for that quadrature point.

$$\frac{dBu}{dt} = \frac{Q_{fission}}{3.6\times10^9 T_f\left(\frac{238\rho_{UO_2}}{238+32}\right)} \tag{B.5}$$

To integrate equation (B.5) over the volume assuming a right cylindrical stack of fuel the approximation from equation (5.6) can be used which states

$$P_{lin} = 2\pi\int_0^{P_r} Q_{fission} r dr \tag{B.6}$$

The average burnup rate is given by

$$\frac{dBu_{avg}}{dt} = \frac{\int_0^{P_r} Q_{fission} r dr}{(3.6\times10^9)T_f\rho_U} \tag{B.7}$$

Rearranging equation (B.6) and substituting into equation (B.7) gives

$$\frac{dBu_{avg}}{dt} = \frac{P_{lin}}{2\pi(3.6\times10^9)T_f\rho_U} \tag{B.8}$$

Factoring out the length component of the density requires the initial mass of uranium to be known. This is the equation used in the source code for the average burnup rate as given by equation (5.15).

$$\frac{dBu_{avg}}{dt} = \frac{P_{lin}L}{(3.6\times10^9)T_f M_U} \tag{B.9}$$

where L is the length of the fuel stack in m and M_U is the initial mass of uranium in the element.

Gaseous Fission Product Swelling Conversion

MATPRO [51] gives the following equation for the fuel swelling to gaseous fission products at temperatures less than 2800 K:

$$\frac{d(\Delta V_{GFP}/V)}{dt} = 8.8(10^{-56})(2800 - T)^{11.73} e^{[-0.0162(2800-T)-8.0(10^{-27})Bu]} \frac{dBu}{dt} \qquad \text{(B.10)}$$

where Bu is in units of fissions m^{-3}. Since the burnup is calculated in MWh kgU^{-1} it makes sense to convert the units of fissions m^{-3} to MWh kgU^{-1} thereby altering equation (B.10). Using the characteristic energy per fission of 200 MeV per fission one fission per unit volume is equivalent to

$$\frac{1 \text{ fission}}{\text{m}^3} = \frac{\text{fission}}{200 \cancel{\text{MeV}}} \times \frac{\cancel{\text{MeV}}}{1.602 \times 10^{-19} \cancel{\text{MJ}}} \times \frac{3600 \cancel{\text{s}}}{1 \cancel{\text{h}}} \times \rho_U \left[\frac{\cancel{\text{kgU}}}{\text{m}^3}\right] \times Bu \left[\frac{\cancel{\text{MWh}}}{\cancel{\text{kgU}}}\right] \qquad \text{(B.11)}$$

The density of uranium is given by equation (B.4) which simplifies to

$$\rho_U = 0.88148 \rho_{UO_2} \qquad \text{(B.12)}$$

Therefore if you have a burnup in fissions per unit volume the conversion to MWh kgU^{-1} becomes

$$\frac{\text{fissions}}{\text{m}^3} = 9.9043 \times 10^{19} \rho_{UO_2} Bu \qquad \text{(B.13)}$$

where Bu is in units of MWh kgU^{-1}. Substituting equation (B.13) into equation (B.10) yields the equation in the source code given by equation (5.31).

$$\frac{d(\Delta V_{GFP}/V)}{dt} = 8.716(10^{-36})(2800 - T)^{11.73} e^{[-0.0162(2800-T)-7.92344(10^{-7})\rho Bu]} \rho \frac{dBu}{dt} \qquad \text{(B.14)}$$

where Bu is in units of MWh kgU^{-1} and $\rho = \rho_{UO_2}$ as required.

Appendix C – Bilinear Plasticity Data

Table C.1 lists the data points used to generate the bilinear family of curves in Figure 5.1. The values for K, n, and m are derived from their formulations in MATPRO [51]. The equations for these parameters are not included here for brevity. The values for the ultimate tensile stress, yield stress, and strains at yield and tensile failure are calculated from equation (5.74), and equations (5.76) through (5.78). The tangent modulus is then determined from equation (5.75).

Below is a list of the abbreviations used in the table:

E = Young's Modulus

K = Strength Coefficient

m = strain rate sensitivity exponent

n = strain hardening exponent

UTS = Ultimate Tensile Strength

TM = Tangent Modulus

Table C.1: Data used for the sheath bilinear plasticity model.

Temperature (K)	E (MPa)	K (MPa)	m	n	UTS (MPa)	Yield Stress (MPa)	Strain at Yield	Strain at UTS	TM (Mpa)
290	97429.0000	1074.3180	0.02	0.098806973	853.0237	655.3995	0.00673	0.09687	2192.3499
400	90840.0000	943.6889	0.02	0.1137432	735.3068	525.1349	0.00578	0.11151	1987.7785
500	84850.0000	799.1870	0.02	0.10945	625.9639	450.4529	0.00531	0.10730	1720.7786
600	78860.0000	640.8737	0.02	0.0940808	512.1410	388.7896	0.00493	0.09224	1412.8625
700	72870.0000	479.1142	0.02	0.0733884	394.9645	321.8208	0.00442	0.07195	1083.0801
750	69875.0000	400.1811	0.023845083	0.06284375	335.8085	283.0794	0.00405	0.06138	919.7639
800	66880.0000	216.6699	0.058121855	0.0531256	184.8315	157.0804	0.00235	0.05021	579.8542
900	60890.0000	85.0949	0.13357	0.0390452	74.6080	65.1501	0.00107	0.03444	283.3875
1000	54900.0000	43.6083	0.1556	0.0369	38.4039	33.1739	0.00060	0.03193	166.9468
1100	48910.0000	112.4104	0.17763	0.0524428	95.4859	80.3106	0.00164	0.04453	353.8150
1135	46871.2500	98.0649	0.1853405	0.05719881	82.4544	67.4462	0.00144	0.04826	320.5772
1200	43800.0000	83.7195	0.19966	0.07344881	68.1911	50.9702	0.00116	0.06122	286.7241
1255	41201.2500	4.1042	0.2117765	0.08719881	3.2626	1.7020	0.00004	0.07196	21.7006
1300	39075.0000	3.8015	0.22169	0.09844881	2.9667	1.3863	0.00004	0.08058	19.6208
1400	34350.0000	3.2533	0.24372	0.12344881	2.4461	0.8824	0.00003	0.09926	15.7583
1500	29625.0000	2.8410	0.26575	0.14844881	2.0668	0.5662	0.00002	0.11728	12.7968
1600	24900.0000	2.5206	0.28778	0.1734488	1.7802	0.3658	0.00001	0.13469	10.5029
1700	20175.0000	2.2480	0.30981	0.1734488	1.5831	0.3329	0.00002	0.13242	9.4423
1800	15450.0000	1.9794	0.33184	0.1734488	1.3899	0.3018	0.00002	0.13023	8.3562
1900	10725.0000	1.6707	0.35387	0.1734488	1.1698	0.2654	0.00002	0.12811	7.0608
2000	6000.0000	1.2780	0.3759	0.1734488	0.8923	0.2168	0.00004	0.12606	5.3603
2100	1275.0000	0.7573	0.39793	0.1734488	0.5273	0.1593	0.00012	0.12408	2.9688
2120	330.0000	0.6339	0.402336	0.1734488	0.4411	0.1706	0.00052	0.12369	2.1965

Appendix D – Example Trelis Journal File

The file attached below and beginning on the next page is the Trelis journal file used to generate the geometry and mesh used in the full fuel element analyses described in section 6.2. The pound symbol (#) indicates comments and or Aprepro commands. Aprepro commands are only used in the variables section for the critical dimensions and calculated values. The commands are encapsulated within the curly braces ({ and }). Although Trelis supports nodesets and sidesets as different entities, meaning there can be a sideset 1 and a nodeset 1, MOOSE does not differentiate between the two. MOOSE looks at the boundary number regardless if it is a sideset or nodeset. Therefore unique numbers are required and sidesets begin at 1 and continue as 2, 3, and so on, whereas nodesets begin numbering as 1001 and continue as 2001, 3001, and so on.

In addition, the convention used for naming the fuel and front backs can be understood from Figure 6.5 in Section 6.2. In the figure, the right side surface of the pellet is considered the fuel front, and the left side is considered the fuel back. The number on the pellet indicates the pellet number, yielding the sideset names FuelFront1 and FuelBack1 for pellet one as an example.

```
## This journal file creates a 3D representation of a full CANDU
## fuel element containing 31 pellets with dishes and chamfers.
## Using this journal file, a full model, half model and quarter
## model can be generated.  The sheath and endcaps are
## included
## By: Kyle Gamble


reset
undo on
## =========
## VARIABLES
## =========

# Critical Dimensions:
# --------------------
# {number_pellets=31}
# {pellet_radius=6.105}
# {pellet_height=15.5}
# {sheath_outer_radius = 6.551}
# {sheath_inner_radius = 6.15}
# {dish_depth=0.2}
# {dish_radius=0.8 * pellet_radius}
# {sphere_move_down=-sphere_move_up}
# {chamfer_axial=0.066}
# {chamfer_radial=0.63}

# Calculated values:
# ------------------
# {sphere_radius=(dish_depth^2 + dish_radius^2) / (2 * dish_depth)}
# {sphere_move_up=pellet_height/2 + sphere_radius - dish_depth}
# {sphere_move_down=-sphere_move_up}
# {sheath_height = number_pellets*pellet_height + 2}
# {sheath_endcap = number_pellets*pellet_height + 7.1}
# {pellet_move_up = pellet_height}
# {pellet_move_down = -pellet_height}

## ===============
## CREATE FUEL GEOMETRY
## ===============

# Create pellet cylinder:
create cylinder radius {pellet_radius} height {pellet_height}

# Create first dish:
create sphere radius {sphere_radius}
volume 2 move 0 0 {sphere_move_up}
subtract volume 2 from volume 1

# Create second dish:
create sphere radius {sphere_radius}
volume 3 move 0 0 {sphere_move_down}
subtract volume 3 from volume 1

# Create chamfers:
tweak curve 1 chamfer radius {chamfer_radial} {chamfer_axial}
tweak curve 2 chamfer radius {chamfer_radial} {chamfer_axial}
```

```
webcut volume all with plane yplane
webcut volume all with plane xplane
webcut volume all with plane zplane
delete volume 7 8 9 10
merge all

## =============
## MESH FUEL
## =============

curve 90 56 76 101 57 91 73 103 104 72 92 60 interval 16
curve 9 45 20 46 84 83 18 17 interval 1
curve 113 124 114 137 interval 14
curve 47 55 75 82 interval 8

mesh surface 31 32 35 51 52 54 41 44 45 56 57 58
# Mesh volume:
mesh volume 1 4 5 6
volume 1 4 5 6 copy reflect z
merge all

##==================
## CREATE SHEATH GEOMETRY
##==================
create cylinder radius {sheath_outer_radius} height {sheath_endcap}
create cylinder radius {sheath_inner_radius} height {sheath_height}
subtract volume 16 from volume 15
webcut volume 17 with zplane
delete volume 17
webcut volume 18 with plane surface 127
webcut volume 18 with cylinder radius {sheath_inner_radius} axis z
webcut volume 18 19 20 with xplane
webcut volume 18 19 20 21 22 23 with yplane
merge all

##==================
## MESH SHEATH GEOMETRY
##==================
curve 287 239 336 237 interval 403
curve 236 286 240 337 interval 3
curve 212 319 216 270 interval 3
curve 351 309 305 314 interval 10
curve 325 324 276 274 285 284 330 332 interval 16

mesh volume 18 19 20 21 22 23 24 25 26 27 28 29
volume 18 19 20 21 22 23 24 25 26 27 28 29 copy reflect z
merge all

##==============
##MAKE MULTIPLE PELLETS
##==============

volume 1 4 5 6 11 12 13 14 copy move 0 0 {pellet_move_up}
volume 1 4 5 6 11 12 13 14 copy move 0 0 {pellet_move_down}
volume 42 43 44 45 46 47 48 49 copy move 0 0 {pellet_move_up}
volume 50 51 52 53 54 55 56 57 copy move 0 0 {pellet_move_down}
```

```
volume 58 59 60 61 62 63 64 65 copy move 0 0 {pellet_move_up}
volume 66 67 68 69 70 71 72 73 copy move 0 0 {pellet_move_down}
volume 74 75 76 77 78 79 80 81 copy move 0 0 {pellet_move_up}
volume 82 83 84 85 86 87 88 89 copy move 0 0 {pellet_move_down}
volume 90 91 92 93 94 95 96 97 copy move 0 0 {pellet_move_up}
volume 98 99 100 101 102 103 104 105 copy move 0 0 {pellet_move_down}
volume 106 107 108 109 110 111 112 113 copy move 0 0 {pellet_move_up}
volume 114 115 116 117 118 119 120 121 copy move 0 0 {pellet_move_down}
volume 122 123 124 125 126 127 128 129 copy move 0 0 {pellet_move_up}
volume 130 131 132 133 134 135 136 137  copy move 0 0 {pellet_move_down}
volume 138 139 140 141 142 143 144 145 copy move 0 0 {pellet_move_up}
volume 146 147 148 149 150 151 152 153  copy move 0 0 {pellet_move_down}
volume 154 155 156 157 158 159 160 161 copy move 0 0 {pellet_move_up}
volume 162 163 164 165 166 167 168 169  copy move 0 0 {pellet_move_down}
volume 170 171 172 173 174 175 176 177 copy move 0 0 {pellet_move_up}
volume 178 179 180 181 182 183 184 185  copy move 0 0 {pellet_move_down}
volume 186 187 188 189 190 191 192 193 copy move 0 0 {pellet_move_up}
volume 194 195 196 197 198 199 200 201  copy move 0 0 {pellet_move_down}
volume 202 203 204 205 206 207 208 209 copy move 0 0 {pellet_move_up}
volume 210 211 212 213 214 215 216 217  copy move 0 0 {pellet_move_down}
volume 218 219 220 221 222 223 224 225 copy move 0 0 {pellet_move_up}
volume 226 227 228 229 230 231 232 233  copy move 0 0 {pellet_move_down}
volume 234 235 236 237 238 239 240 241 copy move 0 0 {pellet_move_up}
volume 242 243 244 245 246 247 248 249 copy move 0 0 {pellet_move_down}
volume 250 251 252 253 254 255 256 257 copy move 0 0 {pellet_move_up}
volume 258 259 260 261 262 263 264 265  copy move 0 0 {pellet_move_down}

volume 1 4 5 6 11 12 13 14 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58
59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84
85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107
108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145
146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164
165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183
184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202
203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221
222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240
241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259
260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278
279 280 281 move 0 -0.045 0

## ===================
## BLOCKS, SIDESETS AND NODESETS
## ===================

#Block: Fuel
block 1 volume  1 4 5 6 11 12 13 14 42 43 44 45 46 47 48 49 50 51 52 53 54 55
56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81
82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124
125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162
163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181
182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219
220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238
239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257
```

```
258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276
277 278 279 280 281
block 1 name "Fuel"

#Block: Sheath
block 2 volume 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38
39 40 41
block 2 name "Sheath"

#Sideset: OuterFuel
sideset 1 surface 1538 1528 1521 1531 1536 1526 1519 1534 1537 1527 1520 1533
1594 1602 1596 1588 1592 1600 1599 1586 1593 1601 1598 1587 1524 1546 1522
1545 1436 1458 1434 1457 1348 1370 1346 1369 1260 1282 1258 1281 1172 1194
1170 1193 1084 1106 1082 1105 996 1018 994 1017 908 930 906 929 820 842 818
841 732 754 730 753 644 666 642 665 556 578 554 577 468 490 466 489 380 402
378 401 292 314 290 313 73 104 67 102 336 358 334 357 424 446 422 445 512 534
510 533 600 622 598 621 688 710 686 709 776 798 774 797 864 886 862 885 952
974 950 973 1040 1062 1038 1061 1128 1150 1126 1149 1216 1238 1214 1237 1304
1326 1302 1325 1392 1414 1390 1413 1480 1502 1478 1501 1568 1590 1566 1589
1603 1583 1595 1574 1515 1495 1507 1486 1427 1407 1419 1398 1339 1319 1331
1310 1251 1231 1243 1222 1163 1143 1155 1134 1075 1055 1067 1046 987 967 979
958 899 879 891 870 811 791 803 782 723 703 715 694 635 615 627 606 547 527
539 518 459 439 451 430 371 351 363 342 119 91 110 81 327 307 319 298 415 395
407 386 503 483 495 474 591 571 583 562 679 659 671 650 767 747 759 738 855
835 847 826 943 923 935 914 1031 1011 1023 1002 1119 1099 1111 1090 1207 1187
1199 1178 1295 1275 1287 1266 1383 1363 1375 1354 1471 1451 1463 1442 1559
1539 1551 1530
sideset 1 name "OuterFuel"

#Sideset: SymX
sideset 2 surface 145 203 199 144 157 156 270 244 265 273 249 239 447 425 441
418 359 337 353 330 105 74 98 66 315 293 309 286 403 381 397 374 469 491 462
485 513 535 506 529 617 594 623 601 573 550 579 557 705 682 711 689 661 638
667 645 881 858 887 865 793 770 799 777 749 726 755 733 837 814 843 821 1585
1562 1591 1569 1497 1474 1503 1481 1409 1386 1415 1393 1321 1298 1327 1305
1233 1210 1239 1217 1145 1122 1151 1129 1057 1034 1063 1041 969 946 975 953
925 902 931 909 1013 990 1019 997 1101 1078 1107 1085 1189 1166 1195 1173
1277 1254 1283 1371 1261 1365 1342 1349 1453 1430 1459 1437 1541 1518 1547
1525
sideset 2 name "SymX"

#Sideset: SymY
sideset 3 surface 176 196 226 206 216 186 256 242 236 247 260 251 440 417 453
432 352 329 365 344 97 65 112 83 321 300 308 285 396 409 388 373 476 497 461
484 520 541 505 528 629 608 616 593 585 564 572 549 717 696 704 681 673 652
660 637 893 872 880 857 805 784 792 769 761 740 748 725 849 828 836 813 1597
1576 1584 1561 1509 1488 1496 1473 1421 1400 1408 1385 1333 1312 1320 1297
1245 1224 1232 1209 1157 1136 1144 1121 1069 1048 1056 1033 981 960 968 937
916 924 901 1025 1004 1012 989 1113 1092 1100 1077 1201 1180 1188 1165 1289
1268 1276 1253 1377 1356 1364 1341 1465 1444 1452 1429 1553 1532 1540 1517
945
sideset 3 name "SymY"

#Sideset: FuelFront1
sideset 4 surface  1536 1526 1519 1534 1537 1527 1520 1533
sideset 4 name "FuelFront1"
```

```
#Sideset: FuelBack1
sideset 5 surface 1556 1555 1542 1548 1557 1554 1543 1549
sideset 5 name "FuelBack1"

#Sideset: FuelFront2
sideset 6 surface 1448 1438 1431 1446 1449 1439 1432 1445
sideset 6 name "FuelFront2"

#Sideset: FuelBack2
sideset 7 surface 1460 1468 1467 1454 1461 1469 1466 1455
sideset 7 name "FuelBack2"

#Sideset: FuelFront3
sideset 8 surface 1360 1350 1343 1358 1361 1351 1344 1357
sideset 8 name "FuelFront3"

#sideset: FuelBack3
sideset 9 surface 1372 1380 1379 1366 1373 1381 1378 1367
sideset 9 name "FuelBack3"

#Sideset: FuelFront4
sideset 10 surface 1272 1262 1255 1270 1273 1263 1256 1269
sideset 10 name "FuelFront4"

#sideset: FuelBack4
sideset 11 surface 1284 1292 1291 1278 1285 1293 1290 1279
sideset 11 name "FuelBack4"

#Sideset: FuelFront5
sideset 12 surface 1184 1174 1167 1182 1185 1175 1168 1181
sideset 12 name "FuelFront5"

#sideset: FuelBack5
sideset 13 surface 1196 1204 1203 1190 1197 1205 1202 1191
sideset 13 name "FuelBack5"

#Sideset: InnerSheath
sideset 14 surface 219 190 223 192 246 268 258 280 248 274 262 282 232 204
230 198
sideset 14 name "InnerSheath"

#Sideset: OuterSheath
sideset 15 surface 185 202 234 213 209 228 200 178 264 271 194 183 240 243
188 179 253 257 217 208 278 281 225 215 254 261 283 276 272 267 250 238
sideset 15 name "OuterSheath"

#Sideset: SymZ
sideset 16 surface 193 72 88 222 220 80 64 189
sideset 16 name "SymZ"

#Sideset: Plenum
sideset 17 surface 219 190 223 192 246 268 258 280 248 274 262 282 232 204
230 198 1538 1528 1521 1531 1536 1526 1519 1534 1537 1527 1520 1533  1594
1602 1596 1588 1592 1600 1599 1586 1593 1601 1598 1587 1524 1546 1522 1545
1436 1458 1434 1457 1348 1370 1346 1369 1260 1282 1258 1281 1172 1194 1170
1193 1084 1106 1082 1105 996 1018 994 1017 908 930 906 929 820 842 818 841
732 754 730 753 644 666 642 665 556 578 554 577 468 490 466 489 380 402 378
```

```
401 292 314 290 313 73 104 67 102 336 358 334 357 424 446 422 445 512 534 510
533 600 622 598 621 688 710 686 709 776 798 774 797 864 886 862 885 952 974
950 973 1040 1062 1038 1061 1128 1150 1126 1149 1216 1238 1214 1237 1304 1326
1302 1325 1392 1414 1390 1413 1480 1502 1478 1501 1568 1590 1566 1589 1603
1583 1595 1574 1515 1495 1507 1486 1427 1407 1419 1398 1339 1319 1331 1310
1251 1231 1243 1222 1163 1143 1155 1134 1075 1055 1067 1046 987 967 979 958
899 879 891 870 811 791 803 782 723 703 715 694 635 615 627 606 547 527 539
518 459 439 451 430 371 351 363 342 119 91 110 81 327 307 319 298 415 395 407
386 503 483 495 474 591 571 583 562 679 659 671 650 767 747 759 738 855 835
847 826 943 923 935 914 1031 1011 1023 1002 1119 1099 1111 1090 1207 1187
1199 1178 1295 1275 1287 1266 1383 1363 1375 1354 1471 1451 1463 1442 1559
1539 1551 1530 1450 1440 1433 1443 1448 1438 1431 1446 1449 1439 1432 1445
1362 1352 1345 1355 1360 1350 1343 1358 1361 1351 1344 1357 1274 1264 1257
1267 1272 1262 1255 1270 1273 1263 1256 1269 1186 1176 1169 1179 1184 1174
1167 1182 1185 1175 1168 1181 1098 1088 1081 1091 1096 1086 1079 1094 1097
1087 1080 1093 1010 1000 993 1003 1008 998 991 1006 1009 999 992 1005 922 912
905 915 920 910 903 918 921 911 904 917 834 824 817 827 832 822 815 830 833
823 816 829 746 736 729 739 744 734 727 742 745 735 728 741 658 648 641 651
656 646 639 654 657 647 640 653 570 560 553 563 568 558 551 566 569 559 552
565 482 472 465 475 480 470 463 478 481 471 464 477 394 384 377 387 392 382
375 390 393 383 376 389 306 296 289 299 304 294 287 302 305 295 288 301 58 54
35 41 56 51 31 45 57 52 32 44 350 340 333 343 348 338 331 346 349 339 332 345
438 428 421 431 436 426 419 434 437 427 420 433 526 516 509 519 524 514 507
522 525 515 508 521 614 604 597 607 612 602 595 610 613 603 596 609 702 692
685 695 700 690 683 698 701 691 684 697 790 780 773 783 788 778 771 786 789
779 772 785 878 868 861 871 876 866 859 874 877 867 860 873 966 956 949 959
964 954 947 962 965 955 948 961 1054 1044 1037 1047 1052 1042 1035 1050 1053
1043 1036 1049  1142 1132 1125 1135 1140 1130 1123 1138 1141 1131 1124 1137
1230 1220 1213 1223 1228 1218 1211 1226 1229 1219 1212 1225 1318 1308 1301
1311 1316 1306 1299 1314 1317 1307 1300 1313 1406 1396 1389 1399 1404 1394
1387 1402 1405 1395 1388 1401 1494 1484 1477 1487 1492 1482 1475 1490 1493
1483 1476 1489 1582 1572 1565 1575 1580 1570 1563 1578 1581 1571 1564 1577
1506 1514 1508 1500 1504 1512 1511 1498 1505 1513 1510 1499  1418 1426 1420
1412 1416 1424 1423 1410 1417 1425 1422 1411 1330 1338 1332 1324 1328 1336
1335 1322 1329 1337 1334 1323 1242 1250 1244 1236 1240 1248 1247 1234 1241
1249 1246 1235 1154 1162 1156 1148 1152 1160 1159 1146 1153 1161 1158 1147
1066 1074 1068 1060 1064 1072 1071 1058 1065 1073 1070 1059 978 986 980 972
976 984 983 970 977 985 982 971 890 898 892 884 888 896 895 882 889 897 894
883 802 810 804 796 800 808 807 794 801 809 806 795 714 722 716 708 712 720
719 706 713 721 718 707 626 634 628 620 624 632 631 618 625 633 630 619 538
546 540 532 536 544 543 530 537 545 542 531 450 458 452 444 448 456 455 442
449 457 454 443 362 370 364 356 360 368 367 354 361 369 366 355 108 118 111
101 106 116 114 99 107 117 113 100 318 326 320 312 316 324 323 310 317 325
322 311 406 414 408 400 404 412 411 398 405 413 410 399 494 502 496 488 492
500 499 486 493 501 498 487 582 590 584 576 580 588 587 574 581 589 586 575
670 678 672 664 668 676 675 662 663 677 674 669 758 766 760 752 756 764 763
750 757 765 762 751 846 854 848 840 844 852 851 838 845 853 850 839 934 942
936 928 932 940 939 926 933 941 938 927 1022 1030 1024 1016 1020 1028 1027
1014 1021 1029 1026 1015 1110 1118 1112 1104 1108 1116 1115 1102 1109 1117
1114 1103 1198 1206 1200 1192 1196 1204 1203 1190 1197 1205 1202 1191 1286
1294 1288 1280 1284 1292 1291 1278 1285 1293 1290 1279 1374 1382 1376 1368
1372 1380 1379 1366 1373 1381 1378 1367 1462 1470 1464 1456 1460 1468 1467
1454 1461 1469 1466 1455 1558 1552 1544 1550 1556 1555 1542 1548 1557 1554
1543 1549
sideset 17 name "Plenum"

#Sideset: FuelFront6
```

```
sideset 18 surface 1096 1086 1079 1094 1097 1087 1080 1093
sideset 18 name "FuelFront6"

#Sideset: FuelBack6
sideset 19 surface 1108 1116 1115 1102 1109 1117 1114 1103
sideset 19 name "FuelBack6"

#Sideset: FuelFront7
sideset 20 surface 1008 998 991 1006 1009 999 992 1005
sideset 20 name "FuelFront7"

#Sideset: FuelBack7
sideset 21 surface  1020 1028 1027 1014 1021 1029 1026 1015
sideset 21 name "FuelBack7"

#Sideset: FuelFront8
sideset 22 surface 920 910 903 918 921 911 904 917
sideset 22 name "FuelFront8"

#Sideset: FuelBack8
sideset 23 surface 932 940 939 926 933 941 938 927
sideset 23 name "FuelBack8"

#Sideset: FuelFront9
sideset 24 surface 832 822 815 830 833 823 816 829
sideset 24 name "FuelFront9"

#Sideset: FuelBack9
sideset 25 surface 844 852 851 838 845 853 850 839
sideset 25 name "FuelBack9"

#Sideset: FuelFront10
sideset 26 surface 744 734 727 742 745 735 728 741
sideset 26 name "FuelFront10"

#Sideset: FuelBack10
sideset 27 surface 756 764 763 750 757 765 762 751
sideset 27 name "FuelBack10"

#Sideset: FuelFront11
sideset 28 surface 656 646 639 654 657 647 640 653
sideset 28 name "FuelFront11"

#Sideset: FuelBack11
sideset 29 surface 668 676 675 662 663 677 674 669
sideset 29 name "FuelBack11"

#Sideset: FuelFront12
sideset 30 surface 568 558 551 566 569 559 552 565
sideset 30 name "FuelFront12"

#Sideset: FuelBack12
sideset 31 surface 580 588 587 574 581 589 586 575
sideset 31 name "FuelBack12"

#Sideset: FuelFront13
sideset 32 surface 480 470 463 478 481 471 464 477
```

```
sideset 32 name "FuelFront13"

#Sideset: FuelBack13
sideset 33 surface 492 500 499 486 493 501 498 487
sideset 33 name "FuelBack13"

#Sideset: FuelFront14
sideset 34 surface 392 382 375 390 393 383 376 389
sideset 34 name "FuelFront14"

#Sideset: FuelBack14
sideset 35 surface  404 412 411 398 405 413 410 399
sideset 35 name "FuelBack14"

#Sideset: FuelFront15
sideset 36 surface 304 294 287 302 305 295 288 301
sideset 36 name "FuelFront15"

#Sideset: FuelBack15
sideset 37 surface 316 324 323 310 317 325 322 311
sideset 37 name "FuelBack15"

#Sideset: FuelFront16
sideset 38 surface 56 51 31 45 57 52 32 44
sideset 38 name "FuelFront16"

#Sideset: FuelFront17
sideset 39 surface 348 338 331 346 349 339 332 345
sideset 39 name "FuelFront17"

#Sideset: FuelFront18
sideset 40 surface 436 426 419 434 437 427 420 433
sideset 40 name "FuelFront18"

#Sideset: FuelFront19
sideset 41 surface  524 514 507 522 525 515 508 521
sideset 41 name "FuelFront19"

#Sideset: FuelFront20
sideset 42 surface 612 602 595 610 613 603 596 609
sideset 42 name "FuelFront20"

#Sideset: FuelFront21
sideset 43 surface 700 690 683 698 701 691 684 697
sideset 43 name "FuelFront21"

#Sideset: FuelFront22
sideset 44 surface 788 778 771 786 789 779 772 785
sideset 44 name "FuelFront22"

#Sideset: FuelFront23
sideset 45 surface 876 866 859 874 877 867 860 873
sideset 45 name "FuelFront23"

#Sideset: FuelFront24
sideset 46 surface 964 954 947 962 965 955 948 961
sideset 46 name "FuelFront24"
```

```
#Sideset: FuelFront25
sideset 47 surface 1052 1042 1035 1050 1053 1043 1036 1049
sideset 47 name "FuelFront25"

#Sideset: FuelFront26
sideset 48 surface 1140 1130 1123 1138 1141 1131 1124 1137
sideset 48 name "FuelFront26"

#Sideset: FuelFront27
sideset 49 surface 1228 1218 1211 1226 1229 1219 1212 1225
sideset 49 name "FuelFront27"

#Sideset: FuelFront28
sideset 50 surface 1316 1306 1299 1314 1317 1307 1300 1313
sideset 50 name "FuelFront28"

#Sideset: FuelFront29
sideset 51 surface 1404 1394 1387 1402 1405 1395 1388 1401
sideset 51 name "FuelFront29"

#Sideset: FuelFront30
sideset 52 surface 1492 1482 1475 1490 1493 1483 1476 1489
sideset 52 name "FuelFront30"

#Sideset: FuelFront31
sideset 52 surface 1580 1570 1563 1578 1581 1571 1564 1577
sideset 52 name "FuelFront31"

#Sideset: FuelBack16
sideset 53 surface 106 116 114 99 107 117 113 100
sideset 53 name "FuelBack16"

#Sideset: FuelBack17
sideset 54 surface 360 368 367 354 361 369 366 355
sideset 54 name "FuelBack17"

#Sideset: FuelBack18
sideset 55 surface 448 456 455 442 449 457 454 443
sideset 55 name "FuelBack18"

#Sideset: FuelBack19
sideset 56 surface 536 544 543 530 537 545 542 531
sideset 56 name "FuelBack19"

#Sideset: FuelBack20
sideset 57 surface 624 632 631 618 625 633 630 619
sideset 57 name "FuelBack20"

#Sideset: FuelBack21
sideset 58 surface 712 720 719 706 713 721 718 707
sideset 58 name "FuelBack21"

#Sideset: FuelBack22
sideset 59 surface 800 808 807 794 801 809 806 795
sideset 59 name "FuelBack22"
```

```
#Sideset: FuelBack23
sideset 60 surface 888 896 895 882 889 897 894 883
sideset 60 name "FuelBack23"

#Sideset: FuelBack24
sideset 61 surface 976 984 983 970 977 985 982 971
sideset 61 name "FuelBack24"

#Sideset: FuelBack25
sideset 62 surface 1064 1072 1071 1058 1065 1073 1070 1059
sideset 62 name "FuelBack25"

#Sideset: FuelBack26
sideset 63 surface 1152 1160 1159 1146 1153 1161 1158 1147
sideset 63 name "FuelBack26"

#Sideset: FuelBack27
sideset 64 surface 1240 1248 1247 1234 1241 1249 1246 1235
sideset 64 name "FuelBack27"

#Sideset: FuelBack28
sideset 65 surface 1328 1336 1335 1322 1329 1337 1334 1323
sideset 65 name "FuelBack28"

#Sideset: FuelBack29
sideset 66 surface 1416 1424 1423 1410 1417 1425 1422 1411
sideset 66 name "FuelBack29"

#Sideset: FuelBack30
sideset 67 surface 1504 1512 1511 1498 1505 1513 1510 1499
sideset 67 name "FuelBack30"

#Sideset: FuelBack31
sideset 68 surface 1592 1600 1599 1586 1593 1601 1598 1587
sideset 68 name "FuelBack31"

#Sideset: SheathMidplane
sideset 69 face 100165 99762 99359 98956 98553 98150 97747 97344 96941 96538
96135 95732 95329 94926 94523 94120 100164 99761 99358 98955 98552 98149
97746 97343 96940 96537 96134 95731 95328 94925 94522 94119 72241 72242 72645
72644 73047 73048 73451 73450 73853 73854 74257 74256 74659 74660 75063 75062
75465 75466 75869 75868 76271 76272 76675 76674 77077 77078 77481 77480 77883
77884 78287 78286
sideset 69 name "SheathInterior"

#Sideset: FuelMidplane
sideset 70 face 661 677 678 662 663 679 680 664 665 681 682 666 667 683 684
668 669 685 686 670 671 687 688 672 673 689 690 674 675 691 692 676 980 996
997 981 982 998 999 983 984 1000 1001 985 986 1002 1003 987 988 1004 1005 989
990 1006 1007 991 992 1008 1009 993 994 1010 1011 995
sideset 70 name "FuelExterior"

#Nodeset: SheathXY
nodeset 1001 vertex 220
nodeset 1001 name "SheathXY"
```

```
#The two nodesets below are not required for the quarter model and are
commented out accordingly.

#Nodeset: SheathX
#nodeset 2001 node 141
#nodeset 2001 name "SheathX"

#Nodeset: SheathXYZ
#nodeset 3001 node 176
#nodeset 3001 name "SheathXYZ"

#Uncomment below to produce a half model
#delete volume 5 6 13 14 21 22 23 27 28 29 33 34 35 39 40 41 44 45 48 49 52
53 56 57 60 61 64 65 68 69 72 73 76 77 80 81 84 85 88 89 92 93 96 97 100 101
104 105 108 109 112 113 116 117 120 121 124 125 128 129 132 133 136 137 140
141 144 145 148 149 152 153 156 157 160 161 164 165 168 169 172 173 176 177
180 181 184 185 188 189 192 193 196 197 200 201 204 205 208 209 212 213 216
217 220 221 224 225 228 229 232 233 236 237 240 241 244 245 248 249 252 253
256 257 260 261 264 265 268 269 272 273 276 277 280 281

#Uncomment below to produce a quarter model
delete volume 5 6 13 14 21 22 23 27 28 29 33 34 35 39 40 41 44 45 48 49 52 53
56 57 60 61 64 65 68 69 72 73 76 77 80 81 84 85 88 89 92 93 96 97 100 101 104
105 108 109 112 113 116 117 120 121 124 125 128 129 132 133 136 137 140 141
144 145 148 149 152 153 156 157 160 161 164 165 168 169 172 173 176 177 180
181 184 185 188 189 192 193 196 197 200 201 204 205 208 209 212 213 216 217
220 221 224 225 228 229 232 233 236 237 240 241 244 245 248 249 252 253 256
257 260 261 264 265 268 269 272 273 276 277 280 281 11 12 18 19 20 24 25 26
50 51 54 55 66 67 70 71 82 83 86 87 98 99 102 103 114 115 118 119 130 131 134
135 146 147 150 151 162 163 166 167 178 179 182 183 194 195 198 199 210 211
214 215 226 227 230 231 242 243 246 247 258 259 262 263 274 275 278 279

#Rescale to millimetres
volume all scale 0.001
block all element type HEX20

#Export Mesh
export mesh "~/projects/trunk/horse/ThirtyOnePelletSheath_quarter.e"
overwrite
```

Appendix E – Example HORSE Input File

An example HORSE input file is including in this appendix. To illustrate the structure of a typical input file used for the HORSE simulations the input file for the comparison against the nominal ELESTRES simulation is provided. The details of the different blocks were provided in Section 6.5. The comma separated value files used for the fission gas addition (nominal_fission_gas2.csv) and the linear power history (nominal2.csv) are provided after the input file.

Input File

```
[GlobalParams]
  #Set global parameters
  disp_x = disp_x
  disp_y = disp_y
  disp_z = disp_z
  order = SECOND
  family = LAGRANGE
[]

[Problem]
  type = ReferenceResidualProblem
  solution_variables = 'disp_x disp_y disp_z temp'
  reference_residual_variables = 'saved_x saved_y saved_z saved_t'
  acceptable_iterations = 10
  acceptable_multiplier = 10
[]

[Mesh]
  file = ThirtyOnePelletSheath_quarter.e
  displacements = 'disp_x disp_y disp_z'
  patch_size = 1000
[]

[Variables]
  [./disp_x]
  [../]

  [./disp_y]
  [../]

  [./disp_z]
  [../]

  [./temp]
    initial_condition = 300.0
  [../]
[]

[AuxVariables]
  [./vonmises]
    order = CONSTANT
    family = MONOMIAL
  [../]
  [./hoop_stress]
    order = CONSTANT
    family = MONOMIAL
  [../]
  [./radial_stress]
    order = CONSTANT
    family = MONOMIAL
  [../]
  [./axial_stress]
    order = CONSTANT
    family = MONOMIAL
```

```
[../]
[./creep_strain_hoop]
  order = CONSTANT
  family = MONOMIAL
[../]
[./creep_strain_axial]
  order = CONSTANT
  family = MONOMIAL
[../]
[./burnup]
  order = CONSTANT
  family = MONOMIAL
[../]
[./saved_x]
[../]
[./saved_y]
[../]
[./saved_z]
[../]
[./saved_t]
[../]
[./gap_cond]
  order = CONSTANT
  family = MONOMIAL
[../]
[./hoop_strain]
  order = CONSTANT
  family = MONOMIAL
[../]
[./axial_strain]
  order = CONSTANT
  family = MONOMIAL
[../]
[./heat_production]
  order = CONSTANT
  family = MONOMIAL
[../]
[./density_aux]
  order = CONSTANT
  family = MONOMIAL
[../]
[./plastic_strain_hoop]
  order = CONSTANT
  family = MONOMIAL
[../]
[./plastic_strain_axial]
  order = CONSTANT
  family = MONOMIAL
[../]
[./total_strain_hoop]
  order = CONSTANT
  family = MONOMIAL
[../]
[./total_strain_axial]
  order = CONSTANT
  family = MONOMIAL
[../]
```

```
  [./avg_burnup]
    order = CONSTANT
    family = MONOMIAL
  [../]
[]

[Functions]
  [./temperature]
    type = ConstantFunction
    value = 583
  [../]

  [./power_history]
    type = PiecewiseConstant
    format = columns
    data_file = nominal2.csv
    direction = right
  [../]
  [./fission_gas_added]
    type = PiecewiseLinearFile
    format = columns
    data_file = nominal_fission_gas2.csv
  [../]

  [./pressure_ramp]
    type = PiecewiseLinear
    x = '0 1'
    y = '0 1'
  [../]
[]

[SolidMechanics]
  [./solid]
    disp_x = disp_x
    disp_y = disp_y
    disp_z = disp_z
    temp = temp
     save_in_disp_x = saved_x
    save_in_disp_y = saved_y
    save_in_disp_z = saved_z
  [../]
[]

[Kernels]
  [./gravity]
    type = Gravity
    variable = disp_y
    value = -9.81
    save_in = saved_y
  [../]

  [./heat]
    type = HeatConduction
    variable = temp
    save_in = saved_t
  [../]
```

```
  [./heat_source]
    type = FissionHeat
    variable = temp
    block = Fuel
    save_in = saved_t
  [../]

  [./heat_ie]
    type = HeatConductionTimeDerivative
    variable = temp
    save_in = saved_t
  [../]
[]

[AuxKernels]
  [./vonmises]
    type = MaterialTensorAux
    tensor = stress
    variable = vonmises
    quantity = vonmises
    execute_on = timestep
    point2 = '0 0 1'
  [../]
  [./hoop_stress]
    type = MaterialTensorAux
    tensor = stress
    variable = hoop_stress
    quantity = hoop
    execute_on = timestep
    point2 = '0 0 1'
  [../]
  [./radial_stress]
    type = MaterialTensorAux
    tensor = stress
    variable = radial_stress
    quantity = radial
    execute_on = timestep
    point2 = '0 0 1'
  [../]
  [./axial_stress]
    type = MaterialTensorAux
    tensor = stress
    variable = axial_stress
    quantity = axial
    execute_on = timestep
    point2 = '0 0 1'
  [../]
 [./creep_strain_hoop]
    type =  MaterialTensorAux
    block = Sheath
    tensor = creep_strain
    variable = creep_strain_hoop
    quantity = Hoop
    execute_on = timestep
    point2 = '0 0 1'
  [../]
  [./creep_strain_axial]
```

```
  type = MaterialTensorAux
  block = Sheath
  tensor = creep_strain
  variable = creep_strain_axial
  quantity = axial
  execute_on = timestep
  point2 = '0 0 1'
[../]
[./burnup]
  type = Burnup
  block = Fuel
  linear_power = power_history
  number_pellets = 31
  variable = burnup
  execute_on = timestep
[../]
[./hoop_strain]
  type = MaterialTensorAux
  tensor = elastic_strain
  variable = hoop_strain
  quantity = hoop
  execute_on = timestep
  point2 = '0 0 1'
[../]
[./axial_strain]
  type = MaterialTensorAux
  tensor = elastic_strain
  variable = axial_strain
  quantity = axial
  execute_on = timestep
  point2 = '0 0 1'
[../]
[./heat_production]
  type = HeatGeneration
  block = Fuel
  variable = heat_production
  execute_on = timestep
[../]
[./density_aux]
  type = DensityAux
  block = Fuel
  variable = density_aux
  execute_on = timestep
[../]
[./plastic_strain_hoop]
  type = MaterialTensorAux
  block = Sheath
  tensor = plastic_strain
  variable = plastic_strain_hoop
  quantity = hoop
  execute_on = timestep
  point2 = '0 0 1'
[../]

[./plastic_strain_axial]
  type = MaterialTensorAux
  block = Sheath
```

```
    tensor = plastic_strain
    variable = plastic_strain_axial
    quantity = axial
    execute_on = timestep
    point2 = '0 0 1'
  [../]
  [./total_strain_hoop]
    type = MaterialTensorAux
    tensor = total_strain
    variable = total_strain_hoop
    quantity = hoop
    execute_on = timestep
    point2 = '0 0 1'
  [../]
  [./total_strain_axial]
    type = MaterialTensorAux
    tensor = total_strain
    variable = total_strain_axial
    quantity = axial
    execute_on = timestep
    point2 = '0 0 1'
  [../]
  [./avg_burnup]
    type = AverageBurnupAux
    block = Fuel
    number_pellets = 31
    linear_power = power_history
    variable = avg_burnup
    execute_on = timestep
  [../]
[]

[AuxBCs]
  [./conductance]
    type = MaterialRealAux
    property = gap_conductance
    variable = gap_cond
    boundary = OuterFuel
  [../]
[]

[Contact]
  [./pellet_sheath_mech]
    master = InnerSheath
    slave = OuterFuel
    penalty = 1e7
    model = glued
    formulation = penalty
    tangential_tolerance = 1e-4
    normal_smoothing_method = nodal_normal_based
    disp_x = disp_x
    disp_y = disp_y
    disp_z = disp_z
        [../]
  [./pellet_pellet_mech1]
    master = FuelBack1
    slave = FuelFront2
```

```
  penalty = 1e7
  tangential_tolerance = 1e-4
  normal_smoothing_method = nodal_normal_based
  disp_x = disp_x
  disp_y = disp_y
  disp_z = disp_z
[../]
[./pellet_pellet_mech2]
  master = FuelBack2
  slave = FuelFront3
  penalty = 1e7
  tangential_tolerance = 1e-4
  normal_smoothing_method = nodal_normal_based
  disp_x = disp_x
  disp_y = disp_y
  disp_z = disp_z
[../]
[./pellet_pellet_mech3]
  master = FuelBack3
  slave = FuelFront4
  penalty = 1e7
  tangential_tolerance = 1e-4
  normal_smoothing_method = nodal_normal_based
  disp_x = disp_x
  disp_y = disp_y
  disp_z = disp_z
[../]
[./pellet_pellet_mech4]
  master = FuelBack4
  slave = FuelFront5
  penalty = 1e7
  tangential_tolerance = 1e-4
  normal_smoothing_method = nodal_normal_based
  disp_x = disp_x
  disp_y = disp_y
  disp_z = disp_z
[../]
[./pellet_pellet_mech5]
  master = FuelBack5
  slave = FuelFront6
  penalty = 1e7
  tangential_tolerance = 1e-4
  normal_smoothing_method = nodal_normal_based
  disp_x = disp_x
  disp_y = disp_y
  disp_z = disp_z
[../]
[./pellet_pellet_mech6]
  master = FuelBack6
  slave = FuelFront7
  penalty = 1e7
  tangential_tolerance = 1e-4
  normal_smoothing_method = nodal_normal_based
  disp_x = disp_x
  disp_y = disp_y
  disp_z = disp_z
[../]
```

```
[./pellet_pellet_mech7]
  master = FuelBack7
  slave = FuelFront8
  penalty = 1e7
  tangential_tolerance = 1e-4
  normal_smoothing_method = nodal_normal_based
  disp_x = disp_x
  disp_y = disp_y
  disp_z = disp_z
[../]
[./pellet_pellet_mech8]
  master = FuelBack8
  slave = FuelFront9
  penalty = 1e7
  tangential_tolerance = 1e-4
  normal_smoothing_method = nodal_normal_based
  disp_x = disp_x
  disp_y = disp_y
  disp_z = disp_z
[../]
[./pellet_pellet_mech9]
  master = FuelBack9
  slave = FuelFront10
  penalty = 1e7
  tangential_tolerance = 1e-4
  normal_smoothing_method = nodal_normal_based
  disp_x = disp_x
  disp_y = disp_y
  disp_z = disp_z
[../]
[./pellet_pellet_mech10]
  master = FuelBack10
  slave = FuelFront11
  penalty = 1e7
  tangential_tolerance = 1e-4
  normal_smoothing_method = nodal_normal_based
  disp_x = disp_x
  disp_y = disp_y
  disp_z = disp_z
[../]
[./pellet_pellet_mech11]
  master = FuelBack11
  slave = FuelFront12
  penalty = 1e7
  tangential_tolerance = 1e-4
  normal_smoothing_method = nodal_normal_based
  disp_x = disp_x
  disp_y = disp_y
  disp_z = disp_z
[../]
[./pellet_pellet_mech12]
  master = FuelBack12
  slave = FuelFront13
  penalty = 1e7
  tangential_tolerance = 1e-4
  normal_smoothing_method = nodal_normal_based
  disp_x = disp_x
```

```
    disp_y = disp_y
    disp_z = disp_z
  [../]
  [./pellet_pellet_mech13]
    master = FuelBack13
    slave = FuelFront14
    penalty = 1e7
    tangential_tolerance = 1e-4
    normal_smoothing_method = nodal_normal_based
    disp_x = disp_x
    disp_y = disp_y
    disp_z = disp_z
  [../]
  [./pellet_pellet_mech14]
    master = FuelBack14
    slave = FuelFront15
    penalty = 1e7
    tangential_tolerance = 1e-4
    normal_smoothing_method = nodal_normal_based
    disp_x = disp_x
    disp_y = disp_y
    disp_z = disp_z
  [../]
  [./pellet_pellet_mech15]
    master = FuelBack15
    slave = FuelFront16
    penalty = 1e7
    tangential_tolerance = 1e-4
    normal_smoothing_method = nodal_normal_based
    disp_x = disp_x
    disp_y = disp_y
    disp_z = disp_z
  [../]
[]

[NodalNormals]
  boundary = 'InnerSheath FuelBack1 FuelBack2 FuelBack3 FuelBack4 FuelBack5
FuelBack6 FuelBack7 FuelBack8 FuelBack9 FuelBack10 FuelBack11 FuelBack12
FuelBack13 FuelBack14 FuelBack15'
[]


[ThermalContact]
  [./pellet_sheath_therm]
    type = GapHeatTransferCANDU
    variable = temp
    master = InnerSheath
    slave = OuterFuel
    avg_fuel_temp = avg_fuel_temp
    avg_sheath_temp = avg_sheath_temp
    volume = gas_volume
    fission_gas_added = fission_gas_added
    plenum_pressure = plenum_pressure
    contact_pressure = contact_pressure
    quadrature = true
  [../]
  [./pellet_pellet_therm1]
```

```
  type = GapHeatTransfer
  variable = temp
  gap_conductivity = 0.15
  master = FuelBack1
  slave = FuelFront2
  quadrature = true
[../]
[./pellet_pellet_therm2]
  type = GapHeatTransfer
  variable = temp
  gap_conductivity = 0.15
  master = FuelBack2
  slave = FuelFront3
  quadrature = true
[../]
[./pellet_pellet_therm3]
  type = GapHeatTransfer
  variable = temp
  gap_conductivity = 0.15
  master = FuelBack3
  slave = FuelFront4
  quadrature = true
[../]
[./pellet_pellet_therm4]
  type = GapHeatTransfer
  variable = temp
  gap_conductivity = 0.15
  master = FuelBack4
  slave = FuelFront5
  quadrature = true
[../]
[./pellet_pellet_therm5]
  type = GapHeatTransfer
  variable = temp
  gap_conductivity = 0.15
  master = FuelBack5
  slave = FuelFront6
  quadrature = true
[../]
[./pellet_pellet_therm6]
  type = GapHeatTransfer
  variable = temp
  gap_conductivity = 0.15
  master = FuelBack6
  slave = FuelFront7
  quadrature = true
[../]
[./pellet_pellet_therm7]
  type = GapHeatTransfer
  variable = temp
  gap_conductivity = 0.15
  master = FuelBack7
  slave = FuelFront8
  quadrature = true
[../]
[./pellet_pellet_therm8]
  type = GapHeatTransfer
```

```
  variable = temp
  gap_conductivity = 0.15
  master = FuelBack8
  slave = FuelFront9
  quadrature = true
[../]
[./pellet_pellet_therm9]
  type = GapHeatTransfer
  variable = temp
  gap_conductivity = 0.15
  master = FuelBack9
  slave = FuelFront10
  quadrature = true
[../]
[./pellet_pellet_therm10]
  type = GapHeatTransfer
  variable = temp
  gap_conductivity = 0.15
  master = FuelBack10
  slave = FuelFront11
  quadrature = true
[../]
[./pellet_pellet_therm11]
  type = GapHeatTransfer
  variable = temp
  gap_conductivity = 0.15
  master = FuelBack11
  slave = FuelFront12
  quadrature = true
[../]
[./pellet_pellet_therm12]
  type = GapHeatTransfer
  variable = temp
  gap_conductivity = 0.15
  master = FuelBack12
  slave = FuelFront13
  quadrature = true
[../]
[./pellet_pellet_therm13]
  type = GapHeatTransfer
  variable = temp
  gap_conductivity = 0.15
  master = FuelBack13
  slave = FuelFront14
  quadrature = true
[../]
[./pellet_pellet_therm14]
  type = GapHeatTransfer
  variable = temp
  gap_conductivity = 0.15
  master = FuelBack14
  slave = FuelFront15
  quadrature = true
[../]
[./pellet_pellet_therm15]
  type = GapHeatTransfer
  variable = temp
```

```
    gap_conductivity = 0.15
    master = FuelBack15
    slave = FuelFront16
    quadrature = true
  [../]
[]

[BCs]
  [./OuterSheath]
    type = ConvectiveFluxFunction
    T_infinity = temperature
    boundary = 'OuterSheath'
    coefficient = 50000
    variable = temp
  [../]

  [./no_x]
    type = DirichletBC
    variable = disp_x
    boundary = 'SymX'
    value = 0.0
  [../]

  [./no_y]
    type = DirichletBC
    variable = disp_y
    boundary = 'SheathXY'
    value = 0.0
  [../]

  [./no_z]
    type = DirichletBC
    variable = disp_z
    boundary = 'SymZ'
    value = 0.0
  [../]

  [./no_flux]
    type = NeumannBC
    variable = temp
    boundary = 'SymX SymZ'
    value = 0.0
  [../]
  [./Pressure]
    [./pressure]
      boundary = 'OuterSheath'
      factor = 10e6
      function = pressure_ramp
   [../]
  [../]

  [./PlenumPressure]
    [./plenumPressure]
      boundary = Plenum
      initial_pressure = 1.01325e5
      startup_time = -200
      R = 8.3143
```

```
      temperature = avg_temp_interior
      volume = gas_volume
      material_input = fission_gas_released
      output = plenum_pressure
      save_in_disp_x = saved_x
      save_in_disp_y = saved_y
      save_in_disp_z = saved_z
    [../]
  [../]
[]

[Materials]
  [./fuel_thermal]
    type = KylesThermalUO2
    block = Fuel
    temp = temp
    burnup = burnup
  [../]

 [./fuel_swelling]
    type = KylesVSwellingUO2
    block = Fuel
    temp = temp
    burnup = burnup
  [../]

  [./fuel_mechanical]
    type = KylesMechUO2
    block = Fuel
    disp_x = disp_x
    disp_y = disp_y
    disp_z = disp_z
    poissons_ratio = 0.316
    youngs_modulus = 1.8e11
    model_youngs_modulus = true
    temp = temp
  [../]

  [./fission_heat]
    type = QFission
    block = Fuel
    linear_power = power_history
    number_pellets = 31
  [../]
  [./sheath_thermal]
    type = KylesThermalZry
    block = Sheath
    temp = temp
  [../]

   [./creep_plas]
    type = KylesMechZry
    block = Sheath
    disp_x = disp_x
    disp_y = disp_y
    poissons_ratio = 0.3
    youngs_modulus = 7e10
```

```
    constitutive_model = combined
    model_youngs_modulus = true
    model_diffusional_creep = false
    temp = temp
    formulation = Nonlinear3D
  [../]

  [./combined]
    type = CombinedCreepPlasticity
    block = Sheath
    submodels = 'creep'
    absolute_tolerance = 1e-5
    max_its = 30
  [../]
  [./creep]
    type = KylesPowerLawCreepModel
    block = Sheath
    relative_tolerance = 1e-5
    absolute_tolerance = 1e-20
    max_its = 30
    temp = temp
  [../]
  [./plas]
    type = KylesIsotropicPlasticity
    block = Sheath
    temp = temp
    max_its = 30
    relative_tolerance = 1e-5
    absolute_tolerance = 1e-5
  [../]

  [./density_sheath]
    type = Density
    block = Sheath
    density = 6551.0
    disp_x = disp_x
    disp_y = disp_y
    disp_z = disp_z
  [../]
[]

[Dampers]
  [./limitT]
    type = MaxIncrement
    variable = temp
    max_increment = 50
  [../]
[]

[Executioner]
  type = Transient

  solve_type = 'PJFNK'

  petsc_options_iname = '-ksp_gmres_restart -pc_type -pc_hypre_type -
pc_hypre_boomeramg_max_iter -pc_hypre_boomeramg_strong_threshold'
```

```
  petsc_options_value = '        201          hypre     boomeramg
4                         0.7'


  line_search = 'none'
  nl_max_its = 25
  nl_abs_tol = 1e-4
  nl_rel_tol = 1e-3

  l_tol = 1e-2
  l_max_its = 100

  start_time = 0.0
  end_time = 28151997

  dtmax = 1e6
  dtmin = 1

  [./TimeStepper]
    type = IterationAdaptiveDT
    timestep_limiting_function = power_history
    max_function_change = 3e20
    force_step_every_function_point = true
    dt = 20
    optimal_iterations = 10
    iteration_window = 4
    linear_iteration_ratio = 100
  [../]

  [./Quadrature]
    order = THIRD
  [../]
[]

[Postprocessors]
  [./avg_temp_interior]
    type = SideAverageValue
    boundary = Plenum
    variable = temp
  [../]
  [./avg_fuel_temp]
    type = SideAverageValue
    boundary = OuterFuel
    variable = temp
  [../]
  [./avg_sheath_temp]
    type = SideAverageValue
    boundary = InnerSheath
    variable = temp
  [../]
  [./gas_volume]
    type = InternalVolume
    boundary = Plenum
  [../]
  [./_dt]
    type = TimestepSize
  [../]
```

```
  [./nonlinear_its]
    type = NumNonlinearIterations
  [../]
  [./flux_from_sheath]
    type = SideFluxIntegral
    variable = temp
    boundary = SheathInterior
    diffusivity = thermal_conductivity
  [../]
  [./flux_from_fuel]
    type = SideFluxIntegral
    variable = temp
    boundary = SheathExterior
    diffusivity = thermal_conductivity
  [../]
  [./element_input_power]
    type = PlotFunction
    function = power_history
    output = both
  [../]
  [./fission_gas_released]
    type = PlotFunction
    function = fission_gas_added
  [../]
[]

[Output]
  file_base = ThirtyOnePelletSheath_refresCANDU1
  interval = 1
  linear_residuals = true
  elemental_as_nodal = true
  output_initial = true
  exodus = true
  perf_log = true
[]

[Debug]
  show_var_residual_norms = true
[]
```

Power History and Fission Gas Added (nominal2.csv, nominal_fission_gas2.csv)

Power History		Fission Gas Added	
Time (s)	**Power (kW/m)**	**Time (s)**	**Fission Gas Added(mol)**
0	51500	0	0
704245.3	51500	704245.3	0
1401719	52000	1401719	4.16E-06
2091237	52600	2091237	6.70E-06
2774262	53100	2774262	9.32E-06
3452180	53500	3452180	1.91E-05
4130098	53500	4130098	1.91E-05
4809286	53400	4809286	2.23E-05
5492311	53100	5492311	2.67E-05
6179217	52800	6179217	0.000030197
6872690	52300	6872690	0.000033482
7571507	51900	7571507	0.000035479
8278498	51300	8278498	0.000037892
8992448	50800	8992448	0.000036477
9714930	50200	9714930	0.000037933
10446153	49600	10446153	0.000037725
11186329	49000	11186329	0.000037809
11937232	48300	11937232	0.000039014
12695990	47800	12695990	0.000034439
13464393	47200	13464393	0.000039514
14242690	46600	14242690	0.000037309
15041559	45400	15041559	0.00004804
15849323	44900	15849323	0.000017844
16668028	44300	16668028	0.000036644
17496079	43800	17496079	0.000024872
18331762	43400	18331762	0.000028617
19179160	42800	19179160	0.000033815
20036574	42300	20036574	0.000024831
20904245	41800	20904245	2.95E-05
21780299	41400	21780299	0.00002271
22664900	41000	22664900	0.000028658
23560421	40500	23560421	0.000026495
24462626	40200	24462626	0.000016221
25373898	39800	25373898	0.000029033
26294422	39400	26294422	0.000018051
27219642	39200	27219642	1.58E-05
28151997	38900	28151997	0.000025871

Appendix F – ELESTRES Input File

An example ELESTRES input file is provided below for the nominal full irradiation history analysis. The only difference between the nominal and overpower cases is the irradiation history provided in Group 10. The full input file of the nominal case is provided and Group 10 of the overpower case is provided afterwards. It can be seen that the input file format is much different than a MOOSE based application's input file.

Nominal

```
' Under the DOS Window on Windows NT 4.0 the command to use to eliminate
' blank lines and comments (lines that start with ') is the following:
' end of NT-DOS filter
'
' type input | findstr /v /b /c:"'" | findstr /v /r /c:"^[     ]*$" > output
' Note that "input" is the name of this file.
' Note that "output" is the name of the result file.
' Note that between the square brackets is a space and a tab character.

' type - echoes input file
' findstr /v /b / c:"'" - find string matching pattern at beginning of line (/v)
'                         and print only those lines that don't match (/b)
'                         /c:"'" - matches a single quote
'                         i.e. print only those lines that do not begin with '
'
' findstr /v /r /c:"^[     ]*$" - find string matching pattern at beginning of the
line (/v)
'                                 using the regular expression (/r) in the string
'                                 "^[    ]*$" - beginning of lines with either/or both
'                                 tab or space zero or more times.
'                                 Note that between the square brackets is a space
'                                 and a tab character.
'
' > output - redirects result to the file output
' end of NT-DOS filter command explanation
'
' Under Unix/Linux
' cat - echoes infile.txt
' grep -v --> will list all lines not beginning with '
' grep -v "^[\t ]*$" --> will list all links that aren't blank or contain
'                       only spaces and/or tabs
' cat infile.txt | grep -v "^'" | grep -v "^[\t ]*$" > infile

' ELESTRES IST 1.20
' Revision History:
' 2010 MARCH 04 - W.R. Richmond
' ELESTRES Input File
'   - outer element - CANDU 6 reactor
'   - nominal design envelope
'
```

```
'
'--  0---|--- 10---|--- 20---|--- 30---|--- 40---|--- 50---|--- 60---|--- 70---|---
80---|--- 90---|
'23456789|123456789|123456789|123456789|123456789|123456789|123456789|123456789|123456
789|123456789|
'*********************************************************************************
'
'  Name of Version of ELESTRES to be run
'
'
ELESTRES-IST 1.2
'
' GROUP 8 - INPUT TITLES
'
'grp|# of| comment cards
'   |    |
    8    2
  CANDU 6 - nominal design power
  Burnup to 360 MWh/kgU
'
' GROUP 1 - PELLET DATA
'
'grp|# of| pellet   | dish     | lip      | central | pellet   | open     |length of|
' 1 |pell| diameter| depth    |          | hole dia| density  | porosity|fuel stck|
'   |ets |   (mm)   | (mm)     | (mm)     | (mm)    | (Mg/m3)  |(fractn)  |   (mm)   |
'   |    |          |          |          |         |          |          |          |
'   |PNOS|    DI    |  DDX     |  WLAND   | DCHOLE  |   RHOI   |    (X)   | STALEN   |
'   |    |          |          |          |         |          |          |          |
    1   31    12.210     0.200     0.591              10.650               480.5
'
' GROUP 2 - ELEMENT DATA
'
'grp|NEL | axial    |diametral| sheath   | filling | plenum   | transduc| fraction|
' 2 |    | gap      |clearance|thickness| gas vol.| void     | cavity  |    of   |
'   |    |          |         |          |or press.|          |         | helium  |
'   |    | (mm)     |  (mm)   |  (mm)    |(mm3-MPa)|(mm3/K)   | vol(mm3)|         |
'   |    |          |         |          |         |          |         |         |
'   |NEL |    HA    |   DC    |  THS     |  FGV    | PLDVD(1)|  TRVM   | FRACHE  |
'   |    |          |         |          |         |          |         |         |
    2   37     2.000     0.090     0.401 -0.101325                          1.0
'
' GROUP 3 - SHEATH PROPERTIES DATA
' SHTH - 0 - Zircaloy 4 (cold worked, stress relieved)
'       99 - type 304 stainless steel
'grp|SHTH| young's | blank    | coeff of| thermal | sheath   | pellet   |coeff. of|
' 3 |shth| modulus |          | thermal | cond. of| ovality  | ovality  |friction |
'   |matl|         |          | expans  |sheath ma|          |          |between  |
'   |    |(GN/m2)  |          |(um/m/K) |kW/(m K))| (mm)     |  (mm)    |fuel/shth|
'   |    |         |          |         |         |          |          |         |
'   |MSHTH  SYM    |  YLSTRN  |    C    |  STC    | OVALSH   | OVALPL   |  PMSU   |
'   |    |         |          |         |         |          |          |         |

    3    0
'
' GROUP 4 - HEAT TRANSFER DATA
'
'grp|ICAN| blank    | sheath   | sheath   | pellet   |
' 4 |CAN-|          | coolant  | inside   | surface  |
'   |LUB |          | htrans   | roughns  | roughns  |
'   |    |          | kW/(m2K) | (um)     |          |
'   |    |          |          |          |          |
'   |ICAN|          |FILMCO(1)|  SRFS    |  SRFP    | Columns 51 - 80 are blank
'   |    |          |          |          |          |          |          |
```

```
    4    1                            1.0       1.0
'
'GROUP 5 - OPERATIONAL PARAMETERS
'
' IFLAG - 0 - axial clearance distributed equally among pellets - req'd for ELDAT file
'       - 1 - axial clearance segregated at one end of the fuel element
'grp|FLAG| coolant | coolant | neutron |thermal  | grain   | NRICH   |diffusion|
' 5 |  0 | temp.   | pressure| inverse |to       | size    |U235 in U|factor   |
'   | or | (K)     | (MPa)   |diffusion|fission  |         |enrichmnt|used with|
'   |  1 |         |         | length  |power    |         |         |diff.coef|
'   |    |         |         | (1/mm)  |ratio    | (um)    |   (%)   |         |
'   |    |         |         |         |         |         |         |         |
'   |IFLAG  TCL(1) |  CPR(1) |  DLINV  | TFPOWR  |  ASIZ   |  ENRCH  |  DIFAC  |
'   |    |         |         |         |         |         |         |         |
'
    5                                       0.925     10.00      0.71
'
'GROUP 6 - not used
'
'GROUP 7 - GROOVE DATA
'
' this input line is used for modelling experiments

'grp|CSGN|  CGW    |  CGD    |  WIREDI | CSGN - number of circum. grooves
' 7 |    |         |         |         | CGW - groove width (mm)
'   |    |         |         |         | CGD - groove depth (mm)
'   |    |         |         |         | WIREDI - dia. of flux detector wire (mm)
'   |    |         |         |         | blank   | blank   | blank   | blank   |
'   |    |         |         |         |         |         |         |         |
'   |NCFGS  CGW    |  CGD    |  WIREDI |    Columns 41 - 80 are not used       |
'   |    |         |         |         |                                       |

'
'GROUP 9 - OUTPUT OPTIONS
' IPRINT - 0 - default - minimal output
'          1 - detailed results output
'          2 - full results output
'          3 to 5 - same as IPRINT = 0
'          6 - similar to 1 except detailed Yth results NOT printed to OUTFILE
'          7 - similar to 1 except finite element info. printed out in TAPE16
'              and in OUTFILE1 data files
'
'          YPRINT(I) - indicates Yth calculation at which conditions for each
'                      annulus are to be printed out
'                    - last YPRINT option produces ELDAT file if FLAG in group
'                      5 is set equal to 1
'
'grp|  I  |YPRINT(1)|YPRINT(2)|YPRINT(3)|YPRINT(4)|YPRINT(5)|YPRINT(6)|YPRINT(7)|
' 9 |  P  |         |         |         |         |         |         |         |
'   |  R  |         |         |         |         |         |         |         |
'   |  I  |         |         |         |         |         |         |         |
'   |  N  |         |         |         |         |         |         |         |
'   |  T  |         |         |         |         |         |         |         |
'
    9    1      72.0
'
'
' GROUP 11 - fuel element and operational data
'
'grp|Numb|PelletGap|Chamfer  |Chamfer  |Axial    |Ramp Time|Max Allow| Blank   |
'11 |er  |axial    |Width    |Height   |Yield    |or Ramp  |Power    |         |
'   |of  |clearance|(radial  |(axial   |Strength |Rate     |Change   |         |
```

```
'   |Pell|between  | directn)| directn)|of Sheath|(s or    |Step     |         |
'   |et  |pellets  |         |         |         |  kW/m/s)|Size     |         |
'   |dish| (mm)    | (mm)    | (mm)    |(MPa)    |         |(kW/m)   |         |
'   |    |         |         |         |         |         |         |         |
'   |NDISH  PELGAP |  WCHAM  |  HCHAM  | SYIELD  |   RR1   |  PINK   |  Blank  |
'   |    |         |         |         |         |         |         |         |
   11    2               0.63     0.066

'
' GROUP 15 - fuel pellet finite element mesh data and modelling options
'            NOTE:  Group 15 must comprise 2 records
'
'grp| B  |maximum  |
'15 | L  |number of|
'   | A  |result   |
'   | N  |lines on | Columns 21 - 80 are not used
'   | K  |each page|
'   |    |         |
'   |    |         |
'   |    |  LMAX   |         |         |         |         |         |         |
'   |    |         |         |         |         |         |         |         |
   15
'
' second line of GROUP 15
'   | N  | N  | B  | I  | BLANK   |     | columns 41 - 80 are blank
'   | O  | O  | L  | F  |         |     | IAN - # of finite diff. annuli in grid
'   | D  | D  | A  | A  |         |     | NODER - # of radial FE nodes along
'   | E  | E  | N  | C  | BLANK   |     |         pellet mid-plane
'IAN| R  | A  | K  | T  |         |     | NODEAX - # of axial FE nodes along
'   |    | X  |    | A  |         |     |          pellet centreline
'   |    |    |    | R  |         |     | IFACTAR - ELDAT 104th line output option
'   |    |    |    |    | BLANK   |IDY  |           1 (default) - ELOCA code output
'   |    |    |    |    |         |     |           2 FACTAR code output
'   |    |    |    |    |         |     |         |         |         |         |
  100
'
' third line of GROUP 15 - optional
'
'        |         |         |         |         |         |         |         |
'        |         |         |         |         |         |         |         |
'DY_WT   | DY_RHOS | DY_B1   | DY_B2   |  DY_B3  |  DY_EXX |  DY_PRX | BLANK   |
'        |         |         |         |         |         |         |         |
'        |         |         |         |         |         |         |         |
' No information

'grp|Numb|PelletGap|Chamfer  |Chamfer  |Axial    |Ramp Time|Max Allow| Blank   |
'11 |er  |axial    |Width    |Height   |Yield    |or Ramp  |Power    |         |
'   |of  |clearance|(radial  |(axial   |Strength |Rate     |Change   |         |
'   |Pell|between  | directn)| directn)|of Sheath|(s or    |Step     |         |
'   |et  |pellets  |         |         |         |  kW/m/s)|Size     |         |
'   |dish| (mm)    | (mm)    | (mm)    |(MPa)    |         |(kW/m)   |         |
'   |    |         |         |         |         |         |         |         |
'   |NDISH  PELGAP |  WCHAM  |  HCHAM  | SYIELD  |   RR1   |  PINK   |  Blank  |
'   |    |         |         |         |         |         |         |         |

' Group 10 input consists of 1 Group 10 card plus NDATA power history points
'
'grp| N  |Power    | NDATA - number of power history steps to be read
'10 | D  |Factor   |
'   | A  |Scaling  |
'   | T  |Factor   |
'   | A  |         |
'   |    |         |
```

```
'   |NDATA POWFAC  |
'   |    |         |

   10   36     1.000
'
' 2nd Section of Group 10
' J | K   | power    | burnup   | coolant | coolant |sheath to| plenum   | ramp     |
' F | F   |+ve value|+ve value| temp     | pressure|coolant   | voidage | rate (s)|
' L | L   | (kW/m)   |(MWh/kgU)|          |         |heat xfer|          | or ramp |
' A | A   |-ve value|-ve value|          |         |coeff.    |          | time     |
' G | G   | total    |(hrs)     |          |         |          |          | (kW/m/s)|
'   |     |  (kW)    |          |  (K)     |  (MPa)  | (kW/m2K)| (mm3/K) |          |
'   |     |          |          |          |         |          |          |          |
'   |KFLAG  POWER  | BURNUP   |   TCL    |   CPR   | FILMCO   |  PLVD    |   RR     |
'   |     |          |          |          |         |          |          |          |
'
                51.5  -195.624    583.00      10.0     50.00
                52.0  -193.743    583.00      10.0     50.00
                52.6  -191.533    583.00      10.0     50.00
                53.1  -189.729    583.00      10.0     50.00
                53.5  -188.311    583.00      10.0     50.00
                53.5  -188.311    583.00      10.0     50.00
                53.4  -188.663    583.00      10.0     50.00
                53.1  -189.729    583.00      10.0     50.00
                52.8  -190.807    583.00      10.0     50.00
                52.3  -192.631    583.00      10.0     50.00
                51.9  -194.116    583.00      10.0     50.00
                51.3  -196.386    583.00      10.0     50.00
                50.8  -198.319    583.00      10.0     50.00
                50.2  -200.690    583.00      10.0     50.00
                49.6  -203.117    583.00      10.0     50.00
                49.0  -205.604    583.00      10.0     50.00
                48.3  -208.584    583.00      10.0     50.00
                47.8  -210.766    583.00      10.0     50.00
                47.2  -213.445    583.00      10.0     50.00
                46.6  -216.194    583.00      10.0     50.00
                45.4  -221.908    583.00      10.0     50.00
                44.9  -224.379    583.00      10.0     50.00
                44.3  -227.418    583.00      10.0     50.00
                43.8  -230.014    583.00      10.0     50.00
                43.4  -232.134    583.00      10.0     50.00
                42.8  -235.388    583.00      10.0     50.00
                42.3  -238.171    583.00      10.0     50.00
                41.8  -241.020    583.00      10.0     50.00
                41.4  -243.348    583.00      10.0     50.00
                41.0  -245.722    583.00      10.0     50.00
                40.5  -248.756    583.00      10.0     50.00
                40.2  -250.612    583.00      10.0     50.00
                39.8  -253.131    583.00      10.0     50.00
                39.4  -255.701    583.00      10.0     50.00
                39.2  -257.006    583.00      10.0     50.00
                38.9  -258.988    583.00      10.0     50.00
 0
' End of irradiation history and input file
```

Overpower

```
'grp| N  |Power    | NDATA - number of power history steps to be read
'10 | D  |Factor   |
'   | A  |Scaling  |
'   | T  |Factor   |
'   | A  |         |
'   |    |         |
'   |NDATA POWFAC  |
'   |    |         |

   10   36     1.000
'
' 2nd Section of Group 10
' J | K  | power    | burnup  | coolant | coolant |sheath to| plenum  | ramp    |
' F | F  |+ve value|+ve value| temp    | pressure|coolant  | voidage | rate (s)|
' L | L  | (kW/m)  |(MWh/kgU)|         |         |heat xfer|         | or ramp |
' A | A  |-ve value|-ve value|         |         |coeff.   |         | time    |
' G | G  | total   |(hrs)    |         |         |         |         | (kW/m/s)|
'   |    |  (kW)   |         |  (K)    |  (MPa)  | (kW/m2K)| (mm3/K) |         |
'   |    |         |         |         |         |         |         |         |
'   |KFLAG  POWER  | BURNUP  |   TCL   |   CPR   | FILMCO  |  PLVD   |   RR    |
'   |    |         |         |         |         |         |         |         |
'
               58.1  -173.401    583.00      10.0     50.00
               58.5  -172.216    583.00      10.0     50.00
               59.1  -170.467    583.00      10.0     50.00
               59.6  -169.037    583.00      10.0     50.00
               60.1  -167.631    583.00      10.0     50.00
               60.1  -167.631    583.00      10.0     50.00
               59.9  -168.191    583.00      10.0     50.00
               59.5  -169.321    583.00      10.0     50.00
               59.1  -170.467    583.00      10.0     50.00
               58.6  -171.922    583.00      10.0     50.00
               58.1  -173.401    583.00      10.0     50.00
               57.4  -175.516    583.00      10.0     50.00
               56.8  -177.370    583.00      10.0     50.00
               56.1  -179.583    583.00      10.0     50.00
               55.5  -181.525    583.00      10.0     50.00
               54.8  -183.843    583.00      10.0     50.00
               54.1  -186.222    583.00      10.0     50.00
               53.5  -188.311    583.00      10.0     50.00
               52.8  -190.807    583.00      10.0     50.00
               52.1  -193.371    583.00      10.0     50.00
               50.8  -198.319    583.00      10.0     50.00
               50.2  -200.690    583.00      10.0     50.00
               49.7  -202.709    583.00      10.0     50.00
               49.1  -205.186    583.00      10.0     50.00
               48.6  -207.297    583.00      10.0     50.00
               48.0  -209.888    583.00      10.0     50.00
               47.4  -212.545    583.00      10.0     50.00
               47.0  -214.354    583.00      10.0     50.00
               46.5  -216.658    583.00      10.0     50.00
               46.0  -219.013    583.00      10.0     50.00
               45.6  -220.935    583.00      10.0     50.00
               45.2  -222.890    583.00      10.0     50.00
               44.7  -225.383    583.00      10.0     50.00
               44.4  -226.906    583.00      10.0     50.00
               44.1  -228.449    583.00      10.0     50.00
               43.8  -230.014    583.00      10.0     50.00
 0
' End of irradiation history and input file
```