Chapter 1

Software Requirement Specification for FP

Table of Units

Throughout this document SI (Système International d'Unités) is employed as the unit system. In addition to the basic units, several derived units are employed as described below. For each unit, the symbol is given followed by a description of the unit with the SI name in parentheses.

m	- for length (metre)
kg	- for mass (kilogram)
s	- for time (second)
K	- for temperature (kelvin)
^oC	- for temperature (centigrade)
J	- for energy (joule, J=$\frac{\text{kgm}^2}{\text{s}^2}$)
cal	- for energy (calorie, cal $\approx 4.2\ \frac{\text{kgm}^2}{\text{s}^2}$)
mol	- for amount of substance (mole)
W	- for power (watt, W=$\frac{\text{kgm}^2}{\text{s}^3}$)

1.1 Reference Material

This section records the information of the SRS in a form that allows easy reference throughout the document.

1.1.1 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The choice of symbols was made with the goal of being consistent with the nuclear physics literature and that used in the FP manual. The SI units are listed in brackets following the definition of the symbol.

Quantities related to Thermal Analysis

C_i	- thermal capacitance terms indexed by i ($\frac{\mathrm{kWs}}{\mathrm{m^oC}}$)
h_b	- coolant film conductance ($\frac{\mathrm{kW}}{\mathrm{m^2C}}$)
h_c	- convective heat transfer coefficient between clad and coolant ($\frac{\mathrm{kW}}{\mathrm{m^2C}}$)
h_{dry}	- convective heat transfer coefficient between fuel surface and coolant at dryout ($\frac{\mathrm{kW}}{\mathrm{m^2C}}$)
h_g	- effective heat transfer coefficient between clad and fuel surface ($\frac{\mathrm{kW}}{\mathrm{m^2C}}$)
h_p	- initial gap film conductance ($\frac{\mathrm{kW}}{\mathrm{m^2C}}$)
k_c	- clad conductivity ($\frac{\mathrm{kW}}{\mathrm{m^oC}}$)
k_{AV}	- average thermal conductivity ($\frac{\mathrm{kW}}{\mathrm{m^oC}}$)
N	- Neutron flux
p_{dry}	- dryout threshold power
q	- heat flux ($\frac{\mathrm{kW}}{\mathrm{m^2}}$)
q'''	- volumetric heat generation ($\frac{\mathrm{kW}}{\mathrm{m^3}}$)
r_c	- clad radius (m)
r_f	- fuel radius (m)
R	- resistance ($\frac{\mathrm{m^oC}}{\mathrm{kW}}$)
R_{FUEL}	- thermal resistance of fuel ($\frac{\mathrm{m^oC}}{\mathrm{kW}}$)
R_{CLAD}	- clad resistance ($\frac{\mathrm{m^oC}}{\mathrm{kW}}$)
R_{GAP}	- gap resistance ($\frac{\mathrm{m^oC}}{\mathrm{kW}}$)
R_{FILM}	- coolant film resistance ($\frac{\mathrm{m^oC}}{\mathrm{kW}}$)
T_{CL}	- centreline temperature (°C)
T_S	- surface temperature (°C)
T_1	- average fuel temperature (°C)
T_2	- average clad temperature (°C)
T_B	- coolant temperature (°C)
t	- time (s)
ρ_1	- fuel density ($\frac{\mathrm{kJ}}{\mathrm{kg^oC}}$)
ρ_2	- clad density ($\frac{\mathrm{kJ}}{\mathrm{kg^oC}}$)
τ_c	- clad thickness (m)
$c_{p,1}$	- specific heat corresponding to fuel average temperature ($\frac{\mathrm{kJ}}{\mathrm{kg^oC}}$)
$c_{p,2}$	- specific heat corresponding to clad average temperature ($\frac{\mathrm{kJ}}{\mathrm{kg^oC}}$)
$c_{p,3}$	- specific heat corresponding to fuel centerline temperature ($\frac{\mathrm{kJ}}{\mathrm{kg^oC}}$)

Quantities related to Nuclear Physics

A_k	- value of trip parameter at t_k
K_i	- response fraction
q'_{MWR}	- metal water reaction heat ($\frac{\mathrm{kW}}{\mathrm{m}}$)
q'_{MWRI}	- integrated metal water reaction heat
q'_N	- linear element power ($\frac{\mathrm{kW}}{\mathrm{m}}$)
q'_{NFRAC}	- relative fuel power
$q'_{N_{\max}}$	- linear element power at full power ($\frac{\mathrm{kW}}{\mathrm{m}}$)
q_{in}	- input heat ($\frac{\mathrm{kW}}{\mathrm{m^2C}}$)

q_{out}	- output heat ($\frac{\text{kW}}{\text{m}^2\text{C}}$)
q'_{out}	- output heat to the coolant ($\frac{\text{kW}}{\text{m}^2\text{C}}$)
W_i	- relative decay heat amplitude for i^{th} group
α_i	- i^{th} delay fraction for incore flux detector signal
β_i	- delayed neutron fraction
γ_i	- decay fraction
λ_i	- delay constant (s^{-1})
τ_A	- amplifier time constant (s)
ψ_i	- i^{th} decay constant for incore flux detector signal (s^{-1})
$\int$	- integration
$\bigtriangledown$	- gradient operator
UO_2	- uranium dioxide
ρ	- reactivity
f	- average flux depression factor
1^*	- prompt generation time (s)
$\Delta H(T_{\text{abs}})$	- fuel stored energy ($\frac{J}{kg}$)

Prefixes

Δ	- finite change in following quantity
d	- infinitesimal change in the following quantity

1.1.2 Abbreviations and Acronyms

SRS	- Software Requirements Specification
TFR	- Temperature Feedback Reactivity
MWR	- Metal Water Reaction
NOP	- Neutron Over Power
IM	- Instance Model
TM	- Theoretical Model
A	- Assumption
PS	- Physical System Description
G	- Goal

1.2 Introduction

This introduction provides an overview of the Software Requirement Specification (SRS) for fuelpin analysis within a nuclear reactor. The requirements are based on an existing theory manual and an existing code developed by a nuclear power generating company, henceforth called FP. This section explains the purpose of this document, the scope of the system, the organization of the document and the characteristics of the intended readers.

1.2.1 Purpose of Document

The main purpose of this document is to assist in the certification process for the FP code. This document provides the requirements that the FP code should implement. In particular, the goals and theoretical models used in the FP code are detailed and refined with an emphasis on explicitly identifying assumptions and unambiguous definitions. The relevant theory for FP that is presented in this SRS includes:

- the lumped parameter fuel modelling technique
- temperature dependent thermodynamic properties
- point neutron kinetics
- decay heat equations
- trip parameter modelling
- metal water reaction model
- fuel stored energy and integrated fuel power calculations.

1.2.2 Scope

The scope of the product is limited to thermal analysis and reactor physics relevant to modelling a single fuelpin. It does not include mechanical analysis or fluid dynamics. The fuelpin is modelled in isolation with no interaction between adjacent fuelpins. Given the appropriate inputs, the code for FP is intended to do the following:

- Predict temperature histories for the reactor fuel and clad.
- Calculate the integrated fuel power and the change in UO_2 enthalpy from room temperature to the average fuel temperature.
- Model the dynamic response of signal amplifiers and trip logic.
- Model dynamically compensated self-powered in-core flux detector signals that form part of the neutron overpower trip systems.

1.2.3 Organization of Document

The organization of this document follows the template for an SRS for Scientific Computing Software proposed by [?] and [?].

1.2.4 Intended Audience

This document will be helpful to Nuclear Safety Analysts to build confidence in the theoretical model and associated code implementing it. This document will also be used as part of the certification process for FP.

1.3 General System Description

General System Description provides general information about the system, identifies the interfaces between the system and its environment, describes the user characteristics and the system constraints.

1.3.1 System Context

1. FP in a larger context of reactor analysis is used for the following:

   - running safety analysis cases.
   - model one pin to give insight into the use of multiple pins.
   - iterative part of design and safety analysis (separation of concerns so that focus can be on one thing at a time).

2. The system takes either fuel power versus time or neutron flux versus time or the reactivity transient as input and predicts the output transient reactor fuel temperature and average clad temperature as follows:

   - If neutron flux versus time is given as input, the transient fuel power is generated from fission power and decay heat components.
   - If the reactivity transient is given as input, the point neutron kinetics model is used to generate the neutron flux transient.

3. The system takes either the neutron flux or the flux detector signal and the shutdown reactivity characteristic, log rate, linear rate and NOP as trip set points to simulate the reactor trip and shutdown.

1.3.2 User Characteristics

The end user of FP should have at least an undergraduate degree in Engineering or Science. Moreover, to understand the theory behind this project, the user should have the equivalent knowledge to an introductory course on each of thermodynamics and reactor physics.

1.3.3 System Constraints

None present.

1.4 Specific System Description

The specific system description includes all of the SRS software requirements in sufficient detail to enable design and testing of a system that will satisfy the requirements [**?**].

1.4.1 Problem Description

FP is a computer program developed to simulate the reactor trip and shutdown by predicting transient reactor fuel and clad temperatures based on a lumped parameter modelling approach, by incorporating a point neutron kinetics model and flexible transient control logic.

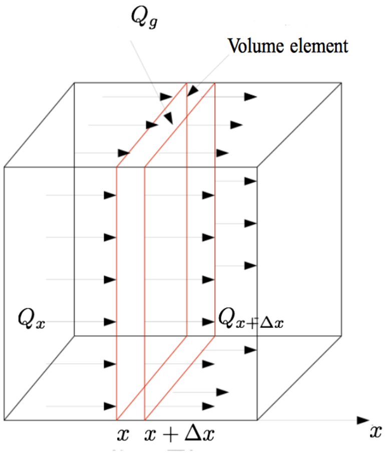

Figure 1.1: One dimensional heat conduction through a volume element

Background

Many concepts are important for understanding the theory behind FP. As a brief summary, the topics include the following: thermodynamics, point neutron kinetics, flexible transient control logic, trip logic, oxidization, enthalpy, lumped parameter modelling, heat transfer, integration and differentiation, flux depression, linear element power, thermal resistance, capacitance and conductivity, fuelpin configuration, specific heat, neutron flux, delayed neutron and prompt generations and linear interpolation.

Understanding the thermodynamic model of a fuelpin requires a basic understanding of heat transfer. To illustrate the rational behind the thermal model of the fuelpin, the heat transfer in one dimension is shown below. Following this, the analogy between heat and electrical conduction is described as it is helpful in understanding the instance models.

Heat transfer in one dimensional Cartesian coordinate system

The general heat transfer equation in 1D Cartesian coordinates can be obtained from an energy balance on a volume element in Cartesian coordinates [**?**, page 34–36]. Figure 1.1 shows a thin element of thickness Δx on a large plate.

Energy balance on this element during small time interval Δt is given as:

$$\begin{gathered}\text{(Rate of heat transfer at } x) - \text{(Rate of heat transfer at } x + \Delta x) + \text{(Rate of heat} \\ \text{generation inside the element)=(Rate of change of energy content of the element)}\end{gathered} \tag{1.1}$$

Using Q for the rate of heat transfer and E for the heat energy content, this equation can be rewritten as:

$$Q_x - Q_{x+\Delta x} + Q_g = \frac{\Delta E}{\Delta t}, \tag{1.2}$$

where Q_x is the rate of heat transfer in the x direction, Q_g is rate of heat generation inside the element and ΔE is the energy content of the element. If the volumetric heat generation in the element is q''' and the area of the plate is A, then the heat generated is:

$$Q_g = q''' A\Delta x \tag{1.3}$$

The change in energy content of the element in time Δt is $\Delta E = E_{t+\Delta t} - E_t$

The temperature (T) can be introduced through the relation that $E = \rho CVT$, where ρ is the density, C is the specific heat capacity and V is the volume of the element. In this case, $V = A\Delta x$.

$$\Delta E = \rho C_p A\Delta x(T_{t+\Delta t} - T_t) \tag{1.4}$$

Substituting 1.3, 1.4 in 1.2,

$$Q_x - Q_{x+\Delta x} + q''' A\Delta x = \rho CA\Delta x \frac{T_{t+\Delta t} - T_t}{\Delta t} \tag{1.5}$$

Dividing by $A\Delta x$ gives,

$$-(\frac{Q_{x+\Delta x} - Q_x}{A\Delta x}) + q''' = \rho C \frac{T_{t+\Delta t} - T_t}{\Delta t} \tag{1.6}$$

Taking the limit as $\Delta x \to 0$ and $\Delta t \to 0$ yields

$$-\frac{1}{A}\frac{\partial Q}{\partial x} + q''' = \rho C \frac{\partial T}{\partial t} \tag{1.7}$$

According to Fourier's law,

$$Q = -kA\frac{\partial T}{\partial x} \tag{1.8}$$

substituting 1.8 into 1.7,

$$-\frac{1}{A}\frac{\partial}{\partial x}(-kA\frac{\partial T}{\partial x}) + q''' = \rho C \frac{\partial T}{\partial t} \tag{1.9}$$

$$\frac{\partial}{\partial x}(k\frac{\partial T}{\partial x}) + q''' = \rho C \frac{\partial T}{\partial t} \tag{1.10}$$

If k is temperature independent, then the above equation simplifies to:

$$k\frac{\partial^2 T}{\partial x^2} + q''' = \rho C \frac{\partial T}{\partial t} \tag{1.11}$$

Analogy between heat conduction and electrical conduction

Figure 1.2 shows the flow of current in a circuit. From Ohm's law, we know that voltage (V) is directly proportional to resistance (R) when current (I) is kept constant; that is,

$$V = IR \tag{1.12}$$

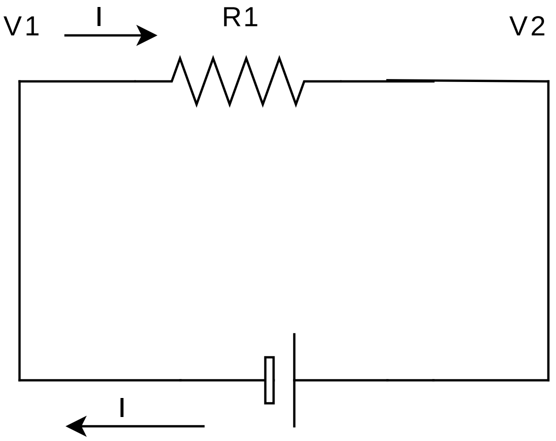

Figure 1.2: Electric circuit

When there are n resistors connected in series with resistances $R_1, R_2, R_3, ...R_n$, the current (I) is same through each resistor. The, voltage drop across all of the resistors is directly proportional to the effective resistance (R_e).

$$V = IR_e, \tag{1.13}$$

where

$$R_e = R_1 + R_2 + R_3..... + R_n \tag{1.14}$$

Figure 1.3 shows the heat flow in a slab. The thermal analogue of Ohm's law is written as,

$$\Delta T = QR_e \tag{1.15}$$

That is, the temperature drop between the surfaces of a slab is directly proportional to the thermal resistance between the surfaces, where Q is the rate of heat conduction. From Fourier's law,

$$Q = -kA\frac{dT}{dx} \tag{1.16}$$

$$dT = -\frac{Q}{kA}dx \tag{1.17}$$

Integrating LHS of 1.17 from T_1 to T_2 and RHS of 1.17 from x_1 to x_2,

$$\int_{T_1}^{T_2} dT = -\frac{Q}{kA}\int_{x_1}^{x_2} dx \tag{1.18}$$

$$\int_{T_2}^{T_1} dT = \frac{Q}{kA}\int_{x_1}^{x_2} dx \tag{1.19}$$

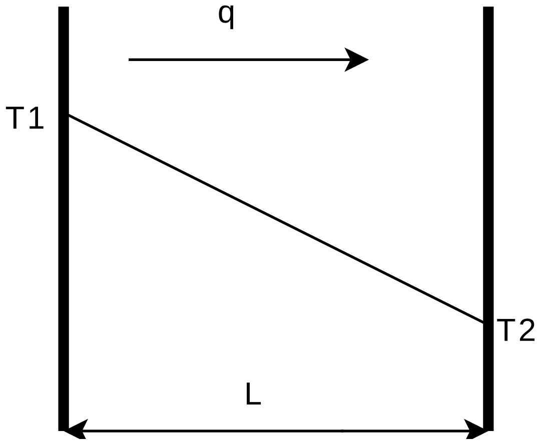

Figure 1.3: Diagram representing heat flow in a slab

$$\int_{T_2}^{T_1} dT = \frac{Q}{kA}(x_2 - x_1) \tag{1.20}$$

Since $x_2 - x_1$ is the length of the element,

$$T_1 - T_2 = Q\frac{L}{kA} \tag{1.21}$$

$$\Delta T = Q\frac{L}{kA} \tag{1.22}$$

Comparing 1.15 with 1.22, the resistance between the surfaces is defined as,

$$R_e = \frac{L}{kA}, \tag{1.23}$$

where A is the effective area of the resistance and k is the thermal conductivity.

Analogy between thermal capacitance and electrical capacitance

For electrical circuits, we have:

$$C\frac{dV}{dt} = I_{\text{in}} - I_{\text{out}}, \tag{1.24}$$

where C is the capacitor,
V is the voltage of the circuit,
I_{in} and I_{out} are the currents coming in and going out of the circuit respectively.
Equation 1.24 is analogous to the heat transfer equation:

$$C\frac{dT}{dt} = q_{in} - q_{out}, \tag{1.25}$$

where C is the capacitor,
T is the temperature,
q_{in} and q_{out} are the heats coming in and going out of the circuit respectively.

Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning. This is provided with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

- Decay heat: The heat released as a result of radioactive decay.
- Delayed neutron: Neutron emitted by one of the fission products anytime from a few milliseconds to a few minutes later.
- Delayed neutron precursors: Neutron-emitting fission fragments that undergo a stage of radioactive decay and yield an additional neutron called a delayed neutron.
- Fuel pellet: a piece of nuclear fuel usually in the shape of a sphere or cylinder.
- Flux depression: The lowering of the particle's flux density in the neighbourhood of an object due to absorption of particles in the object.
- Heat Flux: The rate of heat energy transfer per unit area.
- Linear Element power: The power generated per unit length of the fuelpin.
- Prompt neutron: A neutron immediately emitted by a nuclear fission event.
- Reactor trip: Emergency shutdown in the nuclear reactors.
- Specific heat: heat capacity per unit mass
- Thermal Capacitance: The amount of heat required to change a substance's temperature by a given amount.
- Thermal Conduction: the transfer of heat energy through a substance.
- Thermal Diffusivity: The thermal conductivity divided by density and specific heat capacity at constant pressure.
- Thermal Resistance: Measure of a temperature difference by which an object or material resists a heat flow through it.
- Transient: Changing with time.

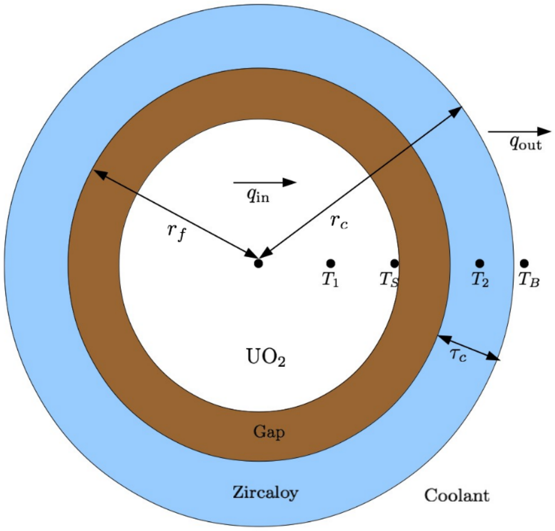

Figure 1.4: Fuel pellet representation

Physical System Description

The physical system of the FP, as shown in Figure 1.4 includes the following elements:

PS1: Fuel pellet made of Uranium dioxide (UO_2).

PS2: The clad material zircaloy covering the pellet.

PS3: Coolant surrounding the clad material.

NOTE: The temperatures T_{CL}, T_1, T_S, T_2, T_B in the Figure 1.4 will be discussed later in this document.

Goal Statements

The goals of FP are as follows:

G1: Given fuel power versus time as input, predict transient reactor fuel and clad temperatures.

G2: Given the neutron flux versus time as input, predict transient reactor fuel and clad temperatures.

G3: Given the reactivity transient as input, predict transient reactor fuel and clad temperatures.

G4: Given the trip setpoints, number of trips to initiate shutdown, shutdown reactivity transient as inputs, simulate reactor trip and shutdown.

1.4.2 Solution Characteristics Specification

This section specifies the solution characteristics of the intended software product. The purpose is to reduce the physical problem described in Section 1.4.1 to one expressed in mathematical terms. The section starts with description of the assumptions that are made, and then describes the theoretical models that are relevant for the FP problem. Data definitions, instanced models, data constraints, and the expected system behaviour are also given.

Assumptions

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the data definition or the instance model in which the respective assumption is used.

A1: Axial conduction in the pellet is ignored [GD2].

A2: The pellet has radial symmetry [GD2].

A3: Averaged thermal conductivity value is considered [DD6].

A4: The Urbanic, Heidrick model is used in modelling metal water reaction [DD5].

A5: Approximation of $\ln \frac{r_o}{r_i}$ as $\frac{\tau_c}{r}$ and r_o as r_i [DD7].

A6: Assume isotropic thermal conductivity [T2].

A7: Cylindrical coordinate system is used [GD2].

A8: The spacial effects are neglected in the reactor kinetics formulations [IM5].

A9: Newton's law of convective cooling applies in the gap between the pellet surface and the clad [DD8].

A10: Newton's law of convective cooling applies between the clad surface and the coolant film[DD9, DD12].

A11: Unit length of fuel rod is being modelled [GD2].

A12: Initially, the average clad temperature (T_2) is less than 1000^oC.

A13: The clad thickness (τ_c) is constant even if the zircaloy is consumed in the metal water reaction.

Theoretical Models

This section focuses on the general equations, laws used to model a fuelpin.

Number	T1
Label	**Conservation of energy**
Equation	$-\nabla \mathbf{q} + q''' = \rho C \frac{\partial T}{\partial t}$
Description	The above equation gives the conservation of energy for a time varying heat transfer in a material of specific heat capacity C and density ρ where $\mathbf{q}$ is the thermal flux vector, q''' is the volumetric heat generation, T is the temperature and ∇ is the gradient operator.

Number	T2
Label	**Constitutive Equation (Fourier's Law)**
Equation	$\mathbf{q}{=}-k(T)\nabla T$
Description	Fourier's law states that the heat flux is propositional to slope or the gradient of temperature, where k is a function of temperature. This law is based on the assumption that the material is isotropic (A6).

Number	T3
Label	**Space-Time kinetics**
Equation	Beyond the scope of this document.
Description	Space-Time kinetics give the relative distribution of the neutrons over space and time.

Number	T4
Label	**Decay Heat Equations**
Equation	Beyond the scope of this document.
Description	Decay heat equations are used in finding the fuel power when neutron flux is given. It is a summation of the fuel power generated by prompt fissions and the fuel power generated by delayed decay heat components due to fission product decay.

General Definitions

This section collects the laws and equations that will be used in deriving the data definitions, which in turn are used to build the instance models.

Number	GD1
Label	**Cylindrical coordinate system**
Units	-
SI equivalent	-
Equation	$\nabla = \hat{r}\frac{\partial}{\partial r} + \hat{\theta}\frac{1}{r}(\frac{\partial}{\partial \theta}) + \hat{z}\frac{\partial}{\partial z}$ where $\hat{r}$, $\hat{\theta}$ and $\hat{z}$ are unit vectors. In matrix notation, this appears as: $\nabla = \begin{bmatrix} \frac{\partial}{\partial r} \\ \frac{1}{r}\frac{\partial}{\partial \theta} \\ \frac{\partial}{\partial z} \end{bmatrix}$ The divergence ∇A is calculated as: $\nabla \mathrm{A} = \frac{\partial(A_r)}{\partial r} + \frac{1}{r}\frac{\partial A_\theta}{\partial \theta} + \frac{\partial A_z}{\partial z}$
Description	The spatial location in a cylindrical coordinate system is expressed in terms of $\hat{r}$, $\hat{\theta}$, $\hat{z}$ as shown in the Figure 1.5. The gradient operator is defined as shown above.
Sources	[**?**, page 12]

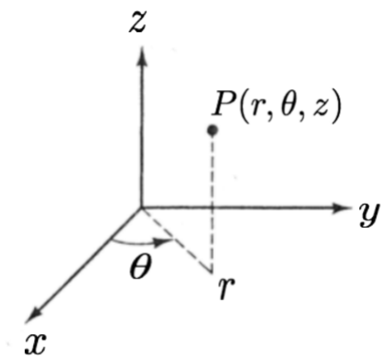

Figure 1.5: Cylindrical coordinate system

Number	GD2
Label	**Average temperature of a hollow cylinder**
Units	$M^0L^0t^0T$
SI equivalent	^oC
Symbol	T_{AVG}
Equation	$T_{\text{AVG}} = \frac{1}{A}\int_A T(r)dA$, with $T(r)$ satisfying $\frac{1}{r}\frac{d}{dr}(kr\frac{dT(r)}{dr}) + q''' = 0$
Description	T_{AVG} is the average temperature of the cylinder, A is the area of the cylinder and $T(r)$ is the temperature at radius r.

Detailed derivation of average temperature:

Applying the Fourier's law from TM2 to Conservation of energy equation in TM1, gives

$$\nabla k \nabla T + q''' = \rho C \frac{\partial T}{\partial t} \tag{1.26}$$

In steady state, the transient features die and (29) changes to

$$\nabla k \nabla T + q''' = 0 \tag{1.27}$$

Applying A7 and writing the above equation in cylindrical coordinate system (GD1),

$$k[\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial T}{\partial r}) + \frac{1}{r^2}\frac{\partial}{\partial \theta}(\frac{\partial T}{\partial \theta}) + \frac{\partial}{\partial z}(\frac{\partial T}{\partial z})] + q''' = 0 \tag{1.28}$$

Ignoring axial conduction (A1), makes $\frac{1}{r^2}\frac{\partial}{\partial \theta}(k\frac{\partial T}{\partial \theta}) = 0$.
and having radial symmetry (A2) makes $\frac{\partial}{\partial z}(k\frac{\partial T}{\partial z}) = 0$.
So, applying A1 and A2 to (31), it simplifies to:

$$\frac{1}{r}\frac{d}{dr}(kr\frac{dT}{dr}) + q''' = 0 \tag{1.29}$$

Average temperature of the hollow cylinder is found by taking the volume averaged value of the temperature at radius r. Let r_1 be the inner radius and r_2 be the outer radius of a hollow cylinder. Then the average temperature of the cylinder is given by

$$T_{\text{AVG}} = \frac{\int_V T(r,z,\theta)dV}{V} \tag{1.30}$$

Taking A1 and A2 and A11 into account,

$$T_{\text{AVG}} = \frac{1}{A}\int_A T(r)dA \tag{1.31}$$

Number	GD3
Label	**Effective thermal resistance**
Symbol	R_{EFF}
Units	$M^{-1}L^{-2}Tt^3$
SI equivalent	$\frac{\text{m}^o\text{C}}{\text{kW}}$
Equation	$R_{\text{EFF}} = \frac{\Delta T}{q}$
Description	In some cases at steady state, the relation between the temperature change (ΔT) and the thermal flux (q) is modelled as ΔT being directly proportional to q. The proportionality constant can be derived using thermodynamic theory and then relabelled as R_{EFF}. This is analoguos to the electric circuit equation of $V{=}IR$. As for the case of electric resistors in series, if n resistors ($R_1, R_2.....R_n$) are connected in series between two temperatures and if constant heat is flowing between those temperatures, then $R_{\text{EFF}} = R_1 + R_2 + ... + R_n$

Number	GD4
Label	**Rate of change of temperature**
Equation	$C\frac{dT_{\text{AVG}}}{dt} = q_{\text{in}} - q_{\text{out}} + q_g$
Description	The basic equation governing the rate of change of temperature with time. where C is the thermal capacitance ($\frac{\text{kW}s}{m^oC}$) $q_{\text{in}}, q_{\text{out}}$ are the linear in and out heat transfer rates respectively ($\frac{\text{kW}}{m}$) and q_g is the rate of internal heat generated.

Detailed derivation of Rate of change of temperature :

Integrating T1 over the volume (V),

$$-\int_V \nabla\mathbf{q}dV + \int_V q'''dV = \int_V \rho C\frac{\partial T}{\partial t}dV \tag{1.32}$$

Applying Gauss's Divergence theorem to the first term over surface S,

$$-\int_S \mathbf{q}\cdot\hat{\mathbf{n}}dS + \int_V q'''dV = \int_V \rho C\frac{\partial T}{\partial t}dV \tag{1.33}$$

Taking A11 into consideration, the volume average gets changed to area average.

$$-\int_S \mathbf{q}\cdot\hat{\mathbf{n}}dS + \int_A q'''dA = \int_A \rho C\frac{\partial T}{\partial t}dA \tag{1.34}$$

Consider a hollow cylinder as in Figure 1.6. The integral over the surface can be summarized as $q_{in} - q_{out}$. The integral of q''' over the area becomes the generated thermal energy q'_g. Hence (1.34) can be written as,

$$q_{\text{in}} - q_{\text{out}} + q_g = \int_A \rho C\frac{\partial T}{\partial t}dA \tag{1.35}$$

Taking the time derivative of GD2.

$$\frac{dT_{\text{AVG}}}{dt} = \frac{1}{A}\int_A \frac{dT}{dt}dA \tag{1.36}$$

Rearranging the above equation,

$$A\frac{dT_{\text{AVG}}}{dt} = \int_A \frac{dT}{dt}dA \tag{1.37}$$

Assuming there are representative values of specific heat (c_p) and density (ρ) and multiplying the above equation with ρC_{rep}:

$$\rho C_{rep}A\frac{dT_{\text{AVG}}}{dt} = \int_A \rho C_{rep}\frac{\partial T}{\partial t}dA \tag{1.38}$$

Replacing the RHS of (1.35) with the LHS of (1.38),

$$\rho C_{rep}A\frac{dT_{\text{AVG}}}{dt} = q_{\text{in}} - q_{\text{out}} + q_g \tag{1.39}$$

Number	GD5
Label	**Newton's law of cooling**
Units	$MLt^{-3}T^0$
SI equivalent	$\frac{\text{kW}}{\text{m}}$
Equation	$q_{\text{newt}} = hA\Delta T(t)$
Description	Newton's law of cooling describes the convection cooling and is stated as "rate of heat loss of a body is proportional to the difference in temperatures between the body and its surroundings." q_{newt} is the thermal flux. h is the heat transfer coefficient (assumed independent of T here) ($\frac{\text{W}}{\text{m}^2\text{K}}$) A is the surface area of the heat being transferred (m^2) $\Delta T(t)$= $T(t) - T_{\text{env}}$ is the time-dependent thermal gradient between environment and object. Newton's law of cooling can be derived from Fourier's law (T2)

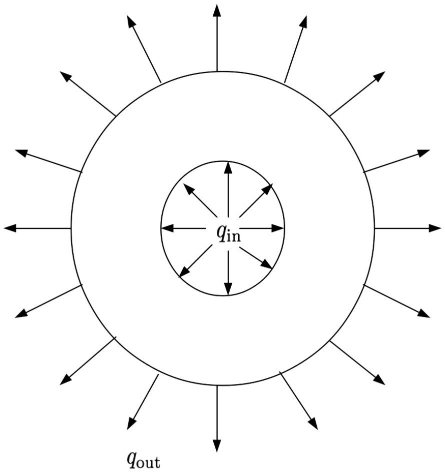

Figure 1.6: Heat transfer in a hollow cylinder

Number	GD6
Label	**Effective heat transfer coefficient**
Units	$M^{-1}L^2t^{-3}T^{-1}$
SI equivalent	$\frac{\text{W}}{\text{m}^2\text{K}}$
Equation	$h_{\text{EFF}} = \frac{q}{A\Delta T(t)}$
Description	q is the thermal flux. h_{EFF} is the effective heat transfer coefficient ($\frac{\text{W}}{\text{m}^2\text{K}}$) A is the surface area of the heat being transferred (m^2) $\Delta T(t)$= $T(t) - T_{\text{env}}$ is the time-dependent thermal gradient between environment and object. The heat transfer coefficient is modelled after Newton's law of cooling. It takes into account all relevant modes of heat transfer.

Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also declared.

Number	DD1
Label	**Relation between linear element power and volumetric heat generation**
Symbol	q'_N
Units	$MLt^{-3}T^0$
SI equivalent	$\frac{\text{kW}}{\text{m}}$
Equation	$q'_N = \pi r_f^2 q'''$
Description	q''' is the volumetric heat generation and r_f is the fuel radius. The linear element power (q'_N) is found by multiplying the volumetric heat generation by the cross-sectional area of the fuel element.
Sources	[**?**, page 2–3];

Number	DD2
Label	**Fuel stored energy**
Symbol	$\Delta H(T_{\text{abs}})$
Units	$M^0L^2t^{-2}T^0$
SI equivalent	$\frac{\text{J}}{\text{kg}}$
Equation	$\Delta H(T_{\text{abs}}) = K_0(K_1\theta((e^{\frac{\theta}{T_{\text{abs}}}} - 1)^{-1} - (e^{\frac{\theta}{298}} - 1)^{-1}) + K_2(T_{\text{abs}}^2 - 298^2) + K_3 e^{\frac{-E_D}{R_D T_{\text{abs}}}})$
Description	The stored energy ($\Delta H(T_{\text{abs}})$) calculated is the change in fuel enthalpy from room temperature ($298^o K$) to the absolute value of the average fuel temperature T_1 (T_{abs}). The values of the constants are given by the Table TB5
Sources	[**?**, page 12];

Number	DD3
Label	**Integrated fuel power**
Symbol	$P_{\text{F,SUM}}$
Units	FPS
SI equivalent	-
Equation	$P_{\text{F,SUM}}(t_i)= \int_0^{t_i} q'_{\text{NFRAC}}(t)dt$
Description	The above equation gives the integrated fuel power at t_i, where q'_{NFRAC} is the relative fuel power and $P_{\text{F,SUM}}(t_i)$ is the integrated fuel power at t_i
Sources	[**?**, page 12];

Number	DD4
Label	**Temperature feedback reactivity**
Symbol	$\rho_{\text{TFB,i}}$
Units	-
SI equivalent	mk
Equation	$\rho_{\text{TFB,i}} = A(T_{1,i} - T_{1,0})$
Description	A is a constant ($\frac{\text{mk}}{^{\circ}\text{C}}$) and $T_{1,0}$ as the initial average temperature ($^{\circ}$C)
Sources	[**?**, page 11];

Number	DD5
Label	**Metal water reaction**
Symbol	q'_{MWR}
Units	$MLt^{-3}T^{0}$
SI equivalent	$\frac{\text{kW}}{\text{m}}$
Equation	$R_{\text{ox}} = [\frac{A}{1.56\delta_{\text{ox}}}]e^{\frac{-B}{R(T_2+273)}}$ if ($\delta_{\text{ox}} \geq \tau_c$) $q'_{\text{MWR}} = 0$ else $q'_{\text{MWR}} = R_{\text{ox}} 2\pi r_c \rho_2 q_r$
Description	δ_{ox} is the reacted zircaloy thickness (m) R_{ox} is the rate of oxidization q'_{MWR} is the metal water reaction heat ($\frac{\text{kW}}{\text{m}}$) ρ_2 is the clad density ($\frac{\text{kg}}{\text{m}^3}$) r_c is the clad radius τ_c is the clad thickness T_2 is the average clad temperature q_r is the heat of reaction ($\frac{\text{kJ}}{\text{kg}}$) and its value is given in (TB1) A, B/R are constants with their values given in (TB1) The Urbanic, Heidrick model is used to predict the oxidation rate from the parabolic rate law (A4)
Sources	[**?**, page 11]

Number	DD6
Label	**Effective thermal resistance of fuel**
Symbol	R_{FUEL}
Units	$M^{-1}L^{-2}t^3T$
SI equivalent	$\frac{\text{m}^\text{o}\text{C}}{\text{kW}}$
Equation	$R_{\text{FUEL}} = \frac{f}{4\pi k_{\text{AV}}}$
Description	R_{FUEL} is the effective thermal resistance between the temperatures T_{CL} and T_S. $\frac{R_{\text{FUEL}}}{2}$ is the effective thermal resistance between T_{CL} and T_1 and between T_1 and T_S f is the flux depression factor k_{AV} is the average fuel conductivity
Sources	[**?**, page 3];

Detailed derivation of R_{FUEL}:

From Equation 1.29 of (GD2),

$$\frac{1}{r}\frac{d}{dr}(kr\frac{dT}{dr}) + q''' = 0 \tag{1.40}$$

Integrating and rearranging the above equation gives:

$$kr\frac{dT}{dr} = -\int_0^r rq'''dr \tag{1.41}$$

$$\Rightarrow kr\frac{dT}{dr} = \frac{-q'''r^2}{2} \tag{1.42}$$

$$\Rightarrow k\frac{dT}{dr} = \frac{-q'''r}{2} \tag{1.43}$$

$$\Rightarrow kdT = \frac{-q'''r}{2}dr \tag{1.44}$$

For a fuel pellet with outer radius r_f, integrating the RHS of (1.44) from 0 to r_f and LHS of (1.44) from T_{CL} to T_S,

$$\int_{T_{\text{CL}}}^{T_S} dT = \frac{-q'''}{2}\int_0^{r_f} \frac{r}{k}dr \tag{1.45}$$

$$\Rightarrow T_S - T_{\text{CL}} = \Delta T = -q'''\frac{r_f^2}{4k} \tag{1.46}$$

Multiplying both sides of (1.46) by minus sign,

$$T_{\text{CL}} - T_S = \Delta T = q'''\frac{r_f^2}{4k} \tag{1.47}$$

Applying A3 to (1.47),

$$T_{\text{CL}} - T_S = \Delta T = q''' \frac{r_f^2}{4k_{\text{AV}}} \tag{1.48}$$

Replacing the volumetric heat generation by the linear element power using the relation from DD1, removes the dependence of ΔT on the pellet radius. Rewriting (1.48),

$$T_{\text{CL}} - T_S = \frac{q'_N}{4\pi k_{\text{AV}}} \tag{1.49}$$

Taking flux depression factor (f) of the fuel pellet into consideration, (1.49) can be written as,

$$T_{\text{CL}} - T_S = \Delta T = (\frac{f}{4\pi k_{\text{AV}}})q'_N \tag{1.50}$$

Comparing the above equation to (GD3) shows that the effective thermal resistance, R_{FUEL} in this case, is:

$$R_{\text{FUEL}} = \frac{f}{4\pi k_{\text{AV}}} \tag{1.51}$$

Number	DD7
Label	R_{CLAD}
Units	$M^{-1}L^{-2}t^3T$
SI equivalent	$\frac{\text{m}^{\text{o}}\text{C}}{\text{kW}}$
Equation	$R_{\text{CLAD}} = \frac{\tau_c}{2\pi r_c k_c}$
Description	The clad resistance is a function of the clad thermal conductivity. It is obtained from the expression for heat transfer by conduction through a hollow cylinder with inner radius r_i and outer radius r_o where k_c is the clad conductivity ($\frac{\text{kW}}{\text{m}^{\text{o}}\text{C}}$) and is given as, $\frac{\Delta T}{q}=\frac{\ln\frac{r_o}{r_i}}{2\pi k_c}$ Taking A5 into consideration, we get $\frac{\Delta T}{q}= \frac{\tau_c}{2\pi r_c k_c}$ Comparison to GD3, shows that effective thermal resistance $R_{\text{CLAD}} = \frac{\tau_c}{2\pi r_c k_c}$
Sources	[?, page 4], [?, page 5] ;

Number	DD8
Label	R_{GAP}
Units	$M^{-1}L^{-2}t^3T$
SI equivalent	$\frac{\text{m}^{\text{o}}\text{C}}{\text{kW}}$
Equation	$R_{\text{GAP}} = \frac{1}{2\pi r_c h_p}$
Description	R_{GAP} is the gap resistance where r_c is the clad radius (m), h_p is the initial gap conductance ($\frac{\text{kW}}{\text{m}^{2\text{o}}\text{C}}$) which is an input parameter
Sources	source code

Derivation of R_{GAP}

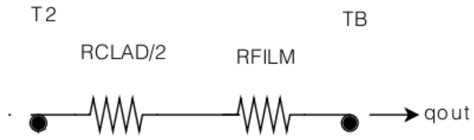

Figure 1.7: Thermal circuit between T_2 and T_B

Taking A9 into consideration, we use Newton's law of cooling to derive R_{GAP}. The area of the clad (A_c) is

$$A_c = 2\pi r_c \tag{1.52}$$

Substituting Equation 1.52 into GD5, and considering h_p as the initial gap conductance (heat transfer coefficient), we get,

$$q_{\text{gap}} = 2\pi r_c h_p \Delta T \tag{1.53}$$

From GD3, the gap resistance (R_{GAP}) can be given as,

$$R_{\text{GAP}} = \frac{\Delta T}{q_{\text{gap}}} \tag{1.54}$$

Substituting Equation 1.53 into Equation 1.54 and simplifying gives,

$$R_{\text{GAP}} = \frac{1}{2\pi r_c h_p} \tag{1.55}$$

Number	DD9
Label	R_{FILM}
Units	$M^{-1}L^{-2}t^{3}T$
SI equivalent	$\frac{\text{m}^{\text{o}}\text{C}}{\text{kW}}$
Equation	$R_{\text{FILM}} = \frac{1}{2\pi r_c h_b}$
Description	R_{FILM} is the coolant film resistance where r_c is the outer clad radius (m), h_b is the coolant film conductance ($\frac{\text{kW}}{m^2C}$) (Figure 1.7) NOTE: Equation taken from the code
Sources	source code

Derivation of R_{FILM}

Assuming that Newton's law of convective cooling applies between the clad and the coolant (A10), we use Newton's law of cooling to derive R_{FILM}. The area of the clad (A_c) is

$$A_c = 2\pi r_c \tag{1.56}$$

Substituting Equation 1.56 into GD5, and considering h_b as the coolant film conductance, we get,

$$q_{\text{coolant}} = 2\pi r_c h_b \Delta T \tag{1.57}$$

From GD3, the coolant film resistance (R_{FILM}) can be given as,

$$R_{\text{FILM}} = \frac{\Delta T}{q_{\text{coolant}}} \tag{1.58}$$

Substituting Equation 1.57 in Equation 1.58 and simplifying gives,

$$R_{\text{FILM}} = \frac{1}{2\pi r_c h_b} \tag{1.59}$$

Number	DD10
Label	R_3
Units	$M^{-1}L^{-2}t^3T$
SI equivalent	$\frac{\text{m}^{\text{o}}\text{C}}{\text{kW}}$
Equation	$R_3 = \frac{1}{2\pi r_f h_g}$
Description	R_3 is the effective thermal resistance between T_S and T_2 (Figure 1.7) where r_f is the fuel radius (m) h_g is the gap film conductance (kw/$m^{2o}C$) which is given by DD19
Sources	[**?**, page 5]

Detailed derivation of R_3:

From Figure 1.8, the effective resistance R_3 between T_S and T_2 is:

$$R_3 = R_{\text{GAP}} + \frac{R_{\text{CLAD}}}{2} \tag{1.60}$$

From GD3, the heat flux between the fuel surface and clad can be given as,

$$q = \frac{\Delta T}{R_3}, \tag{1.61}$$

Taking h_g as the effective heat transfer coefficient between fuel surface and clad, we get the heat flux between the fuel surface and clad from GD6 as

$$q = h_g A_f \Delta T, \tag{1.62}$$

where A_f is the area of the clad given as

$$A_f = 2\pi r_f \tag{1.63}$$

Comparing Equation 1.62 and Equation 1.61,

$$\frac{\Delta T}{R_3} = h_g A_f \Delta T \tag{1.64}$$

Replacing A_f with its value from Equation 1.63 and further simplifying, we get,

$$R_3 = \frac{1}{2\pi r_f h_g} \tag{1.65}$$

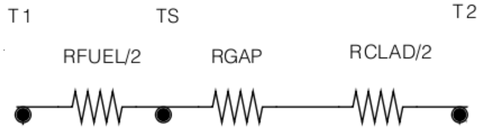

Figure 1.8: Thermal circuit between T_1 and T_2

Number	DD11
Label	R_1
Units	$M^{-1}L^{-2}t^{3}T$
SI equivalent	$\frac{\mathrm{m}^{o}\mathrm{C}}{\mathrm{kW}}$
Equation	$R_1 = \frac{f}{8\pi k_{\mathrm{AV}}} + \frac{1}{2\pi r_f h_g}$
Description	R_1 is the thermal resistance between the average fuel temperature (T_1) and clad temperature (T_2) (Figure 1.8) k_{AV} is the average fuel conductivity r_f is the fuel radius (m) h_g is the gap film conductance (kw/$m^{2o}C$) which is given by DD19
Sources	[**?**, page 4];

Derivation of R_1:

From the Figure 1.8, the effective resistance R_1 between T_1 and T_2 is:

$$R_1 = \frac{R_{\mathrm{FUEL}}}{2} + R_{\mathrm{GAP}} + \frac{R_{\mathrm{CLAD}}}{2} \tag{1.66}$$

From Equation 1.60, since R_3 is the effective resistance of gap and half of the clad, the above equation can be rewritten as,

$$R_1 = \frac{R_{\mathrm{FUEL}}}{2} + R_3 \tag{1.67}$$

Substituting the values of R_{FUEL}, R_3 from DD6 and DD10 respectively into Equation 1.67 gives:

$$R_1 = \frac{f}{8\pi k_{\mathrm{AV}}} + \frac{1}{2\pi r_f h_g} \tag{1.68}$$

Number	DD12
Label	R_2
Units	$M^{-1}L^{-2}t^3T$
SI equivalent	$\frac{\text{m}^{\text{o}}\text{C}}{\text{kW}}$
Equation	$R_2 = \frac{1}{2\pi r_c h_c}$
Description	R_2 is the effective thermal resistance between T_B and T_2 r_c is the outer clad radius (m) h_c is the effective heat transfer coefficient between clad and coolant ($\frac{\text{kw}}{\text{m}^{2o}\text{C}}$) which is given by DD18
Sources	[**?**, page 5];

Derivation of R_2

Assuming that Newton's law of convective cooling applies between the clad and the coolant (A10), we use GD6 to derive R_2. Substituting Equation 1.56 into GD6, and considering h_c as the effective heat transfer coefficient between clad and coolant, we get,

$$q = 2\pi r_c h_c \Delta T \tag{1.69}$$

From GD3, R_2 can be given as,

$$R_2 = \frac{\Delta T}{q} \tag{1.70}$$

Substituting Equation 1.69 into Equation 1.70 and simplifying gives,

$$R_2 = \frac{1}{2\pi r_c h_c} \tag{1.71}$$

Number	DD13
Label	R_{CL}
Units	$M^{-1}L^{-2}t^3T$
SI equivalent	$\frac{\text{m}^{\text{o}}\text{C}}{\text{kW}}$
Equation	$R_{\text{CL}} = \frac{f}{8\pi k_{\text{AV}}}$
Description	R_{CL} is the effective thermal resistance at the centreline, where f is the flux depression factor k_{AV} is the average fuel conductivity
Sources	[**?**, page 5];

Detailed derivation of R_{CL}:

At some radius r, the temperature T_r is given by,

$$T_r = T_{\mathrm{CL}} - q''' \frac{r^2}{4k_{\mathrm{AV}}} \tag{1.72}$$

$$= T_S + q''' \frac{r_f^2 - r^2}{4k_{\mathrm{AV}}} \tag{1.73}$$

From GD2, the average fuel temperature T_1 is defined as the area averaged value of T_r and is given by,

$$T_1 = T_S + \frac{q'''}{4k_{\mathrm{AV}} \pi r_f^2} \int_{r=0}^{r=r_f} (r_f^2 - r^2) 2\pi r dr \tag{1.74}$$

Performing the integration of (1.74) and rearranging, we get,

$$T_1 - T_S = q''' \frac{r_f^2}{8\pi k_{\mathrm{AV}}} \tag{1.75}$$

Comparing the above equation to (GD3) shows that the effective thermal resistance between T_1 and T_S (R_{CL})in this case is:

$$R_{\mathrm{CL}} = \frac{f}{8\pi k_{\mathrm{AV}}} = \frac{R_{\mathrm{FUEL}}}{2} \tag{1.76}$$

Number	DD14
Label	**Thermal capacitance terms**
Units	$ML^2t^{-2}T^{-1}$
SI equivalent	$\frac{\mathrm{kWs}}{\mathrm{m^oC}}$
Equation	$C_1 = \pi r_f^2 c_{p,1} \rho_1$ $C_2 = 2\pi r_c \tau_c c_{p,2} \rho_2$ $C_{\mathrm{CL}} = \pi r_f^2 c_{p,3} \rho_1$
Description	$c_{p,1}, c_{p,2}$ and $c_{p,3}$ are the specific heats corresponding to the fuel average, clad and fuel centreline temperatures respectively ($\frac{\mathrm{kJ}}{\mathrm{kg^oC}}$). ρ_1 and ρ_2 are the fuel and clad densities respectively ($\frac{\mathrm{kJ}}{\mathrm{kg^oC}}$). r_f and r_c are the fuel and clad radius (m) τ_c is the clad thickness
Sources	[**?**, page 5];

Number	DD15
Label	k_c
Units	$ML^1t^{-3}T^{-1}$
SI equivalent	$\frac{\mathrm{kW}}{\mathrm{m^oC}}$
Equation	$k_c = aT_2 + b$
Description	k_c is the clad conductivity where a and b are constants obtained by a least squares fit to tabulated data (TB2).
Sources	[**?**, page 6];

Number	DD16
Label	K_{AV} **as a polynomial**
Units	$ML^1t^{-3}T^{-1}$
SI equivalent	$\frac{\text{kW}}{\text{m}^{\text{o}}\text{C}}$
Equation	$k = x_1 T + x_0$
Description	k_{AV} is a temperature dependant non-linear variable and is represented as a first order polynomial function of temperature. The values of x_0 and x_1 are given in the (TB3).
Sources	-

Number	DD17
Label	$c_{p,1}$ **as a polynomial**
Units	$M^0L^2t^{-2}T^{-1}$
SI equivalent	$\frac{\text{kJ}}{\text{kg}^{\text{o}}\text{C}}$
Equation	$c_p = y_2 T^2 + y_1 T + y_0$
Description	$c_{p,1}$ is a temperature dependant non-linear variable and is represented as a second order polynomial function of temperature. The values of y_0, y_1 and y_2 are given in the (TB4).
Sources	

Number	DD18
Label	h_c
Units	$ML^0t^{-3}T^{-1}$
SI equivalent	$\frac{\text{kW}}{\text{m}^{2\text{o}}\text{C}}$
Equation	$h_c = \frac{2k_c h_b}{2k_c + \tau_c h_b}$
Description	h_c is the effective heat transfer coefficient between the clad and the coolant τ_c is the clad thickness h_b is initial coolant film conductance k_c is the clad conductivity NOTE: Equation taken from the code
Sources	source code

Derivation of h_c:

From Figure 1.7, R_2 is the effective thermal resistance of the coolant film and half of the clad. Adding the R_{FILM} from DD9 with half of the value of R_{CLAD} from DD7, we get

$$R_2 = \frac{1}{2\pi r_c h_b} + \frac{\tau_c}{4\pi r_c k_c} \tag{1.77}$$

Taking the common terms out and rewriting the above equation,

$$R_2 = \frac{1}{2\pi r_c}\left(\frac{1}{h_b} + \frac{\tau_c}{2k_c}\right) \tag{1.78}$$

$$= \frac{1}{2\pi r_c}\left(\frac{2k_c + \tau_c h_b}{2k_c h_b}\right) \tag{1.79}$$

$$= \frac{1}{2\pi r_c\left(\frac{2k_c h_b}{2k_c + \tau_c h_b}\right)} \tag{1.80}$$

But from DD12, R_2 is given as,

$$R_2 = \frac{1}{2\pi r_c h_c} \tag{1.81}$$

Comparing Equation 1.80 and Equation 1.81 and rearranging gives h_c as,

$$h_c = \frac{2k_c h_b}{2k_c + \tau_c h_b} \tag{1.82}$$

Number	DD19
Label	h_g
Units	$ML^0t^{-3}T^{-1}$
SI equivalent	$\frac{\text{kW}}{\text{m}^2{}^\circ\text{C}}$
Equation	$h_g = \frac{2k_c h_p}{2k_c + \tau_c h_p}$
Description	h_g is the gap conductance τ_c is the clad thickness h_p is initial gap film conductance k_c is the clad conductivity NOTE: Equation taken from the code
Sources	source code

Derivation of h_g:

From the Figure 1.8, the effective thermal resistance between T_2 and T_S is, the effective resistance of the gap film and half of the clad. Adding the R_{GAP} from DD8 with half of the value of R_{CLAD} from DD7, we get

$$R_3 = \frac{1}{2\pi r_c h_p} + \frac{\tau_c}{4\pi r_c k_c} \tag{1.83}$$

where R_3 is the effective resistance

Taking the common terms out and rewriting the above equation,

$$R_3 = \frac{1}{2\pi r_c}\Big(\frac{1}{h_p} + \frac{\tau_c}{2k_c}\Big) \tag{1.84}$$

$$= \frac{1}{2\pi r_c}\Big(\frac{2k_c + \tau_c h_p}{2k_c h_p}\Big) \tag{1.85}$$

$$= \frac{1}{2\pi r_c\Big(\frac{2k_c h_p}{2k_c + \tau_c h_p}\Big)} \tag{1.86}$$

But from DD10, R_3 is given as,

$$R_3 = \frac{1}{2\pi r_f h_g} \tag{1.87}$$

Comparing Equation 1.87 and Equation 1.86 and rearranging gives h_g as,

$$h_g = \frac{2k_c h_p}{2k_c + \tau_c h_p} \tag{1.88}$$

Number	DD20
Label	**Incore flux detector signal**
Units	-
SI equivalent	-
Equation	D= $(1 - \hat{\alpha})N + \Sigma_{i=1}^{5} d_i$ $d_i = \hat{\psi}_i(\hat{\alpha}_i N - d_i)$
Description	D is the relative detector signal amplitude $\hat{\alpha_i}$ is the i^{th} delay fraction $\hat{\alpha}$ is the total delayed fraction $\hat{\psi_i}$ is the i^{th} decay constant (s^{-1}) d_i is the relative amplitude of delay group i N is the neutron flux
Sources	[**?**, page 10];

Number	DD21
Label	**Compensated detector signal**
Units	-
SI equivalent	-
Equation	$D'(s) = D(K_1 + \frac{K_2}{1+T_2 s} + \frac{K3}{1+T_3 s})$
Description	The compensated signal D' is given by using the Laplace transformation of $D(s)$ D is the relative detector signal amplitude K_1 is the prompt response fraction K_2, K_3 are the delayed response fractions T_2, T_3 are the delay times (s)
Sources	[**?**, page 10];

Number	D22
Label	**Signal amplifier response**
Units	-
SI equivalent	-
Equation	$A_K^* = A_{K-1}^* e^{\frac{-\Delta t}{\tau_A}} + (1 - e^{\frac{-\Delta t}{\tau_A}})A_K$
Description	A_K is the value of the trip parameter at t_k A_K^* is the filtered value of the trip parameter at t_k τ_A is the amplifier time constant(s) The delay introduced by the signal amplifier is simulated for each trip parameter by the above first order filter. The log rate signal is filtered by two cascaded first order filters.
Sources	[**?**, page 11];

Number	DD23
Label	**Fuel surface temperature**
Units	$M^0L^0t^0T$
SI equivalent	°C
Equation	$T_S = T_2 + \frac{T_1 - T_2}{R_1} R_3$
Description	T_S is the surface fuel temperature T_1 is the average fuel temperature T_2 is the average clad temperature R_1 is the effective resistance between average fuel and average clad temperatures ($\frac{\mathrm{m}^o\mathrm{C}}{\mathrm{kW}}$) R_3 is the effective resistance between the clad and the fuel surface temperatures ($\frac{\mathrm{m}^o\mathrm{C}}{\mathrm{kW}}$)
Sources	[**?**, page 6];

Derivation of T_S:

From GD3, the heat flux generated between T_S and T_2 can be given as,

$$q_{\text{surface}} = \frac{T_S - T_2}{R_{\text{GAP}} + \frac{R_{\text{CLAD}}}{2}}, \tag{1.89}$$

where $R_{\text{GAP}} + \frac{R_{\text{CLAD}}}{2}$ is the effective resistance between T_S and T_2 and $T_S - T_2 = \Delta T$. Replacing effective resistance between T_S and T_2 by R_3 (DD10), Equation 1.89 simplifies to,

$$q_{\text{surface}} = \frac{T_S - T_2}{R_3}, \tag{1.90}$$

Similarly the heat flux generated between T_1 and T_2 can be given as,

$$q_{\text{avgfuel}} = \frac{T_1 - T_2}{R_1}, \tag{1.91}$$

where R_1 is the effective resistance between T_1 and T_2 and $T_1 - T_2 = \Delta T$.

By the continuity of thermal flux, Equation 1.90 becomes equal to Equation 1.91. That is,

$$\frac{T_S - T_2}{R_3} = \frac{T_1 - T_2}{R_1} \tag{1.92}$$

Rearranging the above equation gives,

$$T_S = T_2 + \frac{T_1 - T_2}{R_1} R_3 \tag{1.93}$$

Number	DD24
Label	**Linear and Log rates**
Symbol	$R_{\text{LIN}}, R_{\text{LOG}}$
Units	$M^0L^0t^0T^{-1}$
SI equivalent:	s^{-1}
Equation	$R_{\text{LIN}} = \frac{N_k - N_{k-1}}{\Delta t}$ $R_{\text{LOG}} = \frac{\ln N_k - \ln N_{k-1}}{\Delta t}$
Description	R_{LIN} is the linear rate (s^{-1}) R_{LOG} is the log rate (s^{-1}) N_k is the relative neutron flux at t_k Δt is the solution interval $t_k - t_{k-1}$
Sources	[**?**, page 10];

Number	DD25
Label	**Relation between linear element power and relative fuel power**
Units	$MLt^{-3}T^0$
SI equivalent	$\frac{\text{kW}}{\text{m}}$
Equation	$q'_N = q'_{\text{NFRAC}} q'_{N_{\text{max}}}$
Description	q'_{NFRAC} is the relative fuel power q'_N is linear element power ($\frac{\text{kW}}{\text{m}}$) $q'_{N_{\text{max}}}$ is the full power linear element power ($\frac{\text{kW}}{\text{m}}$)
Sources	[**?**, page 9];

Number	DD26
Label	q'_{MWRI}
Units	-
SI equivalent	-
Equation	$q'_{\text{MWRI}}(t_i) = \frac{1}{q'_{N_{\text{max}}}} \int_0^{t_i} q'_{\text{MWR}}(t)dt$
Description	$q'_{\text{MWRI},N}$ gives the integrated metal water reaction heat at t_N. It is a summation of q'_{MWR} normalized by $q'_{N_{\text{max}}}$ at each time step $q'_{\text{MWR},i}$ is the metal water reaction heat at t_i

Number	DD27
Label	q'_{out}
Units	$Mt^{-3}T^{-1}$
SI equivalent	$\frac{\text{kW}}{\text{m}^{2\text{o}}\text{C}}$
Equation	$q'_{\text{out}} = \frac{1}{q'_{N_{\max}}}\left(\frac{T_2 - T_B}{R_2}\right)$
Description	q'_{out} is the output heat from the reaction that is sent into the coolant R_2 is the effective resistance between the coolant film and the clad ($\frac{\text{m}^{\text{o}}\text{C}}{\text{kW}}$) $q'_{N_{max}}$ is the linear element power at full power ($\frac{\text{kW}}{\text{m}^{\text{o}}\text{C}}$) T_2 is average clad temperature T_B is coolant temperature NOTE: Equation taken from the code

Derivation of q'_{out}:

From the Figure 1.7, the effective resistance R_2 between T_2 and T_B is given by DD12. According to GD3, the R_2 between T_2 and T_B can be given as,

$$R_2 = \frac{\Delta T}{q'_{\text{out}}}, \tag{1.94}$$

where q'_{out} is the heat generated between T_2, T_B and

$$\Delta T = T_2 - T_B \tag{1.95}$$

Substituting Equation 1.95 and DD12 in 1.94 and rearranging gives,

$$q'_{\text{out}} = \frac{(T_2 - T_B)}{R_2} \tag{1.96}$$

Normalizing the above equation by $q'_{N_{\max}}$ (standard presentation) gives,

$$q'_{\text{out}} = \frac{1}{q'_{N_{\max}}}\frac{(T_2 - T_B)}{R_2} \tag{1.97}$$

Number	DD28
Label	dryout
Units	–
SI equivalent	–
Equation	if($q'_{\text{out}} \geq p_{\text{dry}}$) $h_b = h_{\text{dry}}$
Description	We compare the power sent to the coolant (q'_{out}) with the dryout threshold power (p_{dry}) and once it exceeds this maximum permissible power, dryout occurs. When dryout occurs, the coolant film conductance (h_b) becomes equal to the heat transfer coefficient between the fuel surface and the coolant at dryout (h_{dry}). NOTE: The meaning of h_b is given through the definition of R_{FILM} (DD9)

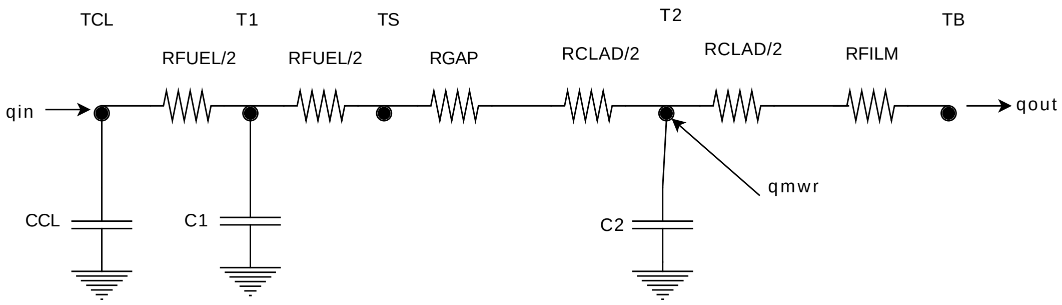

Figure 1.9: Electrical Circuit Analogue

Instance Models

This section reduces the problem defined in the problem description into one which is expressed in mathematical terms. It uses concrete symbols defined in section "Data Definitions" to replace the abstract symbols in the models identified in the section "Theoretical Models" and "General Definitions".

Number	IM1
Label	Rate of change of average fuel temperature
Equation	$C_1 \frac{dT_1}{dt} = q'_N - \frac{T_1 - T_2}{R_1}$
Description	T_1 is the average fuel temperature T_2 is the average clad temperature R_1 is the effective resistance between average fuel and average clad temperatures C_1 is the thermal capacitance of the fuel
Sources	[**?**, page 6];

Derivation of Rate of change of average fuel temperature:

To find the rate of change of average fuel temperature, we use GD4. Consider $c_{p,1}$ as the specific heat evaluated at T_1, ρ_1 as the density of the fuel and A_f as area of fuelpellet which is given as,

$$A_f = \pi r_f^2 \tag{1.98}$$

Now substituting $c_{p,1}$, ρ_1 and Equation 1.98 in Equation 1.39 gives,

$$\rho_1 c_{p,1} \pi r_f^2 \frac{dT_1}{dt} = q_{\text{in}} - q_{\text{out}} + q_g \tag{1.99}$$

From DD14 the capacitance term C_1 is given as,

$$C_1 = \pi r_f^2 c_{p,1} \rho_1 \tag{1.100}$$

Rewriting Equation 1.99 and substituting Equation 1.100 in Equation 1.99 gives

$$C_1 \frac{dT_1}{dt} = q_{\text{in}} - q_{\text{out}} + q_g \tag{1.101}$$

The amount of q_{in} is zero at T_1. That is,

$$q_{\text{in}} = 0 \tag{1.102}$$

The value for q_{out} is given by the flux between T_1 and T_2. From the Figure 1.9,

$$q_{\text{out}} = \frac{T_1 - T_2}{R_1}, \tag{1.103}$$

where T_2 is the average clad temperature and R_1 is the effective thermal resistance between T_1 and T_2 as given in DD2.43.

The integral of heat generation is,

$$q_g = q'_N \tag{1.104}$$

Substituting Equation 1.102, Equation 1.103 and Equation 1.104 in Equation 1.101 and rearranging gives,

$$C_1 \frac{dT_1}{dt} = q'_N - \frac{T_1 - T_2}{R_1} \tag{1.105}$$

Number	IM2
Label	Rate of change of average clad temperature
Equation	$C_2 \frac{dT_2}{dt} = \frac{T_1 - T_2}{R_1} + q'_{\text{MWR}} - \frac{T_2 - T_B}{R_2}$
Description	T_1 is the average fuel temperature T_2 is the average clad temperature T_B is the coolant temperature R_1 is the effective resistance between average fuel and average clad temperatures R_2 is the effective resistance between clad and coolant temperatures C_2 is the thermal capacitance of the clad q_{MWR} is the Metal-Water reaction heat
Sources	[**?**, page 6];

Derivation of Rate of change of average clad temperature:

Similar to the rate of change of average fuel temperature, to find the rate of change of average clad temperature, we use GD4. Consider $c_{p,2}$ as the specific heat evaluated at T_2, ρ_2 as the density of the clad and A_c as area of clad which is given as,

$$A_c = 2\pi r_c \tau_c \tag{1.106}$$

Now substituting $c_{p,2}$, ρ_2 and Equation 1.106 in Equation 1.39 gives,

$$\rho_2 c_{p,2} 2\pi r_c \tau_c \frac{dT_2}{dt} = q_{\text{in}} - q_{\text{out}} + q_g \tag{1.107}$$

From DD14 the capacitance term C_2 is given as,

$$C_2 = 2\pi r_c \tau_c c_{p,2} \rho_2 \tag{1.108}$$

Rewriting Equation 1.107 and substituting Equation 1.108 in Equation 1.107 gives

$$C_2 \frac{dT_2}{dt} = q_{\text{in}} - q_{\text{out}} + q_g \tag{1.109}$$

q_{in} at T_2 is the amount of q_{out} at T_1 (1.103). That is,

$$q_{\text{in}} = \frac{T_1 - T_2}{R_1} \tag{1.110}$$

The value for q_{out} is given by the flux between T_2 and T_B. From the Figure 1.9,

$$q_{\text{out}} = \frac{T_2 - T_B}{R_2}, \tag{1.111}$$

where R_2 is the effective thermal resistance between T_2 and T_B as given in DD12.
The integral of heat generation is,

$$q_g = q'_{\text{MWR}} \tag{1.112}$$

Substituting Equation 1.110, Equation 1.111 and Equation 1.112 in Equation 1.109 and rearranging gives,

$$C_2 \frac{dT_2}{dt} = \frac{T_1 - T_2}{R_1} + q'_{\text{MWR}} - \frac{T_2 - T_B}{R_2} \tag{1.113}$$

Number	IM3
Label	Rate of change of centerline temperature
Equation	$C_{\text{CL}} \frac{dT_{\text{CL}}}{dt} = q'_N - \frac{T_{\text{CL}} - T_1}{R_{\text{CL}}}$
Description	T_1 is the average fuel temperature T_{CL} is the centerline temperature R_{CL} is the effective resistance t the centerline C_{CL} is the thermal capacitance at the centerline q'_N is the linear element power
Sources	[**?**, page 6];

Derivation of Rate of centerline temperature:

Similar to the rate of change of average fuel temperature, to find the rate of change of centerline temperature, we use GD4. Consider $c_{p,3}$ as the specific heat evaluated at T_{CL}, ρ_1 as the density of the fuel and A_f as area of fuel pellet which is given by Equation 1.98. Now substituting $c_{p,3}$, ρ_1 and Equation 1.98 in Equation 1.39 gives,

$$\rho_1 c_{p,3} \pi r_f^2 \frac{dT_{\text{CL}}}{dt} = q_{\text{in}} - q_{\text{out}} + q_g \tag{1.114}$$

From DD14 the capacitance term C_{CL} is given as,

$$C_{\text{CL}} = \pi r_f^2 c_{p,3} \rho_1 \tag{1.115}$$

Rewriting Equation 1.114 and substituting Equation 1.115 in Equation 1.114 gives

$$C_{\text{CL}} \frac{dT_{\text{CL}}}{dt} = q_{\text{in}} - q_{\text{out}} + q_g \tag{1.116}$$

From the Figure 1.9, q_{in} at T_{CL} is the linear element power q'_N. That is,

$$q_{\text{in}} = q'_N \tag{1.117}$$

The value for q_{out} is given by the flux between T_{CL} and T_1. From the Figure 1.9,

$$q_{\text{out}} = \frac{T_{CL} - T_1}{R_{\text{CL}}}, \tag{1.118}$$

where R_{CL} is the effective thermal resistance between T_{CL} and T_1 as given in DD13.
The integral of heat generation is,

$$q_g = 0 \tag{1.119}$$

Substituting Equation 1.117, Equation 1.118 and Equation 1.119 in Equation 1.116 and rearranging gives,

$$C_{\text{CL}} \frac{dT_{\text{CL}}}{dt} = q'_N - \frac{T_{\text{CL}} - T_1}{R_{\text{CL}}} \tag{1.120}$$

Number	IM4
Label	Initial thermal conditions
Equation	$\frac{dT_1}{dt} = 0$ (IM1) $\frac{dT_2}{dt} = 0$ (IM2) $k_c = aT_2 + b$ (DD15) $q'_{\text{MWR}} = 0$ $q'_{\text{N,est}} = 4\pi \int_{T_S}^{T_{\text{CL,est}}} k dT = q'_N$
Description	The initial values that are needed to start the simulation are calculated using the above conditions. T_1 is the average fuel temperature T_2 is the average clad temperature T_S is the surface temperature $T_{\text{CL,est}}$ is the estimate of centerline temperature k_c is the clad conductivity which is given by DD15 a, b are constants with their values given by Table TB2 q'_{MWR} is the Metal-Water reaction heat $q'_{\text{N,est}}$ is the estimate of linear element power k is the fuel conductivity
Sources	[**?**, page 6];

Number	IM5
Label	Point neutron kinetics
Equation	$\dot{N} = [\frac{\rho-\beta}{l^*}]N + \Sigma_{i=1}^{6}\lambda_i c_i$ $\dot{c_i} = -\lambda_i c_i + (\frac{\beta_i}{l^*})N$
Description	N is the neutron flux ρ is the reactivity β_i is the fraction associated with the i^{th} group of delayed neutron precursors β is total delayed neutron fraction l^* is the prompt generation time(s) λ_i is the decay constant associated with the i^{th} group of delayed neutron precursors (s^{-1}) c_i is the delayed neutron precursor for the i^{th} group To make the relative distribution of the neutrons uniform, the space effects on the kinetics equations are to be eliminated. Hence taking A8 into consideration, i.e considering only time effects, the space-time kinetics equations of T3 is reduced to the point kinetics equations. If the reactivity transient is given as input, the transient neutron flux is obtained by solving the point kinetics equations.
Sources	[?, page 6];

Number	IM6
Label	Decay Heat Equations
Equation	$q'_{\text{NFRAC}} = (1-\gamma)N + \Sigma_{i=1}^{3}\lambda_{h,i}W_i$ $\dot{W_i} = -\lambda_{h,i}W_i + \gamma_i N$
Description	N is the neutron flux q'_{NFRAC} is the relative fuel power γ_i is the power fraction associated with the i^{th} decay heat group $\lambda_{h,i}$ is the decay constant associated with the i^{th} decay heat group (s^{-1}) γ is the total delayed power fraction W_i is the relative decay heat amplitude for the i^{th} group The decay equations are used in generating fuel power. The total fuel power is a summation of a prompt neutronic power component (the prompt fission power) and three delayed decay heat components due to fission product decay. If the neutron flux transient is given as input or from the point kinetics routine, the transient fuel power is obtained by solving the decay heat equations.
Sources	[?, page 6];

Data Constraints

This section is to clarify the environmental and system limitations imposed on admissible data. It gives the system constraints on the data to validate the models identified in the section "Instance Models". These constraints are listed in the table below. The column physical constraints gives the physical limitations on the range of the values that can be taken by the variable and are given by the domain expert while the column system constraints gives the system limitations on the range of values that can be taken by the variable and are given by the developer. In this case no system constraints are imposed, so this column is not included.

Variable	Type	Unit	Physical Constraints	Typical value	Property
$\lambda_{h,1}$	Real	s^{-1}	$-\infty < \lambda_{h,1} < \infty$	0.10368	IN
$\lambda_{h,2}$	Real	s^{-1}	$-\infty < \lambda_{h,2} < \infty$	0.000142	IN
$\lambda_{h,3}$	Real	s^{-1}	$-\infty < \lambda_{h,3} < \infty$	0.00476	IN
γ_1	Real	-	$-\infty < \gamma_1 < \infty$	0.0226	IN
γ_2	Real	-	$-\infty < \gamma_2 < \infty$	0.02311	IN
γ_3	Real	-	$-\infty < \gamma_3 < \infty$	0.02078	IN
N	Real		to be discussed		IN-OUT
ρ_1	Real	$\frac{\mathrm{kg}}{\mathrm{m}^3}$	$-\infty < \rho_1 < \infty$	10600	IN
ρ_2	Real	$\frac{\mathrm{kg}}{\mathrm{m}^3}$	$-\infty < \rho_2 < \infty$	6570	IN
ρ (reactivity)	Real		to be discussed	0	IN-OUT
τ_c	Real	m	$-\infty < \tau_c < \infty$	0.00042	IN
τ_g	Real	m	$-\infty < \tau_g < \infty$	0.00004	IN
τ_A	Real	-	$-\infty < \tau_A < \infty$	0.015	IN
ΔT	Real	s	$0 < \Delta T \leq 0.0001$	0.01	IN
f	Real	-	$-\infty < f < \infty$	1.0	IN
l^*	Real	s	$-\infty < l^* < \infty$	0.893×10^{-3}	IN
β_1	Real	-	$-\infty < \beta_1 < \infty$	1.769×10^{-4}	IN
β_2	Real	-	$-\infty < \beta_2 < \infty$	11.498×10^{-4}	IN
β_3	Real	-	$-\infty < \beta_3 < \infty$	10.191×10^{-4}	IN
β_4	Real	-	$-\infty < \beta_4 < \infty$	21.057×10^{-4}	IN
β_5	Real	-	$-\infty < \beta_5 < \infty$	7.726×10^{-4}	IN
β_6	Real	-	$-\infty < \beta_6 < \infty$	1.962×10^{-4}	IN
β_7	Real	-	$-\infty < \beta_7 < \infty$	1.61×10^{-7}	IN
β_8	Real	-	$-\infty < \beta_8 < \infty$	3.23×10^{-7}	IN
β_9	Real	-	$-\infty < \beta_9 < \infty$	1.03×10^{-6}	IN
β_{10}	Real	-	$-\infty < \beta_{10} < \infty$	7.48×10^{-6}	IN
β_{11}	Real	-	$-\infty < \beta_{11} < \infty$	6.61×10^{-6}	IN
β_{12}	Real	-	$-\infty < \beta_{12} < \infty$	1.080×10^{-5}	IN
β_{13}	Real	-	$-\infty < \beta_{13} < \infty$	2.240×10^{-5}	IN
β_{14}	Real	-	$-\infty < \beta_{14} < \infty$	6.52×10^{-5}	IN
β_{15}	Real	-	$-\infty < \beta_{15} < \infty$	2.080×10^{-4}	IN
λ_1	Real	s^{-1}	$-\infty < \lambda_1 < \infty$	12.778×10^{-3}	IN
λ_2	Real	s^{-1}	$-\infty < \lambda_2 < \infty$	31.535×10^{-3}	IN
λ_3	Real	s^{-1}	$-\infty < \lambda_3 < \infty$	122.197×10^{-3}	IN
λ_4	Real	s^{-1}	$-\infty < \lambda_4 < \infty$	32.282×10^{-2}	IN
λ_5	Real	s^{-1}	$-\infty < \lambda_5 < \infty$	1389.289×10^{-3}	IN
λ_6	Real	s^{-1}	$-\infty < \lambda_6 < \infty$	3778.336×10^{-3}	IN
λ_7	Real	s^{-1}	$-\infty < \lambda_7 < \infty$	6.26×10^{-7}	IN
λ_8	Real	s^{-1}	$-\infty < \lambda_8 < \infty$	3.63×10^{-6}	IN
λ_9	Real	s^{-1}	$-\infty < \lambda_9 < \infty$	4.37×10^{-5}	IN
λ_{10}	Real	s^{-1}	$-\infty < \lambda_{10} < \infty$	0.117×10^{-3}	IN

λ_{11}	Real	s^{-1}	$-\infty < \lambda_{11} < \infty$	0.428×10^{-3}	IN
λ_{12}	Real	s^{-1}	$-\infty < \lambda_{12} < \infty$	0.150×10^{-2}	IN
λ_{13}	Real	s^{-1}	$-\infty < \lambda_{13} < \infty$	0.481×10^{-2}	IN
λ_{14}	Real	s^{-1}	$-\infty < \lambda_{14} < \infty$	$0.169\text{*}10^{-1}$	IN
λ_{15}	Real	s^{-1}	$-\infty < \lambda_{15} < \infty$	0.277	IN
$q'_{N,\max}$	Real	$\frac{\text{kW}}{\text{m}}$	$35 \leq q'_{N,\max} \leq 65$	62.9540	IN
h_b	Real	$\frac{\text{kW}}{\text{m}^{2o}\text{C}}$	$-\infty < h_b < \infty$	50	IN
h_p	Real	$\frac{\text{kW}}{\text{m}^{2o}\text{C}}$	$-\infty < h_p < \infty$	10	IN
T_B	Real	^oC	$-\infty < T_B < \infty$	305.0	IN
r_f	Real	m	$-\infty < r_f < \infty$	0.00612	IN
p_{dry}	Real	fp	$-\infty < p_{\text{dry}} < \infty$	1.176	IN
h_{dry}	Real	$\frac{\text{kW}}{\text{m}^{2o}\text{C}}$	$-\infty < h_{\text{dry}} < \infty$	2.0	IN

System Behaviour

This section gives a detailed description of the system's functionalities based on the information in the sections"Data Constraints" and "Instance Models". It formally specifies the flow of processing the data. That is, from getting the input, applying the models and producing the output. The responses to undesired situations such as the errors that are to be generated if the data constraints are not satisfied are also stated. The contents of this section are used in design and testing.

```
Step 1: Read data from the Input file

  1. Read the driving transient data
       If the driving transient is reactivity, then
         Read the reactivity transient data ($\rho(t)$)
       Else if the driving transient is neutron flux, then
         Read the Neutron flux transient data ($N(t)$)
       Else
         Read Fuel power transient data ($q'_{\text{NFRAC}}(t)$)

  2. Read the inputs
       Read the inputs $\Delta t, \tau_c, \tau_g, h_g, h_b, \rho 1, \rho 2, r_i$
       If the driving transient is $\rho(t)$, then
         Read $l^*$, $\beta_i$, $\gamma_i$
       If the driving transient is $\rho(t)$ or $N(t)$, then
         Read the inputs $\gamma_{h,i}$, $\lambda_i$

Step 2: Process/Output:
    For each time step,

  1. If the driving transient is $\rho(t)$, then
       Determine reactivity based on current time
     Else if the driving transient is $N(t)$, then
       Determine neutron flux based on current time
     Else if the driving transient is $q'_{\text{NFRAC}}(t)$, then
       Determine fuel power based on current time
```

2. If the driving transient is $\rho(t)$, then

    (a) Generate the Neutron flux solving Point kinetics equations from IM5

    (b) Output Neutron flux (N)

    Else if the driving transient is $\rho(t)$ or $N(t)$, then

    (a) Generate the relative fuel power by solving Decay heat equations from IM6

    (b) Output the relative fuelpower (q'_{NFRAC})

    Else if the driving transient is $\rho(t)$ or $N(t)$ or $q'_{\text{NFRAC}}(t)$, then

    (a) Change the relative fuel power to linear element power using DD25

    (b) Use the linear element power to determine average fuel temperature, average clad temperature and centerline temperature by solving IM1, IM3 ,IM2 respectively

    (c) Output T_1, T_2, T_{CL}

    (d) Use T_1, T_2 to determine Surface temperature (T_S) by solving DD23

    (e) Output T_S

    (f) Using the generated T_1, find the fuel stored energy $(\Delta H(T_1))$ using DD2 and the power to the coolant (q'_{out}) using DD27

    (g) Find the fuel power (f_p) using DD3
    If Metal water reaction's calculations are desired, then

    (h) Read p_{dry}, h_{dry}, δ_{ox}

    (i) Calculate the Metal water reaction heat using DD5 and DD26

1.4.3 Nonfunctional Requirements

This section specifies system requirements that consider the quality and behaviour of the system as a whole. It provides different specifications for the system, so that it is found acceptable and pleasant to use.These include: Accuracy/performance requirements, maintainability requirements.

Accuracy

The relative error between FP code and HOTSPOT code for the test cases specified in the FP Engineer's manual should not be more than 0.05

Maintainability

The effort put in maintaining the product should be less than 1/4th of the amount of efforts put in building, developing the software.

Solution Validation Strategies

This section establishes the strategies for validating the software product, and the specific tests to be performed to assert it complies with the requirements specification defined in the previous section. To validate the solution produced by the software,

1. Results of the FP code are compared with those of the HOTSPOT code for a test transient.

- An idealized transient involving severe overpower, combined with step changes in coolant temperature and an order of magnitude reduction in convective heat transfer to coolant, should be run with both codes.
- Runs are to performed both with and without average flux depression factor.

1.5 Other System Issues

This section provides additional information "on the side" about FP

1.5.1 Open Issues

None present

1.5.2 Off-the-Shelf Solutions

None present

1.5.3 Waiting Room

None Present

1.6 Traceability Matrix

The purpose of this matrix is to provide an easy reference on what has to be additionally modified if a certain component is changed. Every time a component is changed, then the items in the column of that component which cross into an "X" should be modified as well.

NOTE: Traceability Matrix of a subset of the components is developed to keep the matrix fit in one page. The references between the other items would be documented in a similar manner. Building a tool to automatically generate the graphical representation of the matrix by scanning the labels and references can be a future work.

	T1	T2	T3	A1	A2	A3	GD1	GD2	GD3	GD4	GD5	DD6	DD7	DD8	DD9
GD1															
GD2				X	X		X								
GD3															
GD4	X							X							
GD5		X													
GD6		X													
DD1															
DD2															
DD3															
DD4															
DD5															
DD6								X	X						
DD7									X						
DD8									X		X				
DD9									X		X				
DD10									X						
DD11												X			
DD12									X						
DD13								X	X						
DD14															
DD15															
DD16															
DD17															
Dd18													X		X
DD19													X	X	
DD20															
DD21															
DD22															
DD23						X			X						
DD24															
DD25															
DD26															
DD27									X						
DD28															X
IM1										X					
IM2										X					
IM3										X					
IM5			X												

Table 1.5: Traceability Matrix Showing the Connections Between Items of Different Sections

1.7 Auxillary Constants

TB1 Metal Water Reaction

Constant	Value	Constraint	Units
A	6.48×10^{-8}	$T_2 \leq 1580^oC$	-
	1.0×10^{-6}	$T_2 > 1580^oC$	-
B/R	13586.0	$T_2 \leq 1580^oC$	-
	16014.0	$T_2 > 1580^oC$	-
q_r	6500.0	-	$\frac{\text{kJ}}{\text{kg}}$

TB2 Clad Conductivity

Constant	Value	Constraint	Units
a	1.43×10^{-5}	$T_2 \leq 1000^oC$	-
	2.73×10^{-5}	$T_2 > 1000^oC$	-
b	1.17×10^{-2}	$T_2 \leq 1000^oC$	-
	-1.27×10^{-3}	$T_2 > 1000^oC$	-

TB3 Coefficients for polynomial $K_{\mathbf{AV}}$

Coefficient	Value	Constraint	Units
x_0			
x_1			

TB4 Coefficients for polynomial $c_{p,1}$

Coefficient	Value	Constraint	Units
y_0			
y_1			
y_2			

TB5 Fuel Stored Energy

Constant	Value	Constraint	Units
K_0	15.496	-	-
K_1	19.145	-	$\frac{\text{cal}}{\text{moleK}}$
K_2	7.84733×10^{-4}	-	$\frac{\text{cal}}{\text{moleK}^2}$
K_3	5.64373×10^6	-	$\frac{\text{cal}}{\text{mole}}$
θ	535.285	-	K
E_D/R	1.8971×10^4	-	K

Chapter 2

LP Manual

2.1 Overview

Given relative fuel power (q'_{NFRAC}) as input, the function *fuel_temp_* calculates the following:

1. Average fuel temperature (T_1),
2. Average Clad temperature (T_2),
3. Centerline temperature (T_{CL}),
4. Surface temperature (T_S), and
5. Stored fuel energy ($\Delta H(T_{\text{abs}})$).
6. Integrated fuel power (f_p).
7. Power to the coolant (q'_{out}).
8. Metal water reaction heat (q'_{MWR}).

This function uses

- the material properties of the clad and fuelpellet
- lumped parameter methods and
- initial conditions

It solves the ODEs from the SRS given by IM1, IM3 and IM2 with the initial conditions defined in IM4, which are summarized below:

$$C_1 \frac{dT_1}{dt} = q'_N - \frac{T_1 - T_2}{R_1} \tag{2.1}$$

$$C_2 \frac{dT_2}{dt} = \frac{T_1 - T_2}{R_1} + q'_{\text{MWR}} - \frac{T_2 - T_B}{R_2} \tag{2.2}$$

$$C_{\text{CL}} \frac{dT_{\text{CL}}}{dt} = q'_N - \frac{T_{\text{CL}} - T_1}{R_{\text{CL}}} \tag{2.3}$$

where

- T_B is the coolant temperature
- q'_N is the linear element power (kW/m)
- q'_{MWR} is metal-water reaction heat (kW/m)
- C_1 is the thermal capacitance of the fuel (kWs/(m°C))
- C_2 is the thermal capacitance of the clad (kWs/(m°C))
- C_{CL} is the thermal capacitance at the centerline (kWs/(m°C))
- R_1 is the effective thermal resistance between T_1 and T_2 (m°C/kW)
- R_2 is the effective thermal resistance between coolant and half of the clad (m°C/kW)
- R_{CL} is the thermal resistance between T_{CL} and T_1 (m°C/kW)
- t is time

2.2 Numerical Algorithm

Equations 2.1–2.3 are of the form:

$$\frac{dx}{dt} = a(x)x + b(x)u(x,t), \tag{2.4}$$

where x is the variable under consideration. Taking the Laplace transform of Equation 2.4,

$$x(s) = \frac{x_0}{s-a} + \frac{bu}{s(s-a)} \tag{2.5}$$

with x_0 being $x(t_0)$. To obtain the closed form solution, $a(x)$, $b(x)$ and $u(x,t)$ are assumed to be constant in the interval $[t,\, t+\Delta t]$. The solution to the above ODE is found by taking the inverse Laplace transform of Equation 2.5

$$x(t) = x_o e^{-at} + bu \int_0^t e^{-at} dt \tag{2.6}$$

Denoting an approximation of $x(t_k)$ at t_k by x_k and denoting $\Delta t = t_{k+1} - t_k$, $k \geq 0$, we have:

$$x_{k+1} = x_k e^{-a_k \Delta t} + (1 - e^{-a_k \Delta t}) \frac{-b(x_k)u(x_k, t_k)}{a(x_k)} \tag{2.7}$$

The following table summarizes the values of $a(x)$, $b(x)$ and $u(x,t)$ for Equations 2.1, 2.2 and 2.3:

2.3 Algorithm

fuel_temp_($\Delta t, q_{\text{NFRAC},k}, q'_{N_{\max}}, r_f, f, \rho_1, \rho_2, h_{ib}, h_p, \tau_g, \tau_c, T_b, p_{\text{dry}}, h_{\text{dry}}$, time, $init\_flag, MW\_flag, n$, $h_{b,k}, q'_{N,k}, k_{c,k}, k_{\text{AV},k}, q'_{\text{MWR},k}, f_{p,k}, T_{1,k}, T_{2,k}, T_{\text{CL},k}, T_{S,k}, h_{c,k}, h_{g,k}, C_{1,k}, C_{2,k}, C_{\text{CL},k}, c_{p,1,k}, c_{p,2,k}, c_{p,3,k}, \Delta H(T_{\text{abs},k})$, $\delta_{\text{ox},k}, R_{\text{ox},k}, q'_{\text{out},k}, q'_{\text{MWRI},k}$)

1. Initialization section ($*init\_flag == 1$):

   - Input: Δt, $q_{\text{NFRAC},0}$, $q'_{N_{\max}}$,r_f, f, ρ_1, ρ_2, h_{ib}, h_p, τ_g, τ_c, T_b, *init_flag*.
   - At t_0 compute y_0
   - Output: y_0,

   where $y_0 = \{h_b, q'_N, k_c, k_{\text{AV}}, q'_{\text{MWR}}, f_p, T_1, T_2, T_{\text{CL}}, T_S, C_1, C_2, C_{\text{CL}}, c_{p,1}, c_{p,2}, c_{p,3}, h_c, h_g,$ $\Delta H(T_{\text{abs}}), \delta_{\text{ox}}, R_{\text{ox}}, q'_{\text{out}}, q'_{\text{MWRI}}\}$.
   All elements of the set y_0 are evaluated at the 0^{th} time step.

2. Dynamic section ($*init\_flag == 0$):
   At t_{k+1}, $k \geq 0$,

   - Input:Δt, $q_{\text{NFRAC},k+1}$, $q'_{N_{\max}}$,r_f, f, ρ_1, ρ_2, h_{ib}, h_p, τ_g, τ_c, T_b, p_{dry}, h_{dry}, time, *init_flag*, *MW_flag*, n, y_k.
   - compute y_{k+1}, update n when necessary.
   - Output: y_{k+1}, n,

   where $y_{k+1} = \{h_b, q'_N, k_c, k_{\text{AV}}, q'_{\text{MWR}}, f_p, T_1, T_2, T_{\text{CL}}, T_S, C_1, C_2, C_{\text{CL}}, c_{p,1}, c_{p,2}, c_{p,3}, h_c, h_g,$ $\Delta H(T_{\text{abs}}), \delta_{\text{ox}}, R_{\text{ox}}, q'_{\text{out}}, q'_{\text{MWRI}}\}$.
   All elements of the set y_{k+1} are evaluated at the $k+1^{\text{th}}$ time step.

2.4 Overall function

```
2  ⟨fuel temp function 2⟩ ≡
     void calpro_(float *t, int *i, int *iflag, float *prpval, int *icnt);
     void dryout_(float *htout, float *time);
     void fuel_temp_(const float *delta, float *q_NFRAC, float *q_Nmax, float *r_f, float *f, float
             *rho_1, float *rho_2, float *h_ib, float *h_p, float *tau_g, float *tau_c, float *t_b, float
             *p_dry, float *h_dry, float *time, short int *init_flag, int *MW_flag, int *n, float
             *h_b, float *q_N, float *k_c, float *k_AV, float *q_MWR, float *f_p, float *t_1, float
             *t_2, float *t_CL, float *t_S, float *h_c, float *h_g, float *c_1, float *c_2, float *c_CL, float
             *c_p1, float *c_p2, float *c_p3, float *deltaHT_abs, float *delta_ox, float *rate_ox, float
             *q_out, float *q_MWRI)
     {
       if (*init_flag) { ⟨initialization section 15⟩}
       else { ⟨dynamic section 53⟩}
     }
```
This code is used in chunk 94

The following sections show the connections between the theory and the numerical algorithm to the implementation.

2.5 Naming Conventions

Input Parameters

```
3    void fuel_temp_(const float *delta, float *q_NFRAC, float *q_Nmax, float *r_f, float *f, float *rho_1, float
         *rho_2, float *h_ib, float *h_p, float *tau_g, float *tau_c, float *t_b, float *p_dry, float *h_dry, float
         *time, short int *init_flag, int *MW_flag, int *n, float *h_b, float *q_N, float *k_c, float *k_AV, float
         *q_MWR, float *f_p, float *t_1, float *t_2, float *t_CL, float *t_S, float *h_c, float *h_g, float
         *c_1, float *c_2, float *c_CL, float *c_p1, float *c_p2, float *c_p3, float *deltaHT_abs, float
         *delta_ox, float *rate_ox, float *q_out, float *q_MWRI)
```

On input,

parameter	stores
*delta	Δt
*q_NFRAC	q'_{NFRAC}
*q_Nmax	$q'_{N_{\max}}$
*r_f	r_f
*f	f
*rho_1	ρ_1
*rho_2	ρ_2
*h_ib	h_{ib}
*h_p	h_p
*tau_g	τ_g
*tau_c	τ_c
*t_b	T_B
*p_dry	p_{dry}
*h_dry	h_{dry}
*time	time
*init_flag	0 or 1
*MW_flag	0 or 1
*n	1 or 2
*h_b	h_{ib} or h_{dry}
*q_N	$q'_{N,k}$, $k \geq 0$, if $\neg *init\_flag$
*k_c	$k_{c,k}$, $k \geq 0$, if $\neg *init\_flag$
*k_AV	$k_{\mathrm{AV,k}}$, $k \geq 0$, if $\neg *init\_flag$
*q_MWR	$q'_{\mathrm{MWR},k}$, $k \geq 0$, if $\neg *init\_flag$
*f_p	$P_{\mathrm{F,SUM},k}$ $k \geq 0$, if $\neg *init\_flag$
*t_1	$T_{1,k}$, $k \geq 0$, if $\neg *init\_flag$
*t_2	$T_{2,k}$, $k \geq 0$, if $\neg *init\_flag$

parameter	stores
$*t\_CL$	$T_{\mathrm{CL},k}$, $k \geq 0$, if $\neg * init\_flag$
$*t\_S$	$T_{S,k}$, $k \geq 0$, if $\neg * init\_flag$
$*h\_c$	$h_{c,k}$, $k \geq 0$, if $\neg * init\_flag$
$*h\_g$	$h_{g,k}$, $k \geq 0$, if $\neg * init\_flag$
$*c\_1$	$C_{1,k}$, $k \geq 0$, if $\neg * init\_flag$
$*c\_2$	$C_{2,k}$, $k \geq 0$, if $\neg * init\_flag$
$*c\_CL$	$C_{\mathrm{CL},k}$, $k \geq 0$, if $\neg * init\_flag$
$*c\_p1$	$c_{p,1,k}$, $k \geq 0$, if $\neg * init\_flag$
$*c\_p2$	$c_{p,2,k}$, $k \geq 0$, if $\neg * init\_flag$
$*c\_p3$	$c_{p,3,k}$, $k \geq 0$, if $\neg * init\_flag$
$*deltaHT\_abs$	$\Delta H(T_{\mathrm{abs},k})$, $k \geq 0$, if $\neg * init\_flag$
$*delta\_ox$	$\delta_{\mathrm{ox},k}$, $k \geq 0$, if $\neg * init\_flag$
$*rate\_ox$	$R_{\mathrm{ox},k}$ $k \geq 0$, if $\neg * init\_flag$
$*q\_out$	$q'_{\mathrm{out},k}$ $k \geq 0$, if $\neg * init\_flag$
$*q\_MWRI$	$q'_{\mathrm{MWRI},k}$, $k \geq 0$, if $\neg * init\_flag$

For $*init\_flag = 1$, that is, when time step $k = 0$, all the input variables with subscript k can have any value, as they are not used in any calculations during the initialization.

Output Parameters from the Initialization section

```
5 void fuel_temp_(const float *delta, float *q_NFRAC, float *q_Nmax, float *r_f, float *f, float *rho_1, float
      *rho_2, float *h_ib, float *h_p, float *tau_g, float *tau_c, float *t_b, float *p_dry, float *h_dry, float
      *time, short int *init_flag, int *MW_flag, int *n, float *h_b, float *q_N, float *k_c, float *k_AV, float
      *q_MWR, float *f_p, float *t_1, float *t_2, float *t_CL, float *t_S, float *h_c, float *h_g, float
      *c_1, float *c_2, float *c_CL, float *c_p1, float *c_p2, float *c_p3, float *deltaHT_abs, float
      *delta_ox, float *rate_ox, float *q_out, float *q_MWRI)
```

On output,
if *$*$init_flag* $\equiv 1$,

parameter	stores
$*$n	1
$*$h_b	h_{ib}
$*$q_N	$q'_{N,0}$
$*$k_c	$k_{c,0}$
$*$k_AV	$k_{\mathrm{AV},0}$
$*$q_MWR	$q'_{\mathrm{MWR},0}$
$*$f_p	$P_{\mathrm{F,SUM},0}$
$*$t_1	$T_{1,0}$
$*$t_2	$T_{2,0}$
$*$t_CL	$T_{\mathrm{CL},0}$
$*$t_S	$T_{S,0}$
$*$h_c	$h_{c,0}$
$*$h_g	$h_{g,0}$
$*$c_1	$C_{1,0}$
$*$c_2	$C_{2,0}$
$*$c_CL	$C_{\mathrm{CL},0}$
$*$c_p1	$c_{p,1,0}$
$*$c_p2	$c_{p,2,0}$
$*$c_p3	$c_{p,3,0}$
$*$deltaHT_abs	$\Delta H(T_{\mathrm{abs},0})$
$*$delta_ox	$\delta_{\mathrm{ox},0}$
$*$rate_ox	$R_{\mathrm{ox},0}$
$*$q_out	$q'_{\mathrm{out},0}$
$*$q_MWRI	$q'_{\mathrm{MWRI},0}$

Output Parameters from the Dynamic section

8

On output,

If $\neg\ast\mathit{init\_flag}$,

parameter	stores
$\ast n$	1 or 2
$\ast\mathit{h\_b}$	h_{ib} or h_{dry}
$\ast\mathit{q\_N}$	$q'_{\text{N},k+1}$
$\ast\mathit{k\_c}$	$k_{c,k+1}$
$\ast\mathit{k\_AV}$	$k_{\text{AV},k+1}$
$\ast\mathit{q\_MWR}$	$q'_{\text{MWR},k+1}$
$\ast\mathit{f\_p}$	$P_{\text{F,SUM},k+1}$
$\ast\mathit{t\_1}$	$T_{1,k+1}$
$\ast\mathit{t\_2}$	$T_{2,k+1}$
$\ast\mathit{t\_CL}$	$T_{\text{CL},k+1}$
$\ast\mathit{t\_S}$	$T_{S,k+1}$
$\ast\mathit{h\_c}$	$h_{c,k+1}$
$\ast\mathit{h\_g}$	$h_{g,k+1}$
$\ast\mathit{c\_1}$	$C_{1,k+1}$
$\ast\mathit{c\_2}$	$C_{2,k+1}$
$\ast\mathit{c\_CL}$	$C_{\text{CL},k+1}$
$\ast\mathit{c\_p1}$	$c_{p,1,k+1}$
$\ast\mathit{c\_p2}$	$c_{p,2,k+1}$
$\ast\mathit{c\_p3}$	$c_{p,3,k+1}$
$\ast\mathit{deltaHT\_abs}$	$\Delta H(T_{\text{abs},k+1})$
$\ast\mathit{delta\_ox}$	$\delta_{\text{ox},k+1}$
$\ast\mathit{rate\_ox}$	$R_{\text{ox},k+1}$
$\ast\mathit{q\_out}$	$q'_{\text{out},k+1}$
$\ast\mathit{q\_MWRI}$	$q'_{\text{MWRI},k+1}$

NOTE: The *fuel_temp_* function calls two fuelpin15.f functions- ‘*calpro_*’ which calculates material properties and ‘*dryout_*’ which outputs a message when dryout occurs. The interfaces for these functions are not specified in this document, but the relevant terms that they define are explained as they arise in the documentation.

Local Variables for the Effective thermal resistance in the Initialization section

11

parameter	stores
r_1	$R_{1,0}$
r_2	$R_{2,0}$
r_3	$R_{3,0}$
r_fuel	$R_{\mathrm{FUEL},0}$

Local Variables for the Effective thermal resistance in the Dynamic section

13

parameter	stores
r_1	$R_{1,k+1}$
r_2	$R_{2,k+1}$
r_3	$R_{3,k+1}$
r_CL	$R_{\mathrm{CL},k+1}$

2.6 Initialization section

In this section, we determine initial values (subscript $k = 0$) for:

$h_b, q'_N, k_c, k_{\mathrm{AV}}, q'_{\mathrm{MWR}}, f_p, T_1, T_2, T_{\mathrm{CL}}, T_S, h_c, h_g, C_1, C_2, C_{\mathrm{CL}}, c_{p,1}, c_{p,2}, c_{p,3}, \Delta H(T_{\mathrm{abs}}), \delta_{\mathrm{ox}}, R_{\mathrm{ox}}, q'_{\mathrm{out}}, q'_{\mathrm{MWRI}}$

```
15  ⟨ initialization section 15 ⟩ ≡
      $*n = 1$;      /* n is used to keep track of the dryout output message in the dynamic section */
      float $pi = 3.1416$;
       ⟨ Calculation of $q'_N$ 17 ⟩;
       ⟨ initialization of average clad temperature $T_{2,0}$ 18 ⟩;
       ⟨ Calculation of $k_c$ 19 ⟩;
       ⟨ Calculation of heat transfer coefficient ($h_c$) and the gap conductance ($h_g$) 21 ⟩;
       ⟨ initialization of surface temperature ($T_{S,0}$) 22 ⟩;
       ⟨ convergence routine to determine $k_{\mathrm{AV},0}$ and $T_{\mathrm{CL},0}$ 28 ⟩;
       ⟨ Calculation of $R_1$ 30 ⟩;
       ⟨ initialization of average fuel temperature $T_{1,0}$ 31 ⟩;
       ⟨ declaration of constants for stored energy 32 ⟩;
       ⟨ $\Delta H(T_{\mathrm{abs}})$ 33 ⟩;
       ⟨ Calpro function for $C_1$ and $c_{p,1}$ 35 ⟩;
       ⟨ Calculation of $C_1$ and $c_{p,1}$ 36 ⟩;
      $icnt = 10$;
          /* icnt is given as an argument to the calpro() function for calculating the specific heats and the
            integrals of polynomials */
       ⟨ Calpro function for $C_2$ and $c_{p,2}$ 38 ⟩;
       ⟨ Calculation of $C_2$ and $c_{p,2}$ 39 ⟩;
       ⟨ Calpro function for $C_{\mathrm{CL}}$ and $c_{p,3}$ 41 ⟩;
       ⟨ Calculation of $C_{\mathrm{CL}}$ and $c_{p,3}$ 42 ⟩;
       ⟨ initialization of constants for $R_{\mathrm{ox}}$ 44 ⟩;
       ⟨ Calculation of $R_{\mathrm{ox}}$ 45 ⟩;
       ⟨ Calculation of $q'_{\mathrm{MWR}}$ 46 ⟩;
       ⟨ initialization of $q'_{\mathrm{MWRI}}$ 47 ⟩;
       ⟨ Calculation of $\delta_{\mathrm{ox}}$ 48 ⟩;
       ⟨ Calculation of $q'_{\mathrm{out}}$ 50 ⟩;
       ⟨ initialization of $f_{p,0}$ 51 ⟩;
```

This code is used in chunk 2

2.6.1 Computing q'_N, T_2 and k_c

The input relative fuel power (q'_{NFRAC}) is changed to linear element power (q'_N) by multiplying it with the initial linear element rating ($q'_{\mathrm{N_{max}}}$) as given by DD25 of the SRS.

$$q'_N = q'_{\mathrm{NFRAC}} q'_{N_{\max}}; \tag{2.8}$$

This q'_N is used to determine the relevant temperatures for the fuelpin. We evaluate linear element power as

```
17  ⟨ Calculation of $q'_N$ 17 ⟩ ≡
      $*q\_N = *q\_NFRAC * (*q\_Nmax)$;
```

This code is used in chunks 15 and 57

Now, we evaluate T_2 in steady state by first setting the time derivative term of Equation 2.1 to zero as follows,

$$\frac{T_1 - T_2}{R_1} = q'_N \tag{2.9}$$

Next we set the time derivative term of Equation 2.2 to zero and neglect the metal water heating term to get,

$$\frac{T_1 - T_2}{R_1} = \frac{T_2 - T_B}{R_2} \tag{2.10}$$

Substituting Equation 2.9 in Equation 2.10 and rearranging the equation, we get the steady state case as:

$$T_2 = T_B + q'_N R_2, \tag{2.11}$$

where R_2 is given by DD12 of the SRS as,

$$R_2 = \frac{1}{2\pi r_c h_c} \tag{2.12}$$

From DD18 of the SRS, we have the equation for h_c as,

$$h_c = \frac{2k_c h_b}{2k_c + \tau_c h_b} \tag{2.13}$$

Substituting Equation 2.13 into Equation 2.12, we get,

$$R_2 = \frac{1}{2\pi r_c \left(\frac{2k_c h_b}{2k_c + \tau_c h_b}\right)} \tag{2.14}$$

$$= \frac{1}{2\pi r_c}\left(\frac{2k_c + \tau_c h_b}{2k_c h_b}\right) \tag{2.15}$$

The above equation cannot be evaluated directly in steady state, because R_2 is dependent on T_2 through the clad conductivity (k_c) as given by DD15 of SRS. That is,

$$k_c = aT_2 + b, \tag{2.16}$$

where a and b are given constants obtained by a least squares fit to tabulated data. According to the Assumption A12 of the SRS, since T_2 is less than 1000^oC in the initial state, the values of a and b are given by the Table TB2 of the SRS as,

$$a = 1.43 \times 10^{-5} \tag{2.17}$$

$$b = 1.17 \times 10^{-2} \tag{2.18}$$

So, taking the expression for k_c from Equation 2.16, substituting it into Equation 2.15 gives

$$R_2 = \frac{2(aT_2 + b) + \tau_c h_b}{4\pi r_c h_b (aT_2 + b)}, \tag{2.19}$$

On further simplification, Equation 2.19 becomes,

$$R_2 = \frac{2aT_2 + 2b + \tau_c h_b}{4\pi r_c h_b a T_2 + 4\pi r_c h_b b}, \tag{2.20}$$

where r_c is the outer clad radius and is obtained by the sum of fuel radius (r_f), gap thickness (τ_g) and clad thickness (τ_c).

$$r_c = r_f + \tau_g + \tau_c \tag{2.21}$$

Substituting Equation 2.20 into Equation 2.11 and rearranging gives an equation quadratic in T_2:

$$4\pi r_c h_b a T_2^2 + \left(4\pi r_c h_b b - 4\pi r_c h_b a T_B - 2aq'_N\right)T_2 - (4\pi r_c h_b T_B b + 2q'_N b + q'_N h_b \tau_c) = 0 \tag{2.22}$$

The above equation has to be solved to find the positive root which gives T_2 in steady state. Simultaneously the value k_c from Equation 2.16 is also calculated.

18 ⟨initialization of average clad temperature $T_{2,0}$ 18⟩ ≡ /* declaration of constants */

```
  float a = 1.43 · 10^-05;
  float b = 1.17 · 10^-02;
      /* computation of clad radius 2.21 */
  float r_c = *r_f + *tau_g + *tau_c;
      /* initializing coolant film conductance */
  *h_b = *h_ib;
      /* computation of T2 in steady state */
  float C10 = 2.0 * pi * r_c * (*h_b);
  float C11 = 2.0 * C10 * a;
  float C12 = C10 * (2.0 * b − (2.0 * a * (*t_b))) − (*q_N * 2.0 * a);
  float C13 = −C10 * (*t_b) * 2.0 * b − *q_N * (2.0 * b + ((*h_b) * (*tau_c)));
      /* solving quadratic equation */
  *t_2 = (−C12 + sqrt(C12 * C12 − 4.0 * C11 * C13))/(2.0 * C11);
      /* computation of initial clad conductivity 2.16 */
```

This code is used in chunk 15

19 ⟨Calculation of k_c 19⟩ ≡

```
  *k_c = a * (*t_2) + b;
```

This code is used in chunk 15

2.6.2 Computing h_c, h_g and T_S

Using this clad conductivity (k_c), we compute the heat transfer coefficient (h_c) and the gap conductance (h_g) as DD18 and DD19 of the SRS, respectively. That is,

$$h_c = \frac{2k_c h_b}{2k_c + \tau_c h_b}, \tag{2.23}$$

$$h_g = \frac{2k_c h_p}{2k_c + \tau_c h_p} \tag{2.24}$$

21 ⟨Calculation of heat transfer coefficient (h_c) and the gap conductance (h_g) 21⟩ ≡

```
      /* calculation of heat transfer coefficient */
  *h_c = (2 * (*k_c) * (*h_b))/((2 * (*k_c)) + (*tau_c * (*h_b)));      /* calculation of gap conductance */
  *h_g = (2 * (*k_c) * (*h_p))/((2 * (*k_c)) + (*tau_c * (*h_p)));
```

This code is used in chunks 15 and 60

At each time step, the surface temperature (T_S) is calculated based on the clad and average fuel temperatures as given by DD23 of the SRS as:

$$T_S = T_2 + \frac{T_1 - T_2}{R_1} R_3, \tag{2.25}$$

where R_3 is calculated as given by DD10 of the SRS.

$$R_3 = \frac{1}{2\pi r_f h_g} \tag{2.26}$$

The surface temperature in steady state ($T_{S,0}$) is evaluated using $T_{2,0}$ and by setting Equation 2.1 to zero as shown in Equation 2.9. Substituting Equation 2.9 in Equation 2.25 gives the steady state case of T_S as:

$$T_{S,0} = T_{2,0} + q'_{N,0} R_{3,0}, \tag{2.27}$$

where $R_{3,0}$ is calculated based on $h_{g,0}$.

```
22  ⟨initialization of surface temperature (T_{S,0}) 22⟩ ≡        /* calculation of R_3 */
      float r_3 = 1/(2 * pi * (*r_f) * (*h_g));
          /* calculation of T_{S,0} */
      *t_S = *t_2 + (*q_N * r_3);
```

This code is used in chunk 15

2.6.3 Computing T_{CL} and k_{AV}

Given this T_S and q'_N, the centerline temperature (T_{CL}) is calculated as given by Equation 1.50 of the SRS. That is, in steady state,

$$T_{\mathrm{CL}} = T_S + R_{\mathrm{FUEL}} q'_N, \tag{2.28}$$

where R_{FUEL} is given by DD6 of the SRS as,

$$R_{\mathrm{FUEL}} = \frac{f}{4\pi k_{\mathrm{AV}}}, \tag{2.29}$$

where f is the flux depression factor (constant obtained from the input file) and k_{AV} is the average fuel conductivity.

```
24  ⟨computation of T_CL 24⟩ ≡
      float r_fuel;
      r_fuel = *f/(4.0 * pi * (*k_AV));
      *t_CL = *t_S + (r_fuel * (*q_N));
```

This code is used in chunk 28

The above computation requires the average thermal conductivity (k_{AV}), but this value is not initially known.. Since k_{AV} is a temperature-dependent, non-linear variable, an iterative procedure converging on mutually consistent values for k_{AV} and T_{CL} is needed. An initial estimate of k_{AV} ($k_{\text{AV,est}}$) fixes the $T_{\text{CL,est}}$. To update k_{AV}, we need an estimate of linear element power ($q'_{\text{N,est}}$) which is computed from the current $T_{\text{CL,est}}$. Rewriting Equation 1.45 of the SRS in terms of q'_N by using DD1 of the SRS and taking flux depression factor into consideration gives,

$$\int_{T_{\text{CL}}}^{T_S} dT = \frac{-fq'_N}{2\pi r_f^2} \int_0^{r_f} \frac{r}{k} dr \tag{2.30}$$

Integrating the RHS we have,

$$\int_{T_{\text{CL}}}^{T_S} dT = \frac{-fq'_N}{4\pi k} \tag{2.31}$$

Rearranging Equation 2.31 and integrating the LHS of the equation from T_S to $T_{\text{CL,est}}$ generates the estimate of linear element power ($q'_{\text{N,est}}$) as,

$$q'_{\text{N,est}} = 4\pi \int_{T_S}^{T_{\text{CL,est}}} \frac{kdT}{f} \tag{2.32}$$

where k is a first order polynomial function of temperature and is given by DDL-pkav of the SRS as,

$$k = x_1 T + x_0 \tag{2.33}$$

Let

$$K(T) = \int kdT, \tag{2.34}$$

Hence,

$$\begin{aligned} q'_{\text{N,est}} &= \frac{4\pi}{f} [K(T)]_{T_S}^{T_{\text{CL,est}}}, &(2.35)\\ &= \frac{4\pi}{f} \big(K(T_{\text{CL,est}}) - K(T_S)\big) &(2.36) \end{aligned}$$

25 ⟨estimation of q'_N 25⟩ ≡

```
float q_NEST;
q_NEST = ((4.0 * pi) * (t_e − t_a))/(*f);
    /* t_e and t_a are K(T_CL,est) and K(T_S) respectively which are evaluated by calpro function */
```

This code is used in chunk 28

Substituting Equation 2.29 in Equation 2.28 and rearranging gives,

$$k_{\text{AV}} = \frac{fq'_{\text{N}}}{4\pi(T_{\text{CL}} - T_S)} \tag{2.37}$$

The estimate of the element power from Equation 2.36 is compared to the actual value and used to update k_{AV}. The relationship between k_{AV} and and q'_N is given by the first order Taylor series expansion of k_{AV} with respect to q'_N as,

$$k_{\text{AV,i+1}} = k_{\text{AV,i}} + \frac{dk_{\text{AV}}}{dq'_{\text{N}}} \Delta q'_{\text{N}} \tag{2.38}$$

Differentiating Equation 2.37 with respect to q'_{N} gives,

$$\frac{dk_{\text{AV}}}{dq'_{\text{N}}} = \frac{f}{4\pi(T_{\text{CL}} - T_S)} \tag{2.39}$$

The change in q'_N ($\Delta q'_N$) is the difference between the estimated and actual values.

$$\Delta q'_{\text{N}} = q'_{\text{N,est,i}} - q'_N \tag{2.40}$$

Substituting Equation 2.39 and Equation 2.40 in Equation 2.38,

$$k_{\text{AV,i+1}} = k_{\text{AV,i}} + \frac{(fq'_{\text{N,est,i}} - fq'_N)}{4\pi(T_{\text{CL}} - T_S)} \tag{2.41}$$

26 $\langle$ update k_{AV} 26 $\rangle \equiv$

```
*k_AV = *k_AV + (((*f * q_NEST - (*f * (*q_N))))/(4.0 * pi * (*t_CL - *t_S)));
```

This code is used in chunk 28

We compute the relative error (normalized difference between the actual and estimated values) of q'_N as a condition for convergence.

27 $\langle$ relative error computation 27 $\rangle \equiv$

```
re = (*q_N - q_NEST)/(*q_N);
```

This code is used in chunk 28

Now we can put the above together to evaluate $k_{\text{AV},0}$ and $T_{\text{CL},0}$ using the described convergence routine.

28 $\langle$ convergence routine to determine $k_{\text{AV},0}$ and $T_{\text{CL},0}$ 28 $\rangle \equiv$

```
float t_a, t_e;
int i, iflag;
int icnt;     /* icnt is given as an argument to the calpro() function for calculating the specific heats
       and the integrals of polynomials */
i = 1;
iflag = 1;
icnt = 0;
float ts = *t_S;
calpro_(&ts, &i, &iflag, &t_a, &icnt);
    /* function calpro evaluates t_a which is the integral of polynomial for k_AV at T_S (K(T_S)) */
int idnt = 4;
    /* initial estimate of k_AV */
*k_AV = 0.00255;
    /* initial estimate of relative error for convergence */
float re;
do {
    ⟨computation of T_CL 24⟩;
        /* function calpro evaluates the integral of polynomial for k (t_e) at T_CL */
  float tcl = *t_CL;
```

```
    calpro_(&tcl, &i, &iflag, &t_e, &idnt);
      /* function calpro evaluates $t_e$ which is the integral of polynomial for $k_{AV}$ at $T_{CL}$ ($K(T_{CL})$) */
    ⟨ estimation of $q'_N$ 25 ⟩;
    ⟨ relative error computation 27 ⟩;
    if (fabsf(re) ≤ 0.0004) break;
    ⟨ update $k_{AV}$ 26 ⟩;
  } while (1);
```

This code is used in chunk 15

2.6.4 Computing T_1

With k_{AV} determined from the above routine, we can determine the average fuel temperature (T_1) by setting the time derivative term of Equation 2.1 to zero. That is, in steady state,

$$T_1 = T_2 + q'_N R_1, \qquad (2.42)$$

where R_1 is the effective thermal resistance between T_1 and T_2. The value of R_1 is given by DD11 of the SRS as:

$$R_1 = \frac{f}{8\pi k_{\mathrm{AV}}} + \frac{1}{2\pi r_f h_g} \qquad (2.43)$$

30 ⟨ Calculation of R_1 30 ⟩ ≡

```
    float r_1 = (*f/(8 * pi * (*k_AV))) + (1/(2 * pi * (*r_f) * (*h_g)));
```

This code is used in chunks 15 and 62

31 ⟨ initialization of average fuel temperature $T_{1,0}$ 31 ⟩ ≡

```
    *t_1 = *t_2 + (*q_N * r_1);
```

This code is used in chunk 15

2.6.5 Computing $\Delta H(T_{\mathbf{abs}})$

Now we compute the stored fuel energy, which depends on the average fuel temperature. It is the change in fuel enthalpy from standard room temperature ($T_{std} = 298$K) to the absolute value of the average fuel temperature T_1 and is given by DD2 of the SRS as:

$$\Delta H(T_{\mathrm{abs}}) = K_0\left(K_1\theta\left(\left(e^{\theta/T_{\mathrm{abs}}}-1\right)^{-1} - \left(e^{\theta/T_{std}}-1\right)^{-1}\right) + K_2(T_{\mathrm{abs}}^2 - T_{std}^2) + K_3 e^{-E_D/(R_D T_{\mathrm{abs}})}\right), \qquad (2.44)$$

where $K_0, K_1, K_2, K_3, \theta, E_D, R_D$ are constants whose values are given by the TB5 of SRS as:

Constant	Value	Units
K_0	15.496	-
K_1	19.145	cal/moleK
K_2	7.84733×10^{-4}	cal/(moleK^2)
K_3	5.64373×10^{6}	cal/mole
θ	535.285	K
E_D	37.6946×10^{3}	
R_D	1.987	

```
32  ⟨ declaration of constants for stored energy 32 ⟩ ≡        /* declaration of constants */
      float K0 = 15.49 · 10^{-03};
      float K1 = 19.145;
      float K2 = 7.84733 · 10^{-04};
      float K3 = 5.64373 · 10^{06};
      float THETA = 535.285;
      float E_D = 37.6946 · 10^{03};
      float R_D = 1.987;
      float t_std = 298;
```

This code is used in chunks 15 and 75

Evaluation of the stored energy

```
33  ⟨ ΔH(T_abs) 33 ⟩ ≡
      float t_abs;
      t_abs = *t_1 + 273.0;
      *deltaHT_abs = K0 * (K1 * THETA * ((1/(exp(THETA/t_abs) − 1)) − (1/(exp(THETA/t_std) − 1))) + K2 *
          (t_abs * t_abs − t_std * t_std) + K3 * exp(−E_D/(R_D * t_abs)));
```

This code is used in chunks 15 and 75

2.6.6 Computing C_1, $c_{p,1}$

We initialize the thermal capacitances C_1, C_2, C_{CL} which will be used later in determining the transient temperatures in the dynamic section.
C_1 is the thermal capacitance of the fuel ($\frac{\mathrm{kWs}}{\mathrm{m}^o\mathrm{C}}$) and is given by DD14 of the SRS as,

$$C_1 = \pi r_f^2 \rho_1 c_{p,1}, \tag{2.45}$$

where
ρ_1 is the fuel density ($\frac{\mathrm{kJ}}{\mathrm{kg}^o\mathrm{C}}$)
r_f is the fuel radius (m)
$c_{p,1}$ is the specific heat corresponding to the fuel average temperature ($\frac{\mathrm{kJ}}{\mathrm{kg}^o\mathrm{C}}$)
$c_{p,1}$ is represented as a second order polynomial function of temperature and is given by DD17 of the SRS as,

$$c_{p,1} = y_2 T^2 + y_1 T + y_0 \tag{2.46}$$

The average value of $c_{p,1}$ is explicitly obtained by finding the average $c_{p,1}$ by integrating Equation 2.46 from T_S to $T_{\mathrm{CL,est}}$ and dividing by $T_{\mathrm{CL}} - T_S$. That is,

$$c_{p,1_{\mathrm{AV}}} = \frac{1}{T_{\mathrm{CL}} - T_S} \int_{T_S}^{T_{\mathrm{CL}}} c_{p,1} dT \tag{2.47}$$

Let

$$C_p(T) = \int c_{p,1} dT, \tag{2.48}$$

Hence,

$$c_{p,1_{\mathrm{AV}}} = \frac{1}{T_{\mathrm{CL}} - T_S} [C_p(T)]_{T_S}^{T_{\mathrm{CL}}}, \tag{2.49}$$

$$= \frac{\big(C_p(T_{\mathrm{CL}}) - C_p(T_S)\big)}{T_{\mathrm{CL}} - T_S} \tag{2.50}$$

35 ⟨ Calpro function for C_1 and $c_{p,1}$ 35 ⟩ ≡

```
float t_c, t_d;
    /* function calpro evaluates Cp(TS) */
int j = 2;
int jflag = 3;
ts = *t_S;
float tcl = *t_CL;
calpro_(&ts, &j, &jflag, &t_c, &idnt);
    /* function calpro evaluates Cp(TCL) */
calpro_(&tcl, &j, &jflag, &t_d, &idnt);
```

This code is used in chunk 15

36 ⟨ Calculation of C_1 and $c_{p,1}$ 36 ⟩ ≡ /* calculation of specific heat of the fuel */

```
*c_p1 = (t_d - t_c)/(*t_CL - *t_S);
    /* calculation of C1 */
*c_1 = pi * (*r_f) * (*r_f) * (*rho_1) * (*c_p1);
```

This code is used in chunks 15 and 78

2.6.7 Computing C_2, $c_{p,2}$

C_2 is the thermal capacitance of the clad $(\frac{\text{kWsec}}{\text{m}^{\text{o}}\text{C}})$ and is given by DD14 of SRS as,

$$C_2 = 2\pi r_c \tau_c \rho_2 c_{p,2}, \tag{2.51}$$

where r_c is the outer clad radius (m)
τ_c is the clad thickness (m)
$c_{p,2}$ is the specific heat corresponding to the clad temperature $(\frac{\text{kJ}}{\text{kg}^{\text{o}}\text{C}})$
ρ_2 is the clad density $(\frac{\text{kJ}}{\text{kg}^{\text{o}}\text{C}})$
We evaluate capacitance C_2 for T_2 as:

38 ⟨ Calpro function for C_2 and $c_{p,2}$ 38 ⟩ ≡

```
int k = 3;
int kflag = 2;
    /* function calpro evaluates the specific heat of the clad (cp,2) at T2 */
float t2 = *t_2;
float cp2;
calpro_(&t2, &k, &kflag, &cp2, &idnt);
```

This code is used in chunk 15

39 ⟨ Calculation of C_2 and $c_{p,2}$ 39 ⟩ ≡

```
*c_p2 = cp2;
    /* calculation of C2 */
*c_2 = 2 * pi * r_c * (*tau_c) * (*rho_2) * (*c_p2);
```

This code is used in chunks 15 and 79

2.6.8 Computing $C_{\mathbf{CL}}$, $c_{p,3}$

C_{CL} is the thermal capacitance at the centerline ($\frac{\text{kWs}}{\text{m}^{\text{o}}\text{C}}$) and is given by DD14 of SRS as,

$$C_{\text{CL}} = \pi r_f^2 c_{p,3} \rho_1, \tag{2.52}$$

where r_f is the fuel radius(m)
$c_{p,3}$ is the specific heat corresponding to the fuel centreline temperature ($\frac{\text{kJ}}{\text{kg}^{\text{o}}\text{C}}$).
ρ_1 is the fuel density ($\frac{\text{kJ}}{\text{kg}^{\text{o}}\text{C}}$).

We evaluate capacitance C_{CL} for T_{CL} as:

41 ⟨Calpro function for C_{CL} and $c_{p,3}$ 41⟩ ≡

```
    /* function calpro evaluates the specific heat at the centerline (c_p,3) at T_CL */
  int l = 2;
  int lflag = 2;
  tcl = *t_CL;
  float cp3;
  calpro_(&tcl, &l, &lflag, &cp3, &idnt);
```

This code is used in chunk 15

42 ⟨Calculation of C_{CL} and $c_{p,3}$ 42⟩ ≡

```
  *c_p3 = cp3;
    /* calculation of C_CL */
  *c_CL = pi * (*r_f) * (*r_f) * (*rho_1) * (*c_p3);
```

This code is used in chunks 15 and 80

2.6.9 Computing δ_{ox}, $R_{\mathbf{ox}}$ and $q'_{\mathbf{MWR}}$

The zircaloy clad material oxidizes exothermically when exposed to high temperature steam, resulting in additional heat input (q'_{MWR}) to the clad. The rate of oxidization (R_{ox}) depends on the average clad temperature (T_2) and the thickness of the reacted zircaloy (δ_{ox}) and is given by DD5 of the SRS as,

$$R_{\text{ox}} = \frac{A}{1.56\delta_{ox}} e^{\frac{-B}{R(T_2+273)}}, \tag{2.53}$$

where the values of constants A, B/R are given by Table TB1 of the SRS. According to the Assumption A12 of the SRS, since T_2 is less than 1000^oC in the initial state, the values of A and B/R are given as,

$$A = 6.48 \times 10^{-8} \tag{2.54}$$
$$B/R = 13586.0 \tag{2.55}$$

The thickness of the reacted zircaloy (δ_{ox}) is initialized to 1.0×10^{-6}.

44 ⟨initialization of constants for R_{ox} 44⟩ ≡

```
         /* initialization of δ_ox,0 */
  *delta_ox = 1.0 · 10^-06;
    /* initialization of constants A and B/R */
  float A = 6.48 · 10^-08;
  float BbyR = 13586.0;
```

This code is used in chunk 15

45 ⟨ Calculation of R_{ox} 45 ⟩ ≡

```
*rate_ox = (A/(1.56 * (*delta_ox))) * exp(-(BbyR)/(*t_2 + 273.0));
```

This code is used in chunks 15 and 86

Now using this R_{ox}, the metal water reaction heat (q'_{MWR}) can be calculated as given by DD5 of the SRS.

$$q'_{\mathrm{MWR}} = R_{\mathrm{ox}} 2\pi r_c \rho_2 q_r, \tag{2.56}$$

where q_r is the heat of reaction and its value (6500.0) is given by Table TB1 of the SRS. The integrated metal water heat (q'_{MWRI}) is initialized to zero.

46 ⟨ Calculation of q'_{MWR} 46 ⟩ ≡

```
float q_r = 6500.0;
*q_MWR = *rate_ox * 2 * pi * r_c * (*rho_2) * q_r;
```

This code is used in chunks 15 and 88

47 ⟨ initialization of q'_{MWRI} 47 ⟩ ≡

```
*q_MWRI = 0.0;
```

This code is used in chunk 15

As the reaction takes place, the clad material is oxidized and the thickness of the reacted zircaloy clad material is found by using Euler's method for solving an ODE.

$$\delta_{\mathrm{ox},i+1} = \delta_{\mathrm{ox},i} + \frac{d\delta_{\mathrm{ox}}}{dt}\Delta t \tag{2.57}$$

Since the derivative of oxidized material with respect to time is rate of oxidization, that is,

$$\frac{d\delta_{\mathrm{ox}}}{dt} = R_{\mathrm{ox}} \tag{2.58}$$

Substituting (2.58) in (2.57),

$$\delta_{\mathrm{ox},i+1} = \delta_{\mathrm{ox},i} + R_{\mathrm{ox}}\Delta t \tag{2.59}$$

48 ⟨ Calculation of δ_{ox} 48 ⟩ ≡

```
*delta_ox = *delta_ox + *rate_ox * (*delta);
```

This code is used in chunks 15 and 90

2.6.10 Computing q'_{out} and initializing f_p

The output heat from the reaction is sent into the coolant. This heat to the coolant which is given by DD27 of the SRS is normalized by $q'_{N_{\max}}$ for easier understanding and comparission purposes. In other words the normalization is done since this is a standard form for presenting this information. Hence, the heat out is given as,

$$q'_{\mathrm{out}} = \frac{1}{q'_{N_{\max}}}\Big(\frac{T_2 - T_B}{R_2}\Big), \tag{2.60}$$

where R_2 is the effective resistance between coolant film and the clad and is given by DD12 of the SRS as,

$$R_2 = \frac{1}{2\pi r_c h_c} \tag{2.61}$$

50 ⟨ Calculation of q'_{out} 50 ⟩ ≡

```
float r_2 = 1/(2 * pi * r_c * (*h_c));
*q_out = (*t_2 - *t_b)/(r_2 * (*q_Nmax));
```

This code is used in chunks 15 and 84

The Integrated fuel power (f_p) as given by DD3 of the SRS, is a summation of the fuel powers at each time step. At t_0, no reaction takes place and hence no fuel power is generated. So, initially the integrated fuel power is set to zero.

```
51  ⟨ initialization of $f_{p,0}$ 51 ⟩ ≡
      *f_p = 0.0;
    This code is used in chunk 15
```

2.7 Dynamic section

In this section, we determine transient values (subscript $k > 0$) for

$$q'_N, k_c, k_{\text{AV}}, q'_{\text{MWR}}, f_p, T_1, T_2, T_{\text{CL}}, T_S, h_c, h_g, C_1, C_2, C_{\text{CL}}, c_{p,1}, c_{p,2}, c_{p,3}, \Delta H(T_{\text{abs}}), \delta_{\text{ox}}, R_{\text{ox}}, q'_{\text{out}}, q'_{\text{MWRI}}$$

```
53  ⟨ dynamic section 53 ⟩ ≡
      float pi = 3.1416;
      int icnt = 10;     /* icnt is given as an argument to the calpro() function for calculating the specific
            heats and the integrals of polynomials */
       ⟨ Check for dryout 55 ⟩;
       ⟨ Computing $q'_{N,k+1}$ 57 ⟩;
       ⟨ Computing $k_{c,k+1}$ 58 ⟩;
       ⟨ Computing $h_{c,k+1}$ and $h_{g,k+1}$ 60 ⟩;
       ⟨ Computing $R_{1,k+1}$ and $R_{2,k+1}$ 62 ⟩;
       ⟨ Computing exponential term $e^{\frac{-\Delta t}{R_{1,k+1}C_{1,k}}}$ for $T_1$ 67 ⟩;
       ⟨ Computing exponential term $e^{\frac{-\Delta t(R_{1,k+1}+R_{2,k+1})}{R_{1,k+1}R_{2,k+1}C_{2,k}}}$ for $T_2$ 64 ⟩;
       ⟨ Computing exponential term $e^{\frac{-\Delta t}{R_{\text{CL},k+1}C_{\text{CL},k}}}$ for $T_{\text{CL}}$ 70 ⟩;
       ⟨ Computing $T_{2,k+1}$ 65 ⟩;
       ⟨ Computing $T_{1,k+1}$ 68 ⟩;
       ⟨ Computing $T_{\text{CL},k+1}$ 71 ⟩;
       ⟨ Computing $T_{S,k+1}$ 73 ⟩;
       ⟨ Computing $\Delta H(T_{\text{abs},k+1})$ 75 ⟩;
       ⟨ Computing $P_{\text{F,SUM},k+1}$ 76 ⟩;
       ⟨ Computing $C_{1,k+1} = \pi r_f^2 \rho_1 c_{p,1,k+1}$ 78 ⟩;
       ⟨ Computing $C_{2,k+1} = 2\pi r_c \tau_c \rho_2 c_{p,2,k+1}$ 79 ⟩;
       ⟨ Computing $C_{\text{CL},k+1} = \pi r_f^2 \rho_1 c_{p,3,k+1}$ 80 ⟩;
       ⟨ Computing $k_{\text{AV},k+1}$ 82 ⟩;
      if (*MW_flag ≡ 1) {
          ⟨ Computing $q'_{\text{out},k+1}$ 84 ⟩;
          ⟨ Computing $R_{\text{ox},k+1}$ 86 ⟩;
          ⟨ Computing $q'_{\text{MWR},k+1}$ 88 ⟩;
          ⟨ Computing $\delta_{\text{ox},k+1}$ 90 ⟩;
          ⟨ Computing $q'_{\text{MWRI},k+1}$ 92 ⟩;
      }
    This code is used in chunk 2
```

2.7.1 Checking for Dryout

We check for dryout using the condition given in DD28 of the SRS. If the dryout occurs, we output a message notifying the time and heat out at which it occured and assign the heat transfer coefficient between the fuel surface and the coolant at dryout (h_{dry}) to the coolant film conductance (h_b).

55 ⟨ Check for dryout 55 ⟩ ≡ /∗ check for dryout ∗/

```
if (∗q_out ≥ ∗p_dry ∧ ∗n ≡ 1)
        {
    float qout, tym;
    qout = ∗q_out;
    tym = ∗time;
        /∗ calling fuelpin15.f subroutine ‘dryout_’ to write out fuel sheath dryout time and q'_out ∗/
    dryout_(&qout, &tym);
}
if (∗q_out ≥ ∗p_dry)
        {
    ∗n = 2;
        /∗ assigning dryout heat tranfer coefficient to the coolant conductance ∗/
    ∗h_b = ∗h_dry;
}
```

This code is used in chunk 53

2.7.2 Computing $q'_{N,k+1}$ and $k_{c,k+1}$

The transient linear element power and clad conductivity are determined in the same way as done in 2.6.1. At time t_{k+1}, the q'_N is calculated based on relative fuel power (q'_{NFRAC}) at t_{k+1} and is given by DD25 of the SRS as,

$$q'_{N,k+1} = q'_{\text{NFRAC},k+1} q'_{N_{\max}}; \tag{2.62}$$

We use the same chunk which calculates q'_N in the initialization section to compute $q'_{N,k+1}$, as the piece of code is same for both steady state and the transient state calculations.

57 ⟨ Computing $q'_{N,k+1}$ 57 ⟩ ≡

⟨ Calculation of q'_N 17 ⟩;

This code is used in chunk 53

The value of clad conductivity (k_c) at time t_{k+1} depends on the average clad temperature (T_2) at t_{k} and is given by DD15 of the SRS as,

$$k_{c,k+1} = aT_{2,k} + b, \tag{2.63}$$

where a and b are constants obtained by a least squares fit to tabulated data and are given by Table TB2 of the SRS, where Table TB2 used different values for a and b if the temperature is greater than 1000^oC. We evaluate value of k_c at t_{k+1} as,

58 ⟨ Computing $k_{c,k+1}$ 58 ⟩ ≡

```
float a, b;
```

```
if (*t_2 > 1000.0)
        {
    a = 2.727 · 10^-05;
    b = −1.2727 · 10^-03;
  }
  else
    {
    a = 1.43 · 10^-05;
    b = 1.17 · 10^-02;
  }
  *k_c = a * (*t_2) + b;
```

This code is used in chunk 53

2.7.3 Computing $h_{c,k+1}$ and $h_{g,k+1}$

Now using the $k_{c,k+1}$, we compute the heat transfer coefficient (h_c) and the gap conductance (h_g) at t_{k+1} in the same way we did in 2.6.2 as,

$$h_{c,k+1} = \frac{2k_{c,k+1}h_b}{2k_{c,k+1} + \tau_c h_b}. \tag{2.64}$$

$$h_{g,k+1} = \frac{2k_{c,k+1}h_p}{2k_{c,k+1} + \tau_c h_p}. \tag{2.65}$$

Hence, reusing the chunk that has calculated h_c and h_g in the initialization section, $h_{c,k+1}$ and $h_{g,k+1}$ are computed as,

```
60  ⟨Computing h_{c,k+1} and h_{g,k+1} 60⟩ ≡
      ⟨Calculation of heat transfer coefficient (h_c) and the gap conductance (h_g) 21⟩;
```

This code is used in chunk 53

2.7.4 Computing $R_{1,k+1}$ and $R_{2,k+1}$

R_1 at t_{k+1} is computed in the same way we did in 2.6.4 by taking the value of h_g at t_{k+1} and k_{AV} at t_k as,

$$R_{1,k+1} = \frac{f}{8\pi k_{\mathrm{AV},k}} + \frac{1}{2\pi r_f h_{g,k+1}} \tag{2.66}$$

So, for computing $R_{1,k+1}$, we reuse the same piece of code that computes R_1 at steady state.
R_2 at t_{k+1} is computed taking the value of $h_{c,k+1}$ and is given by DD12 of SRS as,

$$R_{2,k+1} = \frac{1}{2\pi r_c h_{c,k+1}}. \tag{2.67}$$

```
62  ⟨Computing R_{1,k+1} and R_{2,k+1} 62⟩ ≡
      ⟨Calculation of R_1 30⟩;      /* computation of clad radius */
      float r_c = *r_f + (*tau_g) + (*tau_c);
      float r_2 = 1.0/(2.0 * pi * r_c * (*h_c));
```

This code is used in chunk 53

2.7.5 Computing $T_{2,k+1}$

We solve Equation 2.2 for $T_{2,k+1}$. The value of T_2 at time t_{k+1} is computed using $R_{1,k+1}$, $R_{2,k+1}$ and the values of C_2, T_1, T_2, q'_{MWR} at t_k. By taking C_2 of Equation 2.2 to the RHS and rearranging, it simplifies to,

$$\frac{dT_2}{dt} = -\frac{(R_1 + R_2)}{R_1 R_2 C_2} T_2 + \frac{T_1 R_2 + q'_{\mathrm{MWR}} R_1 R_2 + T_B R_1}{R_1 R_2 C_2} \tag{2.68}$$

Comparing (2.68) with (2.4) using the Table 2.2, the solution to Equation 2.2 is given as,

$$T_{2,k+1} = T_{2,k} e^{\frac{-\Delta t(R_{1,k+1}+R_{2,k+1})}{R_{1,k+1}R_{2,k+1}C_{2,k}}} + \left(1 - e^{\frac{-\Delta t(R_{1,k+1}+R_{2,k+1})}{R_{1,k+1}R_{2,k+1}C_{2,k}}}\right) \frac{T_{1,k} R_{2,k+1} + q'_{\mathrm{MWR},k} R_{1,k+1} R_{2,k+1} + T_B R_{1,k+1}}{(R_{1,k+1} + R_{2,k+1})} \tag{2.69}$$

64 ⟨Computing exponential term $e^{\frac{-\Delta t(R_{1,k+1}+R_{2,k+1})}{R_{1,k+1}R_{2,k+1}C_{2,k}}}$ for T_2 64⟩ ≡

```
float g = exp((−(∗delta) ∗ (r_1 + r_2))/(r_1 ∗ r_2 ∗ (∗c_2)));
```

This code is used in chunk 53

65 ⟨Computing $T_{2,k+1}$ 65⟩ ≡

```
∗t_2 = ∗t_2 ∗ g + ((1.0 − g) ∗ (((∗t_1 ∗ r_2) + (∗q_MWR ∗ r_1 ∗ r_2) + ((∗t_b) ∗ r_1))/(r_1 + r_2)));
```

This code is used in chunk 53

2.7.6 Computing $T_{1,k+1}$

We solve Equation 2.1 for $T_{1,k+1}$. The value of T_1 at time t_{k+1} is computed using $R_{1,k+1}$, $T_{2,k+1}$, $q'_{N,k+1}$ and the values of C_1, T_1 at t_k. By taking C_1 of Equation 2.1 to the RHS and rearranging, it simplifies to,

$$\frac{dT_1}{dt} = -\frac{1}{R_1 C_1} T_1 + \frac{T_2 + q'_N R_1}{R_1 C_1} \tag{2.70}$$

Comparing Equation 2.70 with Equation 2.4, using Table 2.2, the solution to Equation 2.1 is given as,

$$T_{1,k+1} = T_{1,k} e^{\frac{-\Delta t}{R_{1,k+1}C_{1,k}}} + \left(1 - e^{\frac{-\Delta t}{R_{1,k+1}C_{1,k}}}\right)(R_{1,k+1} q'_{N,k+1} + T_{2,k+1}) \tag{2.71}$$

67 ⟨Computing exponential term $e^{\frac{-\Delta t}{R_{1,k+1}C_{1,k}}}$ for T_1 67⟩ ≡

```
float j = exp(−(∗delta)/(r_1 ∗ (∗c_1)));
```

This code is used in chunk 53

68 ⟨Computing $T_{1,k+1}$ 68⟩ ≡

```
∗t_1 = j ∗ (∗t_1) + ((1.0 − j) ∗ (r_1 ∗ (∗q_N) + (∗t_2)));
```

This code is used in chunk 53

2.7.7 Computing $T_{\mathbf{CL},k+1}$

Now we solve Equation 2.3 for $T_{\mathrm{CL},k+1}$. The value of T_{CL} at time t_{k+1} is computed using $T_{1,k+1}$, $q'_{N,k+1}$ and the values of C_{CL}, T_{CL} and k_{AV} at t_k. By taking C_{CL} of Equation 2.3 to the RHS and rearranging, it simplifies to,

$$\frac{dT_{\mathrm{CL}}}{dt} = -\frac{1}{R_{\mathrm{CL}}C_{\mathrm{CL}}}T_{\mathrm{CL}} + \frac{T_1 + q'_N R_{\mathrm{CL}}}{R_{\mathrm{CL}}C_{\mathrm{CL}}}, \tag{2.72}$$

where R_{CL} is the one half of the fuel resistance ($\mathrm{R_{FUEL}}$) and is given by DD13 of the SRS as,

$$R_{\mathrm{CL},k+1} = \frac{f}{8\pi k_{\mathrm{AV},k}} \tag{2.73}$$

Comparing Equation 2.72 with Equation 2.4 and using the Table 2.2, the solution to Equation 2.3 is given as,

$$T_{\mathrm{CL},k+1} = T_{\mathrm{CL},k}e^{\frac{-\Delta t}{R_{\mathrm{CL},k+1}C_{\mathrm{CL},k}}} + \left(1 - e^{\frac{-\Delta t}{R_{\mathrm{CL},k+1}C_{\mathrm{CL},k}}}\right)(R_{\mathrm{CL},k+1}q'_{N,k+1} + T_{1,k+1}) \tag{2.74}$$

70 ⟨ Computing exponential term $e^{\frac{-\Delta t}{R_{\mathrm{CL},k+1}C_{\mathrm{CL},k}}}$ for T_{CL} 70 ⟩ ≡

```
float r_CL = *f/(8.0 * pi * (*k_AV));     /* calculation of R_CL,k+1 */
float m = exp(-(*delta)/(r_CL * (*c_CL)));     /* calculation of exponential term */
```

This code is used in chunk 53

71 ⟨ Computing $T_{\mathrm{CL},k+1}$ 71 ⟩ ≡

```
*t_CL = m * (*t_CL) + ((1.0 - m) * (r_CL * (*q_N) + (*t_1)));
```

This code is used in chunk 53

2.7.8 Computing $T_{S,k+1}$

The value of T_S at time t_{k+1} is calculated based on $T_{1,k+1}$, $T_{2,k+1}$ and is given by DD23 of the SRS as,

$$T_{S,k+1} = T_{2,k+1} + \frac{T_{1,k+1} - T_{2,k+1}}{R_{1,k+1}}R_{3,k+1}, \tag{2.75}$$

where $R_{3,k+1}$ is calculated as given by DD10 of the SRS as:

$$R_{3,k+1} = \frac{1}{2\pi r_f h_{g,k+1}} \tag{2.76}$$

73 ⟨ Computing $T_{S,k+1}$ 73 ⟩ ≡

```
float r_3 = 1/(2 * pi * (*r_f) * (*h_g));     /* calculation of gap resistance */
*t_S = *t_2 + ((*t_1 - *t_2)/(r_1) * (r_3));
```

This code is used in chunk 53

2.7.9 Computing $\Delta H(T_{\mathbf{abs}})$ and $P_{F,SUM}$

Now we compute the transient stored fuel energy in the same way we did in the initialization section (2.6.5). The stored fuel energy at time t_{k+1} depends on the value of absolute value of T_1 at t_{k+1} and is given by DD2 of the SRS as:

$$\Delta H(T_{\text{abs},k+1}) = K_0\left(K_1\theta\Big(\big(e^{\theta/T_{\text{abs},k+1}}-1\big)^{-1}-\big(e^{\theta/T_{std}}-1\big)^{-1}\Big)+K_2(T^2_{\text{abs},k+1}-T^2_{std})+K_3e^{-E_D/(R_DT_{\text{abs},k+1})}\right), \tag{2.77}$$

where the values of the constants are given in the initialization section. Reusing the chunks that initialize the constants and compute $\Delta H(T_{\text{abs}})$ in the initialization section, we can compute $\Delta H(T_{\text{abs},k+1})$ as,

75 ⟨ Computing $\Delta H(T_{\text{abs},k+1})$ 75 ⟩ ≡

```
  ⟨ declaration of constants for stored energy 32 ⟩;
  ⟨ ΔH(T_abs) 33 ⟩;
```

This code is used in chunk 53

The integrated fuel power ($P_{\text{F,SUM}}$) at each time step t_{k+1}, is based on the relative fuel power ($q'_{\text{NFRAC},k+1}$) and is given by the numerical approximation of the integral version shown in DD3 of the SRS as:

$$P_{\text{F,SUM},k+1} = \sum_{0}^{i=k+1} q'_{\text{NFRAC},i}\Delta t_i, \tag{2.78}$$

where $q_{\text{NFRAC},i}$ is the relative fuel power at t_i.

76 ⟨ Computing $P_{\text{F,SUM},k+1}$ 76 ⟩ ≡

```
  *f_p = *f_p + (*q_NFRAC * (*delta));
```

This code is used in chunk 53

2.7.10 Computing $C_1, C_2, C_3, c_{p,1}, c_{p,2}, c_{p,3}$

We evaluate the thermal capacitances and the specific heats in the same way we did in the initialization sections 2.6.6, 2.6.7 and 2.6.8. So we reuse the chunks that have implemented the capacitances C_1, C_2, C_3 and their respective specific heats in the initialization section to compute the thermal capacitances and the specific heats at t_{k+1}.
At time t_{k+1}, the average fuel specific heat ($c_{p,1}$) is computed based on $T_{\text{CL},k+1}$ and $T_{S,k+1}$. Taking this $c_{p,1,k+1}$, we evaluate $C_{1,k+1}$ as given in Equation 2.45.

78 ⟨ Computing $C_{1,k+1} = \pi r_f^2\rho_1 c_{p,1,k+1}$ 78 ⟩ ≡

```
  float t_c, t_d;
  int i, iflag;
  i = 2;
  iflag = 3;
  float ts = *t_S;
  float tcl = *t_CL;
      /* function calpro evaluates Cp(T_S,k+1) */
  calpro_(&ts, &i, &iflag, &t_c, &icnt);
      /* function calpro evaluates Cp(T_CL,k+1) */
  calpro_(&tcl, &i, &iflag, &t_d, &icnt);
  ⟨ Calculation of C1 and cp,1 36 ⟩;
```

This code is used in chunk 53

At time t_{k+1}, the specific heat of the clad ($c_{p,2}$) is computed based on $T_{2,k+1}$. Taking this $c_{p,2,k+1}$, we evaluate $C_{2,k+1}$ as given in Equation 2.51.

79 ⟨ Computing $C_{2,k+1} = 2\pi r_c \tau_c \rho_2 c_{p,2,k+1}$ 79 ⟩ ≡

```
      /* calculation of specific heat of the clad ( $c_{p,2,k+1}$) at $T_2$ by calpro */
  i = 3;
  iflag = 2;
  float t2 = *t_2;
  float cp2;
  calpro_(&t2, &i, &iflag, &cp2, &icnt);
   ⟨ Calculation of $C_2$ and $c_{p,2}$ 39 ⟩;
```

This code is used in chunk 53

At time t_{k+1}, the specific heat at the centerline ($c_{p,3}$) is computed based on $T_{\mathrm{CL},k+1}$. Taking this $c_{p,3,k+1}$, we evaluate $C_{\mathrm{CL},k+1}$ as given in Equation 2.52.

80 ⟨ Computing $C_{\mathrm{CL},k+1} = \pi r_f^2 \rho_1 c_{p,3,k+1}$ 80 ⟩ ≡

```
      /* calculation of specific heat $c_{p,3,k+1}$ at $T_{\mathrm{CL}}$ by calpro */
  i = 2;
  iflag = 2;
  tcl = *t_CL;
  float cp3;
  calpro_(&tcl, &i, &iflag, &cp3, &icnt);
   ⟨ Calculation of $C_{\mathrm{CL}}$ and $c_{p,3}$ 42 ⟩;
```

This code is used in chunk 53

2.7.11 Computing k_{AV}

Since k_{AV} is represented as first order polynomial function of temperature, at time t_{k+1}, the average fuel conductivity is explicitly obtained by integrating that expression from T_S to T_{CL}. That is,

$$k_{\mathrm{AV}} = \int_{T_S}^{T_{\mathrm{CL}}} \frac{k dT}{(T_{\mathrm{CL}} - T_S)} \tag{2.79}$$

$$= \frac{1}{(T_{\mathrm{CL}} - T_S)} [K(T)]_{T_S}^{T_{\mathrm{CL}}}, \tag{2.80}$$

where

$$K(T) = \int k dT, \tag{2.81}$$

Hence,

$$k_{\mathrm{AV}} = \frac{K(T_{\mathrm{CL}}) - K(T_S)}{(T_{\mathrm{CL}} - T_S)} \tag{2.82}$$

82 ⟨ Computing $k_{\mathrm{AV},k+1}$ 82 ⟩ ≡

```
  float t_a;      /* evaluation of $K(T)$ at $T_S$ by calpro */
```

```
i = 1;
iflag = 1;
ts = *t_S;
calpro_(&ts, &i, &iflag, &t_a, &icnt);
float t_e;     /* evaluation of K(T) at T_CL by calpro */
tcl = *t_CL;
calpro_(&tcl, &i, &iflag, &t_e, &icnt);
    /* calculation of average fuel conductivity */
*k_AV = (t_e - t_a)/(*t_CL - *t_S);
```

This code is used in chunk 53

2.7.12 Computing q'_{out}

We calculate the heat out (q'_{out}) in the same way as done in the initialization section (2.6.10). The heat out at time t_{k+1} depends on the value of T_2 and h_c at t_{k+1} and is given by DD27 of the SRS as:

$$q'_{\text{out},k+1} = \frac{1}{R_{2,k+1}}\Big(\frac{T_{2,k+1} - T_B}{q'_{N_{\max}}}\Big), \tag{2.83}$$

where $R_{2,k+1}$ is the effective resistance between the clad and the coolant film and is given by DD12 of the SRS as,

$$R_{2,k+1} = \frac{1}{2\pi r_c h_{c,k+1}} \tag{2.84}$$

Reusing the chunk calculating q'_{out} from the initializing section, we can calculate $q'_{\text{out},k+1}$

84 ⟨Computing $q'_{\text{out},k+1}$ 84⟩ ≡

```
    ⟨Calculation of q'_out 50⟩;
```

This code is used in chunk 53

2.7.13 Computing rate of oxidation

We calculate the rate of oxidation (R_{ox}) in the same way as we did in the initialization section (2.6.9). The R_{ox} at time t_{k+1} depends on the value of T_2 at t_{k+1} and δ_{ox} at t_k and is given by DD5 of the SRS,

$$R_{\text{ox},k+1} = \frac{A}{1.56\delta_{\text{ox},k}} e^{\frac{-B}{R(T_{2,k+1}+273)}}, \tag{2.85}$$

where the values of constants A, B/R are given by Table TB1 of the SRS. Table TB1 uses different values for A and B/R if the temperature is greater than 1580^oC. We evaluate value of R_{ox} at t_{k+1} using the same chunk which calculates R_{ox} during the initialization section. However, before the chunk is called, the assignment of values to the variables A and $BbyR$ is done based on the value of T_2 as given by Table TB1 of the SRS.

86 ⟨Computing $R_{\text{ox},k+1}$ 86⟩ ≡

```
float A, BbyR;
```

```
  if (*t_2 ≤ 1580.0) {
    A = 6.48 · 10^{−08};
    BbyR = 13586.0;
  }
  else {
    A = 1.0 · 10^{−06};
    BbyR = 16014.0;
  }
  ⟨ Calculation of R_ox 45 ⟩;
  if (*t_2 ≥ 1850.0) {
    *rate_ox = (A/(1.56 * (*delta_ox))) * exp(−(BbyR)/(1850.0 + 273.0));
  }
```

This code is used in chunk 53

2.7.14 Computing metal water reaction heat (q_{MWR})

We calculate the metal water reaction heat in the same way as we did in the initialization section (2.6.9). The q_{MWR} at time t_{k+1} depends on the value of R_{ox} at t_{k+1} and is given by DD5 of SRS as:

$$q'_{\text{MWR},k+1} = R_{\text{ox},k+1} 2\pi r_c \rho_2 q_r, \tag{2.86}$$

So, for evaluating $q'_{\text{MWR},k+1}$, we reuse the chunk that calculates q'_{MWR} in the steady state. Taking assumption A13 of the SRS into consideration, when all the clad material gets oxidized, that is, when the thickness of the reacted zircaloy (δ_{ox}) becomes equal to or greater than the clad thickness (τ_c), then there will not be any more metal water reaction taking place and hence no more q'_{MWR} will be generated. That is,

$$\delta_{\text{ox}} \geq \tau_c \Rightarrow q'_{\text{MWR}} = 0 \tag{2.87}$$

88 ⟨ Computing $q'_{\text{MWR},k+1}$ 88 ⟩ ≡

```
  ⟨ Calculation of q'_MWR 46 ⟩;
  if (*delta_ox ≥ *tau_c) {
    *q_MWR = 0.0;
  }
```

This code is used in chunk 53

2.7.15 Computing oxidation layer thickness

We calculate the oxidation layer thickness in the same way as we did in the initialization section (2.6.9). The δ_{ox} at time t_{k+1} depends on the value of R_{ox} at t_{k+1} and is given as:

$$\delta_{\text{ox},k+1} = \delta_{\text{ox},k} + R_{\text{ox},k+1}\Delta t \tag{2.88}$$

So, for evaluating $\delta_{\text{ox},k+1}$, we reuse the chunk that calculates δ_{ox} in the steady state. But once the δ_{ox} becomes equal to or greater than the clad thickness (τ_c), as there will not be anymore metal water reaction taking place, the rate of oxidation of the clad becomes zero. That is,

$$\delta_{\text{ox}} \geq \tau_c \Rightarrow R_{\text{ox}} = 0 \tag{2.89}$$

90 ⟨ Computing $\delta_{\text{ox},k+1}$ 90 ⟩ ≡

```
  if (*delta_ox ≥ *tau_c) {
    *rate_ox = 0.0;
  }
  ⟨ Calculation of δ_ox 48 ⟩;
```

This code is used in chunk 53

2.7.16 Computing Integrated metal water reaction heat (q'_{MWRI})

The integrated metal water reaction heat is a summation of q'_{MWR} normalized by $q'_{N_{\max}}$ at each time step. At time t_{k+1}, the q'_{MWRI} is based on $q'_{\mathrm{MWR},k+1}$ and is given by the numerical approximation of the integral form given in DD26 of the SRS as,

$$q'_{\mathrm{MWRI},k+1} = \frac{1}{q'_{N_{\max}}} \sum_{i=0}^{k+1} q'_{\mathrm{MWR},i} \Delta t_i \tag{2.90}$$

92 ⟨ Computing $q'_{\mathrm{MWRI},k+1}$ 92 ⟩ ≡

```
*q_MWRI = *q_MWRI + ((*q_MWR/(*q_Nmax)) * (*delta));
```

This code is used in chunk 53

We store the program into the C file

94 ⟨ fuel_temp.c 94 ⟩ ≡

```
#include <math.h>
#include <assert.h>
#include <stdio.h>
#include <stdlib.h>
   ⟨ fuel temp function 2 ⟩;
```

Equation	x	$a(x)$	$b(x)$	$u(x,t)$
2.1	T_1	$-\frac{1}{R_1C_1}$	$\frac{T_2+q'_NR_1}{R_1C_1}$	1
2.2	T_2	$-\frac{(R_1+R_2)}{R_1R_2C_2}$	$\frac{T_1R_2+q'_{\mathrm{MWR}}R_1R_2+T_BR_1}{R_1R_2C_2}$	1
2.3	T_{CL}	$-\frac{1}{R_3C_{\mathrm{CL}}}$	$\frac{T_1+q'_NR_3}{R_3C_{\mathrm{CL}}}$	1

Table 2.1: Table of functions for ODEs representing different instance models

Bibliography

[1] FPManual. *FP Manual.* Nuclear Power Generation Company, March 1982.

[2] Lei Lai. Requirements documentation for engineering mechanics software: Guidelines, template and a case requirements documentation for engineering mechanics software: Guidelines, template and a case studycase study. Master's thesis, McMaster University, Hamilton, Ontario, Canada, September 2004.

[3] M. Ortuno, A. Marquez, S.Gallego, C.Neipp, and A. Belndez. An experiment in heat conduction using hollow cylinders. *European Journal of Physics*, 32:3–6, 2011.

[4] G. S Sawhney. *Heat and Mass Transfer.* LK International Publishing House Pvt. Ltd, S-25, Green Park Extension, Uphaar Cinema Market, New Delhi- 110 016(India), second edition, 2010.

[5] W. Spencer Smith and Lei Lai. A new requirements template for scientific computing. In J. Ralyté, P. Ȧgerfalk, and N. Kraiem, editors, *Proceedings of the First International Workshop on Situational Requirements Engineering Processes – Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05*, Paris, France, August 2005. In conjunction with 13th IEEE International Requirements Engineering Conference.

Index

List of Refinements