ANL/NSE-17/4 Rev. 1
(ANL/NE-17/4 Rev. 1)

SAM Theory Manual

Nuclear Science and Engineering Division

About Argonne National Laboratory

Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov

DOCUMENT AVAILABILITY

Online Access: U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free at OSTI.GOV (http://www.osti.gov/), a service of the U.S. Dept. of Energy's Office of Scientific and Technical Information

Reports not in digital format may be purchased by the public from the National Technical Information Service (NTIS):

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandra, VA 22312
www.ntis.gov
Phone: (800) 553-NTIS (6847) or (703) 605-6000
Fax: (703) 605-6900
Email: **orders@ntis.gov**

Reports not in digital format are available to DOE and DOE contractors from the Office of Scientific and Technical Information (OSTI):

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
www.osti.gov
Phone: (865) 576-8401
Fax: (865) 576-5728
Email: **reports@osti.gov**

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor UChicago Argonne, LLC, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.

ANL/NSE-17/4 Rev. 1
(ANL/NE-17/4 Rev. 1)

SAM Theory Manual

---

prepared by
Rui Hu, Ling Zou, Guojun Hu, Daniel Nunez, Travis Mui, Tingzhou Fei
Nuclear Science and Engineering Division, Argonne National Laboratory

February 2021

Erratum to accompany "*SAM Theory Manual*" (Argonne National Laboratory report ANL/NSE-17/4 Rev. 1)

August 2021

After initial publication of this report, the authors were made aware of the report number (ANL/NSE-17/4 Rev. 1) does not correctly reflect the original published report (ANL/NE-17/4). As the report number can potentially confuse the readers, another report number (ANL/NE-17/4 Rev. 1) has been added in the present version.

ABSTRACT

The System Analysis Module (SAM) is an advanced and modern system analysis tool under development at Argonne National Laboratory for advanced non-LWR reactor safety analysis. It aims to provide fast-running, modest-fidelity, whole-plant transient analyses capabilities, which are essential for fast turnaround design scoping and engineering analyses of advanced reactor concepts. While SAM is being developed as a system-level modeling and simulation tool, advanced modeling techniques being implemented include a reduced-order three-dimensional module, pseudo 3-D conjugate heat transfer modeling in reactor core, flexible and multi-scale modeling of heat transfer between fluid and structures, in addition to the advances in software environments and design, and numerical methods.

SAM aims to be a generic system-level safety analysis tool for advanced non-LWRs, including Liquid-Metal-cooled fast Reactors (LMR), Molten Salt Reactors (MSR), Fluoride-salt-cooled High-temperature Reactors (FHR), and High-Temperature Gas-cooled Reactors (HTGR). SAM takes advantage of advances in physical modeling, numerical methods, and software engineering to enhance its user experience and usability. It utilizes an object-oriented computational framework (MOOSE), and its underlying meshing and finite-element library and linear and non-linear solvers, to leverage the modern advanced software environments and numerical methods.

This document provides the theoretical and technical basis of the code to help users understand the underlying physical models (such as governing equations, closure models, and component models), system modeling approaches, numerical discretization and solution methods, and the overall capabilities in SAM. As new code capabilities and features are added, the SAM Theory Manual will be updated periodically to keep it consistent with the state of the development.

Contents

List of Figures

List of Tables

1 SAM Overview

The System Analysis Module (SAM) is an advanced system analysis tool being developed at Argonne National Laboratory under the support of U.S. Department of Energy (DOE) Nuclear Energy Advanced Modeling and Simulation (NEAMS) program. It aims to be a modern system analysis code, which takes advantages of the advancements in computing power, software design, numerical methods, and physical models over the past two decades. SAM focuses on modeling advanced reactor concepts such as SFRs (sodium fast reactors), LFRs (lead-cooled fast reactors), and FHRs (fluoride-salt-cooled high temperature reactors) or MSRs (molten salt reactors), and HTGRs (high-temperature gas-cooled reactors). These advanced concepts are distinguished from light-water reactors (LWR) in their use of single-phase, low-pressure (except HTGRs), high-temperature, and non-unity Prandtl number coolants. This simple yet fundamental change has significant impacts on core and plant design, the types of materials used, component design and operation, fuel behavior, and the significance of the fundamental physics in play during transient plant simulations.

SAM is aimed to solve the tightly-coupled physical phenomena including fission reaction, heat transfer, fluid dynamics, and thermal-mechanical response in reactor structures, systems and components in a fully-coupled fashion but with reduced-order modeling approaches to facilitate rapid turn-around for design and safety optimization studies. This document provides the theoretical basis of the code to help users understand the underlying physical models (such as governing equations, closure models, and component models), system modeling approaches, numerical discretization and solution methods, and the overall capabilities in SAM. As new code capabilities and features are added, the SAM Theory Manual will evolve with periodic updates to keep it consistent with the state of the development, implementation, and model additions/revisions.

1.1 Software Structure

SAM is being developed as a system-level modeling and simulation tool with higher fidelity (compared to existing system analysis tools), and with well-defined and validated simulation capabilities for advanced reactor systems. It provides fast-running, modest-fidelity, whole-plant transient analyses capabilities. To fulfill the code development, SAM utilizes the object-oriented application framework MOOSE (Gaston et al., 2009) and its underlying meshing and finite-element library, libMesh (Kirk et al., 2006) and linear and non-linear solvers PETSc (PETSc, 2020), to leverage the available advanced software environments and numerical methods. The high-order spatial discretization schemes, fully implicit and high-order time integration schemes, and the advanced solution method (such as the Jacobian-free Newton–Krylov (JFNK) method) are the key aspects in developing an accurate and computationally efficient model in SAM.

The software structure of SAM is illustrated in Figure 1.1. In addition to the fundamental physics modeling of the single-phase fluid flow and heat transfer, SAM incorporates advances in the closure models (such as convective heat transfer correlations) for reactor system analysis developed over the past several decades. A set of Components, which integrate the associated physics modeling in the component, have been developed for friendly user interactions. This component-based modeling strategy is similar to what is implemented in RELAP-7 (Berry et al., 2015), which is also a MOOSE-based system analysis tool (focused on LWR simulations). A flexible coupling interface has been developed in SAM so that multi-scale, multi-physics modeling capabilities can be achieved by integrating with other higher-fidelity or conventional simulation tools.

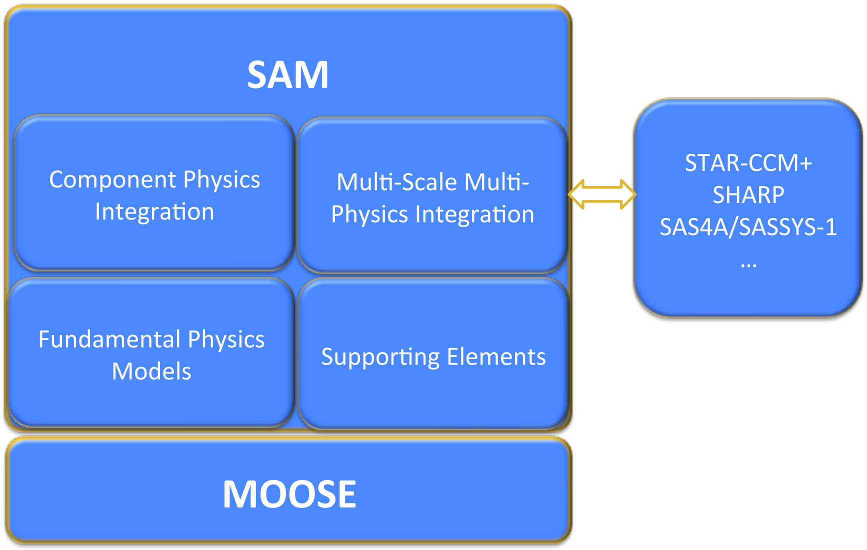


Figure 1.1: SAM Code Structure

1.2 Governing Theory

1.2.1 Fluid dynamics

Fluid dynamics is the main physical model of the SAM code. SAM employs a one-dimensional transient model for single-phase incompressible but thermally expandable flow. The governing equations consist of the continuity equation, momentum equation, and energy equations. A multi-dimensional module is also under development to model the multi-dimensional flow and thermal stratification in the upper plenum or the cold pool of an SFR. The details of the one-D single-phase flow model for incompressible thermally expandable flow are discussed in Chapter 2. The multi-dimensional flow model is detailed in Chapter 9.

1.2.2 Heat transfer

Heat structures model heat conduction inside solids and permit the modeling of heat transfer at interfaces between solid and fluid components. Heat structures are represented by one-dimensional or two-dimensional heat conduction in Cartesian or cylindrical coordinates. Temperature-dependent thermal conductivities and volumetric heat capacities can be provided in tabular or functional form. The modeling capabilities of heat structures can be used to predict the temperature distributions in solid components such as fuel pins or plates, heat exchanger tubes, and pipe and vessel walls, as well as to calculate the heat flux conditions for fluid components. Flexible conjugate heat transfer and thermal radiation modeling capabilities are also implemented in SAM. The details of heat transfer modeling are discussed in Chapter 3.

1.2.3 Closure Models

The fluid equation of state (EOS) model is required to complete the governing flow equations, which are based on the primitive variable formulation; therefore, the dependency of fluid properties and their partial derivatives on the state variables (pressure and temperature) are implemented in the EOS model. Some fluid properties, such as sodium, air, salts like FLiBe and FLiNaK, have been implemented in SAM. It can also utilize the fluid properties available in the MOOSE Fluid Properties Module. The details of built-in fluid and solid property models used in SAM are discussed in Chapter 4.

Empirical correlations for friction factor and convective heat transfer coefficient are also required in SAM because of its one-dimension approximation of the flow field. The friction and heat transfer coefficients are dependent on flow geometries as well as operating conditions during the transient. The details of closure models used in SAM are discussed in Chapter 5.

1.2.4 Mass transport model development

The mass transport modeling capability is needed to model sources and transport of particles for a number of applications, such as tritium transport, delayed neutron precursor drift, radioactive isotope transport for molten salt fueled/cooled systems. A general passive scalar transport model has been implemented in SAM, and it can be used to track any number of species carried by the fluid flow. The passive scalar transport model is discussed in Chapters 2.

1.2.5 Reactor Kinetics model development

SAM employs a built-in point-kinetics model, including reactivity feedback and decay heat modeling. Various reactivity feedback models have been developed and integrated with the point-kinetics module, including fuel axial expansion, core radial expansion, fuel Doppler, and coolant density reactivity. The details of reactor kinetics and reactivity feedback models are discussed in Chapter 6.

1.2.6 Numerical Methods

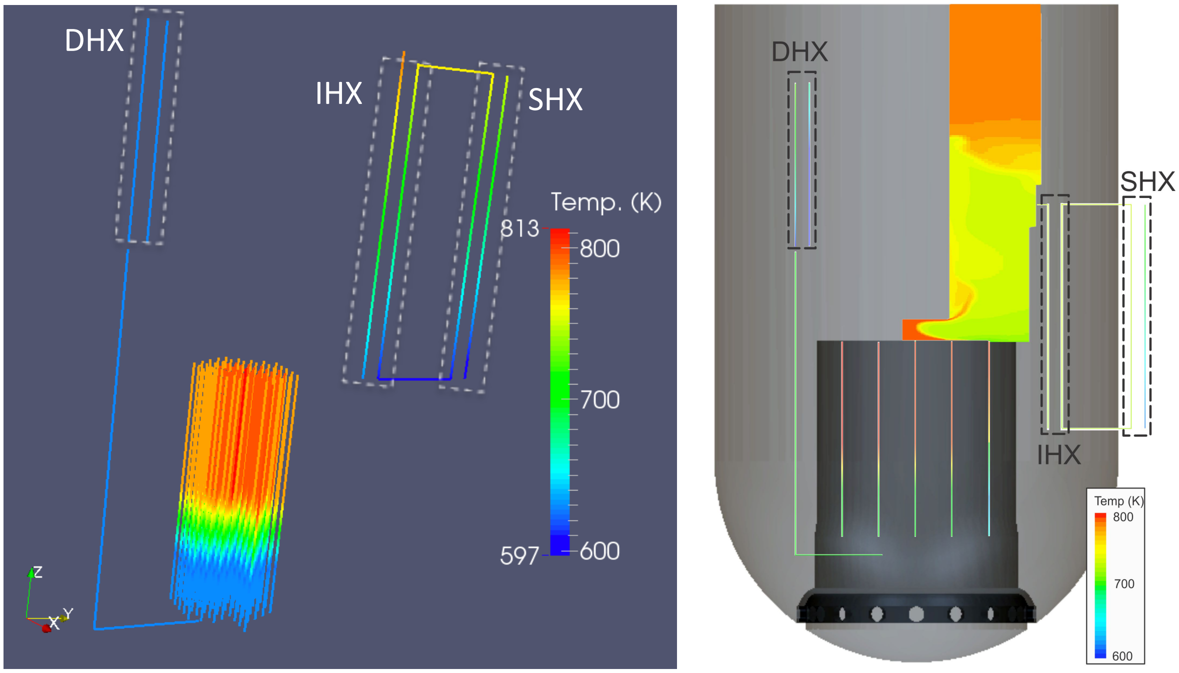
SAM is a finite-element-method based code. The “weak forms” of the governing equations are implemented in SAM. It uses the Jacobian-Free Newton Krylov (JFNK) solution method to solve the equation system. The JFNK method uses a multi-level approach, with outer Newton’s iterations (nonlinear solver) and inner Krylov subspace methods (linear solver), in solving large nonlinear systems. The concept of ‘Jacobian-free’ is proposed, because deriving and assembling large Jacobian matrices could be difficult and expensive. The JFNK method has become an increasingly popular option for solving large nonlinear equation systems and multi-physics problems, as observed in a number of different disciplines (Knoll and Keyes, 2004). One feature of JFNK is that all the unknowns are solved simultaneously in a fully coupled fashion. This solution scheme avoids the errors from operator splitting and is especially suitable for conjugate heat transfer problems in which heat conduction in a solid is tightly coupled with fluid flow. The details of the numerical methods used in SAM are discussed in Chapter 7.

1.3 Overview of Current Capabilities

To develop a system analysis code, numerical methods, mesh management, equations of state, fluid properties, solid material properties, neutronics properties, pressure loss and heat transfer closure laws, and good user input/output interfaces are all indispensable. SAM leverages the MOOSE framework and its dependent libraries to provide JFNK solver schemes, mesh management, and I/O interfaces while focusing on new physics and component model development for advanced reactor systems. The details of component models in SAM are discussed in Chapter 8. The developed physics and component models provide several major modeling features:

1. One-D pipe networks represent general fluid systems such as the reactor coolant loops.
2. Flexible integration of fluid and solid components, able to model complex and generic engineering system. A general liquid flow and solid structure interface model was developed for easier implementation of physics models in the components.
3. A pseudo three-dimensional capability by physically coupling the 1-D or 2-D components in a 3-D layout. For example, the 3-D full-core heat-transfer in an SFR reactor core can be modeled. The heat generated in the fuel rod of one fuel assembly can be transferred to the coolant in the core channel, the duct wall, the inter-assembly gap, and then the adjacent fuel assemblies.
4. Pool-type reactor specific features such as liquid volume level tracking, cover gas dynamics, heat transfer between 0-D pools, fluid heat conduction, etc. These are important features for accurate safety analyses of SFRs or other advanced reactor concepts.
5. A computationally efficient multi-dimensional flow model is under development, mainly for thermal mixing and stratification phenomena in large enclosures for safety analysis. It was noted that an advanced and efficient thermal mixing and stratification modeling capability embedded in a system analysis code is very desirable to improve the accuracy of advanced reactor safety analyses and to reduce modeling uncertainties.
6. A general mass transport capability has been implemented in SAM based on the passive scalar transport. The code can track any number of species carried by the fluid flow for various applications.
7. An infrastructure for coupling with external codes has been developed and demonstrated. The code coupling with STAR-CCM+ (Hu et al., 2014b), SAS4A/SASSYS-1 (Fanning and Hu, 2016; Brunett et al., 2019), Mammoth/RattleSnake (Hu et al., 2019a), Nek5000, and BISON (Martineau et al., 2018) have been demonstrated, while the coupling with the PORTEUS code is undergoing. Details are discussed in Chapter 10.

An example of SAM simulation results of an SFR is shown in Figure 1.2.



(a) SAM model with 61 core channels

(b) Coupled SAM and CFD code simulation

Figure 1.2: SAM simulation results of an SFR.

2 One-Dimensional Fluid Model

2.1 Governing Equations

The transport equations for one-dimensional, single-phase flow can be described by the following set of partial differential equations. The mass, momentum, and energy conservation equations are closed by the equation of state for the fluid.

$$
\begin{aligned}
&\frac{\partial \rho}{\partial t}+\frac{\partial(\rho u)}{\partial z}=0\\
&\frac{\partial(\rho u)}{\partial t}+\frac{\partial(\rho u u)}{\partial z}=-\frac{\partial p}{\partial z}-\rho g-\nabla\cdot\overline{\overline{\tau}}\\
&\frac{\partial(\rho H)}{\partial t}+\frac{\partial(\rho u H)}{\partial z}=\nabla\cdot(k\nabla T)-\nabla q_r''+q'''+\frac{Dp}{Dt}+\phi\\
&\rho=\rho(p,T)
\end{aligned}
\tag{2.1}
$$

where,

t: time;
z: the axial coordinate in flow direction;
ρ: the coolant density;
u: velocity;
g: the acceleration due to gravity;
p: pressure;
$\overline{\overline{\tau}}$: shear stress;
T: temperature;
H: enthalpy, and $H=E+p/\rho$, in which E is internal energy;
k: fluid thermal conductivity;
q_r'': radiation heat flux;
q'': volumetric internal heat source;
ϕ: source terms due to external forces, $\phi=-v\cdot\left(\nabla\cdot\overline{\overline{\tau}}+\rho g\right)$, including frictional dissipation, gravity, etc.

The shear stress in the momentum equation can be simplified using the concept of wall friction coefficient. The energy equation can be simplified by neglecting the energy variation due to fluid conduction, radiation heat, thermal expansion, and the source terms due to external forces, as these terms are negligible compared to surface or internal heating terms. It should be also noted that the modeling of fluid conduction could be included by a user option if users deem it important for certain applications. After the simplifications and applying the continuity equation into the momentum and energy equations, the set of governing equations can be written in the conservative form Eq. (2.2) or in the non-conservative form Eq. (2.3).

$$
\begin{aligned}
&\frac{\partial \rho}{\partial t}+\frac{\partial (\rho u)}{\partial z}=0\\
&\frac{\partial (\rho u)}{\partial t}+\frac{\partial (\rho u u+p)}{\partial z}=-\rho g-\frac{f}{D_e}\frac{\rho u|u|}{2}\\
&\frac{\partial (\rho H)}{\partial t}+\frac{\partial (\rho u H)}{\partial z}=q'''\\
&p=p\left(\rho,\rho u,\rho E\right) \quad \text{or} \quad \rho=\rho\left(p,T\right)
\end{aligned}
\tag{2.2}
$$

In which, f: the friction coefficient; D_e: equivalent hydraulic diameter. When considering the convection heat flux from solid surface q_s'', $q'''=\frac{q_s''P_h}{A_c}$, where P_h and A_c respectively denote heated perimeter and cross-sectional area of the coolant channel.

$$
\begin{aligned}
&\frac{\partial \rho}{\partial t}+\frac{\partial (\rho u)}{\partial z}=0\\
&\rho\frac{\partial u}{\partial t}+\rho u\frac{\partial u}{\partial z}=-\frac{\partial p}{\partial z}-\rho g-\frac{f}{D_e}\frac{\rho u|u|}{2}\\
&\rho C_p\frac{\partial T}{\partial t}+\rho C_p u\frac{\partial T}{\partial z}=q'''\\
&\rho=\rho\left(p,T\right)
\end{aligned}
\tag{2.3}
$$

In which C_p: the specific heat, $C_p=\frac{\partial H}{\partial T}$.

To solve the above set of PDEs, two out of three variables of density, pressure, and temperature (or total energy) must be selected as the state variables, along with the velocity (or mass flux) to compose the three state variables for the system of equations. Two common approaches used in the literature include: (1) primitive variable (or pressure) based formulation, in which the state variables are pressure (p), velocity (u), and temperature (T); and (2) conservative variable (or density) based formulation that uses density (ρ), momentum (ρu), and total energy (ρE) as the state variables.

Historically, the pressure-based approach was developed for low-speed incompressible flows, while the density-based approach was originally designed for high-speed compressible flows (Moureau et al., 2005). Both methods have been extended and reformulated to solve and operate for a wide range of flow conditions beyond their traditional or original intent. The use of conservative variable (ρ, ρu, ρE) as independent variables would require the pressure to be calculated from the equation of state (EOS) model. However, this approach may increase the difficulties in code convergence for applications of incompressible or nearly incompressible liquid flows, since small perturbations in ρ (density) or ρE (volumetric total energy) would cause significant changes in pressure. To use Newton-type nonlinear solvers, the $\frac{\partial p}{\partial \rho}$, $\frac{\partial p}{\partial \rho u}$, $\frac{\partial p}{\partial \rho E}$, $\frac{\partial T}{\partial \rho}$, $\frac{\partial T}{\partial \rho u}$, $\frac{\partial T}{\partial \rho E}$ terms are required for constructing the Jacobian matrix; even if a matrix-free method is used, these derivative terms may still be needed for the preconditioning matrix. The existence of these terms significantly increases the difficulties in deriving the Jacobian or preconditioning matrix and may cause convergence problems when using real fluid EOS.

Although the conservative variable based formulation has the advantages in the applications of compressible flow such as the capability to capture shock waves, the primitive variable based

FEM formulation is more suitable for incompressible or nearly incompressible flows, such as the fluid flow in SFRs, LFRs, or MSRs/FHRs. With the primitive variable approach, using the integral equations in the conservative form Eq. (2.3) will still ensure the conservation laws of the fluid equations. Hence, a primitive variable based formulation is used in this work for SFR system analysis applications. Further details of EOS modeling are discussed in Chapter 4.

2.2 Stabilization Schemes

Finite element analysis of incompressible flows requires stabilization to avoid the potential numerical instabilities. The presence of advection terms (first order terms) in the governing equations can result in spurious node-to-node oscillations (Hu, 2015). Stabilization methods being successfully applied in structure problems, where no convection is present, may totally fail when they are applied to convection-dominated problems, as they occur frequently in fluid mechanics. This is particularly the case with Bubnov-Galerkin methods, which are weighted residual methods, where the test functions are set equal to the shape functions (Moureau et al., 2005). The role of convection in differential equations can be defined by the Peclet number or Reynolds number. The higher these numbers are, the more dominant the convection term and the stronger the oscillations.

The same situation can also be found in the finite difference context. There, the same problem with oscillations occurs when using central differences for the advective operator. In finite difference methods, it is well known that upwind differencing on the convective term does not show oscillatory solutions, but introduces over-diffusive results. A simple Taylor series analysis proves that upwinding is only first order accurate, in contrast to the second order accurate, but oscillatory, central differences. This analysis also elucidates that upwinding can also be interpreted as central differences plus artificial diffusion. Thus, the "right" combination of central and upwind differences may introduce the optimal amount of artificial diffusion, which leads to accurate and oscillation-free solutions (Moureau et al., 2005).

Starting in the 1970s a large number of FEMs arose with different ideas to include the upwind effect in finite element analyses. The Streamline-Upwind/Petrov-Galerkin (SUPG) method, introduced from Brooks and Hughes (1982) can be considered as the first successful stabilization technique to prevent oscillations in convection-dominated problems in the FEM. The main concept is to introduce artificial diffusion in the streamline direction only, interpreted as a modification of the test function of the advection terms, and to enforce consistency, such that this modified test function is applied to all terms of the weak form. The term "artificial diffusion" is not fully applicable any longer, because the stabilized weak form cannot be manipulated such that only a diffusion term is extracted. The exact solution of the original problem still satisfies the SUPG stabilized weak form.

Considering a PDE of the general form,

$$Lu = f \tag{2.4}$$

where L is any differential operator. The SUPG weak form of the problem is:

$$\int_{\Omega} w^{*} \cdot (Lu - f)\, d\Omega = 0 \tag{2.5}$$

The standard Bubnov-Galerkin test functions w are modified by a streamline upwind perturbation of the kind

$$w^* = w + \tau L_{adv} w = w + \tau \nabla w \tag{2.6}$$

Where L_{adv} is the advective part of the whole operator L, and τ is the stabilization parameter that weights the perturbation. Note that the perturbation is multiplied with the residual form of the differential equation. Thereby, consistency is fulfilled from the beginning in that the exact solution also fulfills the stabilized weak form exactly. It is because of this property of the SUPG stabilization (and the PSPG stabilization mentioned later) that numerical oscillations are prevented without introducing excessive numerical diffusion, and therefore without compromising the accuracy of the solution.

The Pressure-Stabilizing/Petrov-Galerkin (PSPG) method is a common technique used for the stabilization of the Stokes equations. For Stokes equations:

$$\begin{aligned} \text{Continuity equation:} \quad & \nabla \cdot u = 0 \\ \text{Momentum equation:} \quad & \nabla \cdot (\nu \nabla \cdot u + p) = f \end{aligned} \tag{2.7}$$

The PSPG stabilization term, similar to that of the SUPG, consists of a perturbation $\tau \nabla q$ multiplied with the residual of the momentum equation, but it is added to the weak form of the continuity equation.

$$\int_\Omega q \cdot (\nabla \cdot u)\, d\Omega + \int_\Omega \tau \nabla q \cdot [\nabla \cdot (\nu \nabla \cdot u + p) - f]\, d\Omega = 0 \tag{2.8}$$

In mixed convection-dominant problems, such as the incompressible Navier-Stokes equations with high Reynolds number, SUPG and PSPG (called herein SUPG/PSPG) stabilization have to be applied to obtain satisfactory results. It should be mentioned that the PSPG stabilization parameter does not necessarily have to be identical with the SUPG stabilization parameter. In this work, the weak forms of the stabilization schemes for incompressible flow are derived as:

$$\begin{aligned}
&\left(\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial z},\ \psi \right) \\
&\quad + \left(\rho \frac{\partial u}{\partial t} + \rho u \frac{\partial u}{\partial z} + \frac{\partial p}{\partial z} + \rho g + \frac{f}{D_e} \frac{\rho u |u|}{2},\ \tau_{PSPG} \nabla \psi \right) = 0 \\[2ex]
&\left(\frac{\partial \rho u}{\partial t} + \frac{\partial \rho u u}{\partial z} + \frac{\partial p}{\partial z} + \rho g + \frac{f}{D_e} \frac{\rho u |u|}{2},\ \psi \right) \\
&\qquad\qquad + \left(\rho \frac{\partial u}{\partial t} + \rho u \frac{\partial u}{\partial z} + \frac{\partial p}{\partial z} + \rho g + \frac{f}{D_e} \frac{\rho u |u|}{2},\ \tau_{SUPG} \nabla \psi \right) = 0 \\[2ex]
&\left(\frac{\partial \rho H}{\partial t} + \frac{\partial \rho u H}{\partial z} - q''',\ \psi \right) + \left(\rho C_p \frac{\partial T}{\partial t} + \rho C_p u \frac{\partial T}{\partial z} - q''',\ \tau_{SUPG} \nabla \psi \right) = 0
\end{aligned} \tag{2.9}$$

in which ψ is the test function; τ_{PSPG} and τ_{SUPG} are the stabilization parameters that weights the perturbations; and $(f, \psi) = \int_\Omega \psi \cdot f d\Omega$, is an expression of the volume integral. Note that the regular residuals of all conservation equations are calculated based on the conservative form while the stabilization terms are calculated based on the non-conservative form. This formulation not only strictly ensures the conservation laws, but also is easy to be implemented. A review of stabilized finite element formulations for incompressible flow, including the SUPG and PSPG schemes, can be found in Tezduyar (1992).

In SUPG and PSPG stabilization schemes, additional stabilization terms are added to the standard Galerkin formulation of the fluid equations. Judicious selection of the stabilization parameters, τ_{PSPG} and τ_{SUPG}, plays a key role in determining the stability and accuracy of the formulations. Various τ formulations were proposed in the literature. The stabilization parameters often involve a measure of the local length scale and other parameters such as the local Reynolds number and Courant numbers. For one-dimensional space-time formulation, the formulation of the stabilization parameters can be defined as:

$$\begin{aligned} \tau &= \left[\left(\frac{2}{\Delta t} \right)^2 + \left(\frac{2|u|}{h} \right)^2 + \left(\frac{4\nu}{h^2} \right)^2 \right]^{-1/2} \\ \tau_{SUPG} &= u \cdot \tau \\ \tau_{PSPG} &= \tau(U) \end{aligned} \tag{2.10}$$

Where h is the "element length", u is local velocity, Δt is the time step size, ν is the kinematic viscosity, and U is a global scaling velocity. If U were not defined in the simulation, the local velocity would be used. This formulation is modified based on the one-dimensional space-time formulation given in Shakib (2010). For one-dimensional fluid model in reactor safety analysis, large elements are commonly used, where $\left(\frac{2|u|}{h}\right) \gg \left(\frac{4\nu}{h^2}\right)$ or $\frac{|u|h}{\nu} \gg 2$. Therefore, the stabilization parameters can be simplified as:

$$\begin{aligned} \tau_{SUPG} &= u \cdot \left[\left(\frac{2}{\Delta t} \right)^2 + \left(\frac{2|u|}{h} \right)^2 \right]^{-1/2} \\ \tau_{PSPG} &= \left[\left(\frac{2}{\Delta t} \right)^2 + \left(\frac{2|U|}{h} \right)^2 \right]^{-1/2} \end{aligned} \tag{2.11}$$

Additionally, an upper bound and lower bound are defined for τ_{PSPG} to ensure the robustness of the stabilization scheme under extreme flow conditions.

2.3 Species Transport

The ability to track various species (e.g. delayed-neutron precursors, decay heat precursors, tritium, heavy metal particles, and fission products) in the primary cooling system of reactor systems is key for MSR transient simulation as well as source term evaluation for other advanced reactors. The mass transport modeling capability has been implemented in SAM based on the general scalar transport model.

The scalar particle transport model is shown in Equation 2.12: the second and third terms on the left-hand side represent particle transport and diffusion, respectively; the fourth term represents isotope decay, and the source term is defined on the right-hand side of the equation.

$$\frac{\partial \rho c_i}{\partial t} + \nabla \cdot (\rho u c_i) - \nabla \cdot (D_i \nabla c_i) + \lambda_i \rho c_i = S_i \tag{2.12}$$

where ρ is the bulk fluid density (kg/m^3); c_i is the scalar particle (species) mass concentration ($\#/kg_{fluid}$); u is the fluid velocity (m/s); D_i is the diffusion coefficient for scalar particle i (m^2/s); S_i is the source term ($\#/m^3$); λ_i is the decay constant for scalar particle i (s^{-1}). The species transport equation is solved together with the fluid conversation equations, using the same stabilization scheme discussed above.

3 Heat Transfer Models

3.1 Heat Conduction Modeling

Heat structures model the heat conduction inside the solids and permit the modeling of heat transfer at the interfaces between solid and fluid components. Heat structures are represented by one-dimensional or two-dimensional heat conduction in Cartesian or cylindrical coordinates. Temperature-dependent thermal conductivities and volumetric heat capacities can be provided in tabular or functional form from user-supplied data. The modeling capabilities of heat structures can be used to predict the temperature distributions in solid components such as fuel pins or plates, heat exchanger tubes, and pipe and vessel walls, as well as to calculate the heat flux conditions for fluid components.

The thermal conduction inside the solid structures is governed by a diffusion equation:

$$\rho C_p \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) - Q''' = 0 \tag{3.1}$$

Where k is the solid thermal conductivity, and Q''' is the volumetric internal heat source in the solid. The equation can be discretized in both Cartesian and cylindrical coordinates. The types of boundary conditions (BC) include: (1) Dirichlet BC, $T = T_0$; (2) Neumann BC, $k\nabla T = q_0''$; or (3) convective BC: $-k\nabla T = h\left(T - T_{fluid}\right)$.

Note that different treatments are needed for the diffusion term $\nabla \cdot (k\nabla T)$ in the heat conduction equation in different geometries.

For 1-D plate type,

$$\nabla \cdot (k \nabla T) = \frac{d}{dx}\left(k \frac{dT}{dx}\right) \tag{3.2}$$

For 2-D plate type,

$$\nabla \cdot (k \nabla T) = \frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial z}\left(k \frac{\partial T}{\partial z}\right) \tag{3.3}$$

For 1-D cylindrical type,

$$\nabla \cdot (k \nabla T) = \frac{1}{r}\frac{d}{dr}\left(kr \frac{dT}{dr}\right) \tag{3.4}$$

For 2-D cylindrical type,

$$\nabla \cdot (k \nabla T) = \frac{1}{r}\frac{\partial}{\partial r}\left(kr \frac{\partial T}{\partial r}\right) + \frac{\partial}{\partial z}\left(k \frac{\partial T}{\partial z}\right) \tag{3.5}$$

For 1-D spherical type,

$$\nabla \cdot (k \nabla T) = \frac{1}{r^2}\frac{d}{dr}\left(kr^2 \frac{dT}{dr}\right) \tag{3.6}$$

3.2 Convective Heat Transfer

In many engineering applications, the fluid flow and solid heat conduction is coupled through convective heat transfer at the solid surfaces. The conjugated heat transfer modeling in SAM at the fluid-structure interface is shown schematically in Figure. The fluid is modeled as one-dimensional flow, and the solid structure is modeled as one-D or two-D heat conduction, and they exchange energy at the fluid-structure interface.

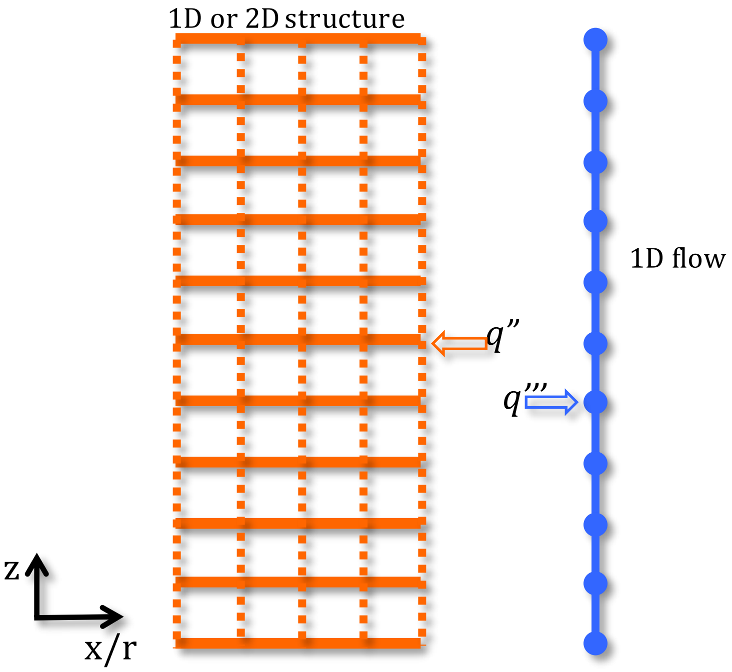


Figure 3.1: The schematic of conjugate heat transfer modeling in SAM

At the fluid-structure interface, the convective heat flux is:

$$q'' = -k\nabla T = h\left(T_w - T_f\right). \tag{3.7}$$

In which h is the heat transfer coefficient, T_w is the wall temperature, and T_f is the fluid temperature.

The weak form of the solid conduction equation (Eq. 3-1) can be written as,

$$\left(\rho C_p \frac{\partial T}{\partial t},\ \psi\right) + (k\nabla T,\ \nabla\psi) - \langle k\nabla T,\ \psi\rangle - \left(Q''',\ \psi\right) = 0 \tag{3.8}$$

In which ψ is the test function; and $(f,\ \psi)$ represents the volume integral, $(f,\ \psi) = \int_\Omega \psi \cdot f d\Omega$; and $\langle f, \psi\rangle$ represents the surface integral, $\langle f, \psi\rangle = \int_\Gamma \psi \cdot f d\Gamma$. For solid elements at the interface, the convective heat flux can be directly applied as boundary conditions for the one-D heat conduction equation, as required by the $\langle k\nabla T, \psi\rangle$ term in Eq. (3.8). However, for fluid elements, it has to be modeled as an additional heat source term in the energy conservation equation,

$$q''' = h\left(T_w - T_f\right)\frac{P_{heated}}{A_c} \tag{3.9}$$

This is implemented as an additional term in the residual calculation at each fluid node (quadrature point):

$$\Delta r_{energy} = \int_{\Omega} q''' \cdot \psi d\Omega = \int_{\Omega} h\left(T_w - T_f\right) \frac{P_{heated}}{A_c} \cdot \psi d\Omega \tag{3.10}$$

Note the above implementation would introduce another term in the stabilized fluid model formulation. For the SUPG scheme used in SAM,

$$\Delta r_{energy} = \int_{\Omega} h\left(T_w - T_f\right) \frac{P_{heated}}{A_c} \cdot \left(\psi + \tau_{supg} \cdot \nabla \psi\right) d\Omega \tag{3.11}$$

3.3 Thermal Radiation Heat Transfer

The radiation heat transfer in engineering systems (enclosures) normally involves nonblack surfaces, which allow multiple reflections to occur. The radiation heat transfer modeling of such enclosures can become very complicated unless some simplifying assumptions are made. A general radiation heat transfer enclosure model is not yet implemented in SAM. Instead, simplified models are implemented to model the radiation heat transfer between two surfaces.

The radiosity, the total radiation energy flux (emitted plus reflected) leaving a surface, can be defined as the difference between the emitted energy flux and the reflected energy flux. The radiosity R_i which corresponds to surface i is given by

$$R_i A_i = \varepsilon_i \sigma T_i^4 A_i + \rho_i \sum_{j=1}^{n} R_j F_{ji} A_j \tag{3.12}$$

in which,

R: radiosity
ε: emissivity
σ: Stefan-Boltzmann constant
T: temperature
ρ : $1-\varepsilon$, reflectivity
F_{ji}: view factor from surface j to surface i
A_j: area of surface j

The radiation enclosure model should also include the reciprocity relation

$$A_i F_{ij} = A_j F_{ji} \tag{3.13}$$

and the unity rule which enforces the following constraint

$$\sum_{j=1}^{n} F_{ij} = 1 \tag{3.14}$$

The net rate of radiation heat transfer between two surfaces in a general enclosure, from surface i to surface j, can be given by

$$\dot{Q}_{i \to j} = A_i R_i F_{ij} - A_j R_j F_{ji} \tag{3.15}$$

So far, only a two-surface enclosure model is implemented in SAM. The net heat transfer rate between the two surfaces in a two-surface enclosure can be given by

$$\dot{Q}_{12} = \frac{\sigma\left(T_1^4 - T_2^4\right)}{\frac{1-\varepsilon_1}{A_1\varepsilon_1} + \frac{1}{A_1F_{12}} + \frac{1-\varepsilon_2}{A_2\varepsilon_2}} = -\dot{Q}_{21} \tag{3.16}$$

For infinitely large parallel plates: $A_1 = A_2, F_{12} = F_{21} = 1$,

$$q_1'' = \frac{\dot{Q}_{12}}{A_1} = \frac{\sigma\left(T_1^4 - T_2^4\right)}{\frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2} - 1} = -q_2'' \tag{3.17}$$

For nfinitely long concentric cylinders: $\frac{A_1}{A_2} = \frac{r_1}{r_2}, F_{12} = 1, F_{21} = \frac{r_1}{r_2}$,

$$\begin{aligned} q_1'' &= \frac{\dot{Q}_{12}}{A_1} = \frac{\sigma\left(T_1^4 - T_2^4\right)}{\frac{1}{\varepsilon_1} + \frac{1-\varepsilon_2}{\varepsilon_2}\left(\frac{r_1}{r_2}\right)} = -q_2''\frac{r_2}{r_1} \\ q_2'' &= \frac{-\sigma\left(T_1^4 - T_2^4\right)}{\frac{1}{\varepsilon_1}\frac{r_2}{r_1} + \frac{1-\varepsilon_2}{\varepsilon_2}} \end{aligned} \tag{3.18}$$

The effects of radiation heat flux on wall surfaces are implemented as an additional term in the residual calculation at each boundary surface node:

$$\Delta r = \int_\Omega \psi \cdot q'' d\Omega \tag{3.19}$$

4 Fluid and Solid Property Model

4.1 Fluid Properties and Equation-of-State

4.1.1 Equation-of-State Modeling in SAM

The fluid equation of state (EOS) model is required to complete the flow governing equations, which are based on the primitive variable formulation. The dependency of fluid properties and their partial derivatives on the state variables (pressure and temperature) are implemented in the EOS model. The following functions/correlations are needed in the SAM EOS model for each fluid.

$$\begin{aligned}
&\rho=\rho\left(p,T\right),\ \frac{\partial\rho}{\partial T}=\xi\left(p,T\right),\ \frac{\partial\rho}{\partial p}=\eta\left(p,T\right)\\
&\beta=\beta\left(p,T\right)=-\frac{1}{\rho}\frac{\partial\rho}{\partial T}\\
&\mu=\mu\left(p,T\right)\\
&k=k\left(p,T\right)\\
&\mathrm{Pr}=\frac{C_p}{\mu k}\\
&H=H\left(p,T\right)\text{or}\,H\left(T\right),\quad T=T\left(H\right)\\
&C_p=C_p\left(p,T\right),\ C_v=C_v\left(p,T\right),\ \frac{\partial C_p}{\partial T}=f\left(p,T\right)
\end{aligned}\tag{4.1}$$

In which:

ρ: density;
β: thermal expansion coefficient;
μ: dynamic viscosity;
Pr: Prandtl number;
H: enthalpy;
C_p: the specific heat at constant pressure;
C_v: the specific heat at constant volume.

Note that the $\frac{\partial\rho}{\partial t}$ and $\frac{\partial\rho}{\partial z}$ terms are needed in the fluid governing equations. They can be obtained using the chain rule:

$$\begin{aligned}
\frac{\partial\rho}{\partial t}&=\xi\frac{\partial T}{\partial t}+\eta\frac{\partial p}{\partial t}\\
\frac{\partial\rho}{\partial z}&=\xi\frac{\partial T}{\partial z}+\eta\frac{\partial p}{\partial z}
\end{aligned}\tag{4.2}$$

And for incompressible but thermally expandable flow, $\frac{\partial\rho}{\partial p}=\eta\left(p,T\right)=0$. Therefore, we have:

$$\begin{aligned}
\frac{\partial\rho}{\partial t}&=\xi\frac{\partial T}{\partial t}\\
\frac{\partial\rho}{\partial z}&=\xi\frac{\partial T}{\partial z}
\end{aligned}\tag{4.3}$$

4.1.2 Liquid Sodium Property Modeling

The liquid sodium property model in SAS4A/SASSYS-1 (Fanning, 2012) is applied in SAM, in which most properties are only dependent on fluid temperature, T in [K].

Liquid Density (ρ_l in [kg/m^3])

$$\begin{aligned}\rho_l &= A_{12} + A_{13}T + A_{14}T^2 \\ \frac{\partial \rho_l}{\partial T} &= A_{13} + 2A_{14}T \\ A_{12} &= 1.00423 \times 10^3 \\ A_{13} &= -0.21390 \\ A_{14} &= -1.1046 \times 10^{-5}\end{aligned} \tag{4.4}$$

Liquid Heat Capacity (C_l in [J/(kg·K)])

$$\begin{aligned}C_l &= \frac{A_{28}}{(T_c - T)^2} + \frac{A_{29}}{T_c - T} + A_{30} + A_{31}(T_c - T) + A_{32}(T_c - T)^2 \\ C_p &= C_v = C_l \\ \frac{\partial C_p}{\partial T} &= \frac{2A_{28}}{(T_c - T)^3} + \frac{A_{29}}{(T_c - T)^2} - A_{31} - 2A_{32}(T_c - T) \\ A_{28} &= 7.3898 \times 10^5 \\ A_{29} &= 3.154 \times 10^5 \\ A_{30} &= 1.1340 \times 10^3 \\ A_{31} &= -2.2153 \times 10^{-1} \\ A_{32} &= 1.1156 \times 10^{-4} \\ T_c &= 2503.3K, \text{the critical temperature}\end{aligned} \tag{4.5}$$

Liquid Specific Enthalpy (h in [J/kg])

The specific enthalpy of liquid sodium is obtained by integrating the heat capacity over temperature, i.e.,

$$h = h_0 + \int_{T_0}^{T} C_p(T)dT$$

in which, $T_0 = 628.15$K and h_0 is set to be $h_0 = C_p(T_0) \times T_0$. The specific enthalpy can be then expressed as,

$$h = h_0 + F(T) - F(T_0)$$

with

$$F(T) = \frac{A_{28}}{T_c - T} - A_{29}\ln(T_c - T) - A_{30}(T_c - T) - \frac{A_{31}}{2}(T_c - T)^2 - \frac{A_{32}}{3}(T_c - T)^3 .$$

Liquid Thermal Expansion Coefficient (β_l in [K^{-1}])

$$
\begin{aligned}
\beta_l &= A_{42} + \frac{A_{43}}{T_c - T} + \frac{A_{44}}{(T_c - T)^2} + \frac{A_{45}}{(T_c - T)^3} + \frac{A_{46}}{(T_c - T)^4} + \frac{A_{47}}{(T_c - T)^5} \\
A_{42} &= 2.5156 \times 10^{-6} \\
A_{43} &= 0.79919 \\
A_{44} &= -6.9716 \times 10^{2} \\
A_{45} &= 3.3140 \times 10^{5} \\
A_{46} &= -7.0502 \times 10^{7} \\
A_{47} &= 5.4920 \times 10^{9}
\end{aligned}
\tag{4.6}
$$

Liquid Thermal Conductivity (k_l in [W/(m·K)])

$$
\begin{aligned}
k_l &= A_{48} + A_{49} T + A_{50} T^2 + A_{51} T^3 \\
A_{48} &= 1.1045 \times 10^{2} \\
A_{49} &= -6.5112 \times 10^{-2} \\
A_{50} &= 1.5430 \times 10^{-5} \\
A_{51} &= -2.46170 \times 10^{-9}
\end{aligned}
\tag{4.7}
$$

Liquid Viscosity (μ_l in [Pa·s])

$$
\begin{aligned}
\mu_l &= A_{52} + \frac{A_{53}}{T} + \frac{A_{54}}{T^2} + \frac{A_{55}}{T^3} \\
A_{52} &= 3.6522 \times 10^{-5} \\
A_{53} &= 0.16626 \\
A_{54} &= -4.56877 \times 10^{1} \\
A_{55} &= 2.87330 \times 10^{4}
\end{aligned}
\tag{4.8}
$$

4.1.3 Salt Property Modeling

Simplified salt property models developed by UC Berkeley (Peterson et al., 2013) are implemented in SAM, in which most properties are only dependent on temperature. Three different types of salts are implemented, including FLiBe, FLiNaK, and Dowtherm A (a type of simulant oil for salt).

FLiBe (temperature in [°C]):

$$
\begin{aligned}
\rho[kg/m^3] &= 2279.92 - 0.488T \\
\mu[kg/m \cdot \mathrm{s}] &= \frac{4.638 \times 10^5}{T^{2.79}} \\
C_p[J/kg \cdot \mathrm{K}] &= 2415.78 \\
h[J/kg] &= C_p T \\
k[W/m \cdot \mathrm{K}] &= 0.7662 + 0.0005T
\end{aligned}
\tag{4.9}
$$

FLiNaK (temperature in K):

$$
\begin{aligned}
\rho[kg/m^3] &= 2729.3 - 0.73T \ , \quad 940 < T < 1170 \\
\mu[kg/m\cdot \mathrm{s}] &= 2.487 \times 10^{-4} \cdot e^{\frac{4478.62}{T}} \ , \quad 770 < T < 970 \\
C_p[J/kg\cdot \mathrm{K}] &= 1905.57 \\
h[J/kg] &= C_p T \\
k[W/m\cdot \mathrm{K}] &= 0.36 + 5.6 \times 10^{-4} T \ , \quad 790 < T < 1080
\end{aligned}
\tag{4.10}
$$

Dowtherm A (temperature in °C):

$$
\begin{aligned}
\rho[kg/m^3] &= 1078 - 0.85T \\
\mu[kg/m\cdot \mathrm{s}] &= \frac{0.130}{T^{1.072}} \\
C_p[J/kg\cdot \mathrm{K}] &= 1518 + 2.82T \\
h[J/kg] &= h_0 + 1518(T - T_0) + 1.41(T^2 - T_0^2) \\
k[W/m\cdot \mathrm{K}] &= 0.142 - 0.00016T
\end{aligned}
\tag{4.11}
$$

with $T_0 = 20°\mathrm{C}$, and $h_0 = C_p(T_0) \times (T_0 + 273.15)$.

For the unspecified but needed properties, the following models are used in SAM.

$$
\begin{aligned}
\frac{\partial \rho}{\partial T} &= \frac{\rho\,(p, T+\varepsilon) - \rho\,(p, T-\varepsilon)}{2\varepsilon}, \quad \varepsilon = 0.5 \\
\frac{\partial \rho}{\partial p} &= 0 \\
\beta &= -\frac{1}{\rho}\frac{\partial \rho}{\partial T} \\
\frac{\partial C_p}{\partial T} &= \frac{C_p\,(p, T+\varepsilon) - C_p\,(p, T-\varepsilon)}{2\varepsilon}, \quad \varepsilon = 0.5 \\
H &= H_0 + \int_{T_0}^{T} C_p dT
\end{aligned}
\tag{4.12}
$$

in which T_0 is the reference temperature and H_0 is the fluid enthalpy at the reference temperature.

4.1.4 Pure Salt Property Modeling

In the Molten Salt Thermophysical Properties Database (Jerden (2019)) under development, temperature dependent correlations for density, viscosity, thermal conductivity, and heat capacity are presented. These correlations are proposed for 27 different pure salts. Out of these 27 salts, 10 of these correlations were complete enough to be included in SAM. The 10 salts and their thermophysical correlations are shown below:

LiF (temperature in K):

$$\begin{aligned} \rho[kg/m^3] &= 1000(2.4-0.00049T) \\ \mu[kg/m\cdot \mathrm{s}] &= 0.001(0.18e^{\frac{22000}{8.3145T}}) \\ C_p[J/kg\cdot \mathrm{K}] &= 2467.289 \\ k[W/m\cdot \mathrm{K}] &= 1.4 \end{aligned} \tag{4.13}$$

NaF (temperature in K):

$$\begin{aligned} \rho[kg/m^3] &= 1000(2.8-0.00064T) \\ \mu[kg/m\cdot \mathrm{s}] &= 0.001(0.12e^{\frac{26000}{8.3145T}}) \\ C_p[J/kg\cdot \mathrm{K}] &= 1667.13 \\ k[W/m\cdot \mathrm{K}] &= 0.9 \end{aligned} \tag{4.14}$$

KF (temperature in K):

$$\begin{aligned} \rho[kg/m^3] &= 1000(2.6-0.00065T) \\ \mu[kg/m\cdot \mathrm{s}] &= 0.001(0.11e^{\frac{24000}{8.3145T}}) \\ C_p[J/kg\cdot \mathrm{K}] &= 1239.313 \\ k[W/m\cdot \mathrm{K}] &= 0.57 \end{aligned} \tag{4.15}$$

BeF_2 (temperature in K):

$$\begin{aligned} \rho[kg/m^3] &= 1000(2.0-0.000015T) \\ \mu[kg/m\cdot \mathrm{s}] &= 0.001(3\times 10^{-7}e^{\frac{240000}{8.3145T}}) \\ C_p[J/kg\cdot \mathrm{K}] &= \frac{10-0.0015T-1.6\times 10^{7}T^{-2}+3.0\times 10^{-9}T^{2}}{0.047} \\ k[W/m\cdot \mathrm{K}] &= 0.8 \end{aligned} \tag{4.16}$$

LiCl (temperature in K):

$$\begin{aligned} \rho[kg/m^3] &= 1000(1.9-0.00043T) \\ \mu[kg/m\cdot \mathrm{s}] &= 0.001(0.11e^{\frac{19000}{8.3145T}}) \\ C_p[J/kg\cdot \mathrm{K}] &= 1486.059 \\ k[W/m\cdot \mathrm{K}] &= 0.73 \end{aligned} \tag{4.17}$$

NaCl (temperature in K):

$$\begin{aligned} \rho[kg/m^3] &= 1000(2.1-0.00054T) \\ \mu[kg/m\cdot \mathrm{s}] &= 0.001(0.089e^{\frac{22000}{8.3145T}}) \\ C_p[J/kg\cdot \mathrm{K}] &= 1163.530 \\ k[W/m\cdot \mathrm{K}] &= 0.49 \end{aligned} \tag{4.18}$$

KCl (temperature in K):

$$\begin{aligned}\rho[kg/m^3] &= 1000(2.4-0.00049T)\\ \mu[kg/m\cdot\mathrm{s}] &= 0.001(0.062e^{\frac{25000}{8.3145T}})\\ C_p[J/kg\cdot\mathrm{K}] &= 992.605\\ k[W/m\cdot\mathrm{K}] &= 0.74\end{aligned} \tag{4.19}$$

MgCl$_2$ (temperature in K):

$$\begin{aligned}\rho[kg/m^3] &= 1000(2.0-0.00027T)\\ \mu[kg/m\cdot\mathrm{s}] &= 0.001(0.18e^{\frac{21000}{8.3145T}})\\ C_p[J/kg\cdot\mathrm{K}] &= \frac{92-2\times10^{-6}T+5.9\times10^{6}T^{-2}}{0.095211}\\ k[W/m\cdot\mathrm{K}] &= 0.2\end{aligned} \tag{4.20}$$

CaCl$_2$ (temperature in K):

$$\begin{aligned}\rho[kg/m^3] &= 1000(2.5-0.00042T)\\ \mu[kg/m\cdot\mathrm{s}] &= 0.001(0.1e^{\frac{30000}{8.3145T}})\\ C_p[J/kg\cdot\mathrm{K}] &= 901.031\\ k[W/m\cdot\mathrm{K}] &= 0.45\end{aligned} \tag{4.21}$$

SrCl$_2$ (temperature in K):

$$\begin{aligned}\rho[kg/m^3] &= 1000(3.4-0.00058T)\\ \mu[kg/m\cdot\mathrm{s}] &= 0.001(0.096e^{\frac{35000}{8.3145T}})\\ C_p[J/kg\cdot\mathrm{K}] &= 630.811\\ k[W/m\cdot\mathrm{K}] &= 0.38\end{aligned} \tag{4.22}$$

4.1.5 Mixed Salt Property Modeling

In the Molten Salt Thermophysical Properties Database, there are temperature dependent correlations for density, viscosity, thermal conductivity, and heat capacity for different salt mixtures. Correlations are set up for 36 different mixed salts, although almost all of them are incomplete. Future work will be needed to have more complete correlations. Of the mixed salts, it was determined that one (LiF-NaF-KF) was complete enough to be included in SAM. For this mixed salt, the density correlation has an uncertainty of 2%, the viscosity correlation has an uncertainty of 2%, the heat capacity correlation has an uncertainty of 10%, and the thermal conductivity correlation has an uncertainty of 5%. Note the LiF-NaF-KF correlations in the database are slightly different from those of FLiNaK that was previously implemented in SAM. Both set of models are available in SAM now.

$\underline{\mathrm{LiF}-\mathrm{NaF}-\mathrm{KF}}$ (temperature in K):
Composition (Mole %): 0.465-0.115-0.42

$$\begin{aligned}
\rho[kg/m^3] &= 1000(2.6-0.00062T) \quad &&, \ 940 < T < 1170 \\
\mu[kg/m\cdot \mathrm{s}] &= 0.001(0.025e^{\frac{4500}{8.3145T}}) \quad &&, \ 770 < T < 970 \\
C_p[J/kg\cdot \mathrm{K}] &= \frac{980+1.1T}{0.126} && \\
k[W/m\cdot \mathrm{K}] &= 0.36-0.00056T \quad &&, \ 790 < T < 1080
\end{aligned} \tag{4.23}$$

4.1.6 MSRE Fuel Property Modeling

Fission product data for numerous and varied samples taken during the Molten Salt Reactor Experiment (MSRE) was compiled and reported in an MSRE report (Compere et al. (1975)). This includes physical properties of the MSRE fuel salt used in the experiment. The temperature dependent models of the MSRE fuel salt that were reported are implemented into SAM. The MSRE fuel salt contains LiF, BeF2, ZrF4, UF4, Cr, Fe, Ni. For this MSRE fuel, the temperature correlations that are currently implemented in SAM are shown below:

$$\begin{aligned}
\rho[kg/m^3] &= 1000(2.575-0.000513(T-273.15)) \quad &&, \ \pm 1\% \text{ Precision} \\
\mu[kg/m\cdot \mathrm{s}] &= 0.001(0.116e^{\frac{3755}{T}}) &&, \ \pm 7\% \text{ Precision} \\
C_p[J/kg\cdot \mathrm{K}] &= 2386.47 &&, \ \pm 3\% \text{ Precision} \\
k[W/m\cdot \mathrm{K}] &= 0.01 &&, \ \pm 10\% \text{ Precision}
\end{aligned} \tag{4.24}$$

4.2 Liquid Metal Property Modeling

The Handbook on Lead-bismuth Eutectic Alloy and Lead Properties, Materials Compatibility, Thermal-hydraulics and Technologies (OECD/NEA (2015)) contains comprehensive temperature dependent property models on density, viscosity, heat capacity, thermal conductivity, and enthalpy. The recommended temperature dependent correlations of lead, bismuth, and LBE are implemented in SAM. The relative precision and uncertainty vary depending on which other available models these correlations are compared to. For density, the deviation of the lead model does not exceed 2%, the deviation of the bismuth model does not exceed 1.3%, and the deviation of lead-bismuth eutectic does not exceed 0.8%. For viscosity, the estimated uncertainty of the lead model is 5%, the deviation of the bismuth model does not exceed 5%, and the deviation of lead-bismuth eutectic does not exceed 8%. For heat capacity, the maximum deviation from accepted models is 10%, the deviation of the bismuth model does not exceed 2%, and the deviation of lead-bismuth eutectic does not exceed 5%. For thermal conductivity, the estimated uncertainty is 15% for lead, 15%, for bismuth, and the deviation of lead-bismuth eutectic from accepted models does not exceed 15%. The correlations implemented in SAM are shown below:

Lead (temperature in K):

$$
\begin{aligned}
\rho[kg/m^3] &= 11441 - 1.2795T \\
\mu[kg/m\cdot \mathrm{s}] &= 4.55\times 10^{-4} e^{\frac{1069}{T}} && ,\ 600.6 < T < 1473 \\
C_p[J/kg\cdot \mathrm{K}] &= 176.2 - 4.923\times 10^{-2}T + 1.544\times 10^{-5}T^2 - 1.524\times 10^{6}T^{-2} && ,\ 600.6 < T < 2021 \\
k[W/m\cdot \mathrm{K}] &= 9.2 + 0.011T && ,\ 600.6 < T < 1300 \\
\Delta h[J/kg] &= \begin{array}{c} 176.2(T-T_m) - 2.4616\times 10^{-2}(T^2-T_m^2)+ \\ 5.147\times 10^{-6}(T^3-T_m^3)+ \\ 1.524\times 10^{6}(T^{-1}-T_m^{-1}) \end{array}
\end{aligned}
\tag{4.25}
$$

Bismuth (temperature in K):

$$
\begin{aligned}
\rho[kg/m^3] &= 10725 - 1.22T \\
\mu[kg/m\cdot \mathrm{s}] &= 4.456\times 10^{-4} e^{\frac{780}{T}} && ,\ 544.6 < T < 1300 \\
C_p[J/kg\cdot \mathrm{K}] &= 118.2 + 5.934\times 10^{-3}T + 7.183\times 10^{6}T^{-2} && ,\ 544.6 < T < 1831 \\
k[W/m\cdot \mathrm{K}] &= 7.34 + 9.5\times 10^{-3}T && ,\ 544.6 < T < 1000 \\
\Delta h[J/kg] &= \begin{array}{c} 118.2(T-T_m) - 2.967\times 10^{-3}(T^2-T_m^2)- \\ 7.183\times 10^{6}(T^{-1}-T_m^{-1}) \end{array}
\end{aligned}
\tag{4.26}
$$

Lead – Bismuth Eutectic (temperature in K):

$$
\begin{aligned}
\rho[kg/m^3] &= 11065 - 1.293T \\
\mu[kg/m\cdot \mathrm{s}] &= 4.94\times 10^{-4} e^{\frac{754.1}{T}} && ,\ 400 < T < 1173 \\
C_p[J/kg\cdot \mathrm{K}] &= 164.8 - 3.94\times 10^{-2}T + 1.25\times 10^{-5}T^2 - 4.56\times 10^{5}T^{-2} && ,\ 400 < T < 1100 \\
k[W/m\cdot \mathrm{K}] &= 3.284 + 1.617\times 10^{-2}T - 2.305\times 10^{-6}T^2 \\
\Delta h[J/kg] &= \begin{array}{c} 164.8(T-T_m) - 1.97\times 10^{-2}(T^2-T_m^2)+ \\ 4.167\times 10^{-6}(T^3-T_m^3) + 4.56\times 10^{5}(T^{-1}-T_m^{-1}) \end{array}
\end{aligned}
\tag{4.27}
$$

4.2.1 Other Built-in EOS Models

Other built-in EOS models in SAM include:

1. Simple air EOS model *AirEquationOfState*, in which tabulate air properties (The Engineering Toolbox, 2017) including density, specific heat, thermal conductivity, expansion coefficient, kinematic viscosity with temperature ranging from -150 °C to 400 °C are implemented.

2. Constant EOS model, *PTConstantEOS*, in which all properties except density are constant user-specified input values. Density is then a linear function of temperature using the provided thermal expansion coefficient.

3. *FluidPropertiesEOS*, in which the fluid properties are directly from *FluidProperties* classes. It can use all the fluid property models (such as ideal gas, water, methane, etc.) implemented in the MOOSE Fluid Properties Module. *FunctionFluidProperties* is also implemented in SAM, in which all fluid properties can be provided in function or table formats in the input file. This gives users the flexibilities to use additional property models without modifying the SAM source codes. This is particular useful for potential MSR/FHR or other reactor developers, as the salt fluid properties may not be widely known and the reactor developer can supply their proprietary properties in the input model.

4.3 Solid Properties

SAM provides users with the option to manually input temperature-dependent solid property models that can be used in simulation. This user option is maintained, but a material library containing correlations for solid properties was implemented. This library will be expanded in the future to provide improved user experience.

4.3.1 Clad Property Modeling

Currently, two different types of stainless steels, HT9 and D9, are implemented in SAM. Correlations for specific heat, thermal expansion, density and thermal conductivity were taken from references Banerjee et al. (2007), Leibowitz and Blomquist (1988), and Yamanouchi et al. (1992). These correlations are shown below.

HT9 (temperature in K):

$$\begin{aligned}
C_p[J/kg\cdot \mathrm{K}] &= \frac{1}{6}(T-227)+500 && , T<527^oC \\
&= \frac{3}{5}(T-527)+550 && , T>527^oC \\
\alpha[1/K] &= \frac{1}{100}\begin{pmatrix} -0.2191+5.678\times 10^{-4}T+ \\ 8.111\times 10^{-7}T^2-2.576\times 10^{-10}T^3 \end{pmatrix} \\
\rho[kg/m^3] &= \frac{7800}{1+3\alpha} \\
k[W/m\cdot \mathrm{K}] &= 17.622+2.428\times 10^{-2}T-1.696\times 10^{-5}T^2 && , 500K<T<1030K \\
&= 12.027+1.218\times 10^{-2}T && , 1030K<T<1200K
\end{aligned} \tag{4.28}$$

D9 (temperature in K):

$$\begin{aligned}
C_p[J/kg\cdot \mathrm{K}] &= 431+17.7\times 10^{-2}T+8.72\times 10^{-5}/T^2 \\
\alpha[1/K] &= \frac{1}{100}\begin{pmatrix} -0.4247+1.282\times 10^{-3}T+ \\ 7.362\times 10^{-7}T^2-2.069\times 10^{-10}T^3 \end{pmatrix} \\
\rho[kg/m^3] &= \frac{7800}{1+3\alpha} \\
k[W/m\cdot \mathrm{K}] &= 7.598+2.391\times 10^{-2}T-8.899\times 10^{-6}T^2 && , 500K<T<1030K \\
&= 7.260+1.509\times 10^{-2}T && , 1030K<T<1200K
\end{aligned} \tag{4.29}$$

4.3.2 Fuel Property Modeling

Thermal physical properties of a few fuel types, i.e. UO2, MOX, U-Pu-Zr, are also implemented in SAM for user friendliness. The irradiated thermal conductivity for UO2 and MOX are also provided. The correlations taken from references (Karahan (2017, 2009); Savage (1968); Maeda and Asaga (2004); Billone (1986); Popov et al. (2000); Fanning (2017)) were combined to allow for complete density, thermal conductivity, and heat capacity correlations. In the table below, the coefficients used for the heat capacity calculations of UO2 and MOX are provided.

Table 4.1: Constants used in heat capacity correlations for UO_2 and MOX.

Constant	UO_2	MOX	Units
C_1	302.27	322.49	J/kg/K
C_2	8.463×10^{-3}	1.4679×10^{-2}	J/kg/K^2
C_3	8.741×10^{7}	0	J/kg
θ	548.68	587.41	K
E_a	18531.7	0	K

UO_2 (temperature in K):

$$\begin{aligned}
\rho_f[kg/m^3] &= \frac{10970}{1+2.04\times10^{-5}(T-273)+8.79\times10^{-2}(T-273)^2} && , \; T < T_{solidus} \\
&= \frac{\rho_{f-solidus}}{1+0.07338\dfrac{T-T_{sol}}{T_{liq}-T_{sol}}} && , \; T_{liquidus} < T < T_{solidus} \\
&= \frac{\rho_{f-liquidus}}{1+\dfrac{0.9285(T-T_{liq})}{8860-0.9285(T-T_{liq})}} && , \; T > T_{liquidus}
\end{aligned}$$

$$C_p[J/kg\cdot \mathrm{K}] = \frac{C_1(\theta/T)^2 e^{\theta/T}}{(e^{\theta/T}-1)^2} + 2C_2T + C_3E_a e^{-E_a/T}T^2$$

$$k_0[W/m\cdot \mathrm{K}] = \begin{array}{r} (0.06059+0.2754\sqrt{2-\dfrac{O}{M}}+2.011\times10^{-4}T)^{-1}+ \\ \dfrac{4.715\times10^{9}}{T^2}e^{\frac{-16361}{T}}+0.2 \end{array} \tag{4.30}$$

MOX (temperature in K):

$$
\begin{aligned}
\rho_f[kg/m^3] &= \frac{11460}{1+2.04\times10^{-5}(T-273)+8.79\times10^{-2}(T-273)^2} && ,\ T < T_{solidus} \\
&= \frac{\rho_{f-solidus}}{1+0.07338\dfrac{T-T_{sol}}{T_{liq}-T_{sol}}} && ,\ T_{liquidus} < T < T_{solidus} \\
&= \frac{\rho_{f-liquidus}}{1+\dfrac{0.9285(T-T_{liq})}{8860-0.9285(T-T_{liq})}} && ,\ T > T_{liquidus} \\
C_p[J/kg\cdot\mathrm{K}] &= \frac{C_1(\theta/T)^2 e^{\theta/T}}{(e^{\theta/T}-1)^2} + 2C_2T + C_3E_a e^{-E_a/T}T^2 \\
k_0[W/m\cdot\mathrm{K}] &= \begin{aligned}&(0.06059+0.2754\sqrt{2-\frac{O}{M}}+2.011\times10^{-4}T)^{-1}+ \\ &\qquad\qquad\qquad \frac{4.715\times10^9}{T^2}e^{\frac{-16361}{T}}\end{aligned}
\end{aligned}
\tag{4.31}
$$

$\mathrm{U-Pu-Zr}$ (temperature in K):

$$
\begin{aligned}
\rho[kg/m^3] &= \left(\frac{W_{Zr}}{6500}+\frac{WUPu}{19000}\right)^{-1}\frac{1-P-0.015Bu}{1+6\times10^{-5}(T-293.15)} \\
C_{p1}[J/kg\cdot\mathrm{K}] &= 26.58+\frac{0.027}{Molar\ Mass}T && ,\ T < 868.15K = T_1 \\
C_{p2}[J/kg\cdot\mathrm{K}] &= 15.84+\frac{0.026}{Molar\ Mass}T && ,\ 868.15K < T < 923.15K = T_2 \\
C_{p3}[J/kg\cdot\mathrm{K}] &= c_{p1}+\frac{c_{p2}-c_{p1}}{T_2-T_1}(T-T_1) && ,\ T > 923.15K \\
k[W/m\cdot\mathrm{K}] &= a+bT+cT^2 \\
a &= 17.5\left(\frac{1-2.23W_{Zr}}{1+1.61W_{Zr}}-2.62W_{Pu}\right) \\
b &= 1.54\times10^{-2}\left(\frac{1+0.061W_{Zr}}{1+1.61W_{Zr}}+0.9W_{Pu}\right) \\
c &= 9.38\times10^{-6}(1-2.7W_{Pu})
\end{aligned}
\tag{4.32}
$$

Irradiated UO_2 (temperature in K):

$$k[W/m\cdot \mathrm{K}] = k_0 F_1 F_2 F_3 F_4$$

$$\begin{aligned}
F_1 &= \left(\frac{1.09}{Bu^{3.265}} + \frac{0.0643}{Bu^{0.5}}T^{0.5}\right)\arctan\left(\left(\frac{1.09}{Bu^{3.265}} + \frac{0.0643}{Bu^{0.5}T^{0.5}}\right)^{-1}\right) \\
F_2 &= 1 + \frac{0.019Bu}{(3-0.019Bu)(1+e^{\frac{T-1200}{-100}})} \\
F_3 &= 1 - \frac{0.2}{1+e^{\frac{T-900}{80}}} \\
F_4 &= 1-2.5P \qquad , \ P < 0.1 \\
&= \frac{1-P}{1+2P} \qquad , \ P \geq 0.1
\end{aligned} \tag{4.33}$$

Irradiated MOX (temperature in K):

$$k[W/m\cdot \mathrm{K}] = k_0 F_1 F_2 F_3 F_4$$

$$\begin{aligned}
F_1 &= \left(\frac{1.09}{Bu^{3.265}} + \frac{0.0643}{Bu^{0.5}}T^{0.5}\right)\arctan\left(\left(\frac{1.09}{Bu^{3.265}} + \frac{0.0643}{Bu^{0.5}T^{0.5}}\right)^{-1}\right) \\
F_2 &= 1 + \frac{0.019Bu}{(3-0.019Bu)(1+e^{\frac{T-1200}{-100}})} \\
F_3 &= 1 - \frac{0.2}{1+e^{\frac{T-900}{80}}} \\
F_4 &= 1-2.5P \qquad , \ P < 0.1 \\
&= \frac{1-P}{1+2P} \qquad , \ P \geq 0.1
\end{aligned} \tag{4.34}$$

5 Closure Models

5.1 Convective Heat Transfer Correlations

5.1.1 Convective heat transfer modeling options in SAM

The closure models selected in SAM have been largely based on the practices of existing nuclear reactor system codes with an emphasis on considering the effects of low-Prandtl number fluids used in liquid-metal cooled fast reactor concepts. From extensive review of closure models used in other system codes and from more recently developed correlations in the literature, a subset of the convective heat transfer correlations have been implemented in SAM as available user options, as well as the default correlation selected based on the geometry and flow conditions. The considerations behind the selections include:

- Fluid type (focused on liquid-metal) and flow geometry,
- Applicable flow regime and geometric ranges,
- Uncertainty of correlations,
- Model complexity / user-accessibility,
- Popularity amongst system code users,
- State-of-the-art investigations/reviews.

The overall structure of the heat transfer model selection in SAM is shown in Figure 5.1. The user is able to select correlations for an individual component by specifying it within the component input block. The default options are listed in Figure 5.1 in bold color font. If the user does not specify a correlation, SAM will determine which correlation to use based on the fluid type (Prandtl number) and the specified flow geometry (pipe or rod bundle). A summary of the implemented heat transfer correlations is listed in Table 4-1. Many of these correlations are also implemented in one or more existing system codes including SAS4A/SASSYS-1 (Fanning, 2012), RELAP5-3D (INL, 2012), TRACE (NRC, 2010), ATHENA (Davis and Shieh, 2000), MARS-LMR (Ha et al., 2010), ANTEO+ (Lodi et al., 2016), ATHLET (Chen and Cheng, 2005), CATHARE (Polidori, 2010), etc.

5.1.2 Low-Prandtl Fluids

The heat transfer characteristics of fluids are highly dependent on the Prandtl number, for which low-Prandtl fluids ($\mathrm{Pr} < 0.1$) require a separate set of correlations. The following sub-sections outline the recommended heat transfer correlations for several flow geometries, determined by survey of use in existing system thermal hydraulic codes.

5.1.2.1 Pipe flow-circular tube forced convection

The default model for pipe flow is the **Seban-Shimazaki** correlation (Seban and Shimazaki, 1951). This correlation is based on work done by Lyon (1949), but developed for liquid flow inside a tube with a constant wall temperature boundary condition. It is widely implemented for liquid metal

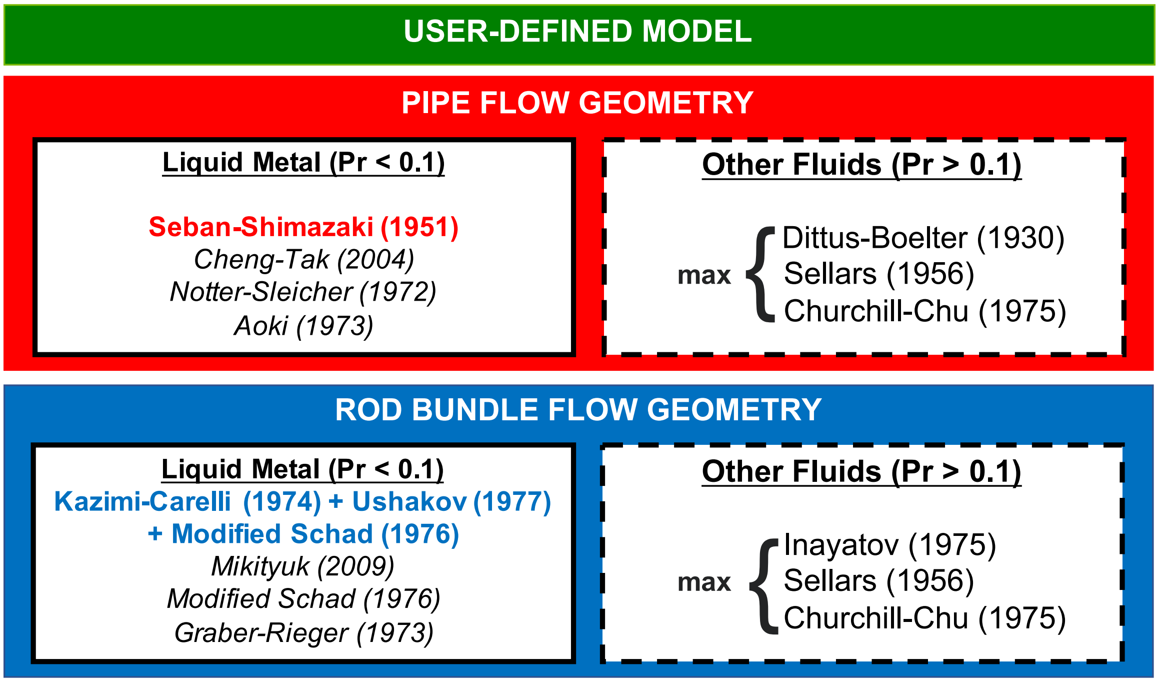


Figure 5.1: SAM modeling options for convective heat transfer

flows across a majority of system thermal hydraulic codes. The correlation is often recommended based on its simple form and universal applicability, and is commonly used amongst system codes. However, it must be noted that this existing Seban-Shimizaki correlation was based on a highly idealized case, with a relatively large range of uncertainty to the original data.

$$\begin{aligned} &\mathrm{Nu} = 5.0 + 0.025\mathrm{Pe}^{0.8}, \qquad \text{if } \mathrm{Pe} > 100 \\ &\text{Valid:} 10^2 \leq \mathrm{Pe} \leq 2 \times 10^4 \\ &\text{Uncertainty: N.R. (No Reference)} \end{aligned} \tag{5.1}$$

The **Notter-Sleicher** correlation (Notter and Sleicher, 1972) is a recommended option for pipe flow. The work of Notter-Sleicher proposed a more rigorous set of equations that encapsulate the full range of Prandtl fluids (with different correlations for liquid metals and non liquid metals),with a well-defined uncertainty. The numerical results of the original study (Notter and Sleicher, 1972) were shown to be in good agreement with experimental data on fully developed heat transfer rates for Prandtl numbers between 0.01 and 10^4.

$$\begin{aligned} &\mathrm{Nu} = 4.8 + 0.0156\mathrm{Pe}^{0.85}\mathrm{Pr}^{0.08} \\ &\text{Valid:}10^4 \leq \mathrm{Re} \leq 10^6, \quad 0.004 \leq \mathrm{Pr} \leq 0.1 \\ &\text{Uncertainty:} \pm 5\% \end{aligned} \tag{5.2a}$$

$$\begin{aligned} &\mathrm{Nu} = 5 + 0.016\mathrm{Re}^a\mathrm{Pr}^b \\ &a = 0.88 - \frac{0.24}{(4+\mathrm{Pr})}, \quad b = 0.33 + 0.5e^{-0.6\,\mathrm{Pr}} \\ &\text{Valid:}10^4 \leq \mathrm{Re} \leq 10^6, \quad 0.1 \leq \mathrm{Pr} \leq 10^4 \\ &\text{Uncertainty:} \pm 10\% \end{aligned} \tag{5.2b}$$

The **Aoki** correlation (Aoki, 1973) is suggested as another option for pipe flow. This correlation has been implemented by KAERI in their fast reactor thermal-hydraulics module MARS-LMR (Ha et al., 2010) specifically for pipe flow geometries, and has shown good results in their work on EBR-II modeling.

$$\begin{aligned} &\mathrm{Nu} = 6.0 + 0.025\left[0.014\mathrm{Re}^{1.45}\mathrm{Pr}^{1.2}\left(1 - \frac{e^{-71.8}}{\mathrm{Re}^{0.45}\mathrm{Pr}^{0.2}}\right)\right]^{0.8} \\ &\text{Valid: N.R. (No Reference)} \\ &\text{Uncertainty: N.R. (No Reference)} \end{aligned} \tag{5.3}$$

The **Cheng-Tak** correlation (Cheng and Tak, 2006) is another option for pipe flow. Although it was originally developed for lead-bismuth eutectic (LBE) flows, it is applicable to other liquid metal flow conditions as well.

$$\begin{aligned} &\mathrm{Nu} = A + 0.018\mathrm{Pe}^{0.80} \\ &A = \begin{cases} 4.5, & \mathrm{Pe} \leq 1000 \\ 5.4 - 0.0009\mathrm{Pe}, & 1000 < \mathrm{Pe} < 2000 \\ 3.6, & \mathrm{Pe} \geq 2000 \end{cases} \\ &\text{Valid:} \quad \mathrm{Pe} \leq 6 \times 10^4 \\ &\text{Uncertainty: N.R. (No Reference)} \end{aligned} \tag{5.4}$$

If the single-phase liquid flow is characterized as fully developed laminar flow (Pe < 10), then the assumption from **Sellars** correlation (Sellars et al., 1956; Incropera et al., 2007) for convection with uniform surface heat flux boundary in circular tube flow is selected as the default model.

$$\begin{aligned} &\mathrm{Nu} = \frac{48}{11} \approx 4.36 \\ &\text{Implemented if Pe} < 10 \\ &\text{Uncertainty: N.R. (No Reference)} \end{aligned} \tag{5.5}$$

5.1.2.2 Rod bundle flow-core channel forced convection

All rod bundle heat transfer correlations have additional dependencies on the bundle geometry, characterized by the pitch-to-diameter-ratio P/D. The suggested default model for bundle flow is the updated **Calamai/Kazimi-Carelli** correlation (Calamai et al., 1974). This correlation is widely implemented for liquid metal bundle flows across a majority of system thermal hydraulic codes. This correlation is simple in nature and covers a significant range, thus presenting a relatively robust option.

$$\begin{aligned}
&\mathrm{Nu} = 4 + 0.16\left(\frac{P}{D}\right)^{5} + 0.33\left(\frac{P}{D}\right)^{3.8}\left(\frac{\mathrm{Pe}}{100}\right)^{0.86} \\
&\text{Valid:}\quad 1.1 \le \frac{P}{D} \le 1.4,\ 10 \le \mathrm{Pe} \le 5\times 10^{3} \\
&\text{Uncertainty: N.R. (No Reference)}
\end{aligned} \tag{5.6}$$

Todreas and Kazimi (2012) showed that the **Kazimi-Carelli** correlation agreed well with experimental data at P/D = 1.15, but underestimated the Nusselt number at P/D = 1.30. This correlation has been predominantly implemented in most existing system codes for fast reactors.

The following correlations are implemented in the default model to account for geometries that lie outside the applicable range of this correlation. The **Ushakov** correlation (Ushakov et al., 1977; Pfrang and Struwe, 2007) extends the existing correlation given by Calamai et al. (1974). Although popularized by Westinghouse for system codes, reviews of the correlation's performance have shown to typically under-predict heat transfer for larger pitch-diameter ratios (above 1.3), and in the case of modeling EBR-II, under-predicting the Nusselt number by as much as 30%. The Ushakov correlation provides a more accurate modeling of larger pitch/diameter ratio rod bundle flows (implemented for P/D $>$ 1.4).

$$\begin{aligned}
&\mathrm{Nu} = 7.55\left(\frac{P}{D}\right) - 20\left(\frac{P}{D}\right)^{-13} + \frac{0.041}{\left(\frac{P}{D}\right)^{2}}\mathrm{Pe}^{\left(0.56+0.19\frac{P}{D}\right)} \\
&\text{Valid:}\quad 1.3 \le \frac{P}{D} \le 2.0,\ 0 \le \mathrm{Pe} \le 4\times 10^{3} \\
&\text{Uncertainty:}\quad \pm 5\%
\end{aligned} \tag{5.7}$$

The **Modified Schad** correlation, also by Kazimi and Carelli (1976), extends existing correlation given by Calamai et al. (1974). This correlation is implemented to cover cases with very tight lattice geometries (P/D $<$ 1.1). Furthermore, since the origins of this correlation come from modeling the core flows of an actual reactor design, the correlation is also presented as a user option to be used in place of the default Calamai correlation:

$$\begin{aligned}
&\text{For } 150 \le \text{Pe} \le 10^3: \\
&\qquad \text{Nu} = \left[-16.15 + 24.96\left(\frac{P}{D}\right) - 8.55\left(\frac{P}{D}\right)^2\right]\text{Pe}^{0.3} \\
&\text{For Pe} \le 150: \\
&\qquad \text{Nu} = 4.496\left[-16.15 + 24.96\left(\frac{P}{D}\right) - 8.55\left(\frac{P}{D}\right)^2\right] \\
&\text{Valid:} \quad 1.05 \le \frac{P}{D} \le 1.5, \quad \text{Pe} \le 10^3 \\
&\text{Uncertainty: N.R. (No Reference)}
\end{aligned} \tag{5.8}$$

The **Mikityuk** correlation developed in recent work Mikityuk (2009) has provided a more generalized correlation for rod bundle convective heat transfer over the complete range of pitch/diameter ratios. Although it is robust and derived with experimental data from both triangular and square rod-bundle geometries, there has not been a large body of literature to reference its usage. Nonetheless, it is suggested that this correlation be implemented as an available user option that would replace the usage of the Calamai and Ushakov correlations if desired.

$$\begin{aligned}
&\text{Nu} = 0.047\left(1 - e^{-3.8\left(\frac{P}{D}-1\right)}\right)\left(\text{Pe}^{0.77} + 250\right) \\
&\text{Valid:} \quad 1.1 \le \frac{P}{D} \le 1.95, \quad 30 \le \text{Pe} \le 5 \times 10^3 \\
&\text{Uncertainty:} \quad \pm 5\%\text{(possibly less)}
\end{aligned} \tag{5.9}$$

The **Gräber-Rieger** correlation (Gräber and Rieger, 1972) is provided as a user option for bundle flows. This correlation has been implemented and recommended in some work by KAERI (Choi and Kim, 2008) for use in describing the shell-side heat transfer in heat exchangers with rod bundles. It is anticipated that this correlation can be used in a similar manner when modeling heat exchangers.

$$\begin{aligned}
&\text{Nu} = 0.25 + 6.2\frac{P}{D} + \left[0.032\left(\frac{P}{D}\right) - 0.007\right]\text{Pe}^{\left(0.8 - 0.024\frac{P}{D}\right)} \\
&\text{Valid:} \quad 1.25 \le \frac{P}{D} \le 1.95, \quad 150 \le \text{Pe} \le 3 \times 10^3 \\
&\text{Uncertainty: N.R. (No Reference)}
\end{aligned} \tag{5.10}$$

5.1.3 Non-Liquid Metal Fluids

For single-phase non-liquid-metal flow, the **Dittus-Boelter (1930)** correlation (Todreas and Kazimi, 2012) is a classic and standard analytical correlation for single-phase heat transfer. It is sufficient to determine the convective heat transfer properties for internal pipe flow. The uncertainty of this correlation is more well defined than other existing correlations.

$$
\begin{aligned}
&\mathrm{Nu} = 0.023\mathrm{Re}^{0.8}\mathrm{Pr}^{n}; \quad n = \begin{cases} 0.4, & \text{heating} \\ 0.33, & \text{cooling} \end{cases} \\
&\text{Valid:} \quad \frac{L}{D} \geq 10,\ \mathrm{Re} \geq 10^{5}(\text{ turbulent}),\ 0.6 \leq \mathrm{Pr} \leq 160 \\
&\text{Uncertainty:} \quad \pm 15\%
\end{aligned}
\tag{5.11}
$$

The adjustment to the classic Dittus-Boelter as proposed by (Inayatov, 1975) to consider the rod-bundle geometric properties is also implemented in SAM. Further literature review should be performed to assess an appropriate justification of the correlation uncertainty.

$$
\begin{aligned}
&\mathrm{Nu} = 0.023\left(\frac{P}{D}\right)\mathrm{Re}^{0.8}\mathrm{Pr}^{n}; \quad n = \begin{cases} 0.4, & \text{heating} \\ 0.33, & \text{cooling} \end{cases} \\
&\text{Valid:} \quad 1.1 \leq \frac{P}{D} \leq 1.6, \quad 6\times 10^{3} \leq \mathrm{Re} \leq 10^{6}, \quad 0.6 \leq \mathrm{Pr} \leq 18 \\
&\text{Uncertainty: N.R. (No Reference)}
\end{aligned}
\tag{5.12}
$$

In the case of low flow conditions, fully developed laminar flow correlation Sellars et al. (1956), as described in Eq. (5.4) is used again for non-liquid metal fluids.

For single-phase natural convection adjacent to a vertical plane, the **Churchill-Chu** correlation (Churchill and Chu, 1975) is implemented, which is consistent with the implementation in RELAP5-3D (INL, 2012).

$$
\begin{aligned}
&\mathrm{Nu} = \left[0.825 + \frac{0.387\mathrm{Ra}^{\frac{1}{6}}}{\left(1+(0.492/\mathrm{Pr})^{\frac{9}{16}}\right)^{\frac{8}{27}}}\right]^{2} \\
&\text{Valid:} \quad \mathrm{Ra} < 10^{12} \\
&\text{Uncertainty: N.R. (No Reference)}
\end{aligned}
\tag{5.13}
$$

For single-phase natural convection adjacent to a horizontal plane (laminar flow), the **McAdams** correlation (McAdams, 1954) was chosen, in line with the implementation in RELAP5. This particular correlation is valid for the lower surface of a heated plane or the upper surface of a cooled plane.

$$
\begin{aligned}
&\mathrm{Nu} = 0.27\mathrm{Ra}^{0.25} \\
&\text{Valid:} \quad 10^{5} < \mathrm{Ra} < 10^{10} \\
&\text{Uncertainty: N.R. (No Reference)}
\end{aligned}
\tag{5.14}
$$

For non-liquid-metal fluid flow, the maximum value of the calculated heat transfer coefficient from the Dittus-Boelter (or Inayatov), Sellars, and Churchill-Chu correlations are used, consistent with the implementation in RELAP5-3D (INL, 2012).

5.1.4 User-specified heat transfer correlation option

User-specified models are also included as an option that allows for greater customization and flexibility to address any shortcomings or necessary modifications to the included models. The provided correlation below is adopted from the implementation in CATHARE (Polidori, 2010) for forced convections.

$$\mathrm{Nu} = \mathrm{Nu}_0 + a\left(\mathrm{Re}^b + c\right)\mathrm{Pr}^d\left(1 + e\mathrm{Re}^f\right)^{0.1} \tag{5.15}$$

For natural convection, a different correlation is used because of the dependence on Rayleigh number.

$$\mathrm{Nu} = \mathrm{Nu}_0 + a\mathrm{Ra}^b \tag{5.16}$$

The above heat transfer correlations implemented in SAM are summarized in Table 5.1, in which the default correlations for different geometry and flow conditions are shaded.

Table 5.1: SAM convective heat transfer models

Name	Correlation	Re	Pr	Pe	P/D	Uncert.	Notes
Circular Pipe Geometry							
Dittus-Boelter (1930)	$\mathrm{Nu} = 0.023\,\mathrm{Re}^{0.8}\mathrm{Pr}^{0.4}$	[1e5, inf)	[0.6, 160]		-	± 15%	*Default, for Pr > 0.1
Seban-Shimazaki (1951)	$\mathrm{Nu} = 5.0 + 0.025\mathrm{Pe}^{0.8}$			[1e2, 2e4]	-	N. R.	*Default, Forced, Turbulent
Sellars et al. (1956)	$\mathrm{Nu} = \frac{48}{11} \approx 4.36$			[0, 10)	-	± 10% (for He)	*Default, Forced, Laminar
Aoki (1973)	$\mathrm{Nu} = 6.0 + 0.025 \times \left[0.014\,\mathrm{Re}^{1.45}\mathrm{Pr}^{1.2} \times \left(1 - \frac{e^{-71.8}}{\mathrm{Re}^{0.45}\mathrm{Pr}^{0.2}}\right)\right]^{0.8}$				-	-	Forced, Turbulent
Cheng-Tak (2004)	$\mathrm{Nu} = A + 0.018\,\mathrm{Pe}^{0.8}$ $A = \begin{cases} 4.5, & \mathrm{Pe} \leq 1000 \\ 5.4 - 0.0009\mathrm{Pe}, & 1000 < \mathrm{Pe} \leq 2000 \\ 3.6, & \mathrm{Pe} \geq 2000 \end{cases}$			[0, 6e4]	-	-	Forced, Turbulent
Notter-Sleicher (1972)	$Nu = 4.8 + 0.0156\,\mathrm{Pe}^{0.85}\mathrm{Pr}^{0.08}$	[1e4, 1e6]	[0.004, 0.1]		-	± 5%	Forced, Turbulent
	$Nu = 5 + 0.016\,Re^{a}\,Pr^{b}$ $a = 0.88 - \frac{0.24}{(4+Pr)}, b = 0.33 + 0.5e^{-0.6Pr}$		[0.1, 1e4]		-	± 10%	
Churchill-Chu (1975)	$Nu = \left[0.825 + \frac{0.387\,Ra^{\frac{1}{6}}}{\left(1+(0.492/Pr)^{\frac{9}{16}}\right)^{\frac{8}{27}}}\right]^2$	Ra: [0, 1e12]			-	N. R.	*Default for Pr > 0.1, Vertical Nat. Conv.
USER (Forced)	$\mathrm{Nu} = \mathrm{Nu}_0 + a\left(\mathrm{Re}^b + c\right)\mathrm{Pr}^d\left(1 + e\,\mathrm{Re}^f\right)^{0.1}$	-	-	-	-	-	
USER (Nat. Circ.)	$\mathrm{Nu} = \mathrm{Nu}_0 + a\mathrm{Ra}^b$	-	-	-	-	-	
Rod Bundle Geometry							
Inayatov (1975)	$\mathrm{Nu} = 0.023\left(\frac{P}{D}\right)\mathrm{Re}^{0.8}\mathrm{Pr}^{0.4}$	[6e3, 1e6]	[0.6, 18]		[1.1, 1.6]	N. R.	*Default, for Pr > 0.1
Kazimi-Carelli (1976)	$\mathrm{Nu} = 4 + 0.16\left(\frac{P}{D}\right)^5 + 0.33\left(\frac{P}{D}\right)^{3.8}\left(\frac{\mathrm{Pe}}{100}\right)^{0.86}$			[10, 5e3]	[1.1, 1.4]	N. R.	*Default within P/D range, Forced, Turbulent
Modified Schad (1976)	$\mathrm{Nu} = \begin{cases} 4.496\left[-16.15 + 24.96\left(\frac{P}{D}\right) - 8.55\left(\frac{P}{D}\right)^2\right], \mathrm{Pe} \leq 150 \\ \left[-16.15 + 24.96\left(\frac{P}{D}\right) - 8.55\left(\frac{P}{D}\right)^2\right]\mathrm{Pe}^{0.3}, 150 \leq \mathrm{Pe} \leq 1000 \end{cases}$			[0, 1e3]	[1.05, 1.5]	-	*Default for P/D range outside Kazimi-Carelli, Forced, Turbulent
Ushakov et al. (1977)	$\mathrm{Nu} = 7.55\frac{P}{D} - 20\left(\frac{P}{D}\right)^{-13} + \frac{0.041}{(P/D)^2}\mathrm{Pe}^{\left(0.56 + 0.19\frac{P}{D}\right)}$			[0, 4e3]	[1.3, 2.0]	± 5%	*Default for P/D range outside Kazimi-Carelli, Forced, Turbulent
Mikityuk (2009)	$\mathrm{Nu} = 0.047\left(1 - e^{-3.8\left(\frac{P}{D} - 1\right)}\right)\left(\mathrm{Pe}^{0.77} + 250\right)$			[30, 5e3]	[1.1, 1.95]	± 5%	Forced, Turbulent
Graber-Rieger (1973)	$\mathrm{Nu} = 0.25 + 6.2\frac{P}{D} + \left(0.032\left(\frac{P}{D}\right) - 0.007\right)\mathrm{Pe}^{\left(0.8 - 0.024\frac{P}{D}\right)}$			[150, 3000]	[1.25, 1.95]	-	Forced, Turbulent

5.2 Wall Friction Correlations

The wall friction (drag) correlations models in SAM have been largely based on practices of existing nuclear reactor system codes such as RELAP5 (INL, 2012) and TRACE (NRC, 2010). After reviewing the closure models used in many existing system codes and the available correlations in the literature, a subset of the wall friction correlations have been implemented in SAM as available user options, as well as the default correlation based on the geometry and flow conditions.

The overall structure of wall friction model selection in SAM is shown in Figure 5.2. he user is able to select correlations for an individual component by specifying it within the component input block. If the user does not specify a correlation, SAM will determine which correlation to use based on the flow conditions (Reynolds number), specified flow geometry (pipe or rod bundle) and pitch-to-diameter (P/D) ratio if applicable. Additionally, SAM has the capability to allow the user to specify multiplier factors in the input model to the calculated laminar and turbulent friction factors (with the transition interpolation adjusted accordingly). These allow the user to 'simulate' the effects of complex geometry using existing correlations.

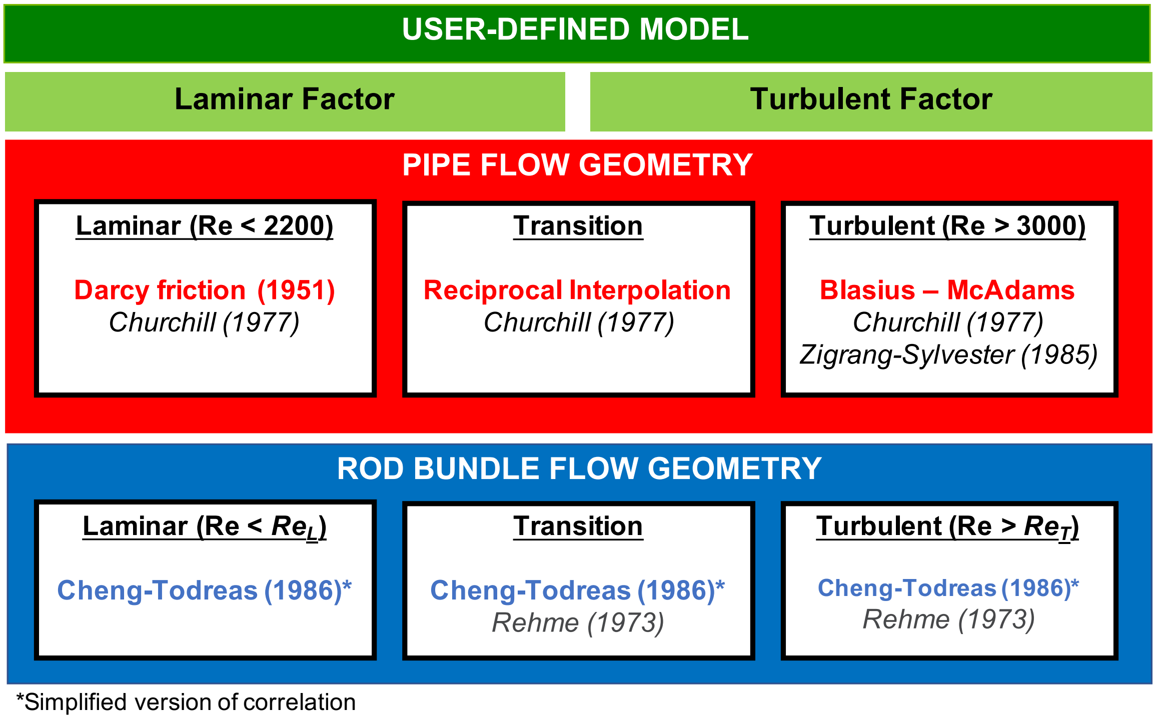


Figure 5.2: SAM modeling options for wall friction coefficient

While thermal properties are very different among different fluids (liquid-metal, salt, air, and water), hydraulic characteristics are quite similar. Therefore, most correlations that have been developed for water as the working fluid can also be used to characterize the pressure drop in advanced reactor systems. Although the effect of material consideration is not as significant, geometric features in advanced reactor designs are markedly different than those found in a typical LWR.

5.2.1 Pipe flow geometry

The **Darcy friction factor** is a dimensionless quantity used in the Darcy–Weisbach equation for the description of friction losses in pipe flow, as well as in open channel flow. It is also known as the Darcy–Weisbach friction facto or resistance coefficient and is four times larger than the Fanning friction factor. For laminar flows, it is considered valid and an exact solution to the Colebrook equation.

$$f = \frac{64}{\mathrm{Re}}, \quad \text{for Re} < 2200 \tag{5.17}$$

For turbulent flow, the **Blasius (1913)** correlation (Todreas and Kazimi, 2012) provides a seminal correlation for the friction factor within a circular pipe. The Blasius correlation is the simplest model for liquid wall friction. It gives reasonable results when applied to turbulent flow in smooth circular tubes. Since it is only a function of Reynolds number, it serves as a generic correlation that works for every fluid. It is often used in system codes:

$$f = \frac{0.316}{\mathrm{Re}^{0.25}}, \quad \text{for } 3 \times 10^3 \leq \mathrm{Re} \leq 3 \times 10^4 \tag{5.18}$$

Another simple yet common approximation for turbulent friction is the **McAdams (1942)** correlation (Todreas and Kazimi, 2012), which is valid for a specific range of the turbulent region:

$$f = 0.184\mathrm{Re}^{-0.2}, \quad \text{for } 3 \times 10^4 \leq \mathrm{Re} \leq 10^6 \tag{5.19}$$

For the transition region between the laminar and turbulent regimes, an interpolation scheme can be used for a continuous change of wall friction coefficients. The approach taken is a “reciprocal interpolation” that was also implemented in RELAP5 (INL, 2012). Weighting factors are applied in a way that smoothly averages the friction factors at the endpoints of the transition regime and interpolates depending on the Reynolds number of the flow:

$$\begin{aligned} f &= (1-w)\,f_{lam,\mathrm{Re}_{min}} + w f_{turb,\mathrm{Re}_{max}} \\ w &= \frac{N}{\mathrm{Re}_{min}} - \frac{N}{\mathrm{Re}} \\ N &= \frac{\mathrm{Re}_{min}\mathrm{Re}_{max}}{\mathrm{Re}_{max} - \mathrm{Re}_{min}} \end{aligned} \tag{5.20}$$

In which Re_{min} and Re_{max} are the lower and upper bounds of the transition region.

The default wall friction model in pipe flow depends on the flow regimes: laminar flow, turbulent flow, and a transition region between the two. Corresponding to most popular definitions, the transition region lies in the range of Re: 2200-3000, where laminar flow is expected for Re < 2200 and turbulent flow is expected for Re > 3000. The entire flow regime and the wall friction model is organized as follows:

Table 5.2: Default SAM wall friction models for internal pipe flow

Regime	Correlation	Re (min)	Re (max)
Minimum	f=1	0	64
Laminar	Darcy	64	2200
Transition	Recip. Interpolation	2200	3000
Turbulent	Blasius; McAdams	3000	3×10^4

The standard definition of wall drag in a smooth circular tube is described by the Colebrook-White equation, which is transcendental in form and not directly solvable. Therefore, two popular correlations that approximate the Colebrook-White friction factor equation have also been implemented in SAM accordingly as user options.

Churchill (1977): This correlation covers the entire flow regime range without need for interpolation, providing a more robust capability. The Churchill correlation gives satisfactory accuracy to approximating the Colebrook-White equation.

$$
\begin{aligned}
& f = 8\left[\left(\frac{8}{\mathrm{Re}}\right)^{12} + \frac{1}{(A+B)^{\frac{3}{2}}}\right]^{\frac{1}{12}} \\
& A = \left[2.457\ln\frac{1}{\left(\frac{7}{\mathrm{Re}}\right)^{0.9} + \frac{0.27\varepsilon}{D}}\right]^{16}, \quad B = \left(\frac{37530}{\mathrm{Re}}\right)^{16} \\
& \text{Valid:} \quad \mathrm{Re} \geq 50 \\
& \text{Uncertainty:} \quad \pm 3.2\%
\end{aligned}
\tag{5.21}
$$

In which ε is the pipe roughness.

Zigrang-Sylvester (1985) (INL, 2012) is a user option for wall friction modeling in the turbulent flow regime. It is only applicable to turbulent flow, thus requires interpolation for the transition regime.

$$
\begin{aligned}
& \frac{1}{\sqrt{f}} = -2\log_{10}\left\{\frac{\varepsilon}{3.7D} + \frac{2.51}{Re}\left[1.14 - 2\log_{10}\left(\frac{\varepsilon}{D} + \frac{21.25}{\mathrm{Re}^{0.9}}\right)\right]\right\} \\
& \text{Valid:} \quad \mathrm{Re} \geq 3 \times 10^3 \\
& \text{Uncertainty:} \quad \pm 5.5\%
\end{aligned}
\tag{5.22}
$$

5.2.2 Wire-wrapped rod bundle geometry

Cheng-Todreas correlation (Cheng and Todreas, 1986): This is the default model incorporated in SAM for wire-wrapped rod bundle geometries. The correlation is widely accepted and considered as an accurate representation of wire-wrap friction pressure drop. In the fast reactor design community, this correlation has been shown to have reasonably good accuracy among the available options, and is a widely adopted model. The simplified model is currently implemented in SAM as the detailed Cheng-Todreas model is only applicable in modeling the effects in a single sub-channel region.

$$f = \begin{cases} C_{fL}/\mathrm{Re}, & \mathrm{Re} \leq \mathrm{Re}_L \\ \dfrac{C_{fL}}{\mathrm{Re}}(1-\psi)^{1/3} + \dfrac{C_{fT}}{\mathrm{Re}^{0.18}}\psi^{1/3}, & \mathrm{Re}_L < \mathrm{Re} < \mathrm{Re}_T \\ C_{fT}/\mathrm{Re}^{0.18}, & \mathrm{Re} \geq \mathrm{Re}_T \end{cases} \tag{5.23}$$

with $\mathrm{Re}_L = 300 \times 10^{1.7\left(\frac{P}{D}-1\right)}$ the laminar-transition Reynolds number, $\mathrm{Re}_T = 10^4 \times 10^{0.7\left(\frac{P}{D}-1\right)}$ the transition-turbulent Reynolds number, and $\psi = \log\left(\frac{\mathrm{Re}}{\mathrm{Re}_L}\right) / \log\left(\frac{\mathrm{Re}_T}{\mathrm{Re}_L}\right)$ the intermittency factor.

For the simplified correlation,

$$C_{fL} = \left[-974.6 + 1612.0\left(\frac{P}{D}\right) - 598.5\left(\frac{P}{D}\right)^2\right]\left(\frac{H}{D}\right)^{0.06-0.085\frac{P}{D}};$$

$$C_{fT} = \left[0.8063 - 0.9022\log\left(\frac{H}{D}\right) + 0.3526\left(\log\left(\frac{H}{D}\right)\right)^2\right]\left(\frac{P}{D}\right)^{9.7}\left(\frac{H}{D}\right)^{1.78-2.0\frac{P}{D}}$$

In which P/D is the pitch-to-diameter ratio, and H/D is the wire-pitch-to-diameter ratio.

For the detailed correlation, C_{fL} and C_{fT} take different values according to the geometric type of the liquid subchannel under consideration: triangular (between three pins), edge (between two pins and a wall), or corner (between one pin and two walls). Its complete expression is detailed in Cheng and Todreas (1986). The RMS error of these correlations on a database of 79 bundles is 7 % (detailed) / 7.6 % (simplified) for turbulent flows and 12.2 % (detailed) / 13.6 % (simplified) for laminar flows.

Rehme (1973): This correlation was established on a set of experiments by Rehme (most of which were conducted before 1967) and considered 75 different geometries.

$$\begin{aligned} f &= \left[\frac{64}{\mathrm{Re}\sqrt{F}} + \frac{0.0816}{(\mathrm{Re}\sqrt{F})^{0.133}}\right] F \frac{P_b}{P_{assembly}} \\ F &= \left(\frac{P}{D}\right)^{0.5} + \left[7.6\frac{D_m}{H}\left(\frac{P}{D}\right)^2\right]^{2.16} \end{aligned} \tag{5.24}$$

The Rehme correlation takes into account the influence of the hexagonal duct. This correlation is valid for the following conditions:

$$\begin{aligned} &10^3 < \mathrm{Re} < 3 \times 10^5 \quad \text{(transitory or turbulent flow regime)} \\ &8 < \frac{H}{D_m} < 50 \\ &1.1 < \frac{P}{D} < 1.42 \\ &7 < N_{rod} < 217 \end{aligned}$$

Where D_m: rod diameter + wire diameter, P: rod pitch, D: rod diameter, Re: Reynolds number, using means bundle average value, H: wire pitch, N_{rod}: number of rods, P_b: rod bundle and wire friction perimeter, $P_{assembly}$: total (with hexagonal duct) friction perimeter.

The Rehme correlation is another widely used correlation based on an effective velocity to take into account the swirl flow velocity around the rod. It considers the effects of *P/D* and *H/D* and the influence of the duct and the number of pins. The accuracy of Rehme's correlation to his own experimental data is ±8% for turbulent flows.

5.2.3 User-specified wall friction correlation option

A simple friction-factor correlation for an exponential function of the Reynolds number has been provided for the user to supply their own friction correlation. The current form is a simple implementation, and can be amended as additional dependencies are found to be significant. At the current stage, the form of this correlation is implemented across the entire flow regime, and will be augmented in a way that a user may specify correlations for the laminar, transition and turbulent flow regimes individually.

$$f = A + B \times \mathrm{Re}^C \tag{5.25}$$

The above wall friction correlations are implemented in SAM and summarized in Table 5.3, in which the default correlations for different geometry and flow conditions are shaded.

Table 5.3: List of SAM wall friction models

Name	Correlation	Re	P/D	H/D	N_{rod}	Uncert.	Notes
Circular Pipe Geometry							
Darcy friction	$f = \frac{64}{Re}$	[64, 2200)				Exact	*Default, Laminar
Blasius (1913)	$f = 0.316/Re^{0.25}$	[3e3, inf)					*Default, Turbulent
McAdams (1942)	$f = 0.184 \cdot \mathrm{Re}^{-0.2}$	[3e4, 1e6]					*Default, Turbulent
Zigrang-Sylvester (1985)	$\frac{1}{\sqrt{f}} = -2\log_{10}\left\{\frac{\epsilon}{3.7D} + \frac{2.51}{Re}\left[1.14 - 2\log_{10}\left(\frac{\epsilon}{D} + \frac{21.25}{Re^{0.9}}\right)\right]\right\}$	[3e3, inf)				± 5.5%	Turbulent
Churchill (1977)	$f = 8\left[\left(\frac{8}{Re}\right)^{12} + \frac{1}{(A+B)^{\frac{3}{2}}}\right]^{\frac{1}{12}}$ $A = \left[2.457\, ln \frac{1}{\left(\frac{7}{Re}\right)^{0.9} + \frac{0.27 \in}{D}}\right]^{16}, B = \left(\frac{37530}{Re}\right)^{16}$	[50, inf)				± 3.2	All regimes
RELAP Interpolation	$f = (1-w)f_{lam,Re_{min}} + wf_{turb,Re_{max}},\quad w = \frac{N}{Re_{min}} - \frac{N}{Re}$	[2200, 3000)					*Default, Transition
Wire-Wrapped Bundle Geometry							
Cheng-Todreas (1986)	$f = C_{fL}/Re$, for Re < Re_L; $f = \frac{C_{fL}}{Re}(1-\psi)^{1/3} + \frac{C_{fT}}{Re^{0.18}}\psi^{1/3}$ for Re_L < Re < Re_T; $f = C_{fT}/Re^{0.18}$ for Re > Re_T, $Re_L = 300 * 10^{1.7\left(\frac{P}{D}-1\right)}\quad Re_T = 10^4 \cdot 10^{1.7\left(\frac{P}{D}-1\right)}$ $\psi = \log\left(\frac{\mathrm{Re}}{Re_L}\right)/\log\left(\frac{Re_T}{Re_L}\right)$	[50, 1e6]	[1.025, 1.42]	[8, 50]	[19, 217]	± 14%	*Default, All regimes, Simplified model
Rehme (1973)	$\mathrm{f} = \left[\frac{64}{\mathrm{Re}\sqrt{\mathrm{F}}} + \frac{0.0816}{(\mathrm{Re}\sqrt{\mathrm{F}})^{0.133}}\right]\mathrm{F}\frac{\mathrm{P_b}}{\mathrm{P_{assembly}}}$ $\mathrm{F} = \left(\frac{\mathrm{p}}{\mathrm{d}}\right)^{0.5} + \left[7.6\frac{\mathrm{d_m}}{\mathrm{H}}\left(\frac{\mathrm{p}}{\mathrm{d}}\right)^2\right]^{2.16}$	[1e3, 3e5]	[1.1, 1.42]	[8, 50]	[7, 217]	± 8%	All regimes

6 Reactor Power Models

6.1 Point Kinetics Model

6.1.1 Point kinetics equations

The analysis of the transient behavior of a nuclear reactor requires the coupled simulation reactor kinetics and thermal-hydraulics of the reactor core. The point kinetics model is widely used for the coupled simulation due to its simplicity to capture the transient behavior of the reactor. In the point kinetics model, it is assumed that the reactor power can be separated into space and time function. The assumption is adequate when the space distribution remains nearly constant during the transient. For stationary fuel reactors, the widely used Point Kinetics Equations (PKE) (Fanning et al., 2017) for m-groups of delayed neutron precursor are

$$\frac{dn}{dt} = \frac{\rho_{ext} - \beta_{eff}}{\Lambda} n + \sum_{i=1}^{m} \lambda_i C_i \tag{6.1}$$

$$\frac{dC_i}{dt} = \frac{\beta_i}{\Lambda} n - \lambda_i C_i \tag{6.2}$$

where $n(t)$ is the time-dependent total neutron density, normalized by the neutron population at full fission power; $C_i(t)$ is the i-th delayed neutron precursor density, $\rho_{ext}(t)$ is the time-dependent net reactivity function, β_i is the i-th delayed neutron fraction, $\beta_{eff} = \sum \beta_i$ is the total delayed neutron fraction, Λ is the neutron generation time, and λ_i is the i-th group decay constant. In practice, 6 delayed neutron precursor groups are used.

Eq. (6.1) is an initial value problem (IVP) with a set of coupled linear ODEs. At first we convert it to a vector form as,

$$\frac{d\mathbf{u}}{dt} = \mathbf{F}(\mathbf{u}, t) \tag{6.3}$$

where

$$\mathbf{u} = \begin{pmatrix} n \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \end{pmatrix}, \mathbf{F}(\mathbf{u}, t) = \begin{pmatrix} \frac{\rho_{ext} - \beta_{eff}}{\Lambda} n + \sum\limits_{i=1}^{6} \lambda_i C_i \\ \frac{\beta_1}{\Lambda} n - \lambda_1 C_1 \\ \frac{\beta_2}{\Lambda} n - \lambda_2 C_2 \\ \frac{\beta_3}{\Lambda} n - \lambda_3 C_3 \\ \frac{\beta_4}{\Lambda} n - \lambda_4 C_4 \\ \frac{\beta_5}{\Lambda} n - \lambda_5 C_5 \\ \frac{\beta_6}{\Lambda} n - \lambda_6 C_6 \end{pmatrix} \tag{6.4}$$

For the case of constant reactivity function $\rho_{ext}(t) = \rho_0$, an analytical solution of the PKE can be obtained rather easily. At first, Eq. (6.3) is converted to a matrix form as

$$\frac{d\mathbf{u}}{dt} = \mathbb{A} \cdot \mathbf{u} \tag{6.5}$$

where

$$\mathbb{A}=\begin{pmatrix} \frac{\rho_0-\beta_{eff}}{\Lambda} & \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 & \lambda_6 \\ \frac{\beta_1}{\Lambda} & -\lambda_1 & 0 & 0 & 0 & 0 & 0 \\ \frac{\beta_2}{\Lambda} & 0 & -\lambda_2 & 0 & 0 & 0 & 0 \\ \frac{\beta_3}{\Lambda} & 0 & 0 & -\lambda_3 & 0 & 0 & 0 \\ \frac{\beta_4}{\Lambda} & 0 & 0 & 0 & -\lambda_4 & 0 & 0 \\ \frac{\beta_5}{\Lambda} & 0 & 0 & 0 & 0 & -\lambda_5 & 0 \\ \frac{\beta_6}{\Lambda} & 0 & 0 & 0 & 0 & 0 & -\lambda_6 \end{pmatrix} \tag{6.6}$$

For given delayed neutron parameters and a constant reactivity ρ_0, matrix $\mathbb{A}$ is constant (does not depend on time) and linear (does not depend on $\mathbf{u}$). The matrix $\mathbb{A}$ has real eigenvalues and can be diagonalized as

$$\mathbb{A}=\mathbb{K}\cdot\mathbb{D}\cdot\mathbb{K}^{-1} \tag{6.7}$$

where matrix $\mathbb{D}$ is diagonal with eigenvalues and $\mathbb{K}$ is the right eigenvector matrix whose columns are right eigenvectors,

$$\mathbb{D}=\mathrm{Diag}\big(\omega_1,\cdots,\omega_7\big) \tag{6.8a}$$

$$\mathbb{K}=\begin{pmatrix}\mathbf{K}_1 & \ldots & \mathbf{K}_7\end{pmatrix} \tag{6.8b}$$

The eigenvalues and eigenvector matrix (and its inverse) can be found numerically with little computational burden.

Substituting the diagonalized matrix $\mathbb{A}$ into Eq. (6.5) and making some arrangements, we obtain

$$\frac{d\mathbf{u}}{dt}=\mathbb{K}\cdot\mathbb{D}\cdot(\mathbb{K}^{-1}\cdot\mathbf{u}) \tag{6.9}$$

Multiplying both sides by $\mathbb{K}^{-1}$, we obtain

$$\mathbb{K}^{-1}\frac{d\mathbf{u}}{dt}=\mathbb{D}\cdot(\mathbb{K}^{-1}\cdot\mathbf{u}) \tag{6.10}$$

Let $\mathbf{v}=\mathbb{K}^{-1}\cdot\mathbf{u}$, we obtain

$$\frac{d\mathbf{v}}{dt}=\mathbb{D}\cdot\mathbf{v} \tag{6.11}$$

which is a set of uncoupled ODEs,

$$\frac{dv_i}{dt}=\omega_i v_i,\ \text{for } i=1,\cdots,7 \tag{6.12}$$

The solution of v_i is trivial,

$$v_i(t)=v_{i,0}\exp(\omega_i t),\ \text{for } i=1,\cdots,7 \tag{6.13}$$

Here $v_{i,0}=v_i(t_0)$ is the initial values. Eventually, the analytical solution is obtained with

$$\mathbf{u}=\mathbb{K}\cdot\mathbf{v} \tag{6.14}$$

6.1.2 Fully implicit BDF scheme

The current default solution scheme for PKE is the implicit Euler (BDF1) or second-order backward differentiation formula (BDF2). Both of these two schemes solves PKE and the nonlinear equations (e.g. fluid equations) simultaneously. The discretization of PKE's temporal terms is straightforward,

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^{n}}{\Delta t} = \mathbf{F}(\mathbf{u}^{n+1}, t^{n+1}), \text{ BDF1} \tag{6.15}$$

$$\frac{\frac{3}{2}\mathbf{u}^{n+1}-2\mathbf{u}^{n}+\frac{1}{2}\mathbf{u}^{n-1}}{\Delta t} = \mathbf{F}(\mathbf{u}^{n+1}, t^{n+1}), \text{ BDF2} \tag{6.16}$$

where vector **u** and vector **F** are defined Eq. (6.4). Note that the size of vector **u** and vector **F** will depend on the group of delayed neutron precursor in the PKE equation.

6.1.3 High-order implicit Runge-Kutta scheme

In order to improve the accuracy for solving the PKE and separate the solution of PKE from the other parts of the system, an auxiliary multi-stage Implicit Runge-Kutta (IRK) (Süli and Mayers, 2003) scheme is developed. For convenience, the PKE is re-written as,

$$\frac{d\mathbf{u}}{dt} = (\mathbb{A}_1 + \mathbb{A}_2)\cdot\mathbf{u} \tag{6.17}$$

where

$$\mathbb{A}_1 = \begin{pmatrix} \frac{-\beta_{eff}}{\Lambda} & \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 & \lambda_6 \\ \frac{\beta_1}{\Lambda} & -\lambda_1 & 0 & 0 & 0 & 0 & 0 \\ \frac{\beta_2}{\Lambda} & 0 & -\lambda_2 & 0 & 0 & 0 & 0 \\ \frac{\beta_3}{\Lambda} & 0 & 0 & -\lambda_3 & 0 & 0 & 0 \\ \frac{\beta_4}{\Lambda} & 0 & 0 & 0 & -\lambda_4 & 0 & 0 \\ \frac{\beta_5}{\Lambda} & 0 & 0 & 0 & 0 & -\lambda_5 & 0 \\ \frac{\beta_6}{\Lambda} & 0 & 0 & 0 & 0 & 0 & -\lambda_6 \end{pmatrix}, \mathbb{A}_2 = \begin{pmatrix} \frac{\rho_{ext}(t)}{\Lambda} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{6.18}$$

Notice that $\mathbb{A}_1$ is constant over time while $\mathbb{A}_2$ is in general time-dependent. For explanation purposes, 6 groups of delayed neutron precursor are used, but the scheme is general for a different number of groups.

Let $h = \Delta t$ be the time step size. The general one step s-stage IRK scheme with given coefficients c_i and a_{ij} is formulated as

$$\mathbf{u}^{n+1} = \mathbf{u}^{n} + h\sum_{i=1}^{s} b_i\mathbf{k}_i \tag{6.19}$$

where the intermediate solution $\mathbf{k}_i$ is solved by

$$\mathbf{k}_i = \mathbf{F}(t^n + c_i h, \mathbf{u}^n + h\sum_{j=1}^{s} a_{ij}\mathbf{k}_j), \text{ for } i = 1, \cdots, s \tag{6.20}$$

Because the PKE is a set of linear OKEs, the vector **F** depends linearly on the intermediate solution vector $\mathbf{k}_i$'s, Eq. (6.20) forms a set of linear equations with respect to unknown vector $\mathbf{k}_i$.

The coefficients c_i, a_{ij}, and b_i can be organized compactly in a Butcher tableau (Süli and Mayers, 2003)

$$\begin{array}{c|cccc} c_1 & a_{11} & \cdots & \cdots & a_{1s} \\ c_2 & a_{21} & \cdots & \cdots & a_{2s} \\ \vdots & \vdots & & & \vdots \\ c_s & a_{s1} & \cdots & \cdots & a_{ss} \\ \hline & b_1 & \cdots & \cdots & b_s \end{array} \tag{6.21}$$

Here we will focus on the 1-stage backward Euler formula (IRK1), 2-stage Radau quadrature formula (IRK2), and 3-stage Radau quadrature (IRK3) formula. The Butcher tableau for IRK1, IRK2, and IRK3 are (Süli and Mayers, 2003)

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}, \quad \begin{array}{c|cc} 0 & \frac{1}{4} & -\frac{1}{4} \\ \frac{2}{3} & \frac{1}{4} & \frac{5}{12} \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array}, \quad \begin{array}{c|ccc} 0 & \frac{1}{9} & \frac{-1-\sqrt{6}}{18} & \frac{-1+\sqrt{6}}{18} \\ \frac{6-\sqrt{6}}{10} & \frac{1}{9} & \frac{88+7\sqrt{6}}{360} & \frac{88-43\sqrt{6}}{360} \\ \frac{6+\sqrt{6}}{10} & \frac{1}{9} & \frac{88+43\sqrt{6}}{360} & \frac{88-7\sqrt{6}}{360} \\ \hline & \frac{1}{9} & \frac{16+\sqrt{6}}{36} & \frac{16-\sqrt{6}}{36} \end{array} \tag{6.22}$$

IRK1 is expected to be first-order accurate and is equivalent to BDF1 scheme, IRK2 is expected to be third-order accurate, and IRK3 is expected to be fifth-order accurate.

For a s-stage IRK formulation, the formula for solving the intermediate solution $\mathbf{k}_1, \cdots, \mathbf{k}_s$ is

$$(\mathbb{M}_1^s + \mathbb{M}_2^s) \cdot \begin{pmatrix} \mathbf{k}_1 \\ \vdots \\ \mathbf{k}_s \end{pmatrix} = \begin{pmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_s \end{pmatrix} \tag{6.23}$$

where $\mathbb{M}_1^s$ and $\mathbb{M}_2^s$ are coefficient matrices, $\mathbf{r}_1$ to $\mathbf{r}_s$ are right hand side source vectors. They are defined as

$$\mathbb{M}_1^s = \begin{pmatrix} \mathbb{I} - ha_{11}\mathbb{A}_1 & \cdots & -ha_{1s}\mathbb{A}_1 \\ \vdots & \cdots & \vdots \\ -ha_{s1}\mathbb{A}_1 & \cdots & \mathbb{I} - ha_{ss}\mathbb{A}_1 \end{pmatrix}, \mathbb{M}_2^s = \begin{pmatrix} -ha_{11}\mathbb{A}_2(\tau_1) & \cdots & -ha_{1s}\mathbb{A}_2(\tau_1) \\ \vdots & \cdots & \vdots \\ -ha_{s1}\mathbb{A}_2(\tau_s) & \cdots & -ha_{ss}\mathbb{A}_2(\tau_s) \end{pmatrix} \tag{6.24}$$

$$\mathbf{r}_i = [\mathbb{A}_1 + \mathbb{A}_2(\tau_i)] \cdot \mathbf{u}^n, \text{ for } i = 1, \cdots, s \tag{6.25}$$

where $\tau_i = t^n + c_i h$. The unknown intermediate solution vectors $\mathbf{k}_i$ can be solved efficiently with a direct linear equation solver, e.g. LU decomposition.

An example which models a step-reactivity insertion, $\rho_{ext}(t) = 0.003$, is designed to verify the order of accuracy of different schemes. The delayed neutron parameters are:

$$\begin{aligned} &\Lambda = 0.00002, \beta_{eff} = 0.007 \\ &\beta_i = (0.000266, 0.001491, 0.001316, 0.002849, 0.000896, 0.000182) \\ &\lambda_i = (0.0127, 0.0317, 0.155, 0.311, 1.4, 3.87) \end{aligned} \tag{6.26}$$

The initial condition assumes an equilibrium state with

$$\mathbf{u}_0 = \left(1, \frac{\beta_1}{\lambda_1\Lambda}, \frac{\beta_2}{\lambda_2\Lambda}, \frac{\beta_3}{\lambda_3\Lambda}, \frac{\beta_4}{\lambda_4\Lambda}, \frac{\beta_5}{\lambda_5\Lambda}, \frac{\beta_6}{\lambda_6\Lambda}\right)^T \tag{6.27}$$

The neutron density at time $t = 1$ s with different time step is used to obtain the observed order of accuracy. The exact solution is $n_e = 2.2248007490664827$. The study results are shown in Table 6.1 and Figure 6.1. As expected, the current BDF1 and BDF2 schemes are observed to be first and second order accurate; and the IRK1, IRK2, and IRK3 schemes are observed to be first, third, and fifth order accurate. Notice that since the IRK3 is very accurate, the truncation error becomes to play when the time step is too small. The BDF1 and IRK1 schemes are inherently identical, which is demonstrated in the results as well.

Table 6.1: Observed order of accuracy in PKE solution of step-reactivity insertion with different schemes.

dt: s	BDF1		BDF2		IRK1		IRK2		IRK3	
	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order
0.1	5.719E-04	NA	2.084E-04	NA	5.719E-04	NA	2.325E-06	NA	2.18E-09	NA
0.05	3.105E-04	0.881	5.369E-05	1.957	3.105E-04	0.881	3.001E-07	2.954	6.99E-11	4.964
0.025	1.613E-04	0.945	1.352E-05	1.989	1.613E-04	0.945	3.816E-08	2.975	2.18E-12	5.001
0.0125	8.211E-05	0.974	3.390E-06	1.996	8.211E-05	0.974	4.813E-09	2.987	8.26E-14	4.724
0.00625	4.143E-05	0.987	8.484E-07	1.998	4.143E-05	0.987	6.043E-10	2.993	1.73E-14	NA
0.003125	2.080E-05	0.994	2.122E-07	1.999	2.080E-05	0.994	7.572E-11	2.997	1.73E-14	NA

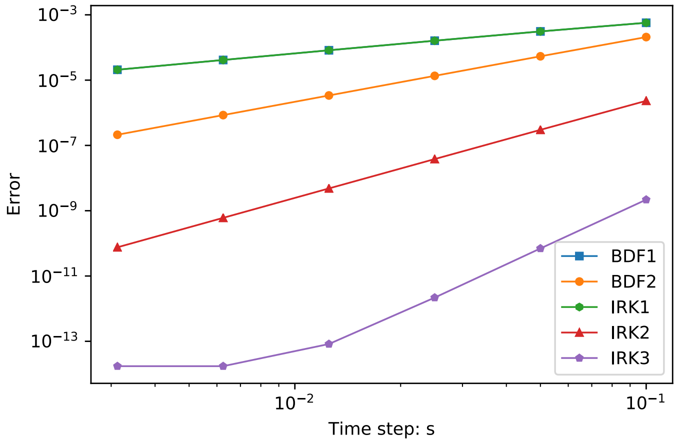


Figure 6.1: Numerical error in PKE solution of step-reactivity insertion with different schemes. Notice that BDF1 and IRK1 are identical.

6.2 Reactivity Feedback Models

The reactivity function $\rho_{ext}(t)$ in Eq. (6.1) represents the net reactivity of the external reactivity (e.g. reactivity due to control rod movement) and various reactivity feedbacks of from the system. Four types of reactivity feedback have been developed for working with the PKE model, including fuel axial expansion, core radial expansion, fuel Doppler, and coolant density reactivity feedback.

6.2.1 Fuel axial expansion reactivity feedback

In advanced nuclear reactors (e.g. Sodium-cooled Fast Reactor), the fuel, especially metallic fuel, expands or shrinks within the cladding in response to the fuel temperature changes during the transient, which imposes a positive or negative reactivity feedback. The fuel axial expansion model was developed to consider the reactivity feedback in response to the fuel temperature changes during the transient. In SAM, the fuel axial expansion reactivity (Fanning et al., 2017; Hu et al., 2019b) is calculated by the difference between the transient and initial values of total reactivity worth in the fuel pins,

$$R_A(t) = \int_{z=0}^{z=L'} \rho_f(\xi,t) f(\xi) A \mathrm{d}\xi \tag{6.28}$$

$$\Delta R_A(t) = R_A(t) - R_A^{ss} \tag{6.29}$$

where ΔR_A is the fuel axial expansion reactivity feedback in the unit of Δ k/k; $\rho_f(\xi,t)$ is the fuel density at transient time t in the unit of kg/m ^{3}; $f(\xi)$ is the fuel reactivity coefficient in unit of Δ k/k / kg; L' and A are the expanded fuel length during the transient and cross-section area, respectively. The integration will consider the axial thermal expansion of the fuel pin.

The fuel axial expansion reactivity model was implemented in SAM using the linear thermal expansion coefficient of fuel and cladding material, which are used in estimating the fuel pin axial displacement due to the temperature change. The axial displacement of fuel pin was calculated by

$$u_i(z) = \int_0^z \alpha_i(\bar{T}_i - \bar{T}_{i,0}) \mathrm{d}z, i = 1,2 \tag{6.30}$$

where u_i, α_i, $\bar{T}_i$, and $\bar{T}_{0,i}$ are the axial displacement (in free expansion), linear thermal expansion coefficient, area-averaged temperature, and stress-free temperature of fuel ($i = 1$) or cladding ($i = 2$), respectively. The current model also considers the condition when the fuel rod and the cladding are in eutectic formation. In eutectic formation, the fuel rod and cladding are assumed to expand together due to the strong shear stress in the rod-cladding boundary. The effective axial displacement (Hetnarski et al., 2009) for eutectic fuel pin is

$$u_e = u_1 \frac{A_1E_1}{A_1E_1 + A_2E_2} + u_2 \frac{A_2E_2}{A_1E_1 + A_2E_2} \tag{6.31}$$

where u_e is the effective axial displacement in the fuel pin, A_i and E_i are the area and Young's modulus of fuel ($i = 1$) and cladding ($i = 2$), respectively.

Considering dividing the fuel into N number of layers in axial direction, when the axial displacement is available, the integral reactivity worth is calculated by,

$$R_A(t) = \sum_{i=1}^{N} m_i f(\xi_i) \tag{6.32}$$

where m_i is the mass of the i-th axial layer which is held constant during the transient calculation due to mass conservation, ξ_i is the axial center location of i-th layer with the axial displacement in consideration, i.e. ξ_i is determined by the axial displacement.

The previous model works well for the solid fuel where the axial expansion is small. However, it faces some issues in case of liquid fuel: the assumption of linear thermal expansion becomes problematic for liquid fuel, such as used in molten-salt reactor, where the axial thermal expansion (and fuel density change) could be significant. Due to thermal expansion, there could be a portion of fuel outside the nominal fuel length. The exact reactivity worth of this portion of fuel needs being provided by the reactor physics calculation. This is understood as an input that should be provided by the SAM users. For liquid fuel, the reactivity worth of this portion of fuel could be important.

An additional fuel axial expansion reactivity feedback model is added into SAM for better treatment of liquid fuel. In this model, the density change of the fuel during the transient is directly used in estimating the reactivity. Let the mesh size be Δz; A be the fuel area; f_i and ρ_i be the reactivity worth and fuel density at the i-th layer. The fuel axial expansion reactivity inside the fixed mesh (i.e. the nominal fuel length region) is evaluated with

$$\Delta R^{'} = \sum_{i=1}^{N} [\rho_i(t) - \rho_i^{ss}] f_i A \Delta z \tag{6.33}$$

where $\rho_i(t)$ is the temperature dependent fuel density of i-th layer. The reactivity contribution of the fuel beyond the fixed mesh is considered by

$$\Delta R^{*} = f^{*} \rho_N(t) A [L^{'} - L] \tag{6.34}$$

where f^{*} is the value of fuel reactivity function beyond the fixed mesh, $\rho_N(t)$ is the density of the last layer of fuel, and $L^{'}$ is the expanded fuel length during the transient. $L^{'}$ is calculated from the fuel density change using mass conservation. The total fuel axial expansion reactivity is thus

$$\Delta R_A(t) = \Delta R^{*} + \Delta R^{'} = f^{*} \rho_N(t) A [L^{'} - L] + \sum_{i=1}^{N} [\rho_i(t) - \rho_i^{ss}] f_i A \Delta z \tag{6.35}$$

This model avoids the assumption of linear thermal expansion by using directly the relation between fuel density and fuel temperature, and is thus valid for liquid fuel.

6.2.2 Core radial expansion reactivity feedback

Due to temperature changes in the cooling system, the reactor core experiences radial thermal expansion, which impose a positive or negative reactivity feedback. For most advanced nuclear reactor design, there are also some constraint systems (e.g. Grid plate, Above core load pad, Top load pad), and the geometry of the reactor core during the transient is also affected by those constraint system (Figure 6.2).

The core radial expansion model is developed to consider the reactivity feedbacks in response to the thermal expansions of reactor cores during the transient. The current model implemented in SAM is able to consider the expansion effects based on multiple constraint systems (Figure 6.2). The reactivity feedbacks due to the expansion effects at different elevations are weighted by user-defined factors (Fanning et al., 2017; Hu et al., 2019b)

$$\Delta R_{RC}(t) = \sum_{n}^{N} \left(\frac{\Delta R}{R}\right)_n w_n \rho_{RC,n} \tag{6.36}$$

where ΔR_{RC} is the core radial expansion feedback in the unit of Δk/k; $\Delta R/R$ is the relative change in the radius of reactor core; $\rho_{RC,n}$ is core radial expansion coefficient at position n in the unit of Δk/k

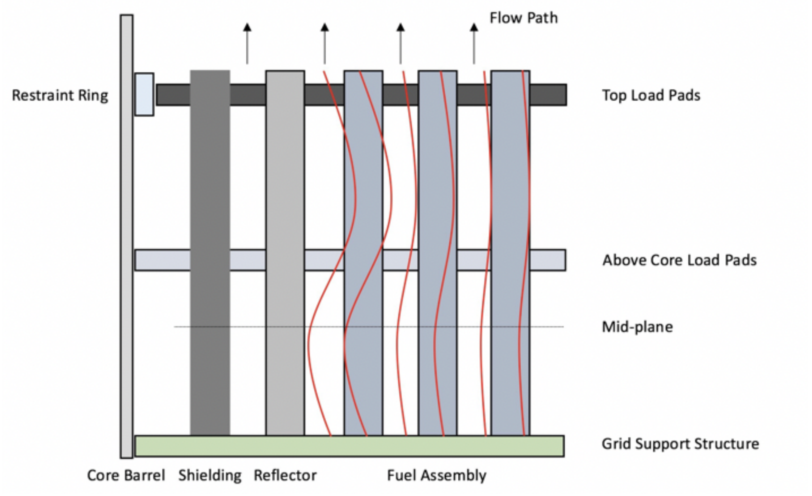


Figure 6.2: Restraint systems in typical SFR and core radial expansion

per $\Delta R/R$; w_n is the user-defined weighting factor; N is the total number of the constraint systems. The relative change in radius of reactor core at different position n is provided by either an external thermal-mechanical calculation or SAM standalone calculation. When SAM standalone calculation is enabled, a linear thermal expansion model will be initialized to calculate the displacement of individual constraint system and thus the relative change in core radius, i.e.

$$\left(\frac{\Delta R}{R}\right)_n = \alpha_n[\bar{T}_n(t) - \bar{T}_n^{ss}] \tag{6.37}$$

where α_n and $\bar{T}_n$ are the thermal expansion coefficient and average temperature of the n-th radial constraint system, specified by the user.

6.2.3 Fuel Doppler reactivity feedback

The fuel Doppler reactivity model is implemented in SAM to consider the reactivity feedbacks in response to the fuel temperature changes during the transient. The Doppler reactivity feedback is integrated over the core channels and provided to the Point Kinetics model for the calculation of fission power. Considering the fuel is split into N axial layers, the Doppler reactivity is calculated by (Fanning et al., 2017; Hu et al., 2019b)

$$R_D(t) = \sum_n^N \alpha_{D,n} \ln \frac{T_{f,n}}{T_{f,n}^{ss}} \tag{6.38}$$

where R_D is the fuel Doppler reactivity feedback in the unit of Δk/k; $\alpha_{D,n}$ is the fuel Doppler reactivity coefficient of n-th layer in unit of Δk/k per $\ln((T+\Delta T)/T)$; $T_{f,n}(t)$ and $T_{f,n}^{ss}$ are the average fuel temperature of n-th layer at the time t and at the beginning (usually a steady-state), respectively. The Doppler reactivity coefficients need being generated from external neutronics calculations. The Doppler reactivity coefficients are provided as user inputs in SAM input model. With user-provided fuel Doppler reactivity coefficients, the fuel temperature changes during the transient impose a positive or negative reactivity feedback on the fission power.

6.2.4 Coolant density reactivity feedback

The coolant density reactivity model is developed to consider the reactivity feedbacks in response to the coolant temperature changes during the transient. The coolant density reactivity feedback is integrated over the flow channels and the difference between the initial and transient values is provided to the Point Kinetics model for the calculation of fission power. Considering the flow channel is split into N volumes, (Fanning et al., 2017; Hu et al., 2019b)

$$R_{CD}(t) = \sum_{n}^{N} \alpha_n \rho_n(t) V_n \tag{6.39}$$

$$\Delta R_{CD}(t) = R_{CD}^{ss} - R_{CD}(t) \tag{6.40}$$

where $\Delta R_{CD}(t)$ is the coolant density reactivity at time t in the unit of Δk/k; α_n is the coolant density reactivity coefficient of n-th volume in unit of Δk/k per kg; $\rho_n(t)$ is the coolant density of n-th volume at the time of t; V_n is the volume of n-th volume. Together with user-provided coolant density reactivity coefficients, the reactivity feedbacks in response to the coolant temperature changes during the transient impose a positive or negative impact on the fission power.

6.3 Decay Heat Model

The decay heat model based on the ANS standard ANSI/ANS-5.1-2005 was implemented in SAM for stationary fuel. All labels used in equations of this subsection are listed in Table 6-2.

Table 6.2: Description of notations used in decay heat equations

Term	Unit	Meaning
$f_j(t)$	MeV/fission-s	Decay power t seconds after a fission event from isotope j.
$\alpha_{i,j}$	MeV/fission-s	Tabulated data from ANS standard.
$\lambda_{i,j}$	s^{-1}	Tabulated data from ANS standard.
$P_d(t)$	W	Decay heat power
Q_f	MeV/fission	Prompt recoverable energy per fission
C_1	J/MeV	$C_1 = 1.602 \times 10^{-13}$
$F(t)$	Fission/s	Total fission rate $F(t) = \frac{P_f(t)}{Q_f C_1}$
$H_{i,j}(t)$	J	Amount of fictitious decay heat precursor
$\beta_{i,j}$		$\beta_{i,j} = \frac{\alpha_{i,j}}{\lambda_{i,j} Q_f}$
j		Index j refers to the four isotopes (U-235, U-238, Pu-239, Pu-241).

In ANS standard, the decay heat power at time t for a fission event from isotope j occurring at time 0 is express as

$$f_j(t) = \sum_i \alpha_{i,j} e^{-\lambda_{i,j} t} \tag{6.41}$$

Then, the decay power

$$P_d(t) = \sum_j \int_{-\infty}^{t} \frac{P_f(t')}{Q_f C_1} C_1 f_j(t-t') dt' \tag{6.42}$$

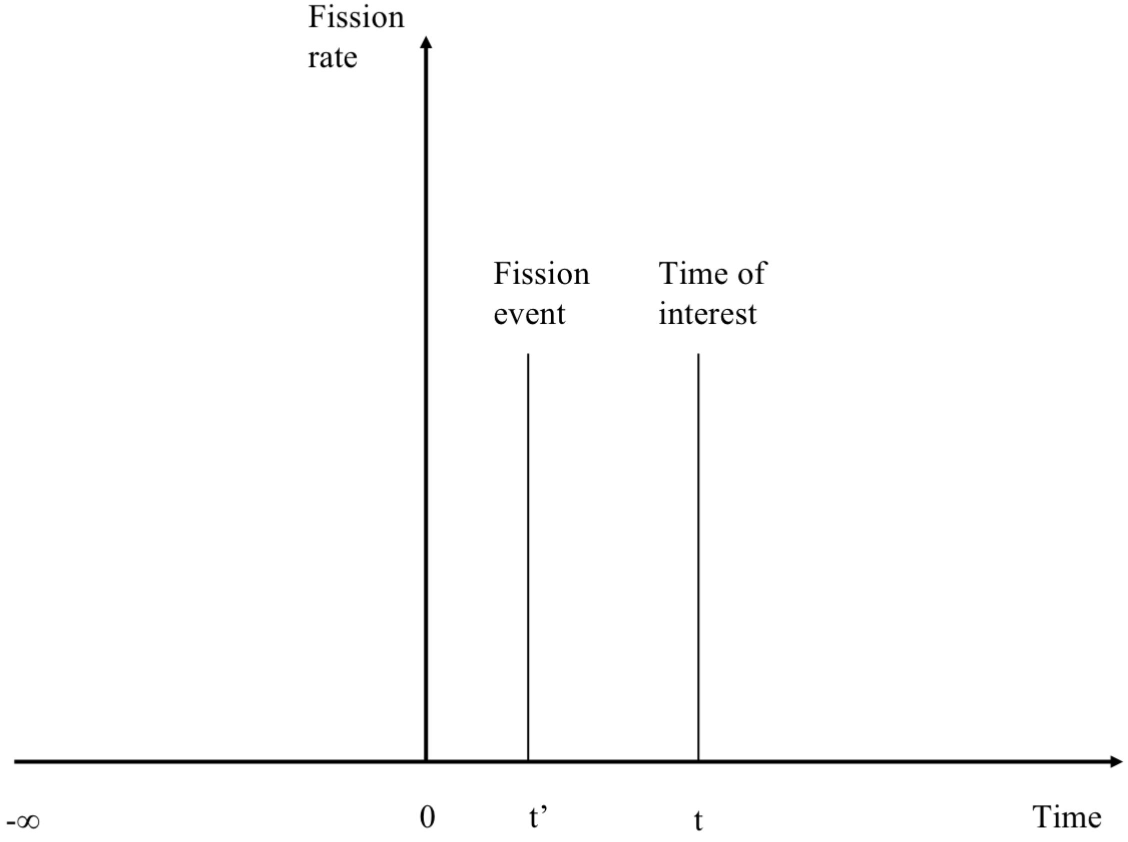


Figure 6.3: Decay heat model time

with t and t' defined in Figure 6-2
$H_{i,j}(t)$ is defined as

$$H_{i,j}(t) = \int_{-\infty}^{t} F(t') \frac{\alpha_{i,j}}{\lambda_{i,j}} e^{-\lambda_{i,j}(t-t')} dt' \tag{6.43}$$

So that

$$P_d(t) = \sum_j \sum_i \lambda_{i,j} H_{i,j}(t) \tag{6.44}$$

The initial condiction is

$$H_{i,j}(0) = \int_{-\infty}^{0} P_f(t') \beta_{i,j} e^{\lambda_{i,j} t'} dt' \tag{6.45}$$

In the current implementation, $P_f(t')$ is assumed to be constant before $t = 0$, so

$$H_{i,j}(0) = \frac{P_f(-\infty) \beta_{i,j}}{\lambda_{i,j}} \tag{6.46}$$

and

$$P_d(0) = \sum_j \sum_i P_f(-\infty)\beta_{i,j} \tag{6.47}$$

At a certain time step t_k,

$$\begin{aligned} H_{i,j}(t_k + \Delta t) &= \int_{-\infty}^{t_k+\Delta t} P_f(t')\beta_{i,j} e^{-\lambda_{i,j}(t_k+\Delta t - t')} dt' \\ &= \int_{-\infty}^{t_k} P_f(t')\beta_{i,j} e^{-\lambda_{i,j}(t_k+\Delta t - t')} dt' + \int_{t_k}^{t_k+\Delta t} P_f(t')\beta_{i,j} e^{-\lambda_{i,j}(t_k+\Delta t - t')} dt' \\ &= H_{i,j}(t_k) e^{-\lambda_{i,j}\Delta t} + \int_{t_k}^{t_k+\Delta t} P_f(t')\beta_{i,j} e^{-\lambda_{i,j}(t_k+\Delta t - t')} dt' \end{aligned} \tag{6.48}$$

It is further assumed that $P_f(t')$ is a 2nd-order polynomial function of t' within Δt. The coefficients of the polynomial can be solved using data from the two earlier time steps. Then, the integral can be analytically integrated.

7 Numerical Schemes

7.1 Finite Element Method Implementation in SAM

Because of its dependence on the MOOSE framework, SAM numerical models are implemented in finite element method (FEM). In essence, FEM tries to find a solution function that is made up of "shape functions" multiplied by coefficients and added together, to approximate the solution to the governing equations. For the basics of FEM, interested readers can refer to a wide range of references available in the literature. It will not be discussed in this document.

SAM uses the continuous Galerkin finite element method and the Language shape functions. Both are available through the MOOSE framework. The key implementation in SAM is thus the weak forms of the governing equations discussed in Chapters 2 and 3.

For the left sides of the fluid flow equations Eq. (2.2), the weak form of each term can be given as:

$$\begin{aligned}
\left(\frac{\partial \rho}{\partial t}, \psi\right) &= \left(\frac{\partial \rho}{\partial T}\frac{\partial T}{\partial t}, \psi\right) \\
\left(\frac{\partial \rho u}{\partial t}, \psi\right) &= \left(u\frac{\partial \rho}{\partial T}\frac{\partial T}{\partial t} + \rho\frac{\partial u}{\partial t}, \psi\right) \\
\left(\frac{\partial \rho H}{\partial t}, \psi\right) &= \left(H\frac{\partial \rho}{\partial T}\frac{\partial T}{\partial t} + \rho C_p\frac{\partial T}{\partial t}, \psi\right) \\
\left(\frac{\partial \left(\rho u\right)}{\partial z}, \psi\right) &= -\left(\rho u, \nabla\psi\right) + \langle \rho u\hat{n}, \psi\rangle \\
\left(\frac{\partial \left(\rho uu\right)}{\partial z}, \psi\right) &= -\left(\rho uu, \nabla\psi\right) + \langle \rho uu\hat{n}, \psi\rangle \\
\left(\frac{\partial \left(\rho uH\right)}{\partial z}, \psi\right) &= -\left(\rho uH, \nabla\psi\right) + \langle \rho uH\hat{n}, \psi\rangle
\end{aligned} \tag{7.1}$$

In which, $(f, \psi) = \int_\Omega \psi \cdot f d\Omega$, and it represents the integration of function f multiplied by the "test" function over volume Ω; $\langle f\hat{n}, \psi\rangle = \int_S f\hat{n} \cdot \psi dS$, and it represents the integration of function f multiplied by the "test" function over surface S, $\hat{n}$ is the normal direction to the surface S. Note that the above weak form utilizes the Gauss's Divergence Theorem for the volume integral of the divergence terms.

$$\int_V \nabla \cdot F dV = \int_S (F \cdot \hat{n})\, dS \tag{7.2}$$

The weak forms of other terms in the fluid equation and the heat conduction equations can be derived in the similar fashion. The resulting terms in volume integral forms are implemented as MOOSE *Kernels*, while the terms in surface integrals are implemented as MOOSE *IntegratedBCs*.

7.1.1 Spatial Discretizations

In SAM, both linear elements (EDGE2 in 1-D and QUAD4 in 2-D in libMesh) and the second-order elements (EDGE3 and QUAD9 in libMesh) are available for use in the finite-element discretization of fluid flow and solid structures. For first-order elements using piece-wise linear Lagrange shape functions, the trapezoidal rule is recommended for the numerical integration; while the

Gaussian quadrature rule is recommended for second-order elements (with second-order Lagrange shape functions) in SAM. In one-dimensional analysis,

$$\text{Trapezoidal rule: } \int_a^b f(x)\,dx = (b-a)\,\frac{[f(a)+f(b)]}{2} \tag{7.3}$$

$$\text{Gaussian quadrature rule: } \int_a^b f(x)\,dx = \sum_{qp} f(x_{qp})\,w_{qp} \tag{7.4}$$

In which x_{qp} is the quadrature point, and w_{qp} is the weight. In SAM, the Gauss-Legendre quadrature is used (through MOOSE and LibMesh); and the quadrature points and weights are well defined.

In one-dimensional analysis, it is well known that trapezoid formula with an interval h gives error of the order $O\left(h^2\right)$. On the other hand, the Gaussian quadrature rule can exactly integrate polynomials of order 2n-1 with n quadrature points, and could have exponential convergence rates. However, the error can be difficult to estimate as it depends on the 2n order derivative. The error bound can be defined as (Kahaner et al., 1989),

$$\text{Error} = \int_a^b f(x)\,dx - \sum_{qp} f(x_{qp})\,w_{qp} = \frac{(b-a)^{2n+1}\,(n!)^4}{(2n+1)\,[(2n)!]^3} f^{(2n)}(\xi), \quad a < \xi < b. \tag{7.5}$$

It can be concluded that SAM spatial discretization scheme is at least second-order accurate with the first-order elements, and could have exponential convergence rates with the second-order elements for continuous problems.

7.1.2 Temporal Discretizations

SAM supports a number of standard time integration methods (available in MOOSE) such as the explicit Euler, implicit Euler (or backward Euler), and BDF2 (backward differentiation formula - 2$^{\text{nd}}$ order method, Crank-Nicolson, and Runge-Kutta methods. For most reactor applications, we recommend to use the implicit Euler or BDF2 methods with SAM.

The backward differentiation formula (BDF) is a family of implicit methods for the numerical integration of ordinary differential equations. They are linear multistep methods that, for a given function and time, approximate the derivative of that function using information from already computed times, thereby increasing the accuracy of the approximation. Note that the first order method of this family, BDF1, is equivalent to the backward Euler method. For a time-step-size Δt, applying the BDF methods to the ordinary differential equation:

$$\frac{\partial u}{\partial t} = f(u,t) \tag{7.6}$$

would result in:

$$f\left(u^{n+1},t^{n+1}\right) = \frac{u^{n+1}-u^n}{\Delta t} + O(\Delta t), \quad \text{Backward Euler or BDF1;} \tag{7.7}$$

$$f\left(u^{n+1},t^{n+1}\right) = \frac{\frac{3}{2}u^{n+1} - 2u^n + \frac{1}{2}u^{n-1}}{\Delta t} + O\left(\Delta t^2\right), \text{BDF2.} \tag{7.8}$$

It can be conclude that SAM temporal discretization can be second-order accurate when using the BDF2 scheme. The performance of the SAM FEM model and the effects of the spatial and temporal discretization schemes can be found in References (Hu, 2015, 2017).

7.2 Solution Methods

The Jacobian-Free Newton Krylov (JFNK) solution method is used to solve the whole SAM system of equations. The JFNK method is a multi-level approach, the outer Newton's iterations (nonlinear solver) and inner Krylov subspace methods (linear solver), in solving large nonlinear systems. The concept of 'Jacobian-free' is proposed, because deriving and assembling large Jacobian matrices could be difficult and expensive. One feature of JFNK is that all the unknowns are solved simultaneously in a fully coupled fashion. This solution scheme avoids the errors from operator splitting and is especially suitable for conjugate heat transfer problems in which the heat conduction in the solid is tightly coupled with the fluid flow.

However, in most applications, the Krylov subspace methods require preconditioning to be efficient. Although the JFNK method is used, the entries of the Jacobian matrix are still derived and implemented for the preconditioning purpose. The execution speed of the code strongly depends on the number of nonlinear (Newton) and linear (Krylov) iterations. The details of the JFNK method and some preconditioning techniques can be found in Knoll and Keyes (2004).

Assuming that the FEM discretized equations form a system of discretized nonlinear equations:

$$\mathbf{F}(\mathbf{u}) = 0 \tag{7.9}$$

Where $\mathbf{F}$ represents the nonlinear equation system and $\mathbf{u}$ is the unknown vector. Newton's method iteratively solves the above nonliner system by solving a series of linearized system:

$$\mathbb{J}\left(\mathbf{u}^k\right)\delta\mathbf{u}^k = -\mathbf{F}\left(\mathbf{u}^k\right) \tag{7.10}$$

with a given initial guess to the unknown vector $\mathbf{u}^0$, and updates the solution as $\mathbf{u}^{k+1} = \mathbf{u}^k + \delta\mathbf{u}^k$. In equations above, $\mathbb{J}$ is the associated Jacobian matrix and $\mathbb{J}\left(\mathbf{u}^k\right) = \mathbf{F}'\left(\mathbf{u}^k\right) = \frac{d\mathbf{F}(\mathbf{u})}{d\mathbf{u}}|_{\mathbf{u}^k}$.

The Newton iteration is terminated if the nonlinear residual is sufficiently small, or based on a required drop in the norm of the nonlinear residual

$$\frac{\left\|\mathbf{F}\left(\mathbf{u}^k\right)\right\|}{\left\|\mathbf{F}\left(\mathbf{u}^0\right)\right\|} < \text{tolerance}, \tag{7.11}$$

and/or a sufficiently small Newton update:

$$\frac{\left\|\delta\mathbf{u}^k\right\|}{\left\|\mathbf{u}^k\right\|} < \text{tolerance}. \tag{7.12}$$

For a scalar problem with n equations and n unknowns, $\mathbf{F}(\mathbf{u}) = \{F_1, F_2, \ldots F_n\}$ and $\mathbf{u} = \{u_1, u_2, \ldots u_n\}$, and the $(i,\ j)$-th element of the Jacobian matrix is

$$J_{ij} = \frac{\partial F_i(\mathbf{u})}{\partial u_j}. \tag{7.13}$$

In the JFNK approach, a Krylov method is used to solve the linear system of equations given by Eq. (7.10). An initial linear residual, $\mathbf{r}_0$, is defined, given an initial guess, $\delta\mathbf{u}_0$, for the Newton correction,

$$\mathbf{r}_0 = -\mathbf{F}(\mathbf{u}) - \mathbb{J}(\mathbf{u})\,\delta\mathbf{u}_0 \tag{7.14}$$

Note that the nonlinear iteration index k in Eq. (7.10) has been dropped since the Krylov iteration is performed at a fixed k. Let j be the Krylov iteration index, the jth iteration $\delta\mathbf{u}_j$ is drawn from the subspace spanned by the Krylov vectors, $\{\mathbf{r}_0, \mathbb{J}\mathbf{r}_0, \mathbb{J}^2\mathbf{r}_0, \ldots, \mathbb{J}^{j-1}\mathbf{r}_0\}$, and can be written as

$$\delta\mathbf{u}_j = \delta\mathbf{u}_0 + \sum_{i=0}^{j-1} \beta_i \mathbb{J}^i \mathbf{r}_0 \tag{7.15}$$

where the scalars β_i minimize the residual $\|\mathbf{F}(\mathbf{u}) + \mathbb{J}(\mathbf{u})\,\delta\mathbf{u}_j\|$ in a least-squares sense.

As evidently shown in above equations, the JFNK method does not require the explicit formation of the Jacobian matrix, instead it only needs the action of the Jacobian only in the form of matrix–vector products, which can be approximated by

$$\mathbb{J}\mathbf{v} \approx \frac{[\mathbf{F}(\mathbf{u}+\varepsilon\mathbf{v}) - \mathbf{F}(\mathbf{u})]}{\varepsilon} \tag{7.16}$$

where ε is a small perturbation.

However, in most applications, the Krylov subspace methods require preconditioning to be efficient. Using right preconditioning, Eq. (7.10) becomes

$$\left(\mathbb{J}\mathbb{P}^{-1}\right)(\mathbb{P}\delta\mathbf{u}) = -\mathbf{F}(\mathbf{u}), \tag{7.17}$$

in which $\mathbb{P}$ represents the precondition matrix (or process) and $\mathbb{P}^{-1}$ is the inverse of the preconditioning matrix.

Right preconditioning is realized through a two-step process. First solve

$$\left(\mathbb{J}\mathbb{P}^{-1}\right)\mathbf{w} = -\mathbf{F}(\mathbf{u}), \tag{7.18}$$

for $\mathbf{w}$. Then solve

$$\delta\mathbf{u} = \mathbb{P}^{-1}\mathbf{w}. \tag{7.19}$$

for $\delta\mathbf{u}$. Note that while we may refer to the precondition matrix $\mathbb{P}$, operationally the algorithm only requires the action of $\mathbb{P}^{-1}$ on a vector. The right-preconditioned version of Eq. (7.16) is:

$$\mathbb{J}\mathbb{P}^{-1}\mathbf{v} \approx \frac{\left[\mathbf{F}\left(\mathbf{u}+\varepsilon\mathbb{P}^{-1}\mathbf{v}\right) - \mathbf{F}(\mathbf{u})\right]}{\varepsilon}. \tag{7.20}$$

This operation is done once per linear iteration, and is actually done in two steps:

1. Preconditioning: Solve for $\mathbf{y}$ in $\mathbb{P}\mathbf{y} = \mathbf{v}$.

2. Perform matrix-free product $\mathbb{J}\mathbf{y} \approx [\mathbf{F}(\mathbf{u}+\varepsilon\mathbf{y}) - \mathbf{F}(\mathbf{u})]/\varepsilon$.

7.3 Preconditioning Matrix

As mentioned above, a Krylov-type of method generally requires preconditioning to be efficient and effective. It is also well known that the closer the preconditioning matrix is to the exact Jacobian matrix, the better the convergence behavior. In SAM, an approximated Jacobian matrix is computed and passed to the underlying numerical solver library (PETSc) for the pre-conditioning purpose, as the exact Jacobian matrix is very difficult to obtain and not necessary. For one-dimensional flow and heat conduction problems, tri-diagonal terms, due to spatial discretization, are included in the preconditioning matrix. Since conjugate heat transfer is a tightly coupled phenomenon between the solid conduction and fluid flow, its Jacobian terms must be included. The Jacobian terms represent the effect of one variable perturbation on the residuals of another variable.

7.3.1 Preconditioning of Flow Equations

The residual terms of fluid flow equations can be given for each primary variable (p, u, T) as (see Chapter 2):

$$
\begin{aligned}
r_p &= \left(\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial z},\ \psi\right) + \left(\rho\frac{\partial u}{\partial t} + \rho u\frac{\partial u}{\partial z} + \frac{\partial p}{\partial z} + \rho g + \frac{f}{D_e}\frac{\rho u|u|}{2},\ \tau_{PSPG}\cdot\nabla\psi\right)\\
r_u &= \left(\frac{\partial \rho u}{\partial t} + \frac{\partial \rho u u}{\partial z} + \frac{\partial p}{\partial z} + \rho g + \frac{f}{D_e}\frac{\rho u|u|}{2}, \psi\right)\\
&\quad + \left(\rho\frac{\partial u}{\partial t} + \rho u\frac{\partial u}{\partial z} + \frac{\partial p}{\partial z} + \rho g + \frac{f}{D_e}\frac{\rho u|u|}{2}, \tau_{SUPG}\cdot\nabla\psi\right)\\
r_T &= \left(\frac{\partial \rho H}{\partial t} + \frac{\partial \rho u H}{\partial z} - q^{'''}, \psi\right) + \left(\rho C_p\frac{\partial T}{\partial t} + \rho C_p u\frac{\partial T}{\partial z} - q^{'''}, \tau_{SUPG}\cdot\nabla\psi\right)
\end{aligned}
\tag{7.21}
$$

The diagonal Jacobian terms for internal nodes can be derived after applying Gauss's Divergence Theorem:

$$
\begin{aligned}
J(p,p) &= \tau_{PSPG}\cdot\nabla\psi\\
J(u,u) &= \left(\rho\frac{d\frac{\partial u}{\partial t}}{du} + \frac{d\rho}{dT}\frac{\partial T}{\partial t}\right)\cdot\psi - 2\rho u\cdot\nabla\psi \pm \frac{f}{D_e}\rho u\cdot\psi\\
&\quad + \tau_{SUPG}\cdot\nabla\psi\cdot\left(\rho\frac{d\frac{\partial u}{\partial t}}{du} + \rho\frac{\partial u}{\partial z} + \rho u\frac{d\frac{\partial u}{\partial z}}{du} + \pm\frac{f}{D_e}\rho u\right)\\
J(T,T) &= \left(\rho C_p + \frac{d\rho}{dT}H\right)\cdot\frac{d\frac{\partial T}{\partial t}}{dT}\cdot\psi - \left(\rho C_p + \frac{d\rho}{dT}H\right)u\cdot\nabla\psi\\
&\quad + \tau_{SUPG}\cdot\nabla\psi\cdot\left(\rho C_p\frac{d\frac{\partial T}{\partial t}}{dT} + \rho C_p u\frac{d\frac{\partial T}{\partial z}}{dT}\right)
\end{aligned}
\tag{7.22}
$$

Non-diagonal terms are derived in a similar fashion and implemented in SAM, but not shown here. Note in the equation above that the dependences of the friction coefficient, f, on the primary fluid variables are neglected, but the dependences of density on the primary variables are considered.

7.3.2 Preconditioning of Heat Conduction Equation

The residual term for heat conduction equation can be given as (see Chapter 3):

$$r_{T_s} = \left(\rho C_p \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) - Q''', \ \psi\right) \tag{7.23}$$

The Jacobian term (for internal nodes) can then be derived as:

$$J(T_s, T_s) = \rho C_p \cdot \frac{d \frac{\partial T}{\partial t}}{dT} \cdot \psi + k \nabla T \cdot \nabla \psi \tag{7.24}$$

7.3.3 Preconditioning of Convective Heat Transfer Modeling

To model the convective heat transfer between fluids and structures, additional residual terms are needed, and can be given as (see Chapter 3):

$$\begin{aligned} \Delta r_{solid} &= \langle h(T_s - T_f), \psi \rangle \\ \Delta r_{fluid} &= \left(h(T_s - T_f)\frac{P_{heated}}{A_c}, \psi + \tau_{supg} \cdot \nabla \psi\right) \end{aligned} \tag{7.25}$$

Note Δ is used together with residual, indicating the residual terms in Eq. (7.22) will be added to the other residual contributions to these entries of the full residual vector.

The Jacobian contributions from conjugate heat transfer can then be derived as:

$$\begin{aligned} J(T_s, T_s) &\approx h\psi \\ J(T_s, T_f) &\approx -h\psi \\ J(T_f, T_s) &\approx -h\frac{P_{heated}}{A_c}(\psi + \tau_{supg} \cdot \nabla \psi) \\ J(T_f, T_f) &\approx h\frac{P_{heated}}{A_c}(\psi + \tau_{supg} \cdot \nabla \psi) \end{aligned} \tag{7.26}$$

In which T_f and T_s represent fluid and solid temperature, respectively. Note in Eq. (7.26) that the dependencies of heat transfer coefficient, h, on the fluid and solid variables are neglected.

For a conjugate heat transfer problem with only one fluid block and one solid block in the computational mesh, the shape of the preconditioning matrix looks like Figure 7.1, in which black lines and dots represent the non-zero entries from the fluid flow within the fluid block and the heat conduction within the solid block; and the red circles and dots represent the non-zero entries from convective heat transfer between fluid and structures. A similar study on the preconditioning of JFNK method for conjugate heat transfer problem is reported in (Zou et al., 2013).

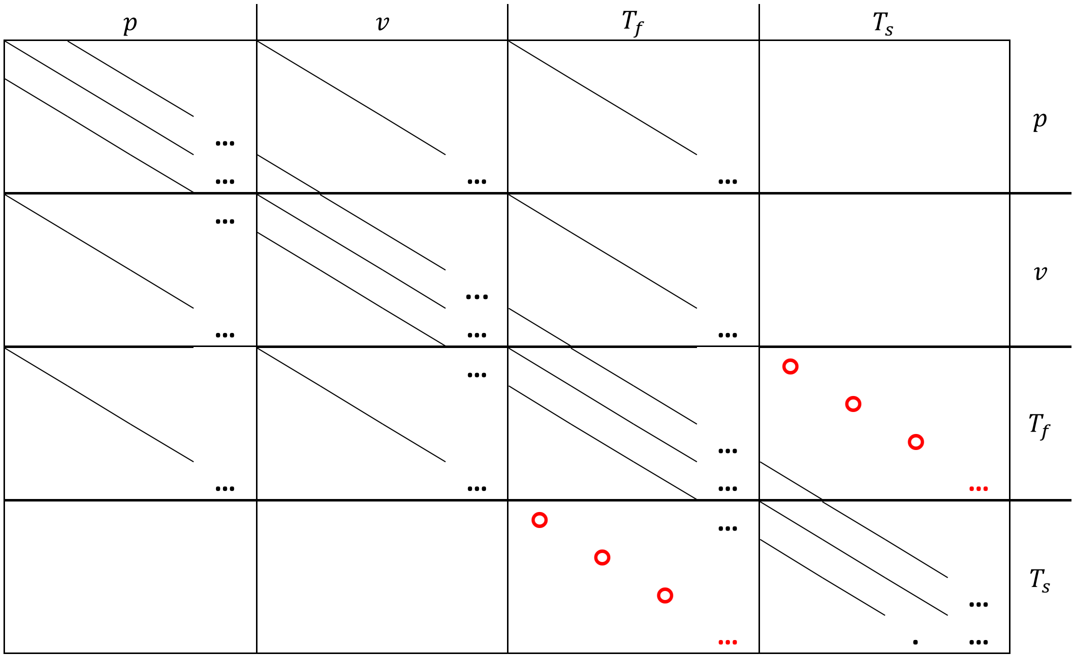


Figure 7.1: Preconditioning matrix for conjugate heat transfer problem (with lines, circles, and dots representing non-zero entries in the matrix)

8 Component Models

The physics modeling (fluid flow and heat transfer) and mesh generation of individual reactor components are encapsulated as Component classes in SAM along with some component specific models. A set of components has been developed based on the models presented in Chapter 2-4 including:

1. basic geometric components;
2. 0-D components for setting boundary conditions;
3. 0-D components for connecting 1-D components;
4. assembly components by combining the basic geometric components and the 0-D connecting components; and
5. non-geometric components for physics integration.

A brief description of major SAM components is listed in Table . The physics models associated with these components will be briefly discussed later in this Chapter.

Table 8.1: Major SAM Components

Component name	**Descriptions**	**Dimension**
PBOneDFluidComponent	Simulates 1-D fluid flow using the primitive variable based fluid model	1-D
HeatStructure	Simulates 1-D or 2-D heat conduction inside solid structures	1-D or 2-D
PBCoupledHeatStructure	The heat structure connecting two liquid components (1-D or 0-D).	1-D or 2-D
PBPipe	Simulates fluid flow in a pipe and heat conduction in the pipe wall.	1-D fluid, 1-D or 2-D structure
PBHeatExchanger	Simulates a heat exchanger, including the fluid flow in the primary and secondary sides, convective heat transfer, and heat conduction in the tube wall.	1-D fluid, 1-D or 2-D structure
PBCoreChannel	Simulates reactor core channels, including 1-D flow channel and the inner heat structures (fuel, gap, and clad) of the fuel rod.	1-D fluid, 1-D or 2-D structure
PBDuctedCoreChannel	Simulates reactor core channels with an outer heat structure of the duct wall.	1-D fluid, 1-D or 2-D structures
PBBypassChannel	Models the bypass flow in the gaps between fuel assemblies.	1-D
FuelAssembly	Models reactor fuel assemblies composed of multiple CoreChannels, representing different regions of a fuel assembly (core, gas plenum, reflector, shield, etc.).	1-D fluid, 1-D or 2-D structure

DuctedFuelAssembly	Model reactor fuel assemblies composed of multiple DuctedCoreChannels.	1-D fluid, 1-D or 2-D structure
MultiChannelRodBundle	Models the rod bundle with a multi-channel model, in which multiple CoreChannels and the inter-channel mixing are defined and created.	1-D fluid, 1-D or 2-D structure
HexLatticeCore	Models a hexagonal lattice core, in which the CoreChannels and HeatStructures are defined and created.	1-D fluid, 1-D or 2-D structure
ReactorCore	Models a pseudo three-dimensional reactor core; It consists of member core channels (with duct walls) and bypass channels.	1-D fluid, 1-D or 2-D structure
HeatPipe	Models a conventional 3-zone cylindrical heat pipe. It consists of wall, wick, and vapor core region.	2-D cylindrical heat structure
PBBranch	Models a zero-volume flow joint, where multiple 1-D fluid components are connected.	0-D
PBSingleJunction	Models a zero-volume flow joint, where only two 1-D fluid components are connected.	0-D
PBPump	Simulates a pump component, in which the pump head is dependent on a pre-defined function.	0-D
PBVolumeBranch	Considering the volume effects of a PBBranch component so that it can account for the mass and energy in-balance between inlets and outlets due to inertia.	0-D
CoverGas	A 0-D gas volume that is connected to one or multiple liquid volumes.	0-D
PBLiquidVolume	A 0-D liquid volume with cover gas (the liquid level is tracked and the volume can change during the transient).	0-D
StagnantVolume	Models a stagnant liquid volume, with connections to other 0-D volumes but no connections to 1-D fluid components.	0-D
PBTDJ	An inlet boundary in which the flow velocity and temperature are provided by pre-defined functions.	0-D
PBTDV	A boundary in which pressure and temperature conditions are provided by pre-defined functions.	0-D
CoupledTDV	A time-dependent-volume boundary in which boundary conditions are provided by other codes in coupled code simulation.	0-D
PipeChain	A non-geometric component for connecting a number of fluid components.	ND
ReactorPower	A non-geometric component describing the total reactor power.	ND
ChannelCoupling	A non-geometric component for coupling two 1-D fluid components (with energy exchange).	ND
RadiationHeatTransferCoupling	A non-geometric component for modeling the radiation heat transfer between two surfaces.	ND

8.1 PBOneDFluidComponent

PBOneDFluidComponent is the most basic fluid component in SAM. It represents a unit one-D component to simulate the one-D fluid flow in a channel. The geometry parameters such as the hydraulic diameter, flow area, and length, are provided in the input file. The wall friction and heat transfer coefficients can be calculated through the closure models based on flow conditions and geometries or provided by the user input. Internal volumetric heating (or cooling) can be specified by the user input as well. The physics models of PBOneDFluidComponent are described in Chapter 2 and 4.

8.2 HeatStructure

HeatStructure is the most basic solid structure component in SAM. It represents a unit one-D or two-D component in Cartesian or cylindrical coordinates to simulate the heat conduction in solid structures. The geometry parameters such as the thickness and length are provided in the input file. Temperature-dependent solid material properties can be provided in tabular or functional form user-supplied data. Internal volumetric heating can be specified by the user input. The physics model of HeatStructure is described in Chapter 3.

8.3 PBCoupledHeatStructure

PBCoupledHeatStructure simulates a HeatStructure with controlled boundary conditions at the two surfaces, such as adiabatic, fixed temperature, convective heat transfer with ambient, or coupled with 0-D liquid volume or 1-D liquid components. Normally users will not directly use HeatStructures to create their models, but use PBCoupledHeatStructure instead.

8.4 PBPipe

PBPipe simulates fluid flow in a pipe and heat conduction in the pipe wall. It is composed of a *PBOneDFluidComponent* and an outer *HeatStructure* (pipe wall). Convective heat transfer between the fluid and the wall is modeled, and various type of boundary conditions at the outer surface of the pipe wall can be modeled, including adiabatic, fixed temperature, fixed heat flux, convective heat transfer with ambient, or coupled with other 0-D liquid volumes or 1-D liquid components.

8.5 PBHeatExchanger

PBHeatExchanger simulates a heat exchanger, including the fluid flow in the primary and secondary sides, convective heat transfer, and the heat conduction in the tube wall. Both countercurrent and concurrent heat exchangers can be modeled. The two sides of the heat exchanger can have different orientation, lengths, flow areas, and hydraulic diameters. This gives the users more flexibilities to model a generic heat exchanger, including advanced heat exchangers being pursued by advanced reactor designs. Note that the two fluid sides of the heat exchanger and the tube wall must have the same number of elements axially.

8.6 PBCoreChannel/ PBDuctedCoreChannel and FuelAssembly/DuctedFuelAssembly

PBCoreChannel simulates the average coolant flow in rod bundles and heat conduction inside a fuel rod, as well as the convective heat transfer between the coolant and the fuel rod. It is composed of a *PBOneDFluidComponent* and a *HeatStructure*. This is also the so-called “Single-Channel” approach to model the fuel assembly. Axial power profiles and the power fractions of total reactor power can be specified for the component. If an outer structure (duct wall) is added to *PBCoreChannel*, it becomes *PBDuctedCoreChannel*, which simulates the ducted fuel assemblies as those in SFRs.

FuelAssembly or *DuctedFuelAssembly* model the reactor fuel assemblies composed of multiple *PBCoreChannels* or *PBDuctedCoreChannels*, representing different axial regions of a fuel assembly including the active core, gas plenum, lower and upper reflector, lower and upper shield, etc. The junction components (*PBSingleJunction*) are also auto-created in *FuelAssembly* or *DuctedFuelAssembly* to model the connection among the fluid parts of *PBCoreChannel* or *PBDuctedCoreChannel*.

8.7 PBBypassChannel

PBBypassChannel is just a *PBOneDFluidComponent* component with additional physics models. It is designed to model the bypass flow in the gaps between fuel assemblies. It includes the modeling of conjugate heat transfer with the neighboring fuel assembly duct walls. It can also model the direct coolant heating as a fraction of the total reactor power and using the same or different axial power shapes.

8.8 Multi-Channel Rod Bundle (MultiChannelRodBundle) Model

To improve the heat transfer between the duct wall and coolant flow, a multi-channel rod bundle model is developed in SAM to account for the temperature differences between the center region and the edge region of the coolant channel in a fuel assembly. Similar approach has been proposed in ENERGY (Yang and Joo, 1999), SAS4A/SASSYS-1 (Fanning, 2012), as well as the multi-region porous medium model reported by Yu et al. (2015). The whole fuel assembly can be divided into a number of regions, as shown in Figure . It is quite remarkable that the volumetric heat flux in region 1 is significantly less than that in other regions, based on analytical calculations. Each inner region is modeled as an average core-channel (i.e., a 1-D coolant channel and an average fuel pin). The edge region can be modeled as one core-channel or six core-channels to account for the differences in heat transfer with each side of the duct wall. This zoning strategy is also inspired from the authors' previous experiences in the CFD simulations of the triangle-lattice pin bundles. As shown in the Hu and Yu (2016), large temperature gradient were observed in the coolant region near the duct wall, but the temperature distribution elsewhere is very uniform except the hot spots due to the wire-wrap effects.

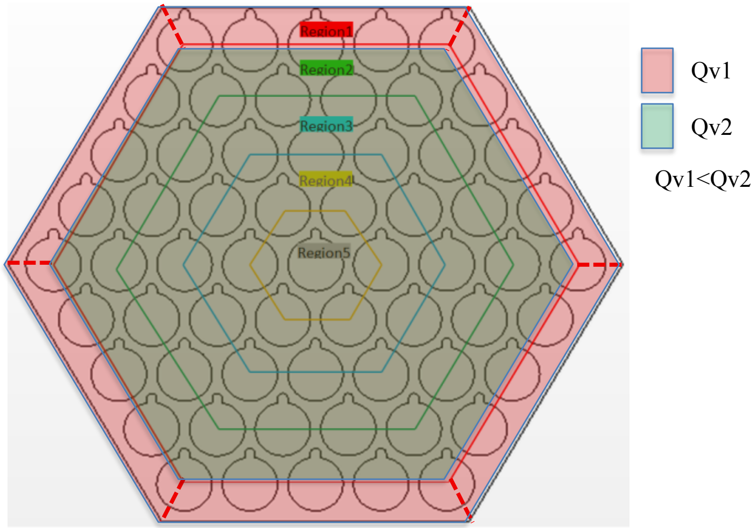


Figure 8.1: Sketch of the regions in the multi-channel model

In the SAM multi-channel model, the fluid regions are modeled as separate parallel channels with the same pressure drop. For simplicity, it is assumed that there are no mass and momentum exchange between channels. However, the energy exchange is allowed at all axial nodes, and the energy exchange rate is modeled as:

$$\frac{d\dot{Q}}{dz} = \beta\left(\rho v\right)_{avg} S\left(h_1 - h_2\right) \tag{8.1}$$

in which, β is the mixing parameter (accounting for both turbulent mixing and directional flow); $\left(\rho v\right)_{avg}$ is the average mass flux between Region 1 and 2; S the total gap width between Region 1 and 2; and h_1 and h_2 are the enthalpies of Region 1 and 2.

This term is modeled as an additional heat source term in the fluid energy conservation equation,

$$q'''_{mix} = \frac{1}{A_c}\frac{d\dot{Q}}{dz} = \beta\left(\rho v\right)_{avg}\left(h_1 - h_2\right)\frac{S}{A_c}. \tag{8.2}$$

Because the SUPG stabilization scheme is used in SAM, the associated stabilization term need to be included in the residual calculation for each finite element:

$$\Delta r_{energy} = \int_{\Omega} \psi \cdot q'''_{mix} d\Omega = \sum_{i,qp}\left(q'''_{mix} \times \left(\psi + \tau_{supg} \cdot \nabla\psi\right)\right), \tag{8.3}$$

In which, i is the index of the test functions and qp is the index of quadrature points for the element.

8.9 HexLatticeCore

HexLatticeCore models a reactor core with a hexagonal lattice such as SFRs. It can automatically generate the core lattice of *MultiChannelRodBundle* or *PBCoreChannel* components, and the inter-assembly structures (including duct walls and inter-assembly gaps), based on the geometry information specified in the input.

8.10 HeatPipe

HeatPipe models a conventional 3-zone cylindrical heat pipe, see Figure 8.2. Along the axial direction, the heat pipe is divided to 3 zones: evaporator, adiabatic, and condenser. Along the radial direction, the heat pipe consists of vapor core, wick, and wall blocks. In the current SAM modeling of the heat pipe, phase transition of the fluid is not considered. The vapor core is modeled as a perfect conductor with high thermal conductivity. This approximation is based on the fact that the thermal resistance in the vapor core is small under normal operation conditions. Flexible boundary condition options are provided in the evaporator and condenser wall outer surface to enable the coupling of heat pipe with other SAM components.

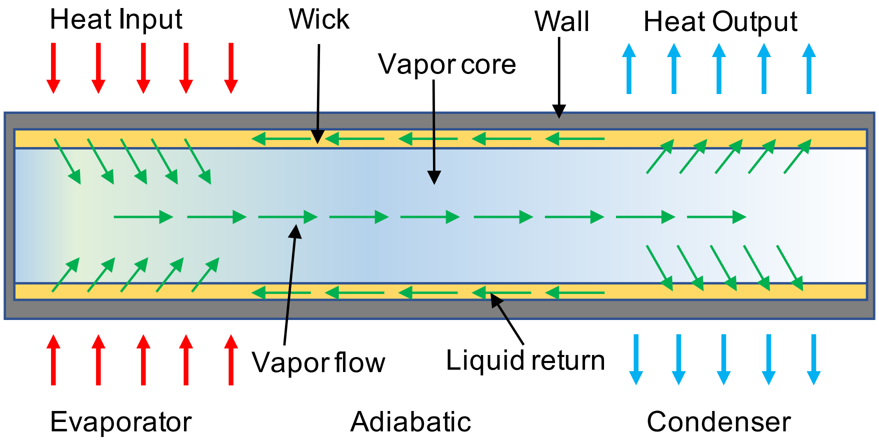


Figure 8.2: Sketch of a conventional 3-zone cylindrical heat pipe

8.11 PBBranch

PBBranch models a 0-D flow junction where multiple 1-D fluid components are connected. The governing equations of the *PBBranch* Component are:

$$\sum_{i=1}^{n} (\rho u A \vec{n})_i = 0 \tag{8.4}$$

$$u_{branch} = \frac{\sum_{i=1}^{n} (\rho u A)_i}{\rho_{branch} A_{branch}} \quad \text{if } (u \cdot \vec{n})_i > 0 \tag{8.5}$$

$$H_{branch} = \frac{\sum_{i=1}^{n} (\rho u A H)_i}{\rho_{branch} u_{branch} A_{branch}} \quad \text{if } (u \cdot \vec{n})_i > 0 \tag{8.6}$$

In which,

ρ: the density at the connecting nodes;
u: the flow velocity at the connecting nodes;
$\vec{n}$: the flow direction at the connecting nodes;
u_{branch}: the flow velocity at the branch;
A: the flow area of the connecting components;
n: the number of connecting components;
H: the enthalpy at the connecting nodes;
H_{branch}: the enthalpy at branch.

Note that simplified models are used to calculate the branch temperature and velocity.

The above models are implemented in SAM using the MOOSE *NodalConstraint* system, along with the use of scalar variables to represent the p, u, and T of the PBBranch. The boundary conditions of the connecting pipe nodes are dependent on the branch conditions: If flow into branch,

$$P_{pipe} = P_{branch} + \Delta P_{acc} + \Delta P_{form} \tag{8.7}$$

If flow out of branch,

$$\begin{aligned} P_{pipe} &= P_{branch} + \Delta P_{acc} - \Delta P_{form} \\ (\rho u)_{pipe} &= \rho_{branch} u_{pipe} \\ (\rho u H)_{pipe} &= \rho_{branch} u_{pipe} H_{branch} \end{aligned} \tag{8.8}$$

In which,

$$\Delta P_{acc} = \frac{1}{2}\left(\rho u^2\right)_{branch} - \frac{1}{2}\left(\rho u^2\right)_{pipe}$$
$$\Delta P_{form} = \frac{1}{2}K\left(\rho u^2\right)_{pipe}$$

8.12 PBSingleJunction

PBSingleJunction is a special junction component, and it models a zero-volume flow joint where only two 1-D fluid components are connected. There are currently two options in the modeling of the single junction:

- Weak formulation. In the weak formulation model, the single junction does not model the mass, momentum, and energy conservation at the junction, but to assure that the two connecting nodes (1 and 2) have consistent boundary conditions. This weak formulation is computationally efficient but does not enforce mass and energy balance well.
- Strong formulation. In the strong formulation model, the mass, momentum, and energy conservation at the junction are modeled the same way as PBBranch, see Eq. 8.4, 8.5, and 8.6. The reference area of the junction is taken as the area of the input pipe. This strong formulation helps enforce mass and energy balance.

The upstream fluid kinetic energy will be used for calculation of the form loss across this junction. Let subscript 1 denote the upstream side.

$$\begin{aligned}
&(\rho u A)_1 = (\rho u A)_2 \\
&\left(p + 0.5\rho u^2\right)_1 = \left(p + 0.5\rho u^2\right)_2 + K\cdot\left(0.5\rho u^2\right)_1 \\
&(\rho u A H)_1 = (\rho u A H)_2
\end{aligned} \tag{8.9}$$

In which, K is the form loss coefficient that can be specified by either built-in model or user-provided value. If K is a positive value, then there will be an additional positive pressure drop along the flow direction.

The built-in form loss coefficient model is based on the Borda-Carnot equation (Chanson, 2004). The Borda–Carnot equation is an empirical description of the mechanical energy losses of the fluid due to a (sudden) flow expansion:

$$\Delta E = \Delta(p + \frac{1}{2}\rho u^2) = \xi\frac{1}{2}\rho(u_1 - u_2)^2 \tag{8.10}$$

where ξ is an empirical loss coefficient between zero and one, $0 \le \xi \le 1$. Notice that the gravity effect has been ignored for the junction. For flow through a sudden expansion of pipe, the loss coefficient ξ for this sudden expansion is approximately equal to one: $\xi \approx 1.0$,

$$\Delta E = \Delta(p + \frac{1}{2}\rho u^2) = \left(1 - \frac{A_1}{A_2}\right)^2\frac{1}{2}\rho u_1^2 \tag{8.11}$$

with $A_1 \le A_2$. Thus, the equivalent form loss coefficient is

$$K_{\text{expansion}} = \left(1 - \frac{A_1}{A_2}\right)^2 \tag{8.12}$$

For flow through a sudden contraction of pipe, assuming a loss coefficient of $\xi \approx 1.0$,

$$\Delta E = \Delta(p + \frac{1}{2}\rho u^2) = \left(\frac{1}{\mu} - 1\right)\left(\frac{A_1}{A_2}\right)^2\frac{1}{2}\rho u_1^2 \tag{8.13}$$

with $A_1 \geq A_2$. And, μ is the Weisbach contraction coefficient (Moffatt, 2005),

$$\mu = 0.63 + 0.37\left(\frac{A_2}{A_1}\right)^3 \tag{8.14}$$

Thus, the equivalent form loss coefficient is

$$K_{\text{contraction}} = \left(\frac{1}{\mu} - 1\right)\left(\frac{A_1}{A_2}\right)^2 \tag{8.15}$$

8.13 PBPump

PBPump is another special junction component, and it simulates a pump component, in which the pump head can be dependent on a pre-defined function. More complex pump models will be developed in future SAM enhancements. Pumping power can be modeled and considered in the energy conservation of the junction.

$$Q_{pump} = P_{head} u_{pump} A_{pump} \tag{8.16}$$

8.14 PBVolumeBranch

PBVolumeBranch considers the volume effects of a junction component so that it can account for the mass and energy in-balance between inlets and outlets due to inertia. The governing equations of the mass and energy conservation for the *PBVolumeBranch* can be given as:

$$-\frac{d(\rho V)}{dt} + \sum_{i=1}^{n} (\rho u A \vec{n})_i = 0 \tag{8.17}$$

$$-\frac{d(\rho H V)}{dt} + \sum_{i=1}^{n} (\rho u A H \vec{n})_i = 0 \tag{8.18}$$

In which,

ρ: the density at the *PBVolumeBranch* component;

V: the volume of the *PBVolumeBranch* component;

t: time.

The momentum energy conservation is more difficult to model in this 0-D component. So instead, the simplified model (Eq. (8.5)) is used to account for various pressure losses in the *PBVolumeBranch* component. The boundary condition modeling for the connected fluid components of the *PBVolumeBranch* is the same as the *PBBranch*, except that the gravity pressure drop is considered in *PBVolumeBranch*. If flow into branch,

$$P_{pipe} = P_{branch} + \Delta P_{acc} + \Delta P_{form} + \Delta P_{grav} \tag{8.19}$$

If flow out of branch,

$$\begin{aligned} P_{pipe} &= P_{branch} + \Delta P_{acc} - \Delta P_{form} + \Delta P_{grav} \\ \Delta P_{grav,i} &= \rho_{branch} \Delta H_i \end{aligned} \tag{8.20}$$

Note that the friction loss is neglected in the model. It is because the friction is dependent on the flow path, and it is very difficult (and unphysical) to model the friction loss in the 0-D component. On the other hand, the friction loss in a large volume is always very small. If it is deemed important, the orifice coefficient can be adjusted to account for it.

8.15 PBLiquidVolume

PBLiquidVolume is a special *PBVolumeBranch*, in which the volume can change, and the liquid level is tracked during the transient. Therefore, an additional scalar variable, liquid level (L), and an auxiliary scalar variable, liquid volume (V), are added in the model.

In addition to the mass conservation, the liquid level and liquid volume have the following constraints:

$$L = \frac{P_{vol} - P_{gas}}{\rho_{vol} g} \tag{8.21}$$

$$V = V_{ref} + \left(L - L_{ref}\right) A_{ref} \tag{8.22}$$

In which, P_{gas} is the cover gas pressure or atmosphere pressure (if it is open to the atmosphere).

The governing equations for the liquid volume physics can be re-organized as:

$$-\left(\rho_{vol} A \frac{dL}{dt} + V \frac{d\rho}{dT}\frac{dT}{dt}\right) + \sum_{i=1}^{n} \left(\rho u A \vec{n}\right)_i = 0 \tag{8.23}$$

$$-\left(\rho_{vol} H_{vol} A \frac{dL}{dt} + V\left(H \frac{d\rho}{dT} + \rho C_p\right)\frac{dT}{dt}\right) + \sum_{i=1}^{n} \left(\rho h u A \vec{n}\right)_i = 0 \tag{8.24}$$

$$P_{vol} = P_{gas} + \rho_{vol} g L \tag{8.25}$$

$$u_{vol} = \frac{\sum_{i=1}^{n} \left(\rho u A\right)_i}{\rho_{vol} A_{vol}} if \left(u \cdot \vec{n}\right)_i > 0 \tag{8.26}$$

In which, P, T, L, and u_{vol} are the primary state scalar variables; and ρ_{vol} and V are auxiliary scalar variables.

As discussion in Chapter 5, to use the JFNK method to solve the nonlinear system equations, it is important to provide approximate (diagonal and off-diagonal) Jacobian terms in the preconditioning matrix for fast convergence. The Jacobian term, $J\left(x_i, x_j\right)$, can be derived as the first order derivative of the Residuals of variable x_i over the variable x_j,

$$J\left(x_i, x_j\right) = \left(R_{x_i}\right)_{x_j} = \frac{\partial R_{x_i}}{\partial x_j} \tag{8.27}$$

Based on the governing equations above, the individual Jacobian terms can be derived:

$$\begin{aligned}
&J\left(L, L\right) = -\rho_{vol} A \frac{d\dot{L}}{dL} - A \frac{d\rho}{dT}\frac{dT}{dt} \\
&J\left(L, T_{vol}\right) = -\frac{d\rho}{dT} A \frac{dL}{dt} - V \frac{d\rho}{dT}\frac{d\dot{T}}{dT} \\
&J\left(L, p_i\right) = 0 \\
&J\left(L, u_i\right) = \rho_i A_i \cdot \vec{n}_i \\
&J\left(L, T_i\right) = 0
\end{aligned} \tag{8.28}$$

In which:

p_i: pressure at the connecting nodes;
u_i: flow velocity at the connecting nodes;
T_i: temperature at the connecting nodes.

$$
\begin{aligned}
&J\left(T_{vol},T_{vol}\right) = -\left(h\frac{d\rho}{dT}+\rho C_p\right)A\frac{dL}{dt} - V\left(h\frac{d\rho}{dT}+\rho C_p\right)\frac{d\dot{T}}{dT} - V\frac{dT}{dt}\left(2\frac{d\rho}{dT}C_p+\rho\frac{dC_p}{dT}\right)\\
&J\left(T_{vol},L\right) = -\rho_{vol}h_{vol}A\frac{d\dot{L}}{dL} - A\left(h\frac{d\rho}{dT}+\rho C_p\right)\frac{dT}{dt}\\
&J\left(T_{vol},p_i\right) = 0\\
&J\left(T_{vol},u_i\right) = \rho_i A_i h_i\cdot\vec{n}_i\\
&J\left(T_{vol},T_i\right) = \rho_i A_i u_i C_{p,i}\cdot\vec{n}_i
\end{aligned}
\tag{8.29}
$$

$$
\begin{aligned}
&J\left(u_{vol},u_{vol}\right) = 1\\
&J\left(u_{vol},p_i\right) = 0\\
&J\left(u_{vol},u_i\right) = \begin{cases}\dfrac{(\rho A\vec{n})_i}{\rho_{vol}A_{vol}}, & (u\vec{n})_i > 0\\ 0, & (u\vec{n})_i < 0\end{cases}\\
&J\left(u_{vol},T_i\right) = 0
\end{aligned}
\tag{8.30}
$$

$$
\begin{aligned}
&J\left(P_{vol},P_{vol}\right) = 1\\
&J\left(P_{vol},T_{vol}\right) = -\frac{d\rho}{dT}gL\\
&J\left(P_{vol},L\right) = -\rho_{vol}g\\
&J\left(P_{vol},\ p_i \text{ or } u_i \text{ or } T_i\right) = 0
\end{aligned}
\tag{8.31}
$$

The other unmentioned off-diagonal Jacobian terms are neglected, such as

$$J\left(x_{i,volume},x_{j,volume}\right)|_{i\neq j} = 0.0 \tag{8.32}$$

As the set of scalar variables used in the *PBLiquidVolume* include all those used in *PBVolumeBranch* and *PBBranch* component. The same set of boundary condition classes can be used to adjust the residual and Jacobian calculation at the end nodes of the connecting pipes of the liquid volume.

8.16 CoverGas

CoverGas component is always used together with *PBLiquidVolume* component. It models a 0-D gas volume that is connected to one or multiple liquid volumes. The gas volume is modeled as an ideal gas, and the heat transfer between the cover gas and the liquid volumes is neglected. Its volume change is decided by the volume changes of all connecting liquid volumes. P_{gas} is an auxiliary scalar variable, and it does not have its own residual or Jacobian entries.

$$\Delta V_{gas} = \sum \Delta V_{liquid\_volume,i} \tag{8.33}$$

$$P_{gas}V_{gas}^{\gamma} = \text{constant} \tag{8.34}$$

8.17 StagnantVolume

StagnantVolume models a stagnant liquid volume, which has no connections to 1-D fluid components but is allowed to connect to a 0-D volume or 1-D or 2-D heat structures for heat transfer. It is assumed that there is no net mass transfer between *StagnantVolume* and the connecting 0-D volumes. The governing equation of the energy conservation for the StagnantVolume can be given as:

$$\frac{d\left(\rho V H\right)}{dt} + \sum_{i=1}^{n} \dot{Q}_i = 0 \tag{8.35}$$

In which,

ρ: average density of the *StagnantVolume* component;
V: total volume of the component;
H: average enthalpy of the volume component;
t: time;
n: the number of coupling heat transfer components;
$\dot{Q}$: heat transfer with coupled heat structures or 0-D volumes.

For convective heat transfer with heat structures,

$$\dot{Q} = \int h_{conv}\left(T_w - T_{vol}\right) dA. \tag{8.36}$$

In which,

h_{conv}: convective heat transfer coefficient;
T_w: structure wall temperature;
T_{vol}: volume temperature.

For heat transfer with other 0-D volumes through thermal mixing,

$$\dot{Q} = \dot{m}_{mix} \Delta H \tag{8.37}$$

In which,

$\dot{m}_{mix}$: the effective mixing flow between 0-D volumes;
ΔH: enthalpy differences between 0-D volumes.

8.18 PBTDJ

PBTDJ is an inlet boundary component in which the flow velocity and temperature are provided by user-defined (time-dependent) functions. It provides boundary conditions to the connecting 1-D fluid components.

8.19 PBTDV

PBTDV is a boundary component in which the pressure and temperature are provided by user-defined (time-dependent) functions. It provides boundary conditions to the connecting 1-D fluid components. Note if the flow is flowing into the *PBTDV*, the temperature boundary condition will not be used by the connecting fluid components.

8.20 CoupledTDV

CoupledTDV is a special *PBTDV* boundary in which boundary conditions are provided by other codes in coupled code simulation.

8.21 PipeChain

PipeChain is a non-geometric component for sequentially connecting a number of fluid components. It will auto-generate the needed *PBSingleJunction* components between the specified fluid components. It is developed for user friendliness.

8.22 ReactorPower

ReactorPower is a non-geometric component for describing the total reactor power, which can be dependent on user-defined functions (such as describing the decay heat curve). The total reactor power variable is used in core components such as *PBCoreChannel* and *PBBypassChannel*.

8.23 ChannelCoupling

ChannelCoupling is a non-geometric component for coupling two 1-D fluid components (with energy exchange). It is intended to model the flow mixing between two parallel channels, using the models described in Equations (8.1) - (8.3).

8.24 RadiationHeatTransferCoupling

RadiationHeatTransferCoupling is a non-geometric component for modeling of the radiation heat transfer between two surfaces. The physics model is described in Chapter 3.

9 Multi-Dimensional Fluid Model

In SAM, a unified multi-dimensional flow model has been developed and implemented to consider both the general multi-dimensional flow model and the porous medium-based flow model. In this chapter, we will discuss the governing equations, types of boundary conditions, and code implementation of the SAM multi-dimensional model.

9.1 Flow Equations

In this section, we will start with the general multi-dimensional flow model in Subsection 9.1.1, followed by a discussion of the porous-medium flow model in Subsection 9.1.2. A unified flow model based on these two flow models will be discussed in Subsection 9.1.3.

9.1.1 General Multi-dimensional Flow Model

For the general multi-dimensional flow model, the set of balance equations for fluid flow in three dimensions includes the mass, momentum, and energy balance equations:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) = 0, \tag{9.1}$$

$$\rho \frac{\partial \mathbf{V}}{\partial t} + \rho (\mathbf{V} \cdot \nabla)\mathbf{V} + \nabla p - \rho \mathbf{g} - \nabla \cdot \tau = 0, \tag{9.2}$$

$$\rho c_p \frac{\partial T}{\partial t} + \rho c_p \mathbf{V} \cdot \nabla T - \nabla \cdot (k \nabla T) - q''' = 0, \tag{9.3}$$

where $\mathbf{V} = [V_x, V_y, V_z]^{\mathrm{T}}$ is the velocity vector. The convection term in the momentum equation, i.e., $\rho(\mathbf{V} \cdot \nabla)\mathbf{V}$, can be explicitly expressed as:

$$\rho (\mathbf{V} \cdot \nabla)\mathbf{V} = \begin{bmatrix} \rho \mathbf{V} \cdot (\nabla V_x) \\ \rho \mathbf{V} \cdot (\nabla V_y) \\ \rho \mathbf{V} \cdot (\nabla V_z) \end{bmatrix} \tag{9.4}$$

If compressibility effects are considered, following Batchelor (1967), the viscous tensor, τ, can be expressed as

$$\tau = \mu \left[\nabla \mathbf{V} + (\nabla \mathbf{V})^{\mathrm{T}} - \frac{2}{3} (\nabla \cdot \mathbf{V}) \mathbf{I} \right], \tag{9.5}$$

and therefore,

$$\nabla \cdot \tau = \mu \nabla^2 \mathbf{V} + \frac{1}{3} \mu \nabla (\nabla \cdot \mathbf{V}), \tag{9.6}$$

where $\nabla \mathbf{V}$ is a second order tensor with $[\nabla \mathbf{V}]_{ij} = \partial V_i / \partial x_j$, for $i, j = x, y, z$; and $\mathbf{I}$ is the identity tensor. The Laplacian term, $\mu \nabla^2 \mathbf{V}$, is defined as

$$\mu \nabla^2 \mathbf{V} = \mu \nabla \cdot (\nabla \mathbf{V}) = \mu \begin{bmatrix} \nabla^2 V_x \\ \nabla^2 V_y \\ \nabla^2 V_z \end{bmatrix}. \tag{9.7}$$

It is clear that the divergence term, $\nabla \cdot \mathbf{V}$, is non-zero for compressible flows where the fluid density depends on pressure, or for thermally expandable flows where the fluid density depends on temperature. Typically, this term only contributes to negligible effects for low-speed flow conditions. For applications of interest to SAM simulations, which typically do not involve highly compressible flow conditions, it is reasonable to ignore this term, and thus,

$$\nabla \cdot \tau = \mu \nabla^2 \mathbf{V}\,. \tag{9.8}$$

With this equality, the momentum equation (Equation 9.2) can be written as

$$\rho\frac{\partial\mathbf{V}}{\partial t}+\rho(\mathbf{V}\cdot\nabla)\mathbf{V}+\nabla p-\rho\mathbf{g}-\mu\nabla^2\mathbf{V}=0. \tag{9.9}$$

Although it is not our intention to develop SAM as a CFD tool, turbulence effects must be considered to obtain physically plausible and numerically stable solutions. There are several turbulence models available, ranging from the very simple zero-equation models to the more complex and widely used two-equation models (i.e., $k-\varepsilon$ and $k-\omega$), and even more complex turbulence models with more equations. Zero-equation models are easy to implement and computationally less expensive; however, they generally can only be applied under very simple flow conditions. More complex turbulence models, such as the $k-\varepsilon$ and $k-\omega$ models, are more attractive options because of their larger validation domains; however, these models are more computationally expensive, and therefore, are not considered in the current SAM development.

Nevertheless, when turbulence effects are considered, the molecular viscosity μ is replaced with the effective viscosity μ_{eff}. The effective viscosity includes the contributions from both the molecular-viscosity and turbulent-viscosity effects, i.e.,

$$\mu_{eff}=\mu+\mu_t, \tag{9.10}$$

where μ_t is the turbulent viscosity. Furthermore, the effects of turbulent flow on heat dissipation in the energy equation also need to be considered. In a similar manner, the fluid thermal conductivity k is replaced with an effective value, i.e.,

$$k_{eff}=k+k_t, \tag{9.11}$$

where k_t is the enhanced thermal conductivity due to turbulence effects, and is related to the turbulent viscosity by

$$k_t=\frac{c_p\mu_t}{Pr_t}, \tag{9.12}$$

where c_p is the specific heat and Pr_t the turbulent Prandtl number. While different values for Pr_t have been reported by various resources (e.g., 0.85 by FLUENT), in this study we use the default value of 0.9 as reported by Yoder (2016).

Currently, SAM provides two options to represent the turbulent viscosity. The first option is a simple zero-equation model implemented in SAM that computes the turbulent viscosity on the basis of Prandtl's mixing-length theory (Iaccarino (2004)):

$$\mu_t=\rho L_{mix}^2\sqrt{2\sum_{i,j=1}^{i,j=3}S_{ij}S_{ij}}, \tag{9.13}$$

where $S_{ij}=\mathbf{S}=\frac{1}{2}\left[\nabla\mathbf{V}+(\nabla\mathbf{V})^{\mathrm{T}}\right]$ is the rate-of-strain tensor. It is noted that more complex zero-equation models exist, which consider wall effects (e.g., the Cebeci-Smith and Baldwin-Lomax models) (Alfonsi (2009)); however, these models were not implemented. Moreover, the second option in SAM allows the user to *import* the turbulent viscosity from an external source. These turbulent-viscosity distributions can be obtained from high-resolution CFD simulations or from machine learning models.

9.1.2 Porous-Medium Flow Model

To address the modeling and simulation needs of pebble-bed reactors such as high-temperature gas-cooled or fluoride-salt-cooled reactors, a porous-medium-based multi-dimensional flow model (porous-medium flow model) has recently been implemented in the SAM code. In nuclear reactor thermal-hydraulic analyses, it is a common approach to utilize porous-medium flows to model the fluid flow and heat transfer in very complex geometries, such as the pebble-bed cores and tube bundles in steam generators. This subsection provides a brief discussion of the porous-medium flow model currently implemented in SAM.

For the porous-medium flow model, we follow the models presented in Nield and Bejan (2013) with some necessary extensions. The complete set of balance equations includes the mass, momentum, and energy equations for the fluid phase, as well as the energy equation for the solid phase:

$$\varepsilon\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\mathbf{v})=0, \tag{9.14}$$

$$\rho\frac{\partial\mathbf{v}}{\partial t}+\frac{\rho}{\varepsilon}(\mathbf{v}\cdot\nabla)\mathbf{v}+\varepsilon\nabla p-\varepsilon\rho\mathbf{g}+\beta\mathbf{v}+\alpha|\mathbf{v}|\mathbf{v}=0, \tag{9.15}$$

$$\varepsilon\rho c_p\frac{\partial T}{\partial t}+\rho c_p\mathbf{v}\cdot\nabla T-\nabla\cdot(\varepsilon k\nabla T)-q'''+a_wh(T-T_s)=0, \tag{9.16}$$

$$(1-\varepsilon)\rho_sc_{p,s}\frac{\partial T_s}{\partial t}-\nabla\cdot(k_{s,eff}\nabla T_s)-q_s'''+a_wh(T_s-T)=0, \tag{9.17}$$

where subscript s denotes the solid phase, ε is the porosity of the pebble bed, and $\mathbf{v}$ is the so-called superficial velocity, which is related to the intrinsic velocity $\mathbf{V}$ by $\mathbf{v}=\varepsilon\mathbf{V}$. For the mass balance equation (Equation 9.14), we follow the same Equation 1.1 presented in Nield and Bejan (2013). For the momentum equation, we use Equation 1.8 from Nield and Bejan (2013) as the basic model, and extend the drag term to include both the linear viscous (Darcy) and quadratic (Forchheimer) terms. Other higher-order partial-derivative terms, such as the Brinkman term analogous to the Laplacian term in the Navier-Stokes equation, are not included. For fluid and solid energy equations, we follow Equations 2.2 and 2.1 of Nield and Bejan (2013), respectively. It is noted that, for the diffusion term of the fluid energy equation, the porosity ε is moved into the divergence, which seems more appropriate if local averaged fluid thermal conductivity is assumed to be εk (see discussion in Section 3.2.1 of Kaviany (1995)). Nevertheless, the contribution of this conduction term to energy transport is normally trivial compared to the convection term, especially under convection-dominant conditions. It is also noted that, for the diffusion term of the solid energy equation, instead of using the simple form of $(1-\varepsilon)\nabla\cdot(k_s\nabla T_s)$ given in Nield and Bejan (2013), we use $\nabla\cdot(k_{s,eff}\nabla T_s)$, which is a more sophisticated approach and requires an additional closure correlation to model $k_{s,eff}$, the effective thermal conductivity of porous media. The last term in both the fluid-flow energy and solid-phase energy equations are the convective heat transfer term, in which a_w is the heating surface area density per unit volume, and h is the heat transfer coefficient.

9.1.3 A Unified Model

As clearly shown in the previous two subsections, the general multi-dimensional fluid flow model and the porous-medium flow model have many similarities. In many applications, there is also a need to include both flow models in the same simulations. For example, in simulations of HTGR core multi-dimensional flows, it is typical to model flows in both the top and bottom plena using the general multi-dimensional flow model and flow through the pebble-bed core using the porous-medium approach. Therefore, a unified model based on these two sets of flow equations has been developed to reduce the effort of code implementation, as well as to ease the model input process when both flow models are needed in the same simulation.

For the unified flow model, the fluid mass, momentum, and energy balance equations are given as follows:

$$\varepsilon\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\mathbf{v})=0, \tag{9.18}$$

$$\rho\frac{\partial\mathbf{v}}{\partial t}+\frac{\rho}{\varepsilon}(\mathbf{v}\cdot\nabla)\mathbf{v}+\varepsilon\nabla p-\varepsilon\rho\mathbf{g}+\mathbf{F}_m=0, \tag{9.19}$$

$$\varepsilon\rho c_p\frac{\partial T}{\partial t}+\rho c_p\mathbf{v}\cdot\nabla T-\nabla\cdot(\varepsilon k\nabla T)-q'''+S_e=0. \tag{9.20}$$

The source term, $\mathbf{F}_m$, in the momentum equation is given as

$$\mathbf{F}_m=\begin{cases}-\mu\nabla^2\mathbf{V} & \text{(general model)}\\ \beta\mathbf{v}+\alpha|\mathbf{v}|\mathbf{v} & \text{(porous-medium model)}\end{cases}, \tag{9.21}$$

and the additional source term in the energy equation, S_e, in the fluid energy equation is given as

$$S_e=\begin{cases}0 & \text{(general model)}\\ a_w h(T-T_s) & \text{(porous-medium model)}\end{cases}. \tag{9.22}$$

It is noted that, if $\varepsilon=1$, this model exactly reduces to the general multi-dimensional flow model given in Subsection 9.1.1. For the simulation of a porous medium, where the solid structure is also modeled, the same solid-phase energy equation, Equation 9.17, is used.

9.2 Boundary Conditions

In this section, the physical boundary conditions are discussed. Their mathematical form and implementation within the FEM will be discussed in the coming sections. We will concern ourselves with a limited set of boundary conditions of engineering interest to SAM simulations, and by no means do we intend to list all possibilities. This limited set of boundary conditions is discussed in the following subsections.

9.2.1 Inlet

For an inlet flow condition, the magnitude of the inlet velocity $|\mathbf{v}_{in}|$ and inlet temperature T_{in} are to be specified. Both parameters could be constant values, or functions of time and/or position. With a given inlet velocity magnitude, local density computed from pressure (as part of the solution), and temperature (from boundary conditions), the inlet mass flux can then be computed. The three velocity components for the inlet flow will also be specified for the momentum equations, and obviously they have to be consistent with the specified value for the magnitude of inlet velocity.

9.2.2 Pressure Outlet

For an outlet flow condition, it is typical to specify the outlet pressure p_{out} , which could be a constant value or a function of time and/or position. No fluid temperature should be given unless a reverse flow condition is expected on part of or on the entire outlet boundary during a transient, which, if possible, should be avoided by moving the pressure outlet further downstream. In addition, it is generally acceptable to deem the viscous effects unimportant, and to allow them to be ignored at the pressure outlet boundary. The same argument could be directly applied to the energy equation, such that the effect of the diffusion term on the boundary is ignored.[1]

---

[1] First, in the context of incompressible flows, an outlet boundary condition is generally a different concept than a prescribed pressure boundary condition. An outlet boundary condition is normally ascribed to an outlet that is far in the downstream direction, where the gradients of all variables (except pressure) are zero (Versteeg and Malalasekera (2013)). In general, it is not necessarily true that the velocity gradients will be zero on a prescribed pressure boundary. However, it is generally safe to assume that those gradients are zero on the "transparent" outlet boundary for the advection-diffusion equation (Griffiths (1997); Halpern (1986)), and it was found that the smaller the viscosity, the smaller the introduced error (Halpern (1986)). The momentum equation is indeed an advection-diffusion equation, and the viscosity, which is the kinematic viscosity for the momentum equation, is generally very small. A similar argument could be made for the energy equation. To the energy equation, the thermal diffusivity is analogous to the kinematic viscosity in the momentum equation. Thermal diffusivities of typical fluids of interest to our simulations are also generally very small, and therefore it is acceptable to use the "transparent" boundary condition as well.

9.2.3 Wall

For solid walls, we apply a non-penetrating condition, such that the mass flux normal to the wall is zero. For the energy equation, several different boundary conditions could be applied. The simplest one to apply is a Dirichlet type of boundary condition for the fluid temperature. A Neumann type of boundary condition can also be applied, in which a wall heat flux is required. An adiabatic condition is a special Neumann boundary condition, where the temperature gradient is zero. For the momentum equation, two types of boundary conditions are considered:

- Non-slip wall

  All velocity components are simply set to zero.

- Slip wall

  Viscous effects at the wall can be neglected (viscous stress-free boundary). It is noted that a rigorous implementation of generic slip-wall boundary conditions on a wall, using an arbitrary geometry, is not a trivial task (Verfürth (1986, 1991); Mucha (2003)). A viscous stress-free boundary is really an engineering simplification, and users should be cautious of the potential inaccuracies that could arise from such a simplification.

9.3 FEM Implementation

As a MOOSE-based application, SAM uses a FEM to obtain the discretized equations, which are then solved with the Jacobian-free Newton-Krylov (JFNK) method provided by the MOOSE framework. Instead of directly solving the PDEs, SAM solves the so-called "weak form" of the PDEs. Using FEM, the weak forms of the discretized equations are obtained by multiplying the governing equations by a test function, ψ, and then integrating over the domain. Here, we use the mass balance equation (Equation 9.18) as an example to introduce several FEM concepts. By using integration by parts and applying the Divergence Theorem, we obtain the weak form for the mass balance equation as

$$\begin{aligned}&\int_{\Omega}\psi\left[\varepsilon\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\mathbf{v})\right]d\Omega\\&=\int_{\Omega}\psi\varepsilon\frac{\partial\rho}{\partial t}d\Omega+\int_{\partial\Omega}\psi\rho\mathbf{v}\cdot\hat{\mathbf{n}}d\Gamma-\int_{\Omega}\nabla\psi\cdot\rho\mathbf{v}d\Omega,\\&=0,\end{aligned} \tag{9.23}$$

where Ω is the domain of interest (i.e., volume), $\partial\Omega$ is the boundary of the domain (i.e., boundary surface), and $\hat{\mathbf{n}}$ is the outward-pointing unit normal of the boundary surfaces. In the context of FEM, it is common to use $(,)$ to denote a volume integral, and $\langle,\rangle$ to denote a surface integral. It is also typical to write the discretized equations in the residual form when the JFNK method is used.

Using the FEM notations, the residual forms of the unified multi-dimensional flow equations are given as follows:

$$r'_c=\left(\varepsilon\frac{\partial\rho}{\partial t},\psi\right)+\left(-\rho\mathbf{v},\nabla\psi\right)+\left\langle\rho\mathbf{v}\cdot\hat{\mathbf{n}},\psi\right\rangle, \tag{9.24}$$

$$\mathbf{r}'_m=\left(\rho\frac{\partial\mathbf{v}}{\partial t},\psi\right)+\left(\frac{\rho}{\varepsilon}(\mathbf{v}\cdot\nabla)\mathbf{v},\psi\right)+\left(\varepsilon\nabla p,\psi\right)+\left(-\varepsilon\rho\mathbf{g},\psi\right)+F_m, \tag{9.25}$$

$$\begin{aligned}r'_e=&\left(\varepsilon\rho c_p\frac{\partial T}{\partial t},\psi\right)+\left(\rho c_p\mathbf{v}\cdot\nabla T,\psi\right)+\left(\varepsilon k\nabla T,\nabla\psi\right)\\&+\left(-q'''+S_e,\psi\right)+\left\langle-\varepsilon k\nabla T\cdot\hat{\mathbf{n}},\psi\right\rangle,\end{aligned} \tag{9.26}$$

and the residual form of the solid-phase energy equation (if applicable) is given as

$$\begin{aligned} r_s =& \Big((1-\varepsilon)\rho_s c_{p,s} \frac{\partial T_s}{\partial t}, \psi \Big) + \Big(k_{s,eff} \nabla T_s, \nabla \psi \Big) \\ &+ \Big(-q''' + S_e, \psi \Big) + \Big\langle -k_{s,eff} \nabla T_s \cdot \hat{\mathbf{n}}, \psi \Big\rangle . \end{aligned} \tag{9.27}$$

In the residual equations discussed above, the superscript "′" indicates that they are not in their final forms, as additional stabilization terms will be added (see next section). The subscripts *c*, *m*, *e*, and *s* denote the mass balance (continuity), momentum, fluid flow energy, and solid-phase energy equations, respectively.

The source term in its weak form, F_m, in the residual form of the momentum equation is

$$F_m = \begin{cases} \Big(\mu \nabla \mathbf{v}, \nabla \psi \Big) + \Big\langle -\mu \nabla \mathbf{v} \cdot \hat{\mathbf{n}}, \psi \Big\rangle & \text{(general model)} \\ \Big(\beta \mathbf{v} + \alpha |\mathbf{v}| \mathbf{v}, \psi \Big) & \text{(porous-medium model)} \end{cases} . \tag{9.28}$$

9.3.1 Stabilization Scheme

It is well understood that stabilization schemes are necessary for the standard Continuous Galerkin FEM to regulate spurious spatial numerical oscillations, when it is applied to convection-dominant problems. In SAM, the SUPG and PSPG schemes are both implemented to resolve numerical instability issues. Both schemes introduce additional terms into the original weak form of the balance equations. For the mass balance equation, an additional PSPG term is added:

$$r_c = r_c' + \Big(R_m, \tau_{\text{PSPG}} \nabla \psi \Big), \tag{9.29}$$

and for the momentum and energy equations, an additional SUPG term is added for each equation:

$$r_m = r_m' + \Big(R_m, \tau_{\text{SUPG,m}} \mathbf{v} \cdot \nabla \psi \Big), \tag{9.30}$$

$$r_e = r_e' + \Big(R_e, \tau_{\text{SUPG,e}} \mathbf{v} \cdot \nabla \psi \Big), \tag{9.31}$$

where R_m and R_e are the raw residual forms of the momentum and the energy equation, respectively. These equations are the original PDEs without being multiplied by the test function or integrated by parts, i.e.,

$$R_m = \rho \frac{\partial \mathbf{v}}{\partial t} + \frac{\rho}{\varepsilon} (\mathbf{v} \cdot \nabla) \mathbf{v} + \varepsilon \nabla p - \varepsilon \rho \mathbf{g} + F_m, \tag{9.32}$$

and

$$R_e = \varepsilon \rho c_p \frac{\partial T}{\partial t} + \rho c_p \mathbf{v} \cdot \nabla T - \nabla \cdot (\varepsilon k \nabla T) - q''' + S_e. \tag{9.33}$$

The stabilization parameters are given as:

$$\tau_{\text{PSPG}} = \left[\left(\frac{2}{\Delta t} \right)^2 + \left(\frac{2V}{h} \right)^2 + \left(\frac{4\nu}{h^2} \right)^2 \right]^{-1/2}, \tag{9.34}$$

$$\tau_{\text{SUPG,m}} = \left[\left(\frac{2}{\Delta t} \right)^2 + \left(\frac{2|\mathbf{v}|}{h} \right)^2 + \left(\frac{4\nu}{h^2} \right)^2 \right]^{-1/2}, \tag{9.35}$$

and

$$\tau_{\mathrm{SUPG,e}} = \left[\left(\frac{2}{\Delta t} \right)^2 + \left(\frac{2|\mathbf{v}|}{h} \right)^2 + \left(\frac{4\alpha}{h^2} \right)^2 \right]^{-1/2}, \tag{9.36}$$

where V is a reference velocity magnitude, which, if not provided, is by default the local velocity magnitude $|\mathbf{v}|$; ν the kinematic viscosity; α the thermal diffusivity; Δt the time-step size; and h the local mesh size (normally taken as the smallest edge length of the local mesh element). For more details on the PSPG/SUPG theory, implementation, and additional references, we refer the reader to Hu (2019). It is also noted that the solid-phase energy equation (Equation 9.17) for the original PDE, and Equation 9.27 for its weak form, do not require stabilization schemes, as they are simply heat conduction equations.

9.4 Closure Correlations

In this section, closure correlations for pressure drop and heat transfer in porous media will be discussed with a main focus on pebble bed applications.

9.4.1 Frictional Pressure Drop in Pebble Beds

It was Blake (Blake (1922)) who first successfully treated the pressure drop problem in porous beds (or "packing materials" as referred to in the original article). Blake suggested a pair of two non-dimensional parameters (Blake (1922); Ergun (1952); Fenech (2013)):

$$f_k \equiv \frac{\Delta p}{\rho u^2} \frac{d_p}{L} \frac{\varepsilon^3}{1-\varepsilon} \tag{9.37}$$

and

$$\mathrm{Re}_m \equiv \frac{\rho u d_p}{\mu\,(1-\varepsilon)} \tag{9.38}$$

and proposed that f_k could be correlated as a function of Re_m, that is, $f_k = f(\mathrm{Re}_m)$. The term f_k was referred to as the Blake-type friction factor by Ergun, and Re_m has been referred to as the modified Reynolds number by many authors. Most empirical correlations proposed to predict frictional pressure drop through pebble beds have followed such a formula.

There are also different ways to define the non-dimensional parameters. Two additional such parameters are introduced here as they are closely related to the validation study presented in this report. In some studies, for example, the KTA correlation (Fenech (2013); Nuclear Safety Standards Commission (1981)), the non-dimensional pressure drag coefficient is given as

$$\Psi \equiv \frac{\Delta p}{\frac{1}{2}\rho u^2} \frac{d_p}{L} \frac{\varepsilon^3}{1-\varepsilon}, \tag{9.39}$$

and it is obvious that,

$$\Psi = 2 f_k;\ \text{and}\ f_k = \Psi/2. \tag{9.40}$$

Currently, we have selected and implemented three empirical correlations to predict frictional pressure drop through porous pebble beds: (1) the classical Ergun correlation; (2) the KTA correlation, which is widely used in pebble-bed reactor analysis applications; and (3) the Eisfeld and Schnitzlein correlation, which explicitly considers the wall effect on pressure drop in pebble beds.

The **Ergun correlation** represents a classical correlation for predicting frictional pressure drop across packed pebble beds. It was developed based on 640 experimental data points on various-sized spheres, sand, pulverized coke, and various gases (carbon dioxide, nitrogen, methane, and hydrogen), and considered pressure drop through porous packed beds as a result of simultaneous kinetic and viscous energy losses (Ergun (1952)):

$$\frac{\Delta p}{L} = 150\frac{(1-\varepsilon)^2}{\varepsilon^3}\frac{\mu}{d_p^2}u + 1.75\frac{1-\varepsilon}{\varepsilon^3}\frac{\rho}{d_p}u^2 \tag{9.41}$$

Based on Equation (9.41), the linear viscous drag coefficient and quadratic drag coefficient of the Ergun correlation are

$$\beta = 150\frac{(1-\varepsilon)^2}{\varepsilon^2}\frac{\mu}{d_p^2}, \tag{9.42}$$

and

$$\alpha = 1.75\frac{1-\varepsilon}{\varepsilon^2}\frac{\rho}{d_p}, \tag{9.43}$$

respectively. The non-dimensional frictional pressure drop coefficient is therefore

$$\Psi = \frac{300}{\mathrm{Re}_m} + 3.5. \tag{9.44}$$

Unfortunately, the validity region and uncertainty range of the Ergun correlation were not provided.

The **KTA correlation** has been developed based on the review of published data and correlations from about 30 papers (Fenech (2013)). The KTA correlation is given as (Fenech (2013); Nuclear Safety Standards Commission (1981)):

$$\frac{\Delta p}{L} = \Psi\frac{1-\varepsilon}{\varepsilon^3}\frac{1}{d_p}\frac{\rho u^2}{2} \tag{9.45}$$

with the non-dimensional drag coefficient,

$$\Psi = \frac{320}{\mathrm{Re}_m} + \frac{6}{\mathrm{Re}_m^{0.1}}. \tag{9.46}$$

Although the expression does not explicitly include the wall effect, the discussion provided in Fenech (Fenech (2013)) suggests that the wall effect has been implicitly considered in the average void fraction of the pebble bed, which is dependent on the pebble diameter-to-bed diameter ratio.

The linear viscous drag coefficient and quadratic drag coefficient are

$$\beta = 160\frac{(1-\varepsilon)^2}{\varepsilon^2}\frac{\mu}{d_p^2}, \tag{9.47}$$

and

$$\alpha = \frac{3}{\mathrm{Re}_m^{0.1}}\frac{1-\varepsilon}{\varepsilon^2}\frac{\rho}{d_p}, \tag{9.48}$$

respectively.

The reported validity region of the KTA correlation is:

- Reynolds number: $1 < \mathrm{Re}_m < 10^5$
- Porosity of the bed: $0.36 < \varepsilon < 0.42$
- Diameter ratio: curve fitting, see Nuclear Safety Standards Commission (1981)
- Height of bed: $H > 5d_p$

and the uncertainty range is:

- $\pm 15\%$ with a confidence level of $\pm 95\%$.

Eisfeld and Schnitzlein (Eisfeld and Schnitzlein (2001)) performed a detailed analysis of more than 2,300 experimental data points, and found that the influence of the container walls on the pressure drop of packed beds should be considered. Their investigation showed that Reichelt's approach to correcting the original Ergun equation for the wall effect is most promising, and they provided improved correlations obtained by fitting the coefficients to their database.

Eisfeld and Schnitzlein provided three fitted correlations, one for spherical particles, one for cylinder particles, and one for both spherical and cylinder particles. As our main focus is to provide modeling and simulation capability for the (spherical) pebble-bed reactor designs, we include only the correlation for spherical particles. The Eisfeld and Schnitzlein correlation for spherical particles is given as

$$\frac{\Delta p}{L} = \left[\frac{154A_w^2}{\mathrm{Re}_m} + \frac{A_w}{B_w}\right]\frac{1-\varepsilon}{\varepsilon^3}\frac{\rho u^2}{d_p} \tag{9.49}$$

with the non-dimensional drag coefficient,

$$A_w = 1 + \frac{2\,(d_p/D_{bed})}{3(1-\varepsilon)}\,, \tag{9.50}$$

and

$$B_w = \left[1.15\left(\frac{d_p}{D_{bed}}\right)^2 + 0.87\right]^2. \tag{9.51}$$

Thus, the linear viscous drag coefficient and quadratic drag coefficient are

$$\beta = 154A_w^2\frac{(1-\varepsilon)^2}{\varepsilon^2}\frac{\mu}{d_p^2}\,, \tag{9.52}$$

and

$$\alpha = \frac{A_w}{B_w}\frac{1-\varepsilon}{\varepsilon^2}\frac{\rho}{d_p}\,, \tag{9.53}$$

respectively. The non-dimensional frictional pressure drop coefficient is therefore

$$\Psi = \frac{308A_w^2}{\mathrm{Re}_m} + \frac{2A_w}{B_w}\,. \tag{9.54}$$

The reported validity region of the Eisfeld and Schnitzlein correlation is:

- Reynolds number: $0.01 < \mathrm{Re}_{dp} < 17635$ in which $\mathrm{Re}_{dp} = \rho u d_p/\mu$
- Porosity of the bed: $0.330 < \varepsilon < 0.882$
- Diameter ratio: $1.624 \leq D_{bed}/d_p \leq 250$
- Height of bed: not provided

and the uncertainty range is:

- Relative root mean square deviation, $\sigma = 0.1613$
- Maximum relative deviation at 95% confidence, $f_{95} = 0.31$
- Maximum relative deviation at 99% confidence, $f_{99} = 0.48$

9.4.2 Pebble-to-fluid Heat Transfer

In pebble beds, in a unit volume, the heat transferred from pebbles to the surrounding fluid is computed from the temperature difference between them, the pebble-to-fluid heat transfer coefficient, as well as the total heat transfer area per unit volume (or heat transfer area density).

For the calculation of the heat transfer coefficient, the KTA model is implemented:

$$\mathrm{Nu} = 1.27\frac{\mathrm{Pr}^{1/3}}{\varepsilon^{1.18}}\mathrm{Re}^{0.36} + 0.033\frac{\mathrm{Pr}^{1/2}}{\varepsilon^{1.07}}\mathrm{Re}^{0.86}\,, \tag{9.55}$$

and the pebble-to-fluid heat transfer coefficient can be obtained as

$$h = \frac{\mathrm{Nu}k}{d_p}\,, \tag{9.56}$$

in which the Reynolds number is defined as

$$\mathrm{Re} = \frac{\rho u d_p}{\mu}\,. \tag{9.57}$$

The heat transfer area density is a pebble bed with pebble diameter d_p can be obtained as

$$a_w = \frac{6(1-\varepsilon)}{d_p}\,, \tag{9.58}$$

which can be found from the following two equations:

$$1-\varepsilon = \frac{n}{6}\pi d_p^3\,, \tag{9.59}$$

and

$$a_w = n\pi d_p^2\,, \tag{9.60}$$

in which n is number of pebbles per unit volume.

The reported validity region of the KTA correlation is:

- Reynolds number: $100 < \mathrm{Re} < 10^5$
- Porosity of the bed: $0.36 < \varepsilon < 0.42$
- Diameter ratio: $D_{bed}/d_p > 20$
- Height of bed: $H > 4d_p$

9.4.3 Wall-to-fluid Heat Transfer in Pebble Beds

In pebble bed HTGR designs, the heat transfer coefficient between the pebble bed and the surrounding reflectors is an important parameter that affects the accurate prediction of core temperatures during transients such as pressurized conduction cooldown (PCC). The wall-to-pebble-bed heat transfer had been the subject of several past studies, from which empirical correlations were proposed, such as the Yagi and Wakao correlation (Yagi and Wakao (1959)) and the Achenbach correlation (Achenbach (1995)). When deriving these correlations, for example, Yagi and Wakao (Yagi and Wakao (1959)) assumed that the fluid and particle phase were at thermal equilibrium. The wall-to-pebble-bed heat transfer is due to the temperature difference between the bed and the wall on the wall-to-pebble-bed interface, and the heat transfer coefficient is lumped into a single parameter analogy to the classic convective heat transfer coefficient between a wall and a flow field. It is worth noting that this assumption is inconsistent with the porous medium flow model of SAM, which does not assume a thermal equilibrium between the two phases, i.e., see equation (9.16) and (9.17).

To address such an inconsistence, additional assumption must be made. We assumes that the temperature of solid phases, including both pebbles and walls that bound the pebble bed, is continuous across the pebble-bed-to-wall interface; and thus the wall-to-pebble-bed heat transfer is applied to the fluid phase only. With this assumption, wall-to-pebble-bed heat transfer coefficient can be seen as the wall-to-fluid heat transfer coefficient.

Yagi and Wakao (Yagi and Wakao (1959)) correlation for the wall-to-pebble-bed heat transfer coefficient, h_w, is given as:

$$\frac{h_w}{c_p \rho u} \mathrm{Pr}^{2/3} = 0.2 \mathrm{Re}^{-0.2} \tag{9.61}$$

in which $\mathrm{Re} = \rho u d_p / \mu$.

Using the conventional definition for the Nusselt number, i.e.,

$$\mathrm{Nu}_w = \frac{h_w d_p}{k} \tag{9.62}$$

in which h_w is the wall-to-pebble-bed heat transfer coefficient and k is the thermal conductivity of the fluid phase. Yagi and Wakao correlation could be rewritten in terms of wall Nusselt number as:

$$\mathrm{Nu}_w = 0.2 \mathrm{Re}^{0.8} \mathrm{Pr}^{1/3} \tag{9.63}$$

Yagi and Wakao suggested that the correlation is valid for $\mathrm{Re} > 20$.

The **Achenbach** correlation (Achenbach (1995)) is quite similar to Yagi and Wakao correlation, but it further considers the pebble-bed-diameter (D_{bed}) effect. The Achenbach correlation is given as:

$$\mathrm{Nu}_w = \left(1 - \frac{d_p}{D_{bed}}\right) \mathrm{Re}^{0.61} \mathrm{Pr}^{1/3} \tag{9.64}$$

with a recommended valid region of $50 < \mathrm{Re} < 2 \times 10^4$.

9.4.4 Effective Thermal Conductivity of Pebble Beds

For HTGR designs, the decay heat removal mechanism is quite different than that of most existing light-water reactor designs. Instead active safety systems, HTGR designs are mainly relying on their passive decay heat removal mechanism that utilizes its large vessel surface area as a way of transferring decay heat. For example, during a postulated pressurized conduction cooldown (PCC) event, the majority of decay heat is transferred from the core region radially to reactor core vessel surface via heat conduction, and then is transferred to the Reactor Cavity Cooling System (RCCS) via thermal radiation and convection.

To support system level safety analysis of HTGR applications, correlations for effective thermal conductivity of pebble bed have been implemented in SAM. Although there exist a large number of such correlations and their variances of different formulations van Antwerpen et al. (2010), most of them share common features to account for several major heat transfer mechanisms, such as heat conduction through pebbles and pebble contact areas, heat conduction/convection through the fluid, and thermal radiation from pebble to pebble. Currently, two correlations have been selected and implemented in SAM, i.e., the Zehner, Bauer, and Schlünder (ZBS) van Antwerpen et al. (2010); You et al. (2017)[2] and IAEA IAEA (2000) correlations.

The **Zehner, Bauer, and Schlünder (ZBS)** model is given as,

$$\frac{k_{eff}}{k_f} = \left(1 - \sqrt{1-\varepsilon}\right) \varepsilon \left[\frac{1}{\varepsilon - 1 + \kappa_G^{-1}} + \kappa_r\right] + \sqrt{1-\varepsilon}\left[\varphi \kappa_s + (1-\varphi)\kappa_c\right] \tag{9.65}$$

---

[2] The original reference for the ZBS model is not available, and the ZBS model has been taken from these two references. A cross check has been done to make sure that these two sources agree with each other.

Of the ZBS model, the non-dimensional effective thermal conductivity related to fluid phase conduction, κ_G, is simplified to be 1. The exact definition of this term is referred to (van Antwerpen et al., 2010) and (You et al., 2017). The non-dimensional effective thermal conductivity related to thermal radiation is given as,

$$\kappa_r = \frac{4\sigma}{2/\varepsilon_r - 1} T^3 \frac{d_p}{k_f}$$

with ε_r being the surface emissivity, σ the Stephan-Boltzmann constant, and d_p the pebble diameter. The non-dimensional effective thermal conductivity related to solid heat conduction is simple $\kappa \equiv k_s/k_f$. The last contribution represents the heat transfer due to solid conduction, fluid conduction, and thermal radiation,

$$\kappa_c = \frac{2}{N}\left\{\frac{B(\kappa+\kappa_r-1)}{N^2\kappa_G\kappa}\ln\frac{\kappa+\kappa_r}{B[\kappa_G+(1-\kappa_G)(\kappa+\kappa_r)]} + \frac{B+1}{2B}\left[\frac{\kappa_r}{\kappa_G} - B\left(1+\frac{1-\kappa_G}{\kappa_G}\kappa_r\right)\right] - \frac{B-1}{N\kappa_G}\right\}$$

with

$$B = 1.25\left(\frac{1-\varepsilon}{\varepsilon}\right)^{10/9}$$

and

$$N = \frac{1}{\kappa_G}\left(1+\frac{\kappa_r - B\kappa_G}{\kappa}\right) - B\left(\frac{1}{\kappa_G}-1\right)\left(1+\frac{\kappa_r}{\kappa}\right)$$

For the **IAEA model**, it argues that there are several types of heat transfer mechanisms in pebble beds, including radiation in void region, conduction of gas (fluid phase), conduction of spherical and convection of gas (fluid phase). The IAEA effective thermal conductivity model includes all except for the gas phase convection part, and therefore, it includes three parts: 1) solid conduction-void radiation-solid conduction process, 2) solid conduction—gas conduction—solid conduction process, and 3) solid conduction—contact area conduction—solid conduction process. The IAEA model is thus given as,

$$k_{eff} = k^r_{eff} + k^g_{eff} + k^c_{eff} \tag{9.66}$$

The solid conduction-void radiation-solid conduction contribution to the total effective thermal conductivity is given as,

$$k^r_{eff} = \left\{\left[1-(1-\varepsilon)^{1/2}\right]\varepsilon + \frac{(1-\varepsilon)^{1/2}}{2/\varepsilon_r - 1}\cdot\frac{B+1}{B}\cdot\frac{1}{1+\frac{1}{(2/\varepsilon_r-1)\Lambda}}\right\}4\sigma T^3 d_p \tag{9.67}$$

with

$$\Lambda = \frac{k_s}{4\sigma T^3 d_p}$$

The solid conduction—gas conduction—solid conduction contribution is given as,

$$k^g_{eff} = k_f\left\{1-\sqrt{1-\varepsilon} + \frac{2\sqrt{1-\varepsilon}}{1-B/\kappa}\left[\frac{(1-1/\kappa)B}{(1-B/\kappa)^2}\ln\frac{\kappa}{B} - \frac{B+1}{2} - \frac{B-1}{1-B/\kappa}\right]\right\} \tag{9.68}$$

The solid conduction—contact area conduction—solid conduction contribution is given as,

$$k^c_{eff} = k_s\left[\frac{3(1-\mu_p)^2}{4E_s}fR\right]^{1/3}\frac{1}{0.531S}\left(\frac{N_A}{N_L}\right) \tag{9.69}$$

with

$$f = p\frac{S_F}{N_A}$$

where N_A and N_L are the number of pebbles per unit area and length, respectively. $R = d_p/2$ is the pebble radius. For the simple cubic arrangement of pebbles,

$$
\begin{aligned}
&S = 1\\
&S_F = 1\\
&N_A = 1/(4R^2)\\
&N_L = 1/(2R)
\end{aligned}
$$

For graphite, the Poisson ratio $\mu_p = 0.136$ and the Young modules $E_s = 9 \times 10^9$ N/m^2 are provided as default values in current implementation. The external pressure p is related to the weight of pebbles in the pebble bed. Although an exact formula was not given in the IAEA model (IAEA, 2000), based on the original discussion provided in (Chan and Tien, 1973), one could reason that such a pressure is pebble bed depth-dependent, and the pebble bed pressure can be computed as,

$$p = \rho_s g(1-\varepsilon)\mathrm{d}z$$

where $\mathrm{d}z = z - z_0$ is the local bed depth with respect to the bed top surface at z_0.

9.5 Species Transport

An additional capability needed for the multi-dimensional flow model component is the ability to model mass transport of a species through the system. Especially for liquid-fueled Molten Salt Reactors (MSRs), being able to model the transport of delayed-neutron precursors (DNPs) and decay heat precursors is important with respect to neutronic and thermal hydraulic behavior. The 3D passive scalar equation to model mass transport is shown below:

$$\rho\frac{\partial c}{\partial t} + \rho\left(u\frac{\partial c}{\partial x} + v\frac{\partial c}{\partial y} + w\frac{\partial c}{\partial z}\right) - D\left(\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} + \frac{\partial^2 c}{\partial z^2}\right) + \lambda\rho c - S = 0 \tag{9.70}$$

Where ρ is the fluid density, c is the species concentration density, D is the diffusion coefficient, λ is the decay constant of the species, S is the production/removal rate of the species, and u, v, and w are the fluid velocities in the x, y, and z directions, respectively.

The weak forms of the discretized equations are obtained by multiplying the governing equations by a test function, and then integrating over the domain. The weak form of the equation above is represented as shown below:

$$\left(\rho\frac{\partial c}{\partial t}, \psi\right) + (\rho\vec{u}\cdot\nabla c, \psi) + (D\nabla c, \nabla\psi) - \langle D\nabla c\cdot\hat{n}, \psi\rangle + (\lambda\rho c, \psi) - (S, \psi) = 0 \tag{9.71}$$

Where, ψ is the test function and $\hat{n}$ is the boundary normal vector. As discussed above, parentheses represent volumetric integrals and brackets represent surface integrals. From this, we can compute the residual terms as shown below.

$$R_i = \left(\rho\frac{\partial c}{\partial t}, \psi_i\right) + (\rho\vec{u}\cdot\nabla c, \psi_i) + (D\nabla c, \nabla\psi_i) - \langle D\nabla c\cdot\hat{n}, \psi_i\rangle + (\lambda\rho c, \psi_i) - (S, \psi_i) \tag{9.72}$$

Note that stabilization schemes are necessary for the standard Continuous Galerkin FEM to regulate spurious spatial numerical oscillations, when it is applied to convection-dominant problems. For the passive scalar transport equation, an additional SUPG term is added, similar to what is discussed above in Section 9.3.1.

10 Control System Models

10.1 Introduction

The SAM control and trip system are aimed to provide the capability to simulate the control and trip systems typically used in hydrodynamic systems. The control system is used to perform the evaluation of algebraic and simple ordinary differential equations (e.g. integraion and differentiation); the trip system is used to perform the evaluation of logical statements.

The fundamental approximation made in the design of control/trip system is that the execution of control/trip system is de-coupled from the other parts of the hydraulic systems. The main execution of individual control/trip units is set at the end of each time step.

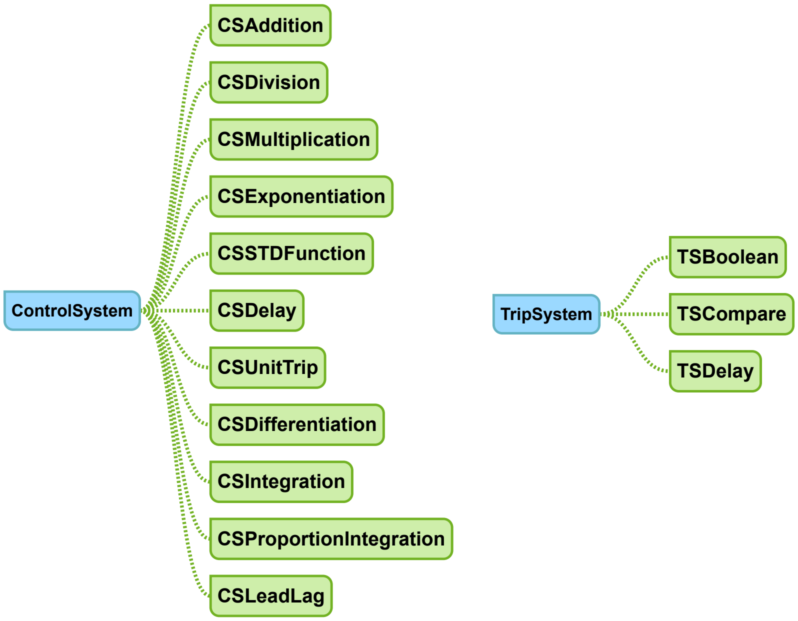


Figure 10.1: SAM control and trip system

The control and trip system are implemented as ControlSystem and TripSystem class, respectively. Inherited from the ControlSystem/TripSystem, a set of basic control/trip components are provided to handle the different operations, e.g. addition, integration, comparison, and etc. A schematic of the design of control and trip system is shown in Figure 10.1.

10.2 Control System

The control system is used to evaluate simultaneous algebraic and ordinary differential equations. The main application of this new capability is to simulate control systems typically used in thermal-hydraulic systems. The control system provides several types of control components. Each component defines a specific function of input variable. The input variable could be the time-advanced quantities in the hydraulic systems, output from control components, and output from trip components. The following list of control components are currently provided:

- CSAddition: addition/subtraction evaluation
- CSDivision: division evaluation
- CSMultiplication: multiplication evaluation
- CSExponentiation: power function evaluation
- CSSTDFunction: standard function evaluation
- CSDelay: delayed output evaluation
- CSUnitTrip: status of trip input
- CSDifferentiation: differentiation evaluation
- CSIntegration: integration evaluation
- CSProportionIntegration: proportional-integration evaluation
- CSLeadLag: lead/lag evalulation

The following subsections provides the mathematical description of the different control components. For different control components, the following common notations are used:

- Y: output variable of control component
- V, V_i: input (i-th input) variable to the control component
- S: constant scaling factor

10.2.1 CSAddition

This control component computes the summation of its input variables.

$$Y = S(A_0 + \sum_{i=1}^{N} A_i V_i) \tag{10.1}$$

where A_0 is a constant bias, A_i is the i-th gain coefficient of i-th input variable, N is the total number of input variables.

10.2.2 CSDivision

This control component compute the division of its input variables. If only 1 input is specified, the inverse of the input variable is computed.

$$Y = \frac{S}{V_1} \text{ or } S\frac{V_2}{V_1} \tag{10.2}$$

10.2.3 CSMultiplication

This control component compute the product of its input variables.

$$Y = S\prod_{i=1}^{N} V_i \tag{10.3}$$

where N is the total number of input variables.

10.2.4 CSExponentiation

This control component compute the power of its input variables. If 2 input variables are specified, the second input is taken as exponent to the power function.

$$Y = SV_1^x \text{ or } SV_1^{V_2} \tag{10.4}$$

where x is exponent to the power function.

10.2.5 CSSTDFunction

This control component computes one of the standard function of its input variables.

$$Y = S\mathbf{F}(V_1, V_2, \cdots) \tag{10.5}$$

where $\mathbf{F}$ can be $\sin(V_1)$, $\cos(V_1)$, $\tan(V_1)$, $\sinh(V_1)$, $\cosh(V_1)$, $\tanh(V_1)$, $|V_1|$, $\ln(V_1)$, $\exp(V_1)$, $\sqrt{V_1}$, $\min(V_1, V_2, \cdots)$, and $\max(V_1, V_2, \cdots)$. Only min and max function may take multiple input variables.

10.2.6 CSDelay

This control component return the input variable at a delayed time.

$$Y = SV_1(t - t_d) \tag{10.6}$$

where t is the current time and t_d is the delay time. History of the input variable is stored in form of (time, value) pair. A user input is used to determine the maximum number of pairs to store. The deleyed function is obtained by a liner interpolation of the stored history value.

10.2.7 CSUnitTrip

This control component return a value of 0.0 or 1.0 depending on the input trip status.

$$Y = SU(\pm t_r) \tag{10.7}$$

where t_r is the input trip. The minus sign represent using complement of the input trip. U is 0.0 or 1.0 depending on the input trip t_r being false or true.

10.2.8 CSDifferentiation

This control component compute derivative of the input variable.

$$Y = S\frac{\mathrm{d}V_1}{\mathrm{d}t} \tag{10.8}$$

The finite difference method is used to calculate the derivative by

$$Y = S\frac{V_1^{n+1} - V_1^n}{\Delta t} \tag{10.9}$$

where Δt, V_1^{n+1}, and V_1^n are the time step, input variable value at the current step, and input variable value at previous step, respectively. Differentiation operation is a noisy process and should be avoided if possible.

10.2.9 CSIntegration

This control component compute integration of the input variable.

$$Y = S\int_{t_0}^{t} V_1 \mathrm{d}t \tag{10.10}$$

where t_0 is the simulation time when the component is added to the system. If initial value of this control component is not provided, a default initial value of 0.0 is used.

The integration is advanced by the trapezoidal approximation

$$Y^{n+1} = Y^n + S(V_1^n + V_1^{n+1})\frac{\Delta t}{2} \tag{10.11}$$

where Δt is the current time step size, $n+1$ denotes the current step, and n denotes the previous step.

10.2.10 CSProportionIntegration

This control component compute proportional-integration of the input variable.

$$Y = S\left(A_1V_1 + A_2\int_{t_0}^{t} V_1 \mathrm{d}t\right) \tag{10.12}$$

where t_0 is the simulation time when the component is added to the system, A_1 is the proportional gain coefficient, and A_2 is the integration gain coefficient. The Laplace transform notation of this component is

$$Y(s) = S\left(A_1 + \frac{A_2}{s}\right)V_1(s) \tag{10.13}$$

This component is advanced in time by

$$I^{n+1} = I^n + (V_1^n + V_1^{n+1})\frac{\Delta t}{2} \tag{10.14}$$

$$Y^{n+1} = S(A_1V_1^{n+1} + A_2I^{n+1}) \tag{10.15}$$

where I is an intermediate variable for storing the integration.

If an initial value is provided Team (2014), then Y^0 is the provided value and

$$I^0 = \frac{1}{A_2}\left(\frac{Y^0}{S} - A_1V_1^0\right) \tag{10.16}$$

otherwise,

$$I^0 = 0 \text{ and } Y^0 = SA_1V_1^0 \tag{10.17}$$

where V_1^0 is initial value of the input variable.

10.2.11 CSLeadLag

This control component computes the lead-lad of the input variable. The lead-lag component is defined in Laplace transform notation as

$$Y(s) = S\left(\frac{1+A_1s}{1+A_2s}\right)V_1(s) \tag{10.18}$$

where A_1 and A_2 are the lead and lag time constant, respectively.

This component is advance in time by

$$Y^{n+1} = \frac{\frac{A_1}{A_2}SV_1^{n+1} + I^n + [S(V_1^n + V_1^{n+1}) - Y^n]\frac{\Delta t}{2A_2}}{1 + \frac{\Delta t}{2A_2}} \tag{10.19}$$

and the intemediate variable I is advanced by

$$I^{n+1} = I^n + [S(V_1^n + V_1^{n+1}) - Y^n - Y^{n+1}]\frac{\Delta t}{2A_2} \tag{10.20}$$

If an initial value is provided Team (2014), then Y^0 is the provided value and

$$I^0 = 0 \tag{10.21}$$

otherwise,

$$Y^0 = SV_1^0 \text{ and } I^0 = \left(1 - \frac{A_1}{A_2}\right)SV_1^0 \tag{10.22}$$

where V_1^0 is initial value of the input variable.

If the lag time constant is specified as zero, which represents a lead operation, the previous evaluation is invalid $A_2 = 0$. In this case, the Laplace transform becomes

$$Y(s) = S(1 + A_1 s)V_1(s) \tag{10.23}$$

or in time domain notation as

$$Y = S\left(V_1 + A_1\frac{\mathrm{d}V_1}{\mathrm{d}t}\right) \tag{10.24}$$

which is advanced in time by

$$Y = S\left(V_1 + A_1\frac{V_1^{n+1} - V_1^n}{\Delta t}\right) \tag{10.25}$$

10.3 Trip System

The trip system is used for the evaluation of logical statements. Each trip object consists of a logical variable that has a true or false result and an associated variable, TIMEOF. The TIMEOF member variable is -1.0 whenever the trip is false, and is set to the time when the trip was last set true. A trip component can be latched or unlatched. A latched trip, once set true, is no longer tested and remains true for the remainder of the problem. An unlatched trip is evaluated and advanced every time step Team (2014).

Two kind of trip componet are currently provided:

- TSCompare: is the type of variable trip and performs the comparison of its input variables
- TSBoolean: is the type of logical trip and performs the boolean of its input signals
- TSDelay: follows the signal of its input trip component but with a time delay

The following subsections provides the mathematical description of the different trip components. For different trip components, the following common notations are used:

- T_r: signal (true or false) of the trip component
- **OP**: operation type

10.3.1 TSCompare

This trip component performs the comparision of its input variables.

$$T_r = V_1 \mathbf{OP} (V_2 + C) \tag{10.26}$$

where C is a user-specified constant bias. The operation OP is one of the following relational operations: EQ is equal, NE is not equal, GT is greater than, GE is greater then or equal, LT is less than, and LE is less than or equal.

10.3.2 TSBoolean

This trip component performs the boolean operation of the input trip signals.

$$T_r = T_{r,1} \mathbf{OP} T_{r,2} \tag{10.27}$$

The operation OP is one of the logical operations: AND, OR (inclusive or), NAND (not-and), NOR (not-ro), XOR (exclusive or), and XNOR (exclusive NOR). An extra operation (NOT) is also available if only 1 input is specified for taking the complement of the input trip.

10.3.3 TSDelay

This trip component returns the signal of its input trip with a delayed time

$$T_r = T_{r,1}(t - t_d) \tag{10.28}$$

where $T_{r,1}$ is the input trip signal, t is the current time, and t_d is the delay time. This compoponent makes use of TIMEOF value of input trip. The output signal is set to true if the input signal is true and $t - t_d$ is larger than TIMEOF value of the input trip.

11 Multi-Scale Multi-Physics Simulations

In additional to the physics and component model development, a flexible coupling interface has been developed so that multi-scale multi-physics modeling capabilities can be achieved by coupling SAM with other high fidelity or conventional simulation tools.

11.1 SAM coupling with CFD Codes

For practical nuclear engineering applications, multi-scale thermal fluid analysis by adopting the combined use of different scale computational tools, such as system thermal-hydraulics and CFD codes, is vital when three-dimensional effects play an important role in the evolution of a given transient or accident scenario. Careful control of data exchange flow and the time-synchronization is essential for a numerically stable and physically valid coupled code simulation. The general issues of coupling of system code and CFD code have been addressed in Hu et al. (2013), including data exchange method, driving mechanism, time synchronization scheme, and the selection of data for exchange. The coupling strategy between SAM and STAR-CCM+ was developed and implemented based on these considerations and the characteristics of each code. STARCCM+ is a general-purpose commercial CFD code that uses a finite volume formulation for the analysis of compressible and incompressible flows and heat transfer, and it has been applied in a wide range of engineering application including the simulations of nuclear reactor systems. The details of the coupling strategy and the implementation can be found in Hu et al. (2013). Similar strategy for the coupling of SAM and STAR-CCM+ is also applied to other multi-scale or multi-physics code coupling applications.

The multiscale coupling capability has been demonstrated in the coupled SAM and STARCCM+ code simulation of the Advanced Burner Test Reactor (Chang et al., 2006) PLOF transient. The importance of the multi-resolution capability was demonstrated by the multi-dimensional flow and the formation of thermal stratification layers in the outlet plenum and cold pool of the ABTR during the postulated transient. The details of the demonstration simulation can be found in Hu et al. (2014a).

11.2 SAM coupling with SAS4A/SASSYS-1

SAS4A/SASSYS-1 is a simulation tool used to perform deterministic analysis of anticipated events as well as design basis and beyond design basis accidents for advanced liquid-metal-cooled nuclear reactors (Fanning, 2012). SAS4A/SASSYS-1 contains a primary and intermediate system modeling component, PRIMAR-4. PRIMAR-4 can represent complex arrangements of coolant system components including pumps, piping, valves, intermediate heat exchangers, air dump heat exchangers, steam generators, etc. In addition to its capabilities, PRIMAR-4 has some shortcomings. The most significant shortcomings are in the form of data management, code structure, and user input limitations. Outdated data management and code structure makes extension of the PRIMAR-4 module difficult. The user input format for PRIMAR-4 limits the number of volumes and segments that can be used to describe a given system. Coupling with SAM will provide an alternative to PRIMAR-4 for primary, secondary, and decay heat coolant system modeling capabilities that are more flexible and extensible.

To combine the advantages of SAS4A/SASSYS-1 and SAM, a coupling strategy has been defined that retains the full complement of core (in the reactor sense) modeling capabilities of SAS4A/SASSYS-1 - coolant channel and sub-channel thermal hydraulics, sodium boiling, fuel restructuring and relocation, in-pin fuel melting, cladding failure, and fuel and clad melting and relocation - and adds the option to use SAM for the primary, intermediate, and decay heat coolant systems. In this approach, the modeling capabilities of PRIMAR-4 will be retained to maintain continuity of simulation capabilities. The coupling effort was successful and is demonstrated by evaluating an unprotected loss of flow transient for the ABTR design. The coupling strategy and the demonstration of the coupled code simulation are detailed in (Fanning and Hu, 2016).

A separate effort was pursued to enable the SAM- SAS4A/SASSYS-1 coupling capability at the solid-liquid interface for potential use of the Versatile Test Reactor (VTR) program. A coupling boundary has been identified at the VTR test vehicle and primary coolant interface, where SAS4A/SASSYS-1 treats primary coolant thermodynamics, while SAM treats all thermodynamic and reactivity behavior within the test vehicle, including the vehicle walls (Brunett et al., 2019). Essential to this integrated tool is its newly developed capability to properly model the conjugate heat transfer process which ensures equality of temperatures and heat fluxes at the vehicle wall interface while ensuring energy conservation.

11.3 Coupling with MOOSE-Based or MOOSE-Wrapped Codes

Because SAM is developed based on the MOOSE framework, the coupling with other MOOSE-based codes and MOOSE-wrapped codes is straightforward by utilizing the MOOSE MultiApp system (Gaston et al., 2015).

As an example, a multi-physics modeling and simulation effort of a heat pipe micro reactor was pursued to utilize several MOOSE-based submodules under NRC's Comprehensive Reactor Analysis Bundle (CRAB), including SAM, MAMMOTH/Rattlesnake, and MOOSE's Tensor Mechanics module. The numerical model consists of the following sub-models coupled to each other through the MOOSE MultiApp system. A schematic representation of interrelation is given in Figure 11.1. Details of this work can be found in (Hu et al., 2019a).

- RK: one whole-core, 3-D MAMMOTH input with homogenized blocks to solve the linearized Boltzmann transport equation.
- HC: one whole-core, 3-D SAM input for the heat conduction calculation in the reactor core.
- HPs: 192 instantiations of a SAM input for modeling the individual heat pipe. Each individual heat pipe is coupled to a cooling pipe to model the secondary heat exchangers.
- RCCS: a SAM input for modeling the reactor cavity cooling system.
- TM-Fuel: 192 instantiations of a Tensor Mechanics input for modeling the axial expansion of the fuel elements.
- TM-Plate: a Tensor Mechanics input for modeling the radial expansion of the reactor core.
- Joint: a dummy input model to initiate the simulation flow of the HC and TM.

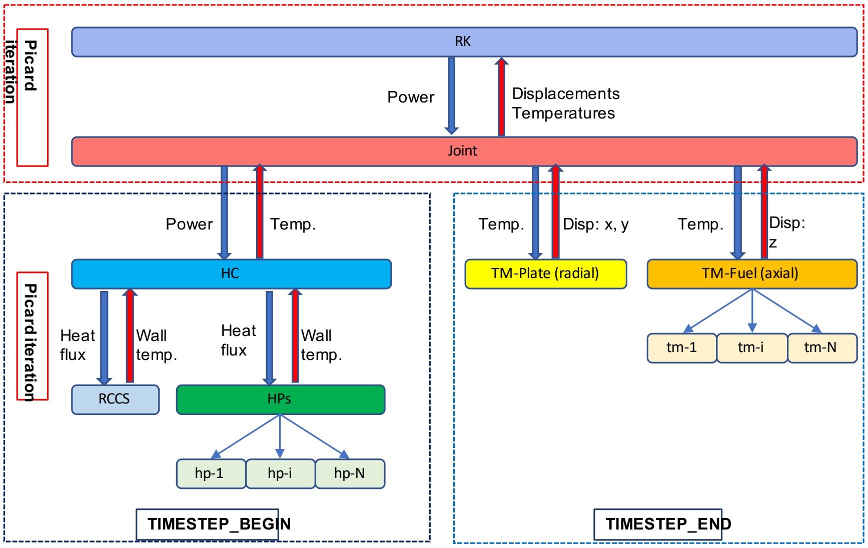


Figure 11.1: Schematic of the multi-physics coupling method for heat pipe micro reactors

ACKNOWLEDGEMENTS

The SAM code development is funded by U.S. Department of Energy Office Nuclear Energy Advanced Modeling and Simulation (NEAMS) program, with additional supports from multiple sources including Argonne Laboratory Directed Research and Development (LDRD), DOE-NE Industry Funding Opportunity Announcement (iFOA), Gateway for Accelerated Innovation in Nuclear (GAIN), Technology Commercialization Fund (TCF), and the U.S. Nuclear Regulatory Commission.

REFERENCES

E. Achenbach. Heat and flow characteristics of packed beds. Experimental thermal and fluid science, 10(1): 17–27, 1995.

G. Alfonsi. Reynolds-averaged navier-stokes equations for turbulence modeling. Applied Mechanics Reviews, 62(4), 2009.

S. Aoki. Current liquid-metal heat transfer research in japan. In O.W. Dwyer, editor, Progress in heat and mass transfer, Vol.7, Heat transfer in liquid metals, pages 569–587. Pergamon Press, 1973.

A. Banerjee, S. Raju, R. Divakar, and E. Mohandas. High Temperature Heat Capacity of Alloy D9 Using Drop Calorimetry Based Enthalpy Increment Measurements. International Journal of Thermophysics, 28: 97–108, 2007.

G. K. Batchelor. An Introduction to Fluid Dynamics. Cambridge University Press, 1967.

R.A. Berry, J.W. Peterson, H. Zhang, R.C. Martineau, H. Zhao, L. Zou, and D. Andrš. RELAP-7 Theory Manual. Technical Report INL/EXT-14-31366, Idaho National Laboratory, 2015.

M.C. Billone. Status of Fuel Element Modeling Codes for Metallic Fuels. In Int. Conf. Reliable Fuels for Liquid Metal Reactors, 1986.

F. C. Blake. The resistance of packing to fluid flow. Transactions of the American Institute of Chemical Engineers, 14(415-421):3, 1922.

A.N. Brooks and T.J.R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. Comp. Methods Appl. Mech. Engrg., 32:199–259, 1982.

Acacia J Brunett, Thanh Q Hua, Guojun Hu, Dan O'Grady, Rui Hu, Thomas H Fanning, and George Zhang. Integrated simulation capabilities for analysis of experiments in the versatile test reactor. Technical report, Argonne National Lab.(ANL), Argonne, IL (United States), 2019.

G.L. Calamai, R.D. Coffield, L. Jossens, J.L. Kerian, J.V. Miller, E.H. Novendstern, G.H. Ursim, H. West, and P.J. Wood. Steady State Thermal and Hydraulic Characteristics of the FFTF Fuel Assemblies. Technical Report ARD-FRT-1582, Westinghouse Electric Corporation, 1974.

C. K. Chan and C. L. Tien. Conductance of packed spheres in vacuum. Journal of Heat Transfer, 95:302–308, 1973.

Y.I. Chang, P.J. Finck, and C. Grandy. Advanced Burner Test Reactor - Preconceptual Design Report. Technical Report ANL-ABR-1 (ANL-AFCI-173), Argonne National Laboratory, 2006.

Hubert Chanson. Hydraulics of open channel flow. Elsevier, 2004.

H.Y. Chen and X. Cheng. The ATHLET-MF Code and Its Application to Heavy Liquid Metal Cooled Systems. Technical Report Wissenschaftliche Berichte FZKA 7165, Forschungszentrum Karlsruhe in der Helmholtz-Gemeinschaft, 2005.

S.K. Cheng and N.E. Todreas. Hydrodynamic models and correlations for bare and wire wrapped hexagonal rod bundles - bundle friction factors, subchannel friction factors and mixing parameters. Nuclear Engineering and Design, 92:227–251, 1986.

X. Cheng and N.I. Tak. Investigation on turbulent heat transfer to lead–bismuth eutectic flows in circular tubes for nuclear applications. Nuclear Engineering and Design, 236:385–393, 2006.

S. Choi and S. Kim. Numerical Validation of Heat Transfer Correlations for Design of the Intermediate Heat Exchanger in a Sodium Cooled Fast Reactor. In Transactions of the Korean Nuclear Society Autumn Meeting, 2008.

S.W. Churchill and H.H.S. Chu. Correlating Equations for Laminar and Turbulent Free Convection From a vertical Plate. International Journal of Heat and Mass Transfer, 18:1323–1329, 1975.

E. L. Compere, S. S. Kirslis, E. G. Bohlmann, F. F. Blankenship, and W.R. Grimes. Fission Product Behavior in the Molten Salt Reactor Experiment. Technical Report ORNL-4865, Oak Ridge National Laboratory, 1975.

C.B. Davis and A.S. Shieh. Overview of the Use of ATHENA for Thermal-Hydraulic Analysis of Systems with Lead-Bismuth Coolant. In Proceedings of the 8th International Conference on Nuclear Engineering 2000, ICONE-8, 2000.

B. Eisfeld and K. Schnitzlein. The influence of confining walls on the pressure drop in packed beds. Chemical engineering science, 56(14):4321–4329, 2001.

S. Ergun. Fluid flow through packed columns. Chem. Eng. Prog., 48:89–94, 1952.

T. H. Fanning. The SAS4A/SASSYS-1 Safety Analysis Code System. Chapter 8 - DEFORM-4: Steady-State and Transient Pre-failure Pin Behavior. Technical Report ANL/NE-16/19, Nuclear Engineering Division, Argonne National Laboratory, 2017.

T.H. Fanning. The SAS4A/SASSYS-1 Safety Analysis Code System. Technical Report ANL/NE-12/4, Argonne National Laboratory, 2012.

T.H. Fanning and R. Hu. Coupling the System Analysis Module with SAS4A/SASSYS-1. Technical Report ANL-ART-74, Argonne National Laboratory, 2016.

TH Fanning, AJ Brunett, and T Sumner. The sas4a/sassys-1 safety analysis code system, version 5. Technical report, Argonne National Lab.(ANL), Argonne, IL (United States), 2017.

H. Fenech. Heat transfer and fluid flow in nuclear systems. Elsevier, 2013.

D. Gaston, C. Newman, G. Hansen, and D. Lebrun-Grandié. MOOSE: A parallel computational framework for coupled systems of nonlinear equations. Nuclear Engineering and Design, 239:1768–1778, 2009.

D. Gaston, C.J. Permann, J.W. Peterson, A.E. Slaughter, D. Andrš, Y. Wang, M.P. Short, D.M. Perez, M.R. Tonks, J. Ortensi, L. Zou, and R.C. Martineau. Physics-based multiscale coupling for full core nuclear reactor simulation. Annals of Nuclear Energy, 84:45–54, 2015.

V.H. Gräber and M. Rieger. Experimentelle Untersuchung des Wärmeübergangs an Flüssigmetalle (NaK) in parallel durchströmten Rohrbündeln bei konstanter und exponentieller Wärmeflussdichteverteilung. Atomkernenergie (ATKE), 19:23–40, 1972.

D. F. Griffiths. The 'No Boundary Condition' Outflow Boundary Condition. International Journal for Numerical Methods in Fluids, 24(4):393–411, 1997.

K.S. Ha, H.Y. Jeong, C. Cho, Y.M. Kwon, Y.B. Lee, and D. Hahn. Simulation of the EBR-II Loss-of-Flow Tests Using the MARS Code. Nuclear Technology, 169:134–142, 2010.

L. Halpern. Artificial Boundary Conditions for the Linear Advection Diffusion Equation. Mathematics of computation, 46(174):425–438, 1986.

Richard B Hetnarski, M Reza Eslami, and GML Gladwell. Thermal stresses: advanced theory and applications, volume 158. Springer, 2009.

G Hu, R Hu, JM Kelly, and J Ortensi. Multi-physics simulations of heat pipe micro reactor. Technical report, Argonne National Lab.(ANL), Argonne, IL (United States), 2019a.

G Hu, G Zhang, and R Hu. Reactivity feedback modeling in sam. Technical report, Argonne National Lab.(ANL), Argonne, IL (United States), 2019b.

R. Hu. An advanced one-dimensional finite element model for incompressible thermally expandable flow. Nuclear Technology, 190:313–322, 2015.

R. Hu. A fully-implicit high-order system thermal-hydraulics model for advanced non-LWR safety analyses. Annals of Nuclear Energy, 101:174–181, 2017.

R. Hu. Three-dimensional flow model development for thermal mixing and stratification modeling in reactor system transients analyses. Nuclear Engineering and Design, 345:209–215, 2019.

R. Hu and Y. Yu. A Computationally Efficient Method for Full-Core Conjugate Heat Transfer Modeling of Sodium Fast Reactors. Nuclear Engineering and Design, 308:182–193, 2016.

R. Hu, J.W. Thomas, and T.H. Fanning. Strategy for Multi-Scale Single-Phase Flow Coupling. Technical Report ANL/NE-13/4, Argonne National Laboratory, 2013.

R. Hu, J.W. Thomas, E. Munkhzul, and T.H. Fanning. Coupled System and CFD Code Simulation of Thermal Stratification in SFR Protected Loss-Of-Flow Transients. In Proceedings of ICAPP 2014, 2014a.

R Hu, JW Thomas, E Munkhzul, and TH Fanning. Coupled system and cfd code simulation of thermal stratification in sfr protected loss-of-flow transients. Proceedings of ICAPP 2014, pages 6–9, 2014b.

G. Iaccarino. Simulation of Turbulent Flows. `https://web.stanford.edu/class/me469b/handouts/turbulence.pdf`, 2004. Accessed: 2020-06-09.

IAEA. Heat Transport and Afterheat Removal for Gas Cooled Reactors Under Accident Conditions. Technical Report IAEA-TECDOC-1163, International Atomic Energy Agency, 2000.

A.Y. Inayatov. Correlation of Data on Heat Transfer Flow Parallel to Tube Bundles at Relative Tube Pitches of 1:1 ¡ s=d ¡ 1:6. Heat Transfer-Soviet Research, 7:84–88, 1975.

F.P. Incropera, D.P. DeWitt, T.L. Bergman, and A.S. Lavine. Fundamentals of Heat and Mass Transfer. John Wiley & Sons, Inc., 2007.

INL. RELAP5-3D Code Manual Volume 1: Code Structure, System Models and Solution Methods. Technical Report INEEL-EXT-98-00834, Rev. 4, Idaho National Laboratory, June 2012.

J. Jerden. Molten Salt Thermophysical Properties Database Development: 2019 Update. Technical Report ANL-CFCT-19-6, Argonne National Laboratory, 2019.

D. Kahaner, C. Moler, and S. Nash. Numerical Methods and Software. Prentice-Hall, New Jersey, USA, 1989.

A. Karahan. Modeling of Thermo-mechanical and Irradiation Behavior of Metallic and Oxide Fuels for Sodium Fast Reactors. PhD thesis, Massachusetts Institute of Technology, 2009.

A. Karahan. Development of a Thermal Conductivity Model for the Irradiated U-Pu-Zr Metallic Fuels. Transactions of the American Nuclear Society, 117:563–566, 2017.

M. Kaviany. Principles of Heat Transfer in Porous Media. Springer, second edition, 1995.

M.S. Kazimi and M.D. Carelli. Heat Transfer Correlation for Analysis of CRBRP Assemblies. Technical Report CRBRP-ARD-0034, Westinghouse, 1976.

B.S. Kirk, J.W. Peterson, R.H. Stogner, and F.C. Graham. libMesh: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations. Engineering with Computers, 22:237–254, 2006.

D. A. Knoll and D. E. Keyes. Jacobian-free Newton-Krylov Methods: a Survey of Approaches and Applications. Journal of Computational Physics, 193:357—397, 2004.

L. Leibowitz and R. A. Blomquist. Thermal conductivity and thermal expansion of stainless steels D9 and HT9. International Journal of Thermophysics, 9(5):873–883, 1988.

F. Lodi, G. Grasso, D. Mattioli, and M. Sumini. ANTEO+: A subchannel code for thermal-hydraulic analysis of liquid metal cooled systems. Nuclear Engineering and Design, 301:128–152, 2016.

K. Maeda and T. Asaga. Change of fuel-to-cladding gap width with the burn-up in FBR MOX fuel irradiated to high burn-up. Journal of Nuclear Materials, 327:1–10, 2004.

Richard Martineau, David Andrs, Joshua Hansel, Cody Permann, Derek Gaston, Steve Novascone, Matthew Bernard, Russel Johns, Rui Hu, Elia Merzari, et al. Extending the capability of nuclear plant systems analysis with advanced tightly-coupled nuclear fuels performance. In AISTech 2018 Iron and Steel Technology Conference and Exposition, pages 1337–1339. Association for Iron and Steel Technology, AISTECH, 2018.

W.H. McAdams. Heat Transmission. New York: McGraw-Hill, third edition edition, 1954.

K. Mikityuk. Heat transfer to liquid metal: review of data and correlations for tube bundles. Nuclear Engineering and Design, 239:680–687, 2009.

HK Moffatt. Prandtl's essentials of fluid mechanics, edited by h erbert o ertel, with contributions by m. b öhle, d. e tling, u. m üller, kr s reenivasan, u. r iedel & j. w arnatz. translated by k atherine m ayes. springer, 2004. 723 pp. isbn 0 387 40437 6.£ 61.50. Journal of Fluid Mechanics, 524:376–379, 2005.

V. Moureau, C. Berat, and H. Pitsch. An Efficient Semi-Implicit Compressible Solver for Large-Eddy Simulations. Technical Report Center for Turbulence Research, Annual Research Briefs 2005, Stanford University, 2005.

P. B. Mucha. On navier–stokes equations with slip boundary conditions in an infinite pipe. Acta Applicandae Mathematica, 76(1):1–15, 2003.

D. A. Nield and A. Bejan. Convection in porous media. Springer, fourth edition, 2013.

R.H. Notter and C.A. Sleicher. A Solution to the Turbulent Graetz Problem - III. Fully Developed and Entry Region Heat Transfer Rates. Chem. Eng. Sci., 27:2073–2093, 1972.

U.S. NRC. TRACE V5.0 Theory Manual. Technical report, U.S. Nuclear Regulatory Commission, 2010.

Nuclear Safety Standards Commission. Reactor Core Design of High-Temperature Gas-Cooled Reactors; Part3: Loss of Pressure through Friction in Pebble Bed Cores. Technical Report KTA 3102.3, 1981.

OECD/NEA. Handbook on Lead-bismuth Eutectic Alloy and Lead Properties, Materials Compatibility, Thermalhydraulics and Technologies, Nuclear Science. OECD Publishing, 2015.

Peterson et al. Temperature-Dependent Thermophysical Properties for Fluoride Salts and Simulant Fluids. Technical Report CIET-DESIGN-001-03, UC Berkeley, 2013.

PETSc. Petsc web page., 2020. URL `http://www.mcs.anl.gov/petsc`.

W. Pfrang and D. Struwe. Assessment of Correlations for Heat Transfer to the Coolant for Heavy Liquid Metal Cooled Core Designs. Technical Report FZKA 7352, FZKA, 2007.

M. Polidori. Implementation of Thermo-Physical Properties and Thermal-Hydraulic Characteristics of Lead-Bismuth Eutectic and Lead on CATHARE Code. Technical Report NNFISS–LP1-001, ENEA, 2010.

S. G. Popov, J. J. Carbajo, V. K. Ivanov, and G. L. Yoder. Thermophysical Properties of MOX and UO2 Fuels Including the Effects of Irradiation. Technical Report ORNL/TM-2000/351, Oak Ridge National Laboratory, 2000.

H. Savage. The Heat Content and Specific Heat of Some Metallic Fast-Reactor Fuels Containing Plutonium. Journal of Nuclear Materials, 25:249–259, 1968.

R. A. Seban and T. T. Shimazaki. Heat transfer to a fuid flowing turbulently in a smooth pipe with walls at constant temperature. Transactions of the ASME, 73:803–809, 1951.

J. Sellars, M. Tribus, and J. Klein. Heat Transfer to Laminar Flow in a Round Tube on Flat Conduit. The Graetz Problem Extended. Trans. Am. Soc. Mech. Engrs., 78:441, 1956.

F. Shakib. Finite element analysis of the compressible euler and navier-stokes equations. PhD dissertation, Stanford University, 2010.

Endre Süli and David F Mayers. An introduction to numerical analysis. Cambridge university press, 2003.

The RELAP5-3D Code Development Team. RELAP5-3D Code Manual Volume I: Code Structure, System Models and Solution Methods. Technical Report INL-EXT-98-00834-V1, Idaho National Laboratory, 2014.

T.E. Tezduyar. Stabilized Finite Element Formulations for Incompressible Flow Computations. Advances in Applied Mechanics, 28:1–44, 1992.

The Engineering Toolbox. Air properties., 2017. URL `http://www.engineeringtoolbox.com/air-properties-d_156.html`.

N.E. Todreas and M.S. Kazimi. Nuclear Systems. Volume 1: Thermal Hydraulic Fundamentals. CRC Press, Taylor & Francis Group, second edition edition, 2012.

P. Ushakov, A. Zhukov, and N. Matyukhin. Heat transfer to liquid metals in regular arrays of fuel elements. High Temperature, 15:868–873, 1977.

W. van Antwerpen, C. G. du Toit, and P. G. Rousseau. A review of correlations to model the packing structure and effective thermal conductivity in packed beds of mono-sized spherical particles. Nuclear Engineering and design, 240(7):1803–1818, 2010.

R. Verfürth. Finite element approximation on incompressible navier-stokes equations with slip boundary condition. Numerische Mathematik, 50(6):697–721, 1986.

R. Verfürth. Finite element approximation of incompressible navier-stokes equations with slip boundary condition ii. Numerische Mathematik, 59(1):615–636, 1991.

H. K. Versteeg and W. Malalasekera. An Introduction to Computational Fluid Dynamics: The Finite Volume Method. Springer, second edition, 2013.

S. Yagi and N. Wakao. Heat and mass transfer from wall to fluid in packed beds. AIChE Journal, 5(1):79–85, 1959.

N. Yamanouchi, M. Tamura, H. Hayakawa, A. Hishinuma, and T. Kondo. Accumulation of Engineering Data for Practical Use of Reduced Activation Ferritic Steel: 8% Cr-2%W-0.2%V-0.04%Ta-Fe. Journal of Nuclear Materials, 191-194:822 – 826, 1992.

W.S. Yang and H.G. Joo. LMR core temperature calculation based on implicit formulation of the ENERGY model and a Krylov subspace method. Ann. Nuc. Eng., 26:629–640, 1999.

D. A. Yoder. Comparison of Turbulent Thermal Diffusivity and Scalar Variance Models. In 54th AIAA Aerospace Sciences Meeting, page 1561, 2016.

Ersheng You, Ximing Sun, Fubing Chen, Lei Shi, and Zuoyi Zhang. An improved prediction model for the effective thermal conductivity of compact pebble bed reactors. Nuclear Engineering and Design, 323: 95–102, 2017.

Y. Yu, E. Merzari, A. Obabko, and J. Thomas. A porous medium model for predicting the duct wall temperature of sodium fast reactor fuel assembly. Nucl. Eng. Des., 295:48–58, 2015.

L. Zou, J.W. Peterson, H. Zhao, and et al. Solving implicit multi-mesh flow and conjugate heat transfer problems with RELAP-7. In Proceedings of M&C 2013, 2013.

Nuclear Science and Engineering Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439

www.anl.gov