ANL/NE-17/4

SAM Theory Manual

*Nuclear Engineering Division

ANL/NE-17/4

SAM Theory Manual

prepared by
Rui Hu
Nuclear Engineering Division, Argonne National Laboratory

March 2017

ABSTRACT

The System Analysis Module (SAM) is an advanced and modern system analysis tool being developed at Argonne National Laboratory under the U.S. DOE Office of Nuclear Energy's Nuclear Energy Advanced Modeling and Simulation (NEAMS) program. SAM development aims for advances in physical modeling, numerical methods, and software engineering to enhance its user experience and usability for reactor transient analyses. To facilitate the code development, SAM utilizes an object-oriented application framework (MOOSE), and its underlying meshing and finite-element library (libMesh) and linear and non-linear solvers (PETSc), to leverage modern advanced software environments and numerical methods.

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). These advanced concepts are distinguished from light-water reactors in their use of single-phase, low-pressure, high-temperature, and low Prandtl number (sodium and lead) coolants. As a new code development, the initial effort has been focused on modeling and simulation capabilities of heat transfer and single-phase fluid dynamics responses in Sodium-cooled Fast Reactor (SFR) systems. The system-level simulation capabilities of fluid flow and heat transfer in general engineering systems and typical SFRs have been verified and validated.

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 the code is still under ongoing development, this SAM Theory Manual will be updated periodically to keep it consistent with the state of the development.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 SAM Overview

The System Analysis Module (SAM) (Hu et al. 2015) 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). These advanced concepts are distinguished from light-water reactors in their use of single-phase, low-pressure, high-temperature, and low Prandtl number (sodium and lead) 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. As a new code development, the initial effort focused on developing modeling and simulation capabilities of the heat transfer and single-phase fluid dynamics responses in reactor systems.

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 the code is still under ongoing development, this 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 (Balay et al., 2017), 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.

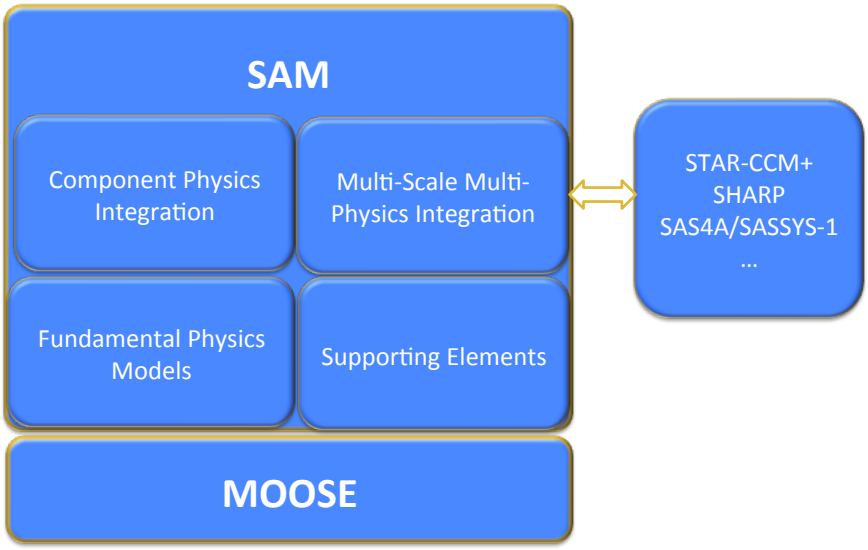


Figure 1-1: SAM Code Structure

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 three-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. Additionally, a subchannel module will be developed for fuel assembly modeling. Both the 3-D module and the sub-channel module will require additional momentum conservation equations. However, they are still under development, thus not discussed further in this document. The details of the single-phase flow model for incompressible thermally expandable flow are discussed in Chapter 2.

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. 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 4.

1.2.4 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 5.

1.3 Overview of Current Capabilities

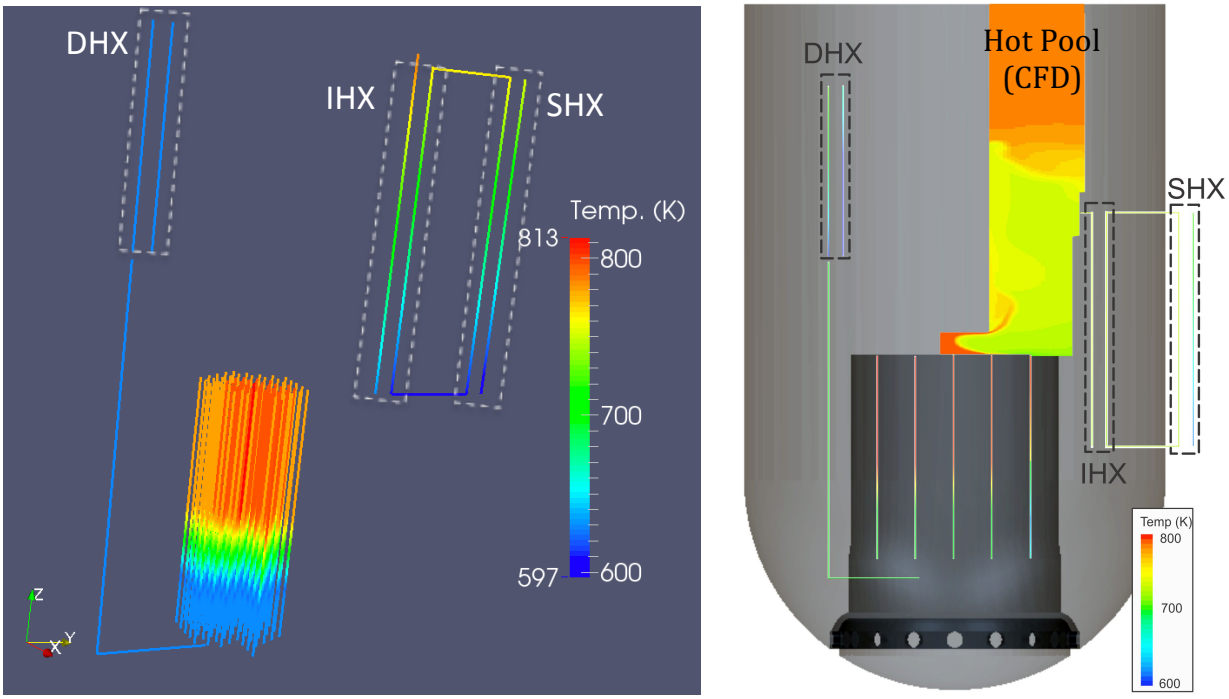
To develop a system analysis code, numerical methods, mesh management, equations of state, fluid properties, solid material properties, neutronics properties, pressure loss and heat transfer closure laws, and good user input/output interfaces are all indispensible. SAM leverages the MOOSE framework and 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 6. 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 infrastructure for coupling with external codes has been developed and demonstrated. Details are discussed in Chapter 7.

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.

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial z} = 0$$

$$\frac{\partial (\rho u)}{\partial t} + \frac{\partial (\rho u u)}{\partial z} = -\frac{\partial p}{\partial z} - \rho g - \nabla \cdot \bar{\bar{\tau}} \qquad (2\text{-}1)$$

$$\frac{\partial (\rho H)}{\partial t} + \frac{\partial (\rho u H)}{\partial z} = \nabla (k \nabla T) - \nabla q_r'' + q''' + \frac{Dp}{Dt} + \phi$$

$$\rho = \rho(p, T)$$

Where,

t: time;

z: the axial coordinate in flow direction;

ρ: the coolant density;

u: velocity;

g: the acceleration due to gravity;

p: pressure;

$\bar{\bar{\tau}}$: 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, $\phi = -v \cdot (\nabla \cdot \bar{\bar{\tau}} + \rho g)$, 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).

$$\frac{\partial \rho}{\partial t}+\frac{\partial(\rho u)}{\partial z}=0$$
$$\frac{\partial(\rho u)}{\partial t}+\frac{\partial(\rho u u+p)}{\partial z}=-\rho g-\frac{f}{D_e}\frac{\rho u|u|}{2} \qquad (2\text{-}2)$$
$$\frac{\partial(\rho H)}{\partial t}+\frac{\partial(\rho u H)}{\partial z}=q'''$$
$$p=p(\rho,\rho u,\rho E)\ or\ \rho=\rho(p,T)$$

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.

$$\frac{\partial \rho}{\partial t}+\frac{\partial(\rho u)}{\partial z}=0$$
$$\rho\frac{\partial u}{\partial t}+\rho u\frac{\partial u}{\partial z}=-\frac{\partial p}{\partial z}-\rho g-\frac{f}{D_e}\frac{\rho u|u|}{2} \qquad (2\text{-}3)$$
$$\rho C_p\frac{\partial T}{\partial t}+\rho C_p u\frac{\partial T}{\partial z}=q'''$$
$$\rho=\rho(p,T)$$

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 ($\rho, \rho u, \rho E$) 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 up-wind 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:

$$\begin{aligned} &\text{Continuity equation: } \nabla \cdot u = 0 \\ &\text{Momentum equation: } \nabla \cdot (\nu \nabla \cdot u + p) = f \end{aligned} \tag{2-7}$$

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:

$$\begin{aligned} &\left(\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial z}, \psi\right) + \left(\rho \frac{\partial u}{\partial t} + \rho u \frac{\partial u}{\partial z} + \frac{\partial p}{\partial z} + \rho g + \frac{f}{D_e} \frac{\rho u |u|}{2}, \tau_{PSPG} \nabla \psi\right) = 0 \\ &\left(\frac{\partial \rho u}{\partial t} + \frac{\partial \rho u u}{\partial z} + \frac{\partial p}{\partial z} + \rho g + \frac{f}{D_e} \frac{\rho u |u|}{2}, \psi\right) \\ &\qquad + \left(\rho \frac{\partial u}{\partial t} + \rho u \frac{\partial u}{\partial z} + \frac{\partial p}{\partial z} + \rho g + \frac{f}{D_e} \frac{\rho u |u|}{2}, \tau_{SUPG} \nabla \psi\right) = 0 \\ &\left(\frac{\partial \rho H}{\partial t} + \frac{\partial \rho u H}{\partial z} - q''', \psi\right) + \left(\rho C_p \frac{\partial T}{\partial t} + \rho C_p u \frac{\partial T}{\partial z} - q''', \tau_{SUPG} \nabla \psi\right) = 0 \end{aligned} \tag{2-9}$$

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:

$$\tau = \left[\left(\frac{2}{\Delta t}\right)^2 + \left(\frac{2|u|}{h}\right)^2 + \left(\frac{4\nu}{h^2}\right)^2\right]^{-1/2}$$
$$\tau_{SUPG} = u \cdot \tau \qquad (2\text{-}10)$$
$$\tau_{PSPG} = \tau(U)$$

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 (1988). 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:

$$\tau_{SUPG} = u \cdot \left[\left(\frac{2}{\Delta t}\right)^2 + \left(\frac{2|u|}{h}\right)^2\right]^{-1/2}$$
$$\tau_{PSPG} = \left[\left(\frac{2}{\Delta t}\right)^2 + \left(\frac{2|U|}{h}\right)^2\right]^{-1/2} \qquad (2\text{-}11)$$

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

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:

$$\rho C_p \frac{\partial T}{\partial t} - \nabla(k\nabla T) - Q''' = 0 \qquad (3\text{-}1)$$

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: $-k\nabla T = h \cdot (T - T_{fluid})$.

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

For 1-D plate type,

$$\nabla(k\nabla T) = \frac{d}{dx}(k\frac{dT}{dx}) \qquad (3\text{-}2)$$

For 2-D plate type,

$$\nabla(k\nabla T) = \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial z}(k\frac{\partial T}{\partial z}) \qquad (3\text{-}3)$$

For 1-D cylindrical type,

$$\nabla(k\nabla T) = \frac{1}{r}\frac{d}{dr}\left(kr\frac{dT}{dr}\right) \qquad (3\text{-}4)$$

For 2-D cylindrical type,

$$\nabla(k\nabla T) = \frac{1}{r}\frac{\partial}{\partial r}\left(kr\frac{\partial T}{\partial r}\right) + \frac{\partial}{\partial z}(k\frac{\partial T}{\partial z}) \qquad (3\text{-}5)$$

For 1-D spherical type,

$$\nabla(k\nabla T) = \frac{1}{r^2}\frac{d}{dr}\left(kr^2\frac{dT}{dr}\right) \qquad (3\text{-}6)$$

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 3-1. 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.

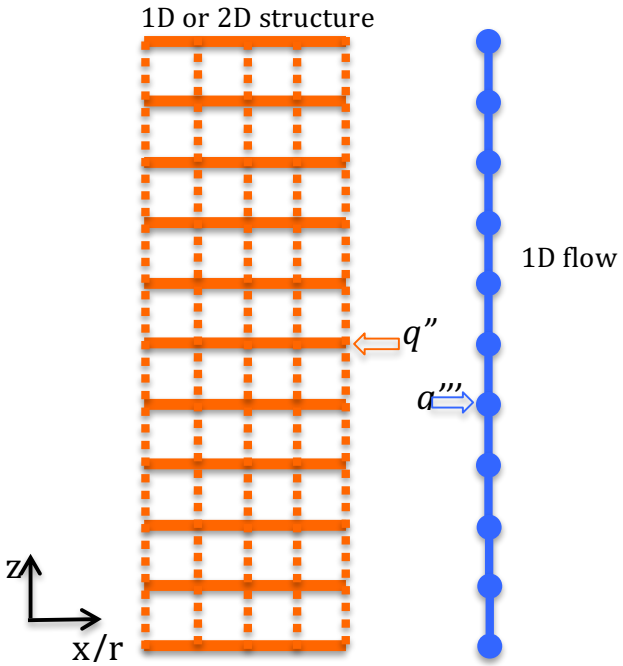


Figure 3-1: The schematic of conjugate heat transfer modeling in SAM

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

$$q" = -k\nabla T = h(T_w - T_f). \quad (3\text{-}7)$$

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,

$$\left(\rho C_p \frac{\partial T}{\partial t}, \psi\right) + (k\nabla T, \nabla\psi) - \langle k\nabla T, \psi\rangle - (Q''', \psi) = 0 \quad (3\text{-}8)$$

In which ψ is the test function; and (f, ψ) 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,

$$q''' = h(T_w - T_f)\frac{P_{heated}}{A_c} \quad (3\text{-}9)$$

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

$$\Delta r_{energy} = \int_{\Omega} q''' \cdot \psi d\Omega = \int_{\Omega} h(T_w - T_f)\frac{P_{heated}}{A_c} \cdot \psi \, d\Omega \quad (3\text{-}10)$$

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

$$\Delta r_{energy} = \int_{\Omega} h(T_w - T_f)\frac{P_{heated}}{A_c} \cdot (\psi + \tau_{supg} \cdot \nabla\psi) \, d\Omega \quad (3\text{-}11)$$

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

$$R_i A_i = \epsilon_i \sigma T_i^4 A_i + \rho_i \sum_{j=1}^{n} R_j F_{ji} A_j \quad (3\text{-}12)$$

in which,

R: radiosity

ϵ: emissivity

σ: Stefan-Boltzmann constant

T: temperature

ρ: $1 - \epsilon$; 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

$$A_i F_{ij} = A_j F_{ji} \quad (3\text{-}13)$$

and the unity rule which enforces the following constraint

$$\sum_{j=1}^{n} F_{ij} = 1 \qquad (3\text{-}14)$$

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

$$\dot{Q}_{i\to j} = A_i R_i F_{ij} - A_j R_j F_{ji} \qquad (3\text{-}15)$$

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

$$\dot{Q}_{12} = \frac{\sigma(T_1^4 - T_2^4)}{\frac{1-\epsilon_1}{A_1\varepsilon_1} + \frac{1}{A_1 F_{12}} + \frac{1-\epsilon_2}{A_2\varepsilon_2}} = -\dot{Q}_{21} \qquad (3\text{-}16)$$

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

$$q_1" = \frac{\dot{Q}_{12}}{A_1} = \frac{\sigma(T_1^4 - T_2^4)}{\frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2} - 1} = -q_2" \qquad (3\text{-}17)$$

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

$$q_1" = \frac{\dot{Q}_{12}}{A_1} = \frac{\sigma(T_1^4 - T_2^4)}{\frac{1}{\varepsilon_1} + \frac{1-\epsilon_2}{\varepsilon_2}\left(\frac{r_1}{r_2}\right)} = -q_2"\frac{r_2}{r_1}$$

$$q_2" = \frac{-\sigma(T_1^4 - T_2^4)}{\frac{1}{\varepsilon_1}\frac{r_2}{r_1} + \frac{1-\epsilon_2}{\varepsilon_2}} \qquad (3\text{-}18)$$

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

$$\Delta r = \int_{\Omega} \psi * q'' \, d\Omega \qquad (3\text{-}19)$$

4 Closure Models

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. Some fluid properties, such as sodium, air, and salts like FLiBe and FLiNaK, have been implemented in SAM. The following functions/correlations are needed in the SAM EOS model for each fluid.

$$\begin{aligned}
&\rho = \rho(p,T), \qquad \frac{\partial \rho}{\partial T} = \xi(p,T), \qquad \frac{\partial \rho}{\partial p} = \eta(p,T) \\
&\beta = \beta(p,T) = -\frac{1}{\rho}\frac{\partial \rho}{\partial T} \\
&\mu = \mu(p,T) \\
&k = k(p,T) \\
&Pr = \frac{C_p}{\mu k} \\
&H = H(p,T) \; or \; H(T), \qquad T = T(H) \\
&C_p = C_p(p,T), \;\; C_v = C_v(p,T), \qquad \frac{\partial C_p}{\partial T} = f(p,T)
\end{aligned} \tag{4-1}$$

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 from a pressure-temperature state relation as:

$$\begin{aligned}
\frac{\partial \rho}{\partial t} &= \xi \frac{\partial T}{\partial t} + \eta \frac{\partial p}{\partial t} \\
\frac{\partial \rho}{\partial z} &= \xi \frac{\partial T}{\partial z} + \eta \frac{\partial p}{\partial z}
\end{aligned} \tag{4-2}$$

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

$$\frac{\partial \rho}{\partial t} = \xi \frac{\partial T}{\partial t}$$
$$\frac{\partial \rho}{\partial z} = \xi \frac{\partial T}{\partial z} \qquad (4\text{-}3)$$

4.1.2 Sodium Property Modeling

The sodium property model in SAS4A/SASSYS-1 (Fanning 2012) in applied in SAM, in which most properties are only dependent on temperature.

Liquid Density (ρ_ℓ in kg/m^3)

$$\rho_\ell = A_{12} + A_{13}T + A_{14}T^2$$
$$\frac{\partial \rho}{\partial T} = A_{13} + 2A_{14}T \qquad (4\text{-}4)$$
$$A_{12} = 1.00423 \times 10^{3}$$
$$A_{13} = -0.21390$$
$$A_{14} = -11046 \times 10^{-5}$$

Liquid Heat Capacity (C_ℓ in J/kg-K)

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

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

$$\beta_l = A_{42} + \frac{A_{43}}{T_c - T} + \frac{A_{44}}{(T_c - T)^2} + \frac{A_{45}}{(T_c - T)^3} + \frac{A_{46}}{(T_c - T)^4} + \frac{A_{47}}{(T_c - T)^5} \quad (4\text{-}6)$$

$A_{42} = 2.5156 \times 10^{-6}$

$A_{43} = 0.79919$

$A_{44} = -6.9716 \times 10^{2}$

$A_{45} = 3.3140 \times 10^{5}$

$A_{46} = -7.0502 \times 10^{7}$

$A_{47} = 5.4920 \times 10^{9}$

Liquid Thermal Conductivity (k_ℓ in W/m-K)

$$k_\ell = A_{48} + A_{49} T + A_{50} T^2 + A_{51} T^3 \quad (4\text{-}7)$$

$A_{48} = 1.1045 \times 10^{2}$

$A_{49} = -6.5112 \times 10^{-2}$

$A_{50} = 1.5430 \times 10^{-5}$

$A_{51} = -2.4617 \times 10^{-9}$

Liquid Viscosity (μ_ℓ in P_a-s)

$$\mu_\ell = A_{52} + \frac{A_{53}}{T} + \frac{A_{54}}{T^2} + \frac{A_{55}}{T^3} \quad (4\text{-}8)$$

$A_{52} = 3.6522 \times 10^{-5}$

$A_{53} = 0.16626$

$A_{54} = -4.56877 \times 10$

$A_{55} = 2.8733 \times 10^{4}$

4.1.3 Salt Property Modeling

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

FLiBe (temperature in °C):

$$\rho = 2279.92 - 0.488 \cdot T \left(\frac{kg}{m^3}\right)$$
$$\mu = \frac{4.638 \times 10^5}{T^{2.79}} (kg/m \cdot s)$$
$$c_p = 2415.78 \ (J/kg \cdot K)$$
$$k = 0.7662 + 0.0005 \cdot T \ (W/m \cdot K)$$

(4-9)

FLiNaK (temperature in K):

$$\rho = 2729.3 - 0.73 \cdot T \left(\frac{kg}{m^3}\right), \quad 940 < T < 1170$$
$$\mu = 2.487 \cdot 10^{-4} \cdot e^{\frac{4478.62}{T}} \left(\frac{kg}{m} \cdot s\right), \quad 770 < T < 970$$
$$c_p = 1905.57 \ (J/kg \cdot K)$$
$$k = 0.36 + 5.6 \cdot 10^{-4} \cdot T \left(\frac{W}{m} \cdot K\right), \quad 790 < T < 1080$$

(4-10)

Dowtherm A (temperature in °C):

$$\rho = 1078 - 0.85 \cdot T \left(\frac{kg}{m^3}\right)$$
$$\mu = \frac{0.130}{T^{1.072}} (kg/m \cdot s)$$
$$c_p = 1518 + 2.82 \cdot T \ (J/kg \cdot K)$$
$$k = 0.142 - 0.00016 \cdot T \ (W/m \cdot K)$$

(4-11)

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

$$\frac{\partial \rho}{\partial T} = \frac{\rho(p, T+\varepsilon) - \rho(p, T-\varepsilon)}{2\varepsilon}, \quad \varepsilon = 0.5$$
$$\frac{\partial \rho}{\partial p} = 0$$
$$\beta = -\frac{1}{\rho}\frac{\partial \rho}{\partial T}$$
$$\frac{\partial C_p}{\partial T} = \frac{C_p(p, T+\varepsilon) - C_p(p, T-\varepsilon)}{2\varepsilon}, \quad \varepsilon = 0.5$$

(4-12)

$$H = H_0 + \int_{T_0}^{T} C_p dT$$

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

4.1.4 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 ^{o}C to 400 ^{o}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.

4.2 Convective Heat Transfer Correlations

4.2.1 Convective heat transfer modeling options in SAM

The closure models in SAM have been largely based on practices of existing nuclear reactor system codes while focusing on the effects of low-Prandtl number fluids existing in liquid-metal cooled fast reactor concepts. After reviewing the closure models used in many existing system codes and the available 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 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 4-1. The user is able to select correlations for an individual component by specifying it within that components input block. The code is set up to first determine whether a user-specified model is provided. If that is not given, then SAM will check whether the component has a pipe or rod-bundle flow geometry (pipe is default, if not specified). The fluid Prandtl number determines whether the fluid is a liquid metal or other fluid (Pr < 0.1). Beyond that, SAM will check if a specific model has been requested using "HTC_user_option", or will use the default options listed in Figure 4-1 in bold color font. Convective heat transfer correlations are applied in SAM as an auxiliary kernel so that the heat transfer coefficients are accessible to both fluid and solid components during runtime. Additional convective heat transfer models can be easily added in the future if deemed necessary. 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 (USNRC 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.

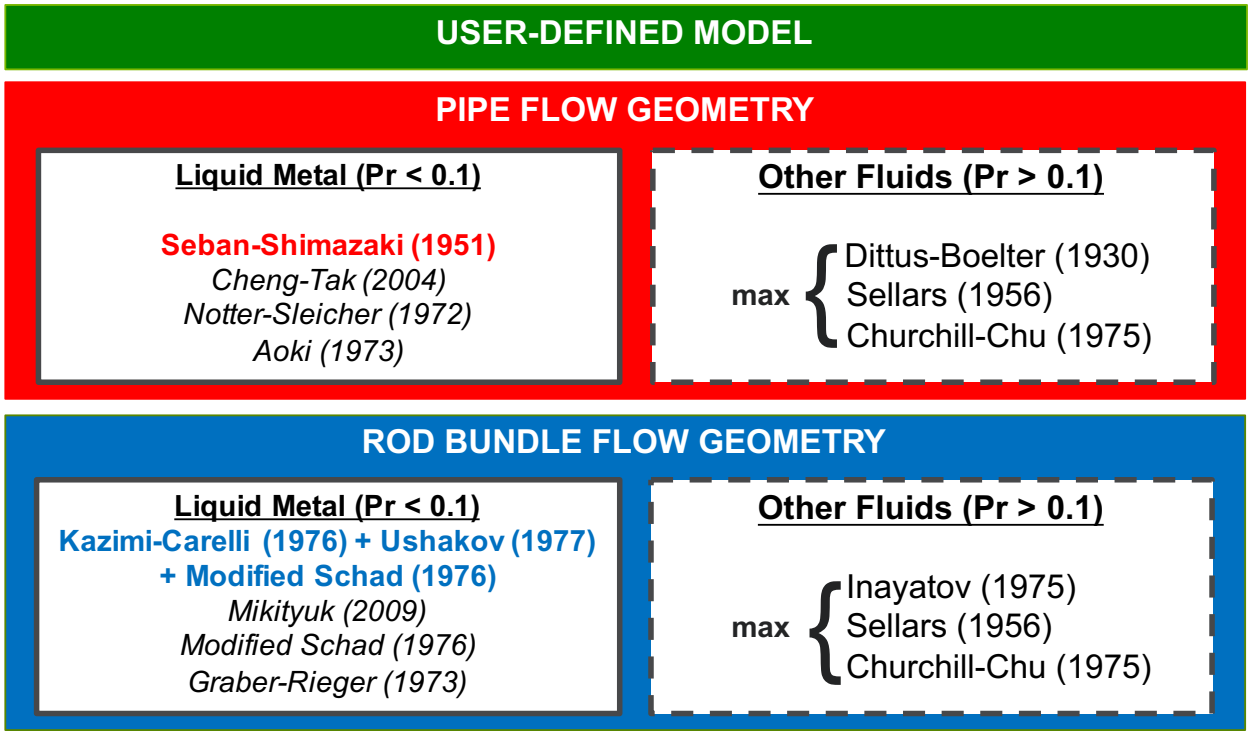


Figure 4-1: SAM modeling options for convective heat transfer

4.2.2 Low-Prandtl Fluids

The heat transfer characteristics of fluids are highly dependent on the Prandtl number, for which low-Prandtl fluids (Pr < 0.1) will 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.

4.2.2.1 Pipe flow – circular tube forced convection

The default model for pipe flow is the **Seban-Shimazaki (1951)** correlation. 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 flows across a majority of system thermal hydraulic codes. The correlation is often recommended based on its simple form and universal applicability. 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. Nonetheless, this form is semi-analytical in nature and is most likely an anticipated correlation that most users will expect to use.

$$Nu = 5.0 + 0.025Pe^{0.8} \; if \; Pe > 100,$$
$$Valid: 10^2 \leq Pe \leq 2\times10^4 \qquad (4\text{-}13)$$
$$Uncertainty: N.R. (No\ Reference)$$

The **Notter-Sleicher (1972)** correlation 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. This correlation is frequently used as a "reference" correlation when comparing other heat transfer models for pipe flow.

$$Nu = 4.8 + 0.0156\, Pe^{0.85} Pr^{0.08}$$
$$Valid: 10^4 \leq Re \leq 10^6, \qquad 0.004 \leq Pr \leq 0.1 \qquad (4\text{-}14a)$$
$$Uncertainty: \pm 5\%$$

$$Nu = 5 + 0.016\, Re^a Pr^b$$
$$a = 0.88 - \frac{0.24}{(4+Pr)}, \qquad b = 0.33 + 0.5e^{-0.6Pr} \qquad (4\text{-}14b)$$
$$Valid: 10^4 \leq Re \leq 10^6, \qquad 0.1 \leq Pr \leq 10^4$$
$$Uncertainty: \pm 10\%$$

The **Aoki (1973)** correlation 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.

$$Nu = 6.0 + 0.025\left[0.014\,Re^{1.45}Pr^{1.2}\left(1 - \frac{e^{-71.8}}{Re^{0.45}Pr^{0.2}}\right)\right]^{0.8}$$

$Valid: No\ reference\ available$ *(4-15)*

$Uncertainty:\ N.R.$

The **Cheng-Tak (2006)** correlation 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.

$$Nu = A + 0.018Pe^{0.80}$$

$$A = \begin{cases} 4.5, & Pe \le 1000 \\ 5.4 - 0.0009Pe, & 1000 \le Pe \le 2000 \\ 3.6, & Pe \ge 2000 \end{cases}$$ *(4-16)*

$Valid: Pe \le 6\times10^4$

$Uncertainty: N.R.$

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

$$Nu = \frac{48}{11} \approx 4.36$$

$Implemented\ if Pe < 10$ *(4-17)*

$Uncertainty: N.R.$

4.2.2.2 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 (1976)** correlation. 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. The following correlations are suggested as extensions for the default model to account for geometries that lie outside the applicable range of this correlation.

$$Nu = 4 + 0.16\left(\frac{P}{D}\right)^5 + 0.33\left(\frac{P}{D}\right)^{3.8}\left(\frac{Pe}{100}\right)^{0.86}$$

$Valid: 1.1 \le \frac{P}{D} \le 1.4, \qquad 10 \le Pe \le 5\times10^3$ *(4-18)*

$Uncertainty: N.R.$

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 **Ushakov (1977)** correlation (Pfrang and Struwe 2007) is suggested as an extension to the existing correlation given by Calamai (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, and should be implemented when appropriate as a default option (implemented for P/D > 1.4).

$$Nu = 7.55\,\frac{P}{D} - 20\left(\frac{P}{D}\right)^{-13} + \frac{0.041}{\left(\frac{P}{D}\right)^{2}} Pe^{\left(0.56+0.19\frac{P}{D}\right)}$$

$$Valid: 1.3 \le \frac{P}{D} \le 2.0, \qquad 0 \le Pe \le 4\times10^{3} \tag{4-19}$$

$$Uncertainty: \pm 5\%$$

The **Modified Schad** correlation, also by Kazimi and Carelli (1976), is suggested as an extension to the existing correlation given by Calamai (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:

For $150 \le Pe \le 10^3$:

$$Nu = \left[-16.15 + 24.96\left(\frac{P}{D}\right) - 8.55\left(\frac{P}{D}\right)^{2}\right] Pe^{0.3}$$

For $Pe \le 150$:

$$Nu = 4.496\left[-16.15 + 24.96\left(\frac{P}{D}\right) - 8.55\left(\frac{P}{D}\right)^{2}\right] \tag{4-20}$$

$$Valid: 1.05 \le \frac{P}{D} \le 1.5, \qquad Pe \le 1\times10^{3}$$

$$Uncertainty: N.R.$$

Recent work by **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.

$$Nu = 0.047\left(1 - e^{-3.8\left(\frac{P}{D}-1\right)}\right)\left(Pe^{0.77} + 250\right)$$

$$Valid: 1.1 \le \frac{P}{D} \le 1.95, \qquad 30 \le Pe \le 5\times10^3 \qquad (4\text{-}21)$$

$$Uncertainty: \pm 5\% \ (possibly\ less)$$

The **Graber-Rieger (1973)** correlation 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.

$$Nu = 0.25 + 6.2\,\frac{P}{D} + \left(0.032\left(\frac{P}{D}\right) - 0.007\right)Pe^{\left(0.8-0.024\frac{P}{D}\right)}$$

$$Valid: 1.25 \le \frac{P}{D} \le 1.95, \qquad 150 \le Pe \le 3\times10^3 \qquad (4\text{-}22)$$

$$Uncertainty: N.R.$$

4.2.3 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.

$$Nu = 0.023Re^{0.8}Pr^n, \qquad n = \begin{cases} 0.4, & heating \\ 0.33, & cooling \end{cases}$$

$$Valid: \frac{L}{D} \ge 10, \qquad Re \ge 10^5 (turbulent), \qquad 0.6 \le Pr \le 160 \qquad (4\text{-}23)$$

$$Uncertainty: \pm 15\%$$

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.

$$Nu = 0.023\left(\frac{P}{D}\right)Re^{0.8}Pr^n, \qquad n = \begin{cases} 0.4, & heating \\ 0.33, & cooling \end{cases}$$

$$Valid: 1.1 \le \frac{P}{D} \le 1.6, \qquad 6\times10^3 \le Re \le 10^6, \qquad 0.6 \le Pr \le 18 \qquad (4\text{-}24)$$

$$Uncertainty: N.R.$$

In the case of low flow conditions, fully developed laminar flow correlation **Sellars (1956),** as described in Eq. (4-16) is used again for non-liquid metal fluids.

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

$$Nu = \left[0.825 + \frac{0.387Ra^{\frac{1}{6}}}{\left(1 + (0.492/Pr)^{\frac{9}{16}}\right)^{\frac{8}{27}}}\right]^2 \qquad (4\text{-}25)$$

$Valid: Ra < 10^{12}$

$Uncertainty: N.R.$

For single-phase natural convection adjacent to a horizontal plane (laminar flow), the **McAdams (1954)** correlation 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. In SAM, this correlation is implemented, but currently not available for use.

$$Nu = 0.27Ra^{0.25}$$

$Valid: 10^5 < Ra < 10^{10}$ (4-26)

$Uncertainty: N.R.$

For non-liquid-metal fluid flow, the maximum value of the Nusselt numbers from the Dittus-Boelter (or Inayatov), Sellars, and Churchill-Chu correlations are used, as the same implemented in RELAP5.

4.2.4 User-specified heat transfer correlation option

A user-specified model is 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). Although the form may provide some seemingly-unnecessary parameters, the correlation was left in its form in trust that its inclusion was deemed necessary in the development of CATHARE.

$$Nu = Nu_0 + a\,(Re^b + c)\,Pr^d\,(1 + e\,Re^f)^{0.1} \qquad (4\text{-}27)$$

The above heat transfer correlations implemented in SAM are summarized in Table 4-1, in which the default correlations for different geometry and flow conditions are shaded.

Table 4-1: SAM convective heat transfer models

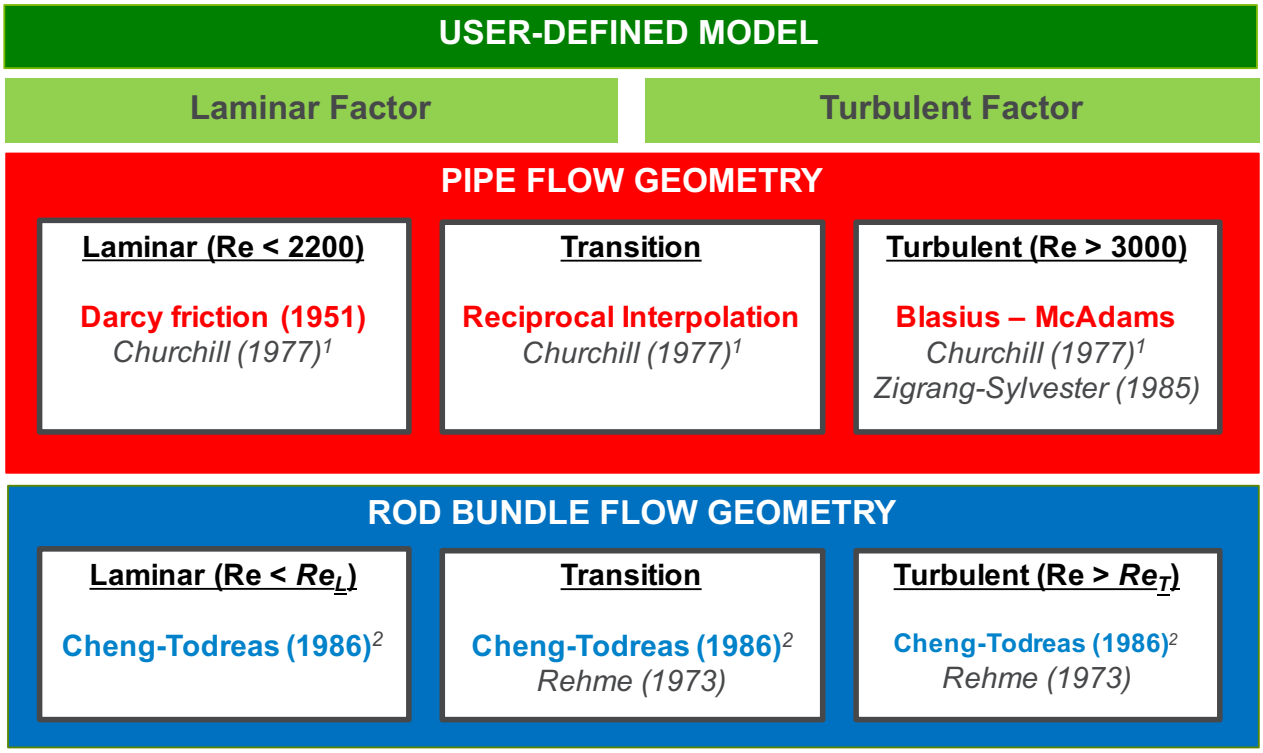
Name	Correlation	Re	Pr	Pe	P/D	Uncert.	Notes
Circular Pipe Geometry							
Dittus-Boelter (1930)	$Nu = 0.023\,Re^{0.8}Pr^{0.4}$	[1e5, inf)	[0.6, 160]		-	± 15%	*Default, for Pr > 0.1
Seban-Shimazaki (1951)	$Nu = 5.0 + 0.025Pe^{0.8}$			[1e2, 2e4]	-	N. R.	*Default, Forced, Turbulent
Sellars et al. (1956)	$Nu = \frac{48}{11} \approx 4.36$			[0, 10)	-	± 10% (for He)	*Default, Forced, Laminar
Aoki (1973)	$Nu = 6.0 + 0.025\left[0.014\,Re^{1.45}Pr^{1.2}\left(1 - \frac{e^{-71.8}}{Re^{0.45}Pr^{0.2}}\right)\right]^{0.8}$				-	-	Forced, Turbulent (MARS LMR)
Cheng-Tak (2006)	$Nu = A + 0.018\,Pe^{0.8}$ $A = \begin{cases} 4.5, & Pe \le 1000 \\ 5.4 - 0.0009Pe, & 1000 < Pe \le 2000 \\ 3.6, & Pe \ge 2000 \end{cases}$			[0, 6e4]	-	-	Forced, Turbulent (ANTEO+)
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)	$Nu = \left[0.825 + \frac{0.387Ra^{\frac{1}{6}}}{\left(1 + (0.492/Pr)^{\frac{9}{16}}\right)^{\frac{8}{27}}}\right]^2$	Ra: [0, 1e12]			-	N. R.	*Default for Pr > 0.1, Vert. Nat. Conv.
USER	$\boldsymbol{Nu = Nu_0 + a\left(Re^b + c\right)Pr^d\left(1 + e\,Re^f\right)^{0.1}}$	-	-	-	-	-	(CATHARE)
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)	$Nu = 4 + 0.16\left(\frac{P}{D}\right)^5 + 0.33\left(\frac{P}{D}\right)^{3.8}\left(\frac{Pe}{100}\right)^{0.86}$			[10, 5e3]	[1.1, 1.4]	N. R.	*Default within P/D range, Forced, Turbulent
Modified Schad (1976)	$Nu = \begin{cases} 4.496\left[-16.15 + 24.96\left(\frac{P}{D}\right) - 8.55\left(\frac{P}{D}\right)^2\right], Pe \le 150 \\ \left[-16.15 + 24.96\left(\frac{P}{D}\right) - 8.55\left(\frac{P}{D}\right)^2\right] Pe^{0.3}, 150 \le Pe \le 1000 \end{cases}$			[0, 1e3]	[1.05, 1.5]	-	*Default for P/D range outside K-C, Forced, Turbulent
Ushakov et al. (1977)	$Nu = 7.55\,\frac{P}{D} - 20\left(\frac{P}{D}\right)^{-13} + \frac{0.041}{\left(\frac{P}{D}\right)^2} Pe^{\left(0.56 + 0.19\frac{P}{D}\right)}$			[0, 4e3]	[1.3, 2.0]	± 5%	*Default for P/D range outside K-C, Forced, Turbulent
Mikityuk (2009)	$Nu = 0.047\left(1 - e^{-3.8\left(\frac{P}{D}-1\right)}\right)\left(\mathrm{Pe}^{0.77} + 250\right)$			[30, 5e3]	[1.1, 1.95]	± 5%	Forced, Turbulent (ANTEO, ATHLET)
Graber-Rieger (1973)	$Nu = 0.25 + 6.2\,\frac{P}{D} + \left(0.032\left(\frac{P}{D}\right) - 0.007\right) Pe^{\left(0.8 - 0.024\frac{P}{D}\right)}$			[150, 3000]	[1.25, 1.95]	-	Forced, Turbulent (ATHLET: HX)

4.3 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 and TRACE. 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 4-2. The user is able to select correlations for an individual component by specifying it within that components input block. If the "user-defined model" option is not selected, the code will auto-select the default wall friction model based on the geometry (pipe vs. rod bundle) and the flow Reynolds number. The Churchill (1977) correlation is the default wall friction model for pipe internal flow, and the Cheng-Todreas (1986) correlation is the default wall friction model for wire-wrapped rod bundle flow. 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 adjusting accordingly). These allow the user to 'simulate' the effects of complex geometry using existing correlations. These are specified with 'lam_factor' and 'turb_factor' in the component inputs.

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. As advanced reactor deployment brings the development of specific technologies (such as the advanced heat exchanger designs), additional correlations can be added in SAM to take into account specific geometry effects.



[1]: alternative (future) default option; [2]: simplified version;

Figure 4-2: SAM modeling options for wall friction coefficient

4.3.1 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 factor, resistance coefficient or simply friction factor and is four times larger than the Fanning friction factor. For laminar flows, it is considered valid and exact to the Colebrook equation.

$$f = \frac{64}{Re}, \quad for\ Re < 2200 \qquad (4\text{-}28)$$

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. It is only a function of Reynolds number, and it is a generic correlation that works for every fluid. It is often used in system codes:

$$f = \frac{0.316}{Re^{0.25}}, for\ 3 \cdot 10^3 \le Re \le 3 \cdot 10^4 \qquad (4\text{-}29)$$

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:

$$f = 0.184 \cdot Re^{-0.2}, \qquad for\ 3 \cdot 10^4 \le Re \le 1 \cdot 10^6 \qquad (4\text{-}30)$$

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. 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:

$$f = (1-w)f_{lam,Re_{min}} + wf_{turb,Re_{max}}$$
$$w = \frac{N}{Re_{min}} - \frac{N}{Re} \qquad (4\text{-}31)$$
$$N = \frac{Re_{min}Re_{max}}{Re_{max} - Re_{min}}$$

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 4-2: 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	3000	$3 \cdot 10^4$
	McAdams	$3 \cdot 10^4$	10^6

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 may be selected as the default model in SAM in future versions due to its ability to cover the entire flow regime range without need for interpolation, providing a more robust capability. It is the implemented default in TRACE as well. The Churchill correlation gives satisfactory accuracy to approximating the Colebrook-White equation.

$$f = 8\left[\left(\frac{8}{Re}\right)^{12} + \frac{1}{(A+B)^{\frac{3}{2}}}\right]^{\frac{1}{12}}$$

$$A = \left[2.457\, ln\frac{1}{\left(\frac{7}{Re}\right)^{0.9} + \frac{0.27\,\epsilon}{D}}\right]^{16}, B = \left(\frac{37530}{Re}\right)^{16} \qquad (4\text{-}32)$$

$$Valid: Re \geq 50$$

$$Uncertainty: \pm 3.2\%$$

In which ϵ is the pipe roughness.

Zigrang-Sylvester (1985) (INL 2012) is a user option for wall friction modeling suggested for modeling the turbulent flow regime. This correlation is the preferred option in RELAP5, and also shows satisfactory accuracy to describing pipe friction. It is only applicable to turbulent flow, thus requires interpolation for the transition regime.

$$\frac{1}{\sqrt{f}} = -2log_{10}\left\{\frac{\epsilon}{3.7D} + \frac{2.51}{Re}\left[1.14 - 2log_{10}\left(\frac{\epsilon}{D} + \frac{21.25}{Re^{0.9}}\right)\right]\right\} \qquad (4\text{-}33)$$

$$Valid: Re \geq 3\times10^3$$

$$Uncertainty: \pm 5.5\%$$

4.3.2 Wire-wrapped rod bundle geometry

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

$$f = C_{fL}/Re, for\ Re < Re_L$$

$$f = \frac{C_{fL}}{Re}(1-\psi)^{1/3} + \frac{C_{fT}}{Re^{0.18}}\psi^{1/3}\ for\ Re_L < Re < Re_T; \qquad (4\text{-}34)$$

$$f = C_{fT}/Re^{0.18}\ for\ Re > Re_T$$

with $Re_L = 300 * 10^{1.7(\frac{P}{D}-1)}$, $Re_T = 10^4 * 10^{1.7(\frac{P}{D}-1)}$, and $\psi = \log\left(\frac{\mathrm{Re}}{Re_L}\right)/\log\left(\frac{Re_T}{Re_L}\right)$.

For the simplified correlation,

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

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

In which $\frac{P}{D}$ is the pitch-to-diameter ration, and $\frac{H}{D}$ 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.

$$f = \left[\frac{64}{Re\sqrt{F}} + \frac{0.0816}{\left(Re\sqrt{F}\right)^{0.133}}\right] F \frac{P_b}{P_{assembly}}$$
$$F = \left(\frac{P}{D}\right)^{0.5} + \left[7.6\frac{D_m}{H}\left(\frac{P}{D}\right)^2\right]^{2.16} \qquad (4\text{-}35)$$

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

$$10^3 < Re < 3 \cdot 10^5 \ (transitory\ or\ turbulent\ flow\ regime)$$
$$8 < \frac{H}{D_m} < 50$$
$$1.1 < \frac{P}{D} < 1.42$$
$$7 < N_{rod} < 217$$

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.

4.3.3 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.

$$f = A + B * Re^{C} \qquad (4\text{-}36)$$

The above wall friction correlations are implemented in SAM and summarized in Table 4-3, in which the default and potentially future default correlations for different geometry and flow conditions are shaded.

Table 4-3: 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, RELAP, ATHENA, MARS
Churchill (1977)	$f = 8\left[\left(\frac{8}{Re}\right)^{12} + \frac{1}{(A+B)^{\frac{3}{2}}}\right]^{\frac{1}{12}}$ $A = \left[2.457\, ln\frac{1}{\left(\frac{7}{Re}\right)^{0.9} + \frac{0.27\epsilon}{D}}\right]^{16}$, $B = \left(\frac{37530}{Re}\right)^{16}$	[50, inf)				± 3.2	*Future default, All regimes, TRACE
Reciprocal Interpolation	$f = (1-w)f_{lam,Re_{min}} + wf_{turb,Re_{max}}$	[2200, 3000]					Transition, RELAP5
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, with $Re_L = 300 * 10^{1.7(\frac{P}{D}-1)}$, $Re_T = 10^4 * 10^{1.7(\frac{P}{D}-1)}$, and $\psi = \log\left(\frac{\mathrm{Re}}{Re_L}\right)/\log\left(\frac{Re_T}{Re_L}\right)$	[50, 1e6]	[1.025, 1.42]	[8, 50]	[19, 217]	± 14%	*Default, All regimes, ANTEO+, MARS
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, ANTEO+

5 Numerical Schemes

5.1 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:

$$\left(\frac{\partial \rho}{\partial t},\psi\right)=\left(\frac{\partial \rho}{\partial T}\frac{\partial T}{\partial t},\psi\right)$$
$$\left(\frac{\partial \rho u}{\partial t},\psi\right)=\left(u\frac{\partial \rho}{\partial T}\frac{\partial T}{\partial t}+\rho\frac{\partial u}{\partial t},\psi\right)$$
$$\left(\frac{\partial \rho H}{\partial t},\psi\right)=\left(H\frac{\partial \rho}{\partial T}\frac{\partial T}{\partial t}+\rho C_p\frac{\partial T}{\partial t},\psi\right)$$
$$\left(\frac{\partial (\rho u)}{\partial z},\psi\right)=-(\rho u,\nabla\psi)+\langle \rho u\hat{n},\psi\rangle \qquad (5\text{-}1)$$
$$\left(\frac{\partial (\rho uu)}{\partial z},\psi\right)=-(\rho uu,\nabla\psi)+\langle \rho uu\hat{n},\psi\rangle$$
$$\left(\frac{\partial (\rho uH)}{\partial z},\psi\right)=-(\rho uH,\nabla\psi)+\langle \rho uH\hat{n},\psi\rangle$$

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.

$$\int_V \nabla\cdot F\, dV=\int_S (F\cdot\hat{n})\, dS \qquad (5\text{-}2)$$

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*.

5.1.1 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,

$$\textit{Trapezoidal rule: } \int_a^b f(x)dx = (b-a)\frac{[f(a)+f(b)]}{2} \qquad (5\text{-}3)$$

$$\textit{Gaussian quadrature rule: } \int_a^b f(x)dx = \sum_{qp} f\left(x_{qp}\right)w_{qp} \qquad (5\text{-}4)$$

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 $O(h^2)$. 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),

$$Error = \int_a^b f(x)dx - \sum_{qp} f\left(x_{qp}\right)w_{qp} = \frac{(b-a)^{2n+1}(n!)^4}{(2n+1)[(2n)!]^3} f^{(2n)}(\xi),\ a<\xi<b. \qquad (5\text{-}5)$$

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.

5.1.2 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 – 2nd 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:

$$\frac{\partial u}{\partial t} = f(u,t) \qquad (5\text{-}6)$$

would result in:

$$f(u^{n+1},t^{n+1}) = \frac{u^{n+1}-u^n}{\Delta t} + O(\Delta t), \text{ Backward Euler or BDF1;} \qquad (5\text{-}7)$$

$$f(u^{n+1}, t^{n+1}) = \frac{\frac{3}{2}u^{n+1} - 2u^n + \frac{1}{2}u^{n-1}}{\Delta t} + O(\Delta t^2), \text{BDF2.} \quad (5\text{-}8)$$

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 and Hu 2017).

5.2 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:

$$F(u) = 0 \quad (5\text{-}9)$$

Where F represents the nonlinear equation system and u is the solution vector. Newton's method will be solving:

$$J(u^k)\delta u^k = -F(u^k) \quad (5\text{-}10)$$

given u^0, In which, $u^{k+1} = u^k + \delta u^k$, J is the associated Jacobian matrix and $J(u^k) = F'(u^k) = \left.\frac{dF(u)}{du}\right|_{u^k}$.

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

$$\left\| \frac{F(u^k)}{F(u^0)} \right\| < tolerance, \quad (5\text{-}11)$$

and/or a sufficient small Newton update:

$$\left\| \frac{\delta u^k}{u^k} \right\| < tolerance. \quad (5\text{-}12)$$

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

$$J_{ij} = \frac{\partial F_i(u)}{\partial u_j}. \quad (5\text{-}13)$$

In the JFNK approach, a Krylov method is used to solve the linear system of equations given by Eq. (5-10). An initial linear residual, r_0, is defined, given an initial guess, δu_0, for the Newton correction,

$$r_0 = -F(u) - J(u)\delta u_0 \quad (5\text{-}14)$$

Note that the nonlinear iteration index k (in Eq. (5-10)) has been dropped since the Krylov iteration is performed at a fixed k. Let j be the Krylov iteration index, the jth iteration δu_j is drawn from the subspace spanned by the Krylov vectors, $\{r_0, Jr_0, J^2 r_0, \ldots, J^{j-1} r_0\}$, and can be written as

$$\delta u_j = \delta u_0 + \sum_{i=0}^{j-1} \beta_i (J)^i r_0 \quad (5\text{-}15)$$

where the scalars β_i minimize the residual ($\left\| F(u) + J(u)\delta u_j \right\|$) in a least-squares sense.

The JFNK method does not require the formation of the Jacobian matrix, instead it only needs the result vector that approximates this matrix multiplied by a vector:

$$Jv \approx \frac{[F(u + \epsilon v) - F(u)]}{\epsilon} \quad (5\text{-}16)$$

where ϵ is a small perturbation.

However, in most applications, the Krylov subspace methods require preconditioning to be efficient. Using right preconditioning, Eq. (5-10) becomes

$$(JP^{-1})(P\delta u) = -F(u)\,, \quad (5\text{-}17)$$

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

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

$$(JP^{-1})w = -F(u), \quad (5\text{-}18)$$

for w. Then solve

$$\delta u = P^{-1} w. \quad (5\text{-}19)$$

for δu. Note that while we may refer to the precondition matrix P, operationally the algorithm only requires the action of P^{-1} on a vector. The right-preconditioned version of Eq. (5-16) is:

$$JP^{-1}v \approx \frac{[F(u + \epsilon P^{-1} v) - F(u)]}{\epsilon}. \quad (5\text{-}20)$$

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

1. Preconditioning: Solve for y in $Py = v$.
2. Perform matrix-free product $Jy \approx [F(u + \epsilon y) - F(u)]/\epsilon$.

5.3 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.

5.3.1 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):

$$
\begin{aligned}
r_p &= \left(\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial z},\ \psi\right) + \left(\rho \frac{\partial u}{\partial t} + \rho u \frac{\partial u}{\partial z} + \frac{\partial p}{\partial z} + \rho g + \frac{f}{D_e}\frac{\rho u|u|}{2},\ \tau_{PSPG} * \nabla\psi\right) \\
r_u &= \left(\frac{\partial \rho u}{\partial t} + \frac{\partial \rho u u}{\partial z} + \frac{\partial p}{\partial z} + \rho g + \frac{f}{D_e}\frac{\rho u|u|}{2}, \psi\right) \\
&\qquad + \left(\rho \frac{\partial u}{\partial t} + \rho u \frac{\partial u}{\partial z} + \frac{\partial p}{\partial z} + \rho g + \frac{f}{D_e}\frac{\rho u|u|}{2}, \tau_{SUPG} * \nabla\psi\right) \\
r_T &= \left(\frac{\partial \rho H}{\partial t} + \frac{\partial \rho u H}{\partial z} - q''', \psi\right) + \left(\rho C_p \frac{\partial T}{\partial t} + \rho C_p u \frac{\partial T}{\partial z} - q''', \tau_{SUPG} * \nabla\psi\right)
\end{aligned}
\qquad (5\text{-}21)
$$

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

$$
\begin{aligned}
J(p,p) &= \tau_{PSPG} \cdot \nabla\psi \\
J(u,u) &= \left(\rho \frac{d\frac{\partial u}{\partial t}}{du} + \frac{d\rho}{dT}\frac{\partial T}{\partial t}\right) \cdot \psi - 2\rho u \cdot \nabla\psi \pm \frac{f}{D_e}\rho u \cdot \psi \\
&\quad + \tau_{SUPG} \cdot \nabla\psi \cdot \left(\rho \frac{d\frac{\partial u}{\partial t}}{du} + \rho \frac{\partial u}{\partial z} + \rho u \frac{d\frac{\partial u}{\partial z}}{du} + \pm \frac{f}{D_e}\rho u\right) \\
J(T,T) &= \left(\rho C_p + \frac{d\rho}{dT} H\right) \cdot \frac{d\frac{\partial T}{\partial t}}{dT} \cdot \psi - \left(\rho C_p + \frac{d\rho}{dT} H\right) u \cdot \nabla\psi \\
&\quad + \tau_{SUPG} \cdot \nabla\psi \cdot \left(\rho C_p \frac{d\frac{\partial T}{\partial t}}{dT} + \rho C_p u \frac{d\frac{\partial T}{\partial z}}{dT}\right)
\end{aligned}
\qquad (5\text{-}22)
$$

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.

5.3.2 Preconditioning of Heat Conduction Equation

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

$$r_{T_s} = \left(\rho C_p \frac{\partial T}{\partial t} - \nabla(k\nabla T) - Q''',\ \psi\right) \tag{5-23}$$

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

$$J(T_s, T_s) = \rho C_p \cdot \frac{d\frac{\partial T}{\partial t}}{dT} \cdot \psi + k\nabla T \cdot \nabla\psi \tag{5-24}$$

5.3.3 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):

$$\begin{aligned} \Delta r_{solid} &= \langle h(T_s - T_f), \psi \rangle \\ \Delta r_{fluid} &= \left(h(T_s - T_f)\frac{P_{heated}}{A_c}, \psi + \tau_{supg} \cdot \nabla\psi\right) \end{aligned} \tag{5-25}$$

Note Δ is used together with residual, indicating the residual terms in Eq. (5-22) 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:

$$\begin{aligned} J(T_s, T_s) &\approx h\psi \\ J(T_s, T_f) &\approx -h\psi \\ J(T_f, T_s) &\approx -h\frac{P_{heated}}{A_c}(\psi + \tau_{supg} \cdot \nabla\psi) \\ J(T_f, T_f) &\approx h\frac{P_{heated}}{A_c}(\psi + \tau_{supg} \cdot \nabla\psi) \end{aligned} \tag{5-26}$$

In which T_f and T_s represent fluid and solid temperature, respectively. Note in Eq. (5-26) that the dependences 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 5-1, 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).

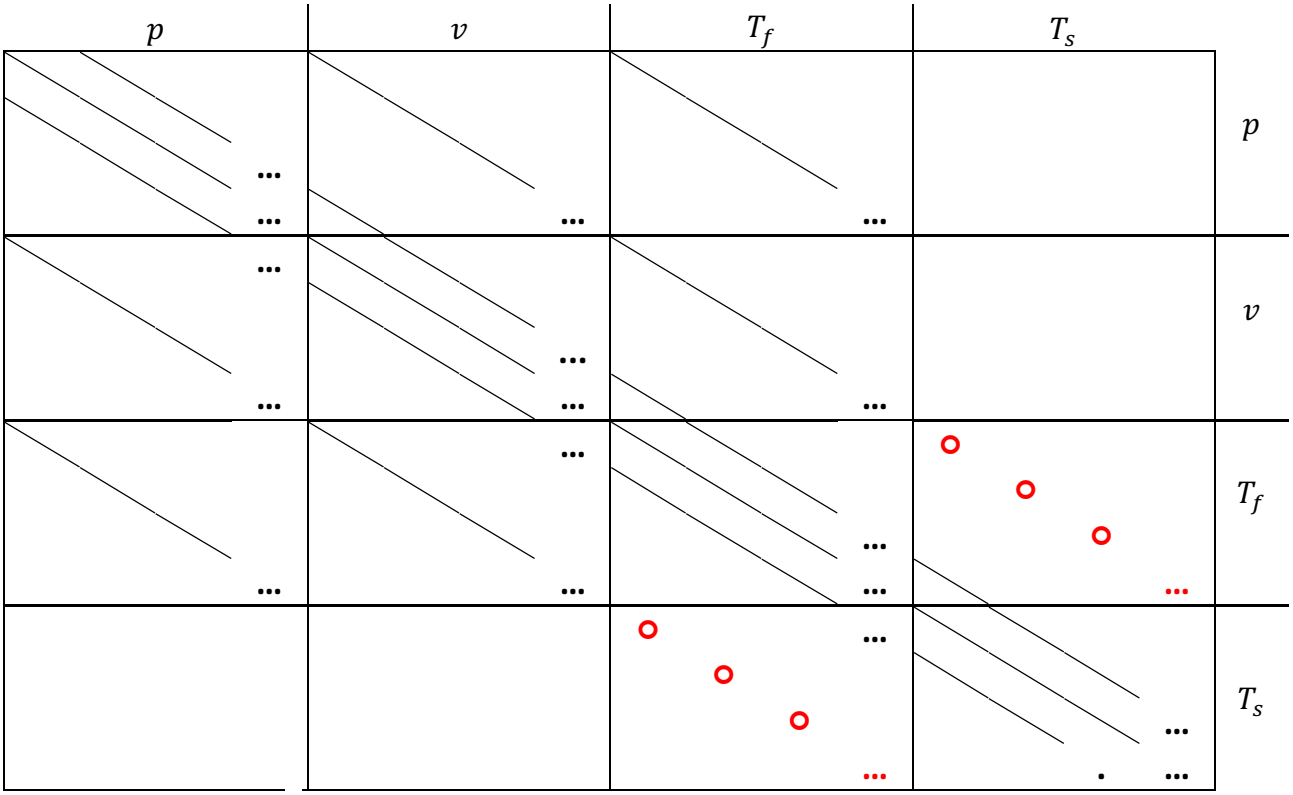


Figure 5-1: Preconditioning matrix for conjugate heat transfer problem
(with lines, circles, and dots representing non-zero entries in the matrix)

6 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 6-1. The physics models associated with these components will be briefly discussed later in this Chapter.

Table 6-1: 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
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

6.1 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.

6.2 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.

6.3 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.

6.4 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.

6.5 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.

6.6 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.*

6.7 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.

6.8 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 6-1. 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.

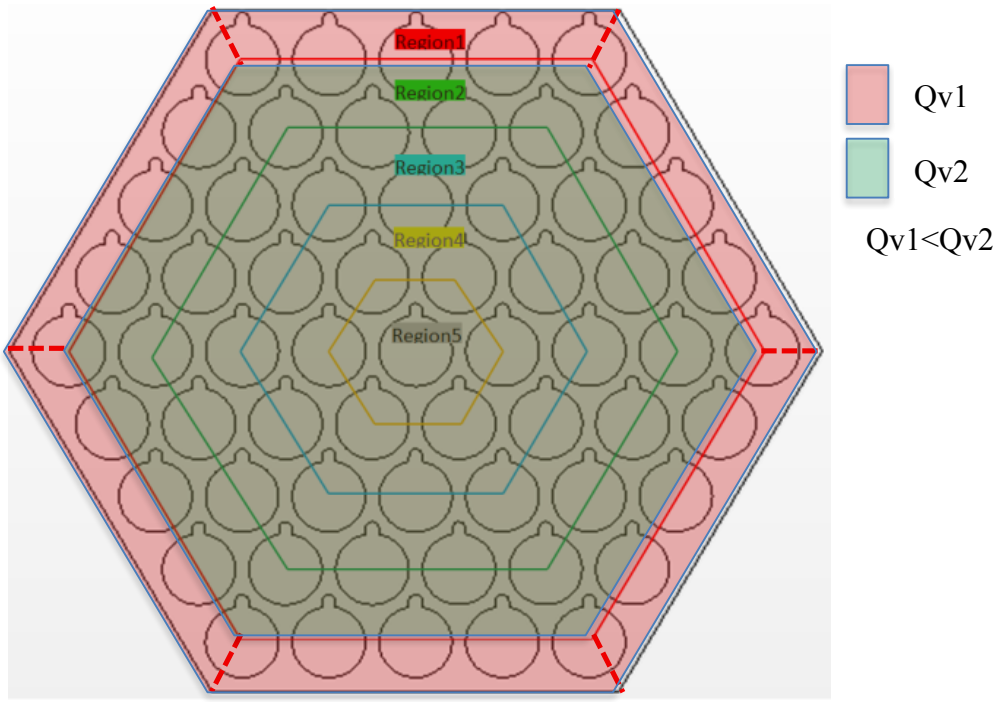


Figure 6-1: Sketch of the regions in the multi-channel model

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:

$$\frac{d\dot{Q}}{dz} = \beta(\rho v)_{avg} S(h_1 - h_2) \qquad (6\text{-}1)$$

in which, β is the mixing parameter (accounting for both turbulent mixing and directional flow); $(\rho v)_{avg}$ 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,

$$q'''_{mix} = \frac{1}{A_c}\frac{d\dot{Q}}{dz} = \beta(\rho v)_{avg}(h_1 - h_2)\frac{S}{A_c}. \quad (6\text{-}2)$$

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:

$$\Delta r_{energy} = \int_{\Omega} \psi \cdot q'''_{mix}\, d\Omega = \sum_{i,qp}\left(q'''_{mix} \times (\psi + \tau_{supg} \cdot \nabla\psi)\right), \quad (6\text{-}3)$$

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

6.9 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.

6.10 PBBranch

PBBranch models a 0-D flow junction where multiple 1-D fluid components are connected.

The governing equations of the *PBBranch* Component are:

$$\sum_{i=1}^{n}(\rho u A\vec{n})_i = 0 \quad (6\text{-}4)$$

$$u_{branch} = \frac{\sum_{i=1}^{n}(\rho u A)_i}{\rho_{branch} A_{branch}} \; if \; (u \cdot \vec{n})_i > 0 \quad (6\text{-}5)$$

$$H_{branch} = \frac{\sum_{i=1}^{n}(\rho u A H)_i}{\rho_{branch} u_{branch} A_{branch}} \; if \; (u \cdot \vec{n})_i > 0 \quad (6\text{-}6)$$

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,

$$P_{pipe} = P_{branch} + \Delta P_{acc} + \Delta P_{form} \tag{6-7}$$

If flow out of branch,

$$\begin{aligned} P_{pipe} &= P_{branch} + \Delta P_{acc} - \Delta P_{form} \\ (\rho u)_{pipe} &= \rho_{branch} u_{pipe} \\ (\rho u H)_{pipe} &= \rho_{branch} u_{pipe} H_{branch} \end{aligned} \tag{6-8}$$

In which,

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

$$\Delta P_{form} = \frac{1}{2} K (\rho u^2)_{pipe}$$

6.11 PBSingleJunction

PBSingleJunction is a special junction component, and it models a zero-volume flow joint where only two 1-D fluid components are connected. It thus does not need to model the mass, momentum, and energy conservations at the junction, but to assure that the two connecting nodes (1 and 2) have consistent boundary conditions.

$$\begin{aligned} (\rho u A)_1 &= (\rho u A)_2 \\ (p + \rho u^2)_1 &= (p + \rho u^2)_2 \\ (\rho u A H)_1 &= (\rho u A H)_2 \end{aligned} \tag{6-9}$$

6.12 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.

$$\dot{Q}_{pump} = P_{head} u_{pump} A_{pump} \tag{6-10}$$

6.13 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:

$$-\frac{d(\rho V)}{dt}+\sum_{i=1}^{n}(\rho uA\vec{n})_i=0 \quad (6\text{-}11)$$

$$-\frac{d(\rho HV)}{dt}+\sum_{i=1}^{n}(\rho uAH\vec{n})_i=0 \quad (6\text{-}12)$$

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. 6-5) 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,

$$P_{pipe}=P_{branch}+\Delta P_{acc}+\Delta P_{form}+\Delta P_{grav} \quad (6\text{-}13)$$

If flow out of branch,

$$P_{pipe}=P_{branch}+\Delta P_{acc}-\Delta P_{form}+\Delta P_{grav} \quad (6\text{-}14)$$

$$\Delta P_{grav,i}=\rho_{branch}\Delta H_i$$

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.

6.14 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:

$$L=\frac{P_{vol}-P_{gas}}{\rho_{vol}g} \quad (6\text{-}15)$$

$$V=V_{ref}+\left(L-L_{ref}\right)A_{ref} \quad (6\text{-}16)$$

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:

$$-\left(\rho_{vol}A\frac{dL}{dt}+V\ \frac{d\rho}{dT}\frac{dT}{dt}\right)+\sum_{i=1}^{n}(\rho uA\vec{n})_i=0 \qquad (6\text{-}17)$$

$$-\left(\rho_{vol}H_{vol}A\frac{dL}{dt}+V\ (H\frac{d\rho}{dT}+\rho C_p)\frac{dT}{dt}\right)+\sum_{i=1}^{n}(\rho huA\vec{n})_i=0 \qquad (6\text{-}18)$$

$$P_{vol}=P_{gas}+\rho_{vol}gL \qquad (6\text{-}19)$$

$$u_{vol}=\frac{\sum_{i=1}^{n}(\rho uA)_i}{\rho_{vol}A_{vol}}\ if\ (u\cdot\vec{n})_i>0 \qquad (6\text{-}20)$$

In which, $P, T, L, and\ u_{vol}$ are the primary state scalar variables; and $\rho_{vol}\ and\ V$ 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, $J(x_i, x_j)$, can be derived as the first order derivative of the Residuals of variable x_i over the variable x_j,

$$J(x_i,x_j)=\left(R_{x_i}\right)_{x_j}=\frac{\partial R_{x_i}}{\partial x_j} \qquad (6\text{-}21)$$

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

$$\begin{aligned}
&J(L,L)=-\rho_{vol}A\ \frac{d\dot{L}}{dL}-\ A\frac{d\rho}{dT}\frac{dT}{dt}\\
&J(L,T_{vol})=-\frac{d\rho}{dT}A\frac{dL}{dt}-V\frac{d\rho}{dT}\frac{d\dot{T}}{dT}\\
&J(L,p_i)=0\\
&J(L,u_i)=\rho_iA_i\cdot\overrightarrow{n_i}\\
&J(L,T_i)=0
\end{aligned} \qquad (6\text{-}22)$$

In which:

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

$$J(T_{vol},T_{vol}) = -\left(h\frac{d\rho}{dT}+\rho C_p\right)A\frac{dL}{dt} - V\left(h\frac{d\rho}{dT}+\rho C_p\right)\frac{d\dot{T}}{dT} - V\frac{dT}{dt}(2\frac{d\rho}{dT}C_p + \rho\frac{dC_p}{dT})$$

$$J(T_{vol},L) = -\rho_{vol}h_{vol}A\frac{d\dot{L}}{dL} - A\left(h\frac{d\rho}{dT}+\rho C_p\right)\frac{dT}{dt} \quad (6\text{-}23)$$

$$J(T_{vol},p_i) = 0$$

$$J(T_{vol},u_i) = \rho_i A_i h_i \cdot \vec{n_i}$$

$$J(T_{vol},T_i) = \rho_i A_i u_i Cp_i \cdot \vec{n_i}$$

$$J(u_{vol},u_{vol}) = 1$$

$$J(u_{vol},p_i) = 0$$

$$J(u_{vol},u_i) = \begin{cases} \dfrac{(\rho A\vec{n})_i}{\rho_{vol}A_{vol}}, & (u\vec{n})_i > 0 \\ 0\ , & (u\vec{n})_i < 0 \end{cases} \quad (6\text{-}24)$$

$$J(u_{vol},T_i) = 0$$

$$J(P_{vol},P_{vol}) = 1$$

$$J(P_{vol},T_{vol}) = -\frac{d\rho}{dT}gL$$

$$J(P_{vol},L) = -\rho_{vol}g \quad (6\text{-}25)$$

$$J(P_{vol},\ \ p_i\ or\ u_i\ or\ T_i) = 0$$

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

$$J(x_{i,volume},x_{j,volume})\big|_{i\neq j} = 0.0 \quad (6\text{-}26)$$

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.

6.15 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.

$$\Delta V_{gas} = \sum \Delta V_{liquid\_volume,i} \quad (6\text{-}27)$$

$$P_{gas} V_{gas}^{\gamma} = constant \quad (6\text{-}28)$$

6.16 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:

$$\frac{d(\rho V H)}{dt} + \sum_{i=1}^{n} \dot{Q}_i = 0 \quad (6\text{-}29)$$

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,

$$\dot{Q} = \int h_{conv}(T_w - T_{vol})\, dA\,. \quad (6\text{-}30)$$

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,

$$\dot{Q} = \dot{m}_{mix} \Delta H. \quad (6\text{-}31)$$

In which,

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

6.17 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.

6.18 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.

6.19 CoupledTDV

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

6.20 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.

6.21 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*.

6.22 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 (6-1) - (6-3).

6.23 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.

7 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.

7.1 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 (ABTR, 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. (2014).

7.2 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). With its origin as SAS1A in the late 1960s, the SAS series of codes has been under continuous use and development for over forty-five years and is currently maintained by the U.S. Department of Energy under the Office of Advanced Reactor Technology.

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, an 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).

7.3 Coupling with Additional Codes

Although the flexible coupling interface is implemented in SAM, additional efforts are required to complete the coupling with additional specific physics simulation tools, such as Nek5000, PROTEUS, BISON, etc. The coupling with these codes is under consideration, but has not been completed or demonstrated.

The Nek5000 computational fluid dynamics solvers are based on the spectral element method developed by Maday and Patera (1989). Nek5000 supports two different formulations for spatial and temporal discretization of the Navier-Stokes equations. The first is the P_N-P_{N-2} method with velocity/pressure spaces based on tensor-product polynomials of degree N and N-2 respectively. The second is the low-Mach number formulation of Tomboulides et al. (1997), which uses consistent order-N approximation spaces for both the velocity and pressure. The low-Mach number formulation is also valid at the zero-Mach (incompressible) limit (Tomboulides et al., 1989). The Nek5000 code has been extensively verified and validated for several benchmark problems and has a proven scalability in existing petascale architectures using over a million MPI ranks (and over a billion degrees-of-freedom).

PROTEUS is a high-fidelity deterministic neutron transport code based on the second-order even-parity formulation (Shemon et al., 2014). The application scope targeted for PROTEUS ranges from the homogenized assembly approaches prevalent in current reactor analysis methodologies to explicit geometry approaches, with the ability to perform coupled calculations to thermal-hydraulics and structural mechanics. The PROTEUS solver has a proven capability of using existing petascale parallel machines to solve problems with demonstrated scalability of over 70% (strong scaling) at over 250,000 processors (on BlueGene/P). These achievements of PROTEUS were made possible by partitioning the space-angle system of equations over the available processors and utilizing established iterative solution techniques from the neutron transport community combined with the parallel algorithms in the PETSc toolbox. Both PROTEUS and Nek5000 are part of the SHARP toolkit, and the coupling between PROTEUS and Nek5000 has been applied in SFR core simulations (Merzari et al. 2015).

BISON (Williamson et al., 2012) is a MOOSE based nuclear fuel performance code applicable to a variety of fuel forms including light water reactor fuel rods, TRISO particle fuel, and metallic rod and plate fuel. It solves the fully-coupled equations of thermomechanics and

species diffusion, for either 1D spherical, 2D axisymmetric or 3D geometries. Fuel models are included to describe temperature and burnup dependent thermal properties, fission product swelling, densification, thermal and irradiation creep, fracture, and fission gas production and release. Plasticity, irradiation growth, and thermal and irradiation creep models are implemented for clad materials. Models are also available to simulate gap heat transfer, mechanical contact, and the evolution of the gap/plenum pressure with plenum volume, and fission gas addition. Because SAM is developed based on the MOOSE framework, the coupling with other MOOSE based codes is straightforward by utilizing the MOOSE *MultiApp* system (Gaston et al., 2015).

ACKNOWLEDGEMENTS

The author sincerely thanks Dr. Thomas Fanning for his valuable comments and suggestions in preparing this document. Special thanks to Mr. Travis Mui for his assistance in enhancing heat transfer and wall friction models in SAM during his internship at Argonne. This work is funded by U.S. Department of Energy Office Nuclear Energy Advanced Modeling and Simulation (NEAMS) program.

REFERENCES

[1] Aoki S., 1973. "Current liquid-metal heat transfer research in Japan", Progress in heat and mass transfer, Vol.7, Heat transfer in liquid metals, Ed. O.W. Dwyer, Pergamon Press, pp. 569-587.

[2] Balay S., Brown J., et al, 2017. PETSc Web page, http://www.mcs.anl.gov/petsc.

[3] Berry R. A., Peterson J. W., Zhang H., et al., 2015. "RELAP-7 Theory Manual," INL/EXT-14-31366, Idaho National Laboratory.

[4] Brooks A.N., Hughes T.J.R., 1982. "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.

[5] Bubelis E., Perez-Martin S., Passerini S., et al. 2016. "IAEA NAPRO Coordinated Research Project: Heat Transfer and Pressure Drop Correlations for Sodium Cooled Systems," ICAPP 2016, San Francisco, CA.

[6] Calamai, G.J., et al., 1974. "Steady State Thermal and Hydraulic Characteristics of the FFTF Fuel Assemblies," ARD-FRT-1582, Westinghouse Electric Corporation.

[7] CD-adapco, 2012. "STAR-CCM+ 7.06 Manual," CD-adapco Ltd, N.Y.

[8] Chang Y. I., Finck P. J., Grandy C., et al., 2006. "Advanced Burner Test Reactor Preconceptual Design Report," ANL-ABR-1 (ANL-AFCI-173), Argonne National Laboratory.

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

[10] Chen S.K., Todreas N.E., Nguyen N.T., 2014. "Evaluation of existing correlations for the prediction of pressure drop in wire-wrapped hexagonal array pin bundles," Nuclear Engineering and Design 267, 109– 131.

[11] Cheng S.K., Todreas N.E. 1986. "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.

[12] Cheng X., Tak N.I. 2006. "Investigation on turbulent heat transfer to lead–bismuth eutectic flows in circular tubes for nuclear applications", Nucl. Eng. Des. 236, 385– 393.

[13] Choi S. and Kim S. 2008. Numerical Validation of Heat Transfer Correlations for Design of the Intermediate Heat Exchanger in a Sodium Cooled Fast Reactor. Transactions of the Korean Nuclear Society Autumn Meeting, PyeongChang, Korea, October 30-31, 2008.

[14] Churchill S.W. 1977. Friction-factor equation spans all fluid-flow regimes. Chemical Engineering 7, 91–92.

[15] Churchill S.W., Chu H.H.S., 1975. "Correlating Equations for Laminar and Turbulent Free Convection From a vertical Plate", International Journal of Heat and Mass Transfer. Vol. 18, pp. 1323-1329.

[16] Davis C. B., Shieh A. S., 2000. "Overview of the Use of ATHENA for Thermal-Hydraulic Analysis of Systems with Lead-Bismuth Coolant," ICONE-2000, Baltimore, MD.

[17] Fanning T.H. and Hu R., 2016. Coupling the System Analysis Module with SAS4A/SASSYS-1: ANL/NE-16/22, ANL-ART-74, Argonne National Laboratory.

[18] Fanning, T.H. (Ed.), 2012. The SAS4A/SASSYS-1 Safety Analysis Code System: ANL/NE-12/4. Argonne National Laboratory.

[19] Gaston D., et al. 2015. "Physics-based multiscale coupling for full core nuclear reactor simulation." Annals of Nuclear Energy 84, 45-54.

[20] Gaston D., Newman C., Hansen G., and Lebrun-Grandi´e D., 2009. "MOOSE: A parallel computational framework for coupled systems of nonlinear equations," Nuclear Engineering and Design, vol. 239, pp. 1768–1778.

[21] Gräber V.H., Rieger M. 1972. Experimentelle Untersuchung des Wärmeübergangs an Flüssigmetalle (NaK) in parallel durchströmten Rohrbündeln bei konstanter und exponentieller Wärmeflussdichteverteilung. Atomkernenergie (ATKE) 19, 23–40.

[22] Ha K.S., Jeong H.Y., Cho C., et al., 2010. "Simulation of the EBR-II Loss-of-Flow Tests Using the MARS Code," Nuc. Tech. 169, 134-142.

[23] Hu R. and Yu Y., 2016. A Computationally Efficient Method for Full-Core Conjugate Heat Transfer Modeling of Sodium Fast Reactors. Nuclear Engineering and Design, Vol. 308, 182-193.

[24] Hu R., Fanning T.H., Sumner T., Yu Y., 2015. Status Report on NEAMS System Analysis Module Development, ANL/NE-15/41, Argonne National Laboratory.

[25] Hu R., Thomas J. W., and Fanning T. H., 2013. Strategy for Multi-Scale Single-Phase Flow Coupling. ANL/NE-13/4, Argonne National Laboratory.

[26] Hu R., Thomas J. W., Munkhzul E., Fanning T. H., 2014. "Coupled System and CFD Code Simulation of Thermal Stratification in SFR Protected Loss-Of-Flow Transients," In Proceedings of ICAPP 2014, Charlotte, USA.

[27] Hu, R., 2015. An advanced one-dimensional finite element model for incompressible thermally expandable flow. Nuclear Technology, 190 (3), 313–322.

[28] Hu, R., 2017. A fully-implicit high-order system thermal-hydraulics model for advanced non-LWR safety analyses. Annals of Nuclear Energy, Vol. 101, 174–181.

[29] Inayatov A. Y., 1975. "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, vol. 7, no. 3, pp. 84–88.

[30] Incropera F. P., DeWitt D. P., Bergman T. L., Lavine A. S., 2007. Fundamentals of Heat and Mass Transfer. John Wiley & Sons, Inc.

[31] INL, 2012. RELAP5-3D Code Manual Volume 1: Code Structure, System Models and Solution Methods, Idaho National Laboratory, INEEL-EXT-98-00834, Rev. 4, June 2012.

[32] Kahaner, D., Moler, C., Nash, S., 1989. Numerical Methods and Software. Prentice-Hall, New Jersey, USA.

[33] Kazimi M. S. and Carelli M. D. 1976. "Heat Transfer Correlation for Analysis of CRBRP Assemblies. " Westinghouse, CRBRP-ARD-0034.

[34] Kirk B. S., Peterson J. W., Stogner R. H., et al., 2006. "libMesh: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations," Engineering with Computers, 22(3-4): 237-254.

[35] Knoll D. A. and Keyes D. E., 2004. "Jacobian-free Newton-Krylov Methods: a Survey of Approaches and Applications," Journal of Computational Physics, Vol. 193, pp.357–397.

[36] Lodi F., Grasso G., Mattioli D., Sumini M., 2016. "ANTEO+: A subchannel code for thermal-hydraulic analysis of liquid metal cooled systems," Nucl. Eng. Des 301, 128-152.

[37] Maday Y. and Patera A.T., 1989. "Spectral element methods for the Navier-Stokes equations," in A.K. Noor and J.T. Oden, editors, State-of-the-Art Surveys in Computational Mechanics, pp. 71–143, ASME, New York.

[38] McAdams W.H., 1954. Heat Transmission (Third ed.). New York: McGraw-Hill. p. 180.

[39] Merzari E., Shemon E., Yu Y., et al., 2015. Multi-Physics Demonstration Problem with the SHARP Reactor Simulation Toolkit: ANL/NE-15/44. Argonne National Laboratory.

[40] Mikityuk K., 2009. Heat transfer to liquid metal: review of data and correlations for tube bundles, Nuclear Engineering and Design 239, 680 – 687.

[41] Moureau V., Berat C., and Pitsch H., 2005. "An Efficient Semi-Implicit Compressible Solver for Large-Eddy Simulations," Center for Turbulence Research, Annual Research Briefs 2005, 43-55, Stanford University.

[42] Notter R.H., Sleicher C.A. 1972. A Solution to the Turbulent Graetz Problem - III. Fully Developed and Entry Region Heat Transfer Rates. Chem. Eng. Sci., 27, 2073-2093.

[43] Peterson P., et al., 2013. "Temperature-Dependent Thermophysical Properties for Fluoride Salts and Simulant Fluids," CIET-DESIGN-001-03, UC Berkeley.

[44] Pfrang W., Struwe D., 2007. Assessment of Correlations for Heat Transfer to the Coolant for Heavy Liquid Metal Cooled Core Designs. FZK report, FZKA 7352.

[45] Polidori M., 2010. "Implementation of Thermo-Physical Properties and Thermal-Hydraulic Characteristics of Lead-Bismuth Eutectic and Lead on CATHARE Code," ENEA.

[46] Rehme K., 1973. Pressure drop correlations for fuel element spacers, Nuclear Technology 17, 15 – 23.

[47] Seban R. A., Shimazaki T. T., 1951. Heat transfer to a fuid flowing turbulently in a smooth pipe with walls at constant temperature. Transactions of the ASME, Vol. 73, 803-809.

[48] Sellars, J., Tribus, M., Klein, J., 1956. “Heat Transfer to Laminar Flow in a Round Tube on Flat Conduit. The Graetz Problem Extended,” Trans. Am. Soc. Mech. Engrs., 78, 441.

[49] Shakib, F., 1988. Finite element analysis of the compressible euler and navier-stokes equations (Ph.D. thesis). Stanford University.

[50] Shemon E.R., Smith M.A., Lee C., 2014. PROTEUS-SN Methodology Manual. Argonne National Laboratory, ANL/NE-14/5.

[51] Tezduyar T.E., 1992. "Stabilized Finite Element Formulations for Incompressible Flow Computations," Advances in Applied Mechanics, Vol. 28, 1-44.

[52] The Engineering Toolbox, 2017. Air Properties. The Engineering Toolbox, http://www.engineeringtoolbox.com/air-properties-d_156.html.

[53] Todreas N.E., Kazimi M.S., 2012. Nuclear Systems. Volume 1: Thermal Hydraulic Fundamentals - Second Edition, CRC Press, Taylor & Francis Group.

[54] Tomboulides A.G., Israeli M., Karniadakis G.E., 1989. “Efficient removal of boundary-divergence errors in time-splitting methods,” Journal of Scientific Computing, 4:291–308.

[55] Tomboulides A.G., Lee J.C.Y., and Orszag S.A., 1997. “Numerical simulation of low Mach number reactive flows,” Journal of Scientific Computing, 12:139–167, June 1997.

[56] US NRC, 2010. TRACE V5.0 Theory Manual, U.S. Nuclear Regulatory Commission.

[57] Ushakov P., Zhukov A., Matyukhin N., 1977. "Heat transfer to liquid metals in regular arrays of fuel elements," High Temperature, vol. 15, pp. 868-873.

[58] Williamson R. L., Hales J. D., Novascone S. R., et al., 2012. “Multidimensional Multiphysics Simulation of Nuclear Fuel Behavior,” Journal of Nuclear Materials, vol. 423, pp. 149–163.

[59] Yang, W.S., Joo, H.G., 1999. LMR core temperature calculation based on implicit formulation of the ENERGY model and a Krylov subspace method. Ann. Nucl. Energy 26, 629–640.

[60] Yu, Y., Merzari, E., Obabko, A., Thomas, J., 2015. A porous medium model for predicting the duct wall temperature of sodium fast reactor fuel assembly. Nucl. Eng. Des. 295, 48–58.

[61] Zhou C., Huber K., Cheng X., “Validation of the modified ATHLET code with the natural convection test of the PHENIX reactor,” Ann. Nuc. Eng. 59, 31-46, 2013.

[62] Zou, L., Peterson, J., Zhao, H., et al., 2013. Solving implicit multi-mesh flow and conjugate heat transfer problems with RELAP-7. In: Proceedings of M&C 2013, Sun Valley, Idaho, May 5–9.

Nuclear Engineering Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439

www.anl.gov

Argonne National Laboratory is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC