Pacific Northwest
NATIONAL LABORATORY

PNNL-33688

COBRA-SFS Version 6.2.1 User Guide

A Thermal Hydraulic Analysis Code for Spent Fuel Storage and Transportation Casks

October 2023

1David J Richmond
2Brian M Hom
3Ben J Jensen
4Christopher L Grant

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL-33688

COBRA-SFS Version 6.2.1 User Guide

A Thermal Hydraulic Analysis Code for Spent Fuel Storage and Transportation Casks

October 2023

1David J Richmond
2Brian M Hom
3Ben J Jensen
4Christopher L Grant

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Acknowledgments

The Authors would like to acknowledge the Department of Energy Integrated Waste Management program for funding the work to develop Version 6.2 and and the continued development and distribution of COBRA-SFS. Additionally Sarah Suffield and Steve Ross put in time and effort to review this report. Finally the authors would like to acknowledge the many code developers that have worked on COBRA-SFS and laid the groundwork to bring the code to this point especially Tom Michener, Judith Cuta and Dave Rector for their mentorship and expertise.

Acronyms and Abbreviations

BWR	Boiling Water Reactor
CFD	Computational Fluid Dynamics
COBRA-SFS	Coolant boiling in Rod Arrays – Spent Fuel Storage
DOE	United States Department of Energy
FD	Finite Difference
LWR	Light Water Reactor
ISFSI	Independent Spent Fuel Storage Installation
IWM	Integrated Waste Management
NRC	United States Nuclear Regulatory Commission
PWR	Pressurized Water Reactor

Contents

Figures

Tables

1.0 Introduction

This document is an updated and reduced version of the complete documentation of the COBRA-SFS thermal analysis code (COolant Boiling in Rod Arrays – Spent Fuel Storage) (Michener et al. 1995). This manual provides the analyst with input instructions and fundamental guidelines for contrasting models using COBRA-SFS. COBRA-SFS is designed to analyze the dry storage and transportation casks that house spent or used nuclear fuel. The primary focus and application is on square light water reactor fuel assemblies, however the code also has the capability to model hexagonal assemblies and is well suited to any geometry that can be defined as an extruded 2-D map.

COBRA-SFS uses a lumped-parameter, finite-difference approach to predict flow and temperature distributions in spent fuel storage systems and fuel assemblies, under forced and natural convection heat transfer conditions, in both steady-state and transients. Derived from the COBRA family of codes (Rowe 1973; Stewart et al. 1977; George et al. 1980; Khan et al. 1981), which have been extensively evaluated against in-pile and out-of-pile data, COBRA-SFS retains all the important features of the COBRA codes for single-phase analysis,[1] and extends the range of application to problems with two-dimensional radiative and three-dimensional conductive heat transfer. With these added capabilities, COBRA-SFS has been used to analyze various single- and multi-assembly spent fuel storage systems containing unconsolidated and consolidated fuel, with a variety of fill media (Cuta et al. 1984; Lombardo et al. 1986b; Cuta and Creer 1986; Wiles et al. 1986; Rector et al. 1986c; McKinnon et al. 1986; Wheeler et al. 1986).

Cycle 0 of COBRA-SFS was released in 1986. Subsequent applications of the code required the development of additional capabilities, leading to the release of Cycle 1 in February 1989. Since then, the code has been subjected to an independent technical review as part of a submittal to the Nuclear Regulatory Commission (NRC) for a generic license to apply the code to spent fuel storage system analysis. Minor modifications and error corrections were developed in response to the reviewers' recommendations. In addition, new capabilities and improvements to the code had been developed, and these changes were combined to form a new release of the code, Cycle 2, in September 1995 (Michener et al. 1995). Subsequently, additional validation and verification work was done to confirm the applicability of the code to transient evaluations, leading to the release of Cycle 3 in 1998 (Rector et al. 1998).

This report constitutes the user oriented documentation of Version 6.2 of COBRA-SFS. It includes a complete set of input instructions, and provides the user with guidance for model construction and "best practices" to fully utilize the capabilities of the code for evaluation of spent fuel storage and transportation. Code theory and solution approach are covered extensively in (PNNL-32245).

The organization of this manual is intended to make it relatively easy to use. It is divided into major sections focused on either input instructions or model development information and organized as follows:

[1] COBRA-SFS is not applicable to two-phase flow analysis as currently formulated. The models for subcooled boiling, the effect of phase change on the energy equation, and momentum effects of phase slip have been removed.

Section 2 *Input Instructions*

Section 3 *Model Development Guide*

Section 4 *Materials Properties*

Section 5 *Fuel Library*

Section 6 *References*

The Model Development guide should be carefully read by any new user to the code because it distills some of the many “lessons learned” in years of experience at Pacific Northwest National Laboratory (PNNL). This code has been used for building models for a large range of complex systems. The materials data will also help modelers by providing commonly used material property and fuel geometry data.

2.0 User Manual

This section is the User Manual for COBRA-SFS, Version 6.2 It contains the line-by-line input instructions for the code, and guidance for the user concerning the important parameters to consider in developing input for a spent fuel storage system model from “the ground up.” Users needing only to access previously created model templates for application of the UNF-ST&DARDS tool should consult the appropriate documentation for that system. An introduction to the overall modeling approach, which may be unfamiliar to new users, is presented in Section 2.3. The input instructions for Version 6.2 of the code are presented in Section 2.4. The auxiliary code RADGEN, used to generate view factor input for the radiative heat transfer model, is described in Section 2.4.7.1. This section includes the input instructions for RADGEN. Additionally, Version 6.2 utilizes an optional simplified input methodology which is outlined in Section 2.5. Section 2.6 contains a description of the common block dimension arrays in COBRA-SFS.

2.1 Version 6.0 Changes

Unless specifically noted there are no significant changes in subsequent minor releases of Version 6. The major changes in Version 6.0 are focused on continuing to make the code simpler and more efficient to use for the analyst. The largest structural change is the full integration of RADGEN into the COBRA-SFS source code. This streamlines the model development and execution process because there is now one input file for the entire model, instead of the need to run RADGEN and generate tape10 files separately. Because the code still generates and reads a tape10 file there is full backwards compatibility for running models that were generated for previous COBRA-SFS versions. One major input change for the models is that the group order as listed in the input manual is now required for models to run correctly. If the groups are not in order the code may not execute some of the setup calculations correctly.

Another major change is that subgroup labels (i.e. *slab.3) are now required for maximum performance. These need to be placed in columns 96-102. What this does is allow the code to better read the input file and determine the necessary array sizes correctly. To maintain maximum backwards compatibility, the code will still run if these line endings are not in place however, it will use an excessive amount of memory and take much longer to run than it should.

2.2 Version 5.0 Changes

Version 5.0 released some major changes and marks a significant move towards a more modern, user friendly, stable and efficient code. Historically COBRA-SFS was only distributed as source code due to the need to change parameters for different model sizes. Starting in Version 5.0 the source code has undergone a major restructuring. COBRA-SFS now takes advantage of the FORTRAN language’s modules and allocatable arrays. The common block structure has been completely eliminated, with the variables now explicitly declared in a central module. Similarly the major arrays that store input and/or calculated parameters have been transitioned to allocatable arrays that are sized by reading the input file.

One key advantage of this from the user perspective is that virtually any size model can be run with the source code as distributed. Additionally the code will always be distributed with a Windows executable along with source code. It is still recommended that the user have a

FORTRAN compiler and debugger to assist in finding errors however, this change will make the cost and time barriers to startup much lower.

2.3 Basic Structure of COBRA-SFS Modeling

COBRA-SFS is a computer code of the class sometimes referred to as “legacy” codes. In this context, “legacy” is a polite euphemism for “old.” For COBRA-SFS, this means that the user interface is neither graphical nor particularly friendly even to experienced users, and can seem actively hostile to the uninitiated. The input is highly structured, with exacting format requirements, and has some peculiar limitations that reflect the basic features of data that was once entered on punched Hollerith[1] cards. For users accustomed to modern codes with tailored GUIs, creating an input file for the COBRA-SFS code can at first seem a little like playing chess blindfolded. Against Bobby Fisher.

Years of experience with the code, during which model builders have made almost every input error possible, often multiple times, have led to the development of various useful internal input checks and advisory messages to help diagnose problems. In addition, there are typical sequential steps that the user should adhere to, which may seem arbitrary at first, but have a certain inevitable logic, in retrospect, once the user has gained some familiarity with the process. Section 3.0 discusses these steps, providing user guidance that outlines an effective “plan of attack” for developing and debugging a new model. But for that advice to be useful, the user must first develop a familiarity with the input requirements and basic structure of a typical model. The best way to do this is to slog through the input instructions in Section 2.4, withholding judgement on the rhyme or reason of the system.

As described in Section 1, COBRA-SFS is a subchannel code, originally designed for thermal-hydraulic analysis of reactor cores and fuel assemblies. The basic structure of any model, therefore, is a series of parallel channels that allow fluid flow along heat generating structures (such as fuel rods) and other solid structures that can conduct heat from or to the fluid. Modifications and enhancements to COBRA-SFS expand this basic capability to include features specific to multi-assembly spent fuel storage systems, such as shipping casks and storage canisters, but the user should bear in mind that these modifications are built on the basic subchannel structure, and that affects how these features could be incorporated into a COBRA-SFS model. The greatest strength of the subchannel modeling approach is that it is intrinsically flexible, and consequently the code is not tailored to a specific cask geometry or coordinate system convention. It has been applied to a wide variety of cask and canister designs, and has successfully accommodated a wide range of geometric configurations.

The flexibility of the subchannel modeling approach and the deliberate generality of the input structure in the code, however, can be a substantial hurdle for the new user. The validation test cases for Cycle 4 (see COBRA-SFS Theory Manual Section 7.0 (Richmond et. al. 2021)) provide illustrative examples of how the code can be applied to specific multiple assembly shipping/storage casks. In addition to these examples, Section 2.3.1 discusses cask modeling optimization, and illustrates some of the main considerations in setting up a cask model. A summary of previously published reports on COBRA-SFS applications is given in Section 2.3.2. The discussion in Section 2.3.2 uses examples from section-of-symmetry models for large systems, but this is merely a reflection of the limitations of computer memory capabilities in the

[1] Readers unfamiliar with this term are referred to Wikipedia (https://en.wikipedia.org/wiki/Punched_card), for background information on this data entry mode.

1980s, when the majority of the modeling work prior to the release of Cycle 2 was performed. Current computer capabilities are such that full 360 models are the standard approach used for most applications, and are specified as a requirement for models developed as templates for UNF-ST&DARDS.

Guidance on how to deal with a calculation in which the code fails to understand the input instructions presented to it is included in Section 2.3.3.

2.3.1 COBRA-SFS Cask Model Optimization

A detailed study of COBRA-SFS modeling optimization has been published elsewhere (Rector and Michener 1989). This section summarizes the results and recommendations of that study, and presents parts of the discussion that might be of interest to a new user of the code. In general, the modeling flexibility and extensive capabilities of the code often tempt the inexperienced user into setting up a more complicated and detailed model than may actually be required for an application. By developing a basic understanding of the modeling features in the code, the user can learn to construct a model that will give the best results for the least effort, both in setup time and computation time.

The accuracy of an analysis with a COBRA-SFS model of a spent fuel storage system is highly dependent on noding distribution and the number and locations of temperature points in the system. Although the code can handle very large problems consisting of hundreds of channels, rods, and axial levels, there is a practical upper limit on model size, generally dictated by computer memory requirements. Also, the larger the problem, the longer it will take to set up and verify the input, and the longer it will take to run; these are constraints in both time and memory costs.

There are basically two areas in model development where the user has significant choices that affect computational efficiency. These are in the geometry modeling for assembly, channel, slab, and axial nodalization, and in the optimization of the numerical solution. The following subsections discuss the considerations and trade-offs to be made in model optimization for these two areas.

2.3.1.1 Cask Model Optimization: Geometry Considerations

The accuracy of the results depends on having enough nodes to adequately resolve the problem, but costs in run-time and memory requirements increase in direct proportion to the number of nodes. Therefore, the objective of model optimization as far as the geometry is concerned is to obtain the desired level of accuracy with the minimum number of nodes. There are basically two ways to reduce problem size: 1) reduce the actual problem to be solved by modeling only a section of symmetry rather than the whole system, and 2) select the noding such that a given node represents the largest possible region that can be characterized with a single temperature (or in the case of a fluid node, a single temperature, velocity, and pressure).

A COBRA-SFS cask model is constructed by dividing the cask into a set of finite volumes or nodes (see COBRA-SFS Theory Manual Section 2.0 (Richmond et. al. 2021) for a detailed discussion of model discretization), consisting of assemblies which are made up of channels to represent the fluid flow paths and rods to represent the fuel. The channels are bounded by solid conduction nodes (aka slabs) that represent the solid material of the cask, such as assembly canisters, support baskets, shielding layers, and inner and outer structural shells. Solid

components not in direct contact with the fluid can also be represented as a network of connected slab nodes, to appropriately capture thermal gradients.

Most cask designs can be described as basically consisting of a large right circular cylinder containing a basket consisting of a square grid of tubes or vertical plates designed to hold spent fuel assemblies or canisters containing assemblies with damaged fuel rods. Typically, assemblies are in their nominal “as built” configuration, but in some cases, cells of the basket may contain canisters full of rods extracted from disassembled bundles (e.g., a limited number of high burn-up test rods, or consolidated fuel rods). The number of nodes required for appropriate resolution of the system is highly problem dependent, and requires the user to develop some intuitive sense of where there might be important thermal gradients in the system that require greater resolution in the noding map. However, it never hurts to augment such experience-based insight with actual noding sensitivity studies for such models.

In cases where problem size may be a significant factor, the model can be constructed in a way that takes advantage of inherent symmetries in the cross-section to reduce the volume of the system that must be represented. There are three conditions that must be satisfied for symmetry in the model. The geometry must be symmetric, the decay heat generation must be the same in geometrically symmetric regions, and the boundary conditions must be the same on geometrically symmetric regions. Geometric symmetry is readily verified by inspection of a cross-sectional diagram of the system. Figure 2.1 shows a diagram of the TN24P cask, which has geometric symmetry on four different axes. Examples of noding patterns that take advantage of this symmetry are shown for a half-section of the cask in Figure 2.2, and for an eighth section in Figure 2.3. All other things being equal, the eighth-section model would give the same results as a full cask model, and would require only about one-eighth the number of nodes. Alternatively, a more detailed model could be constructed for the one-eighth section using the same number of nodes as would be required for a less refined model representing the entire cask. Even if model size does not matter so much for run-time and memory requirements, a partial symmetry model of greater resolution could provide a valuable independent check on the appropriate level of refinement for the full 360° model.

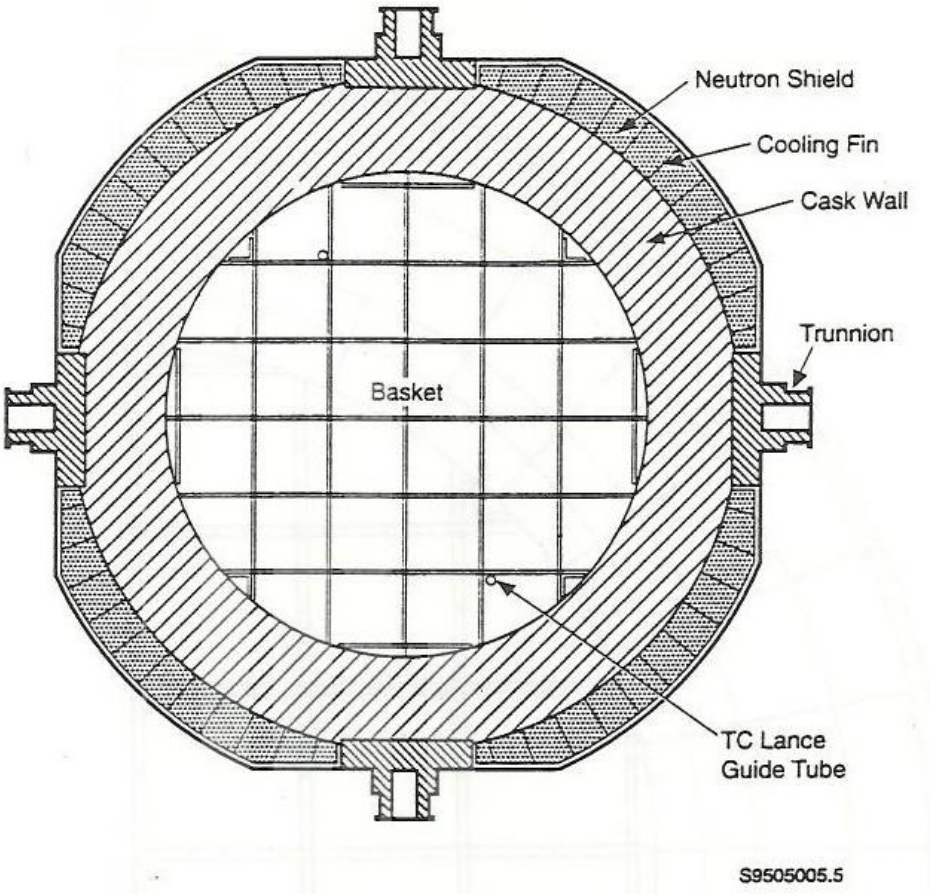


Figure 2.1. TN24P Cask Cross Section

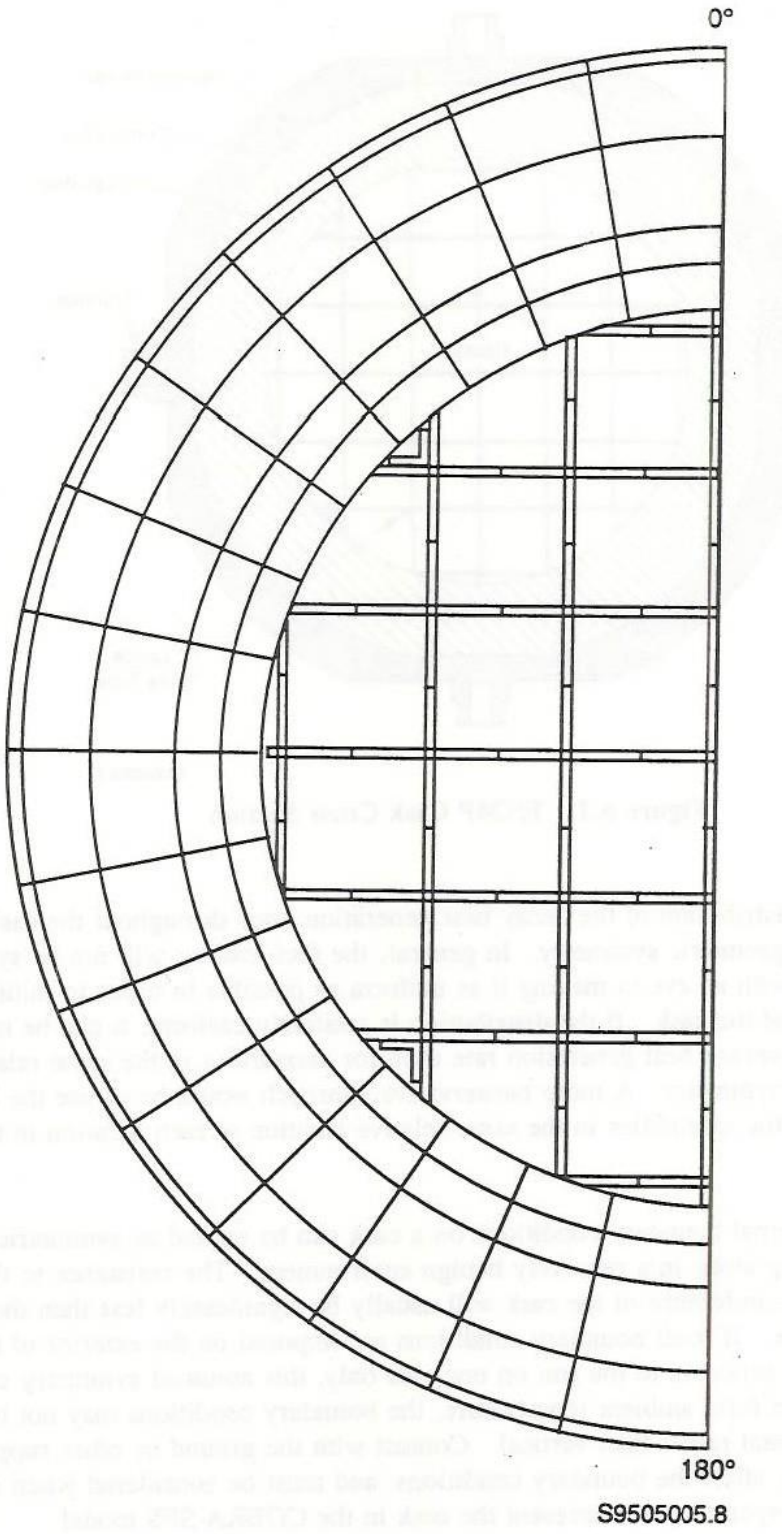


Figure 2.2. TN24P One-Half Section of Symmetry Cask Model

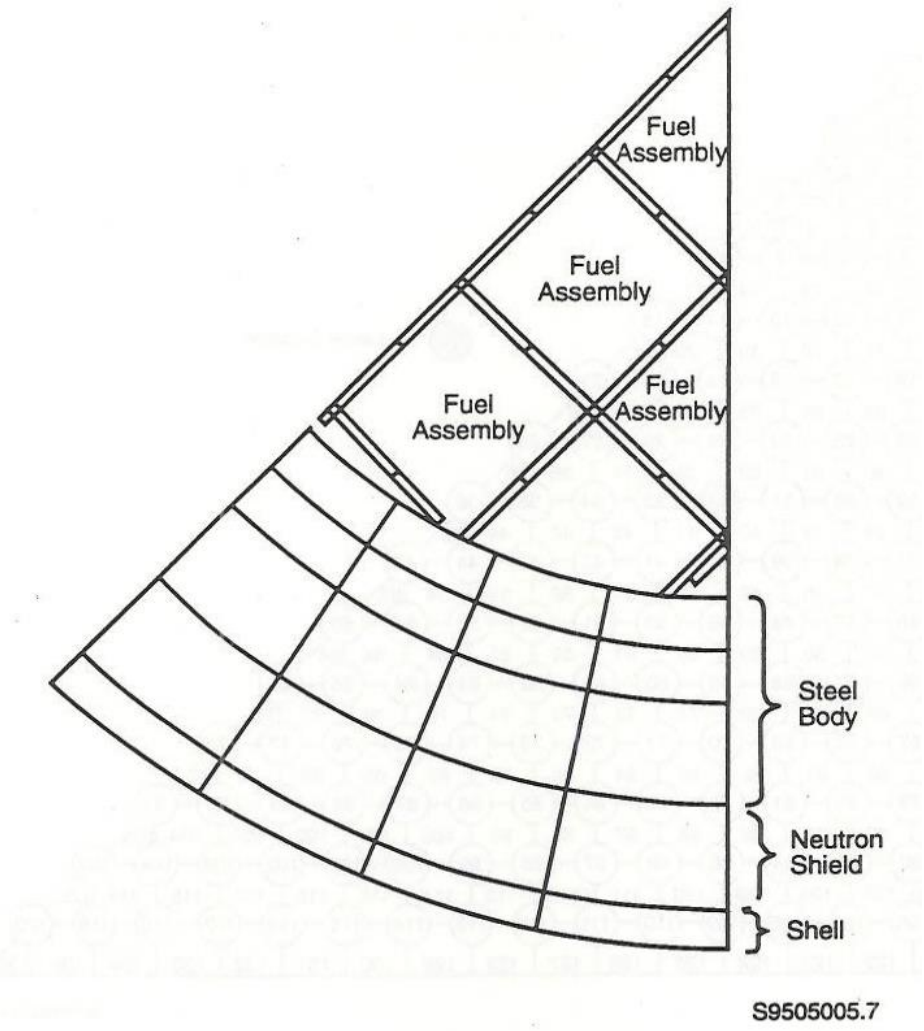


Figure 2.3. TN24P One-Eighth Section of Symmetry Cask Model

Symmetry in the distribution of the decay heat generation rates throughout the cask is usually more problematic than geometric symmetry. This is the main reason for the requirement for full symmetry models in the COBRA-SFS templates constructed for the UNF-ST&DARDS tool. In general, the decay heat in all fuel assemblies loaded in a given cask will not be the same, even for canisters that are licensed only for nominally “uniform” loading, and do not have specific alternative preferential loading patterns defined in their CoC and SAR documents. In practice, the definition of uniform loading for spent fuel canisters makes some reasonable allowance for the fact that spent fuel assemblies do not generally have identical decay heat, even when they are from the same cycle, and spent fuel canisters are often loaded with fuel from more than one cycle, and are consequently of different ages. Actual loading patterns in a given canister are determined based on a number of criteria, such as criticality and dose/exposure limits, and not merely on thermal characteristics. However, it should be noted that fuel assemblies are usually positioned within the basket such that the distribution of decay heat generation is as radially symmetric as possible, to encourage optimum uniform heat removal from the cask in the radial direction, to avoid unnecessary “hot spots” on the canister shell.

In general, the external boundary conditions on a cask can be treated as symmetric, especially if the cask is simply sitting alone in a relatively benign environment. The resistance to thermal conduction around the circumference of the cask will usually be significantly less than the resistance to the ambient temperature. If local boundary conditions are imposed on the exterior of the cask, however, such as a fire or exposure to the sun on one side only, this assumed symmetry can be destroyed. Even with a uniform ambient temperature, the boundary conditions may not be symmetrical if the cask is horizontal rather than vertical. Contact with the ground or other supporting structures can significantly affect the boundary conditions, and must be considered when determining the appropriate section of symmetry to represent the cask in the COBRA-SFS model.

Once an appropriate symmetry has been selected for the model (full or partial), the detail required for resolution of the flow field and temperatures within the assemblies and cask structure must be determined. The level of detail needed in the model depends on the type of information to be obtained from the analysis. Typically, the user wants to determine the peak clad temperature, and often the radial and axial distributions of temperature as well, for the fuel rod cladding and in many cases for other components of the system, such as basket plates, support structures, and shielding material. To obtain an accurate estimate of the temperature distribution for the fuel rods, the geometry resolution of the model should include all rods and all subchannels, including wall and corner channels, as illustrated in Figure 2.4 for a typical 17x17 pressurized water reactor (PWR) fuel bundle.

It is then necessary to assure accurate modeling of the resistances to radial heat flow in the slab noding connecting the walls of the basket cell to the canister internal components and exterior shell. This is as important as the detailed rod-and-subchannel representation of the assembly, and is often much more difficult to obtain and verify.

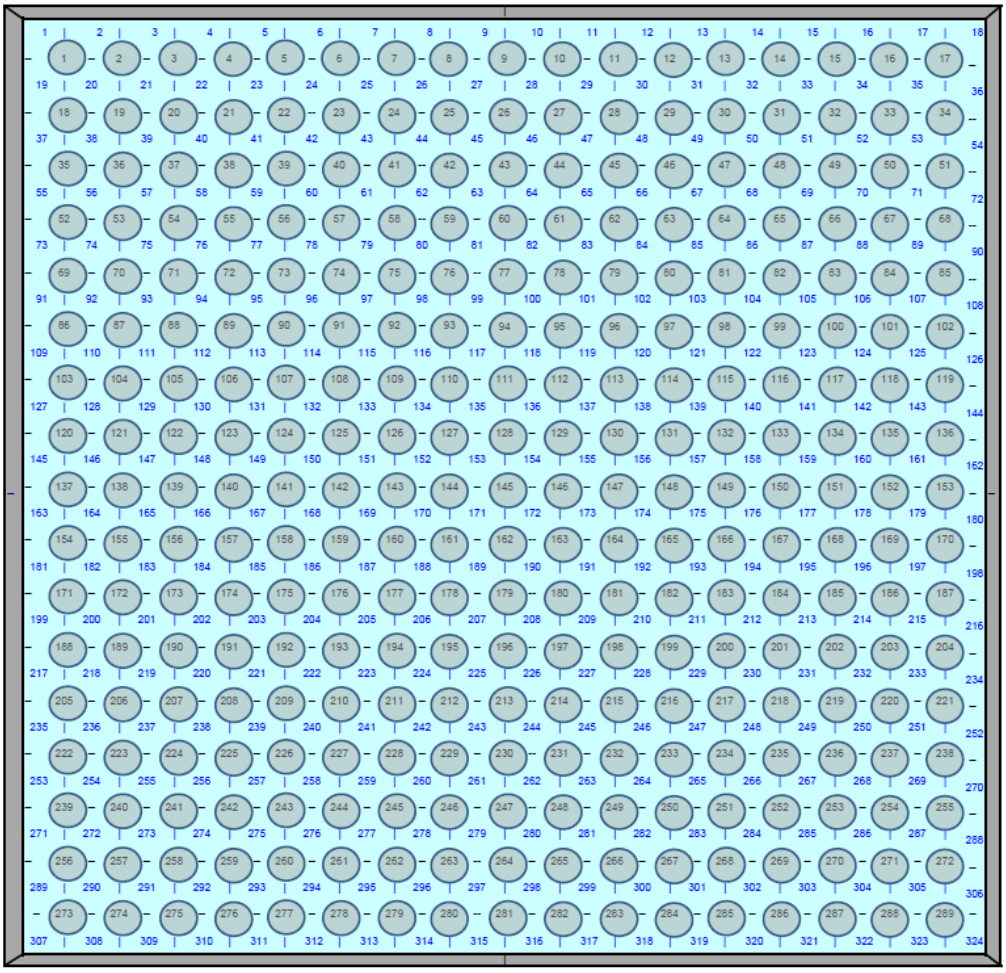


Figure 2.4. Rod-and-Subchannel Model for 17x17 PWR Fuel Assembly (NOTE: Diagram not to scale; rod-to-rod spacing greatly exaggerated for clarity)

The importance of accurate representation of the conduction paths requires that sufficient slab nodes be included in the model. Heat transfer between slab nodes is modeled by assuming a constant heat flux between node points, which implies a linear temperature gradient over that length, if the nodes have the same thermal conductivity. If the heat flux or temperature gradient in a region is relatively constant, then a single node-to-node connection may be sufficient, but in general the slab noding must be detailed enough to resolve any significant non-uniformity in the temperature gradient through the material. The thermal properties of the material being modeled are the best guide to determining the appropriate slab noding resolution, but in some cases sensitivity studies may be useful as well.

Since the basket structures of typical spent fuel canisters are composed of relatively thin plates, it is generally sufficient to represent them with a single node layer. Baskets in some cases are composed of layers of material, generally steel or aluminum plates that are fastened together by intermittent welding or an axial array of short bolts. The material of the basket also usually includes thin neutron poison plates, which are in some manner affixed to the inner faces of the basket cell openings, often by a welded cover plate, but in many cases by bolts or screws. For any geometry consisting of layered plates (for the basket itself or the neutron poison plates), the modeling must take into account the effect of gaps between layers of material, due to fabrication

constraints and differential thermal expansion of dissimilar materials. The happiest solution to this issue is a basket composed of a single material, such as a metal matrix composite that simultaneously provides structural support and neutron absorbing capability in a single layer of material. As of this writing, only one major vendor has licensed such a design, and only for a limited subset of storage canister configurations in their product line.

In a typical COBRA-SFS model, each layer is represented with a single node layer, subdivided in the axial direction with a minimum of two nodes per basket cell face. Figure 2.5 illustrates a typical COBRA-SFS model of the basket and support structure for a canister loaded with boiling water reactor (BWR) fuel assemblies (including channel boxes). Figure 2.6 shows a similar illustration for a canister loaded with PWR fuel assemblies. Figure 2.5 and Figure 2.6 show basket designs consisting of thin plates, and therefore are represented with a single layer of nodes in the COBRA-SFS model. Figure 2.7 shows a model for a basket design that includes multiple layers of metal plates, with individual plates represented with separate slab nodes. Thermal connections between these nodes allow appropriate representation of gaps between the layers of material, which can have a significant effect on radial heat flow from the fuel assemblies to the basket material.

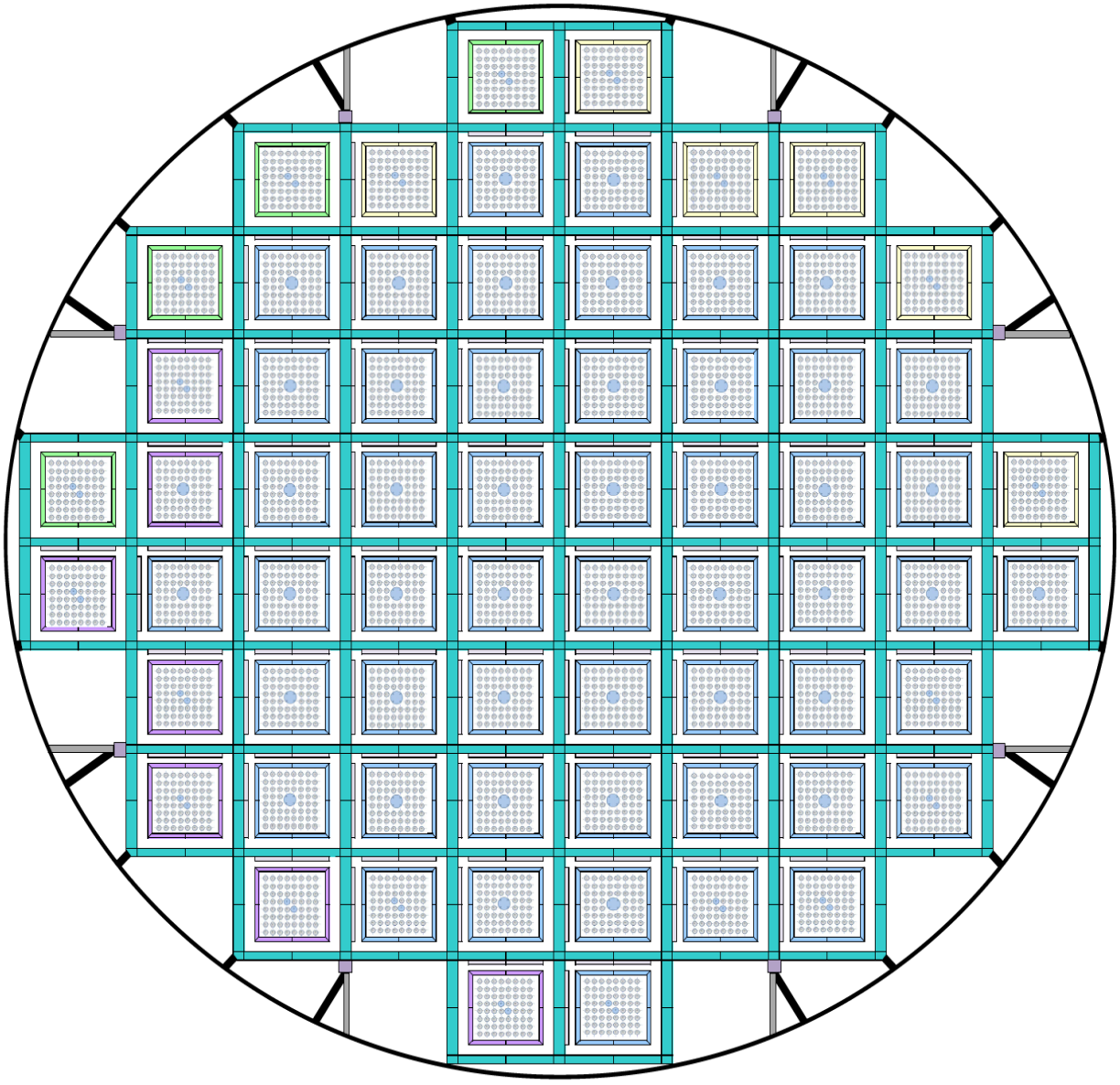


Figure 2.5. Typical COBRA-SFS Model for Canister Loaded with BWR Fuel Assemblies, including Channel Boxes (NOTE: Diagram not to scale; node thicknesses greatly exaggerated for clarity)

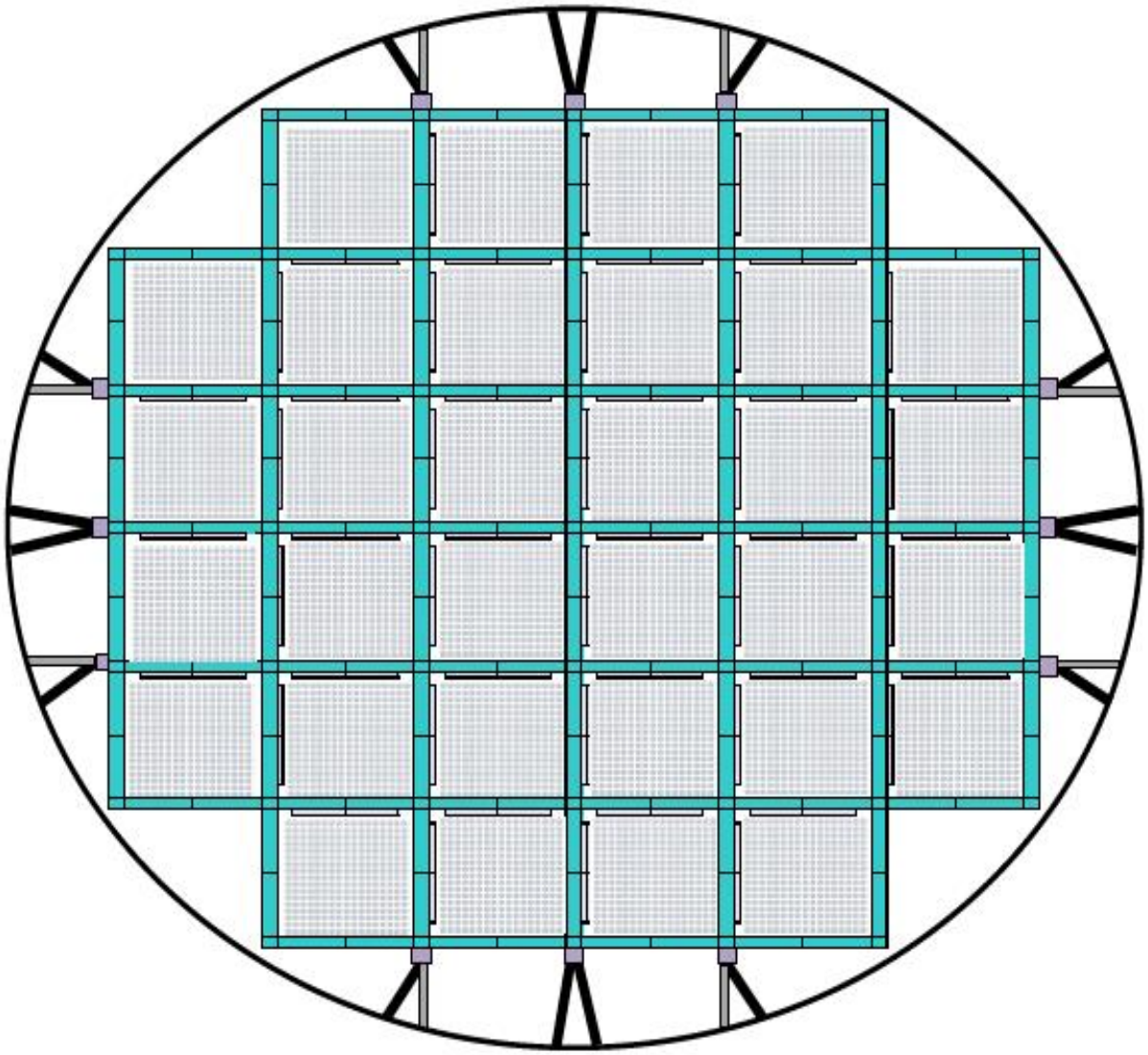


Figure 2.6. Typical COBRA-SFS Model of Canister Loaded with PWR Fuel Assemblies (NOTE: Diagram not to scale; node thicknesses greatly exaggerated for clarity)

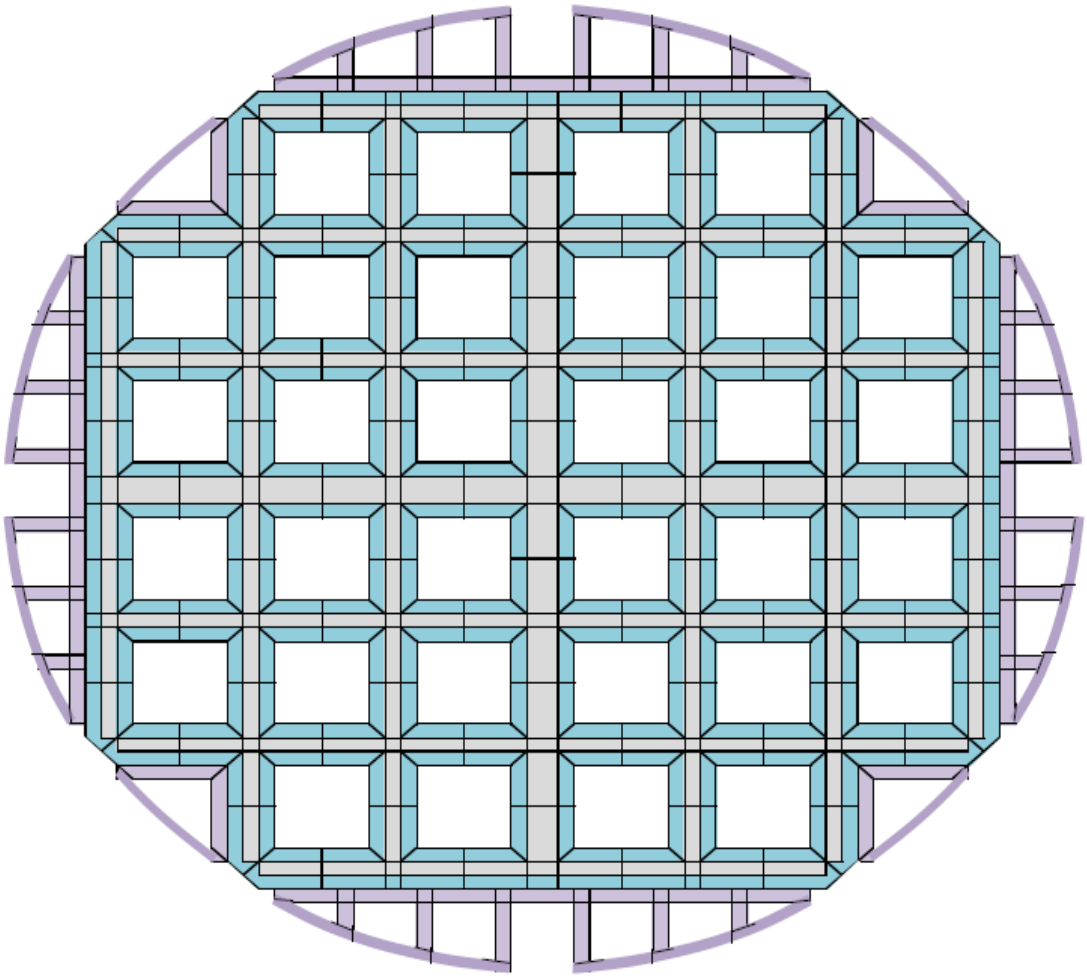


Figure 2.7. Typical COBRA-SFS Model of Canister Basket and Support Rails, with Multiple Layers of Plates (NOTE: Diagram not to scale; node thicknesses greatly exaggerated for clarity)

2.3.1.2 Cask Model Optimization: Solution Parameters

The calculation time required for the numerical solution can be affected by several user-specified parameters. The most important of these are the acceleration parameters for the rod and slab energy equations and the axial rebalancing of the fluid energy solution. The COBRA-SFS numerical solution spends about 80% of the total calculation time in the energy solution, so even a small change in the rate of convergence of this part of the solution can have a significant impact on the overall cost of the simulation.

The acceleration parameters are applied to calculational variables as

$$x^n = \alpha X^n + (1.0 - \alpha) X^{n-1}$$

where

X = calculational variable
α = acceleration factor (input)
n = current iteration number.

The fluid and solid structure energy equations, which are solved simultaneously for all nodes at each axial level, use acceleration factors ACCELH and ACCELW (both defined on CALC.2). The default value for these two acceleration factors is 1.2, and for most problems this is a reasonable value. However, the optimum value can be as high as 1.5 to 1.6, and may decrease the computation time by as much as 20%. For very large cask models, or if a large number of simulations will be run with a given model, it may be wise to run a few sensitivity cases to determine an optimum value for these factors.

The rod energy equation is solved for all nodes at an axial level, with the new fluid enthalpies and slab temperatures held fixed. If the rods are not thermally coupled, either by direct contact conductance or radiative heat transfer, the solution of the rod energy equation requires only one iteration, and the value of the rod solution acceleration factor is irrelevant. If the rods are coupled primarily by contact conductance, the rod energy solution may converge faster if the solution is accelerated by specifying a value for ACCROD that is greater than 1.0. If the rods are coupled by radiative heat transfer, however, it may be necessary to damp the solution because of the highly non-linear terms.

Experience has shown that the value of ACCROD should probably be less than 0.8 for most problems with radiative heat transfer, and it might need to be as low as 0.5 or 0.4 for stability, in some cases. It is recommended that a conservatively low value be selected for ACCROD. The solution computation time does not seem to be readily decreased by adjustments to this parameter, but if it is too large, the solution can become unstable.

Axial rebalancing is done to accelerate the axial propagation of thermal disturbances in the flow field. It is sometimes useful in speeding up convergence, but since axial transport of energy is not generally a dominant heat transfer mechanism in cask thermal-hydraulics, the effect of such acceleration on overall convergence is often very minor. The default value of 1.0 for ACCAX is usually sufficient, and the factor has very little effect on convergence rates for most problems.

2.3.2 Summary of Applications of COBRA-SFS

The basic conservation equations in COBRA-SFS are applicable to any thermal-hydraulic problem that can be adequately described using the assumptions of the subchannel modeling philosophy. The constitutive relations required to achieve closure of the equation set, however, are for the most part far less general, and usually have a limited range of applicability. As a result, the question of whether or not COBRA-SFS can be used to analyze a particular problem is one that should be approached with some careful thought. As a general rule, the code should not be applied without first ascertaining its suitability for the analysis by appropriate data comparisons, assuming such exist somewhere in the real world. If appropriate experimental data is not available, then sensitivity studies and very careful scrutiny of the reasonableness of the results is advisable, based on the underlying physics of the problem.

Cycles 1, 2, and 3 of the COBRA-SFS code have been extensively validated for application to single assembly spent fuel analysis, by means of comparisons with data obtained in electrically heated test sections (e.g., Bates 1986, Fry et al. 1983, Irino et al. 1986). Compared to shipping

casks and storage canisters, these test sections had relatively simple geometries, and as such provide a check on the basic solution method and the formulation of the conservation equations in the code. The code's ability to accurately predict temperature distributions for these tests shows that the conservation equations are in the proper form in the code, and that they correctly model the physical processes of heat transfer and momentum exchange within the fuel rod array and enclosing structures. Documentation of this validation work includes the reports listed in Table 2-1.

Table 2-1. Single-Assembly Validation of the COBRA-SFS Code

ID Number	Date	Title
PNL-6049; Vol. III	1986	COBRA-SFS: A Thermal-Hydraulic Analysis Computer Code; Vol. III: Validation Assessments
EPRI-NP-3764	1984	Thermal-Hydraulic Analysis of Consolidated Spent PWR Fuel Rods
PNL-5781	1986	COBRA-SFS Predictions of Single-Assembly Spent Fuel Heat Transfer Data
EPRI-NP-4593	1986	Comparisons of COBRA-SFS Calculations to Data from Electrically Heated Test Sections Simulating Unconsolidated and Consolidated BWR Spent Fuel

Validation of multi-assembly casks and storage systems is rather more difficult than simulation of single-assembly tests. Due to the greater complexity of the geometry, and the sheer number of nodes that must be included, the possibility of compensating errors is vastly increased. As part of the code validation for multi-assembly applications, the original release of the COBRA-SFS code, and subsequently the Cycle 1 release, were used to obtain pre-test predictions for a number of cask test programs, including the CASTOR-V/21, the TN24P, the Westinghouse MC-10, and the VSC-17 casks.

These calculations were made essentially "blind," in that in many cases the COBRA-SFS models were built and thermal evaluations performed before the tests were conducted. In most cases, discrepancies between the actual test results and the pre-test predictions produced by the code models could be traced to deviations of the experimental design or procedure from the test plan upon which the modeling was initially based. When the code input was modified to make the model more representative of the actual test conditions, the predictions were generally in excellent agreement with the data. Table 2-2 lists the documentation of the multi-assembly validation work performed on COBRA-SFS Cycle 1, and on various subsequent interim developmental versions of the code. Table 2-3 gives a partial listing of systems that Cycle 2 of COBRA-SFS has been used to model, as part of on-going work at PNNL in technical and performance evaluations of spent fuel storage and transportation systems.

Table 2-2. Multi-Assembly Validation of COBRA-SFS Code

ID Number	Date	Title
PNL-6054 EPRI-NP-5128	1986	TN-24P PWR Spent-Fuel Storage Cask: Testing and Analyses
PNL-6631 EPRI-NP-6191	1989	Testing and Analyses of the TN24P PWR Spent Fuel Dry Storage Cask Loaded with Consolidated Fuel
PNL-5777, Vol. II	1986	BWR Spent Fuel Storage Cask Performance Test, Vol. II: Pre-and Post-Test Decay Heat, Heat Transfer, and Shielding Analyses
PNL-5974	1986	CASTOR-1C Spent Fuel Storage Cask Decay Heat, Heat Transfer, and Shielding Analysis
PNL-5802	1986	COBRA-SFS Thermal-Hydraulic Analysis of the CASTOR-1C and REA 2023 BWR Storage Casks Containing Consolidated Spent Fuel
PNL-7839 EPRI-TP-100305	1992	Performance Testing and Analyses of the VSC-17 Ventilated Concrete Cask

Table 2-3 Summary of COBRA-SFS System Models of Spent Fuel Storage and Transportation Packages

System	Model
Holtec	MPC-32, HI-STORM100
Holtec	MPC-24, HI-STORM100
Holtec	MPC-68, HI-STORM100
Holtec	MPC-32, HI-TRAC Transfer Cask
Transnuclear[1]	TN-24PT4
Transnuclear	TN-68
Transnuclear	TN-32PT
Transnuclear	TN-32PT in OS197/OS197L TC
Transnuclear	TN-24PHB in OS197/OS197L TC
Transnuclear	TN-24PTH

[1] Transnuclear Inc. is now AREVA TN Americas; in some references the spent fuel storage and transportation packages listed here as “Transnuclear” are currently identified simply as “AREVA” systems. To avoid confusion, and to be consistent with older references these systems are identified as “Transnuclear” in this document.

System	Model
Transnuclear	TN-24PHB
Transnuclear	TN-32PTH1
Transnuclear	TN-32PTH1 in OS200 TC
Transnuclear	TN-32PT
Transnuclear	TN-24PT1, Advanced NUHOMS
Transnuclear	TN-24PT4, Advanced NUHOMS
Transnuclear	TN-24PT4 in MP197HB Transportation Cask
Transnuclear	TN-32PTH1, HSM-H NUHOMS
Transnuclear	TN-32PTH1, HSM-H NUHOMS
Transnuclear	TN-LC (LWT transportation package)
Transnuclear	TN-69BTH
Transnuclear	TN-32PHB in CCNPP-TC
General Atomics	GA-4 (LWT transportation package)
NAC	MAGNATRAN Transportation Cask (with design basis canister payload)
NAC	MAGNASTOR (with design basis canister in storage overpack)
NAC	NAC-LWT (transportation package)

2.3.3 What to Do When the Code Fails

It may be humbling to have to admit it, but it is usually the user's own fault when the code fails, particularly when the failure occurs in the course of processing the input data. The COBRA-SFS code has seen a great deal of use in a wide variety of applications. Most of the more obvious bugs and errors have been worked out of the code. When it fails in a particular calculation, the problem can usually be traced to something incorrect in the input (e.g., inadvertent errors in the input values, misapplication of particular models, or inappropriate boundary conditions). Therefore, the first step in investigating any problem with the code's performance should be a rigorous scrutiny of the input file.

Examples of code applications are presented in COBRA-SFS Theory Manual Section 7.0 (Richmond et. al. 2021) with the validation test cases, and Section 3.0 of this report provides some guidance in the form of general advice on how to go about setting up and testing a model in a manner that should help uncover input errors or inconsistencies. In addition, the input instructions in Section 2.4 give detailed explanations of the form and usage of each input line in the code. Where necessary, interrelationships between different parts of the input stream are pointed out, and any incompatibilities are noted. There is also some internal checking for errors within the code. But because the input is so extremely flexible, there is a definite limit to how

effective this can be. The code checks for violations of the dimension parameter specifications, and for inconsistencies in the view factor input for radiative heat transfer. Some checking is done on the specification of the channel and slab node interconnections, to make certain that the input conventions have not been violated. But the code has no means of divining what the geometry *should* look like, as compared to how the input describes it.

If careful examination of the input uncovers no obvious errors, incompatible modeling specifications, or inappropriate boundary conditions, the next step is to see how far the solution behavior managed to get before failing. The flow and heat transfer problems solved by the code are generally quite complex, and in many cases highly nonlinear. There are two basic ways the solution can get into trouble: 1) by driving toward a solution far outside the expected reasonable range of the answer, or 2) by being unable to converge on an answer even after many iterations. If there are no input errors, the underlying cause of the failure is similar in both cases; non-linearities in the problem are interfering with the orderly procession of the solution toward convergence. How to deal with the matter, however, is slightly different in each case, as described below.

2.3.3.1 Code Fails with a Cryptic Error Message

The error message is “some variable is not being interpolated properly by subroutine curve.” In most cases, the variable is the fluid enthalpy, and the problem is not with the interpolation algorithm in the code, as the message seems to imply. The interpolation algorithm is probably working fine; the message simply means that the variable has exceeded the range of the properties table for the fluid, as specified in input group PROP (see Section 2.4.2). This message can pop up in the processing of input group OPER (in which case, it is almost certainly a simple input error, either in PROP or OPER). Or it can terminate the calculation after the code has processed all input, and is attempting to run the problem. In either case, the user should first check the group PROP input to determine if the specified range of the table is large enough to accommodate the full range of conditions expected in the problem. If the range seems a little narrow, it might be a good idea to add a line or two to the properties table and try running the case again.

One particularly unusual circumstance in which this error message has manifested is an odd quirk in some modern FORTRAN compilers. Such compilers seem to have a rigid insistence on including a decimal point in the input for the variable storing the fluid temperature in the property table (read on input line PROP.3 in variable TEMLIQ(I,J), see Section 2.4.2). If this data is entered without a decimal point (e.g., instead of “300.” or “300.0,” the value in the input file is specified as simply “300”), these compilers will interpret the value as if it has a format with two decimal places; (i.e., “3.00”, rather than “300.0”). This generally does indeed result in a table with a range inadequate for the particular problem, but the user may not notice the shift in the decimal place in the property table information echoed in the output file with this error message.

The problematic arrays (temperature and usually enthalpy), are displayed in the error message in exponential form, and the user may not immediately notice that the exponent is two orders of magnitude smaller than expected. In examining the property table as defined in the input for group PROP, the user, being merely human, would tend to interpret an entry such as “300” as equivalent to “300.0”, and might not even think of looking more closely at the error message output of the temperature array as the code has interpreted it. However, once discovered, this problem is easily fixed, simply by inserting a decimal point appropriately in the table of temperature values in the input file.

If the output of property values accompanying the error message is consistent with the desired input, such that the range of the table is being properly captured, and the range of the table seems more than adequate for the particular problem, there is one other circumstance where this error might be encountered. If the problem occurs on the very first iteration, there is only a slight possibility that increasing the range of the fluid properties table will take care of matters and let the iteration proceed to a proper solution. For a problem with a specified pressure drop boundary condition (see input group OPER), this error message on the first iteration might simply mean that the initial conditions (specified on line OPER.2) are a very bad first guess. If the value specified in GIN on OPER.2 for the flow rate is significantly less than the final value, unrealistically high fluid temperatures may be calculated in the first iteration. In such a case, it may help to increase the specified flow rate to a value closer to the expected final value.

If, however, the problem occurs after a number of iterations, the trouble is probably more serious, and may indicate an instability in the solution of either the conductive or radiative heat transfer energy exchange. The thermal connections between different slab nodes, and between the slab nodes and the fluid, are implicit within a given axial level. They are connected only semi-implicitly for radiative heat transfer within a given axial level, and for conduction to the axial levels above and below a given level. The heat transfer solution for a case in which slab nodes have strong semi-implicit thermal connections, in comparison to their implicit connections, may experience an instability.

This instability can be diagnosed by running the case again with the maximum number of iterations set to one less than the value at which the code printed the error message. The resulting output for a slab node that is experiencing this instability will show an unexpected (and probably unrealistic) “saw-tooth” temperature profile in the axial direction. The problem can usually be dealt with by specifying some degree of damping for the solution of the slab node or nodes that exhibit the saw-tooth pattern. This is accomplished by specifying a slab damping factor with the variable SLDAMP on CALC.2, then identifying the slab node(s) to be damped, using the input on lines CALC.5 and CALC.6.

The error message may also appear after the solution has proceeded through some number of iterations if there are improperly specified radiative or conductive thermal connections between nodes in the problem. This sort of error has the net effect of creating or destroying energy in the model, so that in some node the enthalpy or temperature may be increasing or decreasing in a physically unrealistic manner with every iteration. The problem is due to an input error of some kind, but it may be rather difficult to find the node that is in error for a case with a large number of nodes and complex interconnections. One relatively simple way to check for this possibility is to perform an isothermal test case. (See Section 3.2 for further discussion of model verification steps.) This will tend to reveal any unintended energy sinks or sources in the model, with localized temperatures that increase or decrease in a manner inconsistent with the laws of thermodynamics.

To run an isothermal case, the power generation rate is set to zero and all boundary temperatures are set to some uniform value. If all thermal connections have been specified properly, the temperature of all fluid, rod and slab nodes should be at very nearly the exact boundary value. If they are not, the region with the largest discrepancy should be examined to determine the source of the problem. If an isothermal case is not sufficient to uncover the problem, selective removal of different radiative heat transfer groups, slab connections, or plenum connections may also be of help in isolating the problem.

2.3.3.2 Code Fails to Converge within the Specified Number of Iterations

The first and most obvious question to ask is how many iterations should be sufficient to achieve convergence. The default of 20 iterations in the code is a hold-over from the halcyon days of reactor core thermal-hydraulic analysis, in which it was not uncommon for the code to be able to converge in less than a dozen iterations, even for very complex problems, due to the driving force of high flow rates and steep thermal gradients. The applications to cask analysis generally involve much more challenging problems, often with very low (or recirculating) flow rates, and at least some regions of relatively flat thermal gradients. It is therefore not unusual for several hundreds, even several thousands of iterations to be required to achieve convergence. For system designs with relatively direct and unencumbered heat transfer paths from the fuel to the outer boundary, and relatively strong flow recirculation patterns, the code can show relatively rapid convergence. In some cases, fewer than 1000 iterations may be required. For such systems, more than a thousand iterations is probably a sign that the code is working too hard, and something probably should be done to find out why the model is behaving that way, and the cause evaluated to determine if some improvement could be made with model revisions.

Many spent fuel storage and transportation systems have design features that include significant bottlenecks in the heat transfer paths, or large non-uniformities in the flow dynamics and heat transfer gradients in different parts of the system. Some systems have virtually no significant flow of fluid, and consequently virtually no convection heat transfer, while others rely on high rates of convection in some parts of the system, and at the same time might have virtually no convection in other parts of the system. These features can result in convergence issues that the user may simply have to learn to live with, depending on the purpose of a given modeling application. A case in point is in the modeling of vertical ventilated storage systems, which typically have two separated flow regions; the inert gas within the canister (which may be undergoing a thermo-siphon recirculation) and the external air flowing up through the annulus between the canister and overpack. It may require more than 1000 iterations, or even several thousand iterations, to obtain a converged solution that encompasses both flow fields, since the rebalancing schemes in the COBRA-SFS numerical solution cannot efficiently accelerate the overall solution with two such separated flow regions. Another system design that can present convergence difficulties consists of canisters with layers of metal discs, interleaved with helium gas layers, such as the horizontal 24P-type Dry Storage Canisters (DSCs) in some NUHOMS modules, and NAC's vertical storage canisters in the NAC-UMS.

In cases where the number of iterations seems excessive, the user should examine carefully the iteration summary produced at the head of the output of results. An example of the code output for the iteration summary is shown here for the first two iterations for a calculation that took only 214 iterations to converge. This summary lists the main solution errors against which the convergence criteria are tested, in the following format.

		peak clad						error			
iteration no.	sweep no.	temp (F)	level	rod	assm.	total flow (lbm/s)	pressure drop (psi)	total energy	flow	fluid energy	rod energy
1	1	381.9	12	13	4	0.338E+01	-0.2353495	-2.3474	56.2054	0.0035	0.0002
	2	382.6	11	13	4			-2.0857	56.2054	0.0233	0.0001
2	1	449.6	9	13	4	0.253E+01	0.0054958	-1.0128	3.0014	0.0780	0.0038
	2	474.4	9	13	4			-0.8751	3.0014	0.1611	0.0007

Figure 2.8 Standard Output Format

In examining the convergence behavior of a given case, the significant items of interest in the iteration summary are the last four columns, summarizing four error terms that are tested

against the convergence criteria specified by the user in the input for group CALC, on line CALC.2. Each iteration includes an upward axial sweep from inlet to exit (sweep no. 1), followed by a downward axial sweep from exit to inlet (sweep no. 2). For each axial sweep, information on peak clad temperature, total flow, pressure drop, and error residuals are written to the summary, as illustrated in the example.

The total energy error is tested against the user-specified convergence criterion (input variable QERROR on CALC.2). The value calculated for testing against the input value of QERROR represents the maximum percentage change in total energy error in all computational cells in the model. The flow error, which represents the maximum percentage change in the flow in a fluid subchannel node, is tested against the input variable FERROR. Similarly, the fluid energy error, which represents the maximum percentage change in the flow energy in a fluid node, is tested against the input variable HERROR. The fluid continuity solution is also tested against the input variable WERRY during the inner iteration, but no information is printed on this in the iteration summary (see Section 2.4.13, input group CALC, line CALC.2 for a complete discussion of this input).

In most cases, these error terms should show a monotonically decreasing trend in the course of the solution iterating toward convergence. In some cases, the errors may show some tendency to oscillate over the first few iterations, but this oscillation should rapidly damp to a steadily decreasing value. If the oscillation persists, the behavior may be due to the stability problem described above. Strong semi-implicit coupling along with relatively weak implicit coupling between nodes for the heat transfer solution may not be severe enough in some cases to cause the code to actually fail. It can, however, result in very slow rates of convergence, or oscillation of the solution about an asymptote that may or may not be approaching the correct solution. In such a case, the problem can be diagnosed in the same way as described above, and much the same action needs to be taken to fix the problem. That is, by looking at the output for saw-tooth patterns in the axial temperature profiles, and damping any nodes that exhibit such behavior.

If the total energy error is not oscillating, but appears to be stuck on some value that does not change from iteration to iteration, the problem may be improperly specified thermal connections. This sort of input error can cause the code to fail, as noted above in Section 2.3.3.1, if the resulting energy error is large enough. Alternatively, it may simply create a small source of energy error that cannot be iterated away. The problem can usually be diagnosed using the same procedures as outlined above, starting with running an isothermal case to look for the existence of an energy source or sink somewhere in the problem.

If the error summary shows steadily decreasing values, but they are going down at an irritatingly slow rate, there are a number of possible causes of the problem and different actions to be taken. Some of the more likely possibilities are discussed below, with suggestions for appropriate action.

If the problem is starting with an initial guess that is too far from the final solution, it may take an abnormally long time to reach convergence. In such a case, the best thing to do is to start with a better initial guess, as defined with the operating conditions in group OPER on line OPER.2. If the case uses a uniform pressure drop boundary condition for a specified total flow (with or without the plenum model), it may be helpful to start with an initial specified flow rate (entered as GIN on OPER.2) that is closer to the expected final value. It may also be helpful to adjust the initial enthalpy (entered as HIN on OPER.2).

Another possible cause of a slow convergence rate is that the specified damping factors on CALC.2 are unnecessarily stringent for a particular case. This difficulty can be reduced or eliminated by appropriate changes in the input for CALC.2. If the damping is not needed, the code will almost always converge more rapidly without the damping factors. However, removing the damping factors could in some cases cause the code to fail due to instabilities, as described above. As disastrous as this result might seem to the user, it is actually a useful improvement over very slow convergence rates. If the failure is due to instabilities of the type described above, the user will be motivated to search out the specific nodes that are hindering convergence. Appropriate damping factors can then be applied specifically to the troublesome nodes, using the damping factor input variable SLDAMP on CALC.2 and identifying the specific nodes to be damped on CALC.5 and CALC.6. This allows damping to be applied where it is needed, without unnecessarily impacting the global convergence rate for the problem.

The optional axial energy rebalancing can sometimes result in a dramatic improvement in the convergence rate. This feature is specified by setting the variable ACCAX to a non-zero value on CALC.2. Axial energy rebalancing consists of performing a one-dimensional energy solution for the system every three iterations, and using the results to make a global adjustment in all nodes at all axial levels. In many cases this will propagate changes through the system much more rapidly than the node-by-node solution, and can result in much faster convergence. It can, however, destabilize the solution, and so should not be used unless it can be shown to be a definite help. The results of the axial rebalancing calculation can be seen directly by setting the variable IPREP on OUTP.1 in group OUTP. However, this yields a very large amount of output even for problems of moderate size, and probably would not be useful except for debugging purposes.

One last option to consider if the code fails to converge is to examine the values selected for the convergence criteria. As with overly stringent damping factors, if the actual convergence criteria are unnecessarily stringent, this input could be forcing the code to calculate many iterations with no real change in the solution. A quick and easy check to determine if the code is doing more iterations than it really needs is to look at the peak clad temperature in the iteration summary. If it is not changing between iterations, this may be an indication that no substantial changes are occurring in the solution. In such a case, the additional iterations are probably a waste of time, and it would be appropriate to relax the convergence criteria slightly. However, it might also be useful to look at overall temperature gradients in the problem, to evaluate the effect of less stringent convergence values on the temperature distribution throughout the model. This would be particularly advisable if the purpose of the calculation is to determine peak temperatures on components in addition to the fuel cladding.

The appropriateness of relaxed convergence criteria can be evaluated by comparing the results obtained with different values for the convergence criteria. If more relaxed convergence criteria yield essentially the same results as the same case with tighter convergence criteria, and does so in fewer iterations, then the relaxed convergence criteria are probably appropriate for the problem. Some degree of judgement is required from the user to determine what constitutes the appropriate standard of convergence for a given application. In general, it is sufficient for a calculated answer to be approximately as accurate as the measured value with which it is validated.

2.3.3.3 Code Fails Due to Divide-by-Zero in the Solution Process

Curiously enough, with many modern compilers, the code does not actually fail when the solution commits a divide-by-zero. Execution blithely continues, writing meaningless “NaN”

values in place of numerical results to the output file, sometimes for thousands of iterations, depending on the value specified by user input on CALC.3 for input variable NTRIES, defining the maximum number of iterations for a calculation. This can be extremely disheartening to the user when opening the output file of a supposedly completed calculation, particularly for a case that has been running for many hours. It is therefore advised, at least in the model debugging phase of an application, for the user to peek at the output file, once the code is no longer stopping with error messages or other obvious difficulties and appears to be finally running the desired calculation.

If the code actually is running as it should, the output file will have begun to accumulate the iteration summary, and usually will have completed two, three, ten, or even more iterations in only a few minutes of execution. The numerical values of the convergence errors may be rather large, at such an early stage in the solution process, but the values will be real numbers, and should be showing at least some tendency to decrease with increasing iteration number, particularly if the solution has proceeded far enough to display more than half-a-dozen or so iterations in the output file. But if the solution has hit a divide-by-zero, it will be writing the letters “NaN” in place of the expected real numbers of the iteration summary. There is no means for the code to recover from this disaster, and the user is advised to abort execution, and begin hunting for the cause of the behavior.

If the very first line of the iteration summary consists of reasonable real numbers, the problem could be due simply to overly constrained convergence criteria. A simple test for this possibility is to replace the convergence criteria input (CALC.2) with the recommended default values (see Section 2.4.13). This almost never works, but it is a quick and easy test, and therefore worth a try before going on to more strenuous efforts to diagnose and correct the problem.

If the “NaN” output occurs in the very first line of the iteration summary, (that is, the upsweep of iteration #1), with NaN printed for the initial value of the peak rod temperature, as well the convergence error summary, the cause of the problem is almost certainly an input error. It will, unfortunately, be a rather subtle error, to have escaped detection by the input error checking gauntlet that the model input has already been through to reach the point of actually beginning to try to execute the case. If the user has access to a FORTRAN debugger, and knows how to use it, this may be the simplest way to track down the problem. It should be possible to determine precisely the point in the model where the divide-by-zero initially occurred, identifying the exact rod, solid node, or fluid node, and axial location. Examination of the input for that node, or input that affects that node, will generally reveal the error, once the user knows where to look.

If a useful debugger is not available, the only other practical option for the user is to start stripping features from the model, for the purpose of isolating the modeling feature (or features) causing the problem. Try an isothermal case (i.e., remove heat generation, set a uniform temperature at all boundaries of the system). Try stripping out thermal radiation (i.e., remove group RADG; and maybe turn down the heat a bit, if thermal radiation is an important component of heat transfer in this case). If the model includes the option for upper and/or lower plenum connections (as per group BDRY), remove the plena entirely, substituting flow or pressure drop boundary conditions for assemblies that would be connected to the plena.

If all else fails, call for help. The telephone numbers of the code developers who are still using the code on a regular basis (as of this writing), are printed on the banner page of the output file.

These individuals have, in all probability, encountered whatever error is causing the problem in a particular case, and may recognize the symptoms on sight. If nothing else, they will be sympathetic, and as helpful as possible.

2.4 Input Instructions for COBRA-SFS Classic Input

The input instructions for COBRA-SFS are provided in this section. The code input is listed line-by-line, with the definition of each input variable. Each set of instructions is preceded by an explanatory paragraph. However, users unfamiliar with COBRA-SFS should refer to Section 2.3 for a discussion of the modeling capabilities and limitations of the code, and the list of references documenting the various applications and validation calculations performed with the code. These reports are intended in part to provide guidance and examples for the code user in developing new input files and applications. It may also be helpful to look through COBRA-SFS – Theory Manual Sections 7.0 (Richmond et. al. 2021) and Section 3.0 of this report, to maintain focus in the face of the complexity and flexibility of the code input structure.

COBRA-SFS input is organized into groups identified by unique four-character flags. The groups are defined by function and reflect a generalized logical approach to setting up input for a problem. COBRA-SFS input falls into the following five basic categories:

- the physical properties of the solid materials and working fluid
- the flow channel geometry and solid node structure
- the constitutive models for the flow and heat transfer solutions
- the boundary conditions
- solution control parameters and output options.

The various input groups under these five categories are summarized in Table 2-4. Some of the input groups are described in this table as optional. However, in the context of constructing models for spent fuel storage and transportation systems, “optional” does not in general mean that a group can actually be omitted from a complete model. It merely means that the code would run without this input, but would produce results that did not take into account the effect of the modeling features specified in that particular input group. The main use of the “optional” feature is that it can be a useful aid in debugging and verifying accuracy of input when building a model. For example, omitting the group RODS input or group RADG input can be a useful step in verifying the accuracy and completeness of the geometry input and slab connections. (See discussion in Section 3.2.)

Table 2-4. Summary of Input Groups

Input Group Flag	Input Group Purpose	Location of Group Line-by-Line Input Instructions
PROP	specifies fluid and solid material properties; fluid properties required for all cases, material properties needed only if model includes solid conduction nodes (i.e., group SLAB).	Section 2.4.2
Geometry Description--		
CHAN	specifies flow field geometry; required for all cases.	Section 2.4.3

Input Group Flag	Input Group Purpose	Location of Group Line-by-Line Input Instructions
VARY	optional input group, to specify axial variation in channel flow area, and axial variation in width of flow path for lateral flow (see COBRA-SFS – Theory Manual 2.3 (Richmond et. al. 2021) for illustration of area definitions for fluid control volume in axial and lateral directions).	Section 2.4.4
RODS	specifies fuel rod geometry and thermal connections between fluid channels and rods; optional, and can be omitted if system being modeled contains no heat generating fuel rods.	Section 2.4.5
SLAB	specifies slab node geometry, defines slab thermal connection types, and interconnecting “map” of solid conduction nodes; optional, and can be omitted if there is no significant heat transfer in solid structures for a particular model.	Section 2.4.6
VIEW	specifies input to define thermal radiation exchange factors for rod arrays and slab nodes and is utilized in scenarios where water rods or guide tubes possessing large outer diameters are important to resolution; optional, and can be omitted if there is no significant energy exchange due to thermal radiation in the system, or if this energy exchange is being temporarily ignored for model debugging purposes.	Section 2.4.7
RADG	specifies input to define thermal radiation exchange factors for rod arrays and slab nodes; optional, and can be omitted if there is no significant energy exchange due to thermal radiation in the system, or if this energy exchange is being temporarily ignored for model debugging purposes.	Section 2.4.8
Constitutive Models--		
HEAT	specifies input for Nusselt number correlations for the total system energy solution (fluid, rods, and slabs; see COBRA-SFS – Theory Manual Sections 2.3, 2.4 and Section 3.0 (Richmond et. al. 2021) for details on how these correlations are used in the conservation equations); this input is not optional; COBRA-SFS has no “hard-wired” heat transfer correlation options; the user must specify the appropriate correlation(s) for the particular modeling application.	Section 2.4.9
DRAG	specifies input for wall friction and form drag correlations for fluid energy solution (see COBRA-SFS – Theory Manual Section 3.0 (Richmond et. al. 2021) for details on how these correlations are used in the conservation equations); this input is not optional; COBRA-SFS has no “hard-wired” constitutive relations for momentum; the user must specify the appropriate correlation(s) for the particular modeling application.	Section 2.4.10
Boundary Conditions--		
BDRY	specifies thermal boundary conditions for the radial side(s) of the model, and defines the geometry and boundary conditions for the optional upper and lower plenum models; this group is optional, if side boundaries are assumed adiabatic and the model considers only inlet flow and exit pressure to define system behavior.	Section 2.4.11

Input Group Flag	Input Group Purpose	Location of Group Line-by-Line Input Instructions
OPER	specifies overall flow or pressure drop axial boundary conditions, and heat generation rate in fuel rods; also can be used to define (optional) forcing functions on boundary conditions for transient calculations. (NOTE: Group REST is also available as a new feature in Cycle 4, to allow a specialized option for defining assembly decay heat values, for interfacing with templates developed for UNF-ST&DARDS.)	Section 2.4.12
Code Control--		
CALC	specifies calculational parameters for numerical solution, including damping options and convergence criteria; specifies time step size and overall duration for (optional) transient calculations.	Section 2.4.13
OUTP	specifies user-defined parameters governing output options for channels, rods, slab nodes, fluid flow gaps; specifies time interval(s) for standard output and other special output options in transient calculations.	Section 2.4.14
ENDD	final group flag to terminate reading of COBRA-SFS input file; signals the code to stop looking for additional input groups, and get on with the job of trying to run the case.	Section 2.4.1 and Section 2.4.15

The new user would be well advised to read through the input instructions entirely before attempting to set up the input for a particular model. It is important for the user to have a reasonably complete picture of the overall structure of the COBRA-SFS input, as it will give order to the large array of options available.

The general format of the input instructions is to give a complete description of all variables in each line of input, including the format for reading the data. Some input lines are repeated, and some groups of lines are repeated in sequence as a set. These repetitive patterns are noted in the instructions, both in the format for the line and in the descriptive text accompanying the input. In many instances, later input will depend on values specified in earlier input lines. These flags are noted on the input line, with a reference to the line on which the flag was defined. For example, the variable defining the number of channels in an assembly, NCHANA, is read on input line CHAN.5. In all subsequent instructions that refer to NCHANA, the origin of this flag is denoted by specifying the variable as NCHANA[CHAN.5].

2.4.1 Problem Initiation Input

Input records COBRA.1 and COBRA.4 are required at the beginning of every COBRA-SFS input file. COBRA.1 defines the execution time allotted for the problem, and offers the user the opportunity to restart a previously executed problem from the tape8 restart dump. The case title is specified on COBRA.4. This title is printed on the output file, and serves as the case identifier, for easy reference. The date and time of the calculation are also written to the output file along with the title. If the restart option is flagged by entering a negative value for MAXT on input line COBRA.1, input line COBRA.2 must also be read, to define the restarted calculation.

COBRA.1 MAXT,IECHO
FORMAT(2I5)

Columns	Variable	Description
1-5	MAXT	CPU time limit (sec.) for problem execution. For restart cases, enter MAXT as a negative number (ABS(MAXT) will be used to define the time limit). (NOTE: The value entered for MAXT cannot supersede any execution time limit specified in the job control language in a batch submittal. The user must ascertain that any such time limits are compatible with the input value of MAXT.)
6-10	IECHO	Flag for printout of input file; = 0; the input file will be printed on the output file (default). = 1; the input file will not be printed on the output file.

COBRA.2 NJUMP,NA,IT,NTT,TTT

Read only if MAXT < 0 on COBRA.1
FORMAT(4I5,F5.0)

Columns	Variable	Description
1-5	NJUMP	Restart option flag; = 1; continue calculation from a previous steady state or transient solution; new input may be read in to define a new case, with the previous calculation as the first guess. = 2; does NOT continue a calculation; this option merely allows the user to capture output information from a previous calculation that is stored on tape8 (which is a binary file, and is not readily readable without appropriate and detailed knowledge of how it was written).
6-10	NA	Number of iterations for restarted case; • for a steady-state calculation, number of additional iterations. • for a transient calculation, number of iterations per time step. (NOTE: If new input is read for group CALC in the input for the restart case, the value for NA specified here will be over-written by the new value for NTRIES on CALC.3.)
11-15	IT	Flag to define type of restart: = 0; restart and continue the previous solution; either • continue iterating in a steady-state case, or • continue time steps in a transient case = 1; restart a converged steady-state case and initiate a transient. (NOTE: If restarting from a steady-state solution, new input in group CALC is required to define the transient time and time-step size. In addition, new input to define the transient forcing functions may be needed, in group OPER. However, the user may have already defined any needed forcing functions in group OPER, within the original steady-state input file, since the transient forcing function input is normalized to the time-zero (i.e., initial steady-state) input. In such case, no additional forcing function input is needed for group OPER.)
16-20	NTT	Number of additional transient time steps for the restarted calculation.
11-15	TTT	Total transient time (seconds) for the restarted calculation.

All cases, including a restart case, must have a title, which is read on line COBRA.3. This is an alphanumeric identifier that will be printed on each page of the output file, along with the date and time of the run.

COBRA.3 NUMLINES,J1,(CASENAMES(I),I = 1,NUMLINES)
FORMAT(5X,I5,A64/10X,A64)

Columns	Variable	Description
3-5	NUMLINES	Number of *additional* lines of text to read after the first line, which is the case title that will be printed on each line of the output file. Enter zero (0) if no additional text lines are needed for the case description.
6-10	J1	Flag for option controlling output of input data; ≤1; print all processed input data from the current input file, then continue with solution. = 10; print all processed input data from the current input file, then stop execution.
11-74	CASENAMES(I)	First line is case title to be written out at top of each page in the output file (maximum of 64 characters). If NUMLINES > 0, (NUMLINES-1) subsequent input lines must be provided in the input file. Note that content of such lines is at the user's discretion; typically, these lines contain model identification or description data, for information and case tracking purposes. This information is not used by the COBRA-SFS code.

When doing a restart calculation, (i.e., NJUMP=1), the user has the opportunity to modify the input by specifying new data for the input groups PROP, VARY, RADG, HEAT, DRAG, BDRY, OPER, and OUTP. The model geometry, however, which is specified in groups CHAN, RODS, and SLAB, generally should not be changed on a restart. Changes in these groups generally constitute a whole new model, and a restart is neither useful nor appropriate in most cases. In some transient applications, changes in thermal connections in group SLAB may be needed, or in some cases changes in boundary conditions may be defined on a restart, but such input changes should be approached with caution, even for an experienced user.

After all modified groups have been entered, or if no changes or modifications are needed in the input on a restart, the user must enter the ENDD group flag to terminate the search for new input for the restart.

COBRA.4 AGROUP

FORMAT(A4)

Columns	Variable	Description
1-4	AGROUP	Enter group flag for input to be read (or redefined on restart; NJUMP > 1 on COBRA.2). See first line of each input group, in Sections 2.4.2 through 2.4.14, for complete first-line input instructions for each group. Enter ENDD, after all needed input groups have been defined, to signal end of input stream for this case. (NOTE: This applies to a normal stand-alone case, or a restarted case.)

2.4.2 Group PROP–Fluid and Material Properties

This group is used to define thermodynamic and thermal properties for the fluid and solid materials included in the model. Properties can be specified for only two different fluids, but any number of solid material properties can be specified in a model, within the limitations of the specified array dimension parameters (see Section 2.3.3). Properties for fluids are defined on PROP.2 and PROP.3; solid material properties are defined on PROP.4. At least one fluid must be specified to define the coolant, but properties for solid materials are required only if the model includes solid conduction nodes (see Section 2.4.6, group SLAB input).

PROP.1 AGROUP,NPROP,NSPROP
FORMAT(A4,1X,2I5)

Columns	Variable	Description
1-4	AGROUP	Enter PROP.
6-10	NPROP	Number of elements in fluid properties table. Enter 0 to use option for two sets of fluid properties. (The number of elements for each table is specified on PROP.2.)
11-15	NSPROP	Number of solid materials for which properties will be entered on PROP.3.

Fluid Properties

If the coolant is a liquid, properties are entered as saturated liquid values for the pressure PLIQ(I). (NOTE that this version of COBRA-SFS is limited to single-phase fluids only, liquid or gas. Phase change models for water as the coolant, which are a feature of many other versions in the COBRA family of codes, are not implemented in this version of COBRA-SFS.) If the coolant is a gas, properties are entered for temperatures at the system pressure (which is specified on OPER.2), and variable PLIQ(I) is not used. Fluid properties for at least two reference temperatures are required to allow interpolation.

Two tables of fluid properties can be specified with this input group, to represent different fluid properties in separate regions of a given model. Typically, this option is used when modeling a ventilated storage module, in which the sealed canister containing the fuel assemblies is backfilled with helium, and there is an air flow annulus between the canister and the storage module overpack, to provide natural convection cooling.

SPECIAL NOTE: The convention in the code for modeling different fluids in a ventilated module is to assume that the fluid properties in the first table defined in the input for PROP.3 represents the air in the annulus, and the second table represents the fluid properties of the inert backfill gas within the canister, typically helium. The fluid regions containing inert gas are connected to the upper plenum and lower plenum, in this type of model. All fluid regions connected to the plena must have the same fluid, as defined in ***second table*** of the PROP.3 input. This convention must be adhered to in assigning fluid properties types to specific assemblies, in group CHAN (see Section 3.4.3).

When the option for two tables of fluid properties is used, NPROP[PROP.1] is set equal to zero, and the number of elements for each of the two tables is defined on PROP.2. If NPROP[PROP.1] is specified as zero, PROP.2 and PROP.3 are read twice in sequence to specify two sets of fluid properties.

If NPROP[PROP.1] is greater than zero, only one fluid properties table will be supplied, and PROP.2 is not read. The value of NPROP from PROP.1 is automatically assigned to NFPROP(1), and only one fluid property table is read on PROP.3.

PROP.2 NFPROP(J),FNAME(J)
FORMAT(I5)

Read only if NPROP[PROP.1] = 0

Columns	Variable	Description
1-5	NFPROP(J)	Number of fluid property table elements to be read on PROP.3 for the J^{th} table.
6-10		BLANK
11-20	FNAME(J)	Alphanumeric identifier for the J^{th} fluid properties table.

PROP.3 (PLIQ(I,J),TEMLIQ(I,J),HLIQ(I,J),CONLIQ(I,J),CPLIQ(I,J),
VLIQ(I,J),VISLIQ(I,J),BLIQ(I,J),I = 1,NFPROP(J))
FORMAT(8E10.5)

Columns	Variable	Description
1-10	PLIQ(I,J)	Saturation pressure for liquid coolant (psia). (NOTE: PLIQ(I) is not used when coolant is a gas.)
11-20	TEMLIQ(I,J)	Temperature (°F).
21-30	HLIQ(I,J)	Enthalpy (Btu/lbm).
31-40	CONLIQ(I,J)	Thermal conductivity (Btu/h-ft-°F).
41-50	CPLIQ(I,J)	Specific heat (Btu/lbm-°F).
51-60	VLIQ(I,J)	Specific volume (ft3/lbm).
61-70	VISLIQ(I,J)	Viscosity (lbm/ft-h).
71-80	BLIQ(I,J)	Coefficient of Thermal expansion (/°F).

If NPROP[PROP.1] > 0, PROP.3 is read NPROP times.

***If NPROP[PROP.1] = 0, PROP.2 and PROP.3 are read twice in sequence ***

***with PROP.3 read NFPROP(J)[PROP.2] times for the J^{th} table ***

Solid Material Properties

Solid material properties are specified for NSPROP[PROP.1] materials. This input is read only if NSPROP on PROP.1 is greater than zero. Specific heat and density of solid nodes are assumed constant for all temperatures. Thermal conductivity can be specified as a function of temperature, as a polynomial fit up to third order, as

$$k = CON0(I) + CON1(I)*T + CON2(I)*T^2 + CON3(I)*T^3$$

where T is absolute[1] temperature, in °R.

For constant thermal conductivity, only CON0(I) is entered, and the other three leading coefficients of the polynomial are left blank, or entered as zero (0.0).

SPECIAL NOTE: When using temperature-dependent thermal conductivity, the user is advised to verify that the range of temperature used to determine the coefficients of the polynomial fit is appropriate for the range of temperatures likely to be seen by the slab nodes represented with that material type. When extrapolated outside the range of the independent variable used to obtain the fitted coefficients, higher order polynomial equations can take unusual and quite unrealistic turns. This can destabilize the numerical solution, causing failure to converge. Or worse, drive convergence to a physically incorrect

[1] Special courtesy reminder for users more familiar with SI, and unfamiliar with the older conventions of Engineering Units used in the COBRA-SFS code; the absolute temperature in the Fahrenheit scale is termed °R for Rankine, with the conversion °R ≡ °F + 459.67.

solution, due to unrealistic values of local thermal conductivity, without any obvious warning to the user.

The material type identification number is used to specify the material properties of the various solid structure nodes input in groups SLAB and BDRY. Material property types must be numbered sequentially from 1 to NSPROP.

PROP.4 (IMAT(I),ANAME(I),CPSOL(I),RHOSOL(I),CONO(I),CON1(I),CON2(I),CON3 (I), NQVT(I),I= 1,NSPROP[PROP.1])

Read only if NSPROP > 0 on PROP.1
FORMAT(I5,A5,6E10.0,I5)

Columns	Variable	Description
1-5	IMAT(I)	Solid material type identification number.
6-10	ANAME(I)	Solid material name (for user convenience; this information is not used in the code except for output of input data).
11-20	CPSOL(l)	Specific heat (Btu/lbm-°F) (needed for transient calculations).
21-30	RHOSOL(I)	Density (lbm/ft^3) (needed for transient calculations).
31-40	CON0(I)	Coefficients of a polynomial expression (up to third order) for solid thermal conductivity (Btu/h-ft-°F). SPECIAL NOTES: A constant thermal conductivity is obtained by entering a value only for CON0(I), and leaving the other three coefficients blank or 0.0. A simple linear fit can be obtained by specifying values only for the first two coefficients, CON0(I) and CON1(I), with CON2(I) and CON3(I) blank or 0.0. A 2nd order fit can be used, particularly if it is more well-behaved at the ‘edges’ of the temperature range, by specifying values only for the first three coefficients, CON0(I), CON1(I), and CON2(I), with CON3(I) blank or 0.0.
41-50	CON1(I)	
51-60	CON2(I)	
61-70	CON3(I)	
71-75	NQVT(I)	Identification number for transient heating rate forcing function for nodes with solid material type IMAT(I). (See related input on OPER.14 and OPER.15.)

PROP.4 is read NSPROP[PROP.1] times.

2.4.3 Group CHAN–Flow Field Geometry

This group is read to define the flow field geometry. The basis of the geometry is subchannel modeling (as described in detail in COBRA-SFS – Theory Manual Sections 1.0 and 2.0 (Richmond et. al. 2021)), in which the flow field is represented as an array of channels that can communicate laterally by crossflow. For convenience in modeling fuel bundles and casks, the channels are grouped into assemblies, which consist of a group of channels that can communicate laterally within an enclosure. Channels of one assembly cannot communicate by crossflow with the channels of other assemblies. Regions of fluid flow that do not contain fuel assemblies, such as open regions in the basket support structures of a canister, are also represented as “assemblies” in a COBRA-SFS model. These are sometimes referred to as “unrodded assemblies,” which appears to be something of an oxymoron, but is merely an artifact of generalizing the modeling feature of an “assembly” to account for other regions of fluid flow in a given model. See COBRA-SFS – Theory Manual Section 7.0 (Richmond et. al. 2021) for illustrations of cask models that contain multiple assemblies, many of which contain fuel rod arrays, but that also have a significant number of “unrodded assemblies” representing open regions of the basket where fluid can recirculate through the system.

Axial Geometry

The axial geometry of the channel region is defined by the axial length, Z, entered on CHAN.3, and the number of axial nodes, NDX, entered on CHAN.1. All channels in all assemblies have the same axial length, and the same axial noding. For a typical COBRA-SFS model of a spent fuel storage canister, Z is generally defined by the axial length of the basket.

The default in the code is a uniform axial node size, which is calculated as DX = Z/NDX. The user has the option of specifying axial nodes of varying sizes, by setting NAZONE on CHAN.1 to an appropriate value and reading in a variable axial node size table on CHAN.4. However, the use of the variable axial noding option can make axial noding sensitivity studies problematic, and therefore this option is not recommended unless needed to capture an unusual global variation in axial geometry (e.g., canisters with a number of axially spaced discs supporting an array of fuel tubes or assembly guide sleeves, rather than a basket composed of an axial grid of support plates).

CHAN.1 AGROUP, NASSEM, NDX, NAZONE, ISHEAR, NSHEAR, NANGLT
FORMAT(A4,1X,6I5)

Columns	Variable	Description
1-4	AGROUP	Enter CHAN.
6-10	NASSEM	Number of assemblies.
11-15	NDX	Number of axial nodes.
16-20	NAZONE	Flag for variable axial node sizes; = 0; uniform axial nodes (default). > 0; number of regions in variable axial node size table (read on chan.4).
21-25	ISHEAR	Flag for fluid shear stress

Columns	Variable	Description
		(NOTE: Used only when modeling plenum regions with multiple channels); = 0; fluid shear stress not considered. = 1; fluid shear stress included.
26-30	NSHEAR	Number of pairs of gaps that are connected by fluid-fluid shear in the lateral direction. (NOTE: Used only when ISHEAR = 1.)
31-35	NANGLT	Number of lateral control volume orientation angles (read on CHAN.2). (NOTE: This input is needed for the fluid-fluid shear option, but it is also used to define the direction of the gravity vector for lateral momentum transport in horizontal or tilted geometries.)

CHAN.2 (ANGLE(I),I = l,NANGLT(CHAN.1))

Read only if NANGLT[CHAN.1] > 0
FORMAT(8e10.5)

Columns	Variable	Description
1-10	ANGLE(I)	Orientation angle for lateral flow, in degrees from vertical; to be assigned to specific gaps on CHAN.9.

NOTE: This input is used only if fluid-fluid shear is included in the model (i.e., ISHEAR= 1 on CHAN.1), or if modeling lateral gravity terms in a horizontal or tilted geometry.

SPECIAL NOTE: Variable ANGLE is defined with a limit of 5 entries in the code common blocks. This in effect limits the value of NANGLT to no more than 5. Technically, this is a bug, and should be fixed in some future revision of the code, possibly with modifications to allow dynamic dimensioning of all arrays. However, this feature is generally used only for rectilinear geometries, and in practical terms, NANGLT is expected to be no more than 4.

CHAN.3 Z,THETA
FORMAT(2F10.5)

Columns	Variable	Description
1-10	Z	Axial length of the channel region (inches).
11-20	THETA	Channel orientation, in degrees from vertical. (Default is 0.0, which means the orientation is VERTICAL. Used to calculate gravity term in the momentum equation).

CHAN.4 (NSTEPS(I),VDX(I),I=l,NAZONE)

Read only if NAZONE > 0 on CHAN.1
FORMAT(8(I5,E5.0))

Columns	Variable	Description
1-5 11-15, etc.	NSTEPS(I)	Number of nodes in zone I. (NOTE: The total number of nodes, ∑ NSTEP(I), must be equal to NDX, specified on CHAN.1.)
6-10, 16-20, etc.	VDX(I)	Axial node length (inches) in zone I. (NOTE: the sum of all node length values, for all NAZONE zones must be equal to Z on CHAN.2.)

****up to four pairs of NSTEPS, VDX data may be entered per record.****

****If NAZONE is greater than 4, CHAN.4 is read repeatedly until ****

****all NAZONE pairs of data have been specified.****

Assembly Geometry Description

The input lines CHAN.5 through CHAN.7 are read in sequence as a set NASSEM[CHAN.1] times, once for each assembly in the model. All channels are identified in this manner, assembly-by-assembly. An assembly may contain any number of channels. The assembly number and the assembly type number are read on CHAN.5. Each assembly must have a unique number (1 through NASSEM), but different assemblies can have the same assembly **type** number, if they have the same geometry.

An assembly type is defined by the number of channels and their interconnections, as specified by input on CHAN.7. The geometry data for an assembly type is entered only once with the CHAN.7 input for the *first assembly of that type*. For subsequent assemblies of that type, the value entered for ITYPA on CHAN.5 is recognized as a previously defined type, and CHAN.7 is

not read. The geometry input for the channels of any subsequent assembly of the same type is automatically copied into the appropriate arrays in the code.

The flow and heat transfer correlations to be used must be defined separately for each assembly, however, even if the **assembly type** has been previously defined. This information is supplied on CHAN.6, along with the index of the axial heat generation profile table, which is required if the assembly contains rods. See input instructions for groups HEAT and DRAG for the numbering convention of the user-specified correlations, and group OPER for that of the axial heat generation profile. In developing a model, the user must keep track of the relationship between the correlations flagged on CHAN.6 for each assembly, and the specific correlations specified in HEAT and DRAG, and the profiles in OPER, to assure that the model is using the appropriate selection for the given model.

CHAN.5 NASS,ITYPA,NCHANA,INTAPE,IFREE,TMNCVL(NASS)
FORMAT(4I5,E5.0)

Columns	Variable	Description
1-5	NASS	Assembly number.
6-10	ITYPA	Assembly type number (corresponding to the channel geometry data entered on CHAN.7).
11-15	NCHANA	Number of channels in assembly NASS.
16-20	INTAPE	Flag for I/O unit source of the channel geometry input on CHAN.6 for assembly NASS: = 0; read from the input file (default). = N; read from I/O unit N.
21-25	IFREE	Flag for specifying wall shear boundary condition option for assembly NASS: = 0; wall shear specified by friction factor correlation (default). = 1; laminar zero-slip wall boundary condition. (NOTE: This option is used only when modeling large open plenum regions where fluid-fluid shear must be taken into account.)
26-30	TMNCVL (NASS)	Factor to adjust transverse momentum control volume length in assembly NASS so that the length is defined as $L_{NASS} = f_k$/TMNCVL(NASS) in the calculation of the turbulent crossflow, w´. (Default for TMNCVL(NASS) is 0.0, so that L_{NASS} = 1.0 for w´.)

SPECIAL NOTE: The convention in the code for the assembly numbering scheme assumes that all assemblies with rods are defined first, in sequence, before any assemblies that do not contain rods are defined in the input stream. If there are N assemblies with rods, these must be numbered 1 through N. Any

additional assemblies without rods are then numbered N+1 through NASSEM.

CHAN.6 NAFLX(NASS),NFLMC(NASS),NHFVT(NASS),NPFVT(NASS),MDFLT, NFASS(NASS),ITDPA(NASS)
FORMAT(7I5)

Columns	Variable	Description
1-5	NAFLX (NASS)	Identification number of the axial heat flux profile table [specified on OPER.16 and OPER.17] for assembly NASS (default is 1).
6-10	NFLMC (NASS)	Identification number of the heat transfer coefficient correlation [specified on HEAT.2] for assembly NASS (default is 1). (NOTE: If NTHEAT > 0 on HEAT.1, this is also the identification number of the heat transfer coefficient correlation for lateral flow in assembly NASS. IF NFHEAT > 0 on HEAT.1, this is also the identification number of the heat transfer coefficient correlation for free convection in assembly NASS.)
11-15	NHFVT (NASS)	Identification number of the heat generation versus time forcing function [specified on OPER.13 and OPER.14] to use in assembly NASS during a transient (default is 0; initial value is used throughout the transient.)
16-20	NPFVT (NASS)	Identification number of the pressure drop or flow versus time forcing function [specified on OPER.11 and OPER.12J for assembly NASS (default is 0; initial value is used throughout the transient).
21-25	MDFLT (NASS)	Identification number for the friction factor correlation(s) [specified on DRAG.2 for axial flow and optionally on DRAG.13 for lateral flow], to be used in assembly NASS (default is 1). (NOTE: This value can be superseded in a given channel by specifying a non-zero value for N on CHAN.7 for an individual channel of this assembly type.)
26-30	NFASS (NASS)	Identification number of the fluid properties table [specified on PROP.3] for assembly NASS. (Default is table 1.)
31-35	ITDPA (NASS)	Flag for flow or pressure drop boundary condition for assembly NASS: = 0; specified flow boundary condition. = 1; uniform pressure drop (specified in OPER). = 2; pressure drop resulting in uniform inlet mass flux.

Columns	Variable	Description
		= 3; uniform pressure drop calculated for total mass flow rate (FTOTAL on OPER.2). = 4; zero flow at both top and bottom boundaries. (NOTE: ITDPA(NASS) is needed only if different assemblies have different boundary conditions. If all assemblies see the same boundary condition, ITDPA(NASS) may be entered as 0 for each assembly, and the appropriate boundary condition option is then specified with variable ITDP on OPER.1 for all assemblies. However, there is no penalty for correctly specifying this information here on CHAN.6 for each assembly and on OPER.1 for all assemblies collectively.)

CHAN.7 (N,I,AC,PW,PH,(NANGLE(K),LC(K),GAPS(K),DIST(K),K=1,4),I= 1, NCHANA[CHAN.5])

Read only for a new value of assembly type in ITYPA[CHAN.5].
FORMAT(I1,I4,3E5.3,4(I1,I4,2E5.2))

Columns	Variable	Description
1	N	Index number of friction factor correlation to be applied to channel I; must correspond to a correlation entered on DRAG.2. SPECIAL NOTE: If N is blank or zero, the default index for the friction factor correlation for channel I is MDFLT, entered on CHAN.6, to be applied to all channels in the assembly. This is the recommended approach for typical spent fuel storage system models. (NOTE: If N>0, and the option for lateral friction factor correlations is used (see group DRAG), N also must correspond to the index number of a lateral friction factor correlation entered on DRAG.13.)
2-5	I	Channel identification number (must be entered sequentially, 1 through NCHANA[CHAN.5]).
6-10	AC	Channel nominal area (in^2.)
11-15	PW	Channel wetted perimeter (in.)
16-20	PH	Channel heated perimeter (in.)
21, 36, 51, 66	NANGLE(K)	Optional input, read only if NANGLE > 0 on CHAN.1; if NANGLE = 0, enter zero or blank.

Columns	Variable	Description
		if NANGLE > 0, enter index number of a lateral control volume orientation angle read on CHAN.2. (NOTE: This input is optional, and is needed only when the option for fluid-fluid shear or gravity forces in the transverse direction are modeled; see Section 2.3.2.)
22-25, 37-40, 52-55, 67-70	LC(K)	Identification number of adjacent channel for the L^{th} connection to channel I, where LC(K) > I SPECIAL NOTE: Each connection should be identified only once, as a connection from the lower-numbered channel to the higher-numbered channel.
26-30, 41-45, 56-60, 71-75	GAPS(K)	Width of flow connection between channel I and channel LC(K), (inches).
31-35, 46-50, 61-65, 76-80	DIST(K)	Transverse control volume length (inches) between channel I and channel LC(K). (NOTE: This input is optional. It defines the transverse length of the momentum cell and also the conduction length in the transverse direction. A default value for this distance can be defined for all flow connections by specifying the variable SL on CHAN.8. If this option is used, DIST(K) can be entered as zero.)

CHAN.7 is read NCHANA[CHAN.5] times for a given assembly type.

CHAN.5 through CHAN.7 are read sequentially NASSEM[CHAN.l] times.

Lateral Flow Connection Parameters

All lateral flow connections between the channels within each assembly are defined in the input on CHAN.7. The flow solution for these connections requires an empirical term to define the crossflow resistance (refer to the description of the lateral momentum equation in COBRA-SFS – Theory Manual Section 2.3.3 (Richmond et. al. 2021)). For analysis of rod bundle arrays using subchannel modeling, a single number is usually sufficient to characterize all lateral connections. This input is read on CHAN.8.

The user also has the option of specifying different loss coefficients for different gaps, as appropriate to model geometries in which the lateral control volumes are not all essentially identical. This input is supplied in group DRAG. However, a value still must be entered for KIJ(1) on CHAN.8, for use in gaps where the lateral resistance is not defined in subsequent input. Even if lateral resistance values will be defined for all gaps in subsequent input, KIJ(1) on CHAN.8 must be defined, even if it is specified as zero. The code cannot look ahead in the input stream to see whether or not this input will be required.

Similarly, a value must also be entered for the width-to-length ratio SL, to be used to define the centroid length for any gap that is not specified with a non-zero value for DIST(K) on CHAN.7. If all gaps have a non-zero DIST(K) value specified, SL is not used, and therefore can be entered as zero.

The index numbers of pairs of gaps that exchange momentum due to fluid-fluid shear must be identified by input on CHAN.9, if this option has been specified by input on CHAN.1. The user must understand the gap numbering convention in COBRA-SFS in order to correctly define this input. The gaps are numbered automatically in the code, in the order in which they are specified by input on CHAN.1. For example, in an array of subchannels such as that shown in Figure 2.4 for a fuel assembly, the input for CHAN.7 for subchannels 1 through 6 is as follows;

Sample of CHAN.7 input, to illustrate translation of gap connection input to gap numbering convention in COBRA-SFS:

```
                         1         2         3         4         5         6
column number: 12345678901234567890123456789012345678901234567890123456789012345678890
 CHAN.7 input: 0   1 0.14 0.58 0.17    2 0.20 0.49   19 0.20 0.49
               0   2 0.30 1.22 0.66    3 0.20 0.50   20 0.14 0.49
               0   3 0.30 1.22 0.66    4 0.20 0.50   21 0.14 0.49
```

By the COBRA-SFS gap number convention, the gaps are automatically numbered as follows:

CHAN.7 line	subchannel II	subchannel JJ	gap number
1	1	2	1
	1	19	2
2	2	3	3
	2	20	4
3	3	4	5
	3	21	6

This pattern of gap numbering is continued for all subchannels and their respective interconnections identified by input on CHAN.7.

CHAN.8 KIJ(I),SL,FTM
FORMAT(3E5.0)

Columns	Variable	Description
1-5	KIJ(1)	Nominal flow resistance in lateral connections between channels (default is 0.5).
6-10	SL	Width-to-length ratio for transverse momentum control volume. (NOTE: This is used only for gaps where DIST(K) is entered as zero on CHAN.7 default is 0.5).
11-15	FTM	Factor for turbulent mixing of momentum and energy (default is 0.0).

CHAN.9 (III(L),JJJ(L),L = 1,NSHEAR[CHAN.1])

Read only if NSHEAR > 0
FORMAT (12I5)

Columns	Variable	Description
1-5, 11-15, etc.	III(L)	Identification number of one gap in the L^{th} pair that are connected by lateral fluid shear. (NOTE: See discussion just prior to CHAN.8 input line for an explanation and example of the gap numbering convention in the code.)
6-10, 16-20, etc.	JJJ(L)	Identification numbers of the other gap in the L^{th} pair that are connected by lateral fluid shear.

****Up to six pairs of III,JJJ gap numbers may be entered per record.****

****If NSHEAR is greater than six, repeat this line until all NSHEAR****

****pairs of data have been specified.****

2.4.4 Group VARY–Geometry Variations

This group is used to specify axial variations in channel flow area and lateral flow connection width. It is optional. If this group is not used, all channel areas and lateral flow connection widths are assumed constant over the entire axial length. This option may be required in some cases to appropriately capture the geometry variation for the system. However, as noted for the

variable axial noding option (group CHAN, input flag NAZONE), using the optional group VARY input can make axial noding sensitivity studies problematic. Group VARY input will usually have to be modified to remain consistent with any axial noding variations investigated.

VARY.1 AGROUP,NAFACT,NAXL,NARAMP,NGAPS,NGXL
FORMAT(A4,1X ,5I5)

Columns	Variable	Description
1-4	AGROUP	Enter VARY.
6-10	NAFACT	Number of channels with area variations.
11-15	NAXL	Number of axial locations for channel area variations. (NOTE: All area variation tables use the same table of axial locations.)
16-20	NARAMP	Number of iterations for gradual insertion of area variations into the calculation (default is 1).
21-25	NGAPS	Number of lateral flow connections for which the width varies axially.
26-30	NGXL	Number of axial locations for lateral flow connection width variations. (NOTE: All gap width variation tables use the same axial locations.)

Channel Area Variations

The channel area variations are read only if NAFACT and NAXL on VARY.1 are both greater than zero. The channel area variations are specified on VARY.2 through VARY.4. VARY.2 is read once to define the axial locations of the area variations, then the input lines VARY.3 and VARY.4 are read sequentially, NAFACT times, to define the area variations and the channels affected. Line VARY.3 is read to specify the assembly number and index number of a channel in the assembly that has area variations, then VARY.4 is read to define the table of area variation factors in that channel, at the locations defined by input on VARY.2.

VARY.2 (AXL(J),J = l,NAXL[VARY.1])

Read only if NFACT > 0 and NAXL > 0 on VARY.1
FORMAT(12F5.3)

Columns	Variable	Description
1-5, 11-15, etc.	AXL(J)	Relative axial location (x/L) of the J^{th} channel area variation.

VARY.3 NASS,I

Read only if NAFACT > 0 and NAXL > 0 on VARY.1
FORMAT(2I5)

Columns	Variable	Description
1-5	NASS	Identification number of assembly containing channel I.
6-10	I	Identification number of channel in assembly NASS for which area variations are being specified.

VARY.4 (AFACT(L,J),J=l,NAXL[VARY.1])

Read only if NAFACT > 0 and NAXL > 0 on VARY.1
FORMAT(12F5.3)

Columns	Variable	Description
1-5, 6-10, etc.	AFACT(L,J)	Channel area variation factor (A_{ij}/AC_i) for the L^{th} area variation table at axial level AXL(J), as defined on VARY.2. (NOTE: Area variation factor tables are numbered sequentially, in the order they are read in on VARY.4. Index L of array AFACT is 1 to NAFACT.)

*** VARY.3 and VARY.4 are read sequentially NAFACT[VARY.1] times.***

Lateral Flow Connection Width Variations

The flow connection width variations specified on VARY.5 through VARY.7 are read only if NGAPS and NGXL on VARY.1 are both greater than zero. This input follows the same pattern used in defining the channel area variations. VARY.5 is read once to define the axial locations where gap width variations occur. Then VARY.6 and VARY.7 are read sequentially NGAPS times. VARY.6 is read to define the assembly number and the index number of a gap that has varying width. Then VARY.7 is read to define the table of width variation factors for that gap.

VARY.5 GAPXL(J),J=1,NGXL[VARY.1])

Read only if NGXL > 0 and NGAPS > 0 on VARY.1
FORMAT(12F5.3)

Columns	Variable	Description
1-5, 6-10, etc.	GAPXL(J)	Relative axial location (x/L) of the J^{th} gap width variation.

VARY.6 NASS,K

Read only if NGXL > 0 and NGAPS > 0 on VARY.1
FORMAT(2I5)

Columns	Variable	Description
1-5	NASS	Identification number of assembly containing lateral flow connection K.
6-10	K	Identification number of gap with width variation.

VARY.7 (GFACT(L,J),J= 1,NGXL[CHAN.1])

Read only if NGXL > 0 and NGAPS > 0 on VARY.1
FORMAT(12F5.3)

Columns	Variable	Description
1-5, 6-10, etc.	GFACT(L,J)	Flow connection width variation factor ($GAP_{kj}/GAPS_k$) for the L^{th} variation table at axial level GAPXL(J), as specified on VARY.5. (NOTE: Gap width variation tables are numbered sequentially in the order in which they are read on VARY.7. Index L of array GFACT(L,J) is 1 to NGAPS.)

VARY.6 and VARY.7 are read sequentially NGAPS[VARY.1] times.

SPECIAL NOTE: The geometry variations in this input group affect only channel area and gap width. The perimeter of the flow channel is unaffected by this geometry variation input. The local hydraulic diameter of the channel calculated in the code, therefore, reflects only the change in area.

2.4.5 Group RODS–Fuel Rod Geometry

This group is read to define the geometry and material properties for the fuel rods that are contained in the assemblies defined in group CHAN. If a given model does not contain fuel rods or heater rods simulating fuel rods, the input for this group is not needed and can be omitted from the input stream.

The representation of the fuel rod(s) in a given model is defined by the value specified for variable NC on RODS.1.

If NC = 0, the heat flux at the outer surface of the cladding is treated as a boundary condition in the calculation. In such a case, the code does not solve the rod energy equation for surface temperatures for the rods, and thermal radiation heat transfer within the rod array is not included in the solution. (This option is not typically used in evaluations of spent fuel casks, but can be useful for other applications of the code.)

If NC = 1, the heat flux at the inner surface of the cladding is treated as a boundary condition, and the solid conduction solution for the rod is performed for the cladding only. This yields rod surface temperatures, and therefore thermal radiation within the rod array can be included in the

heat transfer calculation. This option is useful for steady-state calculations, mainly as an efficiency measure for large models, since it reduces memory requirements. But it is not suitable for transient evaluations of spent fuel canisters and casks, since it omits the effect of the thermal inertia of the fuel inside the rods.

If NC>4, the heat generation rate within the fuel is a boundary condition, and the complete rod energy equation is solved for (NC-1) nodal temperatures within the fuel pellet, as well as the cladding surface temperature. Note that a minimum of four nodes are required for representation of heat transfer within the fuel pellet, and therefore NC must be greater than 4 to satisfy this condition since one node represents the fuel cladding material. This is in most cases more than sufficient, since the temperature gradient in the fuel pellet is generally quite flat for spent fuel, due to the low heat generation rate, compared to in-reactor conditions. As with the NC = 1 option, thermal radiation within the rod array can be included in the heat transfer solution, since the NC>4 option also yields rod surface temperatures. The option NC>4 is required for thermal evaluations of spent fuel for transient conditions, in order to properly capture the effect of thermal inertia of the fuel.

Input must be specified in this group for each assembly defined in group CHAN, even if a given assembly does not contain fuel rods. A rod configuration type is specified for each assembly; assemblies that do not contain rods are assigned a configuration type of zero. A configuration type for an assembly that contains rods is defined by the geometry of the rod array; specifically, the total number of rods in the assembly, the nominal rod diameter, radial power distribution, fuel type, and rod-to-channel connection patterns for heat transfer.

SPECIAL NOTE: Consistent with the assembly input for CHAN.5, assemblies with fuel rods must be defined first in a model, in sequence 1 through N, where N is the total number of assemblies that contain fuel rods. Assemblies that do not contain fuel rods, N+1 through NASSEM, must be defined subsequently, with configuration type zero.

RODS.1 AGROUP,NC,NFUELT,NQAX,NRODTP,NTHETA,NRCON,NSCON
FORMAT(A4,1X,7I5)

Columns	Variable	Description
1-4	AGROUP	Enter RODS.
6-10	NC	flag for fuel model option; = 0; no fuel model; cladding outer surface heat flux boundary condition only. (Cladding surface temperatures NOT calculated.) = 1; only the rod cladding is modeled; cladding inner surface heat flux boundary condition only. (NOTE: This option considers all modes of surface heat transfer for the cladding, and cladding surface temperatures are calculated, but it does not include internal fuel conduction. It is applicable to steady state calculations only.) > 4; fuel model with (NC-1) finite-difference nodes in the fuel, plus one node in the cladding. (NOTE: this option considers all modes of surface heat transfer for the rod, plus internal fuel conduction. It can be used in steady state calculations, and is *required* for transient calculations, to provide appropriate thermal inertia terms in the rod energy equation.)
11-15	NFUELT	Number of fuel types for which thermal properties are to be specified on RODS.4. (NOTE: this parameter is not used if NC = 0.)
16-20	NQAX	Flag for temperature dependent fuel properties; = 0; constant fuel properties. = 1; temperature-dependent fuel properties for fuel type 1 only. (NOTE: This option is available only if NC > 4).
21-25	NRODTP	Number of tables of axial fuel type variations. (Applicable only if NC > 4.)
26-30	NTHETA	Number of circumferential nodes in a fuel rod (default is 1). (WARNING: The view factors input in group RADG must be consistent with this input variable.)
31-35	NRCON	Number of rod-to-rod thermal connections for conduction heat transfer.
36-40	NSCON	Number of rod-to-slab thermal connections for conduction heat transfer.

CAUTION: The options flagged by NTHETA, NRCON, and NSCON, for multiple circumferential nodes in the fuel rods and rod-to-rod or rod-to-wall contact for

thermal conduction, were developed for detailed modeling of consolidated fuel assemblies packaged in square canisters in tightly packed hexagonal arrays. Since this approach to spent fuel packaging has been essentially abandoned, there has been no interest in fully developing this feature in the COBRA-SFS code, and it has not been subjected to additional development or testing since the release of Cycle 2. The feature is included in this release, but additional V&V is recommended for any application utilizing this capability in Cycle 4.

Rod Configuration Description

A rod configuration type is defined by the number of rods it contains, the nominal rod diameter, the radial power distribution, and the connections for heat transfer between the rods and channels in an assembly. Each assembly described in group CHAN must be assigned a rod configuration type on RODS.2. If an assembly contains no fuel rods, the rod configuration type must be specified as zero. The rod configuration description for a given non-zero type is entered on RODS.3. As with the assembly types in group CHAN, a rod configuration type description is entered only once. The rod geometry arrays for subsequent assemblies with the same rod type are automatically filled with the appropriate input. The rod configuration description input is read one assembly at a time for all NASSEM[CHAN.1] assemblies.

RODS.2 NOA,ITYPA,NORODS,INTAPE
FORMAT(4I5)

Columns	Variable	Description
1-5	NOA	Assembly number; (NOTE: Must be entered sequentially, 1 through NASSEM[CHAN.1].)
6-10	ITYPA	Rod configuration type number for assembly NOA. (NOTE: All assemblies with rods must be defined before assemblies without rods. See instructions for CHAN.5. If there are no rods in assembly NOA, ITYPA must be entered as zero.)
11-15	NORODS	Number of rods in configuration type ITYPA. (NOTE: If ITYPA is zero, NORODS must be zero.)
16-20	INTAPE	I/O unit number from which the rod input for rod configuration type ITYPA is to be read (default is zero, which causes the code to read this information from the input file).

RODS.3 (N,I,DIA(I),RADIAL(I),(LR(I,L),PHI(I,L),L= 1,8),I1= 1,NORODS[RODS.2])

Read only if NORODS > 0 on RODS.2.
FORMAT(1X,I1,I3,2E5.2,8(I5,E5.2))

Columns	Variable	Description
1-2	N	Rod material properties type flag; Enter as 0 or blank, to obtain default of rod material type 1 for the I^{th} rod of this fuel assembly type. To specify rod material property type other than 1 for the I^{th} rod of this fuel assembly type, enter the rod material type index, as defined by input on RODS.4. (Note that the rod material type index is determined in the code by automatic sequential numbering of rod material type(s) entered on RODS.4.) To specify axially varying rod material properties for the I^{th} rod of this fuel assembly type, enter a value for N that corresponds to the identification number of the appropriate table of axially varying fuel type, as defined by input specified on RODS.7 and RODS.8. (Note that NRODTP must be defined as greater than zero on RODS.1 to access this option.)
3-5	I	Rod identification number. (NOTE: Must be entered in sequence, from 1 to NORODS[RODS.2].)
6-10	DIA(I)	Rod outside diameter (inches).
11-15	RADIAL(I)	Radial heat generation factor for rod I, normalized to the average rod heat generation rate.
16-20, 26-30, etc.	LR(I,L)	Identification number of the L^{th} channel connected to rod I (up to 8 channels can be connected to rod I).
21-25, 31-35, etc.	PHI(I,L)	Fraction of the perimeter of rod I connected to adjacent channel LR(I,L).

RODS.3 is read NORODS times for each new rod configuration type.

RODS.2 and (optionally) RODS.3 are read sequentially as an input.

set NASSEM[CHAN.1] times.

SPECIAL NOTE: The standard width of an input line for COBRA-SFS is no more than 80 columns, based on the typical “useable area” for data entry with Hollerith cards. Modifications in Cycle 4 to enhance radiation modeling for fuel assemblies has resulted in the total format width being expanded to

95 columns on RODS.3. This change is transparent to most users, and is of no concern for “backward compatibility” of the RODS.3 input for files initially created for Cycle 3 or other earlier versions of the code.

However, some users routinely use the space beyond column 80 to insert comments and other documentation information into a COBRA-SFS model input file. The presence of such data in columns 81 through 95 of a RODS.3 input line will cause the code to fail, usually with an error message related to variable type miss-match, as it attempts to read alphanumeric or other format-incompatible data into the fifth and sixth entries of the LR and PHI arrays. The appropriate correction for this problem is to simply move the comment data at least 15 columns to the right, or delete it entirely, if it is no longer needed.

Fuel Material Properties

Fuel material properties must be entered if the model includes nodes in the cladding or fuel pellet (i.e., NC > 0 on RODS.1). The fuel material properties are defined by fuel type, and are automatically numbered sequentially in the order they are read in on RODS.4. (The user must keep track of this numbering sequence for models with NFUELT > 1, for appropriate entry in variable N on RODS.3 for a particular fuel assembly type. Other than the default of N=1, the code has no means of automatically linking a given fuel material type to a particular fuel assembly type.)

If NC > 0 on RODS.1, at least one fuel type must be specified using the input on RODS.4. If the axial fuel type variations option has been flagged (i.e., NRODTP > 0 on RODS.1), a minimum of two fuel types must be defined.

RODS.4 (KFUEL(I),CFUEL(I),RFUEL(I),DFUEL(I),KCLAD(I),CCLAD(I),RCLAD(I), TCLAD(I),HGAP(I),DROD(I),GEOMF(I),DFUELI(I),QVOID(I), I=1, NFUELT[RODS.1])

Read only if NFUELT > 0 on RODS.1
FORMAT(12E5.0,E10.0)

Columns	Variable	Description
1-5	KFUEL(I)	Thermal conductivity (Btu/h-ft-°F) of the fuel material for fuel type I.
6-10	CFUEL(I)	Specific heat (Btu/lbm-°F) of the fuel material for fuel type I.
11-15	RFUEL(I)	Density (lbm/ft^3) of the fuel material for fuel type I.
16-20	DFUEL(I)	Pellet diameter (inches) for fuel type I.
21-25	KCLAD(I)	Cladding thermal conductivity (Btu/h-ft-°F) for fuel type I.
26-30	CCLAD(I)	Cladding specific heat (Btu/lb-°F) for fuel type I.
31-35	RCLAD(I)	Cladding density (lbm/ft^3) for fuel type I.
36-40	TCLAD(I)	Cladding thickness (inches) for fuel type I.
41-45	HGAP(I)	Fuel-clad gap conductance coefficient (Btu/h-ft^2-°F) for fuel type I.

Columns	Variable	Description
46-50	DROD(I)	Outside diameter (inches) of the fuel rod, including the cladding, for fuel type I.
51-55	GEOMF(I)	Fuel rod geometry flag; = 0; solid cylindrical fuel rod with internal heat generation. = 1; annular cylindrical fuel rod with internal heat generation. = 2; annular cylindrical fuel rod with a heat flux boundary condition on the inner fuel surface.
56-60	DFUELI(I)	Inner diameter (inches) of an annular fuel rod of fuel type I (used only if GEOMF(I) > 0).
61-70	QVOID(I)	Heat generation rate (Btu/sec-ft^3) in the central void of an annular fuel rod of type I (used only if GEOMF(I) = 1).

RODS.4 is read NFUELT[RODS.1] times.

Special Warning: As a modeling simplification, guide tube “rods” within an assembly are typically assigned the same fuel type number (N on RODS.3) as the surrounding fuel rods. Since the guide tubes typically have zero decay heat, the very small inconsistency introduced in “fuel rod type” data applied to them has no discernable effect on calculated temperatures. The effort of defining special fuel types for the guide tube rods may be worth investigating, with appropriate modeling sensitivity studies, for a particular application in which the guide tubes have some heat generation, such as that due to being loaded with non-fuel hardware (which is a documented practice at some sites). It may also be worth investigating for transient applications, since representing the guide tubes as fuel rods results in a small increase in the thermal inertia of the assembly. But in general, it is expected that uncertainties and general conservatisms in overall assembly decay heat and rod-by-rod decay heat distribution would tend to swamp any sensitivity of results to this simplification in the modeling approach for the guide tubes. The user is strongly advised to perform appropriate sensitivity studies, if tempted to remove this simplification from a specific model.

Temperature-Dependent Fuel Properties

The user has the option of specifying temperature-dependent material properties for fuel type 1, using the input on RODS.5 and RODS.6. This option is selected by setting NQAX to 1 on RODS.1. However, all other fuel types will have material properties that are constant for all temperatures. For spent fuel thermal analysis, variation in fuel properties with temperature is generally expected to be insignificant, and this option would be used only in very special applications.

RODS.5 NTNODE

Read only if NQAX = 1 on RODS.1
FORMAT(I5)

Columns	Variable	Description
1-5	NTNODE	Number of entries in the temperature-dependent material properties table for fuel type 1.

RODS.6 (TVARY(I),VARYK(I),VARYCP(I),VARYR(I),I= l,NTNODE[RODS.5])

Read only if NQAX = 1 on RODS.1
FORMAT(12E5.0)

Columns	Variable	Description
1-5, 21-25, 41-45	TVARY(I)	Temperature (°F).
6-10, 26-30, 46-50	VARYK(I)	Thermal conductivity (Btu/h-ft-°F) of fuel type 1 at temperature TVARY(I).
11-15, 31-35, 51-55	VARYCP(I)	Specific heat (Btu/lb-°F) of fuel type 1 at TVARY(I).
16-20, 36-40, 56-60	VARYR(I)	Density (lb/ft^3) of fuel type 1 at TVARY(I).

The temperature-varying material properties must be entered as a monotonically increasing table, with TVARY(1) the lowest temperature in the table, and TVARY(NTNODE) the highest temperature.

Axial Fuel Variation

This input is optional, and is read only if NRODTP is greater than zero on RODS.1. It allows the user to account for axial variations in material composition of rods of type 1, using the fuel types defined by input on RODS.4. The fuel types are assigned to axial ranges in tables of axial location versus fuel type, using the input lines RODS.7 and RODS.8. The entries must be monotonically increasing on axial location. The tables are numbered 1 through NRODTP, in the order they are entered on RODS.7 and RODS.8. When this option is used, the value specified for N on RODS.3 for each rod must correspond to the index of the table containing the appropriate varying fuel type profile, as entered on RODS.7 and RODS.8.

RODS.7 KNZ

Read only if NRODTP > 0 on RODS.1
FORMAT(I5)

Columns	Variable	Description
1-5	KNZ	Number of axial zones in the I^{th} table of axial distance versus fuel type, where I is from 1 to NRODTP[RODS.1].

RODS.8 ((ZEND(I,K),IZTYP(1IK),K = 1,KNZ[RODS.7]),I= l,NRODTP[RODS.1])

Read only if NRODTP > 0 on RODS.1.
FORMAT(6(E5.0,I5))

Columns	Variable	Description
1-5, 11-15, etc.	ZEND(I,K)	Relative axial location (x/L) of the end of the K^{th} fuel zone of the I^{th} axially varying fuel type table.
6-10, 16-20	IZTYP(I,K)	Index of material type (from the order of entry on RODS.4), in fuel zone ending at ZEND(I,K) in the I^{th} etc. table of axial distance versus fuel type.

RODS.7 and RODS.8 are read in sequence, NRODTP times.

Rod-to-Rod Thermal Connections

This input is optional, and is read only if NRCON[RODS.1] is greater than zero. It is used to specify heat transfer paths due to direct physical contact between rods. It is assumed that the contact conductance and geometry factor are the same for all rods in contact with each other. This input is entered on RODS.9. The specific rods that are in contact with other rods must be identified on RODS.10. This input is read for all NRCON[RODS.1] rods. Up to 6 rods may be in contact with a given rod, based on the maximum possible packing, which would be a hexagonal array. All rods in contact with a given rod must be identified; this input is not processed with the automatic reciprocity utilized in other thermal connection input options in the code.

CAUTION: As noted in the instructions for RODS.1, the option for multiple circumferential nodes in the fuel rods and rod-to-rod contact for thermal conduction was developed for detailed modeling of consolidated fuel assemblies packaged in square canisters in tightly packed hexagonal arrays. This feature has not been subjected to additional development or testing since the release of Cycle 2. It is included in the current release, but additional V&V is recommended for any application utilizing this capability in Cycle 4.

RODS.9 RCON,RRDIM

Read only if NRCON > 0 on RODS.1
FORMAT(2E10.5)

Columns	Variable	Description
1-10	RCON	Contact resistance between rods, (h-ft-°F/Btu).
11-20	RRDIM	Geometry factor of thermal connections for rod-to-rod contact, defined as the ratio of one-half the cladding node thickness over the perimeter of the rod circumferential node. That is, $$RRDIM = \frac{\frac{1}{2}TCLAD}{\frac{\pi d_{rod}}{NTHETA}}$$

RODS.10 NTH,NRR(NTH),(NCLAD(NTH,I),I=1,6)

Read only if NRCON[RODS.1] > 0
FORMAT(8I5)

Columns	Variable	Description
1-5	NTH	Index number of connection for rod-to-rod contact resistance (from 1 to NRCON).
6-10	NRR(NTH)	Index number of the N^{th} rod in contact with other rods.
11-15, 16-20, 21-25, 26-30, 31-35, 36-40	NCLAD (NTH,I)	Index numbers of the NTHETA rods that are in contact with rod NRR(NTH).

****RODS.10 is read NRCON[RODS.1] times.****

Rod-to-Slab Thermal Connections

This input is optional, and is read only if NSCON[RODS.1] is greater than zero. It is used to specify heat transfer paths due to direct contact between rods and structures modeled with slabs.

CAUTION: As noted in the instructions for RODS.1, the option for multiple circumferential nodes in the fuel rods and rod-to-wall contact for thermal conduction was developed for detailed modeling of consolidated fuel assemblies packaged in square canisters in tightly packed hexagonal arrays. This feature has not been subjected to additional development or testing since the release of Cycle 2. It is included in the current release, but additional V&V is recommended for any application utilizing this capability in Cycle 4.

RODS.11 WCON,WRDIM

Read only if NSCON > 0 on RODS.1
FORMAT(2E10.5)

Columns	Variable	Description
1-10	WCON	Contact resistance between rods and walls, (h-ft-°F/Btu)
11-20	WRDIM	Geometry factor of thermal connections for rod-to-wall contact, defined as the ratio of one-half the wall surface node thickness over the perimeter of the wall node.

RODS.12 NTH,NRW(NTH),(NWALL(NTH,I),I =1,6)

Read only if NSCON[RODS.1] > 0
FORMAT(8I5)

Columns	Variable	Description
1-5	NTH	Index number of connection for rod-to-wall contact resistance (from 1 to NSCON).
6-10	NRW(NTH)	Index number of the N^{th} wall node in contact with rod(s).
11-15, 16-20, 21-25, 26-30, 31-35, 36-40	NWALL (NTH,I)	Index numbers of the rods that are in contact with wall node NRW(NTH).

****RODS.12 is read NSCON[RODS.1] times.****

2.4.6 Group SLAB–Solid Structure Geometry

This group defines the geometry, material properties, and heat transfer connections for the solid structures that comprise a cask or other physical system being analyzed. It can be used to model such things as the basket that holds the fuel assemblies, the support structure for the basket, the various shells comprising the cask body (e.g., the metal of the cask body, layers of shielding material), and other internal structures of the cask. Conduction heat transfer is modeled within and between slab nodes, and slab nodes can also exchange energy with the fluid by conduction and convection. Thermal radiation exchange with the fuel rods and with other slab nodes can also be modeled.

SPECIAL NOTE: Before venturing to enter actual input data for this group, the user is well advised to have created a detailed noding map for the model, and have a clear definition of all thermal connections (slab-to-slab and slab-to-fluid) needed to appropriately represent the geometry and heat transfer paths in the system. (Spreadsheets are very useful for this purpose.) The input in this group is highly interdependent, and cannot be approached in a linear fashion. The user must try to keep the “big picture” in mind at all times. See COBRA-SFS – Theory Manual Section 7.0 (Richmond et. al. 2021) for illustrative examples, and Section 3.0 of this report for additional discussion related to efficient model construction.

As discussed in COBRA-SFS – Theory Manual Section 2.4.2 (Richmond et. al. 2021), the energy equation for the solid structure nodes is solved as a network of thermal connections, with heat transfer paths defined for the various faces of each node. Solid structure nodes are typically connected to physically adjacent nodes, usually on more than one face, and can also have thermal connections to fluid nodes in adjacent channels. The equations in COBRA-SFS – Theory Manual Section 2.4.2 (Richmond et. al. 2021) are presented with a very generalized thermal conductance between adjacent nodes and with the fluid. The input in group SLAB permits the user to define these connections to represent the geometry of the system being modeled, in as fine a level of detail as the purpose of the modeling might require. This input is extremely flexible, allowing the user to model almost any reasonable geometry configuration. Although the network might be intimidatingly complex, the structure of this input is, in application, relatively simple.

Figure 2.8 shows the basic form of a slab-to-slab connection for thermal conduction between nodes that can exchange energy with each other. The simplest formulation of this connection would be for nodes forming a continuous material. However, it is also possible to account for gaps between adjacent material and differences in material composition for adjacent nodes, as discussed here and in the detailed instructions for the specific input lines, SLAB.2 and SLAB.3. The thermal resistance for heat transfer between two slab nodes is defined simply as

$$R = \frac{F_{G,N} + F_{G,M}}{k} \tag{2.1}$$

where

$F_{G,N}$ = geometry factor for node N of the pair
$F_{G,M}$ = geometry factor for node M of the pair
k = thermal conductivity (Btu/s-ft-°F) of the material in the conduction path between the centers of nodes N and M

It should be noted that as formulated in Eq. (2.1), the thermal resistance implicitly assumes that the two adjacent nodes N and M are comprised of the same material. As implemented in the code, however, the thermal resistance is actually calculated directly from the material properties specified for each node. The main purpose of the definition in Eq. (2.1) is to illustrate the convention used in COBRA-SFS to define the solid conduction network in terms of the geometry of the noding. The user therefore can define the network with input based on the geometry of the noding alone. This is because the geometry factor for a given connection is defined as

$$F_G = \frac{W}{L} \tag{2.2}$$

where

W = distance from the solid node center to the edge facing the adjacent node
L = length of the solid node at the face in contact with the adjacent node

Figure 2.8 illustrates the application of the resistance definition in Eq. (2.1) and the geometry factor definition in Eq. (2.2) with a diagram showing a segment of typical slab-to-slab connections in a model. For node N in this figure, there are two paths for thermal energy exchange with adjacent nodes, and therefore two connection types would be defined in the input.

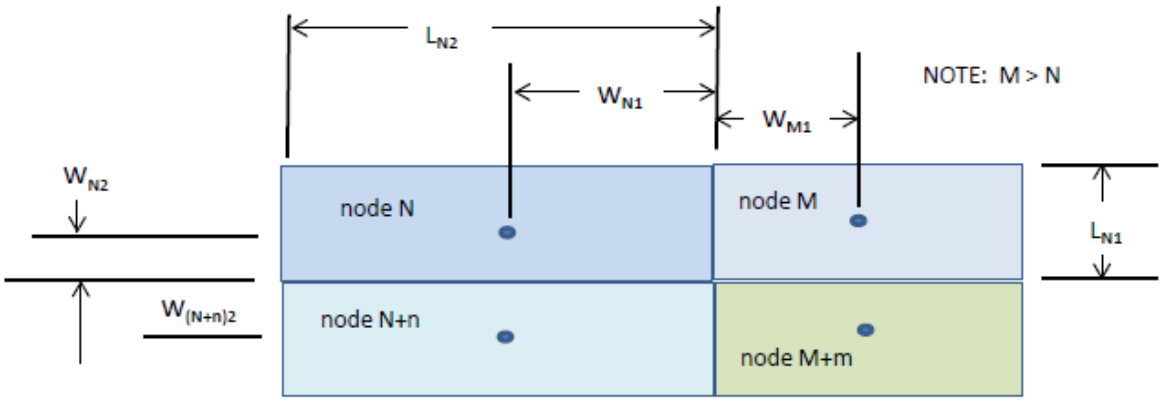


Figure 2.9. Diagram of Basic Slab-to-Slab Thermal Resistance Connection Geometry

For the connection between node N and node M, the thermal resistance in Eq. (2.1) would be determined from the geometry as

$$R_1 = \frac{F_{G,N1} + F_{G,M1}}{k}$$

The corresponding geometry factors would be defined as

$$F_{G,N1} = \frac{W_{N1}}{L_{N1}}$$

$$F_{G,M1} = \frac{W_{M1}}{L_{N1}}$$

Similarly, the connection between node N and node (N+n) would be represented in Eq. (2.1) with corresponding geometry factors defined as

$$F_{G,N2} = \frac{W_{N2}}{L_{N2}}$$

$$F_{G,(N+n)2} = \frac{W_{(N+n)2}}{L_{N2}}$$

For simplicity, this illustration is presented for a rectangular grid of nodes, but the thermal connection network in the COBRA-SFS model is NOT constrained to any grid geometry. This connection logic can be extended to any geometry, unconstrained by any particular coordinate system, since the geometry factors are based simply on the length of the conduction path between the node centers.

The default for slab-slab connections in the thermal connection network is to assume perfect contact between adjacent slab nodes. For nodes representing continuous material, this is obviously realistic. It is also an appropriate representation for nodes representing structures expected to be in intimate contact with substantial contact pressure (i.e., on the order of thousands of pounds per square inch), such as plates comprising the base of a vertical package, which bear the full weight of the fuel assemblies in the package, and possibly a substantial load due to the structural body of the package. It may also be an appropriate assumption for components that are expected to be in intimate contact at operating conditions, due to differential thermal expansion, although this assumption should always be verified in a given model with relevant structural calculations, based on as-built conditions, and should not be assumed *a priori* for any design.

In general, manufacturing tolerances and fabrication constraints are likely to result in small gaps between adjacent components of a spent fuel storage canister or cask. Typically, these gaps are too small to represent explicitly as fluid channels, and it is more efficient to represent them as an additional thermal resistance between the faces of the adjacent slab nodes. Figure 2.9 shows a diagram illustrating a gap due to imperfect contact between adjacent slab nodes.

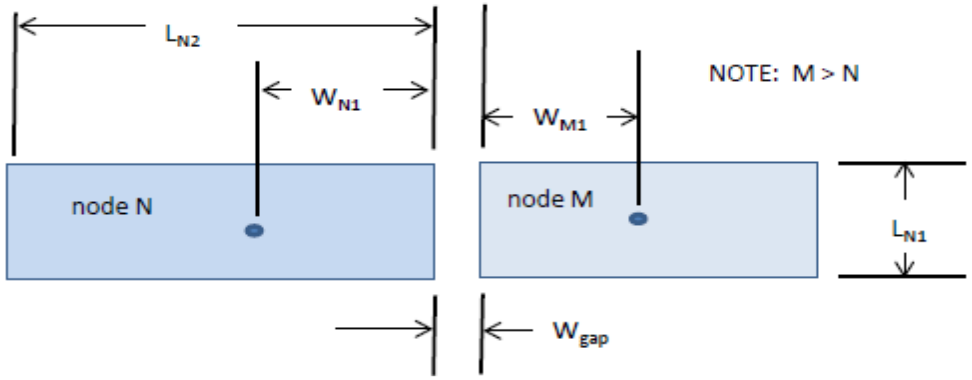


Figure 2.10. Diagram of Slab-to-Slab Thermal Resistance Connection Geometry, Including Gap between Adjacent Nodes

The resistance to heat transfer due to the gap between the adjacent nodes is calculated as an additional resistance in series with the resistance through the solid material of the adjacent nodes, and is added to the resistance calculated using the relationship in Eq. (2.1). The conduction resistance of the gap is determined with the same formula, as simply

$$R_{gap} = \frac{W_{gap}}{L_{gap}}\left(\frac{1}{k_{gap}}\right) \quad (2.3)$$

where

W_{gap} = width of gap between the two slab nodes
L_{gap} = length of the node face that sees the adjacent slab node (in the example in Figure 2.9, $L_{gap} = L_{N1}$)
k_{gap} = thermal conductivity (Btu/s-ft-°F) of the material in the gap (usually the backfill gas, typically helium)

The user also has the option of specifying thermal radiation as a component of heat transfer across the gap. This is discussed in the detailed instructions for this input; see SLAB.2 below.

SPECIAL NOTE: The astute reader may have noticed at this point that the thermal resistance terms defined using Eqs. (2.1) through (2.3) actually should be on a *per area* basis, to be properly consistent with the slab energy equation as presented in COBRA-SFS – Theory Manual Section 2.4.2 (Richmond et. al. 2021). However, since models are generally constructed based on a cross-sectional map of the slab node connections showing heat flow paths in the radial direction (as illustrated in examples shown in COBRA-SFS – Theory Manual Section 7.0 (Richmond et. al. 2021) and in Section 3.0 of this report), the COBRA-SFS input defining these heat transfer paths is constructed such that the user need only supply the geometry dimensions in the plane of the map. These length terms are automatically multiplied by the node axial length (defined by input in group CHAN) when used in the solution of the conservation equations, to determine the appropriate local radial area for a given thermal connection. This allows effortless accommodation of variable axial noding, and permits the user to change the specified axial noding in a model simply by changing the input for the appropriate line (or lines, if using variable axial noding) in group CHAN, without having to redefine all the solid node thermal connections, as well. The resulting change in radial heat transfer area at a given location within the conduction network, and therefore the effect on local thermal resistance between nodes, is automatically propagated through the geometry of the model.

Thermal connections between slab nodes and fluid nodes (i.e., channels), are specified in a manner similar to the slab-slab thermal resistance defined in Eq. (2.1). The only significant difference is that the thermal resistance on the fluid node side of the relationship is accounted for in the fluid energy equation, and as a result, only a single geometry factor is needed, for the solid node portion of the connection. The complete geometry factor for the slab node in a slab-fluid connection, therefore, is simply

$$F_G = \frac{W}{L_f} \quad (2.5)$$

where

W = distance from the solid node center to the node edge facing the channel
L_f = Length of the portion of the slab node face that sees the channel

Figure 2.10 illustrates the geometry factor definition for slab-fluid connection, with example connections between node N and three adjacent fluid channels. In this example, it would be necessary to define three connection types to represent slab-fluid connections for node N and all geometrically identical nodes.

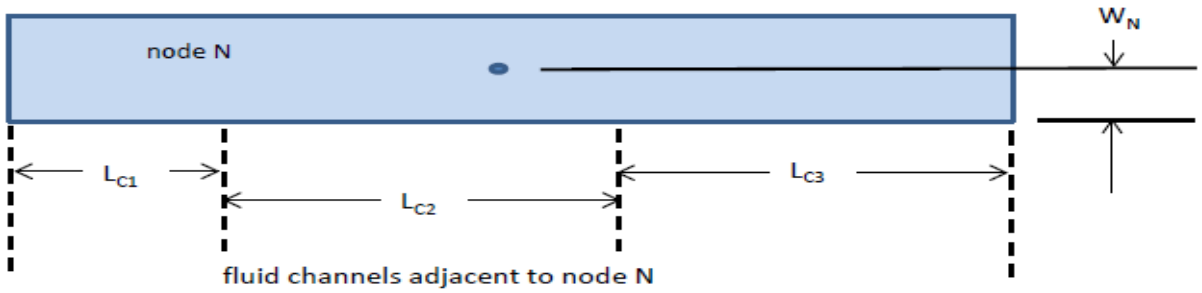


Figure 2.11. Diagram of Basic Slab-to-Fluid Thermal Resistance Connection Geometry

$$F_{G,N1} = \frac{W_N}{L_{C1}}$$

$$F_{G,N2} = \frac{W_N}{L_{C2}}$$

$$F_{G,N3} = \frac{W_N}{L_{C3}}$$

With these definitions of the slab-to-slab and slab-to-fluid thermal resistances in mind, and a detailed map of the cross-section of the noding developed for a model, the user can begin the long march through the group SLAB input. On SLAB.1, the user must specify the total number of slab nodes in the model, with the value entered for NWK. In the expectation that many of the slab-to-slab and slab-to-fluid interconnections will be identical, due to the regular and repetitive geometry of a typical multi-assembly spent fuel package (see examples in COBRA-SFS – Theory Manual Section 7.0 (Richmond et. al. 2021)), the thermal connections are indexed by type, for efficiency of input processing. In this context, a thermal connection type is defined mainly by the geometry of a connection, as discussed above. The number of types of solid-to-solid connections, (NKSS), and the number of types of solid-to-fluid connections (NKSF), for a given model must be specified by the input on SLAB.1.

The default for slab material properties is to assume that the slab is the same material axially from top to bottom. However, the code includes the option of defining axially varying material properties for selected slab nodes. This option is flagged by a non-zero value for NMAT on SLAB.1.

SLAB.l AGROUP,NKSS,NKSF,NWK,NMAT
FORMAT(A4,1X,1315)

Columns	Variable	Description
1-4	AGROUP	Enter SLAB.
6-10	NKSS	Number of solid-to-solid thermal connection types.
11-15	NKSF	Number of solid-to-fluid thermal connection types.
16-20	NWK	Number of solid structure nodes.
21-25	NMAT	Number of different axially varying material type profiles to be defined by input on lines SLAB.8 and SLAB.9. Enter zero (0) for axially uniform material properties for all slab nodes.

Solid-to-Solid Thermal Connection Types

A total of NKSS[SLAB.1] thermal connection types must be defined by the input on SLAB.2. The user has two options for specifying the thermal resistance of a given connection type. The typical approach is to simply define the appropriate geometry factors for the connection, as defined in Eq. (2.2) and illustrated in the example shown in Figure 2.8. The code then automatically calculates the appropriate thermal resistance, using the thermal conductivity associated with the material property type for each of the two nodes in a given connection. The geometry factors for the two adjacent nodes of a given type are entered in RDIMA(IC) and RDIMB(IC) on SLAB.2.

The convention for this input is that RDIMA(IC) is the geometry factor for the lower-numbered node of any pair of nodes connected by thermal connection type IC, and RDIMB(IC) is the geometry factor for the higher-numbered node of the pair. This input uses the convention of “forward-looking only” that is implemented wherever possible in the input stream for COBRA-SFS. This means that for any given node N, the user has to enter only connections to higher-numbered nodes M (i.e., N < M). This is something to bear in mind when defining the node numbering pattern in the noding map. In most geometries, there will be some unavoidable reflexivity in the numbering scheme and therefore in the connection type definitions, where some node connection geometry N2-to-M2 will be a mirror image of some other N1-to-M1 connections, but careful use of the flexibility of the node numbering scheme can be helpful in reducing the number of connect types that must be defined by input. This is generally worth some effort, as it will simplify model verification (see Section 3.2 for further discussion of this issue).

Alternatively, the user may choose to specify the connection input as the value of R for a specific connection, having calculated the appropriate value independently, using Eq. (2.1). This value is entered in variable RSER(IC) on SLAB.2. When the thermal resistance for a given connection type is entered in this manner, the geometry factors specified with RDIMA(IC) and RDIMB(IC) are not used in the code, and can be left blank, or entered as zero (0.0). Conversely, when the option to enter the geometry factors with RDIMA(IC) and RDIMB(IC) is used, RSER(IC) is left blank, or entered as zero (0.0).

In most cases, entering the individual geometry factors for a connection type, using RDIMA(IC) and RDIMB(IC), will be more convenient than calculating RSER(IC). In order to enter the thermal resistance using RSER(IC), the user must first calculate both geometry factors for each node, then sum them and divide by the average thermal conductivity of the connection path. The main purpose of this option originally was to simplify the input for averaged nodes representing multiple layers of material with different thermal conductivities. The current recommendation in COBRA-SFS modeling is to represent such layers as individual nodes within the network, since memory limitations are generally no longer a problem in model building. The code can easily handle the material mapping on a one-to-one basis, without the simplification of determining averaged or effective conductivities for homogenized regions, and this option is therefore less likely to be needed. However, it is retained in Cycle 4, to maintain code flexibility and backward compatibility.

If the connection type includes a gap between adjacent nodes N and M, the geometry factor for the gap is specified by input RPAR(IC), calculated using the relationship in Eq. (2.3). Thermal radiation across the gap can be included in the connection, simply by specifying an appropriate surface emissivity for each face of the nodes connected across the gap, by means of input EMA(IC) and EMB(IC). The area for thermal radiation exchange is calculated in the code, from the value entered in ARAD(IC). This input value corresponds to the connection length in the geometry factor, as defined in Eq. (2.2). As with the local area for connections between nodes in intimate contact, the radial area for heat transfer across the gap is calculated in the code, using the value entered for ARAD(IC) times the local axial node length (defined by input in group CHAN).

If there is no gap for a given connection type, RPAR(IC) is entered as zero (0.0), and no input is needed for EMA(IC), EMB(IC), and ARAD(IC).

SLAB.2 (IC,RDIMA(IC),RDIMB(IC),RSER(IC),RPAR(IC),EMA(IC),EMB(IC), ARAD(IC), IC = l,NKSS[SLAB.1])

Read only if NKSS > 0 SLAB.1
FORMAT(I5,4E10.0,3F5.0)

Columns	Variable	Description
1-5	IC	Identification number of a solid-to-solid thermal connection type. (NOTE: Must be entered in sequence from 1 to NKSS.)
6-15	RDIMA(IC)	Geometry factor, (as defined in Eq. (2.2)), for the lower numbered node of any pair of nodes connected by thermal connection type IC. (NOTE: Not used if RSER(IC) is specified.)
16-25	RDIMB(IC)	Geometry factor, (as defined in Eq. (2.2)), for the higher-numbered node of any pair of nodes connected by thermal connection type IC. (NOTE: Not used if RSER(IC) is specified.)
26-35	RSER(IC)	Thermal resistance (s-ft-°F/Btu), (as defined in Eq. (2.1)), between the centers of adjacent nodes connected by thermal connection type IC. (NOTE: Not used if RDIMA(IC) and RDIMB(IC) are specified.)
36-45	RPAR(IC)	Thermal resistance (s-ft-°F/Btu) of the gap between adjacent solid structure nodes connected by thermal connection type IC, as defined in Eq. (2.3). (NOTE: This is optional input. Enter zero if there is no significant gap resistance between nodes of this type.)
46-50	EMA(IC)	Surface emissivity of lower-numbered node of the pair, for radiative heat transfer across the gap between any pair of nodes connected by thermal connection type IC. (Enter zero if RPAR(IC) is zero.)
51-55	EMB(IC)	Surface emissivity of higher-numbered node of the pair, for radiative heat transfer across the gap between any pair of nodes connected by thermal connection type IC. (Enter zero if RPAR(IC) is zero.)
56-60	ARAD(IC)	Length of node (inches) on the face looking at the gap, for calculation of heat transfer area for energy exchange across the gap between adjacent nodes connected by thermal connection type IC. (Enter zero if RPAR(IC) is zero.)

SLAB.2 is read in NKSS[SLAB.1] times.

Solid Structure Node Description

This input describes each of the NWK solid nodes in the model, identifies its material type, and specifies all interconnections for heat transfer with other solid nodes. The specification of the heat transfer connections between slabs follows the same “forward-looking” convention as that for specifying connections between channels for lateral flow. A connection is identified only once by defining it from the lower-numbered node to the higher-numbered node of the pair. For example, if node 2 connects to node 7, the connection is identified only on the input line for node 2. This connection should not be defined again on the input line for node 7.

SLAB.3 (KW,MATTYP(KW),WALLXC(KW),QSLAB(KW),NAXK(KW), NSLAB,(KWAL(L,KW),ICON(L,KW),L= 1,NSLAB))

Read only if NWK > 0 on SLAB.1.
FORMAT(2I5,2F5.0,12I5)

Columns	Variable	Description
1-5	KW	Identification number of a solid structure node. (NOTE: Must be entered in sequence, from 1 to NWK[SLAB.1].)
6-10	MATTYP(KW)	Material type identification for node KW: If NMAT = 0 on SLAB.1 all slabs have axially uniform material properties, and MATTYP(KW) must correspond to a material type specified on PROP.3. If NMAT > 0 on SLAB.1, but this particular slab KW has axially uniform material properties; MATTYP(KW) is specified as a material type defined on PROP.3. If NMAT > 0 on SLAB.1, and this particular slab KW has axially varying material properties; MATTYP(KW) is specified as the index of an axially varying material property profile defined on SLAB.8 and SLAB.9. (See discussion of input for SLAB.8 and SLAB.9.)
11-15	WALLXC(KW)	Solid node cross-sectional area (inches^2) in the axial direction, for axial conduction. (NOTE: Enter zero to neglect axial conduction in slab node KW for this model.)
16-20	QSLAB(KW)	Solid node volumetric heat generation rate (Btu/h-ft^3). (NOTE: Solid conduction nodes are not generally used to represent structures with heat generation in a COBRA-SFS model. This option has been exercised only for limited special modeling applications. Use with caution.)

21-25	NAXK(KW)	Identification number of the axial heat generation profile table [specified on OPER.16] to be applied to QSLAB(KW) (default is 1).
26-30	NSLAB	Number of thermal connections to adjacent solid nodes that have index numbers greater than KW.
31-35, 41-45, etc.	KWAL(L,KW)	Identification number of the L^{th} solid node adjacent to node KW, where KWAL(L,KW) > KW.
36-40, 46-50, etc.	ICON(L,KW)	Identification number of the thermal connection type for the connection between node KW and node KWAL(L,KW). (NOTE: This must correspond to a thermal connection type identified on SLAB.2.)

A maximum of five pairs of KWAL,ICON values may be specified on SLAB.3 for given slab node. If NSLAB is greater than five, continue the connection input on SLAB.4 to define all connections to node KW before going on to the input for the next slab node.

SLAB.4 (KWAL(L,KW),ICON(L,KW),L=6,NSLAB)

Read only if NSLAB > 5 on SLAB.3.
FORMAT(30X,10I5)

Columns	Variable	Description
1-30		BLANK
31-35, 41-45, etc.	KWAL(L,KW)	Identification number of the L^{th} solid node adjacent to node KW, where KWAL(L,KW) > KW.
36-40, 46-50, etc.	ICON(L,KW)	Identification number of the thermal connection type for the connection between node KW and node KWAL(L,KW). (NOTE: This must correspond to a thermal connection type identified on SLAB.2.)

SLAB.4 is read with five pairs of entries per line.

SLAB.3 (with SLAB.4, if needed) is read NWK[SLAB.1] times.

Solid-to-Fluid Thermal Connection Types

This input is required only if solid structure nodes have heat transfer connections to flow channels. If NKSF[SLAB.1] is zero, this input is not read. As with the solid-to-solid thermal connections, the solid-to-fluid connections are defined by type. Connections of the same type have the same thermal resistance, where the thermal resistance of a connection is defined as described in Eq. (2.1), with geometry factor defined as in Eq. (2.5), and illustrated in Figure

2.10. In a similar manner as with the slab-slab connection types defined by input on SLAB.2, the user has two options for specifying the thermal resistance of a slab-fluid connection type. In the first option, the user can simply define the individual geometry factor, F_G, for the connection, using Eq. (2.5). This geometry factor is entered in RDIMF(IC) on SLAB.5. Alternatively, using the second option, the resistance can be specified as the value of R calculated using Eq. (2.1), with a geometry factor defined as in Eq. (2.5). This value is entered in variable RWAL(IC) on SLAB.2. For either option, the length used to define the geometry factor must be entered in variable WID(IC).

SLAB.5 (IC,RDIMF(IC),RWAL(IC),WID(IC),IC=1,NSKF[SLAB.1])

Read only if NKSF > 0 on SLAB.1
FORMAT(I5,2E5.0,E10.0)

Columns	Variable	Description
1-5	IC	Solid-to-fluid thermal connection type identification number. (NOTE: Must be entered in sequence, from 1 to NKSF.)
6-10	RDIMF(IC)	Geometry factor, as defined in Eq. (2.5), from the center of the solid node to the surface that faces the channel. (NOTE: Not used if RWAL(IC) is specified.)
11-15	RWAL(IC)	Thermal resistance (s-ft-°F/Btu) (as defined in Eq. (2.1), with geometry factor defined as in Eq. (2.5)), from the center of the solid node to the node surface that faces the channel. (NOTE: Not used if RDIMF(IC) is specified.)
16-25	WID(IC)	Length (inches) of the face of the solid node that sees a fluid channel node. (NOTE: As illustrated in Figure 2.10, this length for a given connection type may not be the entire length of the face of a node that sees the fluid with connection type IC. A given solid node may have more than one connection type defined to fully capture its solid-fluid connections. (See COBRA-SFS – Theory Manual Section 7.0 (Richmond et. al. 2021) for further examples.))

SLAB.5 is read NKSF[SLAB.1] times.

Solid-to-Fluid Connection Description

This input specifies the connections between solid nodes and fluid channels. It is read only if NKSF on SLAB.1 is greater than zero. Connections are defined using this input line only for nodes that see fluid nodes. In many cases, the number of nodes with fluid connections will be less than the total number of solid nodes, NWK, specified on SLAB.1. To signal the end of this input, a line must be entered with zero for KW on SLAB.6, after all nodes with fluid connections have been identified.

SPECIAL NOTE: Neglecting to enter the final SLAB.6 input line with KW = 0 will cause the code to fail with a misleading error message about inappropriately formatted input, as it attempts to read subsequent non-SLAB.6 input lines with the SLAB.6 input formatting.

SLAB.6 (KW,NCHN,(IASSM(L,KW),IKW(L,KW),ICF(L,N),L= 1,NCHN))

Read only if NSKF > 0 on SLAB.1
FORMAT(2I5,4(3I5))

Columns	Variable	Description
1-5	KW	Identification number of solid node with thermal connection(s) to channel(s). Enter zero for KW on a final SLAB.6 input line, to terminate this input when all solid-to-fluid connections have been identified.
6-10	NCHN	Number of thermal connections to adjacent channels for node KW.
11-15, 26-30, etc.	IASSM(L,KW)	Index number of assembly containing the L^{th} channel with thermal connection to node KW. (Must correspond to the index number of an assembly defined by input in group CHAN.)
16-20, 31-35, etc.	IKW(L,KW)	Identification number of L^{th} channel in assembly IASSM(L,KW) that has a solid-fluid thermal connection to node KW.
21-25, 36-40, etc.	ICF(L,KW)	Identification number of solid-to-fluid thermal connection type for the connection between node KW and channel IKW(L,KW). (NOTE: This must correspond to a thermal connection type specified on SLAB.5.)

A maximum of four sets of values of IASSM, IKW, and ICF may be specified on SLAB.6. If NCHN is greater than four, continue the connection input on SLAB.7 to define all connections to node KW before going on to the input for the next node with connections to the fluid.

SLAB.7 (IASSM(L,KW),IKW(L,KW),ICF(L,N),L =5,NCHN))

Read only if NCHN > 4 on SLAB.5
FORMAT(15X,4(3I5))

Columns	Variable	Description
1-10		BLANK
11-15, 26-30, etc.	IASSM(L,KW)	Index number of assembly containing the L^{th} channel with a thermal connection to node KW. (Must correspond to the index number of an assembly defined by input in group CHAN.)
16-20, 31-35, etc.	IKW(L,KW)	Identification number of L^{th} channel in assembly IASSM(L,KW) that has a solid-fluid thermal connection to node KW.
21-25, 36-40, etc.	ICF(L,KW)	Identification number of solid-to-fluid thermal connection type for the connection between node KW and channel IKW(L,KW). (NOTE: This must correspond to a thermal connection type specified on SLAB.5.)

SLAB.7 is read with 4 sets of (IASSM, IKW, ICF) entries per line.

SLAB.6 (with SLAB.7, if needed), is read as many times as necessary

to define the connections between solid nodes and fluid nodes.

The input on SLAB.6 is terminated by entering a line with zero entered for KW, after all nodes with connections to flow channels have been identified.

Axial Variation of Slab Material

This option allows the user to specify different material properties at different axial locations along the length of a slab node. The material type flag MATTYP(KW) on SLAB.3 for a specific slab node KW modeled with axially varying material properties is set to the appropriate profile index number defined by input on SLAB.8 and SLAB.9. This makes the input chain a little circular, as the input for SLAB.8 and SLAB.9 needs to have been thought out (and perhaps even already generated) before the user can assign the appropriate profile number to MATTYP(KW) on SLAB.3.

When this option is used, the input variable MATTYP(KW) on SLAB.3 (for a specific node KW) has been in effect redefined for that specific node; it is no longer a flag for a single material type defined in group PROP. It is now a flag for a *profile* of axial material types defined with SLAB.8 and SLAB.9. This is an artifact of developing the option as a new feature added to the code in Cycle 2, to improve code flexibility for variable structural modeling, without adversely affecting “backward compatibility” with previous usage of the code. The modern user inherits this imbedded history of code development, and must simply accept it as *the way things are* in Cycle 4.

The input on SLAB.8 and SLAB.9 defines a given profile in terms of material properties types from input in group PROP, which are to be applied at specific axial locations, for specific slab nodes. The profile index numbers must be defined sequentially, starting from NSPROP+1 and

ending with NSPROP+NMAT. (Note that NSPROP is the total number of material properties types, defined on PROP.1.) The user must specify NMAT[SLAB.1] profiles.

SLAB.8 MAT,NSET

Read only if NMAT > 0 on SLAB.1
FORMAT (2I5)

Columns	Variable	Description
1-5	MAT	Axially varying material type profile identification number. Used in SLAB.3 to identify axially varying material type profile for specific slab nodes. The first profile must be identified by index number NSPROP+1; profile numbers must be incremented sequentially, to NSPROP+NMAT. (NOTE: NSPROP is defined by user input on PROP.1.)
6-10	NSET	Number of regions in this profile with different material properties.

SLAB.9 (K1,K2,MATJ,N= 1,NSET[SLAB.8])

Read only if NMAT > 0 on SLAB.1
FORMAT(4(3I5))

Columns	Variable	Description
1-5, 16-20 46-50	K1	Axial node number at the beginning of the N^{th} region of axially varying material profile MAT. (NOTE: The first entry for K1 must be node 2. This input is based on axial node numbering, not axial level numbering. See discussion of node and level numbering conventions in the numerical discretization, in COBRA-SFS – Theory Manual Section 2.0 (Richmond et. al. 2021).)
6-10, 21-25, 36-40,	K2	Number of axial nodes in the N^{th} region of profile MAT. (NOTE: The total number of nodes for all NSET regions in profile MAT must sum to the total number of axial nodes, NDX[CHAN.1].)
11-15, 26-30, 41-45, 56-60	MATJ	Identification number of material type for the N^{th} region of profile MAT. (NOTE: This must correspond to a material type defined by input for PROP.3.)

Four sets of entries (K1,K2,MATJ) are read on each line of SLAB.9. This line is repeated until all NSET regions are read for profile MAT.

SLAB.8 and SLAB.9 are read as a group NMAT[SLAB.1] times.

2.4.7 Group VIEW–View factor calculation

NOTE: The number of assembly types is limited to 6 in COBRA-SFS Versionversion 6.0 and forward.

2.4.7.1 View Factors for Thermal Radiation Modeling: RADGEN Code

The RADGEN code is an auxiliary code that can be used to generate the grey body view factors for thermal radiation heat transfer within an enclosure modeled as an assembly in COBRA-SFS. RADGEN is documented in detail elsewhere (Rector 1987). It is not the purpose of this section to give a complete description of this auxiliary code. However, it is virtually impossible to run any significant spent fuel storage analysis problem with COBRA-SFS without using view factors of the sort generated with the RADGEN code. Thermal radiation is typically a major mode of heat transfer in dry storage systems, particularly near the beginning of dry storage, when decay heat levels are highest for a given system. The RADGEN code has been developed in parallel with COBRA-SFS, to provide input defining view factors[1] that are appropriate for the system being modeled.

This section presents a general description of the RADGEN code in COBRA-SFS – Theory Manual Section 7.0 (Richmond et. al. 2021) including a description of the fuel rod arrangements and enclosure geometries for which this code can determine the grey body view factors. Section 2.4.7.2 gives the input instructions for the RADGEN code.

2.4.7.2 RADGEN/VIEW Input Instructions

The RADGEN code was originally designed to accept input in interactive mode, but also has the option to read input from an external file. The RADGEN input convention in Version 6 has been fully integrated into the main input file and is replaced by the VIEW input group. The ability to read tape10 files generated via RADGEN externally to the COBRA-SFS solution algorithm is retained in Version 6. The input sequence for generating view factors via this method is identical to that of the VIEW input group. The interactive mode has been deactivated in Cycle 4, but has been retrained as a feature in the code, in case it may be desirable at some point to go back to this option. The approach of requiring an input file has been adopted for Cycle 4 mainly as an aid to quality assurance tracking and verification of code applications, particularly for use within the UNF-ST&DARDS computational framework.

If generating view factors external to the main COBRA-SFS solution algorithm, the input to the RADGEN code must be specified on a local file named *input*, defined in the code as corresponding to logical unit 9. As with the COBRA-SFS input, the RADGEN input is formatted,

[1] The user has the option of using other methods to obtain view factors for thermal radiation modeling. However, the view factor arrays obtained by any such independent methods must be translated into the input conventions of the COBRA-SFS code, as described in input group RADG (see Section 2.4.8).

as described in the line-by-line input instructions presented in this section for the VIEW input group.

The major portion of the input to VIEW consists of describing the geometry of the surfaces for which view factors are to be calculated. The user must also be aware of the numbering convention assumed in VIEW for the rods of the array and wall nodes of the enclosure. These numbers are “place-holder” numbers, rather than actual rod and wall node numbers in the COBRA-SFS model, but the ordering must be maintained in the COBRA-SFS model input, so it is important for the user to understand this correspondence. (It is needed primarily for input in group RADG; see Section 2.4.8.) The numbering convention assumed in VIEW is illustrated with simple examples in Figure 2.11. For an illustration of the rod numbering convention for square array assemblies with large guide tubes, such as CE 14x14 assemblies, refer to the rod number mapping in Figure 2.15. For arrays of this geometry within a rectangular enclosure, the wall node mapping is the same as illustrated in Figure 2.11 (part (a)) for square array assemblies.

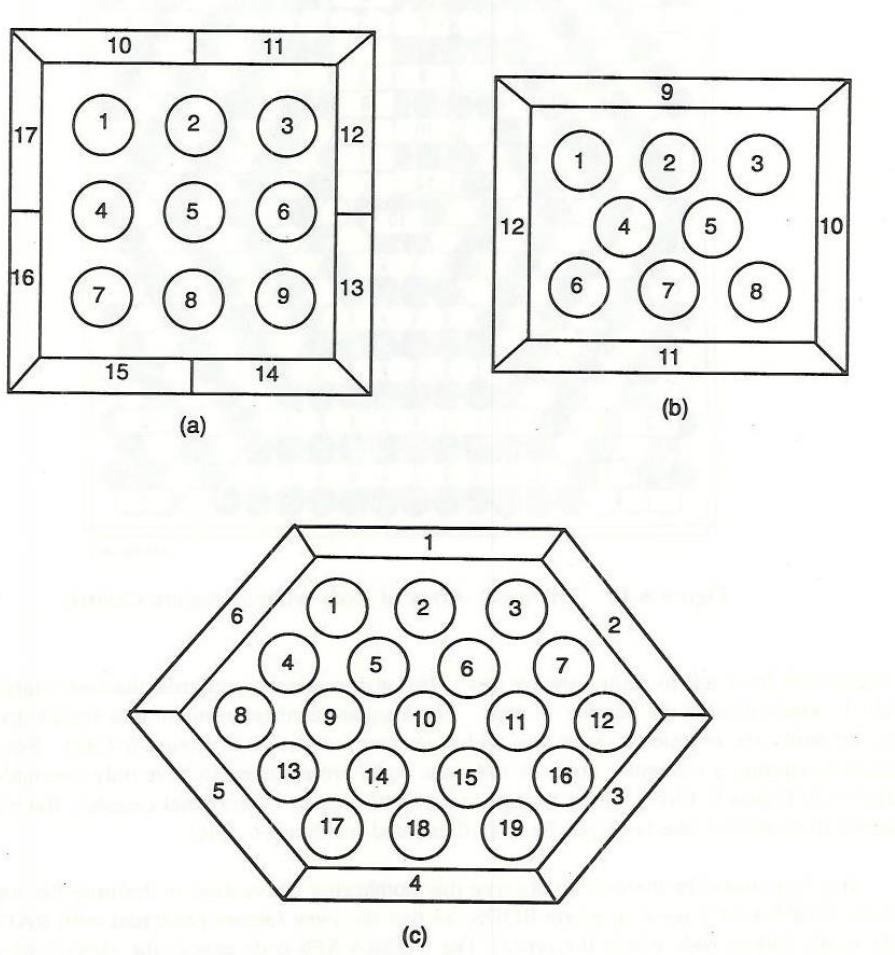


Figure 2.12. Illustration of RADGEN Rod and Wall Node Numbering Convention for Square Array and Triangular Array Assemblies

It is important for the user to be aware that this is the numbering scheme for the rods and wall nodes that has been used to create the tape10 file. The tape10 file provides the view factor input to the COBRA-SFS model, usually for more than one assembly. The group RADG input

(input line RADG.10) that maps the actual slab node numbers in the model representing the enclosure walls for assemblies that use the tape10 input ***must appropriately correspond to this numbering convention***. The COBRA-SFS code checks the view factor input from tape10 for each assembly relative to the geometry input for the slab nodes identified as defining the walls of the enclosure for that assembly (see input line RADG.10), to verify that the view factors sum appropriately within that assembly. Error messages from COBRA-SFS relating to improper view factors will in many cases be due to inconsistencies in the mapping of this VIEW numbering convention onto the actual slab node numbers in the COBRA-SFS input.

VIEW can also be used to generate view factors for an enclosure that does not contain rods. In such case, the input is greatly simplified, compared to that required for enclosures containing rod arrays. The required input simply describes the geometry of the enclosure, by specifying the number of wall surfaces, the emissivities of those surfaces, and their geometric relationship to each other. The enclosure can be of virtually any shape; the only restriction is that all wall surfaces must be flat or concave. That is, a wall surface cannot shadow itself or other surfaces from the line-of-sight view of any other surface. (NOTE: It is possible to model open enclosures that include corners or convex surfaces, but the view factors cannot be calculated using VIEW. They must be calculated by the user, and supplied to the COBRA-SFS input stream using the special input option for user-specified black body view factors in group RADG of the COBRA-SFS input. See Section 2.4.8.)

The input required to define the geometry of an open enclosure so that VIEW can calculate the view factors is relatively simple. The user must supply the (x,y) coordinates of the endpoints of each wall surface, relative to a Cartesian grid. This grid can be imposed in any manner the user finds convenient for the geometry of the enclosure, since the (x,y) points are meaningful only in relation to each other. The simplest approach is to place the left endpoint of surface #1 at the origin, and define all other node endpoints relative to that origin. Figure 2.12 illustrates this with an enclosure consisting of both flat and curved surfaces. As with the wall node numbering illustrated in Figure 2.11 for assemblies containing rods, the arbitrary wall node numbers of an enclosure required for the VIEW input must be mapped with geometric consistency onto actual slab node numbers for the enclosure as represented in the COBRA-SFS input, in group RADG.

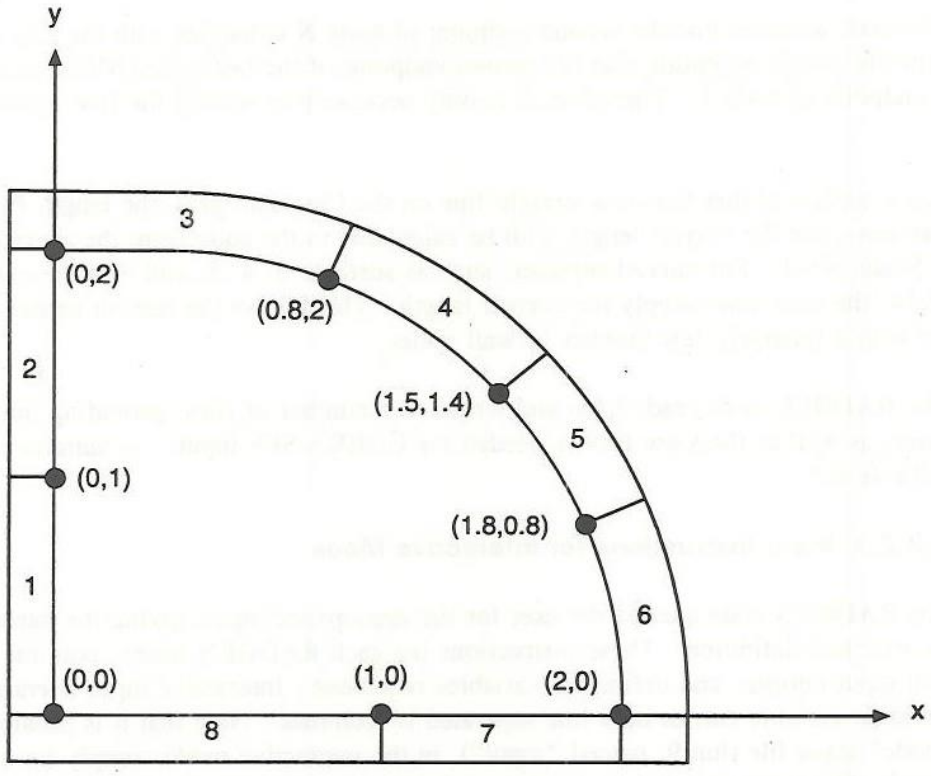


Figure 2.13. Illustration of Node Numbering Convention and Node (x,y) Endpoints on a Cartesian Grid for RADGEN Calculation of View Factors in an Open Enclosure

VIEW.1 AGROUP, NASS

FORMAT(A4,1X,1315)

Columns	Variable	Description
1-4	AGROUP	Enter VIEW
5-10	NASS	Number of assembly types for which view factors will be generated (limit 6)

VIEW.2 TEXT

Format (16A4)

Columns	Variable	Description
1-64	TEXT	Title for RADGEN output identification.

VIEW.3 IGEOM, PDR, NROWS, NCOL, IFIRST

Format (I5,F5.3,3I5)

Columns	Variable	Description
1-5	IGEOM	Rod array geometry type flag: = 0 – rods on square pitch, in rectangular enclosure = 1 – rods on triangular pitch, in rectangular enclosure = 2 – rods on triangular pitch, in hexagonal enclosure = 3 – enclosure without rods.
6-10	PDR	rod array pitch-to-diameter ratio
11-15	NROWS	If IGEOM = 0, number of rod rows in the square array If IGEOM = 1, number of rods in shortest row of the triangular pitch array within the rectangular enclosure If IGEOM = 2, number of rods in the line immediately adjacent to a wall, for triangular pitch in a hexagonal enclosure If IGEOM = 3, this input is not used; enter 0 or blank
16-20	NCOL	If IGEOM = 0, number of rod columns in the square array If IGEOM = 1, number of rods in longest row of the triangular pitch array within the rectangular enclosure If IGEOM > 1, this input is not used; enter 0 or blank
21-25	IFIRST	If IGEOM = 1, (for rods on a triangular array in a rectangular canister), enter IFIRST = 1, if top row is a long row IFIRST = -1, if top row is a short row If IGEOM ≠ 1, this variable is not used; enter 0 or blank

VIEW.4 DIA, (XR(I), I=1,4)

Read only if IGEOM < 2 and NCOL > 0 or IGEOM = 2 on VIEW.3
Format (5F5.0)

Columns	Variable	Description
1-5	DIA	Enter rod outside diameter, to specify array of uniform rods (all rods assumed to have same diameter) for this assembly type.

		Enter 0.0 to specify that the COBRA-SFS model of this assembly type includes rods of two different diameters, as specified in input group RODS (see Section 2.4.5). When DIA is entered as zero, additional input is required; see RADGEN.4a, RADGEN.4b, and RADGEN.4c). (NOTE: The standard approach in COBRA-SFS modeling is to assume that all rods in an assembly have the same diameter (as specified on input line RODS.3). The option for two different diameters is a new optional feature of Cycle 4.)
6-10	XR(1)	Clearance distance (inches) between top row of rods and enclosure wall (Clearance distances are defined assuming the user has a cross-sectional diagram of the rod array with rods and wall nodes numbered as illustrated in Figure 2.15.) (NOTE: For IGEOM = 2, option for non-uniform wall gaps is not available for a triangular array of rods in a hexagonal enclosure (i.e., the array is assumed to be centered within the enclosure); the value entered for XR(1) is used for all six wall gaps.)
11-15	XR(2)	Clearance distance (inches) between the right-side column of rods and the enclosure wall. (NOTE: For IGEOM = 1, (triangular array in rectangular enclosure), this distance is the clearance for the long rows of rods; the code calculates the appropriate clearance for the short rows automatically.)
16-20	XR(3)	Clearance distance (inches) between the bottom row of rods and enclosure wall.
21-25	XR(4)	Clearance distance (inches) between the left-side column of rods and enclosure wall. (NOTE: For IGEOM = 1, (triangular array in rectangular enclosure), this distance is the clearance for the long rows of rods; the code calculates the appropriate clearance for the short rows automatically.)

Input Conventions for Allowing Two Different Rod Diameters in an Assembly

This feature has been developed to allow consideration of the effect that the difference in size of the guide tubes within an array (compared to the fuel rod diameter) could have on thermal radiation heat transfer within the rod array. In general, this is expected to be a relatively small effect, as demonstrated by the ability of the COBRA-SFS code to match experimental data with the standard modeling simplification of all rods within an array (fuel rods, guide tubes, and instrumentation tubes) assumed to be at the nominal diameter of the fuel rods. (See discussion of COBRA-SFS validation history in COBRA-SFS – Theory Manual Section 7.0 (Richmond et. al. 2021), and detailed comparisons with test data in COBRA-SFS – Theory Manual Section 7.6 and Section 7.9 (Richmond et. al. 2021).)

For most applications of the COBRA-SFS code, it seems to be sufficient to appropriately capture the effect of guide tubes on the assembly decay heat distribution (with zero decay heat for the “rods” at the locations of the guide tubes), and capture the effect on hydrodynamic behavior with appropriate variation in local subchannel geometry (i.e., with accurate flow area, wetted perimeter, and heated perimeter for the subchannels surrounding the guide tube locations (see input for CHAN.7)). The inherent simplification in radiation view factor reflections in the vicinity of the guide tubes due to treating them as having the same diameter as the fuel rods is not likely to have a strong effect on heat transfer due to thermal radiation, in large part because the guide tube locations are relatively deep within the rod array, in regions where rod-to-rod temperature gradients are generally not large.

However, due to increasing interest in obtaining detailed temperature histories of spent fuel over the lifetime in dry storage, and the expectation that some fuel might be moved to dry storage at higher decay heat loads than heretofore considered, the new feature allowing more accurate calculation of thermal radiation within the rod array was deemed a potentially useful development. At a bare minimum, this new feature could provide an additional check on the uncertainties associated with thermal modeling of spent fuel in long-term dry storage.

This new feature in the modeling flexibility of the COBRA-SFS package has been implemented in a way that preserves backward compatibility with earlier versions of VIEW/RADGEN and COBRA-SFS (i.e., Cycle 3 and the initial release of Cycle 4 in 2015.) New input lines VIEW.4a, VIEW.4b and VIEW.4c are required when this feature is utilized within a COBRA-SFS model. The specific instructions for these new input lines are presented here. Illustrative examples of this input are presented, following the instructions for VIEW.4c.

VIEW.4a DIA1, DIA2

Read only if DIA = 0.0 on VIEW.4.
Format (2F5.0)

Columns	Variable	Description
1-5	DIA1	Rod outside diameter for fuel rods in array (all fuel rods assumed to have the same diameter). (NOTE: This includes burnable poison rods and “dummy” rods, which are assumed to be the same diameter as the fuel pins in the array.)
6-10	DIA2	Rod outside diameter for control rod guide tubes in array (all guide tubes assumed to have the same diameter). SPECIAL NOTE: This feature has been tested for WE-type control rods, and for CE-type control rods. This feature has not yet been tested with BWR fuel assembly designs with oversized water rods or the atrium designs. (NOTE: Instrumentation guide tubes are assumed to be the same diameter as control rod guide tubes, if the array is such that the control rods each displace only one rod location in the array (e.g., WE fuel assemblies). For arrays where large guide tubes displace more than 1 array position (e.g., CE assemblies), instrumentation guide tubes (if present in the array) must be treated as having the same diameter as the fuel pins (that is, DIA1).)

VIEW.4b NRODX

Read only if DIA = 0.0 on VIEW.4.
Format (I5)

Columns	Variable	Description
1-5	NRODX	Number of rods in array with rod outside diameter DIA2.

VIEW.4c NR(I), I=1,NRODX

Read only if DIA = 0.0 on VIEW.4.
Format (16I5)

Columns	Variable	Description
1-5, 11-15, …, 76-80	NR(I)	Index number of i^{th} rod with diameter DIA2. SPECIAL NOTE: For assembly configurations in which the control rods each displace a single rod in the array, without disturbing the regularity of the array (e.g., WE-type assemblies), the index numbering captured in input variable NR(I) is the same as the rod array numbering in the model. For assembly configurations in which the control rods displace a "block" of fuel rods (e.g., CE-type assemblies), this input is somewhat more complicated, to account for the disruption of the rod array. See the following illustrative examples, showing appropriate NR(I) input for both types of assemblies.

VIEW.4c is read with up to 16 elements per input line. If NRODX > 16, repeat this input line until the required number of rods with diameter DIA2 have been identified.

Example of NR(I) input for WE-type assembly:

As noted above, the index numbering for input array NR should correspond to the normal rod index numbering for assemblies of this type. Figure 2.14 illustrates this with a diagram showing the rod index numbering for a typical WE14x14 assembly. (SPECIAL NOTE: This diagram is only an illustration of the rod numbering scheme within the array. It is not a mesh diagram). In this diagram, the array locations of the guide tubes are shown with grey blocks. In the group RODS.3 input for assemblies of this type, variable DR(I) for these rods must be specified with the same value as that entered for DIA2 in the VIEW input, on line VIEW.4a.

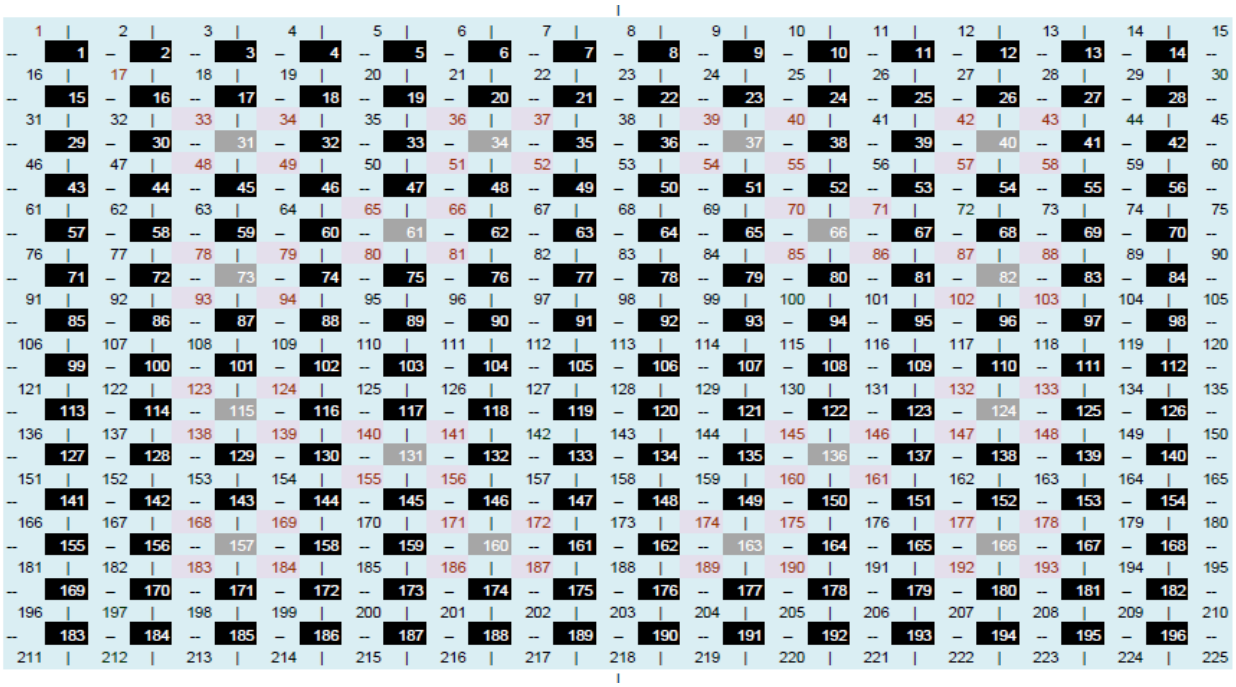


Figure 2.14. Diagram Illustrating Rod and Subchannel Numbering Schemes in COBRA-SFS Model of WE14x14 Fuel Assembly

In Figure 2.13, the locations of the guide tubes (with diameter DIA2) in this fuel assembly are indicated by grey blocks. As a gentle reminder of the interconnectedness of much of the geometry input in a COBRA-SFS model, the locations of subchannels that are affected by the difference in diameter between the fuel rods (DIA1) and the guide tubes (DIA2) are also indicated on this diagram. The index numbers of these subchannels are highlighted in pale violet. Appropriate adjustment of the geometry input for the affected subchannels should be specified in group CHAN, input line CHAN.7. This applies to the conventional approach of using a single diameter for all fuel rods, as well as to the new option of specifying DIA1 for the fuel rods and DIA2 for the guide tube rods. However, there is no automatic check in the code for this geometry adjustment, and it is not a fatal error to neglect it.

The input for VIEW.4c, array NC(I), for the assembly type illustrated in Figure 2.13 is as follows;

```
  31   34   37   40   61   66   73   82   115   124   131   136   157   160   163   166
```

For this example, NRODX = 16 (on VIEW.4b). Note that this input must be specified using the 16I5 formatting defined for input line VIEW.4c.

Example of NR(I) input for CE-type assembly:

The diagram shown in Figure 2.14 illustrates the rod array and subchannel numbering scheme expected in COBRA-SFS for assemblies with large guide tubes that displace a block of four fuel rods within the array, utilizing the new feature allowing two different rod diameters for thermal radiation within the rod array. This diagram is for a representative CE14x14 fuel assembly. In the COBRA-SFS input for group RODS, the fuel rods must be represented with diameter (DR(I)[RODS.3]) that is the same value as DIA1[VIEW.4a], and the guide tube “rods” must be represented with diameter (DR(I)[RODS.3]) that is the same value as DIA2[VIEW.4a], in order

to be compatible with the tape10 generated with this new option in VIEW. As noted for the WE14x14 diagram in Figure 2.13, the diagram for the CE14x14 assembly in Figure 2.14 highlights in pale violet the subchannels with geometry input (see line CHAN.7) affected by the larger diameter of the guide tubes. This geometry adjustment applies to the conventional approach of using a single diameter for all fuel rods, as well as the new option of specifying DIA1 for the fuel rods and DIA2 for the guide tube rods. Note also that in either approach, the subchannel at the center of the 4-rod block displaced by the guide tube must be omitted from the geometry input for the assembly, as specified in group CHAN.

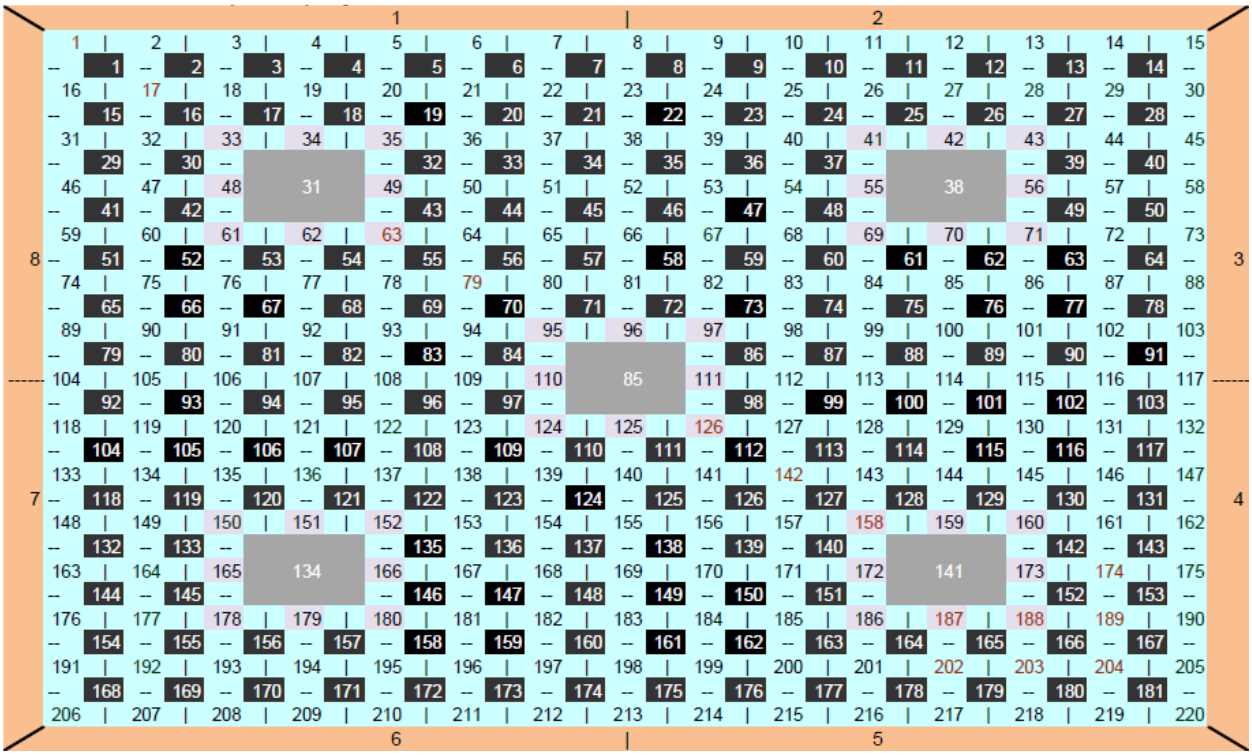


Figure 2.15. Diagram Illustrating Rod and Subchannel Numbering Schemes in COBRA-SFS Model of CE14x14 Fuel Assembly, when Modeling the Array with the Fuel Rods and the Guide Tubes

The input required for VIEW.4c, however, is not based on the numbering scheme shown for the rods in Figure 2.14. Since this new feature is built on the original assumption of an uninterrupted rod array, and to assure backward compatibility of the code, the input for VIEW.4c must reflect back to the standard numbering scheme for a rod array with only a single rod diameter (i.e., DIA1) represented. Figure 2.15 shows this diagram, which corresponds to the original approach to modeling CE-type assemblies in COBRA-SFS. In this standard approach with a single rod diameter for all rods in the array, a guide tube is represented by a block of four rods with diameter DIA1 and zero decay heat generation for the affected rods, in their respective RODS.3 input lines in group RODS. The appropriate geometry adjustments for the surrounding subchannels (including elimination of the subchannel at the center of the 4-rod block) should be made in the input for CHAN.7 in group CHAN, regardless of which option for the rod geometry and radiation modeling is utilized in a particular model.

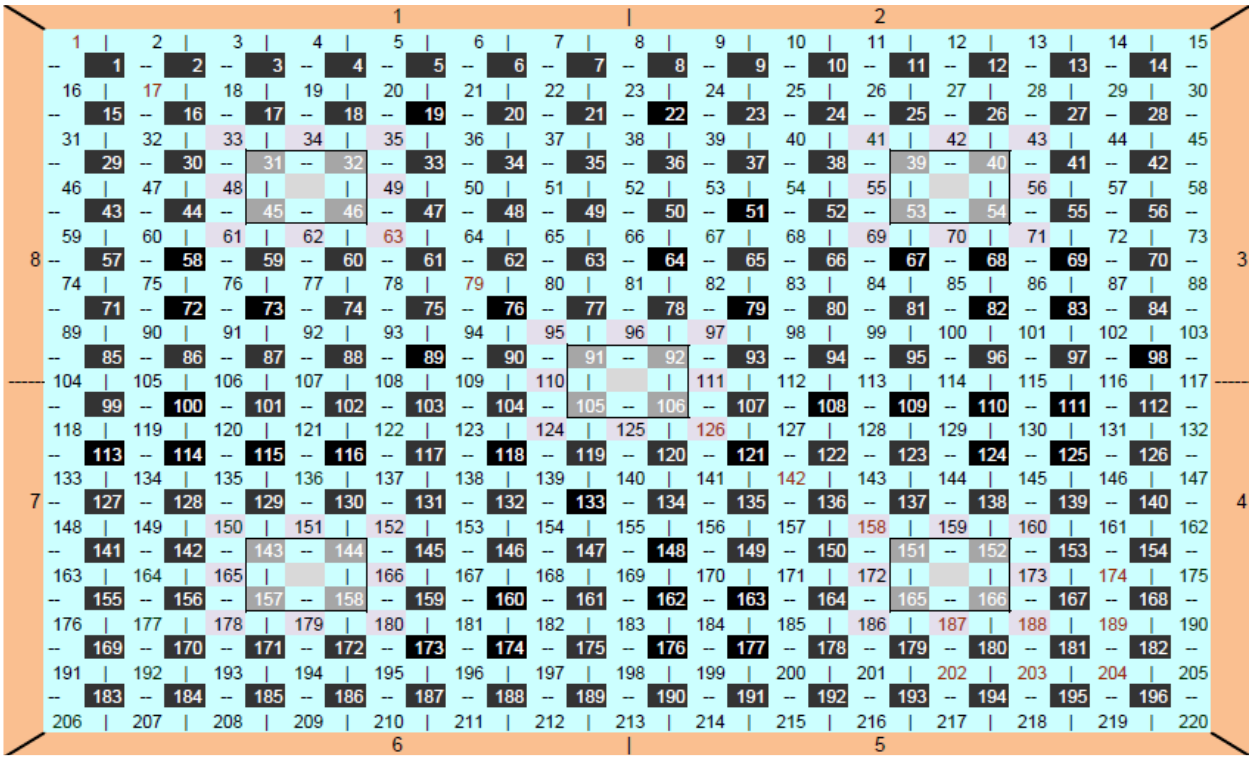


Figure 2.16. Diagram Illustrating Rod and Subchannel Numbering Schemes in COBRA-SFS Model of CE14x14 Fuel Assembly, when Modeling the Array with Only a Single Rod Diameter

The input required for the NR(I) array of VIEW.4c for the CE14x14 assembly type must be drawn from the rod numbering scheme in Figure 2.14, based on the nominal undisturbed array, rather than the actual rod numbering scheme being used in the COBRA-SFS model of the assembly type, which is illustrated in Figure 2.15. The input for VIEW.4c consists of identifying the index number of the first rod in each of the 4-rod blocks that are displaced by a guide tube, *as these numbers appear in the original undisturbed array*.

For this illustrative example, the required input for NR(I) on VIEW.4c is as follows;

```
31   39   91   143   151
```

For this example, NRODX = 5 (on VIEW.4b). Note that the NC(I) array must be specified using the 16I5 formatting defined for input line VIEW.4c.

Preliminary results obtained in testing this feature for CE14x14 and CE16x16 arrays with relatively low decay heat loads (e.g., ~50% of design basis for the particular canisters evaluated), the difference in peak cladding temperature using these two options for representing thermal radiation was determined to be on the order of one degree Fahrenheit for the specific test cases. This does not constitute a definitive study of the sensitivity of peak rod temperature or radial temperature distribution to this refinement in the modeling of thermal radiation in assemblies with large guide tubes, but it does suggest that the simpler single-rod-diameter modeling for an array, which has been used in typical applications of the COBRA-SFS code for many years, is a reasonable approach. More detailed and systematic evaluation of the

sensitivity of fuel cladding temperatures to this modeling refinement is suggested as a possible consideration for future work.

VIEW.5 JBEG(I), NLINE(I), I=1,NROWS

Read only if NCOL = 0 on VIEW.3.
Format (16I5)

Columns	Variable	Description
1-5, 11-15,…, 71-75	JBEG(I)	Index number of first rod in the i^{th} row. (NOTE: Must begin with rod #1, as shown in example rod and wall numbering patterns in Figure 2.13.)
6-10, 16-20,…, 76-80	NLINE(I)	Number of rods in the i^{th} row.

Input line VIEW.5 is read with 8 pairs of (JBEG,LNINE) entries per line, until NROWS of pairs have been specified. The last line may contain fewer than eight pairs, if NROWS is not an even multiple of 8.

VIEW.6 NWALL

Read only if IGEOM = 3 on VIEW.3.
Format (I5)

Columns	Variable	Description
1-5	NWALL	Number of wall surfaces in the enclosure. = 8 for square array of rods in rectangular enclosure = 4 for triangular array of rods in rectangular enclosure = 6 for triangular array of rods in hexagonal enclosure

VIEW.7 X(I), Y(I), PERIM(I), I=1, NWALL

Read only if IGEOM = 3 on VIEW.3.
Format (3F5.0)

Columns	Variable	Description
1-5	X(I)	Left end-point coordinate for the i^{th} surface on Cartesian grid defined for the enclosure (See example in Figure 2.12).
6-10	Y(I)	Right end-point coordinate for the i^{th} surface on Cartesian grid defined for the enclosure. SPECIAL NOTE: The Y(NWALL) endpoint must coincide with the X(1) endpoint. The enclosure must fully close.
11-15	PERIM(I)	Length of i^{th} surface facing the enclosure. (NOTE: If surface I lies on a straight line on the Cartesian grid, PERIM(I) can be entered as 0.0, and the code will automatically calculate the correct length assuming linear geometry. For curved surfaces, however, the user must supply the correct value for the actual length of surface I.)

Input line VIEW.7 is read NWALL times, for each wall surface node, in ascending order from 1 to NWALL.

VIEW.8 EMROD, EMWALL

Format (2F5.0)

Columns	Variable	Description
1-5	EMROD	Emissivity of rod surfaces (NOTE: All rods assumed to have same emissivity.)
6-10	EMWALL	Emissivity of enclosure wall surfaces (all surfaces assumed to have same emissivity) Enter 0.0 for EMWALL to specify non-uniform wall node emissivities (on RADGEN.9)

VIEW.8a NSID, NSET, NSEG, NSLAB

Format (4F5.0)

Columns	Variable	Description
1-5	NSID	Number of wall nodes in enclosure (NOTE: Must be 8 in Cycle 4 and 4a, for rectangular enclosure containing square array fuel assembly.)
6-10	NSET	NSET, NSEG, and NSLAB not used currently; placeholder input variables for extension of option to multiple surfaces on a single rod.
11-15	NSEG	
16-20	NSLAB	

VIEW.9 EMW(I), I=1, NWALL

Format (8F5.0)

Columns	Variable	Description
1-5, 6-10,…, 36-40	EMW(I)	Emissivity of i^{th} wall node. (NOTE: Must begin with wall node #1, following example rod and wall numbering patterns in Figure 2.15.)

The input for VIEW.9 is redundant if EMWALL is non-zero on VIEW.8, but is nevertheless required by the code as currently formulated, to assure the ability to represent non-uniform wall emissivities.

If NASS > 1 (on VIEW.1), repeat the whole block of input (as appropriate to the fuel assembly type being modeled) until all NASS types have been defined.

2.4.8 Group RADG–Radiative Heat Transfer Exchange Factors

This group describes the exchange factors for radiative heat transfer within assemblies. Exchange factors are dimensionless geometry parameters that define how well a given surface can “see” another surface for radiative heat transfer. The radiative heat transfer exchange factor groups are specified by type, where a type is defined by the geometry and emissivities of the surfaces in the assembly. A radiative heat transfer exchange factor group type consists of an array of grey body view factors for every surface in the assembly as it exchanges thermal energy by radiation with every other surface in the assembly. The emissivities of the surfaces are also included in the definition of the group type.

There are two ways the user can define radiative heat transfer exchange factor group types for a given case. The grey body view factors and emissivities for some or all groups can be read from a separate input file, on logical unit 10 (opened with the file name *tape10*). The tape10 file is generated by the auxiliary program VIEW/RADGEN, and must be consistent with the

geometry input specified in groups RODS and CHAN for the model. Section 2.4.7.1 discusses the capabilities of the Cycle 4 version of the VIEW/RADGEN code, and includes detailed input instructions.

The user also has the option of specifying thermal radiation heat transfer in assemblies that do not contain rods. VIEW/RADGEN can calculate grey body view factors for some types of these "unrodded" assembly geometries, and supply this input on tape10. However, this feature is not fully generalized in VIEW/RADGEN, and the COBRA-SFS code contains an alternative option allowing the user to specify black body view factors for such assemblies. The COBRA-SFS code will automatically calculate the corresponding grey body exchange factors from the black body view factors and surface emissivity values specified by user input for the slab nodes corresponding to the walls of the particular assemblies. Both options can be used within the same model, to obtain thermal radiation exchange factors in all assemblies where this is a significant mode of heat transfer. Typically, a tape10 file generated by the VIEW/RADGEN code will be used to obtain the view factors for the assemblies containing rods, and the option for user-specified black body view factors will be used for assemblies modeling open regions of the system, such as the basket support region, which do not contain rods.

SPECIAL NOTE: When view factors generated by VIEW/RADGEN are used to determine thermal radiation heat transfer for a square assembly containing a rod array on a square pitch, the slab nodes representing the walls of the assembly must be modeled with *eight* slab nodes, two on each face of the assembly enclosure. (See COBRA-SFS – Theory Manual Section 7.0 (Richmond et. al. 2021) and Section 3.0 of this report for illustrative examples. This is another one of the many things it would be helpful for the user to be aware of before attempting to set up a new model.) For assemblies with rods on a triangular pitch within a square enclosure, the walls of the assembly must be modeled with four slab nodes, one for each face of the enclosure. Similarly, for assemblies with rods on a triangular pitch within a hexagonal enclosure, the walls of the assembly must be modeled with six slab nodes, one for each flat face of the enclosure. For assemblies where black body view factors are specified by the user, however, the walls can be modeled with any number of nodes.

A given assembly (as defined in group CHAN input) can have only one radiative heat transfer group type. However, a given solid structure node may belong to more than one group, if it has surfaces facing more than one assembly.

RADG.1 AGROUP,NASSR,NT10,NRADG
FORMAT(A4,1X,3I5)

Columns	Variable	Description
1-4	AGROUP	Enter RADG.
6-10	NASSR	Number of assemblies for which radiative heat transfer groups are to be specified.
11-15	NT10	Number of radiative heat transfer groups to be read from logical unit 10 (maximum of 6; these radiative heat transfer exchange factor groups are calculated using RADGEN, or some other appropriate ray-tracing software that can be subverted into presenting the exchange factors in the manner expected by the COBRA-SFS input for this group; see Section 2.4.7.1.)
16-20	NRADG	Number of radiative heat transfer exchange factor group types for which the user will supply black body view factors and emissivities, using RADG.2 and RADG.3.

Black Body Radiative Exchange Factor Types

This input is read only if NRADG on RADG.1 is greater than zero. It is used to define radiative heat transfer view factors for enclosures that cannot be defined using program VIEW/RADGEN; see Section 2.4.7.1.

RADG.2 NRAD,NSURF

Read only if NRADG > 0 on RADG.1
FORMAT(2I5)

Columns	Variable	Description
1-5	NRAD	Radiative heat transfer exchange factor group type identification number. (NOTE: Groups must be numbered sequentially, from 1 to NRADG)
6-10	NSURF	Number of surfaces in radiative heat transfer exchange factor group type NRAD.

Black Body Radiative Exchange Factors

For each radiative exchange factor group type NRAD specified on RADG.2, the user must describe the geometry of the enclosure, and the black body view factors for all surfaces. These factors are area-weighted and can be calculated using the Hottel crossed-string correlation method (Cox 1977), such that

$$A_n F_{nm} = A_m F_{mn} \tag{2.7}$$

where

A_n,A_m = area of surfaces n and m, respectively
F_{nm} = grey body view factor from surface n to m
F_{mn} = grey body view factor from surface m to n.

Refer to Section 2.4.7.1 for a discussion of the black body view factors, and examples of the input expected for this option in the COBRA-SFS code.

RADG.3 IS,AREAS(NR,IS),EMX(IS),(ISX,FVW(NR,IS,ISX), ISX=1,NSURF[RADG.2]),NFACES

Read only if NRADG > 0 on RADG.1.
FORMAT(I5,2F5.0,6(I5,F5.0),I5)

Columns	Variable	Description
1-5	IS	Generic index number for a surface in radiative heat transfer group type NRAD. A total of NRAD[RADG.2] surfaces must be specified for a given group type. (See Section 2.4.7.1 for examples of this input convention.)
6-10	AREAS(NR,IS)	Length (inches) of surface IS in group NRAD. (NOTE: As with the length values specified for geometry factors in group SLAB, this input length is multiplied by the axial node length, to define the local radial heat transfer area for thermal radiation exchange. See Section 2.4.7.1 for additional discussion of this input convention.)
11-15	EMX(IS)	Emissivity of surface IS in group NRAD.
16-20, 26-30, etc.	ISX	Generic index number of a surface that can exchange thermal radiation with IS in group NRAD. (NOTE: For a given surface IS, the pairs of values (ISX, FVW(NR,IS,ISX) are entered in sequence from 1 to NSURF[RADG.2].)

Columns	Variable	Description
21-25, 31-35, etc.	FVW(NR,IS, SX)	Black body view factor from surface IS to surface ISX. (NOTE: This input utilizes the “forward-looking” logic employed in other connection input in the code, and it is necessary only to enter values for view factors for connections where ISX ≥ IS. View factors for connections where ISX < IS can be entered as 0.0, and the code will automatically calculate the reflexive FVW values for these connections, based on the previous lines of input for the lower numbered surfaces, using the relationship in Eq. (2.7).)
76-80	NFACES	Total number of surfaces that can exchange energy by thermal radiation with surface IS.

Up to six pairs of values for ISX,FVW(NR,IS,ISX) can be entered on RADG.3. If NSURF[RADG.2] is greater than six, the remaining input for surface IS is read on RADG.4 before going on to the input for the next surface.

RADG.4 (ISX,FVW(NR,IS,ISX),ISX =7,NSURF[RADG.2])

Read only if NSURF > 6 on RADG.2.
FORMAT(15X,6(I5,F5.0))

Columns	Variable	Description
1-15		BLANK
16-20, 26-30, etc.	ISX	Index number of a surface that can exchange thermal radiation with IS in group NRAD.
21-25, 31-35, etc.	FVW(NR,IS, ISX)	Black body view factor from surface IS to surface ISX.

Up to six pairs of values for ISX,FVW(NR,IS,ISX) may be entered on RADG.4. RADG.4 is read as many times as necessary to define all view factors to all surfaces seen by surface IS in group NRAD.

RADG.2 and RADG.3, (with RADG.4, if required), are read

sequentially, NRADG[RADG.1] times.

SUMMARY of INPUT READ from tape10:

Input lines RADG.5 through RADG.9 are read from logical unit 10, which can be generated using the auxiliary program RADGEN or using the VIEW input group which integrates RADGEN into the main COBRA-SFS solution algorithm. These input lines are described here ***for***

information only. The user does NOT supply these lines in the normal input stream for COBRA-SFS.

RADG.5 NROWS,PDR,EMW,EMR,T10TL
Read from I/O unit 10

Read only if NT10 > 0 on RADG.1
FORMAT(I5,3E5.0,I5,16a4)

Columns	Variable	Description
1-5	NROWS	Number of rows of rods in the rod bundles modeled by this radiative heat transfer group.
6-10	PDR	Nominal pitch-to-diameter ratio for the rod bundle modeled by this radiative heat transfer group.
11-15	EMW	Emissivity of the slab nodes modeling the walls of the assembly for this radiative heat transfer group.
16-20	EMR	Emissivity of the rods for this radiative heat transfer group.
21-84	T10TL	Title for tape10 file content. Optional; provided for user convenience.

RADG.6 MMl,MM2,MM3,MM4,IRATIO

Read from I/O unit 10, only if NT10 > 0 on RADG.1.
FORMAT(5I5)

Columns	Variable	Description
1-5	MM1	Total number of surfaces in radiative heat transfer group MM2.
6-10	MM2	Identification number for this radiative heat transfer group. (NOTE: 1 ≤ MM2 ≤6)
11-15	MM3	Flag for number of surfaces on a rod. (NOTE: Must correspond to the value for NTHETA entered on RODS.1).
16-20	MM4	Number of wall surfaces in this radiative heat transfer group (must be 8 for a square-array unconsolidated rod bundle and 4 for consolidated rod bundle with triangular pitch in a square enclosure).
21-25	IRATIO	Flag for reading surface perimeter ratios on RADG.8; = 0; read only wall surface ratios = 1; read all surface ratios (defined when tape10 includes option for more than a single rod diameter)

RADG.7 (EMV(MM2,N),N=1,MM4)

Read from I/O unit 10, only if NT10 > 0 on RADG.1.
FORMAT(8E15.0)

Columns	Variable	Description
1-15, 16-30, 21-45, etc.	EMV(MM2,N)	Emissivity of the N^{th} slab node enclosing an assembly with radiative heat transfer group MM2. (NOTE: Default is EMW[RADG.5] for all wall nodes; this line is non-zero only if option for variable emissivities is used when creating the tape10 file)

RADG.8 (RRATIO(MM2,N),N=1,MM4 or MM1)

Read from I/O unit 10, only if NT10 > 0 on RADG.1.
FORMAT(8E15.0)

Columns	Variable	Description
1-15, 16-30, 31-45, etc.	RRATIO (MM2,N)	Ratio of the perimeter of the N^{th} node of the assembly wall to the rod perimeter, for radiative heat transfer group MM2. (NOTE: If IRATIO[RADG.6] = 0, the RRATIO array in the tape10 file is the ratio of the perimeter of the N^{th} node of the assembly wall to the rod perimeter. If IRATIO[RADG.6] = 1, the RRATIO array in the tape10 file is the ratio of the perimeter of the N^{th} radiation surface (including rods and wall nodes) to the reference rod perimeter.)

RADG.9 (FVn(N,M),M= 1,MM1),N= 1,MM1)

Read from I/O unit 10 only if NT10 > 0 on RADG.1.
FORMAT(8E15.5)

Columns	Variable	Description
1-15, 16-30, 31-45, etc.	FVn(N,M)	Grey body exchange factor from surface N to surface M for radiative heat transfer group MM2.

RADG.9 is read for MM1 sets of exchange factors.

RADG.6 through RADG.9 are read sequentially NT10 times from logical unit 10. This is the end of the input read from logical unit 10.

END of INPUT from Logical Unit 10.

User must continue defining input for group RADG in the normal input stream, with line RADG.10.

Assembly Radiative Heat Transfer Group Information

The generic surface indexing defined for a radiation group type (as defined by input for RADG.3 for assemblies without rods, or by input from tape10), is assigned to specific nodes in specific assemblies by means of the input on line RADG.10. This is another one of those pieces of the input when a detailed node map of the slab network layout is more or less indispensable. See COBRA-SFS – Theory Manual Section 7.0 (Richmond et. al. 2021) for examples. The wall nodes of all assemblies with radiative heat transfer must be identified with this input, whether their respective view factors are from input on RADG.3 or from tape10.

RADG.10 IASS,ITYPR(IASS),MSID(IASS),(KSIDE(IASS,L),L = 1,12)
FORMAT(3I5,12(I5))

Columns	Variable	Description
1-5	IASS	Identification number of an assembly with radiative heat transfer.
6-10	ITYPR(IASS)	Radiative heat transfer group type: • for a radiative group type read from logical unit 10, enter a positive number (must correspond to a group type identified on RADG.6). • for a radiative group calculated from user-specified black body view factors (using RADG.2 and RADG.3), enter the *negative* of the group identification number (NOTE: The absolute value must correspond to the number of a group type identified on RADG.2).
11-15	MSID(IASS)	Number of slab nodes forming the walls of assembly IASS that have surfaces participating in radiative exchange factor group ITYPR(IASS). (SPECIAL NOTE: If view factors for group ITYPR(IASS) are for a rod array defined with input from tape10, and • if assembly IASS is a rectangular enclosure containing a rod array on a square pitch, MSID(IASS) must be 8 • if assembly IASS is a rectangular enclosure containing a rod array on a triangular pitch, MSID(IASS) must be 4 • if assembly IASS is a hexagonal enclosure containing a rod array on a triangular pitch, MSID(IASS) must be 6 • if view factors for group ITYPR(IASS) are for a group type specified by input on RODS.2 and RODS.3,

		MSID(IASS) must be the same as NSURF on RODS.2 for group type ABS[ITYPR(IASS)].)
16-20, 21-25, etc.	KSIDE(IASS,L)	Identification number of the slab node corresponding to generic index number L of the corresponding surface of radiative exchange factor group ABS[ITYPR(IASS)].

Up to 12 values for KSIDE can be entered on RADG.10; if MSID(IASS) is greater than 12, the remaining input for assembly IASS is read on RADG.11 before going on to the input for the next assembly. With the special constraints noted in the instructions for RADG.10 on the permitted values for MSID(IASS) when the radiative group type is read from tape10, it is obvious that RADG.11 would be needed only for group types specified with input lines RADG.2 and RADG.3.

RADG.11 (KSIDE(IASS,L),L = 13,MSID(IASS))
Read only if MSID(IASS)[RADG.10] > 12
FORMAT(15X,12(I5))

Columns	Variable	Description
1-15		BLANK
16-20, 21-25, etc.	KSIDE(IASS,L)	Identification number of the slab node corresponding to index number L of the corresponding surface of radiative exchange factor group ABS[ITYPR(IASS)].

RADG.10 (with RADG.11 if MSID(IASS) > 12) is read NASSR[RADG.1] times.

2.4.9 Group HEAT–Heat Transfer Correlations

This group defines the heat transfer coefficient correlations for convection heat transfer between the fluid and solid structures of the model (i.e., fuel rods and slab nodes). (Refer to Section 2.4.6 for discussion of input required for conduction heat transfer between the fluid and the structures represented by slab nodes.) The user must coordinate the input in group HEAT with correlations identified for specific assemblies in the input specified in group CHAN. Specifically, the value entered for NFLMC(NASS) on CHAN.6 for each assembly must correspond to heat transfer correlation(s) defined with input on line HEAT.2, or HEAT.5, or HEAT.6. The number of specific heat transfer correlation sets to be defined in group HEAT is determined by user input for variables NHEAT, NTHEAT, and NFHEAT on HEAT.1.

The input in group HEAT also allows the user to define specific constitutive models for turbulent energy exchange between fluid flowing in adjacent channels connected by gaps. These models were developed primarily for representing liquid or two-phase coolant flow in reactor cores, reflecting the history of the COBRA series of codes. Their relevance to gas flow at the relatively low velocities encountered in spent fuel storage canisters and casks is limited. The recommended selection for this option is NSCBC = 0; that is, to neglect energy exchange by turbulent mixing between adjacent channels. Unless a particular application has some specific feature that may make these models important in gas flow, these options can generally be ignored.

The option to consider conduction in the fluid in the radial direction, from channel-to-channel through the lateral gap connections, is a feature of COBRA-SFS that can be utilized by setting the flag NFCON = 1 on HEAT.1. It is recommended that this option generally should be included in a typical cask or canister model, particularly if the backfill gas is a relatively high-conductivity gas such as helium. In many storage systems, convection heat transfer is severely limited due to low or nearly zero flow circulation velocities resulting from the geometry of the package internal structure (i.e., basket design) or because the canister is stored in a horizontal orientation. Conduction heat transfer in the fluid, by contrast, is generally unaffected by geometry, and particularly in the absence of convection, can be a significant mode of heat transfer in the radial direction.

HEAT.1 AGROUP,NHEAT,NSCBC,NFCON,NTHEAT,NFHEAT
FORMAT(A4,1X,5I5)

Columns	Variable	Description
1-4	AGROUP	Enter HEAT.
6-10	NHEAT	Number of heat transfer correlation sets to be read in on HEAT.2.
11-15	NSCBC	Flag for single-phase turbulent mixing model: = 0; no turbulent mixing (default; recommended for low velocity gas flow) = 1; turbulent mixing defined using Eq. (2.10) = 2; turbulent mixing defined using Eq. (2.11) = 3; turbulent mixing defined using Eq. (2.12) = 4; turbulent mixing defined using Eq. (2.13)
16-20	NFCON	Flag for radial conduction in the fluid: = 0; no conduction in the fluid. = 1; fluid conduction between channels through the gap connections.
21-25	NTHEAT	Number of lateral convection heat transfer correlation sets to be read in on HEAT.5.
26-30	NFHEAT	Number of axial free convection heat transfer correlations to be read in on HEAT.6.

Axial Flow Forced Convection Heat Transfer Correlations

Since spent fuel storage systems are generally required to be passively cooled systems, it might seem odd to include an option for axial flow forced convection heat transfer correlations. This feature is in part an artifact of the history of the code for thermal-hydraulic modeling of reactor cores, but it is also a useful feature for representing the fluid behavior of spent fuel storage systems with significant internal convection heat transfer, particularly vertical storage systems. Although such systems are passively cooled, they do not follow the typical flow patterns that most natural convection heat transfer correlations are derived for.

These designs rely on internal thermo-siphon recirculation of backfill gas within the canister to move heat from the rods to the canister outer shell, and external air circulation through the overpack annulus to remove heat from the canister surface and carry it out to the external environment. Rather than “natural convection”, it is more precisely correct to characterize the flow and heat transfer patterns in these types of systems as *naturally driven forced convection*, where the driving force is a gravity head due to a density difference in different parts of the system, rather than a mechanical pump. As such, heat transfer due to the resulting fluid flow can be readily characterized with standard forced convection heat transfer correlations, selected for an appropriate range of applicability (typically defined by Reynolds number) for the particular system.

Forced convection heat transfer correlations are specified in the code using the standard formula,

$$H = (a_1 Re^{a_2} Pr^{a_3} + a_4)\frac{k}{D_e} \tag{2.8}$$

where

a_1, a_2, a_3, a_4	=	empirical coefficients
Re	=	Reynolds number
Pr	=	Prandtl number
k	=	thermal conductivity of the fluid
D_e	=	channel hydraulic diameter

The coefficients of the formula in Eq. (2.8) are specified for a correlation, using the input line HEAT.2. The convention in the code is to enter heat transfer correlations in pairs, with one set of coefficients for laminar flow and one set for turbulent flow, such that Eq. (2.8) becomes

$$H_{laminar} = (a_{l1} Re^{a_{l2}} Pr^{a_{l3}} + a_{l4})\frac{k}{D_e}$$
$$H_{turbulent} = (a_{t1} Re^{a_{t.2}} Pr^{a_{t3}} + a_{t4})\frac{k}{D_e} \tag{2.9}$$

The local heat transfer coefficient for a node is evaluated as the maximum value obtained with the turbulent and laminar equations for the local Reynolds number and Prandtl number,

$$H_{axial} = \max\left(H_{laminar}, H_{turbulent}\right)$$

As illustrated in Figure 2.16, the typical coefficients based on experimental data obtained for laminar and turbulent correlations of this form yield curves that allow automatic sorting of the flow regime, without the necessity of precisely defining the laminar-to-turbulent transition. This greatly simplifies the input to the code for heat transfer correlations, and can help prevent non-physical disturbance to the numerical solution due to arbitrary and abrupt transition between the two flow regimes.

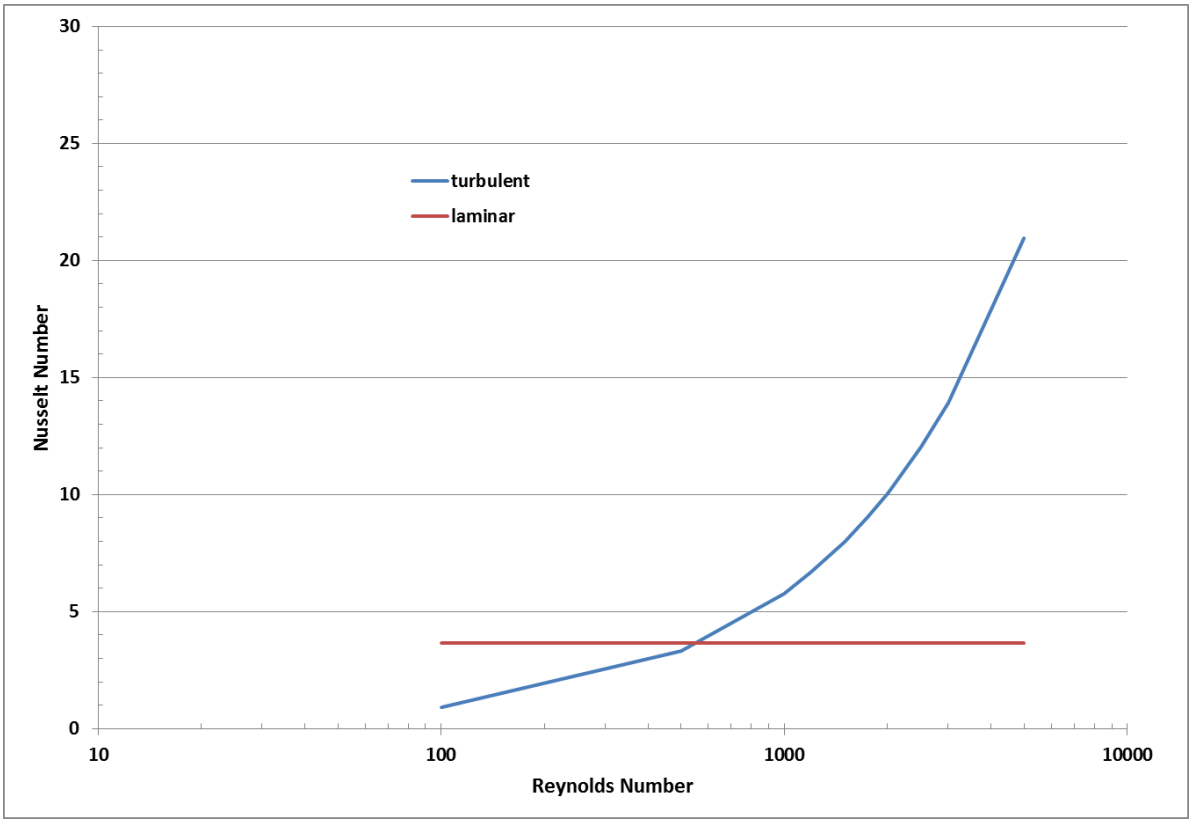


Figure 2.17. Typical Nusselt Number as a Function of Flow Regime, Based on Experimental Data

The heat transfer correlations are numbered sequentially in the code, from 1 to NHEAT, in the order they are read in on HEAT.2. If NHEAT > 1, the user must keep track of this sequence, in order to match the desired correlation with specific assemblies in the group CHAN input. Specific correlations are flagged for specific assemblies by the value entered for variable NFLMC(NASS) on CHAN.6, to identify the heat transfer correlation to be used by a given assembly (see group CHAN input, in Section 2.4.3). In the group CHAN input, each assembly is given a specific value for NFLMC, so that it is possible to assign a different heat transfer correlation to each assembly, if this is desirable for a given model. Assemblies of the same type in the group CHAN input are permitted to use different heat transfer correlations, at the user's discretion.

HEAT.2 (AH1(I),AH2(I),AH3(I),AH4(I),AHL1(I),AHL2(I),AHL3(I),AH l4(I), I = 1,NHEAT)

Read only if NHEAT > 0 on HEAT.1
FORMAT(12F5.0)

Columns	Variable	Description
1-5	AH1(I)	Reynolds number multiplier coefficient of I^{th} axial heat transfer correlation for turbulent flow (i.e., a_{tl} in Eq. [6.9]).
6-10	AH2(I)	Reynolds number exponent of I^{th} axial heat transfer correlation for turbulent flow (i.e., a_{t2} in Eq. [6.9]).
11-15	AH3(I)	Prandtl number exponent of I^{th} axial heat transfer correlation for turbulent flow (i.e., a_{t3} in Eq. [6.9]).
16-20	AH4(I)	Additive coefficient of I^{th} axial heat transfer correlation for turbulent flow (i.e., a_{t4} in Eq. [6.9]).
21-25	AHL1(I)	Reynolds number multiplier coefficient of I^{th} axial heat transfer correlation for laminar flow, (i.e., a_{fl} in Eq. [6.9]).
26-30	AHL2(I)	Reynolds number exponent of I^{th} axial heat transfer correlation for laminar flow (i.e., a_{f2} in Eq. [6.9]).
31-35	AHL3(I)	Prandtl number exponent of I^{th} axial heat transfer correlation for laminar flow (i.e., a_{f3} in Eq. [6.9]).
36-40	AHL4(I)	Additive coefficient of I^{th} axial heat transfer correlation for laminar flow (i.e., a_{f4} in Eq. [6.9]).

Subchannel Turbulent Mixing Correlations

As noted above, the relevance of local turbulent mixing between adjacent subchannels at gas flow velocities typically encountered in spent fuel storage canisters and casks is limited. The recommended selection for this option is NSCBC = 0; that is, to neglect energy exchange by turbulent mixing between adjacent channels. Unless a particular application has some specific feature that may make these models important in gas flow, these options can generally be ignored. However, in the event that these models are relevant to a particular application of the code, the following models are available, based on the value specified for NSCBC[HEAT.1].

if NSCBC = 1; W'_T = AMIX * ($S_K \bar{G}$)	Eq. (2.10)
if NSCBC = 2; W'_T = AMIX * (Re ** BMIX) * ($S_K \bar{G}$)	Eq. (2.11)
if NSCBC = 3; W'_T = AMIX * (Re ** BMIX) * ($\bar{D} \bar{G}$)	Eq. (2.12)
if NSCBC = 4; W'_T = AMIX * (Re ** BMIX) * (S_K/ℓ_K)($\bar{D} \bar{G}$)	Eq. (2.13)

where

W'_T = turbulent crossflow

Re = Reynolds number (based on axial velocity)

$\bar{G}$ = average axial mass flux of the two channels connected by lateral flow connection k

S_k = lateral width of lateral flow connection k
$\overline{D}$ = average hydraulic diameter of the two channels connected by lateral flow connection k
ℓ_K = transverse momentum control volume length for flow connection k
AMIX, BMIX = user-defined constants; see HEAT.3.

Note that S_k and ℓ_K correspond to input defining channel-to-channel lateral flow connections in group CHAN, on input line CHAN.7. The user must supply a total of MAXTYP pairs of AMIX, BMIX entries; one pair for each assembly type defined by input in group CHAN. The code calculates MAXTYP by summing the number of unique values of ITYPA[CHAN.5] supplied by the input. The user must do the same.

HEAT.3 (AMIX(N),BMIX(N),N=1,MAXTYP)

Read only if NSCBC > 0 on HEAT.1
FORMAT(2E5.0)

Columns	Variable	Description
1-5	AMIX(N)	Multiplicative coefficient for turbulent mixing correlation, of form specified by value of NSCBC on HEAT.1, to be applied to all assemblies of type N.
6-10	BMIX(N)	Reynolds number exponent for turbulent mixing correlation of form specified by value of NSCBC on HEAT.1, to be applied to all assemblies of type N. If NSCBC = 1, BMIX(N) is not used, and should be entered as zero (0.0).

if NSCBC > 0 on HEAT.1, HEAT.3 is read MAXTYP times, once for each assembly type

Radial Fluid Conduction

If NFCON = 1, radial heat conduction in the fluid will be calculated for the model. Heat conduction through the fluid is treated as occurring between adjacent channels through the gap connections. The conduction path is defined by the centroid length of the lateral control volume, which is specified by user input in variable DIST(K) on CHAN.7. The input on HEAT.4 can be used to modify the conduction length for the lateral connections in a given assembly, to account for variations in the connection lengths that may not be considered in the value entered for DIST. However, the value specified for GK(N) for a given assembly type applies to all channels of all assemblies of that type.

HEAT.4 (GK(N),N = 1,MAXTYP)

Read only if NFCON = 1 on HEAT.1.
FORMAT(16E5.0)

Columns	Variable	Description
1-5, 6-10, 11-15, etc.	GK(N)	Multiplier on the lateral control volume length for thermal conduction between channels in assemblies of type N: • if GK(N) is specified as 1.0, the length specified in DIST on CHAN.7 for a particular gap connection is used for the conduction length. • if GK(N) is specified as 0.0, radial fluid conduction is zero in assemblies of type N.

MAXTYP is the number of unique assembly types specified by user input in group CHAN, as described above for input line HEAT.3.

Up to 16 values can be read on a line for HEAT.4. If there are more than 16 assembly types, this line is repeated as many times as necessary to specify a GK(N) value for each of MAXTYP assembly types.

Lateral Flow Forced Convection Heat Transfer Correlations

This input is optional, and is needed only if NTHEAT > 0 on HEAT.1. It was originally developed for studies of natural circulation in horizontal test assemblies and storage canisters. Subsequent work has shown that for most such configurations, lateral flow is negligible, and can be omitted as a modeling simplification. The feature is retained in the code, to maintain flexibility. Users wishing to employ this model are strongly urged to perform independent validation for their specific application, including sensitivity studies.

The input for heat transfer correlations for forced convection due to lateral flow follow the same convention as the correlations for axial flow, as described above for input line HEAT.2. Coefficients are read in for a turbulent and a laminar formulation

$$H_T = \frac{k}{D_r}\left(a_{t1} {Re_D}^{a_{l2}} Pr^{a_{l3}} + a_{t4}\right)$$
$$H_L = \frac{k}{D_r}\left(a_{l1} Re^{a_{l2}} Pr^{a_{l3}} + a_{l4}\right) \quad (2.14)$$

where

a_{t1}, a_{t2}, a_{t3}, a_{t4} = empirical coefficients for turbulent correlation
a_{l1}, a_{l2}, a_{l3}, a_{l4} = empirical coefficients for laminar correlation
Re_D = Reynolds number, based on rod diameter
Pr = Prandtl number
k = thermal conductivity of the fluid
D_r = rod nominal diameter

The local heat transfer coefficient is evaluated as the maximum of the two;

$$H_{lateral} = max(H_T, H_L)$$

These heat transfer correlations are numbered sequentially in the code, from 1 to NTHEAT, in the order they are read in on HEAT.5. The heat transfer correlation to be used in a particular lateral flow connection is determined by the value specified for NFLMC(IASS) on CHAN.6 for the assembly containing the lateral flow connection. The input for NFLMC(IASS) on CHAN.6 is thus made to do double duty, since it also specifies the heat transfer correlation for axial flow in the channels of the assembly, by identifying the index of a correlation defined by input on HEAT.2.

It is expected that in a given application, either the axial or the lateral convection would be dominant, and the user would select the corresponding correlations appropriately. If the option for lateral heat transfer correlations is specified (with NTHEAT>0 on HEAT.1), the user must make certain that the index number entered on CHAN.6 in variable NFLMC(IASS) for an assembly type corresponds to the correct lateral heat transfer correlation (from HEAT.5) for the lateral connections between channels in the assembly. The user must also enter appropriate coefficients for the axial heat transfer correlation (from HEAT.2) with the corresponding index number, to assure that it will not interfere with the performance of the lateral heat transfer correlation. (The simplest way to do this is to enter coefficients that are equivalent to Nu = 1 for the corresponding axial heat transfer correlation.)

HEAT.5 (AHT1(I),AHT2(I),AHT3(I),AHT4(I),AHTL1(I),AHTL2(I),AHTL3(I), AHTL4(I),I= 1,NTHEAT[HEAT.1])

Read only if NTHEAT > 0 on HEAT.1
FORMAT(8F5.0)

Columns	Variable	Description
1-5	AHTl(I)	Reynolds number multiplier coefficient of I^{th} lateral heat transfer correlation for turbulent flow (i.e., a_{tl} in Eq. [6.14]).
6-10	AHT2(I)	Reynolds number exponent of I^{th} lateral heat transfer correlation for turbulent flow (i.e., a_{t2} in Eq. [6.14]).
11-15	AHT3(I)	Prandtl number exponent of I^{th} lateral heat transfer correlation for turbulent flow (i.e., a_{t3} Eq. [6.14]).
16-20	AHT4(I)	Additive coefficient of I^{th} lateral heat transfer correlation for turbulent flow (i.e., a_{t4} Eq. [6.14]).
21-25	AHTL1(I)	Reynolds number multiplier coefficient of I^{th} lateral heat transfer correlation for laminar flow (i.e., a_{11} in Eq. [6.14]).
26-30	AHTL2(I)	Reynolds number exponent of I^{th} lateral heat transfer correlation for laminar flow (i.e., a_{11} in Eq. [6.14]).
31-35	AHTL3(I)	Prandtl number exponent of I^{th} lateral heat transfer correlation for laminar flow (i.e., a_{13} in Eq. [6.14]).
36-40	AHTL4(I)	Additive coefficient of I^{th} lateral heat transfer correlation for laminar flow (i.e., a_{14} in Eq. [6.14]).

HEAT.5 is read NTHEAT[HEAT.1] times.

Axial Flow Free Convection Heat Transfer Correlations

Heat transfer correlations for free convection due to buoyancy-driven axial flows can be specified in the form,

$$H_f = \frac{k}{D_r}\left(a_{f1} Ra^{a_{f2}}\right) \quad (2.15)$$

where

a_{fl}, a_{f2} = empirical coefficients
Ra = local Rayleigh number of the flow, (defined as the Grashof number times the Prandtl number)
k = thermal conductivity of the fluid
Dr = rod diameter.

These heat transfer correlations are numbered sequentially in the code, from 1 to NFHEAT, in the order they are read in on HEAT.6. The heat transfer correlation to be used for free convection in a given assembly is determined by the value specified for NFLMC(IASS) on CHAN.6. As noted above for the HEAT.5 input, the user must ensure that entries for HEAT.2 and HEAT.5 do not interfere with the code's interpretation of the correct free convection heat transfer correlation (from HEAT.6).

HEAT.6 (AHF1(I),AHF2(I),I=1,NFHEAT[HEAT.1])

Read only if NFHEAT > 0
FORMAT(2F5.0)

Columns	Variable	Description
1-5	AHF1(I)	Rayleigh number multiplier coefficient of Ith axial heat transfer correlation for free convection (i.e., a_{f1} in Eq. [6.15]).
6-10	AHF2(I)	Rayleigh number exponent of Ith axial heat transfer correlation for free convection (i.e., a_{f2} in Eq. [6.15]).

HEAT.6 is read NFHEAT[HEAT.1] times

2.4.10 Group DRAG–Friction Factors and Loss Coefficients

This group describes the axial pressure losses due to wall friction and local obstructions in the flow field. The optional network model can also be specified, by setting the flag NETWK on DRAG.1 (refer to COBRA-SFS – Theory Manual Section 5.1.2.2 (Richmond et. al. 2021) for a discussion of the network model). However, the network model is unlikely to be applicable to typical models of spent fuel storage systems, and the user can generally ignore it. Similarly, the option for local axial blockages of the flow field, specified by setting flag NBLOCK on DRAG.1, is unlikely to be applicable to storage systems, except in unusual configurations, and can be safely ignored for most models.

DRAG.1 AGROUP,NFRICT,NOLC,NRAMP,NLCFF,NLCFP,NETWK,NOGRP, NBLOCK,NVISCW,NFTRAN

FORMAT(A4,1X,10I5)

Columns	Variable	Description
1-4	AGROUP	Enter DRAG.
6-10	NFRICT	Number of friction factor correlation sets to be read in (default is 1).
11-15	NOLC	Number of assembly types for which loss coefficients are to be specified.
16-20	NRAMP	Number of iterations over which the loss terms are to be ramped into the solution (default is 1).
21-25	NLCFF	Number of loss coefficient versus Reynolds number forcing functions to be defined (default is 0).
26-30	NLCFP	Number of points in the loss coefficient versus Reynolds number forcing functions.
31-35	NETWK	Flag for optional pressure drop network model: = 0; not used (recommended selection). = 1; pressure drop calculated for user-specified assembly network.
36-40	NOGRP	Number of assembly groups for the network model (used only if NETWK > 0).
41-45	NBLOCK	Number of axial locations where axial flow blockages occur. = 0; not used (recommended selection).
46-50	NVISCW	Flag for hot wall correction to viscosity in friction factor correlation: = 0; no hot wall correction. = 1; apply hot wall correction to channels with heated surfaces.
51-55	NFTRAN	Number of lateral flow fraction loss correlation sets to be read in on DRAG.13.

Friction Factor Correlations

Wall friction losses are described in COBRA-SFS using the Darcy friction factor correlation form. This is defined in terms of the pressure drop such that the axial frictional pressure gradient is given by

$$\frac{dP}{dX} = \frac{f|m|m}{2g_c D_h \rho A} \tag{2.16}$$

The friction factor, f, is determined from the Blasius relation,

$$f = aRe^b + c \tag{2.17}$$

Each friction factor correlation consists of a formulation with a set of turbulent and a set of laminar coefficients, of the form

$$f_T = aRe^b + cRe^d + e \tag{2.18}$$

$$f_L = a_L Re^{b_L} + c_L \tag{2.19}$$

The code evaluates the local friction factor as the maximum of the turbulent and laminar values,

$$f = \max(f_T, f_L)$$

A total of NFRICT[DRAG.1] correlations are specified by input on DRAG.2. These correlations are numbered sequentially in the order they are read in on DRAG.2. These identification numbers are used when specifying the values for N on CHAN.7, in the geometry description for each channel of an assembly type. The default for N is type 1, and in most cases, this will be sufficient for all channels. However, the user can specify as many friction factor correlations as needed to represent the various wall resistances in a given model.

Alternatively, instead of specifying N for each channel on CHAN.7, a friction factor correlation can be specified for the assembly type, in the value for MDFLT on CHAN.6. The value for N on CHAN.7 can then be entered as zero, and all channels in assemblies of that type will be assigned the friction factor correlation identified in MDFLT on CHAN.6. (This is by far the preferable way to assign this input, unless the individual channel-by-channel input for N on CHAN.7 is absolutely necessary for a given model.)

DRAG.2 (AA(I),BB(I),CC(I),DD(D,EE(I),AAL(I),BBL(I),CCL(I),I= 1,NFRICT[DRAG.l])

Read only if NFRICT > 0 on DRAG.1.
FORMAT(8F5.0)

Columns	Variable	Description
1-5	AA(I)	First multiplicative coefficient on the Reynolds number for turbulent flow friction factor correlation (a in Eq. [6.18]).
6-10	BB(I)	First Reynolds number exponent for turbulent flow friction factor correlation (b in Eq. [6.18]).
11-15	CC(I)	Second multiplicative coefficient on the Reynolds number for turbulent flow friction factor correlation (c in Eq. [6.18]).
16-20	DD(I)	Second Reynolds number exponent for turbulent flow friction factor correlation (d in Eq. [6.18]).
21-25	EE(I)	Additive coefficient for turbulent flow friction factor correlation (e in Eq. [6.18]).

26-30	AAL(I)	Multiplicative coefficient on the Reynolds number for laminar flow friction factor correlation (a_L in Eq. [6.19]).
31-35	BBL(I)	Reynolds number exponent for laminar flow friction factor correlation (b_L in Eq. [6.19]).
31-35	CCL(I)	Additive coefficient for laminar flow friction factor correlation (c_L in Eq. [6.19]).

*** DRAG.2 is read NFRICT[DRAG.1] times. ***

Axial Form Loss Coefficients

The pressure loss due to form drag on local obstructions in the flow field such as grid spacers and orifice plates is given by

$$\Delta P = C_D \frac{m|m|}{2 g_c \rho A^2}$$

where

C_D = loss coefficient
m = upstream mass flow rate (lbm/s).

At the low velocities typical of convection within a spent fuel canister, the effect of form losses can generally be expected to be minor. Wall friction effects are more likely to dominate the momentum conservation solution. This is a fortunate circumstance, as there is usually a dearth of information on loss coefficients for structures in the system (such as grid spacers for proprietary fuel assembly designs). Sensitivity studies can be readily performed to assess the effect of uncertainty in loss coefficient values on predicted temperatures, and in most cases will show that a very large variation is required to produce a noticeable change. In general, it can be fairly simple to show that reasonable engineering values based on standard handbook formulas for specific structural configurations (e.g., orifices, orifice plates, and large area changes) are adequate for most applications.

Axial form loss coefficients can be specified for any channel or group of channels, at any axial location in the model. Local loss coefficients are specified by assembly type. Assemblies of the same type see the same form losses. In each assembly type having local losses, the user must define the axial levels and channels where the losses occur, and specify the loss coefficient to be applied at each location. This input is specified on DRAG.3 and DRAG.4. These two input records are read in sequence NOLC[DRAG.1] times.

DRAG.3 LCASST(I),NB1(I),LCFF(I)

Read only if NOLC > 0 on DRAG.1.
FORMAT(3I5)

Columns	Variable	Description
1-5	LCASST(I)	Identification number of an assembly type with loss coefficients.
6-10	NB1(I)	Number of sets of channels in assembly type LCASST(I) that have loss coefficients.

		(NOTE: A channel set is defined as a group of sequentially numbered channels that have the same loss coefficient(s) at the same axial location(s).)
11-15	LCFF(I)	Identification number of loss coefficient forcing function to be applied to loss coefficients in assembly type LCASST(I) (optional input; must correspond to a forcing function entered on DRAG.6).

DRAG.4 (with DRAG.5, if needed) is read for L=1,NB1(I) sets of channels with local losses in assemblies of type LCASST(I).

DRAG.4 (NLEVEL(I,L),ILCS(I,L),ILCE(I,L),(FACTOR(I,L,J),XCD(I,L,J),J = 1,6)

Read only if NOLC > 0 on DRAG.1.
FORMAT(3I5,12F5.0)

Columns	Variable	Description
1-5	NLEVEL(I,L)	Number of axial locations in the L^{th} location set for assembly type LCASST(I). (NOTE: L= l,NB1(I), as defined on DRAG.3)
6-10	ILCS(I,L)	Identification number of starting channel in the L^{th} location set of assembly type LCASST(I).
11-15	ILCE(I,L)	Identification number of ending channel in the L^{th} location set of assembly type LCASST(I). (NOTE: All channels from ILCS(I,L) to ILCE(I,L) will be assigned the same losses at the same locations, as specified by the values for the FACTOR and ZCD arrays.)
16-20, 26-30, etc.	FACTOR(I,L,J)	Relative height (x/L) of the J^{th} axial location for the L^{th} channel set in assembly type LCASST(I).
21-25, 31-35, etc.	XCD(I,L,J)	Loss coefficient to be applied in channels ILCS(I,L) through ILCE(I,L) at axial location FACTOR(I,L,J).

Up to six pairs of (FACTOR,XCD) values can be entered on DRAG.4. If NLEVEL(I,L) is greater than six, the remaining pairs of (FACTOR,XCD) are read on DRAG.5 for the current location set before going on to the next location set.

DRAG.5 (FACTOR(I,L,J),XCD(I,L,J),J = 7, NLEVEL(I,L))

Read only if NLEVEL(I,L) > 6 on DRAG.4.
FORMAT(15X,12F5.0)

Columns	Variable	Description
1-15		BLANK
16-20, 26-30, etc.	FACTOR(I,L,J)	Relative height (x/L) of the J^{th} axial location for the L^{th} channel set in assembly type LCASST(I).
21-25, 31-35, etc.	XCD(I,L,J)	Loss coefficient to be applied in channels ILCS(I,L) through ILCE(I,L) at axial location FACTOR(I,L,J).

DRAG.3 and DRAG.4 (with DRAG.5, if needed) are read in sequence NOLC[DRAG.1] times, with DRAG.4 (and DRAG.5) read NB1(I) times for each time DRAG.3 is read.

Losses for Network Model

This input is required only if the network model has been specified by NETWK = 1 on DRAG.1. The assembly grouping for the network is defined by reading DRAG.6 for each assembly. Each assembly is assigned a group number, inlet loss parameters, and optional loss coefficient forcing functions. A loss coefficient for the total flow in all assemblies is specified on DRAG.7. Loss parameters for individual groups of assemblies are specified on DRAG.8 for NOGRP[DRAG.1] groups of assemblies.

DRAG.6 (NETGRP(N),NINFF(N),NOUTFF(N),RAIN(N),PWIN(N),HAIN(N),RAOUT(N), PWOUT(N),HAOUT(N),N = 1,NASSEM[CHAN.1])

Read only if NETWK = 1 on CHAN.1
FORMAT(3I5,6F10.0)

Columns	Variable	Description
1-5	NETGRP (N)	Assembly grouping number to be used with assembly N, corresponding to the assembly groups described in DRAG.9 (default is 1).
6-10	NINFF(N)	Identification number of loss coefficient forcing function versus Reynolds number, (defined on DRAG.5 and DRAG.6), to be used with assembly inlet loss, RAIN(N) (default is 0 for no forcing function).
11-15	NOUTFF(N)	Identification number of loss coefficient forcing function versus Reynolds number, to be used with assembly outlet loss RAOUT(N) (default is 0 for no forcing function).

16-25	RAIN(N)	Inlet loss parameter $(ft\text{-}lbm)^{-1}$ to be applied at inlet of assembly N.
26-35	PWIN(N)	Wetted perimeter (inches) associated with inlet loss RAIN (used only in Reynolds number calculation when NINFF(N) > 0; default is 1.0).
36-45	HAIN(N)	Gravitational head length (inches) associated with the assembly inlet loss.
46-55	RAOUT(N)	Outlet loss parameter $(ft\text{-}lbm)^{-1}$ to be applied at outlet of assembly N.
56-55	PWOUT(N)	Wetted perimeter (inches) associated with outlet loss RAOUT(N). (Used only in Reynolds number calculation when NOUTFF(N) > 0; default is 1.0).
66-75	HAOUT(N)	Gravitational head length (inches) associated with the assembly outlet loss.

DRAG.6 is read NASSEM[CHAN.1] times.

DRAG.7 RTIN

Read only if NETWK > 0 on DRAG.1
FORMAT(F10.5)

Columns	Variable	Description
1-10	RTIN	Loss parameter $(ft\text{-}lbm)^{-1}$ to be applied to the total flow rate (sum of all assembly flow rates; Reynolds-number independent).

DRAG.8 (RGIN(L),HGIN(L),RGOUT(L),HGOUT(L),L= 1,NOGRP)

Read only if NOGRP > 0 on DRAG.1
FORMAT(4F10.5)

Columns	Variable	Description
1-10	RGIN(L)	Loss parameter $(ft\text{-}lbm)^{-1}$ to be applied at assembly grouping inlet (Reynolds number independent).
11-20	HGIN(L)	Gravitational head length (inches) associated with the inlet pressure loss for group L.
21-30	RGOUT(L)	Loss parameter $(ft\text{-}lbm)^{-1}$ to be applied at assembly grouping outlet (Reynolds number independent).
31-40	HGOUT(L)	Gravitational head length (inches) associated with the outlet pressure loss for group L.

DRAG.8 is read NOGRP[DRAG.1] times.

Loss Coefficient Forcing Functions

This input is read only if NLCFP > 0 on DRAG.1. Loss coefficients are specified as constants (in XCD(I,L,J) on DRAG.4 and DRAG.5), but the user has the option of varying loss coefficients as a function of the Reynolds number. The forcing functions are numbered sequentially in the order they are defined on DRAG.7.

The variations in loss coefficients are specified as a function of Reynolds number. The same table of Reynolds number values applies to all forcing functions. The variations in loss coefficient with Reynolds number are expressed as multiplicative factors normalized to the value of XCD(I,L,J) entered on DRAG.4.

DRAG.9 (RECL(I),I = 1,NLCFP[DRAG.1])

Read only if NCLFP > 0 on DRAG.1
FORMAT(12E5.0)

Columns	Variable	Description
1-5, 6-10, etc.	RECL(I)	Reynolds number for the I^{th} point in the loss coefficient forcing function profiles.

NOTE: The table of Reynolds number values must be monotonically increasing.

DRAG.10 ((FFLC(J,L),J=l,NLCFP[DRAG.1]),L=l,NLCFF[DRAG.1]

Read only if NCLFF > 0 and NCLFP > 0 on DRAG.1.
FORMAT(12E5.0)

Columns	Variable	Description
1-5, 6-10, etc.	FFLC(J,L)	The loss coefficient factor for Reynolds number RECL(I) in the L^{th} forcing function profile.

DRAG.10 is read NLCFF[DRAG.1] times.

Local Blockages

This input allows the user to specify flow blockages at some axial locations in some channels. For the typical low velocities for gas flow in a spent fuel storage or transportation system, it is highly unlikely that this input would be needed or useful, except for specialized models of unusual configurations. But if it is needed, it is available. The axial flow at the blocked level is set to zero in the affected channels. Note that this input is specified by axial level number; if the axial noding is changed in a model, this input must be adjusted to reflect the new locations of

the blocked axial levels, or the blockages may turn up in the wrong places. This feature is optional and is read only if NBLOCK is greater than zero on DRAG.1.

DRAG.11 NBLOKA,NBLOKC

Read only if NBLOCK > 0 on DRAG.1.
FORMAT(2I5)

Columns	Variable	Description
1-5	NBLOKA	Axial level where the blockage is located.
6-10	NBLOKC	Number of channels blocked at axial level NBLOKA.

DRAG.12 (IBLOKA(I),IBLOKC(I),I = l, NBLOKC[DRAG.11])

Read only if NBLOCK > 0 on DRAG.1
FORMAT(16I5)

Columns	Variable	Description
1-5, 11-15, etc.	IBLOKA(I)	Identification number of assembly containing blocked channel IBLOKC(I).
6-10, 16-20, etc.	IBLOKC(I)	Identification number of the I^{th} blocked channel.

DRAG.11 and DRAG.12 are read in sequence NBLOCK times to define all blocked channels.

Lateral Friction Loss Correlations

The default approach in COBRA-SFS is to treat lateral friction and form losses with a constant value for all gaps. This is specified by input as the gap resistance KIJ(1), defined in group CHAN on CHAN.8. The optional input for DRAG.13 provides an alternative, in which the lateral friction loss can be expressed as a Blasius-type friction factor correlation.

The lateral pressure loss due to flow through a gap can be expressed in the form

$$\Delta P = \frac{f'|w|w}{2g_c\rho S^2} \tag{2.20}$$

where

f´ = friction loss coefficient for flow through a single gap.

The friction loss coefficient is expressed in terms of the transverse Reynolds number, which is defined with the rod diameter as the characteristic length, such that

$$Re_D = \frac{VD_{rod}\,\rho}{\mu}$$

The friction loss coefficient is represented by a set of turbulent and laminar formulations, as follows:

$$f'_T = a_T {Re_D}^{b_t} + c_T {Re_D}^{d_t} + e_t$$
$$f'_L = a_L {Re_D}^{b_L} + c_L \qquad (2.21)$$

The code evaluates the lateral friction factor as the maximum of the turbulent and laminar formulations as follows:

$$f' = \max\,(f'_T, f'_L)$$

The correlations are numbered sequentially from 1 to NFTRAN, in the order they are read in on DRAG.13. The friction factor correlation to be used in a particular lateral flow connection is determined by the value specified for N on CHAN.7 for the lower-numbered channel of the pair of channels that make up the lateral flow connection. The input for N on CHAN.7 is thus made to do double duty, since it also specifies the axial friction factor correlation (defined by input on DRAG.2) used to determine the channel friction pressure loss.

For example, suppose some channel II uses axial friction factor correlation number 3, specified on DRAG.2. For channel II that is connected to channel JJ through gap K, if NFTRAN is specified as greater than zero, then the lateral pressure loss in gap K will be determined using lateral friction factor correlation number 3, as specified on DRAG.13. The user must make certain that the value entered on CHAN.7 in variable N for a given channel corresponds to the correct axial friction factor for the channel *and* to the correct lateral friction factor correlation for the channel's lateral connections to other channels.

If the value of N on CHAN.7 for some channel II specifies an axial friction factor correlation number that has no corresponding lateral friction factor correlation number from DRAG.13, then the friction loss in any lateral flow connection from channel II to adjacent channels is calculated with the default loss coefficient, KIJ(1), entered on CHAN.8.

DRAG.13 AAT(I),BBT(I),CCT(I),DDT(I),EET(I),AATL(I),BBTL(I),CCTL(I), I=1,NFTRAN)

Read only if NFTRAN > 0 on DRAG.1.
FORMAT(8F5.0)

Columns	Variable	Description
1-5	AAT(I)	First multiplicative coefficient on the Reynolds number for turbulent flow lateral wall friction factor correlation (a_T in Eq. [6.21]).

6-10	BBT(I)	First Reynolds number exponent for turbulent flow lateral wall friction factor correlation (b_T in Eq. [6.21]).
1-15	CCT(I)	Second multiplicative coefficient on the Reynolds number for turbulent flow lateral wall friction factor correlation (c_T Eq. [6.21]).
16-20	DDT(I)	Second Reynolds number exponent for turbulent flow lateral wall friction factor correlation (i.e., d_T in Eq. [6.21]).
21-25	EET(I)	Additive coefficient for turbulent flow lateral wall friction factor correlation (e_T in Eq. [6.21]).
26-30	AATL(I)	Multiplicative coefficient on the Reynolds number for laminar flow lateral wall friction factor correlation (a_L in Eq. [6.21]).
31-35	BBTL(I)	Reynolds number exponent for laminar flow lateral wall friction factor correlation (b_L in Eq. [6.21]).
36-40	CCTL(I)	Additive coefficient for laminar flow lateral wall friction factor correlation (c_L in Eq. [6.21]).

*** DRAG.13 is read in NFTRAN[DRAG.1] times. ***

2.4.11 Group BDRY–Thermal Boundary Conditions

The input for this group defines the heat transfer boundary conditions for the calculation. In most instances, this will be the interface between the system being modeled and the environment, but the user has a great deal of flexibility in specifying the heat transfer boundary conditions for a model.

The default in COBRA-SFS is to assume that the side boundaries of a system are adiabatic. This is an artifact of the code's original development for reactor core modeling. The ability to specify side boundary conditions is one of the major enhancements to the code for application to spent fuel storage and transportation systems. However, it means that if there is heat transfer to the environment in the radial direction, boundary conditions must be defined explicitly for the sides of the system.

In COBRA-SFS, heat is transferred out the boundaries of a model through a series of user-defined boundary regions, each of which can use a different boundary connection type. The general structure of a boundary connection type allows the connection to represent conduction, convection and radiation heat transfer in the boundary region. It is defined as a semi-empirical correlation for the boundary surface heat flux q_b'' and has the form

$$q_b'' = C_1[C_2(T_i - T_{i+1})]^{C_3}(T_i - T_{i+1}) + \sigma\left(\frac{1}{\epsilon_1} + \frac{1}{\epsilon_{i+1}} - 1\right)^{-1}(T_i^4 + T_{i+1}^4) \quad (2.22)$$

where

C_n = user-defined constants
σ = Stefan-Boltzmann constant, 0.1714(10^{-8}) Btu/h-ft^2-°R^4
ε_i, ε_{i+1} = surface emissivity of boundary nodes i and i + 1
T_i, T_{i+1} = temperature of nodes i and i + 1.

The total number of boundary types, NBTYP, is defined for a given model by the user on BDRY.1. These boundary connection types are defined by input on BDRY.2, with coefficients C_n for correlations of the form shown in Eq. (2.22). The total number of slab nodes that use boundary connection types of the form defined by BDRY.2 input to define the side boundary of the model is specified in the value defined for NWSID on BDRY.1.

The form of Eq. (2.22) is extremely generalized, and the user must specify appropriate values for C_n to define boundary types for the specific conditions of a given model. For connections that are conduction only, as is typically required for layers of material in the plenum modeling, and may also be needed for side boundaries in some models, the definitions of the C_n are fairly straightforward. The term $C_1(T_i - T_{i+1})$ alone is all that is needed to capture the heat flux due to conduction, and the remainder of the equation can be quietly dropped. The term $[C_2(T_i - T_{i+1})]^{C3}$ is eliminated by the simple expedient of setting C_2=1.0 and C_3=0.0, which reduces this term to unity. The radiation heat transfer component in the relationship can be eliminated by omitting surface emissivites from the input for the specific boundary type. This reduces Eq. (2.22) to simply the conduction equation, captured by means of the C_1 coefficient as

$$C_1 = \frac{k}{dx}$$

where

k = thermal conductivity of the material comprising the boundary connection type
dx = thickness of the connection

For boundary connection types that capture the effect of convection heat transfer (with or without thermal radiation), however, the appropriate definitions of the C_n values are more complicated. Natural convection heat transfer correlations are typically of the form

$$Nu = \frac{HL}{k} = a(GrPr)^b$$

where the Grashof number is defined as

$$Gr = \frac{g\beta L^3}{\left(\frac{\mu}{\rho}\right)^2}$$

Inserting this definition of the Nusselt number into Eq. (2.22) yields the following definitions for the C_n coefficients;

$$C_1 = a\left(\frac{k}{L}\right)$$

$$C_2 = \frac{g\beta L^3 Pr}{\left(\frac{\mu}{\rho}\right)^2}$$

$$C_3 = b$$

With this general parsing of the coefficients for Eq. (2.22), boundary connection types can readily be defined for any appropriate Nusselt number the user wishes to apply in a given model.

The temperature of the external environment seen by the boundary node is defined using axial profiles of boundary temperatures, read on BDRY.3. The number of axial boundary temperature profiles to be read in is defined by the user, as NBTEMP on BDRY.1. The user can read in as many axial profiles as needed to characterize a given model. The boundary condition types seen by specific slab nodes are defined by input on BDRY.5 through BDRY.7. This input also includes the option to define axially varying boundary connection types.

If the plenum model is not used, the boundaries at the top and bottom of the system are assumed adiabatic for heat transfer, and the flow boundary conditions are defined by the input in group OPER. In this case, the inlet flow boundary condition treats the entering fluid as coming from an essentially infinite reservoir at the user-specified fluid boundary temperature and pressure, as defined in OPER. At the exit, the fluid vanishes into another essentially infinite reservoir at the specified boundary pressure. If flow should reverse at the exit, the properties of the incoming fluid are defined at the boundary pressure and some user-specified temperature.

The group BDRY input includes the alternative option of defining inlet and outlet plena to model a closed flow system, with heat transfer connections to the external environment. When the optional plenum model is used, the boundary conditions specified in group OPER are used to define the initial values for inlet flow rate, temperature, and pressure. Subsequently, conditions in the inlet plenum are calculated for each iteration by summing the flows from all channels having negative inlet flow. This temperature and pressure are used to define the conditions for flow into all channels with positive inlet flow. In the upper plenum, the conditions are calculated by summing the flow from channels with positive flow at the exit. These conditions define the temperature and pressure of flow into channels with negative flow at the top.

The fluid conditions in the inlet plenum and the exit plenum are treated as one-dimensional. It is assumed that the flow mixes instantaneously in these regions, and no radial gradients are maintained in the fluid. Thermal boundary conditions on the upper and lower plena can vary radially, however. In addition, different plenum boundary regions can connect axially to different slab nodes, to appropriately model heat transfer paths at the ends of the storage system represented in a given model. It is also possible to specify separate top and side boundary conditions on the plenum regions. The plenum input is specified on lines BDRY.8 through BDRY.14.

BDRY.1 AGROUP,NBTYP,NBTEMP,NWSID,NPR
FORMAT(A4,1X,4I5)

Columns	Variable	Description
1-4	AGROUP	Enter BDRY.
6-10	NBTYP	Number of thermal boundary connection types.
11-15	NBTEMP	Number of axial boundary temperature profiles.
16-20	NWSID	Number of solid structure nodes connected to the side thermal boundary.
21-25	NPR	Plenum model flag:

		= 0; no plenum heat transfer model. = 1; one plenum only is modeled, either the upper or the lower plenum. = 2; lower plenum and upper plenum are both modeled.

Thermal Boundary Connection Types

The thermal boundary connection types are structured to be as general as possible. They are formulated assuming that the heat flux in the boundary region can be expressed as a function of the temperature difference between adjacent boundary nodes, in the form shown in Eq. (2.22).

BDRY.2 NN,C1,C2,C3,EM1,EM2
FORMAT(I5,3F10.0,2F5.3)

Columns	Variable	Description
1-5	NN	Thermal boundary connection type identification number. (NOTE: Must be entered in sequence, from 1 to NBTYP[BDRY.1].)
6-15	C1	Leading coefficient for user-specified heat flux correlation defining the boundary connection type (C_1 in Eq. [6.22]). (NOTE: C_1 is in units of Btu/s-in.^{2}-°F. This is an historical artifact of the code's development path, and cannot be changed.)
16-25	C2	Second coefficient for user-specified heat flux correlation defining the boundary connection type (C_2 in Eq. [6.22]). (NOTE: C_2 has units of °F^{-1}.)
26-35	C3	Third coefficient for user-specified heat flux correlation defining the boundary connection type (C_3 in Eq. [6.22]). This coefficient provides the exponent on the temperature difference term in Eq. [6.22], which can be eliminated by specifying C3=0.0, if not needed for the connection type. (NOTE: C_3 is dimensionless.)
36-40	EM1	Surface emissivity for boundary node i of a pair connected by thermal boundary connection type NN. (NOTE: Enter zero if there is no radiative heat transfer for this boundary connection type.)

Columns	Variable	Description
41-45	EM2	Surface emissivity for boundary node i + 1 of a pair connected by thermal boundary connection type NN. (NOTE: Enter zero if there is no radiative heat transfer for this boundary type. However, if EM1 > 0.0 for this type, and EM2 is entered as zero or blank, the default for EM2 is 1.0, on the assumption that this connection type defines the outermost boundary of the system, and there is thermal radiation from surface i to an infinite ambient.)

*** BDRY.2 is read NBTYP[BDRY.1] times. ***

Boundary Axial Temperature Profiles

This input defines the temperature of the environment seen by the system. A boundary temperature can be specified as uniform by entering a two-point table, with the same temperature at relative axial locations 0.0 and 1.0. The axial locations are normalized to the model axial length, specified with input variable Z on CHAN.3. A total of NBTEMP[BDRY.1] boundary temperature profiles must be entered.

BDRY.3 (N,NZONET(N),(ZENDT(I,N),BT(I,N),I =1,6)

Read only if NBTEMP > 0 on BDRY.1.
FORMAT(2I5,6(2F5.0))

Columns	Variable	Description
1-5	N	Side boundary temperature profile identification number. (NOTE: Must be entered in sequence, from 1 to NBTEMP.)
6-10	NZONET(N)	Number of pairs of entries in the N^{th} table of boundary temperature versus axial position. (NOTE: Each table must include at least two points; for relative axial locations 0.0 and 1.0.)
11-15, 21-25, 31-35, etc.	ZENDT(I,N)	Relative axial location (x/L) of the I^{th} element of boundary temperature profile N, where L is defined by input variable Z on CHAN.3. (NOTE: Array ZENDT must begin with 0.0 and increase monotonically to 1.0.)
16-20, 26-30,	BT(I,N)	Boundary temperature at location ZENDT (I,N).

Columns	Variable	Description
36-40, etc.		

Up to six pairs of values for (ZENDT[I,N],BT[I,N]) can be entered on BDRY.3. If NZONET(N) > 6, the remaining pairs of points are read on BDRY.4, with 6 pairs of points per line. All points for the N^{th} profile are entered before going on to the input for the next profile.

BDRY.4 (ZENDT(I,N),BT(I,N),I=7,NZONET(I))

Read only if NZONET(N) > 6.
FORMAT(10X,6(2F5.0))

Columns	Variable	Description
1-10		BLANK
11-15, 21-25, 31-35, etc.	ZENDT(I,N)	Relative axial location (x/L) of the I^{th} element of boundary temperature profile N. (ZENDT must begin with 0.0 and increase monotonically to 1.0.)
16-20, 26-30, etc.	BT(I,N)	Boundary temperature at location ZENDT(I,N).

*** BDRY.3 (with BDRY.4, if needed) is read NBTEMP[BDRY.1] times***

Side Thermal Boundary Connections

Heat transfer connections to the boundary must be defined for NWSID[BDRY.1] slab nodes. The heat transfer connections linking a given slab node to the boundary are defined as occurring through a series of sequentially connected regions, each of which can use a different boundary connection type to define the heat transfer through that region. In most cases, one region is sufficient to define the heat transfer connection from the slab node constituting the edge of the system to the environment, (e.g., from the outer surface of the cask to the ambient). In some cases, however, there may be additional material between the boundary slab node and the environment that is not modeled explicitly with slab nodes using the input in group SLAB. The effect of this material can be included by using appropriately defined boundary regions, connected by user-defined boundary connection types. The connection types specified between the boundary regions must define the heat flux through the regions using the relation in Eq. (2.22).

The total number of regions connecting a given slab node to the environment is defined by the user, in variable NHSID(I) on BDRY.5 for each of the NWSID nodes with connections to the boundary. The first region connects to the slab node on one side, and to the next region out on the other side. The remaining regions, 2 through NHSID(I), are connected in series, each to the adjacent region by an appropriate boundary type. The NHSID(I)th region connects to the

specified boundary temperature. (If NHSID(I) is 1, the slab node is connected directly to the boundary temperature, using the specified boundary connection type, with no intermediate regions.)

The user can also specify a heat source term in the outermost radial region of a thermal boundary connection. This allows the user to model heat loading on the cask due to external conditions, such as fire or solar radiation.

For the IWSID(N) solid structure node identified on BDRY.5, the user must define the radial boundary region associated with that node. BDRY.6 is read NHSID(N) times to define all radial boundary regions connecting node IWSID(N) to the boundary temperature.

BDRY.5 (IWSID(N),NHSID(N),AWSID(N),NBTTYP(N),QBSRC(N),N = 1,NWSID[BDRY.1])

Read only if NWSID > 0 on BDRY.1.
FORMAT(2I5,E5.3.I5,E5.3)

Columns	Variable	Description
1-5	IWSID(N)	Index number of N^{th} solid structure node connected to a side thermal boundary temperature.
6-10	NHSID(N)	Number of radial boundary regions connecting node IWSID(N) to the boundary temperature.
11-15	AWSID(N)	Length (inches) of node IWSID(N) facing the boundary. (NOTE: This term is properly an area, but the input required is length of boundary node N on a radial cross-section of the model. The length specified by the user is automatically multiplied by the local axial node length in the model, thus defining the correct area. See discussion in Section 3.4.6, for thermal resistance connection types in group SLAB.)
16-20	NBTTYP(N)	Identification number of boundary temperature profile seen by node IWSID(N) on the far side of region NHSID(N). (NOTE: Must correspond to the index of a profile entered on BDRY.3.)
21-25	QBSRC(N)	Optional boundary heat flux (Btu/h-ft^2); to be applied to the $NHSID(N)^{th}$ radial region (default is 0.0).

BDRY.6 ((I,SPER(I,N),NZONEB(I,N),(ZENDB(I,N,K),NBCTYP(I,N,K),I = 1,NHSID(N), K=1,6)

Read only if NHSID(N) > 0 on BDRY.5.
FORMAT(I5,F5.0,I5,6(F5.0,I5))

Columns	Variable	Description
1-5	I	Index of a radial boundary region associated with node IWSID(N). (NOTE: Boundary regions must be numbered sequentially from 1 to NHSID(N).)
6-10	SPER(I,N)	Perimeter of I^{th} boundary region of node IWSID(N), normalized to AWSID(N) (dimensionless).
11-15	NZONEB(I,N)	Number of axial zones to be read for a table of boundary connection type versus axial position for region I. (Default is 1, for an axially uniform boundary connection.)
16-20, 26-30, etc.	ZENDB(I,N,K)	Relative axial location (x/L) of the end of the K^{th} boundary connection type zone in region I. (NOTE: If NZONEB(I,N) is 1, the boundary is axially uniform, and ZENDB(I,N,1) must be entered as 1.0.)
21-25, 31-35, etc.	NBCTYP(I,N,K)	Boundary thermal connection type for the K^{th} axial segment of region I. (This index must correspond to a boundary connection type defined on BDRY.2).

NOTE: The first six pairs of (ZENDB,NBCTYP) for any radial boundary region I of the $NWSID(N)^{th}$ boundary node are read on BDRY.6. If NZONEB(I,N) > 6, the remainder are read on BDRY.7, with six entries on a line, before going on to the input for the next region.

BDRY.7 (ZENDB(I,N,K),NBCTYP(I,N,K),K=7,NZONEB(I,J))

Read only if NZONEB(I,N) > 6 on BDRY.6.
FORMAT(15X,6(F5.0,I5))

Columns	Variable	Description
1-15		BLANK
16-20, 26-30, etc.	ZENDB(I,N,K)	Relative axial location (x/L) of the end of the K^{th} boundary connection type zone in region I.
21-25, 31-35, etc.	NBCTYP(I,N,K)	Boundary thermal connection type for the K^{th} axial segment of region I (this index must correspond to a boundary connection type defined on BDRY.2).

BDRY.6 (with BDRY.7, if needed) is read NHSID(N)[BDRY.5] times for node IWSID(N), which is connected to a thermal boundary.

BDRY.5 and BDRY.6 are read as a set NWSID[BDRY.1] times.

Plenum Boundary Connections

This input group defines the flow and heat transfer connections to the upper and lower plena. This input is optional; if NPR[BDRY.1] is zero, the upper and lower heat transfer boundaries are assumed adiabatic, and the flow boundary is determined by the input in group OPER. When this option is used, the model may include the lower plenum only, the upper plenum only, or both the lower plenum and upper plenum.

The upper and lower plena are basically specialized heat transfer boundary models, but also include a lumped parameter fluid node. Flow from all channels connected to a plenum is assumed to be instantaneously and completely mixed at a single temperature and pressure. The flow channels of all assemblies are assumed to connect to the plena, unless an assembly is explicitly specified as not connected, using the input on BDRY.13 and BDRY.14. The lower plenum provides an inlet fluid temperature for positive flow entering the channels at the inlet, and the upper plenum provides the fluid temperature for negative flow entering the channels at the exit. Natural recirculation is calculated in the momentum solution.

Heat transfer paths can be specified through the plena, connecting the slab nodes with the appropriate boundary temperatures by means of detailed connections between the plenum regions and the top or bottom of the slab nodes. The plena are defined with nominal heat transfer areas in the axial and radial directions. These areas model surfaces through which heat can move through the sides and outer surface (top or bottom) of the plenum. The heat transfer paths through the plenum to the boundary are defined using a series of plenum boundary regions, in the same manner as the boundary regions for the side boundary nodes. These regions connect radially or axially from the plenum to the boundary, and the connections between regions are specified by thermal boundary connection type(s), as defined on BDRY.2. The ambient temperature of the environment beyond the upper or lower plenum is specified in TAMBP(1,N) for the radial regions, and in TAMBP(2,N) for the axial regions, both on BDRY.8. The user may also specify heat sources in the outermost region of the plenum.

The input for each plenum is read on BDRY.8 through BDRY.12. If NPR[BDRY.1] is 1, only one plenum is specified. It may be either the upper plenum or the lower plenum, as specified by the flag N on BDRY.8. If NPR[BDRY.1] is specified as 2, both the lower plenum and upper plenum must be defined. In this case, the lower plenum is defined in the first set of inputs for BDRY.8 through BDY.12, and the upper plenum is defined in the second set of inputs.

BDRY.8 N,AWPS(N),AWPAX(N),NPSID(N),NPAX(N),TAMBP(1,N),TAMBP(2,N), QPSRC(1,N),QPSRC(2,N)

Read only if NPR > 0 on BDRY.1
FORMAT(I5,2F5.3,2I5,4F5.3)

Columns	Variable	Description
1-5	N	Flag for plenum region under consideration: = 1; lower plenum. = 2; upper plenum.
6-10	AWPS(N)	Nominal plenum radial heat transfer area (in.^{2}).
11-15	AWPAX(N)	Nominal plenum axial heat transfer area (in.^{2}).
16-20	NPSID(N)	Number of radial plenum boundary regions.
21-25	NPAX(N)	Number of axial plenum boundary regions.
26-30	TAMBP(l,N)	Plenum radial boundary temperature (°F).
31-35	TAMBP(2,N)	Plenum axial boundary temperature (°F).
36-40	QPSRC(1,N)	Optional plenum radial boundary heat flux (Btu/h-ft^2). (NOTE: This is applied to the outermost plenum radial region only.)
41-45	QPSRC(2,N)	Optional plenum axial boundary heat flux (Btu/h-ft^2). (NOTE: This is applied to the outermost plenum axial region only.)

BDRY.9 (I,SPERP(K,1,N),NPTYP(K,1,N),NPWN, (IPINTP(1,NPW),MATYPP(NPW), AREACP(NPW),DXPLEN(NPW),J = 1,3)

Read only if NPSID(N) > 0 on BDRY.8
FORMAT(I5,F5.0,2I5,3(2I5,2FS.0))

Columns	Variable	Description
1-5	I	Plenum radial region number. (NOTE: Regions must be numbered in sequence from 1 to NPSID(N).)
6-10	SPERP (K,1,N)	Plenum radial region heat transfer area multiplier, normalized to AWPS(N) on BDRY.8 (dimensionless).
11-15	NPTYP (K,1,N)	Thermal connection type number for connections between slab nodes and plenum radial region I. (NOTE: This must correspond to a boundary connection type defined on BDRY.2.)

Columns	Variable	Description
16-20	NPWN	Number of slab nodes connected to plenum radial region I.
21-25, 41-45, 61-65	IPINTP (I,NPW)	Identification number of the J^{th} slab node connected to plenum radial region I. (NOTE: Must correspond to a node entered on SLAB.3.) **Warning**: Do NOT attempt to connect boundary nodes (defined by input on BDRY.5) to plenum regions. This is fatal.
26-30, 46-50, 66-70	MATYPP (NPW)	Identification number of the material type for the connection between node IPINTP(1,NPW) and plenum axial region I.
31-35, 51-55, 71-75	AREACP (NPW)	Heat transfer area ($in.^2$) for connection between node IPINTP(1,NPW) and plenum region I. (Default is node IPINTP(1,NPW) cross-sectional area for axial conduction, WALLXC, specified on SLAB.3.)
36-40, 56-60 76-80	DXPLEN (NPW)	Length (inches) for heat transfer from node IPINTP(1,NPW) to the center of plenum radial region I.

NOTE: Up to three sets of values for (IPINTP, MATYPP, AREACP, DXPLEN) may be read on BDRY.9. If NPWN[BDRY.9] is greater than three, then the remaining sets of values are read on BDRY.10, with 3 sets of values per line, before going on to the input for the next plenum region.

BDRY.10 (IPINTP(1,NPW),MATYPP(NPW),AREACP(NPW),DXPLEN(NPW),J=4,NPWN)

Read only if NPWN > 3 on BDRY.9
FORMAT(20X,3(2I5,2F5.0))

Columns	Variable	Description
1-20		BLANK
21-25, 41-45, 61-65	IPINTP(I,NPW)	Identification number of the J^{th} slab node connected to plenum radial region I. (NOTE: Must correspond to a node entered on SLAB.3.)
26-30, 46-50, 66-70	MATYPP(NPW)	Identification number of the material type for the connection between node IPINTP(I,NPW) and plenum axial region I.
31-35, 51-55, 71-75	AREACP(NPW)	Heat transfer area ($in.^2$) for connection between node IPINTP(1,NPW) and plenum region I. (Default is cross-sectional area for axial conduction for node IPINTP(1,NPW), defined by input for WALLXC, specified on SLAB.3.)

Columns	Variable	Description
36-40, 56-60, 76-80	DXPLEN(NPW)	Length (inches) for heat transfer from node IPINTP(1,NPW) to the center of plenum radial region I.

*** BDRY.9 (with BDRY.10, if needed), is read K = 1,NPSID(N)[BDRY.8] times. ***

BDRY.11 (I,SPERP(K,2,N),NPTYP(I,2,N),NPWN,(IPINTP(1,NPW),MATYPP(NPW), AREACP(NPW),DXPLEN(NPW),J = 1,3)

Read only if NPAX(N) > 0 on BDRY.8
FORMAT(I5,F5.0,2I5,3(2I5,2F5.0))

Columns	Variable	Description
1-5	I	Plenum axial region number. (NOTE: Regions must be numbered sequentially from 1 to NPAX(N).)
6-10	SPERP(K,2,N)	Plenum axial region heat transfer area multiplier, normalized to AWPAX(N) on BDRY.8.
11-15	NPTYP(K,2,N)	Thermal connection type for connections between slab nodes and plenum axial region I. (NOTE: This must correspond to a boundary connection type defined on BDRY.2.)
16-20	NPWN	Number of solid nodes connected to plenum axial region I.
21-25, 41-45, 61-65	IPINTP(l,NPW)	Identification number of the J^{th} solid node connected to plenum axial region I. (NOTE: Must correspond to a node entered on SLAB.3.) **Warning**: Do NOT attempt to connect boundary nodes (defined by input on BDRY.5) to plenum regions. This is fatal.
26-30, 46-50, 66-70	MATYPP(NPW)	Identification number of the solid material type for the connection between IPINTP(l,NPW) and plenum axial region I.
31-35, 51-55, 71-75	AREACP(NPW)	Heat transfer area ($in.^2$) for connection from slab node IPINTP(l,NPW) and plenum axial region I (default is node IPINTP(l,NPW) area for axial conduction, WALLXC, specified on SLAB.3).

36-40, 56-60, 76-80	DXPLEN(I,J)	Length (inches) for heat transfer from node IPINTP(l,NPW) to center of plenum axial region I.

NOTE: Up to three sets of values for (IPINTP, MATYPP, AREACP, DXPLEN) may be read on BDRY.11. If NPWN[BDRY.11] is greater than three, then the remaining sets are read on BDRY.12, with three sets of values per line, before going on to the next plenum region.

BDRY.12 (IPINTP(I,J),MATTYPP(I,J),AREACP(I,J),DXPLEN(J),J =4,NPWN)

Read only if NPWN > 3 on BDRY.11
FORMAT(20X,3(2I5,2F5.0))

Columns	Variable	Description
1-20		BLANK
21-25, 41-45, 61-65	IPINTP (1,NPW)	Identification number of the J^{th} slab node connected to plenum axial region I. (NOTE: Must correspond to a node entered on SLAB.3.)
26-30, 46-50, 66-70	MATYPP (NPW)	Identification number of the solid material type for the connection between IPINTP(1,NPW) and plenum axial region I.
31-35, 51-55, 71-75	AREACP (NPW)	Heat transfer area ($in.^2$) for connection from slab node IPINTP(l,NPW) and plenum axial region I. (Default is node IPINTP(1,NPW) area for axial conduction, WALLXC, specified on SLAB.3).
36-40, 56-60, 76-80	DXPLEN(I,J)	Length (inches) for heat transfer from node IPINTP(1,NPW) to center of plenum axial region I.

BDRY.11 (with BDRY.12, if needed), is read K = l,NPAX(N)[BDRY.8] times.

The input lines BDRY.8 through BDRY.12 are read as a set NPR[BDRY.1] times,

to define the plena for a given model.

Assemblies Not Connected to Plena

This input is read only when the plenum model is used. It allows the user to specify assemblies that are not connected to the plena. The total number of unconnected assemblies is specified in the variable NNPA in BDRY.13, and the index numbers of the unconnected assemblies are read on BDRY.14. If all assemblies are connected to the plena, then NNPA is specified as zero, and BDRY.14 is not read.

BDRY.13 NNPA

Read only if NPR > 0 on BDRY.1
FORMAT(I5)

Columns	Variable	Description
1-5	NNPA	Number of assemblies that are not connected to the plenum model.

BDRY.14 IUNCA(I), I=1,NNPA

Read only if NNPA > 0 on BDRY.13
FORMAT(12I5)

Columns	Variable	Description
1-5, 6-10, 11-15, 16-20 . . . 56-60	IUNCA(I)	Index number of the I^{th} assembly that is **not** connected to the plenum model.

BDRY.14 is read as many times as necessary to supply NNPA entries,

with up to 12 entries on a record.

2.4.12 Group OPER–Operating Conditions

The input for this group defines the flow boundary conditions for the system, and the heat generation rate for the rods. The structure of this input group strongly reflects the roots of COBRA as a code for modeling reactor core thermal-hydraulics. The approach to defining the boundary conditions in COBRA-SFS has been greatly modified by the addition of the plenum models, defined in group BDRY, and thus renders many options related to flow boundary conditions in group OPER unusable. However, these features have been retained in COBRA-SFS, for flexibility and because they often turn out to be rather useful for unexpected applications of the code. Not simply for sentimental reasons.

When the plenum model in group BDRY is not used, group OPER is used to specify the flow boundary conditions at the inlet and outlet of the system. The user specifies the temperature and flow rate for fluid entering the system at the inlet to the channels and the channel exit pressure. This constitutes a complete set of flow boundary conditions for normal upflow conditions. If the flow could be negative at a channel exit, the user must also specify an enthalpy for fluid carried into the system by this reverse flow.

If the lower plenum model is specified in group BDRY, the inlet conditions are calculated as part of the overall solution, based on fluid conditions in the plenum. In such a case, the boundary

conditions specified in group OPER are used as the first guess in the iterative solution. If the upper plenum model is used, the properties of the fluid transported into the channels if flow reverses at the exit are determined from the solution of the fluid conditions in the upper plenum.

This group can also be used to define forcing functions for transient calculations. The transient capability in COBRA-SFS has been verified and validated (Rector et al. 1998) with the release of Cycle 3. The COBRA-SFS code has been used extensively to analyze wet-to-dry storage transfer operations, particularly vacuum drying transients in spent fuel canisters, primarily at design basis, but also for site-specific initial decay heat loading conditions (which are generally significantly below design basis). The code has also been used to evaluate SNF package response to postulated HAC fire conditions for many different transportation packages.

In addition, COBRA-SFS has been used to evaluate transportation package response to conservative representations of actual severe transportation fire accident conditions (NUREG/CR-6886, Rev. 2, 2009; NUREG/CR-6894, Rev. 1. 2007; NUREG/CR-7206, 2016; NUREG/CR-7207, 2016; NUREG/CR-7209, 2017). However, there is not an extensive body of experimental data from actual testing of spent fuel storage systems for transient conditions under normal, off-normal, and accident conditions. If such data were ever to become available, additional validation of the code for transient applications would be appropriate.

Axial and radial distributions of the assembly heating rates, which are needed to represent realistic loading conditions for spent fuel storage systems, are also specified in this group.

OPER.1 AGROUP,IHH,IG,ITDP,NRPF,NP,NH,NG,NGPRFL,NHX,NQ,NQPRFL,
NDPA,NRAMPH,NAXP
FORMAT(A4,1X,14I5)

Columns	Variable	Description
1-4	AGROUP	Enter OPER.
6-10	IHH	Flag for units of enthalpy or temperature on inlet (or initial) fluid conditions: = 0; uniform inlet (or initial) enthalpy (Btu/lbm) in all channels. = 1; uniform inlet (or initial) temperature (°F) in all channels. = 2; define inlet enthalpy (Btu/lbm) for each channel on OPER.3. = 3; define inlet temperature (°F) for each channel on OPER.3. (NOTE: Option IHH = 2 (or 3) is often needed for models with regions unconnected to the plenum, such as a model of a vertical ventilated storage system, in order to specify different inlet enthalpy (or temperature) boundary condition values for assemblies that are NOT connected to the lower plenum, compared to assemblies that are connected to the plenum. The unconnected assemblies (representing the air annulus) usually have a fixed boundary temperature corresponding to ambient air temperature, which in most cases is much lower than the lower plenum inlet temperature (calculated in the solution) for

Columns	Variable	Description
		the gas (usually helium) recirculating within the canister in assemblies containing spent fuel. The simple expedient of specifying a uniform enthalpy or temperature for *all* assemblies in the model, (by entering IHH=0 or 1), can result in convergence problems, particularly for a model with relatively high heat load (e.g., at or near design basis.) These convergence issues can occur if the uniform value specified on OPER.2 is relatively far from the final converged solution for inlet temperature seen by assemblies connected to the lower plenum. Generally, a low initial value (nominally in the range 80-100°F) would be necessary to assure that the unconnected assemblies would see the correct inlet boundary condition, usually corresponding to ambient air temperature. If such convergence issues occur for a given model with IHH < 2, it is necessary to select IHH = 2 or 3, and use the input for OPER.3 to specify different inlet enthalpies or temperatures in the channels of the unconnected assemblies, compared to the values specified for the connected assemblies, such that the solution in all regions starts from initial conditions closer to the final result. A typical initial temperature value for assemblies connected to the plena would be on the order of 300-400°F, for design basis storage conditions.
11-15	IG	Flag for option to specify inlet (or initial) flow distribution: = 0; average inlet mass flux (Mlbm/h-ft^2). = 2; average inlet mass flux (Mlbm/h-ft^2), but assembly inlet flows will be determined by user-supplied flow fractions on OPER.4. Channel mass flux will be uniform within each assembly. = 3; average inlet mass flux (Mlbm/h-ft^2), but channel flows in each assembly will be determined by user-supplied flow fractions on OPER.7. = 4; average inlet mass flux (Mlbm/h-ft^2), but flows will be determined by both assembly flow fractions on OPER.4 and channel flow fractions on OPER.6. (NOTE: When the lower plenum is modeled, IG = 0 is the logical option for this input, unless it is necessary to specify non-uniform inlet flow conditions for assemblies that are not connected to the plenum.)

Columns	Variable	Description
16-20	ITDP	Flag for flow or pressure drop boundary condition to be applied to all assemblies: = -1; boundary condition flag set for each assembly by using ITDPA(NASS) on CHAN.6. = 0; specified flow boundary condition. = 1; uniform pressure drop (specified in DPS on OPER.2). = 2; uniform inlet mass flux (required pressure drop calculated as part of the solution). = 3; uniform pressure drop calculated for zero net inlet mass flow rate (FTOTAL on OPER.2. This is valid only for FTOTAL = 0.0). = 4; zero flow at both top and bottom boundaries. (NOTE: ITDP=4 can be used only when the plenum model is NOT used.) SPECIAL NOTE: In a spent fuel storage system or transportation package model of any complexity, it is strongly recommended that the user specify ITDP as -1 on OPER.1, and define the appropriate boundary flag in ITDPA(NASS) on CHAN.6 for each assembly. If the plenum model is used, the appropriate boundary flag is 3 for assemblies connected to the plena, and a value of 1 or 2 is typically used for assemblies that are not connected to the plena. Note that if any assemblies are NOT connected to the plena, then ITDP ***must*** be -1, and individual boundary flags must be specified on CHAN.6.
21-25	NRPF	Flag for assembly heating rate specification: = 0; all assemblies have the same average heat generation rate (PDN on OPER.2). = 1; average heating rate is specified in PDN on OPER.2, and relative heat generation factors normalized to PDN are specified for each assembly on OPER.8. = 2; heat generation rate is read for each assembly, on OPER.8, (PDN on OPER.2 is not used). SPECIAL NOTE: This option can be superseded by utilizing subgroup REST; see Section 2.4.12.1. If this subgroup is included in the input stream, NRPF must be entered as 0.
26-30	NP	Number of pairs of values in table for transient forcing function on system pressure, to be read on OPER.9.

Columns	Variable	Description
31-35	NH	Number of pairs of values in transient forcing function table on inlet enthalpy or temperature, to be read on OPER.10. (NOTE: If the lower plenum model is specified in group BDRY, this option can be used to define forcing functions *only for channels in assemblies NOT connected to the lower plenum*. Forcing functions cannot be applied to the inlet flow of channels in assemblies connected to the lower plenum.)
36-40	NG	Number of pairs of values in transient forcing function table on the flow boundary , to be read on OPER.11 and OPER.12: • if ITDP = 0 or 2, the forcing function is on the inlet flow rate. • if ITDP = 1 or 3, the forcing function is on the pressure drop boundary condition. • if ITDP = 4, this option cannot be used. (NOTE: If the lower plenum model is specified in group BDRY, this option can be used to define inlet forcing functions *only for channels in assemblies NOT connected to the lower plenum*. Forcing functions cannot be applied to the flow boundary condition on channels in assemblies connected to the lower plenum.)
41-45	NGPRFL	Number of transient forcing functions for inlet mass flux or pressure drop profiles to be read on OPER.12 (used only if NG > 0).
46-50	NHX	Number of pairs of values in transient forcing function table on exit enthalpy or temperature, to be read on OPER.15. (NOTE: If the upper plenum model is specified in group BDRY, this option can be used to define a forcing function *only for channels in assemblies NOT connected to the upper plenum*. A forcing function cannot be applied to the exit enthalpy boundary condition on channels in assemblies connected to the upper plenum.)
51-55	NQ	Number of entries in table(s) of transient forcing function(s) on average heat generation rate. The table of time-values to be used by all forcing functions on heat generation rate is read on OPER.13. (NOTE: All tables of heat generation rate forcing functions must use the same set of time values.)
56-60	NQPRFL	Number of heat generation rate transient forcing function tables to be read on OPER.14 (used only if NQ > 0).

Columns	Variable	Description
61-65	NDPA	Flag for option to specify different pressure drop boundary conditions for each assembly when using the network model. = 1; assembly pressure drops to be specified on OPER.18. = 0; uniform pressure drop or flow boundary condition. (NOTE: The network model is specified by NETWK > 0 on DRAG.1. If NETWK is specified as zero, the input for NDPA is not used. See the input instructions for group DRAG.) (WARNING: The plenum model specified in group BDRY cannot be used with the network option. If the network option is used, ITDP must be specified as 1 or 2, and all assemblies must use the same boundary condition option.)
66-70	NRAMPH	Number of iterations over which the heat generation is to be ramped into the solution (default is 1).
71-75	NAXP	Number of axial heat generation profiles to be entered on OPER.16 and OPER.17. (Default is zero, for uniform axial heat generation distribution for all rods). (NOTE: Specifying this option as NAXP = 0 assumes that the uniform heat generation profile extends ***over the full axial length of the model***. That is, the heated length is identical to the total axial length, Z[CHAN.3], and the model does not include the unheated ends of the fuel rods within the assemblies. If the model includes unheated regions at the ends of the fuel rods, then the axial heat generation profile is NOT uniform, within the definition of the term in the COBRA-SFS input, even if the heat generation profile is uniform over the length of the actual fuel region of a rod. In such case, a value greater than 0 must be entered for NAXP, and appropriate input provided for the “nominally” uniform profile, with the input provided on OPER.16 and OPER.17 for that profile.)

Operating Conditions

This input defines the nominal operating conditions for the system. When the plenum model is not used, the input on OPER.2 defines the steady-state system pressure, inlet flow enthalpy or temperature, average heat generation rate, and flow or pressure drop boundary condition on the system. When the plenum model is used, this input defines the initial state of the system at the beginning of the steady-state iterative solution.

OPER.2 PEXIT,HIN(1),GIN,PDN,HOUT,DPS,FTOTAL
FORMAT(7F10.0)

Columns	Variable	Description
1-10	PEXIT	System pressure (psia). Used to calculate the boiling temperature limit, based on the PLIQ[PROP.2] array. (NOTE: This variable is not used when the fluid is a gas, and can be left blank in this input line.)
11-20	HIN	If IHH = 0 or 2 on OPER.1, average inlet (or initial) enthalpy (Btu/lbm). If IHH = 1 or 3, average inlet (or initial) temperature (°F). SPECIAL NOTE: When there are two fluid properties tables defined in input group PROP, the options for uniform inlet enthalpy (or temperature), specified with IHH = 0 [or 1] on OPER.1, may not be appropriate for the model, as this will result in channel inlet conditions for both fluids being assigned the same inlet enthalpy [or temperature]. See discussion of IHH options with OPER.1 instructions.
21-30	GIN	Average inlet (or initial inlet) mass flux (Mlbm/h-ft^2).
31-40	PDN	Average heat generation rate (MBtu/h-ft^3): • When the fuel model is used (NC>4 on RODS.1), PDN is based on the volume of the fuel (defined using fuel diameter, DROD[I], specified on RODS.4). • When the fuel model is not used (NC=0 or NC=1 on RODS.1), PDN is based on volume of the rod, (defined using rod diameter, DIA[I], specified on RODS.3). (NOTE: See instructions for special optional group REST, in Section 2.4.12.1 for an alternative method for specifying the heat generation per assembly.)

Columns	Variable	Description
41-50	HOUT	If IHH = 0 or 2 on OPER.1, exit enthalpy (Btu/lbm). If IHH = 1 or 3, exit temperature (°F). (NOTE: HOUT is used to define fluid conditions for reverse flow at the exit of any channel. If the upper plenum model is used, however, HOUT is applied only to reverse exit flow in channels of assemblies that are not connected to the plenum. Default is HOUT = HIN.)
51-60	DPS	Total system pressure drop (psia) (used only when ITDP = 1 or 2 on OPER.1, or if ITDPA[NASS] = 1 or 2 on CHAN.6 for one or more assemblies.)
61-70	FTOTAL	Total inlet flow rate (lbm/s) (used only when ITDP = 3 on OPER.1 or CHAN.6).

Nonuniform Operating Parameter Distributions

The input for lines OPER.3 through OPER.18 are read only if appropriate option flags have been set on OPER.1. Because of the large number of options that must be defined on OPER.1, there is great potential for confusion in interpreting the requirements for input lines OPER.3 through OPER.18. All of these input lines are in some sense “optional.” but may have become “required” due to the modeling options selected for a particular model. The user is advised to proceed with caution.

Inlet Enthalpy/Temperature Distribution

This input is optional; it is read only if IHH is specified as 2 or 3 on OPER.1.

OPER.3 (HINLET(I),I= 1,NCHANL)

Read only if IHH > 1 on OPER.1
FORMAT(12E5.0)

Columns	Variable	Description
1-5, 6-10, 11-15, etc.	HINLET(I)	If IHH = 2 on OPER.1, inlet enthalpy (Btu/lbm) for channel I. If IHH = 3 on OPER.l, inlet temperature (°F) for channel I. SPECIAL NOTE: For this input, NCHANL is the ***total number of channels in the model***; that is, the sum of the NCHANA[CHAN.5] entries for all NASSEM[CHAN.1] assemblies. This may be a very large number, for a typical full-symmetry detailed model of a spent fuel storage system, and may require hundreds of input lines. Use of spreadsheets or other automatic data generating software is highly

		recommended for the process of generating this input. (See diagrams of large system models in COBRA-SFS – Theory Manual Section 7.0 (Richmond et. al. 2021) and Section 3.0 of this report of this report for examples.)

Inlet Flow Distribution by Assembly

This input is optional; it is read only if IG is specified as 2 or 4 on OPER.1.

OPER.4 (ZDUM(I),I = 1,NASSEM[CHAN.1])

Read only if IG = 2 or 4 on OPER.1
FORMAT(12E5.0)

Columns	Variable	Description
1-5, 6-10, 11-15, etc.	ZDUM(I)	Inlet flow fraction for assembly I, normalized to the average mass flux, GIN on OPER.2. Within assembly I, the inlet flow for each channel II is calculated as F(II,1) = ZDUM(I)*[GIN*(l0)6/3600.]*A(I,1)

Inlet Flow Distribution by Channel

This input is optional; it is read only if IG is specified as 3 or 4 on OPER.1. The total number of assemblies that have specified inlet flow distributions is specified as NASSIN on OPER.5. Then OPER.6 and OPER.7 are read in sequence NASSIN[OPER.5] times to define the specified inlet flow for the channels in the individual assemblies.

OPER.5 NASSIN

Read only if IG = 3 or 4 on OPER.1
FORMAT(I5)

Columns	Variable	Description
1-5	NASSIN	Number of assemblies for which channel inlet flow distribution is to be specified using the input on OPER.6 and OPER.7.

OPER.6 NASS

Read only if NASSIN > 0 on OPER.5
FORMAT(I5)

Columns	Variable	Description
1-5	NASS	Identification number of an assembly with specified channel inlet flow distribution.

OPER.7 TERM1(II), II=NCHAN1,NCHAN2)

Read only if NASS > 0 on OPER.6
FORMAT(12E5.0)

Columns	Variable	Description
1-5, 6-10, 11-15, etc.	TERM1(II)	Inlet mass flow fraction for channel II of assembly NASS[OPER.6]; this value must be normalized to GIN on OPER.2. The mass flow fraction is used to define the channel inlet flow as F(II,l) = TERM1(II)*[GIN*106/3600]*A(II,1)

OPER.6 and OPER.7 are read in sequence NASSIN[OPER.5] times.

Assembly Heat Generation Distribution

This input is technically optional, since it is read only if NRPF is specified as greater than zero on OPER.1. However, in most spent fuel modeling applications, the fuel assemblies are unlikely to be at the same decay heat value, even in canisters that are nominally "uniformly loaded." Therefore, in practical terms, NRPF is generally greater than zero, and the user will need to specify the heating rate for each individual assembly. When this option is used, assemblies that do not contain fuel rods by definition have a heating rate of zero.

SPECIAL NOTE: For users who cannot relate to a heating rate specified in input units of MBtu/h-ft^3, there is an alternative available in Cycle 4 that allows this input to be specified in Watts. See instructions for group REST, in Section 2.4.12.1.

OPER.8 (PDNA(N),N = 1,NASSEM[CHAN.1])

Read only if NRPF > 0 on OPER.1
FORMAT(12E5.0)

Columns	Variable	Description
1-5, 6-10, 11-15, etc.	PDNA(N)	Heating rate for assembly N: if NRPF = 1, PDNA(N) is normalized to PDN on OPER.2. if NRPF = 2, PDNA(N) is the assembly heating rate in units of MBtu/h-ft^3 (PDN on OPER.2 is not used).

Transient Forcing Function Tables

The input on lines OPER.9 through OPER.15 can be used to specify forcing functions on the system pressure, inlet enthalpy or temperature, inlet flow rate or pressure drop, and the heat generation rate. Forcing functions on the inlet flow conditions are ignored for assemblies connected to a lower plenum model. They are also ineffective when using the zero-flow boundary condition option, specified with ITDP = 4 on OPER.1.

System Pressure Transient Forcing Function

This input is optional; it is read only if NP is specified as greater than zero on OPER.1.

OPER.9 (YP(I),FP(I),I = 1,NP)

Read only if NP > 0 on OPER.1
FORMAT(12E5.0)

Columns	Variable	Description
1-5, 11-15, etc.	YP(I)	Transient time (sec) for the I^{th} element of system pressure forcing function table. (NOTE: This array must be entered as monotonically increasing from time zero.)
6-10, 16-20, etc.	FP(I)	Fraction of steady-state system pressure at time YP(I), normalized to PEXIT on OPER.2.

Inlet Enthalpy or Temperature Transient Forcing Function

This input is optional; it is read only if NH is specified as greater than zero on OPER.1.

OPER.10 (YH(I),FH(I),I=I,NH)

Read only if NH > 0 on OPER.1
FORMAT(12E5.0)

Columns	Variable	Description
1-5, 11-15, etc.	YH(I)	Transient time (sec) for the I^{th} element of inlet enthalpy or temperature forcing function table. (NOTE: This array must be entered as monotonically increasing from time zero.)
6-10, 16-20, etc.	FH(I)	If IHH = 0 or 2, FH(I) is the fraction of inlet enthalpy at time YH(I), normalized to HIN entered on OPER.2. If IHH = 1 or 3, FH(I) is the fraction of inlet temperature at time YH(I), normalized to HIN entered on OPER.2.

Inlet Flow or Pressure Drop Transient Forcing Functions

This input is optional. OPER.11 and OPER.12 are read only if NG is specified as greater than zero on OPER.1. OPER.11 is read only once to define the time array to be used with all NGPRFL[OPER.1] forcing functions. OPER.12 is read NGPRFL[OPER.1] times to define each of the inlet flow or pressure drop forcing functions. The tables are numbered sequentially in the code, in the order in which they are read in on OPER.12. These sequence numbers are assigned to specific assemblies by the values entered for NPFVT(NASS) on CHAN.6.

OPER.11 (YG(J),J =I,NG)

Read only if NG > 0 on OPER.1
FORMAT(12E5.0)

Columns	Variable	Description
1-5, 6-10, etc.	YG(J)	Transient time (sec) for the J^{th} element of inlet flow or pressure drop forcing function tables. (NOTE: Must be entered as monotonically increasing from time zero.)

OPER.12 ((FG(J,L),J = 1,NG),L = 1,NGPRFL)

Read only if NG > 0 and NGPRFL > 0 on OPER.1
FORMAT(12E5.0)

Columns	Variable	Description
1-5, 6-10, 11-15, etc.	FG(J,L)	If ITDP = 0 or 2 on OPER.l, the inlet flow fraction at time YG(J) for the L^{th} table, normalized to GIN entered on OPER.2. SPECIAL NOTES for flow or pressure drop forcing functions: If ITDP = -1, the inlet flow forcing function[s] can be applied only to assemblies with ITDPA[IASS] defined as 0 or 2 on CHAN.6.) Note that if ITDP = -1, the pressure drop forcing function(s) can be applied only to channels in assemblies with ITDPA[IASS] defined as 1 on CHAN.6.) If ITDP = 1 on OPER.1, the pressure drop multiplier at time YG(J) for the L^{th} table, is normalized to DPS entered on OPER.2.

OPER.12 is read NGPRFL[OPER.1] times.

Heat Generation Rate Transient Forcing Functions

This input is optional. OPER.13 and OPER.14 are read only if NQ is specified as greater than zero on OPER.1. OPER.13 is read only once to define the time array to be used with all NQPRFL[OPER.1] forcing functions. OPER.14 is read NQPRFL[OPER.1] times to define the heat generation rate forcing functions. The tables are numbered sequentially in the code, in the order in which they are read in on OPER.14. These sequence numbers are assigned to specific assemblies by the values entered in NHFVT(NASS) on CHAN.6.

OPER.13 (YQ(J),J = 1,NQ) Read only if NQ > 0 on OPER.1
FORMAT(12E5.0)

Columns	Variable	Description
1-5, 6-10, etc.	YQ(J)	Transient time (sec) for the J^{th} element of heat generation rate forcing function tables. (NOTE: Must be entered as monotonically increasing from time zero.)

OPER.14 ((FQ(J,L),J = 1,NQ),L= 1,NQPRFL)

Read only if NQ > 0 and NQPRFL > 0 on OPER.1
FORMAT(12E5.0)

Columns	Variable	Description
1-5, 6-10, etc.	FQ(J,L)	The heat generation rate multiplier, normalized to PDN on OPER.2, at transient time YQ(J), for the L^{th} table.

*** OPER.14 is read NQPRFL[OPER.1] times. ***

Exit Enthalpy Transient Forcing Function

This input is optional; it is read only if NHX is specified as greater than zero on OPER.1.

OPER.15 (YHX(I),FHX(I),I = 1,NHX)

Read only if NHX > 0 on OPER.1
FORMAT(12E5.0)

Columns	Variable	Description
1-5, 11-15, etc.	YHX(I)	Transient time (sec) for the I^{th} element of exit flow conditions forcing function table. (NOTE: Must be entered as monotonically increasing from time zero.)
6-10, 16-20, etc.	FHX(I)	If IHH = 0 or 2, the fraction of exit enthalpy entered in HOUT on OPER.2. If IHH = 1 or 3, the fraction of exit temperature entered in HOUT on OPER.2.

Axial Heat Generation Profiles

Unless specified otherwise, the axial heat generation profile is assumed to be a uniform distribution along the total axial length Z, specified in group CHAN on CHAN.3. This default option is obtained simply by entering NAXP as zero on OPER.1. In most applications, however, the axial heat generation rate will not be axially uniform, even if it is uniform over the active length of the fuel. Typically, storage system models include structures above and below the active fuel length, such that Z on CHAN.3 encompasses more than merely the active length of the fuel region of an assembly. Therefore, the axial heat generation profile would be by definition non-uniform, as far as the COBRA-SFS code is concerned.

Nonuniform axial heat generation profiles can be specified as tables of relative axial power factors versus axial distance. This option is specified by setting NAXP on OPER.1 to the number of axial heat generation rate profiles needed for the model. Tables describing the axial heat generation rate distributions are then entered using the input on OPER.16 and OPER.17.

The axial heat generation rate profiles are numbered sequentially from 1 to NAXP, in the order they are entered on OPER.16 and OPER.17. If the option for the default uniform profile is used, NAXP is entered as zero (or blank) and the profile is automatically generated in the code and assigned the index number 1 (which is the index number that should be specified for each assembly, in group CHAN, on CHAN.6).

SPECIAL NOTE: If a uniform axial profile is required for some particular assembly (or assemblies) in a given model, but other assemblies have non-uniform profiles, such that NAXP > 0 on OPER.1, the user must explicitly enter the uniform profile using the input on OPER.16 and OPER.17, along with the input for the non-uniform profiles. The automatically generated uniform profile is available to the user only if it is the *only* axial profile needed for the model.

The particular profile to be used in an assembly is specified by input on CHAN.5, by the value specified for NAFLX(NASS).

OPER.I6 NAX(I)

Read only if NAXP > 0 on OPER.1
FORMAT(I5)

Columns	Variable	Description
1-5	NAX(I)	Number of pairs of entries in the I^{th} axial heat generation rate profile table.

OPER.17 (Y(L,I),AXIAL(L,I),L=1,NAX(I))

Read only if NAXP > 0 on OPER.1
FORMAT(12F5.0)

Columns	Variable	Description

1-5, 11-15, 21-25, etc.	Y(L,I)	Relative location (X/L) of the L^{th} element of axial heat generation rate profile I. (NOTE: Must be entered monotonically increasing from 0.0 to 1.0.)
6-10, 16-20, 26-30, etc.	AXIAL(L,I)	Relative heat generation rate at Y(I,L), normalized to average heat generation rate, in PDN on OPER.2.

*** OPER.16 and OPER.17 are read in sequence NAXP[OPER.1] times. ***

Non-Uniform Pressure Drop Boundary Condition

This input is optional. It is read only if NDPA is greater than zero on OPER.1. It is used to specify a separate pressure drop boundary condition for each assembly. This option can be used only with ITDP = 1 on OPER.1, or if some assemblies have ITDPA(NASS) = 1 specified on CHAN.5. When this option is used, the pressure drop entered in DPA on OPER.2 is ignored.

OPER.18 (DPA(I),I=l,NASSEM[CHAN.1])

Read only if NDPA > 0 on OPER.1
FORMAT(12E5.0)

Columns	Variable	Description
1-5, 6-10, etc.	DPA(I)	Pressure drop (psi) for assembly I. (NOTE: Supersedes DPS on OPER.2. This option cannot be used with the network model, flagged by NETWK > 0 on DRAG.1.)

2.4.12.1 Group REST – Special Option for Assembly Decay Heat Input

Group REST is a special addition to simplify interfacing COBRA-SFS model templates with UNF-ST&DARDS. The input for PDN on OPER.2 is another quaint artifact of COBRA history, and can be difficult to implement for spent fuel canisters and casks. When the REST option is used, the heat generation rate for each assembly is specified in Watts, and the standard method of using NRPF on OPER.1 and PDN on OPER.2 is overridden. When this optional input is used, NRPFF on REST.1 must equal NASSM on CHAN.1.

REST.1 AGROUP,NRPFF
FORMAT(A4,1X,I5)

Columns	Variable	Description
1-4	AGROUP	Enter REST.
6-10	NRPFF	Number of assemblies in the model. (NOTE: This MUST be the same value as specified in group CHAN for variable NASSEM on CHAN.1.)

REST.2 DHWATTS(N), N=1,NRPFF
FORMAT(12F5.0)

Columns	Variable	Description
1-5, 6-10, 11-15, etc.	DHWATTS(N)	Heat generation (Watts) in assembly N. SPECIAL NOTE: Enter zero (0.0) for assemblies that do not contain heat generating elements; that is, assemblies that do not consist of fuel rods. See discussion of COBRA-SFS definition of “assembly” for group CHAN and RODS input, in Sections 2.4.3 and 2.4.5.

*** REST.2 is read as many times as necessary to supply NRPFF values, with 12 entries per line. ***

2.4.13 Group CALC–Calculation Parameters

This group defines the calculation parameters needed to specify the manner in which the model will be solved. The input specifies the various convergence limits, damping factors, acceleration factors, and iteration limits required for the steady state solution. The simulation time and time step size for transients is also defined by this input. The general structure of this group reflects modeling and convergence concerns related to reactor core thermal-hydraulics, but additional convergence parameters have been added to specifically address the thermal behavior peculiar to spent fuel storage systems. This includes low-velocity gas flow, recirculation, and predominately radial heat transfer gradients.

Experience has shown that as a general rule, the lateral momentum solution (see flag ISCHEME on input line CALC.1) can be neglected in thermal analysis of typical spent fuel storage systems for normal operation. If there are concerns about the effect of lateral momentum for a particular application, sensitivity studies are recommended (i.e., running cases with ISCHEME = 1 and with ISCHEME = 0, and comparing the resulting peak temperatures and temperature distributions). If there is no substantial difference in the results, it is recommended that the model be run with the simpler ISCHEME = 1 option selected.

SPECIAL NOTE: For very low velocity gas flows in subchannel arrays, particularly for applications where convection heat transfer is severely inhibited, the option ISCHEME = 1 may be required, to prevent non-physical numerical

oscillations in the lateral flow solution from destabilizing the overall solution, or unnecessarily slowing convergence rates.

CALC.1 AGROUP,ISCHEME,ITSTEP,NODUMP
FORMAT(A4,1X,I5,5X,2I5)

Columns	Variable	Description
1-4	AGROUP	Enter CALC.
6-10	ISCHEME	Lateral momentum solution flag: = 0; lateral momentum equation solved for crossflows (default). = 1; lateral flows set to zero (no solution of lateral momentum equation).
11-15		BLANK
16-20	ITSTEP	Number of entries in variable time step table. (Default is zero, for uniform time step size throughout a transient.) (NOTE: This input is used only for transient calculations.
21-25	NODUMP	Flag for option to write a binary restart file on logical unit 8: = 0; create a restart file (default). = 1; do not create a restart file. SPECIAL NOTE: The restart file (tape8) can be very large for a typical spent fuel storage system model; if the user is quite confident that the restart option will not be needed for a given application, the recommended approach is to set NODUMP = 1.

Solution Damping and Acceleration Factors

Damping and acceleration factors used in the solution are entered on CALC.2. All factors are applied as

$$X^n = \alpha X^n + (1 - \alpha) X^{n-1}$$

where

X = calculation variable
α = damping or acceleration factor
n = current iteration number.

Convergence limits are defined in non-dimensional form so that the relative magnitude of the affected variable does not affect the rate of convergence.

CALC.2 TTIME,WERRY,FERROR,HERROR,QERROR,DAMPNG,ACCELY,ACCELF, ACCRHO,ACCELW,ACCROD,ACCAX,SLDAMP
FORMAT(13E5.0)

Columns	Variable	Description
1-5	TTIME	Total transient time (minutes) (NOTE: For a steady state calculation, specify TTIME = 0.0.)
6-10	WERRY	Inner loop convergence criterion for the recirculation solution scheme (default is 0.001).
11-15	FERROR	External axial flow convergence criterion, defined for the implicit axial momentum equation as the maximum allowable error for iterative axial flows. If the error is greater than FERROR, another iterative sweep of the entire bundle is made (default is 0.01).
16-20	HERROR	Convergence criterion for fluid enthalpy (default is 0.001).
21-25	QERROR	Convergence criterion for total energy balance (default is 0.001).
26-30	DAMPNG	No longer used; leave blank.
31-35	ACCELY	No longer used; leave blank.
36-40	ACCELF	Damping factor for iterative axial and lateral flows (default is 0.7).
41-45	ACCRHO	Damping factor for change in fluid density (default is 1.0).
46-50	ACCELW	Acceleration factor for the solid structure energy equation solution (default is 1.2).
51-55	ACCROD	Damping factor for the rod energy solution (default is 0.5).
56-60	ACCAX	Damping factor for axial energy rebalancing (default is 0.0, no axial energy rebalancing).
61-65	SLDAMP	Damping factor in the energy solution for slab conduction SPECIAL NOTE: Slab conduction damping is defined in the same manner as described above for other damping factors, but is implemented in the code such that different damping factors can be applied to different regions of the model. As a result, the input for SLDAMP is treated in a slightly different manner. The option for no damping is the default in the code, and is specified by SLDAMP = 0.0 in the input stream. The input value of 0.0 is automatically converted to a damping factor of 1.0.

Columns	Variable	Description
		However, a non-zero value for SLDAMP is directly equivalent to α in the above relationship. See instructions for CALC.5 and CALC.6 to define regions where this damping will be applied.

Time Step and Iteration Control

This input specifies the number of time steps for a transient, and the time step size, if the variable time step option is used. For steady state calculations, the number of time steps must be specified as zero. The number of iterations for the steady state solution is specified with NTRIES on CALC.3. This value is used for all time steps in a transient, as well.

CALC.3 NDT,NTRIES
FORMAT(2I5)

Columns	Variable	Description
1-5	NDT	Total number of time steps. (NOTE: Enter zero for steady-state calculations.)
6-10	NTRIES	Maximum number of external iterations allowed for any time step, including the steady state at time zero (default is 20).

The default in the code is a uniform time step size for a transient, calculated from NDT on CALC.3 and TTIME on CALC.2 as DT = TTIME/NDT. Nonuniform time step sizes are specified by setting ITSTEP > 0 on CALC.1, and entering a table of time step size versus time on CALC.4.

CALC.4 (YT(I),FT(I),I =I ,ITSTEP)

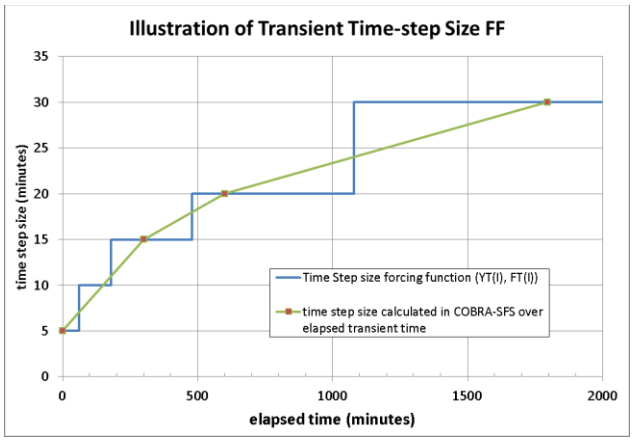
Read only if ITSTEP > 0 on CALC.1.
FORMAT(12E5.0)

Columns	Variable	Description
1-5, 11-15, 21-25, etc.	YT(I)	Elapsed transient time (minutes) when the first time step that is FT(I) minutes long begins.
6-10, 16-20, 26-30, etc.	FT(I)	Time step size (minutes) from time YT(I).

When this option is used, the time step size at any point in the elapsed transient time is calculated by linear interpolation in the table entered on CALC.4. A simple example illustrating this input and how it is interpreted in the code is shown in the following input table and graph.

Example: CALC.4 Input for Variable Time-step Size

YT(I)	FT(I)
elapsed time (minutes)	time-step size (minutes) beginning at YT(I)
0	5
300	15
600	20
1795.2	30



This option allows the user only limited control of the exact time-step size at any given point in the transient, but permits reasonable variation in the discretization of the solution in the dimension of time. It is useful for executing relatively long transients in a reasonable number of time steps, without the necessity of using the restart option. It is useful for transients of the type frequently encountered in vacuum drying operations, where rapid changes in boundary conditions or system response are anticipated in the initial stages of the transient, but in later stages temperature changes may be slow, often due to approaching a steady-state asymptote. Small time steps are advisable initially, but in general, time steps can be much larger later in the transient. It would also be possible to use this option to specify increasing and then decreasing time-step size over time, to represent cyclic drying transients.

Damping For Heat Transfer Solution of Slab Nodes

This input is optional, and is read only when SLDAMP is greater than zero on CALC.2. It allows the user to specify a damping factor on the heat transfer solution of specific slab nodes in a model. This is a particularly useful capability for cases where certain nodes are not well-coupled with the fluid energy solution. See Section 3.0 for further discussion of this option.

CALC.5 NSLGRP

Read only if SLDAMP > 0.0 on CALC.2.
FORMAT(I5)

Columns	Variable	Description
1-5	NSLGRP	Number of groups of slabs for which the conduction solution will be damped. (NOTE: For this input, a group is defined as a set of slabs with sequential identification numbers.)

CALC.6 (BSLGRP(I),ESLGRP(I),I= 1,NSPGRP)

Read only if NSLGRP > 0 on CALC.5.
FORMAT(12I5)

Columns	Variable	Description
1-5, 11-15, 21-25, etc.	BSLGRP(I)	The index number of the first slab in the I^{th} group of slabs with the conduction solution damped.
6-10, 16-20, 26-30, etc.	ESLGRP(I)	The index number of the last slab in the I^{th} group of slabs with the conduction solution damped.

2.4.14 Group OUTP–Output Options

This group allows the user to select the information to be written to the output file. The default is to print everything (i.e., all channels, rods, lateral flow connections, and solid structure nodes) at every axial level for the steady state solution and for every time step in a transient. For large models, or long-running transients, this can result in enormous output files.

The options in group OUTP allow the user to limit the output, either by restricting the number of channels, rods, lateral flow connections, or solid structure nodes for which output will be produced, or by limiting the frequency of output generated during a transient. This input allows considerable flexibility in specifying the output, so that the user can print only what is actually needed in the way of results.

The first five flags on line OUTP.1 are used to define the specific output desired for a given model. These flags determine what is written to the output file. However, a new option for output of cladding temperatures only, added to Cycle 4 with line OUTP.7, can modify the effect of these flags. See input instructions for OUTP.7.

OUTP.1 AGROUP,NCOUT,NROUT,NGOUT,NWOUT,NALL,NSKIPX,NSKIPT,
NPCHAN,NPROD,NPNODE,NPGAP,NPAVG,NPWALL,IPREB,IOTEMP
FORMAT(A4,1X,5I1,10I5)

Columns	Variable	Description
1-4	AGROUP	Enter OUTP.
6	NCOUT	Flag for channel output: = 0; no channel printout. = 1; channel printout controlled by NPCHAN.
7	NROUT	Flag for rod output: = 0; no rod printout. = 1; rod printout controlled by NPROD. SPECIAL NOTE: If ASHOT > 0 on OUTP.7, NROUT must be specified as 1, or the code will not be able to write out the special output files with cladding temperature only.
8	NGOUT	Flag for lateral flow connection output: = 0; no lateral flow connection printout. = 1; lateral flow connection printout controlled by NPGAP.
9	NWOUT	Flag for solid node output: = 0; no solid node printout. = 1; solid node printout controlled by NPWALL.
10	NALL	Flag for output of all information: = 0; output is governed by NCOUT, etc. flags. = 1; everything is printed, regardless of values assigned to other flags. = 2; same as 0, plus graphics dump for post-processing. = 3; same as 1, plus graphics dump for post-processing.
11-15	NSKIPX	Number of axial nodes to skip when printing results (default is 1, which results in every axial node being printed).
16-20	NSKIPT	Number of time steps to skip between printed results (default is 1, which results in every time step being printed).
21-25	NPCHAN	Flag for number of channels to be printed if NCOUT = 1. = 0; print all channels (default). > 0; print NPCHAN channels (see OUTP.2).
26-30	NPROD	Flag for number of rods to be printed if NROUT = 1: = 0; print all rods (default). > 0; print NPROD rods (see OUTP.3). SPECIAL NOTE: If ASHOT > 0 on OUTP.7, NPROD must be specified as 0; when this option is flagged, the user cannot limit the number of rods for which the code will write out the special output files with cladding temperature only.

Columns	Variable	Description
31-35	NPNODE	Flag for fuel temperature node printout, (used only if NROUT = 1); = 0; print only fuel centerline temperature, fuel surface temperature, and cladding surface temperature (default). > 0; print cladding surface temperature and all internal nodal temperatures (used only with the conduction model; NC > 4 on RODS.1).
36-40	NPGAP	Flag for number of lateral flow connections to be printed (if NGOUT = 1): = 0; print all lateral flow connections (default). > 0; print NPGAP lateral flow connections (see OUTP.4.)
41-45	NPAVG	Flag for assembly average and channel exit values to be printed: = 0; no additional output (default). = 1; channel exit values only. (Warning: Produces large volume of output information for typical spent fuel storage system models.) = 2; assembly-average values only. (Recommended; provides condensed summary of fluid solution results on a per-assembly basis, and can be particularly useful in the model checking process.) = 3; channel exit and assembly-average values.
46-50	NPWALL	Flag for solid node printout if NWOUT = 1: = 0; print all solid nodes (default). > 0; print NPWALL solid nodes (see OUTP.5).
51-55	IPREB	Flag for solution description output; = 0; normal iteration summary only (default). > 0; axial rebalancing summary and detailed iteration summary. (Warning: Produces large volume of output information for typical spent fuel storage system models.)
56-60	IOTEMP	Flag for output units on temperature. = 0; output in Fahrenheit = 1; output in Celsius.

User-Defined Output Options

The following input lines, OUTP.2 through OUTP.5, are optional. They are needed only if the corresponding options to limit the output to user-defined items have been specified by setting NPCHAN, NPROD, NPGAP, NPWALL or NSKIPT to non-zero values on OUTP.1. The default in the code is to print all information about all modeled structures.

Channels to be Printed

This input is optional; it is read only if NCOUT is 1 and NPCHAN is greater than zero on OUTP.1, but the user does not want to see output for ALL channels in the model. A total of NPCHAN pairs of assembly index number and channel index number must be specified on OUTP.2.

OUTP.2 (PRINTA(I),PRINTC(I),I = 1,NPCHAN)

Read only if NCOUT = 1 and NPCHAN > 0 on OUTP.1
FORMAT(16I5)

Columns	Variable	Description
1-5, 11-15, etc.	PRINTA(I)	Identification number of assembly containing channel PRINTC(I).
6-10, 16-20, etc.	PRINTC(I)	Index number of the I^{th} channel in assembly PRINTA(I) for which output will be printed.

Rods to be Printed

This input is optional; it is read only if NROUT is 1 and NPROD is greater than zero on OUTP.l, but the user does not want to see output for ALL rods in the model. A total of NPROD pairs of assembly index number and rod index number must be specified on OUTP.3.

OUTP.3 (PRINTA(I),PRINTR(I),1= 1,NPROD)

Read only if NROUT = 1 and NPROD > 0 on OUTP.1
FORMAT(16I5)

Columns	Variable	Description
1-5, 11-15, etc.	PRINTA(I)	Index number of assembly containing rod PRINTR(I).
6-10, 16-20, etc.	PRINTR(I)	Index number of the I^{th} rod in assembly PRINTA(I) for which output will be printed.

Lateral Flow Connections to be Printed

This input is optional; it is read only if NGOUT is 1 and NPGAP is greater than zero on OUTP.1, but the user does not want to see output for ALL lateral flow connections in the model. A total of NPGAP pairs of assembly index number and gap index number must be specified on OUTP.4.

OUTP.4 (PRINTA(I),PRINTG(I),I = 1,NPGAP)

Read only if NGOUT = 1 and NPGAP > 0 on OUTP.1
FORMAT(16I5)

Columns	Variable	Description
1-5, 11-15, etc.	PRINTA(I)	Index number of assembly containing lateral flow connection PRINTG(I).
6-10, 16-20, etc.	PRINTG(I)	Index number of the I^{th} lateral flow connection in assembly PRINTA(I) for which output will be printed.

Solid Nodes to be Printed

This input is optional; it is read only if NWOUT is 1 and NPWALL is greater than zero on OUTP.1, but the user does not want to see output for ALL solid nodes in the model. A total of NPWALL slab node index numbers must be specified on OUTP.5.

OUTP.5 (PRINTW(I),I=1,NPWALL)

Read only if NWOUT = 1 and NPWALL > 0 on OUTP.1
FORMAT(12I5)

Columns	Variable	Description
1-5, 6-10, 11-15, etc.	PRINTW(I)	Index number of I^{th} slab node for which temperatures are to be printed.

Transient Output Interval

This input is optional; it is read only if NSKIPT is greater than zero on OUTP.1. To define the output interval using only NSKIPT, enter zero for TRANT. Otherwise, output will be produced every NSKIPT time steps, or every TRANT seconds, whichever comes first.

OUTP.6 TRANT

Read only if NSKIPT > 1 on OUTP.1
FORMAT(F5.3)

Columns	Variable	Description
1-5	TRANT	Time interval (sec) between successive output time steps in a transient calculation (default is 0.0, which means that the output interval will be controlled by the value of NSKIPT alone).

The input for OUTP.7 is optional; it is read only if TTIME > 0.0 on CALC.2 or TTT > 0.0 on COBRA.2. That is, it is read only for transient calculations. This is a special feature developed for application to vacuum drying transients, but can be used for any transient where the user wishes to keep close track of the surface temperatures at all axial locations for all rods in the system.

The special output files are written in JSON syntax, for ease of data processing within UNF-ST&DARDS. This syntax means that the files contain data in ASCII format, allowing for them to be read using either JSON compliant codes, or alternatively, the data can be extracted using any standard editing tool. The naming convention for the files written by COBRA-SFS in the course of the transient is to concatenate the word “Transient” with the elapsed transient time. (NOTE: As currently defined, the only way to change this naming convention is by editing the appropriate lines of the COBRA-SFS source code. This feature is still under development, and may undergo revision with more user experience.)

As currently implemented, a vacuum drying transient of 100 timesteps, using time-varying time step size and extending over approximately 100 hours, would produce 101 special output files of the form;

Transient_____0.valid.json	contains all cladding temperatures for time zero (the initial steady-state condition at the beginning of the transient calculation)
Transient_____300.valid.json	contains all cladding temperatures at time 300 seconds (the first timestep of the transient, in this example)
Transient_____611.valid.json	contains all cladding temperatures at time 611 seconds (the second timestep of the transient, in this example)
and so on, to	
Transient__356395.valid.json	contains all cladding temperatures at time 356,395 seconds (98.998 hours; the final timestep in the example transient)

OUTP.7 ASHOT

Read only if TTIME > 0.0 on CALC.2 or TTT > 0.0 on COBRA.2
FORMAT(I5)

Columns	Variable	Description
1-5	ASHOT	Flag for special output files containing rod cladding surface temperatures, to be written to a separate file for each timestep of the transient. = 0; no special output files produced, cladding temperatures written to the normal output file only. = 1; all cladding temperatures written to the normal output file and to files containing the cladding temperature only. WARNING: For typical models, and transients of up to 100 timesteps, this can generate a huge output file. Be sure you really need the output file, and your system can handle it, before using this option. = 2; cladding temperatures written to special output files for each timestep; output of rod temperatures to the normal output file is suppressed.

2.4.15 Termination of Input File

After the input for all groups needed to define a particular model and application of the COBRA-SFS code have been specified, the user must tell the code that the input stream has come to an end, and it is time to begin processing it for output to the input file, and (eventually) execution of the solution. The end of the input stream is indicated by entering the endflag ENDD for the group input name.

SETIN.1 AGROUP
FORMAT(A4)

Columns	Variable	Description
1-4	AGROUP	Enter ENDD, to signal end of input stream for this case.

Congratulations! You can now start debugging your input file.

2.5 Input Instructions for COBRA-SFS Simplified Input (Beta)

In COBRA-SFS Version 6.0, development was started on the COBRA-SFS simplified input. This will be the first release of this new feature. The goal of this input is to give a greater degree of user friendliness and flexibility to the code. This new input file makes use of hardwired information for fuel geometry heat transfer correlations and material properties. Also, unlike the classic input file, the new input is not fixed format and is instead list-directed with space-delimited entries. Each line starts with a group identifier that labels the line for readability and allows the code to interpret the subsequent data correctly. This section lists the groups in the order that they should appear in the input file, along with brief descriptions of the data required and the data that the code fills in. Only currently available options are listed. In future versions, more input options will be added to increase functionality. A working understanding of the classic input structure and COBRA-SFS model building is recommended no matter which input file type is used. Currently, the new input functionality is limited to vertical PWR casks and contains a limited set of fuel designs and materials available.

In Version 6.0, the simplified input does not have all of the code options that have been used for code validation and should be considered beta and for information purposes only. More code development work will result in a more stable version of the simplified input. When the stability for different problem types and validation cases can be confirmed the new input will be considered the primary input file for COBRA-SFS. Over time, the classic input will be phased out in favor of the simplified input.

2.5.1 Group “cobra”

This group is the problem initialization and output file flags. By starting the input file with this line the code will recognize that the user is using the simplified input format.

Group Name	Problem Type Flag (Integer)	Output Type Flag (Integer)
cobra	0 = steady state calculation (NOTE: only steady state is available in Version 6.0)	1 = Standard Output File and All BOA outputs

2.5.2 Group “calc”

This group controls the calculational options of the code.

Group Name	Number of Axial Divisions (Integer)	Number of Steady State Iterations (Integer)
calc	Typical models need approximately 40 axial levels to converge. It is recommended that the user start with a smaller number (10-20) to speed up execution in the model testing stage then test the convergence behavior for final calculations.	Typical models need approximately 1000 iterataions to converge fully. If the total energy error converges before the iterations specified are reached the code will finish execution and print output

2.5.3 Group “case”

Group Name	Case name (string)
case	This case name is for user identification of model files. (NOTE: it does not need to match the input file names and will not affect the labeling of input and output files by the code)

2.5.4 Group “type”

This group identifies the type of model that is being analyzed. It controls the orientation of flow angles and whether different boundary conditions are needed from the user.

Group Name	Canister Type (string)
type	vvc = vertical ventilated canister system

2.5.5 Group “canister”

This group contains canister geometry information

Group Name	Canister internal diameter (in.) (real)	Canister internal height (in.) (real)	Canister Wall Thickness (in.) (real)
canister	For a canister system this is the internal diameter of the canister. For a bolted system enter the internal cavity diameter	For a canister system this is the internal height of the canister. For a bolted system enter the internal cavity height	For a canister system this is the side wall thickness of the canister. For a bolted system enter the cavity liner thickness.

2.5.6 Group “fill”

This group controls the canister gas environment. This includes the fill gas type and pressure.

Group Name	Fill gas type	Absolute pressure (atm) (real)
fill	He = Helium fill	If canister fill is specified by density the user will need to use an iterative calculation to determine the correct pressure entry

2.5.7 Group “atmosphere”

This group is used for ventilated casks and determines the environment that is seen through the annulus ventilation path.

Group Name	ventilation gas type	Ventilation gas temperature pressure (°F) (real)	Atmospheric Pressure (atm) (real)
atmosphere	air = air atmosphere	If canister fill is specified by density the user will need to use an iterative calculation to determine the correct pressure entry	This is used to adjust for altitude variations. Recommend value is 1 atm

2.5.8 Group “boundary”

This group specifies the temperature of the cask’s side and bottom. This group also includes the subgroup “side.” This is the equivalent input of types in BDRY Section 2.4.11.

Group Name	Side temperature (°F) (real)	Top Temperature (°F) (real)	Bottom Temperature (°F) (real)	Side Boundary Node Count (integer)
boundary	Side boundary temperature	Top boundary temperature	Bottom boundary temperature	Number of slab nodes connected to the side boundary
Group Name	**Boundary node id (integer)**	**Boundary node face length (in.) (real)**	**Side Surface Heat Flux (Btu/hr-ft^2)**	
side	ID number of node entered in a node group	The face length of the boundary node. The code multiplies this by the node height to determine the face area	Applied surface heat flux	

2.5.9 Group “basket”

This group is used to get general information about the basket and is used to inform the code of expected inputs elsewhere.

Group Name	Basket Cell Count (integer)	Basket cell width in. (real)	Basket Axial Length (in.) (real)	Downcomer Count (integer)	Mouse hole flag (integer)
basket	Number of basket cells containing fuel assemblies	Width of cells containing fuel (square cells area assumed in Version 6.0)	The axial length for the subchannel model	Number of non-rodded zones defined as assemblies in COBRA-SFS classic	0 = no mouse holes 1 = mouse holes

2.5.10 Group “solid_connection”

This group uses the same definitions as the SLAB.2 input in Section 2.4.6 to define the connection types for the solid structure model. In this case the “low” is referring to the lower numbered node and the high is referring to the higher numbered node

Group Name	Solid Connection id (integer)	Low-High Geometry Factor (unitless) (real)	High-Low Geometry Factor (unitless) (real)	Gap Resistance (s*ft*btu/) (real)	Low Emissivity (unitless) (real)	High Emissivity (unitless) (real)	Face Area (in.) (real)
solid_connection	Identification number for the solid connection enter from 1 to the total connection count	See COBRA-SFS classic input instructions for SLAB.2 RDIMA(IC)	See COBRA-SFS classic input instructions for SLAB.2 RDIMB(IC)	See COBRA-SFS classic input instructions for SLAB.2 RPAR(IC)	See COBRA-SFS classic input instructions for SLAB.2 EMA(IC)	See COBRA-SFS classic input instructions for SLAB.2 EMB(IC)	See COBRA-SFS classic input instructions for SLAB.2 ARAD(IC)

2.5.11 Group “node”

This group replaces the functions of the SLAB.3 input of COBRA-SFS classic in section 2.4.6. The node group defines the properties and geometry of a given node and connection are defined with the subgroup “solid_solid”. Each model will include a large number of node groups. All connections are defined from the lower numbered node to the higher numbered node, and the code automatically calculates the reciprocal connections.

Group Name	**Solid Node id (integer)**	**Material (string)**	**Cross-sectional Area (in^2) (real)**	**Heat Generation ($btu/hr\text{-}ft^3$) (real)**	**Solid Connection Count (integer)**
node	Enter from 1 to total nodes in the model	304ss = 304 stainless steel Carbs = carbon steel A1100 = Aluminum 1100 Cu152 = Copper Poisn = metal matrix composite poison plate (eg. Boral™, Metamic HT™) Resin = neutron absorbing resin Concr = concrete	This is used for transient calculations and heat generation	Recommend using 0.0 except for special modeling applications	Number of higher numbered nodes connected Defines number of solid_solid subgroups to be read
Sub Group Name	**Connected Node id (integer)**	**Connection type (integer)**			

solid_solid	Enter the higher numbered node that this node is connected to	Enter the connection type			

2.5.12 Group “assembly”

This group defines the basket cells and fuel assemblies in each position, including the type of fuel and power level associated with each fuel. The assembly must be defined by a boundary of 8 solid nodes. These nodes do not need to be concurrently numbered but are required to be entered in clockwise order from the top left of the fuel compartment.

Group Name	**Assembly id (integer)**	**Fuel type (string)**	**Assembly Power (W) (real)**	**Power Profile (string)**
assembly	Integer assigned to each fuel assembly region	WE17x17OFA = Westinghouse 17x17 pwr assembly	This is applied evenly across each fuel pin in the assembly	Standard = standard DOE limiting profile
Sub Group Name	**Wall node id (integer)**	**Wall node Thickness (in.) (real)**	**Emissivity (unitless) (real)**	
wall	Node id of the wall node for this assembly must be entered in clockwise order from the top left	Thickness of the wall node forming this assembly (NOTE: All thicknesses must be equal in Version 6.0)	(NOTE: All emissivities must be equal in Version 6.0)	

2.5.13 Group “downcomer”

This group defines the fluid geometry and wall view factors defining non-rodded “assemblies.” The subgroups wall and view are read several times based on the number of wall nodes in the downcomer. This section is used to replace the SLAB.5, SLAB.6 and RADG inputs of sections 2.4.6 and 2.4.8.

Group Name	**Downcomer id (integer)**	**Downcomer cross sectional area (in.^{2}) (string)**	**Wetted perimeter (in.) (real)**	**Heated Perimeter (in.) (string)**	**Wall Node Count (integer)**
downcomer	Number from 1 to total number of downcomers	Used to calculate flow area	Perimeter of all wall nodes	Only used if slab heat generation is used	Number of wall nodes forming the enclosure
Sub Group Name	**Wall node id (integer)**	**Wall node face length (in.) (real)**	**Wall node Thickness (in.) (real)**	**Wall Node Emissivity (unitless) (real)**	
wall	This corresponds to the node ids entered in the node group They may be entered in any order. It is recommend to follow a clockwise pattern.	This is used to calculate the area for radiation heat transfer			
Sub Group Name	**Wall node id (integer)**	**View Factor (real)**			
view	Radiation connections are read in the order of the wall group. Start entries with the node entered on wall for self view factor. Then enter view factors for the nodes down the list	Enter zero if there is no view factor to this surface			

2.5.14 Group “annulus”

This group defines the fluid geometry and wall view factors for the annulus region of a vertical ventilated cask. The subgroup’s wall and view are read several times based on the number of wall nodes in the annulus. In this case, it is required for the user to build the model such that that the annulus subchannels and wall nodes line up. For example, if there are six annulus

channels, the canister nodes must also be divided into six pieces. The nodes must then be numbered clockwise around the canister then clockwise around the annulus liner. This section is used to replace the SLAB.5, SLAB.6 and RADG inputs of Sections 2.4.6 and 2.4.8.

Group Name	**Annulus inner diameter (in.) (real)**	**Annulus Outer diameter (in.) (real)**	**Vent elevation distance (in.) (real)**	**Annulus channel count) (integer)**
annulus	Typically canister outer diameter	Typically overpack inner diameter	This is used to help calculate pressure drop	Number of channels that the annulus is divided into
Sub Group Name	**Wall node id (integer)**	**Wall node Thickness (in.) (real)**	**Wall Node Emissivity (unitless) (real)**	
wall	Slab node ID numbers corresponding to input group “node” that border the annulus region	Radial thickness of slab node	Emissivity of slab node surface that see’s the annulus region	
Sub Group Name	**Wall node id (integer)**	**View Factor (real)**		
view	Slab node ID of slab node that contributes to radiative heat exchange within the annulus	View factor value from “wall” slab node to “view” slab node		

2.5.15 Group “plenum_connection”

This is the equivalent input of thermal boundary connection types in BDRY Section 2.4.11

Group Name	Plenum connection id (integer)	C1 (Btu/s-in.^{2}-°F) (real)	C2 (1/°F) (real)	C3 (dimensionless) (real)	EM1 (unitless) (real)	EM2 (unitless) (real)
plenum_connection	Number from 1 to the total plenum connection count	See input instructions for COBA-SFS classic input group BDRY.2				

2.5.16 Group "bottom_plenum"

This is the equivalent input of thermal boundary connection types in BDRY Section 2.4.11

Group Name	Nominal axial plenum area (in.^{2}) (real)	Nominal radial plenum area (in.^{2}) (real)	Axial plenum region count (integer)	Radial plenum region count (integer)	Axial Boundary Heat Flux (Btu/hr-ft^2) (real)	Radial Boundary Heat Flux (Btu/hr-ft^2) (real)
bottom_plenum	This is the area that the area multiplier entries are based off	This is the area that the area multiplier entries are based off	Number of axial plenum regions to be modeled	Number of radial plenum regions to be modeled	Optional heat flux applied to the axial boundary	Optional heat flux applied to the outermost radial boundary
Sub Group Name	**Region id (integer)**	**Area multiplier (in.^{2}) (real)**	**Connection Type id (integer)**			
axial_region	This group is read from 1 to the axial plenum region count	Heat transfer area multipler normalized to the axial plenum area	Plenum connection type defined in plenum_connection group			
Sub Group Name	**Region id (integer)**	**Area multiplier (in.^{2}) (real)**	**Connection Type id (integer)**			
radial_region	This group is read after the	Heat transfer area multipler normalized to the axial plenum area	Plenum connection type defined in plenum_connection group			

2.5.17 Group “top_plenum”

Group Name	Nominal axial plenum area (in.^{2}) (real)	Nominal radial plenum area (in.^{2}) (real)	Axial plenum region count (integer)	Radial plenum region count (integer)	Axial Boundary Heat Flux (Btu/hr-ft^2) (real)	Radial Boundary Heat Flux (Btu/hr-ft^2) (real)
top_plenum	This is the area that the area multiplier entries are based off	This is the area that the area multiplier entries are based off	Number of axial plenum regions to be modeled	Number of radial plenum regions to be modeled	Optional heat flux applied to the axial boundary	Optional heat flux applied to the outermost radial boundary
Sub Group Name	**Region id (integer)**	**Area multiplier (in.^{2}) (real)**	**Connection Type id (integer)**			
axial_region	This group is read from 1 to the axial plenum region count	Heat transfer area multiplier normalized to the axial plenum area	Plenum connection type defined in plenum_connection group			
Sub Group Name	**Region id (integer)**	**Area multiplier (in.^{2}) (real)**	**Connection Type id (integer)**			
radial_region	This group is read after the	Heat transfer area multiplier normalized to the axial plenum area	Plenum connection type defined in plenum_connection group			

2.5.18 Group “end”

This group is entered to signal the end of the input file.

Group Name	
end	

2.6 COBRA-SFS I/O

Previously the input and output file structure made file management difficult during model development and production runs. Starting in Version 5.0, I/O will maintain backwards compatibility while increasing trackability and usability. All input files must be ASCII text files with no special characters or there may be errors in input reading. When the code is executed it will look for a command line argument that includes the case name. Additionally, a number 1 can be added to the command line after the case name to flag terminal output of the code. If no case name is provided, it will default to the old format. The output files will now all be created with a date stamp for trackability and to avoid inadvertently overwriting an existing output file. The input file version is determined automatically when the code starts.

Table 2-5 COBRA-SFS I/O Summary

File Type	Default	Version 6.0+ Format
Input	input	casename.input
Tape 10 view factors	tape10	casename.tape10
Output	output	casename_YYYYMMDD_HHMM.output
.out summary file	cobra.out	casename_YYYYMMDD_HHMM_cobra.out
Tape 8 restart file	tape8	casename.tape08

3.0 Model Development Guide

In general, a COBRA-SFS model is most efficiently developed from the inside out; that is, from the heat-generating fuel assemblies to the external boundary of the system, where that thermal energy is being deposited. It is therefore useful to begin by taking a long, close look at the design drawings[1], with particularly close attention to the cross-sectional diagrams, since heat removal from most systems or packages is almost entirely in the radial direction; that is, out the sides. Very few packages have significant heat transfer in the axial direction, such that in many cases, a convenient simplification of the modeling effort is to assume adiabatic conditions on the ends. This is a conservative assumption, in that it will tend to result in slightly higher temperature estimates, but it is not generally that large a conservatism, and can serve as a bounding assumption, particularly for initial scoping calculations.

The purpose of this preliminary study of the drawings is to obtain a clear overview of the general features of the design, and develop an engineering appreciation of its strengths and weaknesses as a “thermal machine.” This study should be undertaken with questions such as the following in mind:

- What are the main heat transfer paths available to move heat from the fuel assemblies to the outer boundary?
- What does the support structure for the basket or fuel tubes containing the fuel assemblies look like?
  - Thermal conduction through solid structures is likely to be an important mode of heat transfer in almost any design, so where are the bottlenecks?
  - Where might structural geometry turn a broad heat transfer road into a narrow path? How are adjacent structures fastened together?
  - What might happen to surface-to-surface contact conductance as a result of differential thermal expansion at operating temperatures?
- Other than in the fuel assemblies and the basket structure(s) enclosing them, where is thermal radiation likely to be important?
- If the package is oriented vertically, does the design permit development of a significant thermo-siphon natural convection recirculation pattern? That is, does the design allow gas flow to transport heat from the fuel rods as it flows upward through the fuel assemblies and then deposit that heat at or near to the outer wall as the gas flows downward through open channels between the fuel support structure and the wall of the cask or canister cavity?
- If thermo-siphon recirculation is not possible in the design, where else might natural convection possibly have a significant role? (Hint: if a design does not allow thermo-siphon, convection is unlikely to be a major mode of heat transfer. But it should not be discounted out of hand.)
- If the canister containing the fuel assemblies is enclosed within an overpack (either in a storage module or transportation cask), what does the interface between the canister and the overpack consist of? A small annular gap (possibly filled with helium)? A somewhat

[1] As-built drawings would be preferable, but modelers are seldom afforded such luxuries as knowing the actual dimensions of systems for which they are expected to produce precise and reliable temperature estimates.

large gap, partially filled with a liner shell (which may be aluminum alloy)? A large gap with a complicated geometry of its own?

- If the package is oriented horizontally, it is reasonable to expect significant convection heat transfer in any region of the system?

With this general overview of the thermal design in mind, it is can be a little easier to make informed decisions on the level of detail needed to appropriately represent the various significant pieces of the system in the individual steps of the process of model development. Modeling is, by definition, a process of simplifying the representation of a system, relative to its full complex reality. For spent fuel storage systems and transportation packages, knowing where the heat is going, and understanding the physical processes driving that heat transfer, can make the modeling simplification process more obvious and more rational, and might make some of the model-building steps easier to implement.

3.1.1 Modeling the Fuel Assemblies

The logical place to begin when developing a model for any spent fuel storage system or transportation package is with appropriate representation of the fuel assemblies. This generally constitutes the bulk of the input for groups CHAN and RODS, with a separate “assembly” input block for each fuel assembly in the basket. The standard numbering convention for the subchannel array and the rod array representing a fuel assembly in COBRA-SFS is illustrated in Figure 3.1 for a typical 17x17 assembly.

NOTE: The diagram in this figure is an index numbering map for the subchannel numbers and rod numbers required in groups CHAN and RODS, respectively; it is not a diagram of the model, and does *not* include any geometry information; the rod-and-subchannel grid shown in the figure is rectilinear only because it was generated in a spreadsheet. The rods (shown as black squares) and guide tubes (shown as grey squares) are actually circular in cross-section, and fuel assembly arrays are square, not rectangular.

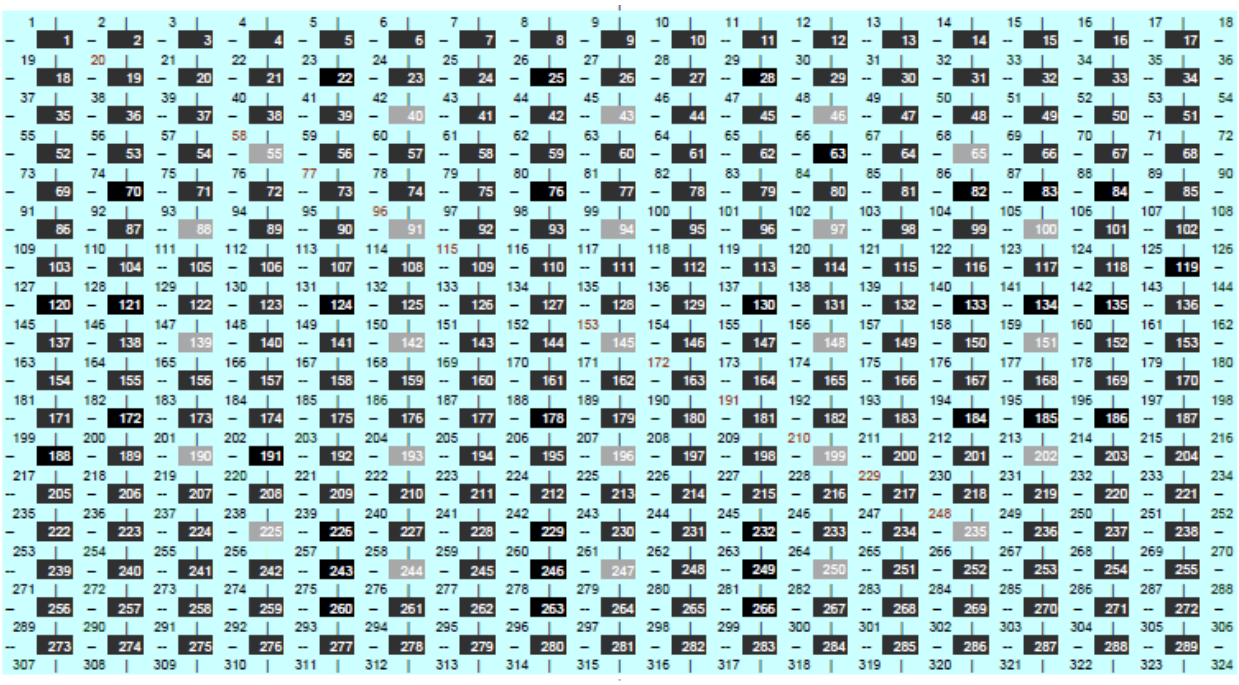


Figure 3.1. Map of Subchannel and Rod Indexing Scheme for Typical 17x17 Fuel Assembly, for Input Groups CHAN and RODS

This numbering convention for the array of rods and subchannels representing a fuel assembly is so engrained in COBRA-SFS modelers that it may not have been explicitly mentioned anywhere else in this document that this numbering scheme is *required*, if the model includes thermal radiation view factors generated with the RADGEN code (i.e., on tape10). The geometry input for group CHAN (flow area, perimeters, and gap connections), and group RODS (rod diameter and fraction of rod surface facing surrounding channels), is readily calculated from the fuel assembly dimensional information (i.e., rod diameter, array pitch, and control rod diameter, plus the internal width of the basket opening containing the fuel assembly). As can be discerned from Figure 3.1, there are only four distinct subchannel geometry “types” in the 17x17 array, consisting of

- the “corner” subchannels
- the “wall” subchannels (on the outer edges of the array “sides”)
- the interior subchannels formed by four adjacent rods (sometimes referred to as “matrix subchannels”)
- and the interior subchannels formed by three adjacent rods and a guide tube (sometimes referred to as ‘”thimble subchannels”)

The regularity of the subchannel array, and the limited number of unique subchannel geometries make constructing the group CHAN input for this type of assembly, regardless of array size, relatively easy, particularly with the aid of a spreadsheet program.

Figure 3.2 shows a map of the required indexing scheme for a CE14x14 assembly model, which follows a pattern similar to that of the 17x17 array, but with the significant difference of having five large guide tubes, each of which displaces a block of four rods within the fuel rod array. In this convention, the outsized guide tubes (which are round, not square, as conventialized in the

map) must be numbered “in line” with the first of the two rows of rods that each one affects. A model of a CE16x16 assembly would follow the same pattern, imposed on a 16x16 array. Similarly, the array numbering for BWR assemblies with large water rods must be numbered using the same convention, in a COBRA-SFS model.

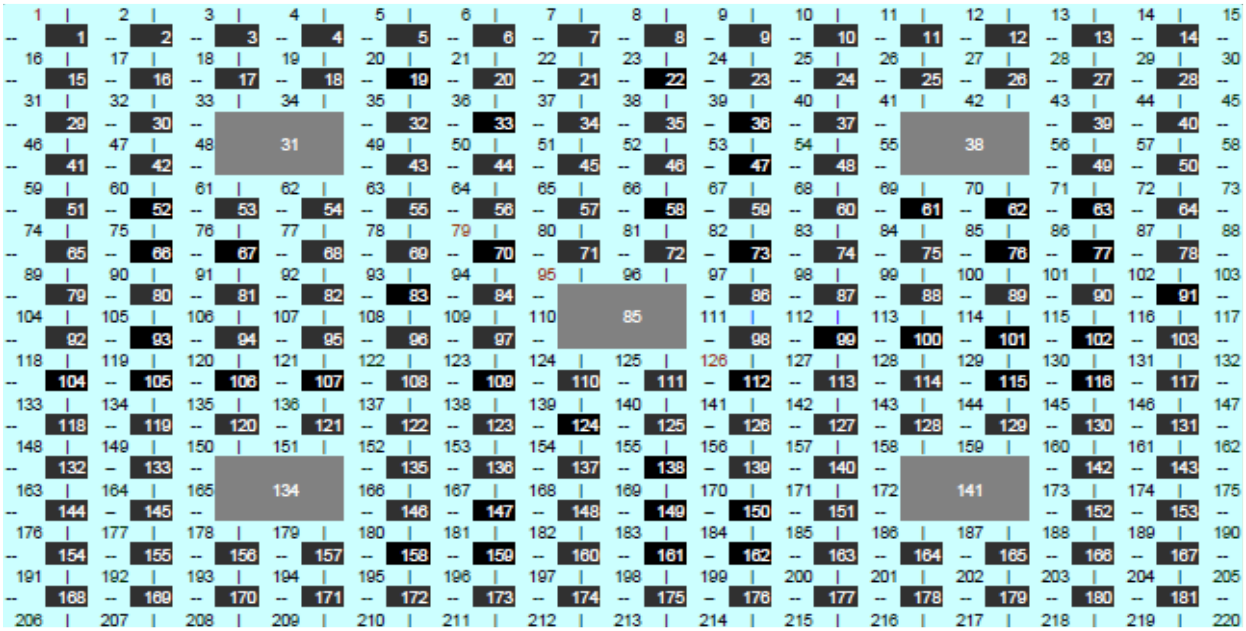


Figure 3.2. Subchannel and Rod Indexing Scheme for CE14x14 Fuel Assembly, for Input Groups CHAN and RODS

The geometry input for groups CHAN and RODS for the illustrated CE14x14 array is also amenable to generation using a spreadsheet, but is somewhat more complicated, due to the outsize guide tubes, which result in a larger number of different subchannel geometries and some local irregularities in the channel-to-channel connection pattern for subchannels surrounding the guide tubes.

The rod definition input in group RODS allows the model to account for heat generation in a fuel assembly on a rod-by-rod basis (see input instructions in Section 2.4.5). However, information on individual rod decay heat is seldom available for a given modeling effort, since it can be determined only if the detailed operating history and end-of-cycle burnup is available for each assembly. Such information is generally not available for thermal modeling applications, and the convention most commonly used is to assume that all fuel rods in a given assembly have the same decay heat. That is, the individual rods in the assembly are represented as having a decay heat fraction of 1.0, relative to the assembly average decay heat (see Section 2.4.5, input line RODS.3). Similarly, burnable poison rods within the array are represented with a decay heat fraction of 1.0, while rods representing guide tubes and instrument tubes are assigned a decay heat fraction of 0.0 in the group RODS input.)

The pin-by-pin radial distribution of decay heat in a spent fuel assembly is a function of the burnup history of the particular assembly. One specific example assessed recently for moderately high burnup fuel showed a variation of only about ±3% in the individual rod decay heat values, relative to an assumed uniform average decay heat per rod. Other examples have cited a peak to average variation on the order of 10-15%, which suggests that there is a wide range of variation in actual fuel assemblies with regard to the uniformity of pin-by-pin decay

heat. But regardless of how accurate this assumed uniformity might be for a specific fuel assembly, it is often the *only* assumption that can be made, based on available information. If that is the case, the modeler should bear in mind that this is an additional source of uncertainty in the thermal modeling, and will affect the range of uncertainty in any temperature predictions obtained with a model that must assume uniform radial decay heat in the fuel assemblies, due to lack of information on actual pin-by-pin decay heat.

Having developed a reasonable representation of the fuel assemblies for a model, the next item to consider is the appropriate representation of the solid nodes enclosing the fuel assembly. These nodes form the outer wall of the corner and edge subchannels, and are thermally connected to the fluid in the subchannels (see discussion of slab-fluid thermal connections in Section 2.4.6.)

3.1.2 Modeling the Structure Immediately Enclosing the Fuel Assemblies

There are basically two general types of basket design in spent fuel storage canisters and casks; an array of individual fuel tubes, each of which can accommodate a single fuel assembly, or a grid of relatively thin plates forming an “egg-crate” array of cells, with one fuel assembly per cell. (The CASTOR-V/21 basket, illustrated in COBRA-SFS – Theory Manual Section 7.4 through 7.6 (Richmond et. al. 2021), is an example of a basket consisting of fuel tubes. The TN-24P cross-section shown in COBRA-SFS – Theory Manual Section 7.7 through 7.9 (Richmond et. al. 2021) illustrates a basket consisting of thin plates.)

The basket and support structure are represented in a COBRA-SFS model with slab nodes, as discussed in Section 3.1.3, with guidance on appropriate definition and resolution of the geometry for efficient modeling. However, a first concern, before diving into the complexities of the slab noding to represent the entire basket, is proper representation of the slab nodes as seen by the fuel assembly rod-and-subchannel portion of the model. This portion of the model defines the interface between the fuel assemblies and the solid structure of the basket, and has some fairly specific requirements that must be satisfied, if thermal radiation is to be represented in a manner consistent with the modeling assumptions in COBRA-SFS.

As discussed in Section 2.4.7.1, the auxiliary code RADGEN can be used to generate grey-body viewfactors for thermal radiation within a fuel assembly and between the assembly rods and the enclosing walls of the basket. For a square-array fuel assembly, the COBRA-SFS model must have exactly 8 slab nodes comprising the wall nodes that are in contact with the fluid, and which can exchange energy with the rods by thermal radiation. Figure 3.3 illustrates this convention for a generic 17x17 assembly.

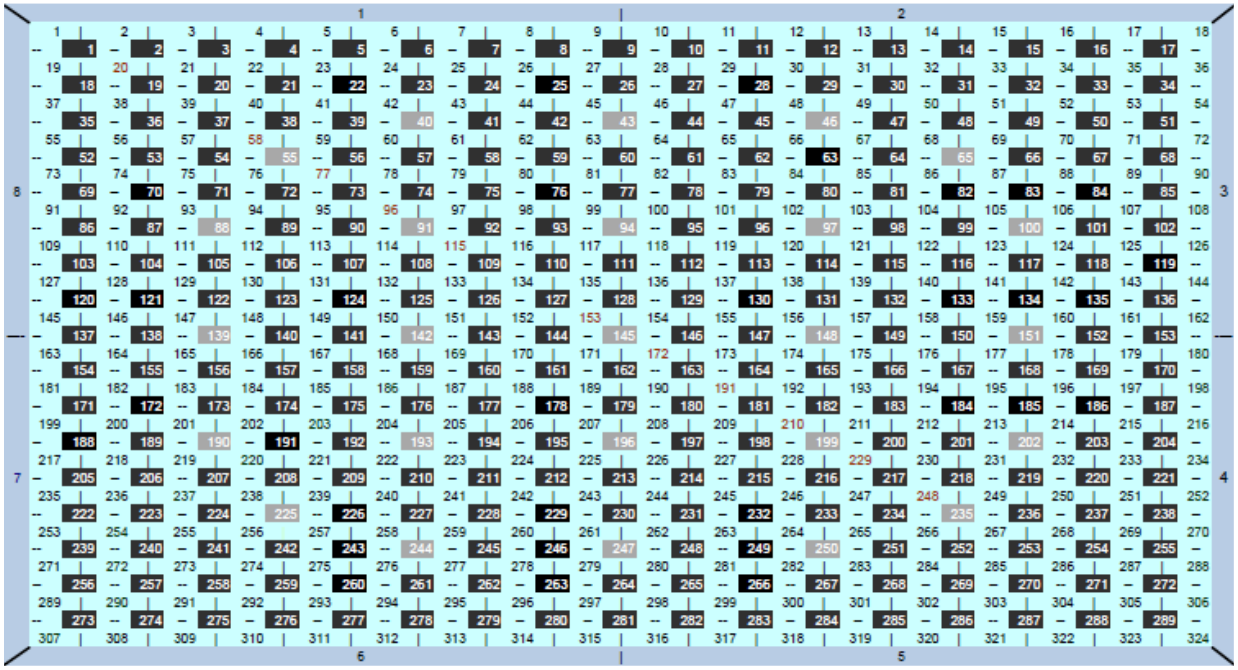


Figure 3.3. Map of Subchannel and Rod Arrays for Typical 17x17 Fuel Assembly, Illustrating Wall Nodes at Interface between Fluid and Solid Nodes in a COBRA-SFS Model

This is one of the steps in the model building process where it is possible to make subtle and frustratingly difficult to find errors in the COBRA-SFS input, mainly because there are two independent places in the input stream where the geometry of these 8 wall nodes must be defined consistently. These are:

- In the geometry input for the distance from the “edge” rods to the walls of the fuel enclosure, defined by input to the RADGEN code to generate the tape10 for the assembly (see Section 2.4.7.1).
- The geometry input for the slab-to-fluid connections defining the thermal connection(s) between the wall nodes and the fluid nodes (see Section 2.4.6, input line SLAB.5).

Inconsistency in the input in these two separate parts of the input stream can result in a potentially confusing error message from the code to the effect that something is wrong with the grey body viewfactor input from file tape10. This error message is the result of a check to ensure that the thermal radiation modeling within the code does not inadvertently create or destroy energy due to incompatible geometric input. It is in effect reporting that the surface area of the wall nodes as represented in the fluid-slab thermal connections input in group SLAB (as per SLAB.5) is not compatible with the surface area implicit in the specified input for the RADGEN code (used to generate the tape10). The error message does not actually say that, unfortunately. Instead, it says something about view factors not summing to 1.0, and prints out incompatible area ratios. The code developers at the time knew exactly what that meant; modelers must live and learn.

In most cases, the user will automatically define these separate input pieces correctly, if basing the input in both places on the geometry of the fuel assembly (i.e., rod pitch and fuel rod diameter) and the internal dimensions of the fuel tube or basket cell opening. The easiest way to get into trouble with this input is to attempt to capture the effect of eccentric positioning of the fuel assembly within the fuel tube or basket cell by defining non-uniform wall nodes for this 8-

node region. To avoid this problem, the modeler should follow three simple rules when defining the fluid-thermal input for the 8 wall nodes:

- Assume that the two wall nodes on a given side of the enclosure are the same length on the edge facing the fluid (i.e., the edge connected to the subchannels).
- Assume the pairs of wall nodes on opposite faces of the enclosure have the same total length in contact with the fluid in the adjacent subchannels (that is, match the lengths of nodes 1 and 6, 2 and 5, 3 and 8, 4 and 7, as labeled in the diagram in Figure 3.3).
- Capture any geometric variation due to eccentricity of the fuel assembly within the fuel tube or basket cell with the input for VIEW that defined the distance between the outermost row of rods and the wall on each side of the assembly (see COBRA-SFS – Theory Manual Section 6.0 (Richmond et. al. 2021)).

When this approach is used, the grey body viewfactors in the tape10 file will account for the correct path-lengths between the rods and the wall nodes, including the effect of any eccentricity of the position of the fuel assembly within the enclosure. No additional information is needed.

On the subject of the assumed position of a fuel assembly within a fuel tube or basket cell, it should be noted that in general, the modeler will have little or no information on this small detail of the system. The most reasonable approach for vertical storage systems is to assume that the assembly is exactly centered within the enclosure. This is obviously bounding, and may even be reasonably accurate, since the opening is generally only slightly larger than the footprint of the fuel assembly. Eccentricity of position in a vertical basket could be only small fractions of an inch, for any reasonable design. The effect of this assumption on uncertainty in modeling results would be easily swamped by other larger uncertainties in the overall modeling input, such as uncertainties or unknowns in the decay heat distribution and total decay heat of the assembly.

For a horizontal system, the eccentricity of position of the fuel assembly would be obvious from the geometry of the fuel assembly and enclosure dimensions, and would be readily captured in RADGEN input (as per rule #3 above). However, as a simplifying assumption, the fuel assembly is treated as centered axisymmetrically within the basket opening, with a cavalier disregard for the force of gravity. This is obviously bounding, and conservative, generally at very little cost of increased uncertainty of predicted temperatures for the fuel rods. As noted for the effect of eccentricity of position in a vertical basket, the effect of this simplification for horizontal systems is similarly swamped by other larger uncertainties in the overall modeling input, particularly those related to decay heat within the assembly.

These recommendations address the appropriate modeling of thermal radiation heat transfer between the fuel assembly and the enclosure wall nodes. Another possible mode of heat exchange is by radial conduction due to direct contact between the fuel assembly and the wall. This mode of heat transfer is neglected when modeling spent fuel storage systems with COBRA-SFS, for both vertical and horizontal systems. It is not included in the basic modeling capabilities of the code, for the following reasons:

- Information for realistic modeling is seldom available for such detailed representation of the geometry of the assembly and enclosure, in particular with regard to details of fuel assembly grid design, which would constitute the actual structures in contact with the walls of the enclosure. This information is generally proprietary to a given fuel vendor, and is not available to the modeler. Therefore it is virtually impossible to verify or validate the effect of

such possible direct contact on overall fuel cladding or system temperatures. The only rational approach is to neglect it, as an application of reasonable engineering judgement.

- Even in the rare case of having complete information on the geometry of a given fuel design, the fuel rods in an intact fuel assembly would not normally come in direct contact with the walls. Possible points of contact would very likely be limited to no more than “dimples” on the outer straps of the grid spacers of an assembly, which would include only a very small fraction of the total area of the outer “side” of a fuel assembly or the enclosure wall, and therefore would constitute at best only a minor heat transfer path.

The long history of V&V of the COBRA-SFS code for modeling of spent fuel storage systems in vertical and horizontal testing configurations, as discussed in COBRA-SFS – Theory Manual Section 7.0 (Richmond et. al. 2021) and Section 3.0 of this report, demonstrate conclusively that neglecting this potential heat transfer path is justifiable. Assuming the fuel assembly is geometrically centered within the fuel tube or basket cell is a reasonable modeling simplification, and does not adversely affect modeling results. The effect is well within the uncertainty due to other more significant inputs, such as the fuel assembly decay heat distribution and total decay heat load.

3.1.3 Modeling the Basket

The discussion in Section 3.1.2 is focused primarily on appropriately capturing heat transfer at the inner surface of the slab nodes in direct contact with the rod-and-subchannel region of the model. Once this 8-node piece of the model has been defined, there is still the rest of the basket to consider, and this can in many cases be the most labor-intensive and complicated part of the model-building effort. Section 2.3 describes typical COBRA-SFS models of several different storage systems, and the models constructed for the CASTOR-V/21 and TN-24P as part of the Cycle 4 validation are outlined in COBRA-SFS – Theory Manual sections 7.4 through 7.6 and 7.7 through 7.9 (Richmond et. al. 2021), respectively.

As these examples illustrate, there are basically two types of baskets; an array of relatively sturdy fuel tubes held in position with some type of support structure, or a more minimalist “egg-crate” grid of plates, usually consisting of several layers, often of different metals. But regardless of the general basket design, the first step in constructing the detailed network of interconnected slab nodes representing the basket is to ensure that it interfaces neatly with the requirements discussed above in Section 3.1.1 for the slab nodes in immediate contact with the fluid subchannels. The discussion in Section 3.1.2 focuses on the interface of these wall nodes with the fluid; this section focuses on how these wall nodes connect to the rest of the basket. If the basket is an array of fuel tubes, very often these wall nodes simply comprise the fuel tube itself. If the basket is a grid of plates, at least one or two of the sides will probably consist of a neutron absorber plate (aka “neutron poison plate,” or simply “poison plate”). Typical noding patterns for these two options are illustrated in Figure 3.4 and Figure 3.5.

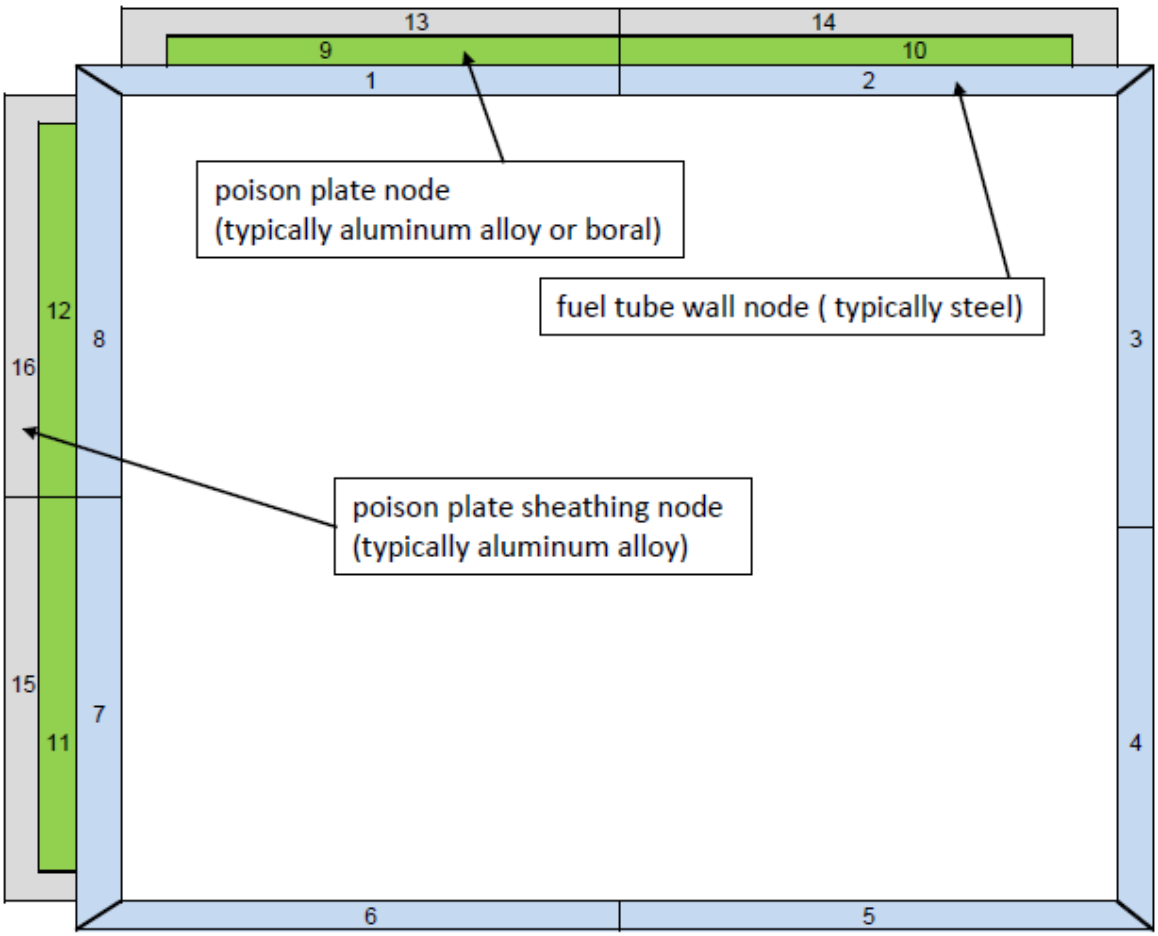


Figure 3.4. Typical COBRA-SFS Model Slab Noding Diagrams for a Fuel Tube with Poison Plate on Exterior Surfaces (NOTE: Diagram not to scale; plate thicknesses greatly exaggerated for clarity)

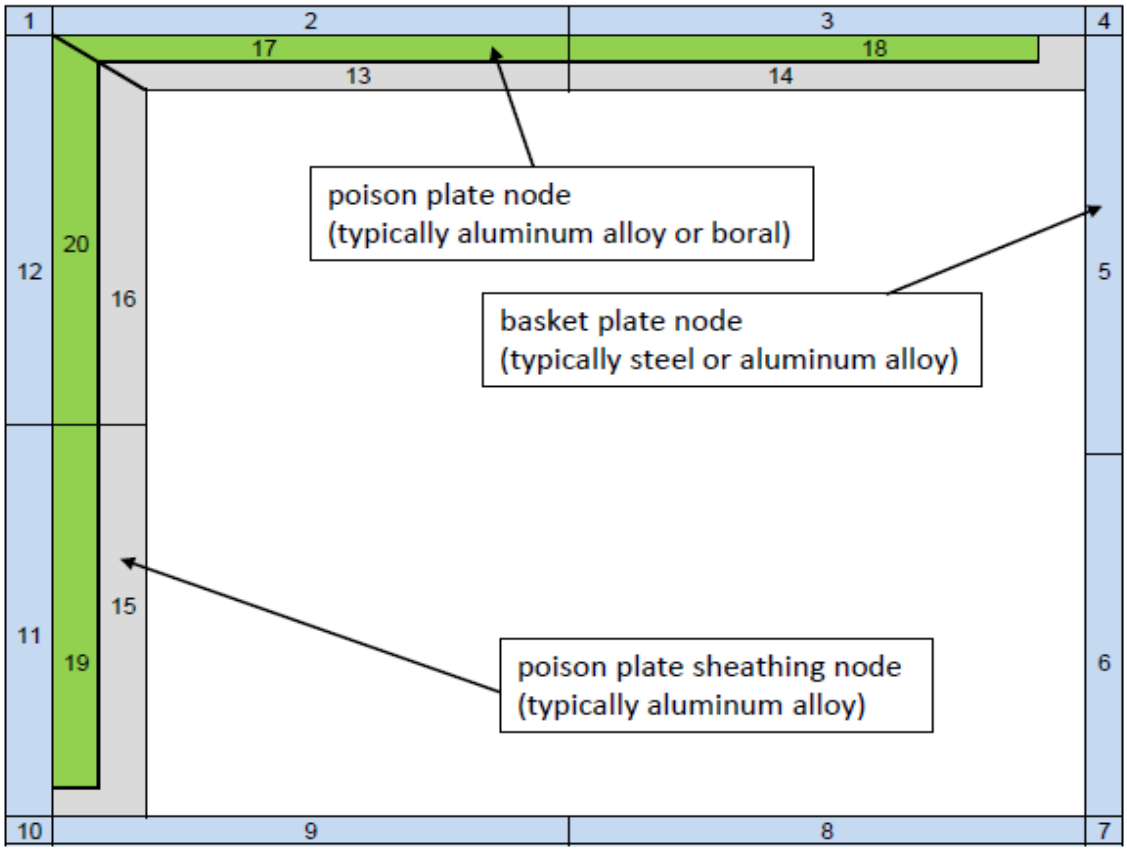


Figure 3.5. Typical COBRA-SFS Model Slab Noding Diagram for Basket Cell with Poison Plate on Interior Surfaces (NOTE: Diagram not to scale; plate thicknesses greatly exaggerated for clarity)

These two figures nicely illustrate how the thermal network modeling of the slab nodes allows the COBRA-SFS model to easily and economically represent adjacent thin plates. In particular, the poison plates are generally quite thin, and the aluminum sheathing (if included in the design) is generally even thinner. This poses a challenge to most CFD codes, and layers of thin plates are generally represented with a single layer of cells with averaged thermal properties, including a calculated effective thermal conductivity (which may be anisotropic) to account for the different materials and possible gaps between them. The COBRA-SFS modeling approach allows direct representation of each layer of material, and direct modeling of the effect of gaps between plates, by means of the thermal connections between adjacent slab nodes (see group SLAB input instructions in Section 2.4.6).

Comparing the examples shown in Figure 3.4 and Figure 3.5, it should be noted that the diagram in Figure 3.5 includes “corner nodes” in the basic square defining the basket cell (i.e., light blue nodes), but Figure 3.4 for the fuel tube does not. The corner nodes should generally be included in the representation of a grid-type basket cell as a typical modeling feature, to ensure adequate resolution of the temperature gradient along a given radial path in the basket grid. This can readily be seen in the diagram in COBRA-SFS – Theory Manual Figure 7.40 (Richmond et. al. 2021), illustrating the COBRA-SFS model of the TN-24P. Including these “corner nodes” in the basket essentially doubles the number of temperature points in any given center-to-edge run of the basket plates, and also automatically gives an appropriate temperature at the perpendicular intersections of the rectilinear basket plates. Less obvious

from the diagram, but often of significant importance in thermal performance of a basket design, including such corner nodes at the intersection point allows the model to take into account the effect of any potential gaps due to imperfect contact of the abutting plate. Inclusion or omission of the effect of imperfect contact at such intersections (by means of the contact conductance assigned to the slab-to-slab node connection, see Section 2.4.6) can have a significant effect on predicted (and actual) thermal performance of the basket design.

Such corner nodes can often be omitted in basket designs consisting of fuel tubes, since the tubes are not generally directly connected to one another, and there usually is not a direct heat transfer path from tube to tube, as is typical in an array of cells formed by basket plates. However, the COBRA-SFS model of the CASTOR-V/21 cask (see COBRA-SFS – Theory Manual Figure 7.8 (Richmond et. al. 2021)) includes corner nodes for most of the fuel tubes. The model in this figure was constructed by an experienced COBRA-SFS modeler, with a clear understanding of the importance of capturing as accurately as possible the complicated geometry of the plate interconnections of the basket support structure, and the resultant complicated interconnections of heat transfer paths in this cask. This is where a thorough understanding of the thermal behavior of the system being represented is needed, to make intelligent decisions regarding the level of detail to include in the network of slab interconnections representing the basket.

3.1.4 Basket Support Structures

Once the noding of the basket has been developed, the modeler is confronted with the question of how to represent the structures that interface the basket to the cask or canister inner wall. This can present interesting modeling challenges due to the geometry of the interface, and because the model may need to accommodate appropriate fluid flow paths between the basket and the cavity wall for backfill gas recirculation, as well as account for direct conduction from the basket to the inner wall of the cavity. The number of different approaches developed for basket support structures in cask and canister designs is large enough to warrant some breakdown of the discussion of how to model them. Section 3.1.4.1 discusses some important points for baskets consisting of fuel tubes. Section 3.1.4.2 discusses some of the variety of supports encountered in baskets consisting of a grid of plates.

3.1.4.1 Support Structures for Fuel Tube Baskets

For baskets consisting of fuel tubes, there are two basic approaches to the design of the support structure; one quite complicated and one comparatively simple. The complicated approach is illustrated by the array of interconnected plates and flux traps in the CASTOR-V/21 basket. In contrast to this complexity, there is also a comparatively simple design used by some vendors that supports the array of fuel tubes with solid discs spaced along the axial length of the tubes. Fuel tube baskets that have complicated plate support structures such as exemplified in the CASTOR-V/21 cask generally also have relatively unique structural connections to the cask or canister wall. They also often have significant paths for fluid recirculation that must be considered in the model. In many cases, the modeling concerns for these structures would be similar to those for some types of plate baskets, and these designs are therefore discussed in Section 3.1.4.2.

In fuel tube baskets with disc supports, as represented by some DSC configurations for NUHOMS horizontal storage modules and some NAC vertical storage modules, the fuel tubes are supported within up to two dozen or more steel discs, typically on the order of 3 to 4 inches

thick, spaced every 4 to 6 inches along the axial length of the tubes. The appropriate representation of the discs with slab nodes is somewhat more challenging, compared to the representation of layers of basket plates, simply from the standpoint of geometric calculations, as well as the concern to appropriately represent thermal gradients. Figure 3.6 shows a diagram of the disc plate noding[1] developed for the DSC-24P canisters stored at the Rancho Seco Independent Spent Fuel Storage Installation (ISFSI). The axial representation of this design requires utilizing the variable axial noding feature (see Section 2.4.4, input group VARY) in the COBRA-SFS code, and also the option for axially varying material properties (see Section 2.4.6, input group SLAB), to appropriately capture the stack of steel discs separated by layers of helium gas.

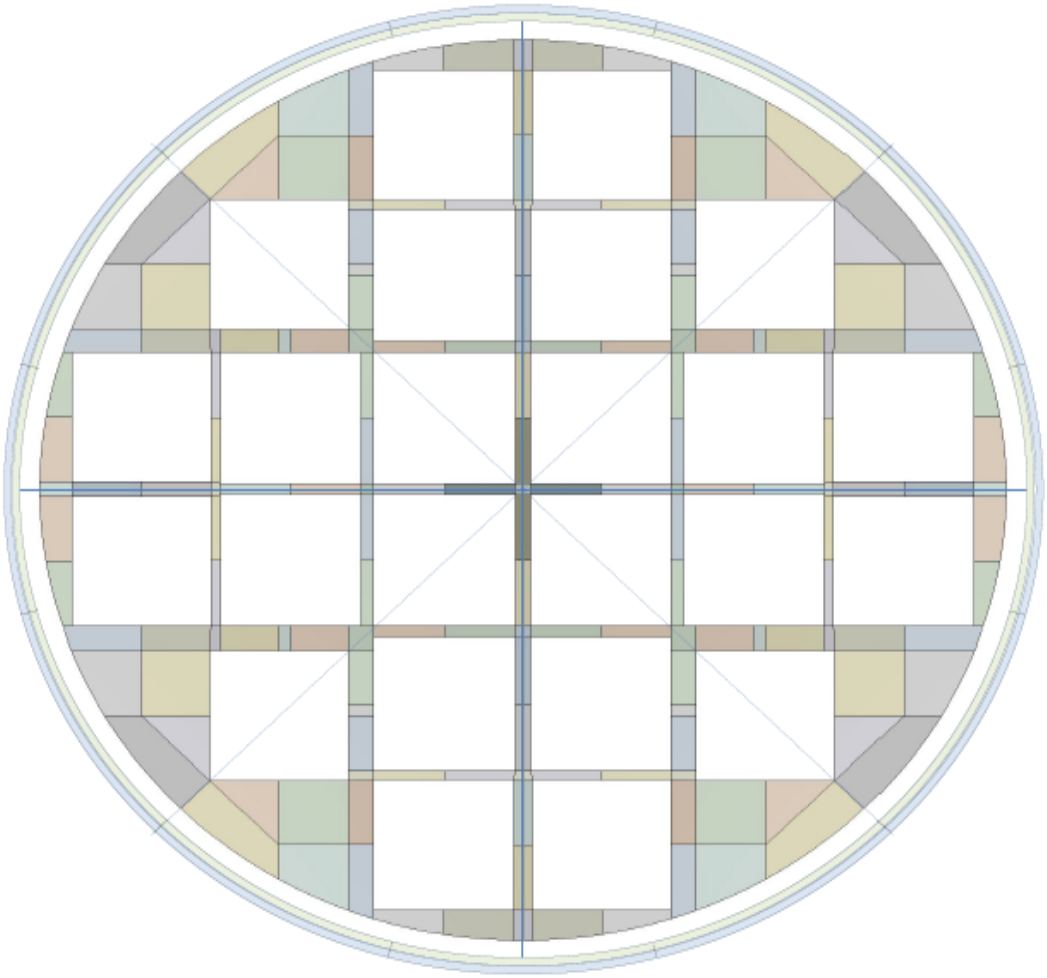


Figure 3.6. Example Illustrating Slab Noding for Support Discs in COBRA-SFS Model of NUHOMS 24P DSC Design

The interface with the canister wall is extraordinarily simple in this design; it consists of the discs resting against the inner wall. For horizontal storage, the appropriate connections can be defined by a simple application of Euclidian geometry, with some assistance from structural considerations of the relative compliance of the canister shell and the steel discs. For vertical

[1]This model is available as a template for UNF-ST&DARDS, and the inexperienced modeler probably will not have to deal with such a challenging geometry. Not on the first try, anyway.

storage, the connection can be most easily represented with the bounding assumption that the discs are axisymmetric within the cylindrical cavity, with a gap width determined by the difference in diameter of the discs and that of the inner cavity (typically on the order of a few fractions of an inch).

The illustration in Figure 3.6 shows the bounding assumption of axisymmetric positioning, which can also be assumed as an unrealistic but bounding case for horizontal storage as well as vertical storage. The thermal connections between slab nodes representing the disc and those representing the canister inner wall should include conduction through the backfill gas filling the gap, and thermal radiation across the gap.

For vertical orientation of a disc-supported basket, the presence of the discs precludes any possibility of significant axial recirculation of the backfill gas. There can be some minor recirculation within the individual fuel tubes (in the subchannel arrays of the fuel assemblies), but no special effort is required to represent this type of flow in the COBRA-SFS model; it is automatically captured in the conservation equations (mass, momentum, and energy) for the subchannels.

For horizontal orientation of this type of cask, it is reasonable to suppose that there might be some natural recirculation of the backfill gas sandwiched between adjacent discs. However, the essentially discontinuous fluid regions between the discs cannot be readily modeled in COBRA-SFS as fluid flow channels. The recommended approach for modeling this region is to treat the helium backfill gas as part of the solid slab nodes comprising the slab noding for the discs (as illustrated in Figure 3.6), and utilizing the option for variable material properties *in these slab nodes* to represent the interleaved layers of gas and metal.

There are two main thermodynamic compromises in this modeling approach. Effect of natural circulation within the disc-to-disc gaps is not captured in the model. Thermal radiation from disc-to-disc is not captured. The thermal paths through discs and gas alike are treated as conduction only. Fortunately, heat transfer in this region is very likely to actually be dominated by conduction. The node surfaces facing across the relatively narrow spacing between discs will be at similar temperatures for any given pair of discs, which would tend to minimize the importance of thermal radiation as a mode of heat transfer. Similarly, the local temperature gradients available to drive natural convection would be expected to be relatively small, such that gas velocities would be relatively low. Natural convection would not necessarily be totally negligible under such conditions, but it would not be expected to dominate the heat transfer behavior of the system.

When developing models of horizontal storage canisters such as NUHOMS DSCs for UNF ST&DARDS templates, the analysis approach utilizes boundary temperatures on the outer shell defined from a database designed to capture the effect of storage in a horizontal module. The boundary temperatures are defined for a large range of boundary conditions (e.g., total decay heat and ambient air temperature) based on calculations with a detailed CFD model of an entire NUHOMS storage module that includes a detailed representation of the canister internals. The CFD model used to develop the canister outer shell temperature database fully accounts for backfill gas recirculation between the discs and thermal radiation between the discs, the outer surfaces of the fuel tubes, and the canister wall. When this database is used to define the outer shell surface temperatures in a COBRA-SFS model, the boundary conditions in effect compensate for potential effects of the modeling compromises in the COBRA-SFS representation of the canister internals.

3.1.4.2 Support Structures for Plate Grid Baskets

Plate grid baskets fall into two general categories, when considered in terms of the basket support structure; those with simple line-contact supports at fairly widely spaced locations around the periphery of the basket, and those with structures referred to as “support rails” that effectively fill the space between the basket periphery and the cavity wall. Due to these significant differences in geometry, there are significant differences in the thermal performance of these two types of support structures.

Line-Contact Support Structures

In general, baskets with line-contact support structures are designed for vertical orientation only, and the support structures are not intended to support the weight of the basket and fuel assemblies in a horizontal orientation. The CASTOR-V/21 is an exception, in that it is a dual purpose package, designed for horizontal transportation as well as vertical storage, but the general character of the support structures, as far as it matters to heat transfer, is essentially the same. These line-contact supports are generally thermal bottlenecks to heat transfer from the basket structure to the wall, and the structures define relatively open regions of backfill gas between basket plates and the wall. Typical geometries for line-contact supports are illustrated in Figure 3.7 and Figure 3.8.

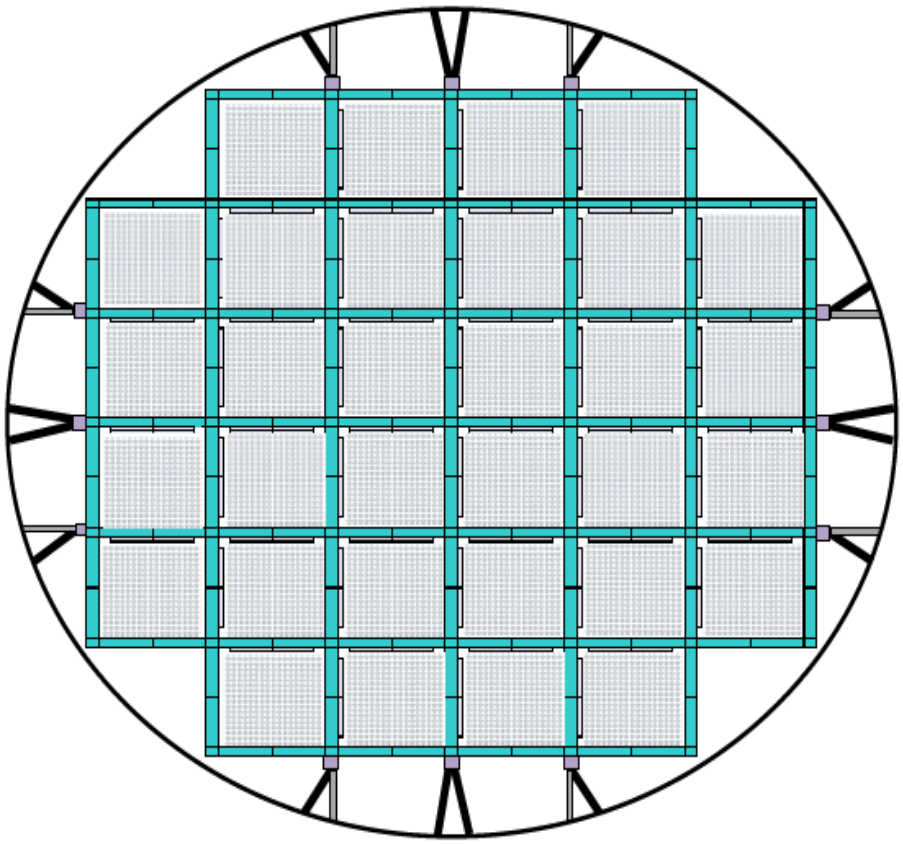


Figure 3.7. Illustration of Noding Diagram for Typical Point-contact Basket Supports for Vertical Storage Canister

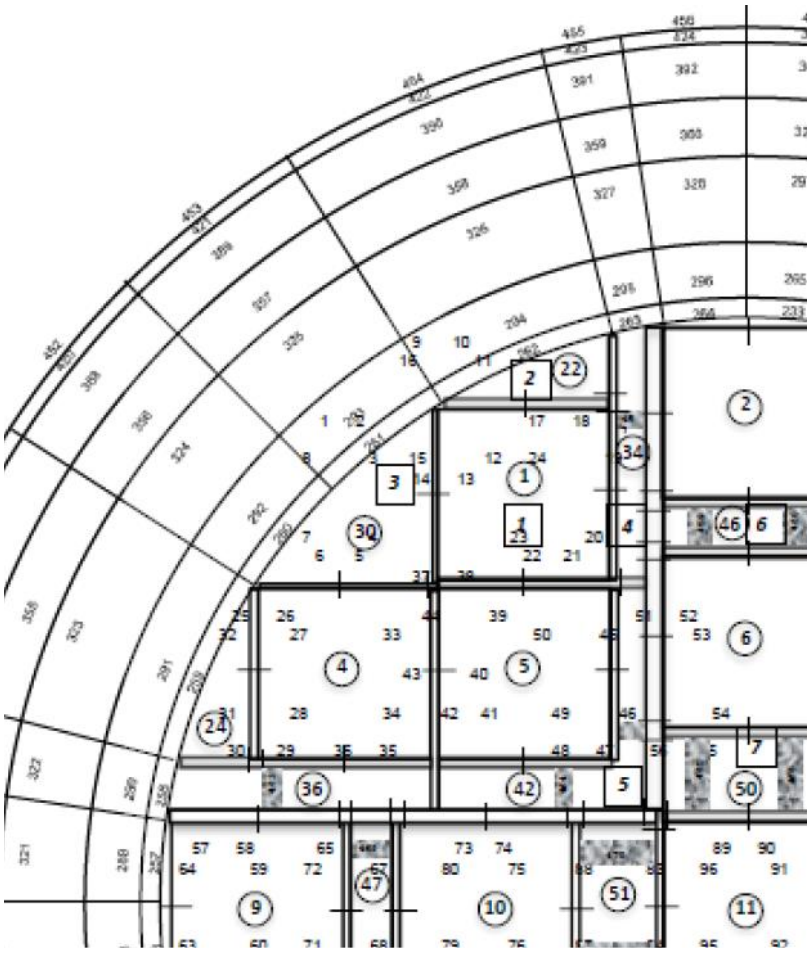


Figure 3.8. Illustration of Segment of Noding Diagram for CASTOR-V/21, Showing Wall Contact Capable of Supporting Basket in Vertical or Horizontal Orientation

In both configurations, as illustrated in Figure 3.7 and Figure 3.8, the COBRA-SFS model must appropriately accommodate the contact conductance at the locations where the plate ends touch the wall. In these examples, the support structures are not physically attached to the cavity wall, but are shimmed “as necessary” to ensure a snug and stable fit of the basket within the cavity. The appropriate thermal resistance to represent this sort of physical contact depends on factors that are generally unknown to the modeler, and may in fact be effectively unknowable, since as-built and final loaded configuration information is almost never available for thermal modeling purposes. Therefore, it is necessary to estimate the thermal resistance of the contact, based on the geometry of the connecting structures and reasonable engineering judgement.

The maximum possible area of contact generally can be determined reasonably accurately from dimensions on the relevant drawings for the particular system, but the contact pressure at those points is very difficult to even guess at in most cases, particularly with the notation requiring that the structures should be shimmed “as necessary.” This is a notation for materials and fabrication constraints, and it is highly unlikely that thermal effects were considered, except possibly with a dismissive nod to thermal expansion considerations.

From a thermal modeling standpoint, it is generally assumed that unless contact pressure is very high, (i.e., in the range of thousands of pounds per square inch), there will not be “perfect” contact between structures that are nominally touching each other. This also applies to the contact between the basket plates and the support structure, as illustrated in Figure 3.7, in addition to the contact between the support structure and the cavity wall. Therefore, the heat transfer path cannot be treated as a continuous material on either the inner or outer side of the support structure. In the absence of more accurate information, the recommended modeling approach is to assume that a small gap exists between the nodes representing the support structure and the basket plate, and between the support structure and the wall.

A conservative and possibly bounding value for this gap is typically cited as 0.01 inch, and the contact resistance of the gap is defined to include conduction through the backfill gas filling the gap. Optionally, the contact resistance can also include thermal radiation across the gap. This may or may not be significant, depending on the temperature gradient between the adjacent nodes, but it does no harm to include it. If the temperature difference between the adjacent nodes is not significant, including it will not affect the overall thermal results. If it is significant, omitting it will. Sensitivity studies may be a good idea, depending on the purpose of a particular modeling effort.

As the noding diagrams in Figure 3.7 and Figure 3.8 show, the point-contact support structures result in relatively large open areas between the basket periphery and the cavity wall. These open regions can serve as the downward flow paths for thermo-siphon recirculation in the cavity. (These regions are sometimes referred to as “downcomers,” in a fuzzy sort of parallel with reactor vessel hydrodynamics, even though the downflow paths in the cavity are not in the strict sense of the term designed as actual downcomers.) These flow paths are typically represented in a COBRA-SFS model as an array of one-dimensional flow paths that allow hot backfill gas to flow downward from the upper plenum to the lower plenum, transferring heat by convection to the outer wall of the cavity.

In addition to this convection heat transfer, which is the major component of the thermo-siphon effect that so efficiently moves heat from the fuel assemblies to the cavity wall in some canister designs, the COBRA-SFS model must include thermal radiation from the outer surfaces of the basket plates to the cavity wall. (See Section 2.4.7.1 for discussion of how to use VIEW/RADGEN to determine black body view factors for open enclosures of this type[1] in a COBRA-SFS model.) Thus, in the region between the basket structure and the cavity wall, all three modes of heat transfer – conduction through the solid structures of the supports, convection due to the flowing fluid in the open paths formed by the nodes of the basket, supports, and wall, and thermal radiation across the open regions – is captured in the COBRA-SFS model. All three modes are needed to appropriately represent the heat transfer behavior of this type of system, and to obtain reliable and reasonably realistic predictions of thermal behavior of the system (within the constraints of input uncertainties), including peak fuel cladding temperatures within the basket.

[1] The modeler also has the option of hand-calculating the black body view factors for thermal radiation within an enclosure defined by the slab nodes modeling the enclosure walls, and entering them directly into the code input stream, using input options in group RADG (see Section 2.4.8). This can afford an opportunity for the modeler to finally make professional use of the beauties of Euclidian geometry. For those who do not recall high school geometry class with any great fondness (or even for those who do), the chapter on mensuration formulae in the CRC Handbook *Standard Mathematical Tables* is highly recommended as an invaluable aid in such an endeavor.

Support Structures

The rather delicate support structures illustrated in Figure 3.7 are, as noted in the discussion, generally designed for vertical storage canisters. For horizontal storage or transport, more substantial support structures are needed, in order to support the full weight of the basket structure and fuel assemblies. An example of supports of this type is illustrated in Figure 3.9. These types of supports are sometimes termed “support rails,” but terminology is not consistent across vendors, or even within the full range of a given vendor’s products. (For simplicity, the term “rail support structures” will be used, to compare and contrast with the “line-contact supports” discussed previously.)

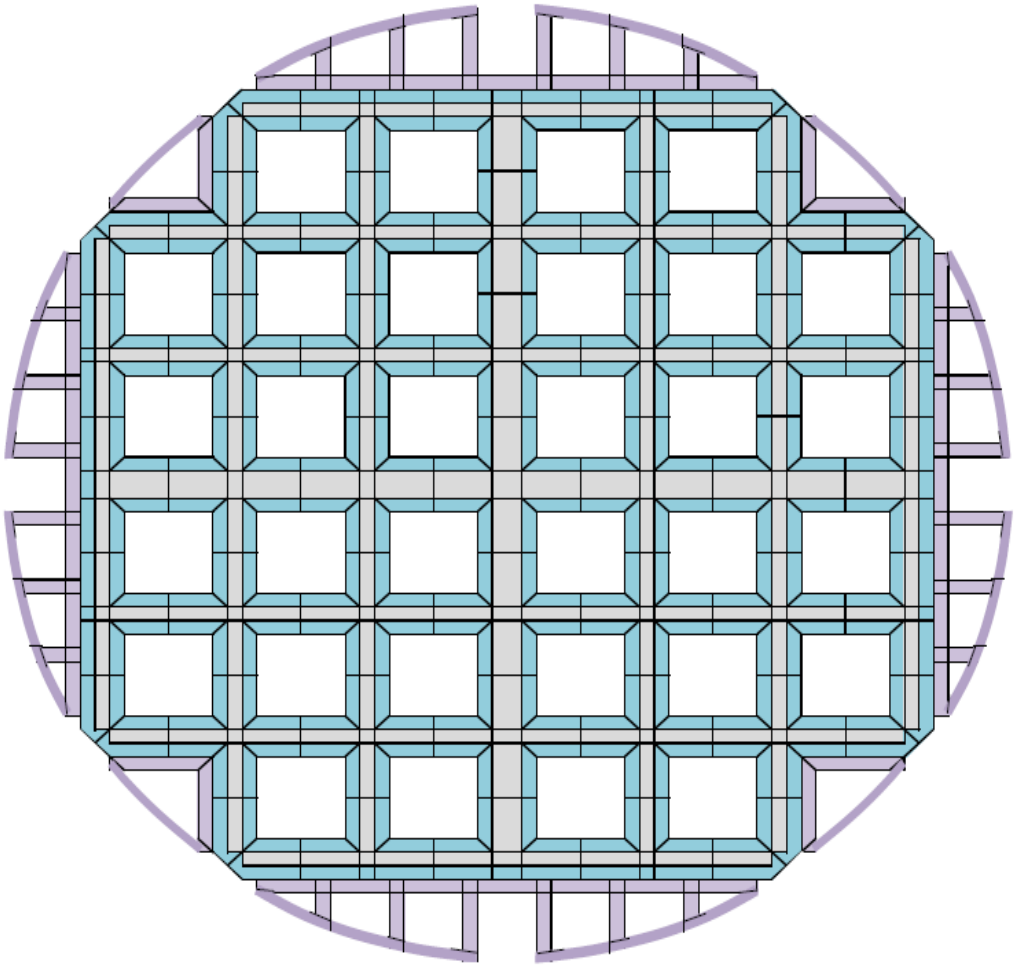


Figure 3.9. Illustration of Noding Diagram for Typical Rail Support Structures for Basket Support, for Vertical or Horizontal System

The area of contact is generally much larger between the rail support structure and the basket or the cavity wall, and generally encompasses a much larger number of slab nodes, compared to the line-contact support structures. However, the considerations of defining appropriate contact areas and contact resistances are exactly the same. Supports of this type are often of aluminum alloy, and may be fastened to the basket plates with bolts or in some cases fastened to the cavity wall, instead of attached to the basket. Appropriate representation of the contact resistance on the rail nodes facing the basket plates, and on the rail nodes facing the cavity wall

is crucial to capturing the thermal behavior of the system. Particularly with high-conductivity aluminum rails, over-estimating the effective contact at these interfaces can result in a substantial (and possibly unrealistic) reduction in the fuel cladding temperatures (including the peak cladding temperature) predicted with the thermal model.

As a bounding case, it is recommended that when modeling systems with this type of support rails, at least one case should be run with the most conservative possible assumptions for gap widths and contact resistances. This defines the highest temperatures that could reasonably be expected for the particular design. Subsequent cases, inserting more optimistic estimates of the thermal connections can then be developed, with the limiting case serving as a reality check on the reasonableness of the results.

The openings in the type of rail support structures illustrated in Figure 3.9 can also serve as flow paths for thermo-siphon recirculation when the package is oriented vertically, if these openings connect to the lower plenum. These openings are often somewhat smaller than those illustrated in Figure 3.7 for the point-contact basket supports, but they can serve the same function, and exactly the same modeling considerations apply. They should be represented as 1-D flow channels, and the modeling should accommodate all modes of heat transfer – conduction, convection, and thermal radiation – within each of the rail openings. If the package is horizontal, the same considerations apply, except that it can be expected that the COBRA-SFS code will predict essentially no heat transfer via convection.

In some systems, the rail support structures consist of solid billets of aluminum alloy, rather than the strut-like cross-section of the structure with open regions illustrated in Figure 3.9. For solid billet supports, there are obviously no flow channels to support thermo-siphon, and no openings where thermal radiation can move heat from the inner nodes of the support structure to the outer rim. Heat transfer in this type of support structure is by conduction only. The slab noding representing this type of support structure should be sufficient to ensure capturing the radial thermal gradient through the cross-section (as shown in the example in Figure 3.10), but exactly the same considerations as discussed above for the open rails and line supports, regarding contact resistance at the interface with the basket and with the cavity wall apply.

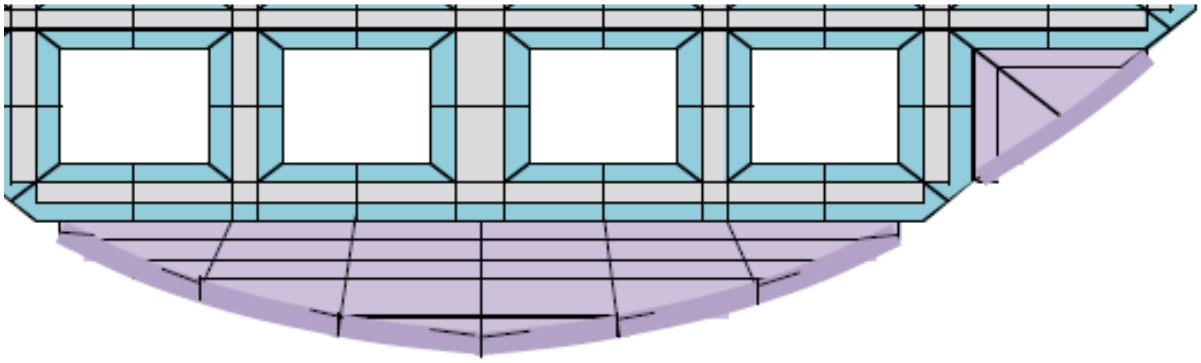


Figure 3.10. Illustration of Noding Diagram for Typical Rail Support Structures Composed of Solid Billets

In general, the thermal resistance at these interfaces will be significantly higher than that of the high-conductivity aluminum billets, and there typically will not be a significant thermal gradient radially through the solid nodes representing this structure. The significant temperature drops will be at the interface with the nodes representing the basket side, and at the interface with the

cavity wall. If a particular model yields results that show a significant temperature gradient in a support rail consisting of aluminum billets, the modeler would be well-advised to take a long, close look at the modeling input defining the thermal connections and nodes in this region.

3.1.5 Beyond the Cavity Inner Wall

Once the model has been defined as far as the cavity wall, the slab node network often suddenly gets much simpler. The structures to be represented from that point outward generally consist of concentric layers of material, usually with nicely cylindrical geometry, as illustrated by the noding of the cask body of the CASTOR-V/21 (see COBRA-SFS – Theory Manual Figure 7.8 (Richmond et. al. 2021)) and the somewhat simpler TN-24P (see COBRA-SFS – Theory Manual Figure 7.40 (Richmond et. al. 2021)). Figure 3.11 shows a slightly more complicated example, illustrating the COBRA-SFS model developed for the MP-187 transportation cask, developed for evaluation of transfer operations (vertical and horizontal) at the Rancho Seco ISFSI. The COBRA-SFS model of this system includes a DSC-24P canister inside the MP-187, but the canister has been omitted from the diagram in Figure 3.11, for clarity.

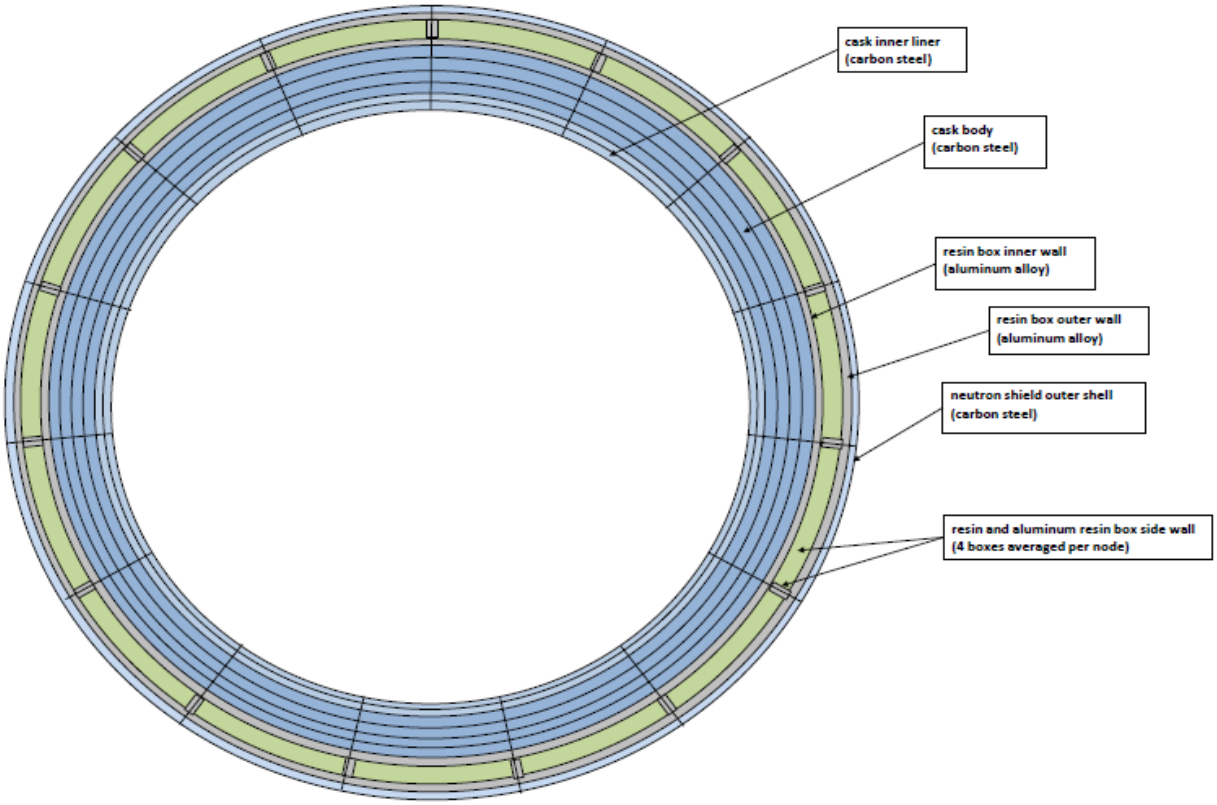


Figure 3.11. Illustration of Noding Diagram for COBRA-SFS Model of MP-187 Transportation Cask, Used as On-site Transfer Cask in Loading Operations at Rancho Seco ISFSI

The only complicated noding in this outer portion of the model consists of capturing the finned internal structure within the resin-filled neutron shield of the MP-187, as illustrated in Figure 3.12. The resin itself has a relatively low thermal conductivity, but the fins are steel overlaid with aluminum; a design feature to enhance thermal performance of the cask, as well as an aid to mechanical strength and ease of fabrication. The thermal network of the solid

conduction nodes in the COBRA-SFS modeling approach makes modeling of this type of structure, consisting of thin plates of dissimilar materials, much simpler and often much more precise than is typically possible in advanced CFD codes.

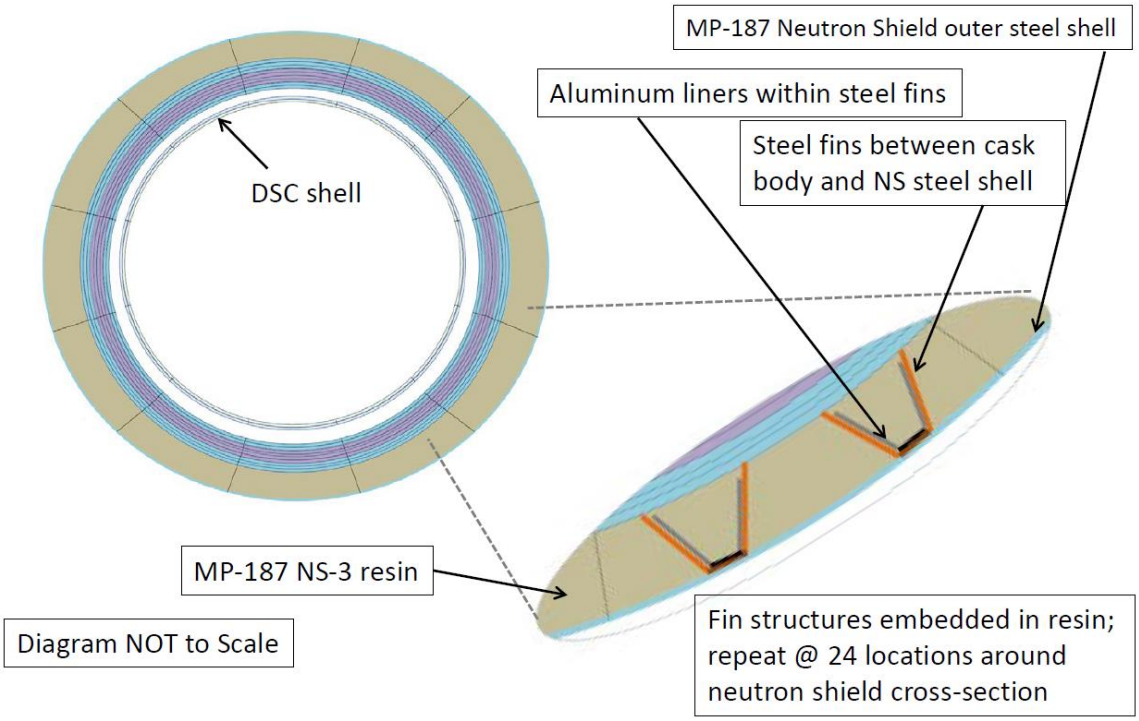


Figure 3.12. Illustration of Detailed Noding of Resin-filled Neutron Shield in COBRA-SFS Model of MP-187 Transportation Cask

A similar simplicity of the overall thermal network noding is possible when modeling vertical ventilated storage modules, such as the Holtec HI-STORM and NAC UMS systems. The “overpack,” as the sturdy concrete structure enclosing the storage canister is often termed, is generally a relatively simple structure consisting of a thin steel inner shell, a thick layer of concrete, and a relatively thin outer steel shell. Figure 3.13 illustrates the slab noding for the concrete overpack in a COBRA-SFS model of a HI-STORM100 system.

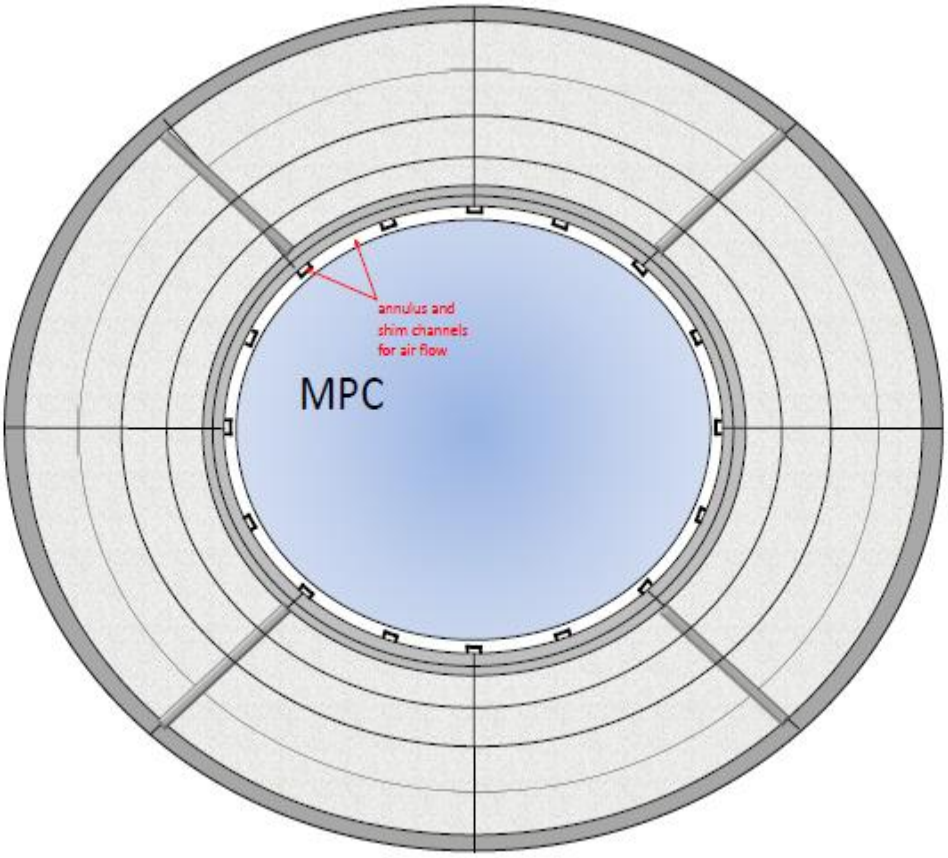


Figure 3.13. Illustration of Noding Diagram for COBRA-SFS Model of Typical Ventilated Concrete Overpack Containing a Multi-assembly Storage Canister

The most important consideration for the COBRA-SFS model in representing the structures beyond the cavity wall is that the geometry of the slab nodes and their interconnections is to preserve the geometric symmetry of the system. Failure to observe this practice (which is inarguably an actual “best practice” by any definition of the term) will introduce unwanted and physically unrealistic asymmetries in the temperature field associated with the slab node network that could in some circumstances destabilize the numerical solution of the slab energy equation. Even if the numerics are unaffected, the resulting temperature field will have unintended biases, and it will be much more difficult to verify the accuracy of the geometry and the correctness of node interconnections defining the system. The symmetry of the temperature field when symmetrical boundary conditions are applied is in fact the single most important test in the process of evaluating and verifying a model before placing any real faith in the predictions derived therefrom. (See discussion in Section 3.2 on model verification.)

3.1.6 Plenum Modeling

The ability to model a gas plenum above and below the fuel assemblies within the basket region is the main feature of COBRA-SFS that allows the code to model spent fuel storage systems efficiently and with appropriate representation of the hydrodynamic behavior of the backfill gas. It is also the simplest modeling feature in the code, consisting of a one-dimensional representation of the region above the basket (the upper plenum) and the region below the

basket (the lower plenum), and should not be overtaxed with complexities it is not designed to handle.

As a first approximation for any model, it is strongly recommended that the plenum regions be represented as simply as possible. Or the plena could simply be omitted from the model, if possible. This is advisable for the model verification steps (see Section 3.2), when the focus is primarily on verifying the accuracy and completeness of the slab node network comprising the basket and walls of the system. But in many cases, this may prove to be sufficient for the purposes of a particular model, and no further refinement of the plena regions may be required to obtain desired results from the modeling effort.

In this minimalist approach, the upper plenum is represented simply as a gas volume for the region above the basket, plus a single layer of material representing the inner layer of the canister or cask lid. This layer is then connected to a boundary region representing an adiabatic boundary (see Section 2.4.11). The lower plenum is similarly represented as a gas volume that is nominally below the basket, plus a layer of material representing the base of the cask or canister. In actual practice, there is unlikely to be an open gas volume below the basket such as is typical above the basket. In vertical systems, the basket structure and the fuel assemblies sit directly on the base of the cavity. The lower plenum, therefore, is usually represented as equivalent to a physical volume of a height defined by the relatively large openings in the basket plates that allow recirculation of backfill gas between and among fuel assemblies and from the downcomer flow channels external to the basket, in designs that allow thermo-siphon recirculation. For systems that are not designed to encourage thermo-siphon hydrodynamics, the ends of the basket plates will usually have openings to assist in draining water from the cask in transfer operations. The height of these openings could be used in a model of such a system to define a reasonable estimate of the gas volume of the lower plenum.

The lower plenum gas region for horizontal systems is generally defined in the same manner as for vertical systems. Although it is sometimes assumed, for modeling purposes, that there is a physical gap between the fuel assemblies (and possibly the basket structures) at the base of the cavity in horizontal systems, this assumption is a modeling conservatism, not generally a physical reality. Casks and canisters to be stored horizontally are loaded vertically, and are very gently handled when down-ending and inserting into the horizontal storage module (or at least as gently as one can handle something that weighs in excess of a hundred tons). There is not likely to be any great movement of the internal components from the initial loading configuration that could form a discernable lower plenum gap between the fuel assemblies and base of the canister.

If this first approximation minimalist approach is not sufficient for a particular application, the plenum model can be expanded to include additional layers representing the material of the lid and base of the system. Figure 3.14 shows an example of such a model, representing a vertical system with thermal connections to ambient air at the top of the system, and to the ground sink temperature at the bottom.

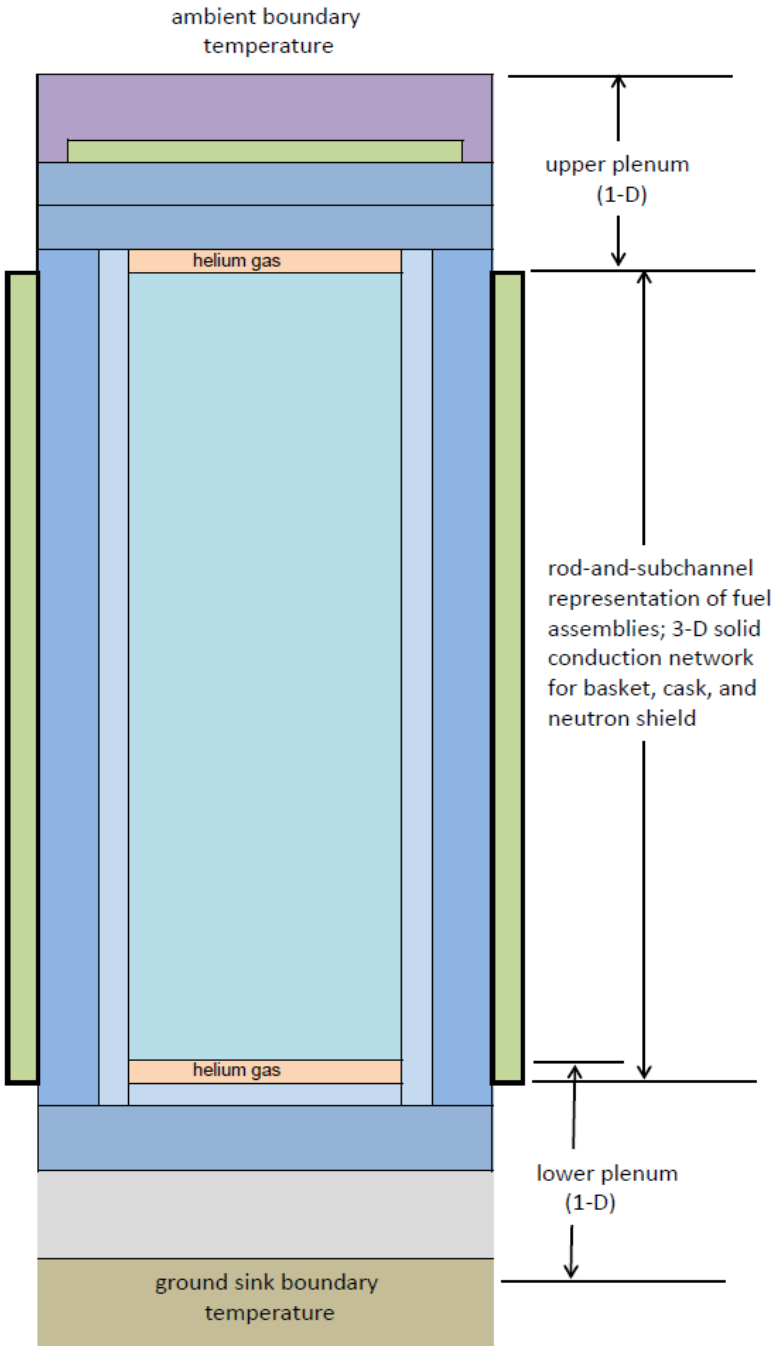


Figure 3.14. Illustration of Extended Plenum Modeling in a COBRA-SFS Model of a Vertical Storage System

It is strongly recommended that plenum modeling be extended only to account for potential one-dimensional heat transfer paths in the axial direction. It is possible to include heat transfer paths in the radial direction, but this is something that should be undertaken only when absolutely necessary for complete representation of a given system, and only by an experienced modeler. If such detailed modeling is needed for a particular application, it can be extremely helpful to have a parallel effort modeling the system with a CFD code (e.g., FLUENT, STAR-CD, STAR-CCM+) that is capable of representing the end regions in full three-dimensional detail. The

results of such modeling can be used to verify the appropriateness and completeness of the more simplified representation of the end structures in the COBRA-SFS model. Given the case-by-case nature of detailed plenum modeling, and because detailed modeling of the end structures is not usually needed for COBRA-SFS modeling of spent fuel storage and transportation system, there are no generic “best practices” to recommend. The best advice available can be summed up in the blanket warning; ***proceed with caution***.

3.2 Model Verification

Having successfully run the gauntlet of the input construction process, and developed an input file that the COBRA-SFS code will actually process and execute, the modeler is faced with the task of verifying that the model correctly represents the system being modeled, within the constraints of the modeling simplifications and assumptions. The input error checking in the code will readily capture many typical actual mistakes in the input stream, but it will not generally capture errors in the consistency of the input, particularly with respect to the specification of the slab noding network connections and connect types. Verifying that the model noding is actually what the modeler intended is an exercise that requires human intervention.

This can be a daunting prospect for a typical large system model, since there are many ways to bury mistakes in the complexity of the overall system geometry and decay heat loading, particularly if the system is loaded non-uniformly. Long experience with this process has shown that the following steps can reliably uncover most, if not all, such errors in consistency of input.

1. Run the model with zero decay heat generation, uniform temperature specified on all boundaries, omitting thermal radiation modeling (i.e., no tape10, no group RADG input)
    a. Favorable Results: solution converges in minimum number of iterations (default is 3), with convergence criteria met within the specific criteria defined by user input; all temperatures within the entire model the same as the specified boundary temperature (within temperature convergence limit).
    b. Unfavorable Results: solution grinds on to maximum number permitted by group CALC input, temperatures are not the same as the specified boundary temperature (may be drifting up, or down, or perhaps both, in different regions of the model).

Results consistent with the description in 1(b) indicate that there are strange and unusual connections in the slab nodes, and energy is being created or destroyed in the course of the solution. The simplest way to track down the problem(s) is to extract the temperatures from slab node block of the output file, insert it into a spreadsheet, and plot the convolutions in the temperature field. It is usually sufficient to capture the peak temperature from each slab node, which is considerably easier than trying to look at the entire temperature field at once. Plots of the peak temperatures over related structures can be useful. Examining temperatures from pairs or sets of nodes that are at geometrically symmetrical positions within the network can uncover errors or omissions in the specified network connections. Note that the output of processed input data, at the front of the output file, provides a detailed table, showing all connections (slab-to-slab and slab-to-fluid) for each slab node in the system. This is an invaluable aid in verifying connections in a model. This table can also be extracted from the output file and inserted into a spreadsheet, for ease of searching and input tracking.

Once the model successfully passes the test of Step #1, the modeler can go on to Step #2.

2. Run the model with zero decay heat generation, uniform temperature specified on all boundaries, add in thermal radiation modeling in the fuel assemblies (include tape10 input, but group RADG input specifies only radiation for assemblies that include fuel rods).

    a. Favorable Results: same as in Step #1 above; solution converges in minimum number of iterations, all temperatures within the entire model the same as the specified boundary temperature, and the modeler has the added satisfaction of satisfying energy conservation with conduction, convection, and thermal radiation included in the fuel assemblies.

    b. Unfavorable Results: same symptoms as in 1(b), but the problem is probably due to inconsistencies in the viewfactor input.

Errors related to tape10 input will generally be severe enough that the code simply will not run, and some appropriate error message will be generated in the input processing. If problems occur in Step #2, after successfully passing through Step #1, this suggests that there is some very subtle inconsistency in the tape10. This may require professional help. In general, if Step #1 has been successfully passed, Step #2 will be just a “feel good” check, almost guaranteed to be successful. Almost.

If the model successfully passes the test of Step #2, the modeler can go on to Step #3.

3. Same conditions as Step #2, but expand group RADG to include user-defined black body viewfactors in open assemblies, in group RADG. (NOTE: If the model has no open regions with user-defined black body viewfactor input, this step is obviously not needed.)

    c. Favorable Results: same as in Step #1 above; solution converges in minimum number of iterations, all temperatures within the entire model the same as the specified boundary temperature, and the modeler has the satisfaction of satisfying energy conservation with conduction, convection, and thermal radiation included in *all* assemblies, not just assemblies with rods, included in the model.

    d. Unfavorable Results: same symptoms as in 1(b), but the problem is probably due to inconsistencies in the viewfactor input specified for assemblies without rods.

Check the input for the user-specified black body view factors. This could be due to round-off error in the limited format field allowed by the fixed formatting in the code input. Or it could be due to incorrect mapping of one or more black body viewfactor type onto the corresponding nodes of the enclosure it is applied to. Check input for RADG.3 and RADG.10.

When the model successfully passes the test of Step #3, the modeler can go on to Step #4.

4. Run the model with all thermal radiation modeling included (i.e., tape10 and complete input for group RADG), and uniform decay heat generation rate in all assemblies with fuel rods (note that the same axial heat generation profile should be used in all assemblies for this step of the model testing, but the axial profile need not be uniform).

For this step, the same symmetry checks of temperatures and temperature profiles recommended for checking for energy sources/sinks in Step #1 are advisable. Illustrations of this type of checking using a spreadsheet are shown in Figure 3.15 through Figure 3.17. (Note that the plots in these figures are “working notes” from actual models built with COBRA-SFS and are not intended to be “report quality” graphics. The plots are in °F, because that is the default temperature unit in the COBRA-SFS code. The labels identifying the temperature regions plotted are idiosyncratic to the particular problem, and are meaningful only to the

modeler. The purpose of these plots is to illustrate the process, not present results for specific cases.) Figure 3.15 shows an illustration of a symmetry check that indicates a possible node connection error manifested in the temperature profiles in circumferential rings of nodes representing the cask body in that particular COBRA-SFS model. Figure 3.16 shows a similar illustration of temperatures from another model, in a region of concentric rings of nodes where thermal connections are appropriately symmetrical and correct, consistent with the geometry of the model.

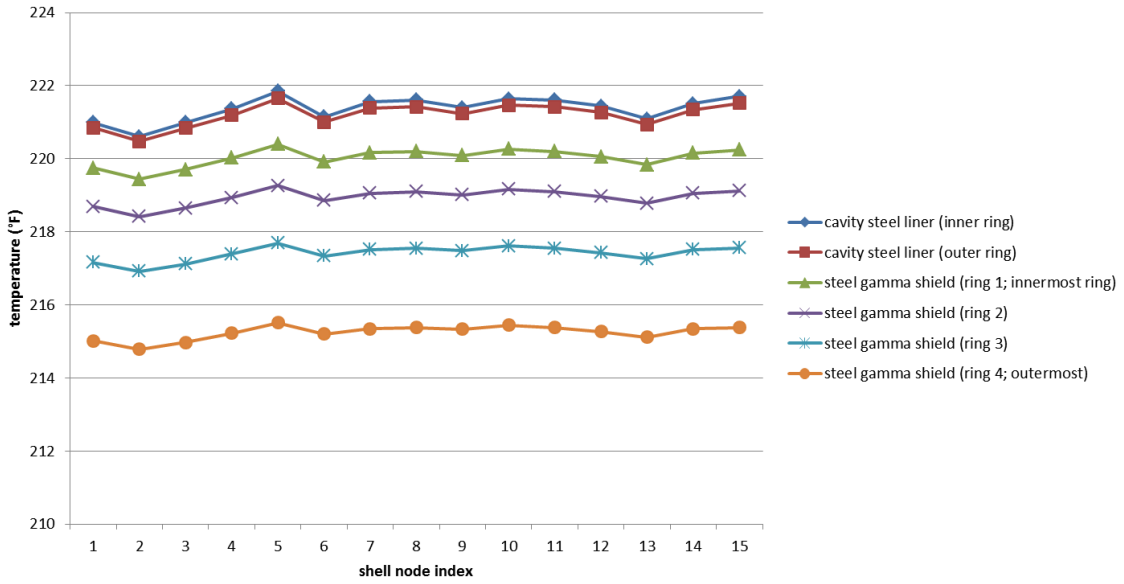


Figure 3.15. Illustration of Slab Peak Temperature Profiles for Connected Nodes that are Expected to be Symmetrically Connected (but obviously are not...)

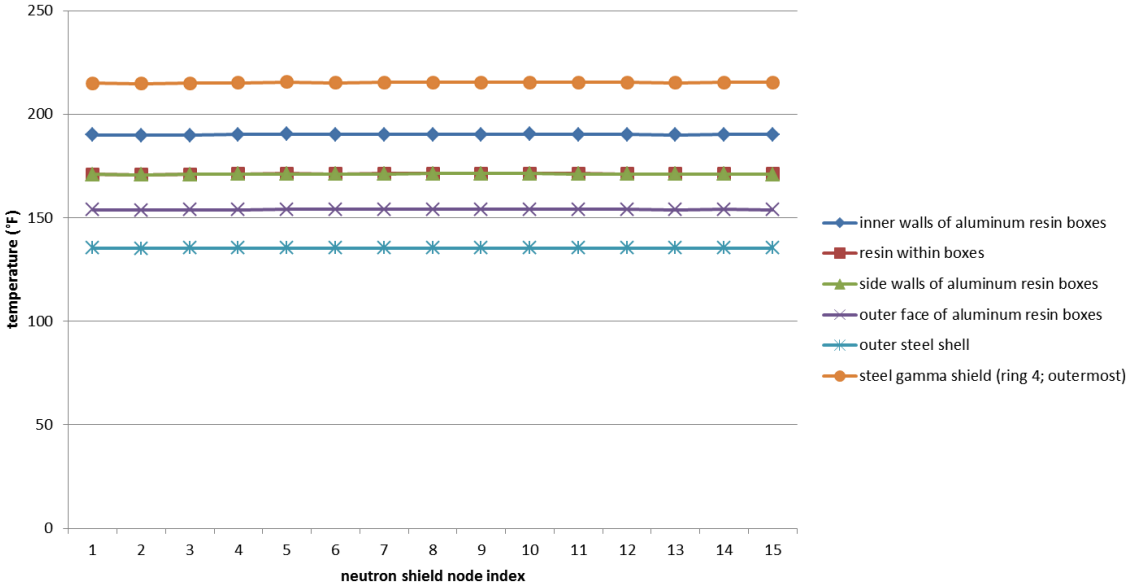


Figure 3.16. Illustration of Slab Peak Temperature Profiles for Connected Nodes that are Connected Symmetrically, as Intended in the Model

A simple bar chart matching symmetric locations in a model of the basket structure, with uniform heat generation imposed on all assemblies, is shown in Figure 3.17. The paired “left side” and “right side” temperatures should be identical or very nearly so (as they appear to be for pairs 3 and 6, and are almost identical for pairs 2, 4, and 5, but are clearly not for pairs 1 and 7). Plots of this type are almost as good as direct text messages from the code telling the modeler where to look for inconsistent input in the thermal connections between slab nodes representing the basket structure. Often, errors of this type tend to run in packs; locating and fixing one can often lead the modeler to a whole series of them, and they can all be fixed at the same time, before running the case again, and replotting the results, to see if the asymmetries have disappeared.

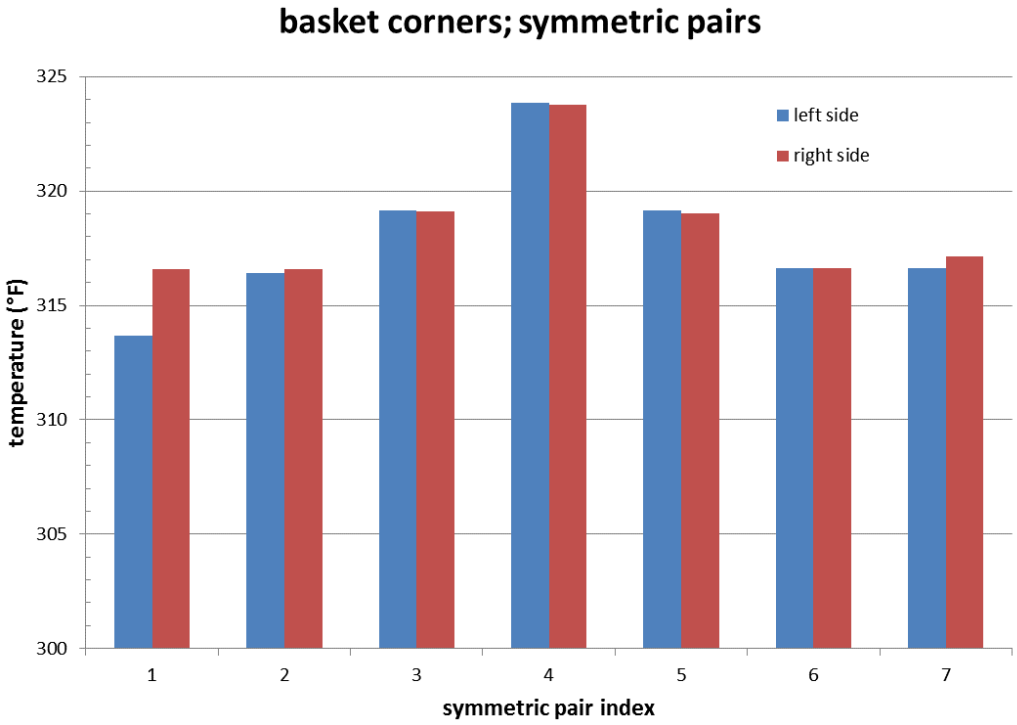


Figure 3.17. Illustration of Plot Revealing Inconsistent Slab Connection Input by Comparing Peak Temperatures in Nodes at Geometrically Symmetrical Locations within the Model

When evaluation of the code results from Step #4 shows that symmetries within the temperature field match the geometric symmetries of the model, within nominal convergence criteria (typically 0.1 deg.F or less), and the overall thermal behavior of the system appears consistent with physical reality, the model can be considered “verified by inspection.” The modeler can have reasonable confidence that the thermal network is correctly representing the geometry of the system, within the constraints of the modeling assumptions. At this point, the heat generation input can be revised to reflect the actual (usually non-uniform) distribution of heat generation in the system, and the work of generating reportable results can begin.

Building detailed thermal models with COBRA-SFS is a process that requires patience and perseverance, but can be extremely rewarding, in terms of obtaining useful understanding of thermal behavior of spent fuel storage and transportation systems. The validation and verification history of the code gives confidence that an appropriately developed and verified model is predicting accurate and meaningful results, within the constraints of the assumptions underlying the (generally incompletely known) boundary conditions. This section provides a

summary of what can be termed “best practices” for the process of model development and verification. But it should be borne in mind that what constitutes best practices can (and probably should) evolve over time, to be congruent with modeling needs and changes in computational capabilities. The recommendations given here are based on experience, and are suitable for current applications of the COBRA-SFS code and modern (i.e., circa 2015) computer capabilities. But the single most important recommendation offered is that the user should develop a good engineering appreciation of the thermal behavior in such systems, and let that be the primary guiding principle in determining how to approach modeling any particular system.

4.0 Materials Property Library

This section contains information on the fluid and solid material properties included in the COBRA-SFS properties library. These material properties can be used in the PROP.2 and the PROP.3 sections of an input file for fluids and solids, respectively. These are also being implemented into the simplified input file format. This section presents selected properties. For fluids, not all backfill pressures are listed. The variables used in this section are defined as follows:

- T represents temperature in degrees Fahrenheit (°F),
- h represents enthalpy in units Btu/lbm,
- k represents thermal conductivity in units Btu/hr-ft-°F,
- C_p represents specific heat in units Btu/lb-°F,
- v represents specific volume in units ft^3/lbm,
- μ represents viscosity in units lbm/ft-hr, and
- ρ represents density in units lbm/ft^3.

4.1 Properties of Commonly Used Fluids

NOTE: All Fluid Properties are derived from the NIST Chemistry WebBook, SRD 69 (Lemmon et. al 2020)

4.1.1 Helium, 1 atm

Helium is an inert gas often used for backfilling canisters after vacuum drying.

Table 4-1 Tabulated material properties of Helium at 1 atm.

T [°F]	h [Btu/lbm]	k [Btu/hr-ft-°F]	C_p [Btu/lbm-°F]	v [ft^3/lbm]	μ [lbm/ft-hr]
0	572.9	0.0807	1.2412	83.91	0.0432
200	821.2	0.1036	1.2412	120.39	0.0553
400	1069.4	0.1245	1.2412	156.88	0.0665
600	1317.6	0.1440	1.2412	193.37	0.0769
800	1565.8	0.1625	1.2412	229.85	0.0868
1000	1814.1	0.1801	1.2412	266.34	0.0963

1200	2062.3	0.1970	1.2412	302.83	0.1054
1400	2310.5	0.2133	1.2412	339.31	0.1143
1600	2558.8	0.2291	1.2412	375.80	0.1228
1800	2807.0	0.2445	1.2412	412.29	0.1312
2000	3055.2	0.2594	1.2412	448.77	0.1393
2200	3303.5	0.2740	1.2412	485.26	0.1473

4.1.2 Nitrogen, 1 atm

Nitrogen is a common diatomic gas and a major component of air.

Table 4-2 Tabulated material properties for Nitrogen at 1 atm.

T [°F]	h [Btu/lbm]	k [Btu/hr-ft-°F]	C_p [Btu/lbm-°F]	v [ft^3/lbm]	μ [lbm/ft-hr]
0	113.9	0.0132	0.24896	11.97	0.0382
100	138.8	0.0154	0.24891	14.59	0.0445
200	163.7	0.0174	0.24927	17.20	0.0504
300	188.7	0.0194	0.25018	19.81	0.0559
400	213.7	0.0213	0.25171	22.42	0.0611
500	239.0	0.0232	0.25381	25.03	0.0660
600	264.5	0.0250	0.25639	27.64	0.0707
700	290.3	0.0269	0.25931	30.24	0.0752
800	316.4	0.0287	0.26244	32.85	0.0795

4.1.3 Oxygen, 1 atm

Diatomic oxygen is a major component of air. It is never used in its pure form for any SNF applications, but it is included in this library to allow the user to easily incorporate it into non-standard gas mixtures.

Table 4-3 Tabulated material properties for Oxygen at 1 atm.

T [°F]	h [Btu/lbm]	k [Btu/hr-ft-°F]	C_p [Btu/lbm-°F]	v [ft^3/lbm]	μ [lbm/ft-hr]
0	99.7	0.0133	0.21877	10.48	0.0436
100	121.7	0.0159	0.22025	12.77	0.0512
200	143.8	0.0185	0.22297	15.05	0.0582
300	166.3	0.0211	0.22657	17.34	0.0649
400	189.1	0.0237	0.23065	19.62	0.0711
500	212.4	0.0263	0.23486	21.91	0.0771
600	236.1	0.0288	0.23897	24.19	0.0827
700	260.2	0.0313	0.24285	26.47	0.0882
800	284.7	0.0338	0.24641	28.75	0.0934

4.1.4 Argon, 1 atm

Argon is an inert gas and is included as a minor component of air.

Table 4-4 Tabulated material properties for Argon at 1 atm.

T [°F]	h [Btu/lbm]	k [Btu/hr-ft-°F]	C_p [Btu/lbm-°F]	v [ft^3/lbm]	μ [lbm/ft-hr]
0	57.1	0.0089	0.12479	8.39	0.0479
100	69.5	0.0106	0.12463	10.23	0.0565
200	82.0	0.0121	0.12454	12.06	0.0646
300	94.4	0.0135	0.12449	13.89	0.0722
400	106.9	0.0148	0.12446	15.72	0.0794
500	119.3	0.0161	0.12444	17.55	0.0863
600	131.8	0.0173	0.12442	19.38	0.0929
700	144.2	0.0185	0.12441	21.20	0.0991
800	156.7	0.0196	0.1244	23.03	0.1052

4.1.5 Air, 1 atm

Air is a mixture of nitrogen, oxygen, argon, and trace gases. The properties listed are for a standard mixture of air consisting of 78% nitrogen, 21% oxygen, and 1% argon. Trace gases are ignored in this mixture. Each of the component gas properties is sourced from NIST and listed above.

Table 4-5 Tabulated material properties for Air at 1 atm.

T [°F]	h [Btu/lbm]	k [Btu/hr-ft-°F]	C_p [Btu/lbm-°F]	v [ft^3/lbm]	μ [lbm/ft-hr]
0	110.3364	0.013161	0.241378	11.62381	0.039402
100	134.4823	0.015438	0.241649	14.16155	0.046026
200	158.6882	0.017599	0.2425	16.69671	0.052178
300	183.0101	0.019687	0.243965	19.23066	0.057937
400	207.5033	0.02173	0.246015	21.76362	0.063367
500	232.2271	0.023745	0.248537	24.29657	0.068522
600	257.2243	0.025739	0.251412	26.82853	0.073438
700	282.5181	0.027713	0.254504	29.36049	0.078147
800	308.1293	0.029663	0.257693	31.89167	0.082678

4.2 Properties of Commonly Used Solids

4.2.1 Aluminum Alloy 6061

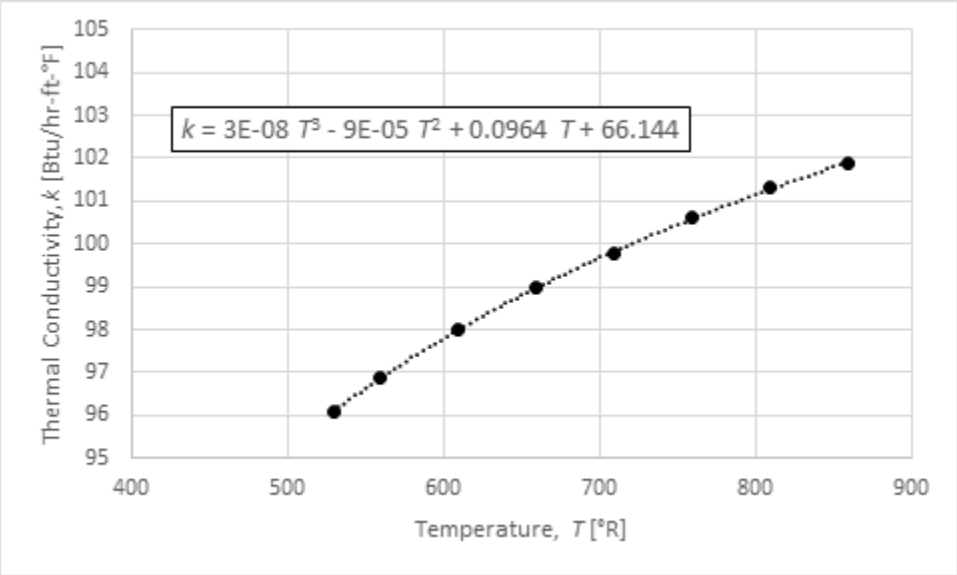
This alloy is also called any of the following: al6061; UNS A96061; ISO AlMg1SiCu; Aluminium 6061-T6, AD-33 (Russia); AA6061-T6; 6061T6, UNS A96061; ISO AlMg1SiCu; Aluminium 6061-T651, AD-33 (Russia); AA6061-T651. The tabulated material properties are shown below.

Table 4-6 Tabulated material properties for 6061 Aluminum

T [°F]	*k* [Btu/hr-ft-°F]	*ρ* [lbm/ft^3]	C_p [Btu/lbm-°F]
70	96.1	169.34	0.213

100	96.9	169.34	0.215
150	98.0	169.34	0.218
200	99.0	169.34	0.221
250	99.8	169.34	0.223
300	100.6	169.34	0.226
350	101.3	169.34	0.228
400	101.9	169.34	0.230

A third-order polynomial fit of thermal conductivity as a function of temperature (in °R) has been plotted below.



4.2.2 Aluminum Alloy 2024

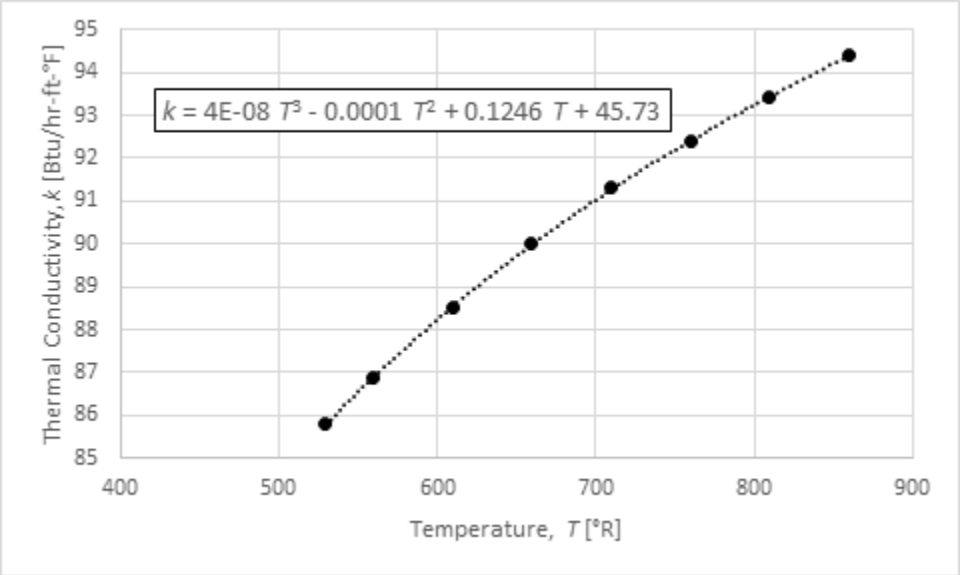
This alloy is also called any of the following: Aluminium 2024-T351; AA2024-T351, Aluminium 2024-T4; UNS A92024; ISO AlCu4Mg1; NF A-U4G1 (France); DIN AlCuMg2; AA2024-T4, ASME SB211; CSA CG42 (Canada).

Table 4-7 Tabulated material properties for 2024 Aluminum

T [°F]	k [Btu/hr-ft-°F]	ρ [lbm/ft^3]	C_p [Btu/lbm-°F]

70	85.8	173.55	0.209
100	86.9	173.55	0.211
150	88.5	173.55	0.214
200	90.0	173.55	0.217
250	91.3	173.55	0.220
300	92.4	173.55	0.222
350	93.4	173.55	0.224
400	94.4	173.55	0.228

A third-order polynomial fit of thermal conductivity as a function of temperature (in °R) has been plotted below.



4.2.3 Carbon Steel (Material Group A)

The carbon steels in Material Group A all have similar thermal properties. Material Group A includes plain carbon steel, SA-516, and ASTM A36 carbon steel. Other appropriate metals in this group can be identified using ASME BVPC Section 2 Part D.

Table 4-8 Tabulated material properties for Carbon Steel (Material Group A)

T [°F]	k [Btu/hr-ft-°F]	ρ [lbm/ft^3]	C_p [Btu/lbm-°F]
70	34.9	490.75	0.102
100	34.7	490.75	0.105
150	34.2	490.75	0.109
200	33.7	490.75	0.112
250	33.0	490.75	0.115
300	32.3	490.75	0.118
350	31.6	490.75	0.120
400	30.9	490.75	0.122
500	29.4	490.75	0.126
600	28.0	490.75	0.132
800	25.3	490.75	0.145
900	23.8	490.75	0.153
1000	22.4	490.75	0.161
1250	18.6	490.75	0.190
1500	15.5	490.75	0.190

4.2.4 Carbon Steel (Material Group B)

The carbon steels in Material Group B all have similar thermal properties. Material Group B includes ASTM A203 carbon steel and other carbon steels with significant nickel content. Other appropriate metals in this group can be identified using ASME BVPC Section 2 Part D.

Table 4-9 Tabulated material properties for Carbon Steel (Material Group B)

T [°F]	k [Btu/hr-ft-°F]	ρ [lbm/ft^3]	C_p [Btu/lbm-°F]
70	27.3	490.75	0.105
100	27.6	490.75	0.108

150	27.8	490.75	0.112
200	27.8	490.75	0.116
250	27.6	490.75	0.119
300	27.3	490.75	0.122
350	26.9	490.75	0.125
400	26.5	490.75	0.127
500	25.7	490.75	0.131
600	24.9	490.75	0.136
800	23.2	490.75	0.148
900	22.3	490.75	0.156
1000	21.1	490.75	0.163
1250	17.6	490.75	0.194
1500	14.9	490.75	0.177

4.2.5 High Chrome Steel (Material Group J, Stainless Steel)

The stainless steels in Material Group J all have similar thermal properties. Material Group J includes SS304 stainless steel and other stainless steels with significant chromium content. Other appropriate metals in this group can be identified using ASME BVPC Section 2 Part D.

Table 4-10 Tabulated material properties for High Chrome Steel (Material Group J and Stainless)

T [°F]	k [Btu/hr-ft-°F]	ρ [lbm/ft^3]	C_p [Btu/lbm-°F]
70	8.6	499.39	0.114
100	8.7	499.39	0.115
150	9.0	499.39	0.117
200	9.3	499.39	0.119
250	9.6	499.39	0.122

300	9.8	499.39	0.123
350	10.1	499.39	0.125
400	10.4	499.39	0.126
500	10.9	499.39	0.129
600	11.3	499.39	0.130
800	12.3	499.39	0.134
900	12.7	499.39	0.135
1000	13.1	499.39	0.135
1250	14.3	499.39	0.140
1500	15.3	499.39	0.142

4.2.6 Incoloy 800

Incoloy 800 is an alloy of iron, chromium, and nickel that is often used in high-temperature applications due to its resistance to oxidation and carburization. Incoloy 800 typically has a maximum working temperature of 1100 °F, but properties up to 1500 °F are included for modeling purposes.

Table 4-11 Tabulated material properties for Incoloy 800

T [°F]	k [Btu/hr-ft-°F]	ρ [lbm/ft^3]	C_p [Btu/lbm-°F]
70	6.7	495.94	0.111
100	6.8	495.94	0.110
150	7.1	495.94	0.111
200	7.4	495.94	0.112
250	7.7	495.94	0.113
300	8.0	495.94	0.115
350	8.3	495.94	0.116
400	8.5	495.94	0.117

500	9.1	495.94	0.120
600	9.6	495.94	0.122
800	10.6	495.94	0.126
900	11.1	495.94	0.129
1000	11.6	495.94	0.131
1250	13.0	495.94	0.139
1500	14.5	495.94	0.148

4.2.7 Kaowool® Thermal Ceramic

Kaowool® is a ceramic produced by Morgan Advanced Materials and is primarily comprised of alumina and silica. It can be used as a thermal insulator. The properties included in this document are for the insulating boards made of Kaowool® 80 which have a continuous use temperature limit of 2000 °F. Kaowool® can also be obtained in blanket form or in bulk.

Table 4-12 Tabulated material properties for Kaowool® Thermal Ceramic

T [°F]	k [Btu/hr-ft-°F]	ρ [lbm/ft^3] (average)
500	0.042	19.0
1000	0.058	19.0
1500	0.083	19.0

4.2.8 Lead (ASTM B29)

Lead is a ductile, dense metal used in nuclear shielding applications. It has a notably low melting point (622 °F) so caution should be used when using lead in high-temperature modeling.

Table 4-13 Tabulated material properties for Lead (ASTM B29)

T [°F]	k [Btu/hr-ft-°F]	ρ [lbm/ft^3]	C_p [Btu/lbm-°F]
68 - 212	19.1	707.96	0.031

5.0 Fuel Library

These assemblies have been selected for the COBRA-SFS fuel library because they are relatively common and have information that is publicly available. The information in the following sections is used in the COBRA-SFS simplified input. This geometry and fuel information is used to determine flow, radiation, and fuel conduction parameters. An example of the parameters referred to in COBRA-SFS is shown in Figure 5.1.

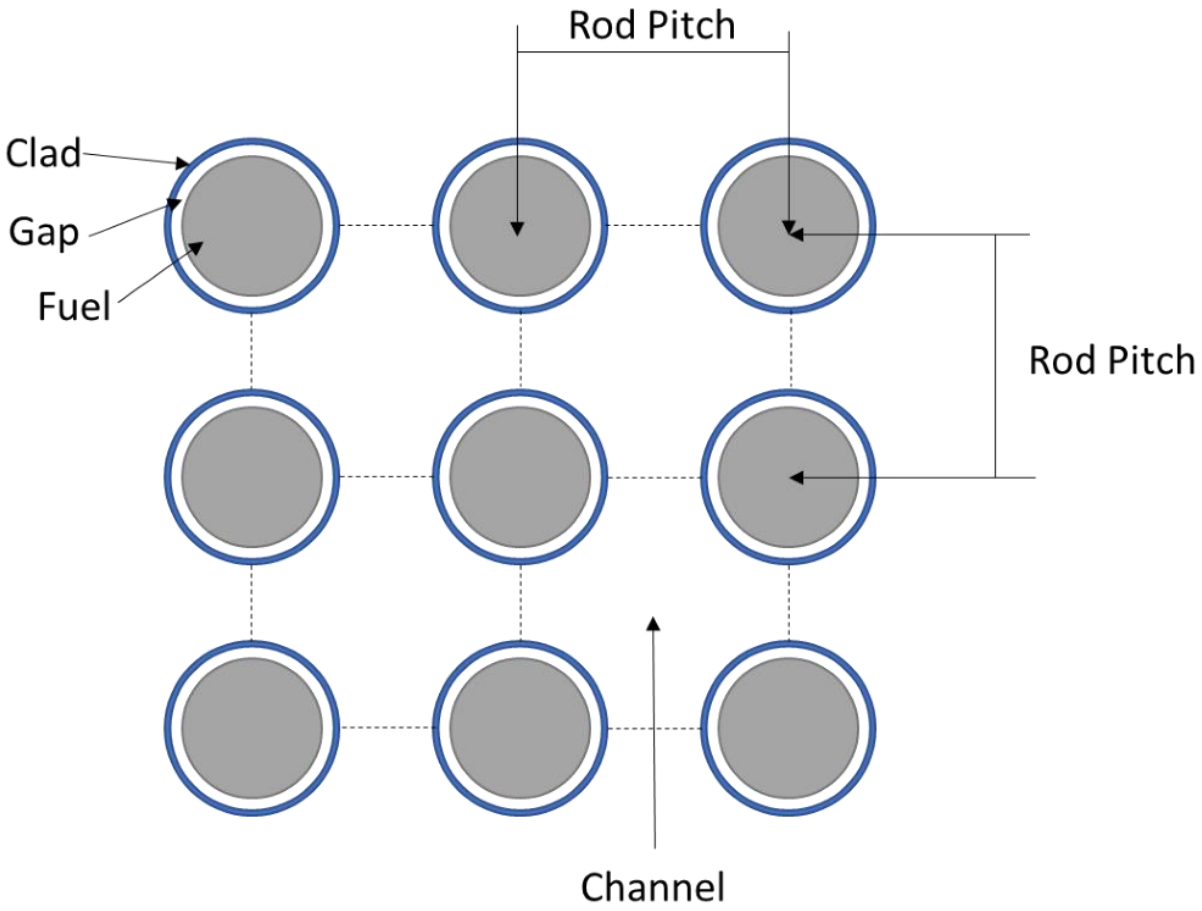


Figure 5.1 Example Assembly Diagram

5.1 Boiling Water Reactor Fuel Assemblies

This section details available BWR fuel assembly data that is planned to be integrated into COBRA-SFS.

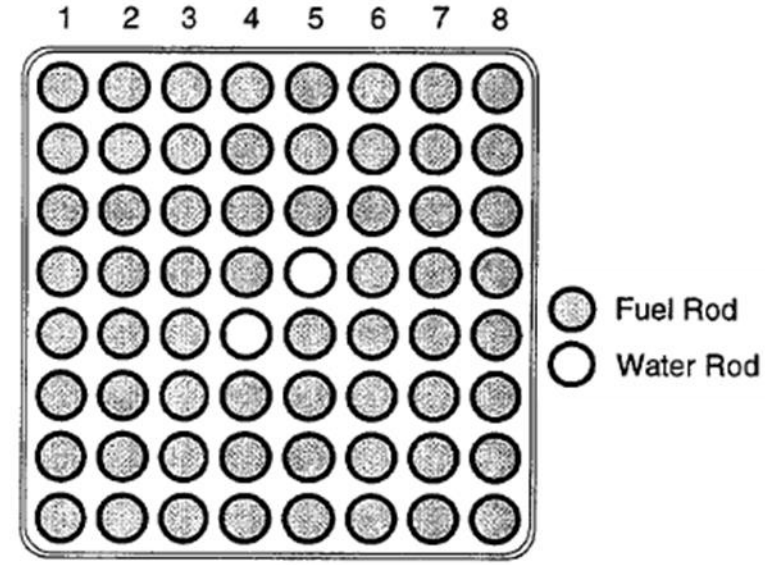
5.1.1 GE 8x8R

The GE 8x8 Retrofit (8x8R) BWR fuel assembly is a further revision to succeed the original GE 8x8 assembly. This 8x8R design has two water rods versus one water rod in the previous GE 8x8 standard design. The standard 8x8 and 8x8R assemblies are prevalent in BWR reactors.

This GE 8x8R is representative of an 8x8 BWR assembly for many spent fuel thermal modeling applications utilizing COBRA-SFS.

Table 5-1 GE 8x8 critical dimensions

Fuel Type	GE 8x8R
Cladding Diameter (in.)	0.483
Rod Pitch (in.)	0.64
Pellet Diameter (in)	0.487
Cladding Thickness (in)	0.032
Gap Thickness (in)	0.009
Assembly Width (in)	5.258
Channel Box (in)	0.0120 thick
Channel Box Material	Zircaloy-4
Number of Rods per Assembly	62
Number of Water Rods	2
Water Rod Guide Tube ID (in.)	Slightly Larger than Fuel
Pellet Enrichment (%)	
Fuel Rod Length (in)	160.782
Active Length (in)	144
Cladding Material	Zircaloy-2



8x8 with 2 small water rods. Other 8x8s have 1 large water rod (4 array locations)

Figure 5.2 GE 8x8 array map

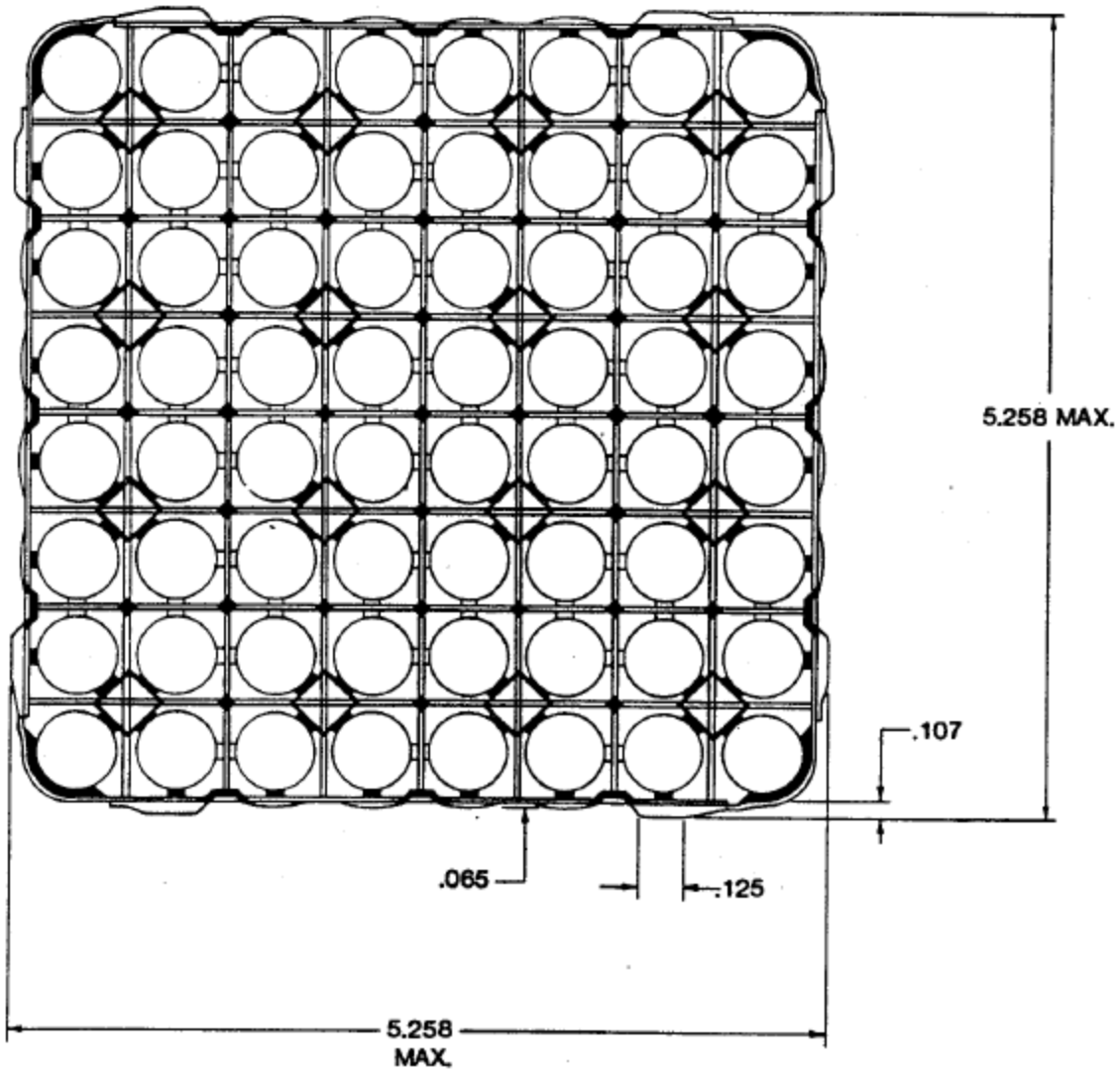


Figure 5.3 GE 8x8R grid diagram

5.1.2 GE 9x9 Assembly

Table 5-2 GE 9x9 critical dimensions

Fuel Type	9x9 BWR Simulator
Cladding Diameter (in.)	0.44
Rod Pitch (in.)	0.567
Pellet Diameter (in)	
Cladding Thickness (in)	
Gap Thickness (in)	
Assembly Width (in)	
Channel Box (in)	5.276 ID, 5.472 OD
Number of Rods per Assembly	74 Full, 66 Partial
Number of Water Rods	2
Water Rod Guide Tube ID (in.)	0.98 OD, 0.921 ID
Pellet Enrichment (%)	
Fuel Rod Length (in)	155.90 Full, 102.75 Partial
Active Length (in)	
Cladding Material	Zircaloy-2 Channel and water rod

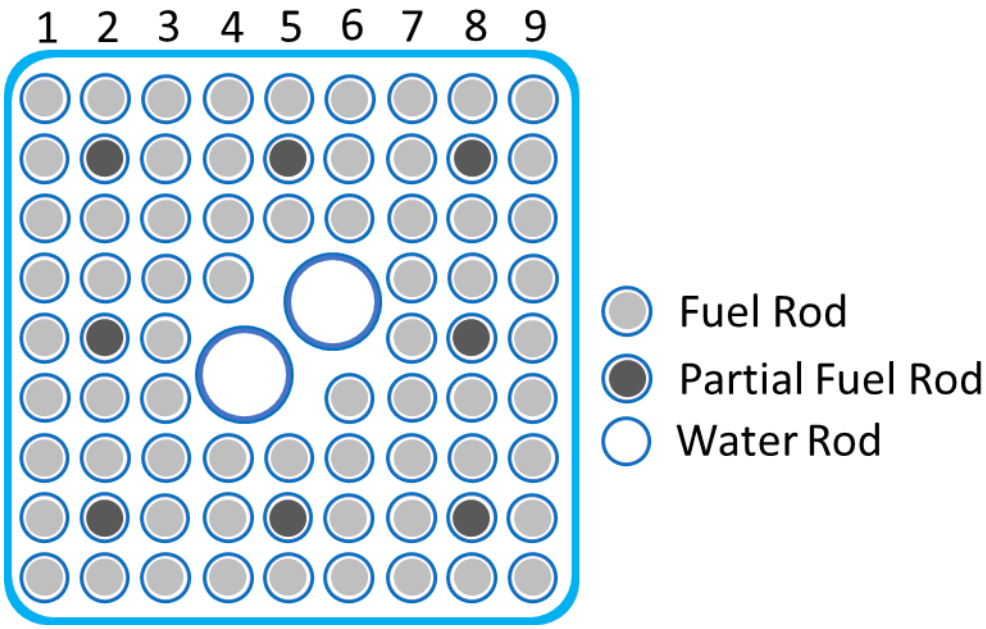


Figure 5.4 GE 9x9 diagram

5.1.3 GE 10x10 ANF

Table 5-3 GE 10x10 critical dimensions

Fuel Type	GE 10x10 4,5,6 ANF
Cladding Diameter (in.)	0.3787
Rod Pitch (in.)	0.51
Pellet Diameter (in)	0.3224
Cladding Thickness (in)	0.0248
Gap Thickness (in)	0.00335
Assembly Width (in)	5.25
Channel Box (in)	
Number of Rods per Assembly	92
Number of Water Rods	2
Water Rod Guide Tube ID (in.)	0.92
Pellet Enrichment (%)	
Fuel Rod Length (in)	163.84
Active Length (in)	150
Cladding Material	Zircaloy-2

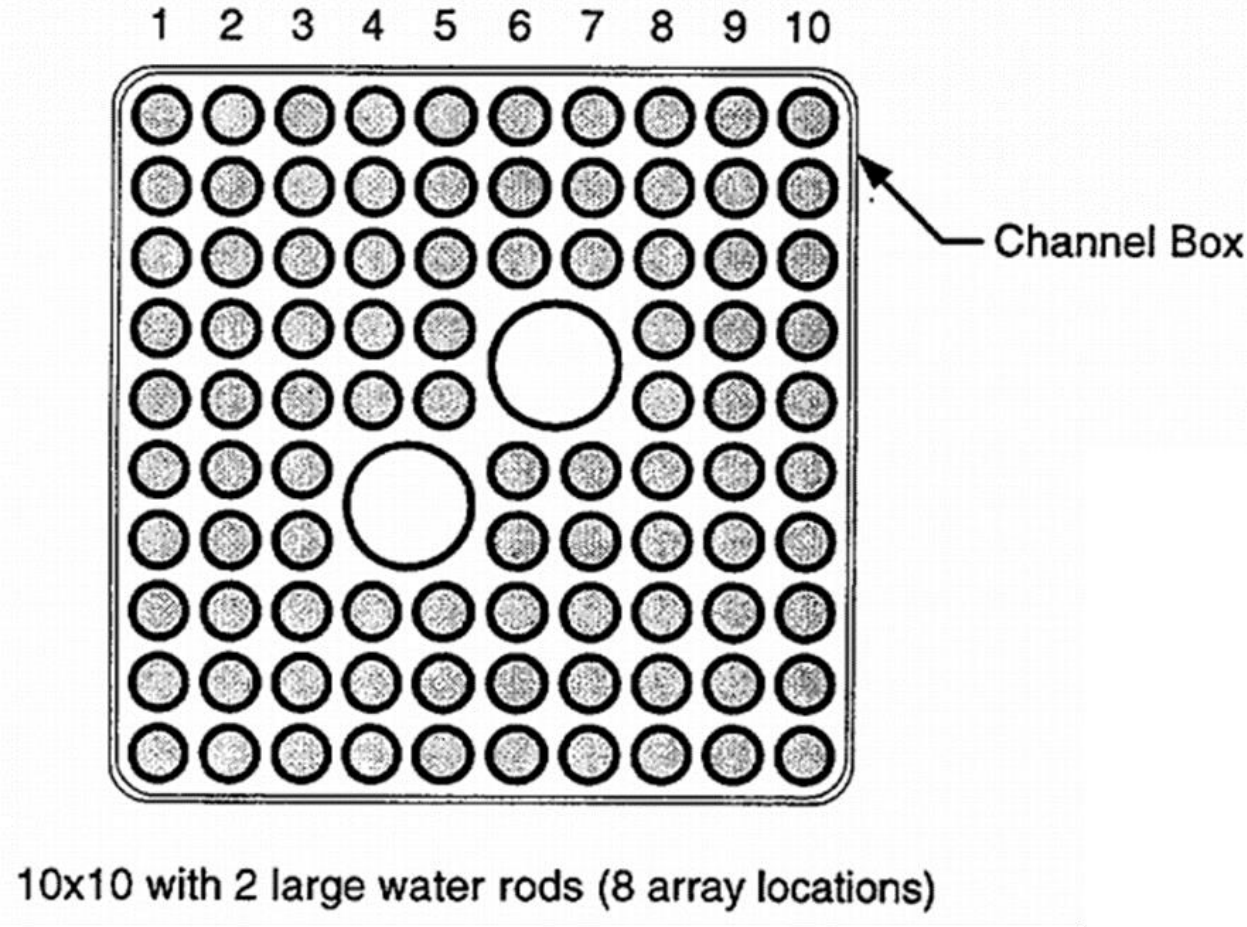


Figure 5.5 GE 10x10 diagram

5.2 Pressurized Water Reactor Fuel Assemblies

This Section details available information on PWR fuel assemblies that are planned to be integrated into COBRA-SFS.

5.2.1 Westinghouse 14x14

The Westinghouse 14x14 PWR fuel assemblies are used in first-generation Westinghouse PWR plants. This 14x14 PWR fuel design is representative of 14x14 fuel assemblies used in Westinghouse plants for most modeling applications with COBRA-SFS.

Table 5-4 Westinghouse 14x14 Critical Dimensions

Fuel Type (PWR)	WE 14x14
Cladding Diameter (in.)	0.422
Rod Pitch (in.)	0.556
Pellet Diameter (in)	0.3659
Pellet Composition (% U-235)	
Cladding Thickness (in)	0.0243
Gap Thickness (in)	0.0075
Assembly Width (in)	7.763
Number of Fuel Rods per Assembly	179
Number of Control Rods	16
Control Rod Diameter (in.)	0.543
Fuel Rod Length (in)	161.1
Active Length (in)	144
Clad Material	Zircaloy-4

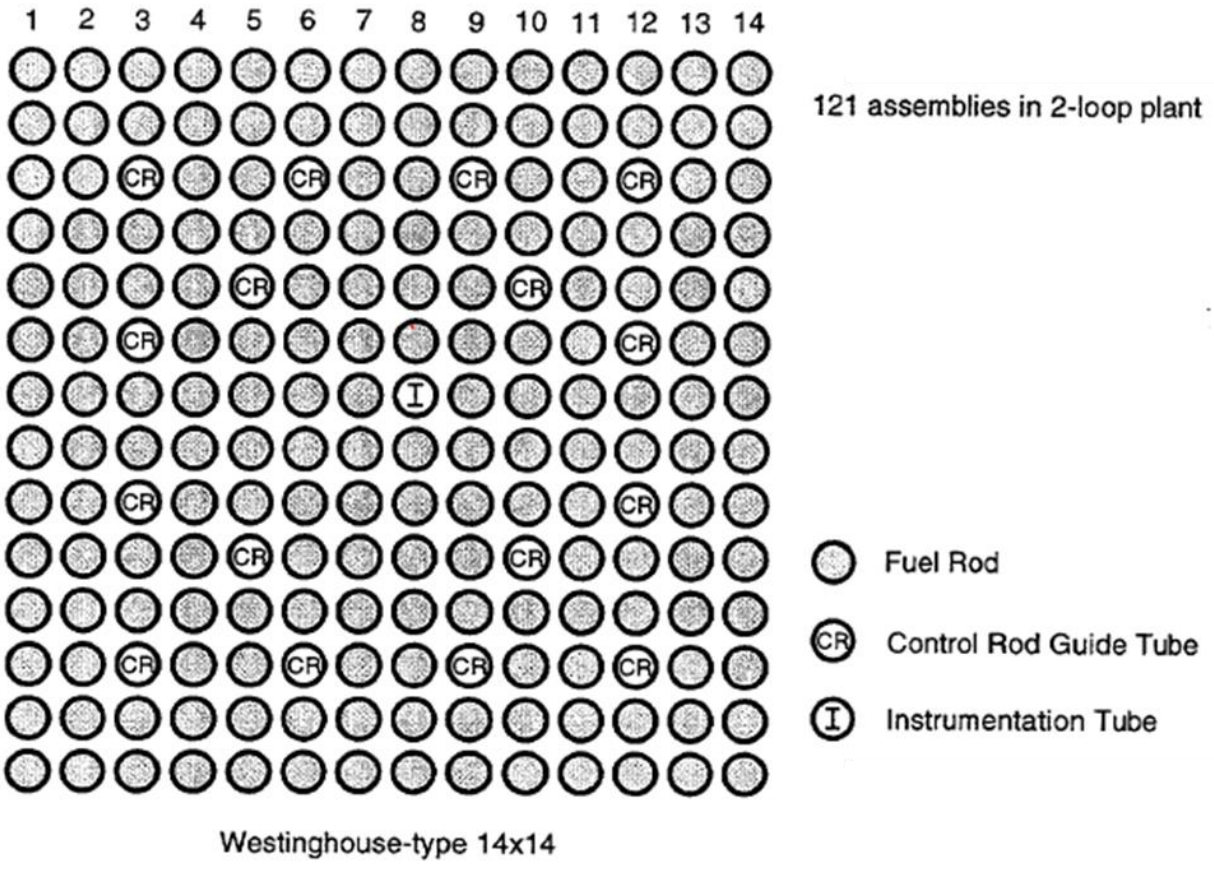


Figure 5.6 Westinghouse 14x14 assembly diagram

5.2.2 Combustion Engineering 14x14 Standard

The Combustion Engineering (CE) 14x14 Standard PWR fuel assembly is the generic 14x14 fuel used in Combustion Engineering plants.

Table 5-5 CE 14x14 Critical Dimensions

Fuel Type (PWR)	CE Std.
Cladding Diameter (in.)	0.44
Rod Pitch (in.)	0.58
Pellet Diameter (in)	0.3765
Pellet Composition (% U-235)	
Cladding Thickness (in)	0.028
Gap Thickness (in)	0.00375
Assembly Width (in)	8.1
Number of Fuel Rods per Assembly	164
Number of Control Rods	5
Control Rod Diameter (in.)	1.035
Fuel Rod Length (in)	147
Active Length (in)	137
Clad Material	Zircaloy-4

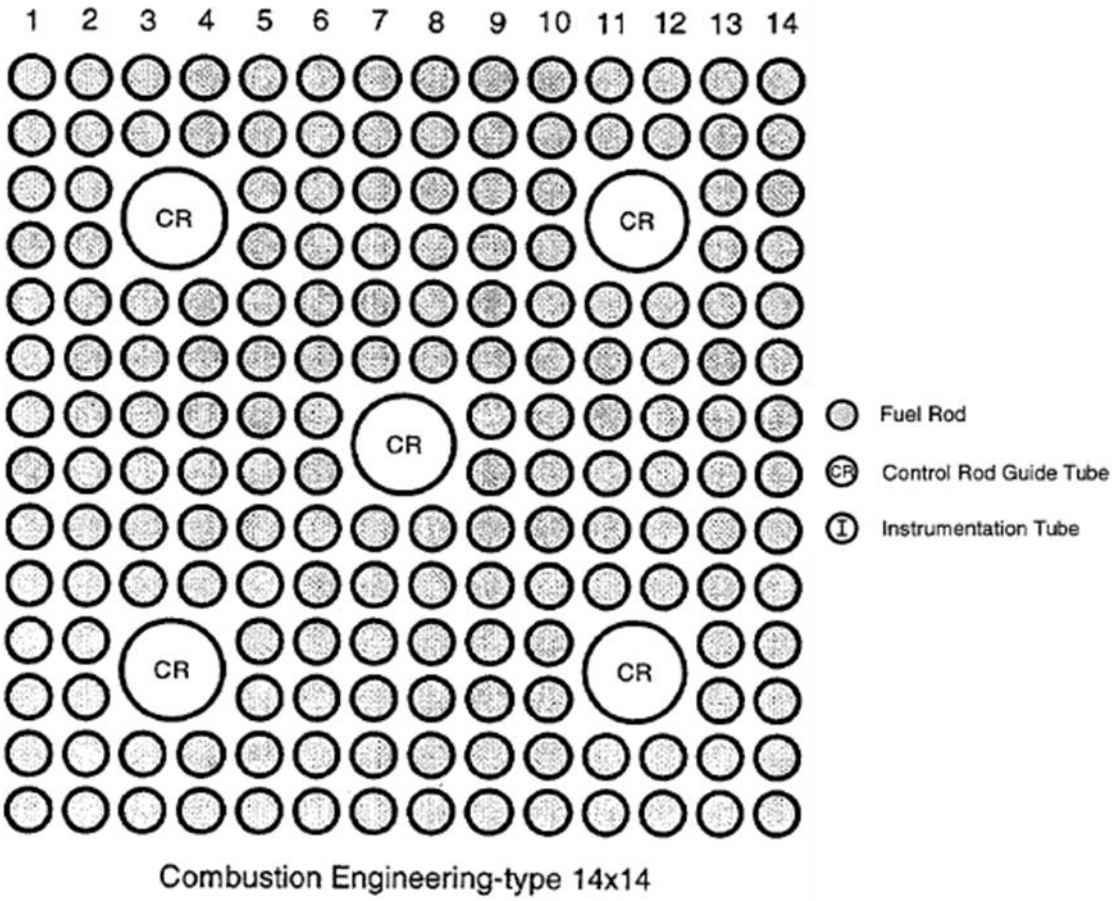


Figure 5.7 Combustion Engineering 14x14 assembly diagram

5.2.3 Westinghouse 15x15

Table 5-6 Westinghouse 15x15 critical dimensions

Fuel Type (PWR)	WE 15x15
Cladding Diameter (in.)	0.422
Rod Pitch (in.)	0.563
Pellet Diameter (in)	0.366
Pellet Composition (% U-235)	
Cladding Thickness (in)	0.024
Gap Thickness (in)	0.0038
Assembly Width (in)	
Number of Fuel Rods per Assembly	204
Number of Control Rods	20
Control Rod Diameter (in.)	0.515
Fuel Rod Length (in)	149.7
Active Length (in)	144
Clad Material	Zirccaloy-4

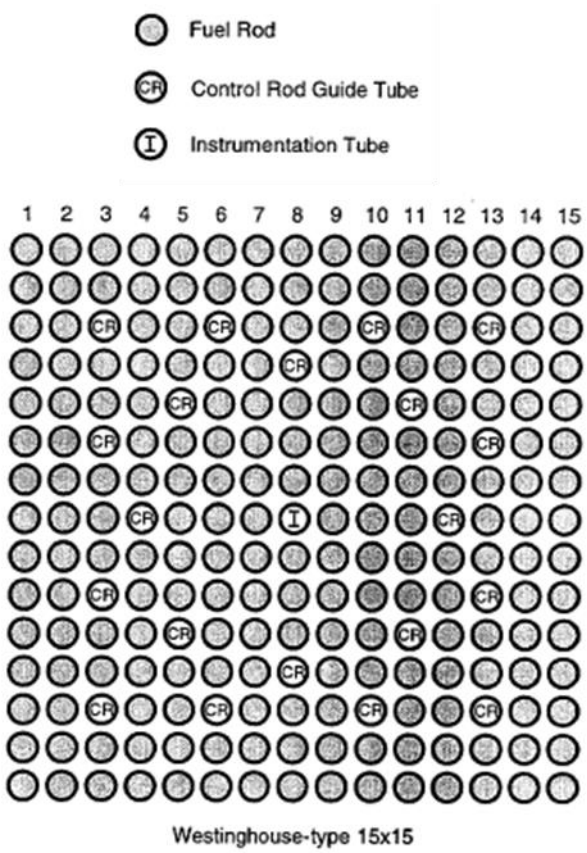


Figure 5.8 Westinghouse 15x15 assembly diagram

5.2.4 Combustion Engineering 16x16

Table 5-7 CE 16x16 Critical Dimensions

Fuel Type (PWR)	CE 16x16
Cladding Diameter (in.)	0.382
Rod Pitch (in.)	0.506
Pellet Diameter (in)	0.325
Pellet Composition (% U-235)	
Cladding Thickness (in)	0.025
Gap Thickness (in)	0.0035
Assembly Width (in)	8.25
Number of Fuel Rods per Assembly	224
Number of Control Rods	5
Control Rod Diameter (in.)	0.98/0.035 thick
Fuel Rod Length (in)	161
Active Length (in)	150
Clad Material	Zircaloy-4

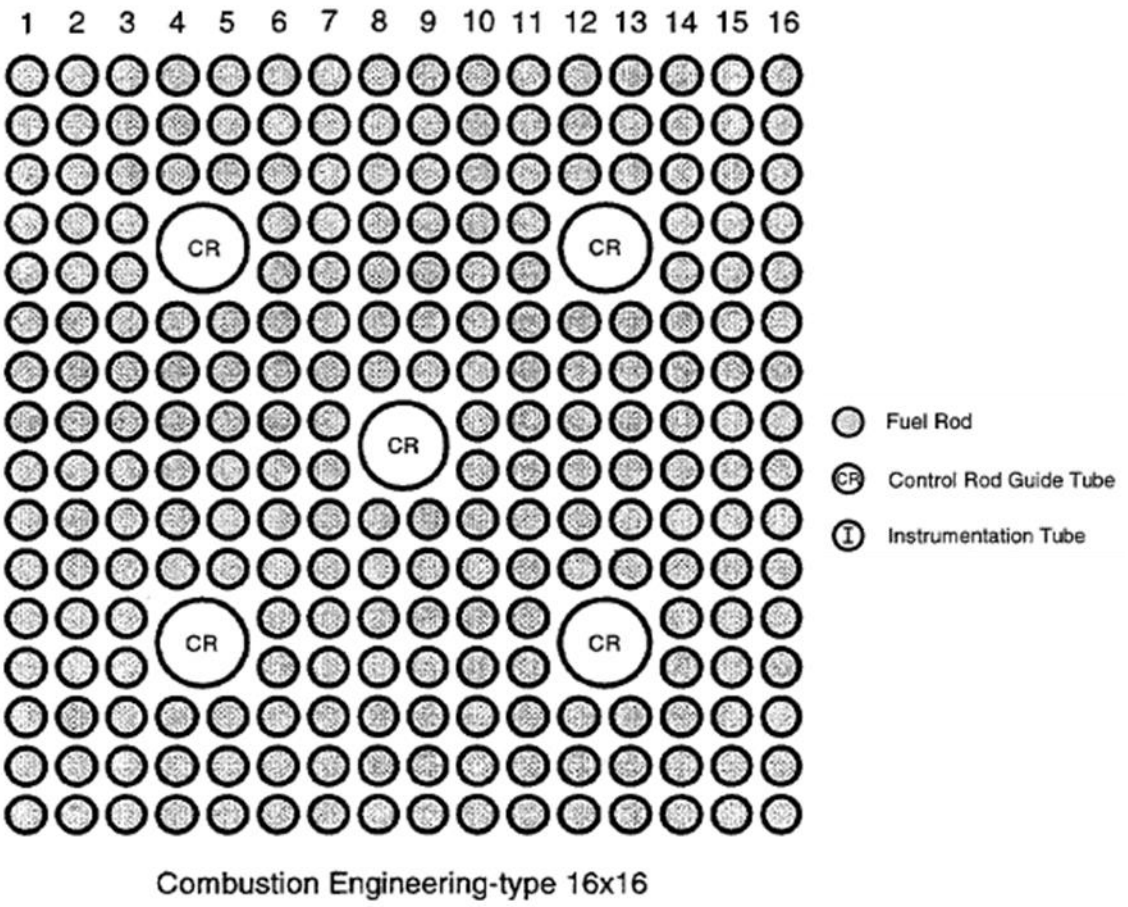


Figure 5.9 CE 16x16 Assembly Diagram

5.2.5 Westinghouse 17x17OFA

Table 5-8 WE 17x17OFA critical dimensions

Fuel Type	WE 17x17 OFA
Cladding Diameter (in.)	0.36
Rod Pitch (in.)	0.496
Pellet Diameter (in)	0.3088
Pellet Material	Uranium Oxide
Pellet Enrichment (% U235)	1.6-3.8
Cladding Thickness (in)	0.0225
Gap Thickness (in)	0.0031
Assembly Width (in)	8.434
Number of Fuel Rods per Assembly	264
Number of Control Rods	24
Control Rod Diameter (in.)	0.45
Fuel Rod Length (in)	151.635
Active Length (in)	144
Clad Material	Zircaloy-4

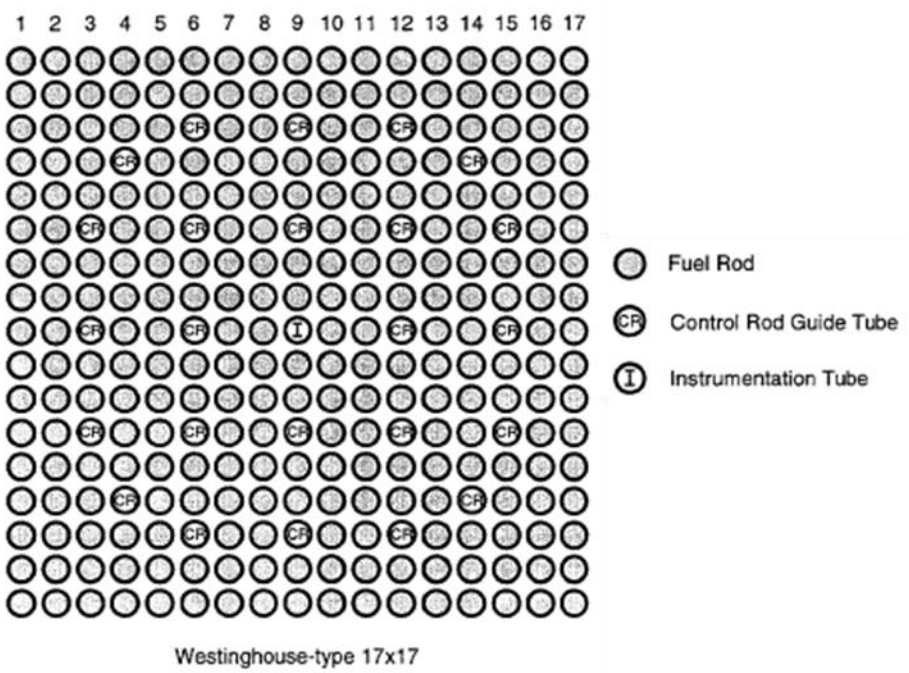


Figure 5.10 Westinghouse 17x17 Fuel Diagram

6.0 References

American Society of Mechanical Engineers. (2015). ASME Boiler and Pressure Vessel Code, Part D: Properties (Customary). New York: American Society of Mechanical Engineers, Boiler and Pressure Vessel Committee.

Banerjee K, JM Scaglione, and RA LeFebvre. 2014. *Integrated Data and Analysis Tool for Used Nuclear Fuel Management*. Transactions of American Nuclear Society (ANS) – Winter Meeting, 111, 338-341, November 9-13, 2014. Anaheim, California.

Bates JM. 1986. *Single PWR Spent Fuel Assembly Heat Transfer Data for Computer Code Evaluations*. PNL-5571, Pacific Northwest Laboratory, Richland, Washington.

Cox RL. 1977. *Radiation Heat Transfer in Arrays of Parallel Cylinders*. ORNL-5239, Oak Ridge National Laboratory, Oak Ridge, Tennessee.

Creer JM, RA McCann, MA McKinnon, JE Tanner, ER Gilbert, RL Goodman, C Dziadosz, EV Moore, DH Schoonen, M Jensen, and C Mullen. 1986. *The CASTOR-V/21 PWR Spent-Fuel Storage Cask: Testing and Analyses.* EPRI NP-4887, Electric Power Research Institute, Palo Alto, California.

Creer JM, TE Michener, MA McKinnon, JE Tanner, ER Gilbert, and RL Goodman. 1987. *The TN-24P PWR Spent-Fuel Storage Cask: Testing and Analyses*. EPRI-NP-5128/PNL-6054, Electric Power Research Institute, Palo Alto, California.

Croff AG. 1980. *ORIGEN-2 -- A Revised and Updated Version of the Oak Ridge Isotope Generation and Depletion Code*. ORNL-5621, Oak Ridge National Laboratory, Oak Ridge, Tennessee.

Cuta JM and JM Creer. 1986. *Comparisons of COBRA-SFS Calculations to Data from Electrically Heated Test Sections Simulating Unconsolidated and Consolidated BWR Spent Fuel*. EPRI-NP-4593, Electric Power Research Institute, Palo Alto, California.

Cuta JM, DR Rector, and JM Creer. 1984. *Comparisons of COBRA-SFS Calculations with Data from Simulated Sections of Unconsolidated and Consolidated BWR Spent Fuel*. EPRI-NP-3764, Electric Power Research Institute, Palo Alto, California.

Eric W. Lemmon, Mark O. McLinden and Daniel G. Friend, "Thermophysical Properties of Fluid Systems" in NIST Chemistry WebBook, NIST Standard Reference Database Number 69, Eds. P.J. Linstrom and W.G. Mallard, National Institute of Standards and Technology, Gaithersburg MD, 20899, https://doi.org/10.18434/T4D303, (retrieved October 1, 2020).

Fort JA, Richmond DJ, Cuta, JM, Suffield SR. 2018 Thermal Modeling of TN-32B Cask for High Burnup Spen Fuel Data Project. PNNL-2549 Rev.2 Pacific Northwest National Laboratory, Richland, Washington.

Fry CJ, E Livesey, and GT Spiller. 1983. *Heat Transfer in a Dry, Horizontal LWR Spent Fuel Assembly*. In ***Proceedings of Seventh International Symposium***, New Orleans, Louisiana.

Oak Ridge National Laboratory, Oak Ridge, Tennessee. From ***Packaging and Transportation of Radioactive Materials (PATRAM'83) Symposium***.

Gebhart B. 1957. *Unified Treatment for Thermal Radiation Transfer Processes – Gray, Diffuse Radiators and Adsorbers*. Paper No. 57-A-34. ASME. December 1957.

Gebhart B. 1961. *Surface Temperature Calculations in Radiant Surroundings of Arbitrary Complexity for Gray, Diffuse Radiation*. *Int. J. Heat Mass Transfer*, vol. 3(4), 341-346.

Gebhart B. 1971. **Heat Transfer**. 2nd edition. McGraw-Hill Book Company. New York, New York.

George TL, KL Basehore, CL Wheeler, WA Prather, and RE Masterson. 1980. *COBRA-WC: A version of COBRA for Single-phase Multi-assembly Thermal-Hydraulic Transient Analysis*. PNL-3259, Pacific Northwest Laboratory, Richland, Washington.

Gosman AD, R Herbert, SV Patankor, R Potter, and DB Spalding. 1973. *The SABRE Code for Prediction of Coolant Flows and Temperatures in Pin Assemblies Containing Blockages*. Presented at ***International Meeting on Reactor Heat Transfer***, Karlsruhe, Germany. HTS/73/47, Imperial College of Science and Technology.

Hirt CW and JL Cook. 1972. *Calculating Three-Dimensional Flows Around Structures and Over Rough Terrain*. ***Journal of Computational Physics*** 10:324-340.

Hottel HC and AF Sarofin. 1967. **Radiative Heat Transfer**. McGrawHill, Inc., New York, New York.

Ingesson L and S Hedberg. 1970. *Heat Transfer Between Subchannels in a Rod Assembly*. Paper No. FC7.11. Presented at the ***Fourth International Heat Transfer Conference***, Versailles, France.

Irino M, M Oohashi, T Irie, and T Nishikawa. 1986. *Study on Surface Temperatures of Fuel Pins in Spent Fuel Dry Shipping/Storage Casks*. IAEA-SM-286/139P, pp. 585-598.

JA Barsic, ME Conner, AM Everhard, et al. 17x17 Next Generation Fuel (17x17 NGF) Reference Core Report. Westinghouse Electric Company LLC, Monreovill, PA, March 2008

Khan EU, WA Prather, TL George, and JM Bates. 1981. *A Validation Study of the COBRA-WC Computer Program for LMFBR Thermal-Hydraulic Analysis.* PNL-4128, Pacific Northwest Laboratory, Richland, Washington.

Lindgren, Eric R., and Durbin, Samuel. Materials and Dimensional Reference Handbook for the Boiling Water Reactor Dry Cask Simulator.. United States: N. p., 2017. Web. doi:10.2172/1559567.

Lombardo NJ, JM Cuta, TE Michener, DR Rector, and CL Wheeler. 1986a. *COBRA- SFS: A Thermal-Hydraulic Analysis Computer Code; Volume III: Validation Assessments*. PNL-6049, Vol. 3, Pacific Northwest Laboratory, Richland, Washington.

Lombardo NJ, TE Michener, CL Wheeler, and DR Rector. 1986b. *COBRA-SFS Predictions of Single-Assembly Spent Fuel Heat Transfer Data*. PNL-5781, Pacific Northwest Laboratory, Richland, Washington.

Malin, D H. Domestic light water reactor fuel design evolution. Volume III. United States: N. p., 1981. Web.

Morgan Advanced Materials, “Kaowool® Organic Boards Product Data Sheet.” October 2, 2020.

McKinnon MA, JM Cuta, and UP Jenquin. 2002. *Effect of Loading Pattern on Thermal and Shielding Performance of a Spent Fuel Cask*. In ***Proceedings of 10th International Conference on Nuclear Engineering*** **(ICONE10)**, Arlington, Virginia, April 14-18, 2002.

McKinnon MA, RE Dodge, RC Schmitt, LE Eslinger, and G Dineen. 1992. *Performance Testing and Analyses of the VSC-17 Ventilated Concrete Cask*. EPRI-TR-100305/PNL-7839, Electric Power Research Institute, Palo Alto, California.

McKinnon MA, TE Michener, MF Jensen, and GR Rudman. 1989. *Testing and Analyses of the PWR Spent Fuel Dry Storage Cask Loaded with Consolidated Fuel*. EPRI-NP-6191/PNL-6631/UC-85, Electric Power Research Institute, Palo Alto, California.

Metals Handbook, Vol.2 - Properties and Selection: Nonferrous Alloys and Special-Purpose Materials, ASM International 10th Ed. 1990.

Michener TE, DR Rector, JM Cuta, RE Dodge, and CW Enderlin. 1995. *COBRA-SFS: A Thermal-Hydraulic Code for Spent Fuel Storage and Transportation Casks*. PNL-10782, Pacific Northwest Laboratory, Richland, Washington.

NUREG/CR-6886, Rev. 2. 2009. *Spent Fuel Transportation Package Response to the Baltimore Tunnel Fire Scenario.* U.S. Nuclear Regulatory Commission, Washington, D.C.

NUREG/CR-6894, Rev. 1. 2007. *Spent Fuel Transportation Package Response to the Caldecott Tunnel Fire Scenario.* U.S. Nuclear Regulatory Commission, Washington, D.C.

NUREG/CR-7206. 2016. *Spent Fuel Transportation Package Response to the MacArthur Maze Fire Scenario*, U.S. Nuclear Regulatory Commission, Washington, D.C.

NUREG/CR-7207. 2016. *Spent Fuel Transportation Package Response to the Newhall Pass Fire Scenario*, U.S. Nuclear Regulatory Commission, Washington, D.C.

NUREG/CR-7209. 2017. *A Compendium of Spent Fuel Transportation Package Response Analyses to Severe Fire Accident Scenarios*, U.S. Nuclear Regulatory Commission, Washington, D.C.

Poljak G. 1935. *Analysis of Heat Interchange by Radiation between Diffuse Surfaces*. *Tech. Phys. USSR*, Vol. 1 (5-6), 555-590.

Rector DR and TE Michener. 1989. *COBRA-SFS Modifications and Cask Model Optimization*. PNL-6706, Pacific Northwest Laboratory, Richland, Washington.

Rector DR, CL Wheeler, and NJ Lombardo. 1986a. *COBRA-SFS: A Thermal-Hydraulic Analysis Computer Code, Volume 1: Mathematical Models and Solution Method*. PNL-6049, Vol. 1, Pacific Northwest Laboratory, Richland, Washington.

Rector DR, JM Cuta, and NJ Lombardo. 1986b. *COBRA-SFS Thermal-Hydraulic Analyses of the CASTOR-IC and REA 2023 BWR Storage Casks Containing Consolidated Spent Fuel*. PNL-5802, Pacific Northwest Laboratory, Richland, Washington.

Rector DR, RA McCann, UP Jenquin, CM Heeb, JM Creer, and CL Wheeler. 1986c. *CASTOR-1C Spent Fuel Storage Cask Decay Heat, Heat Transfer, and Shielding Analyses*. PNL-5974, Pacific Northwest Laboratory, Richland, Washington.

Rector DR, TE Michener, and JM Cuta. 1998. *Verification and Validation of COBRA-SFS Transient Analysis Capability*. PNNL-11883, Pacific Northwest National Laboratory, Richland, Washington.

Rector DR. 1987. *RADGEN: A Radiation Exchange Factor Generator for Rod Bundles*. PNL-6342, Pacific Northwest Laboratory, Richland, Washington.

Reid RC, JM Prausnitz, and BE Poling. 1987. **The Properties of Liquids and Gases**. McGraw-Hill International, Fourth Edition. New York.

Richmond DJ, Jensen BJ, Grant CL, and Hom BM. 2021. *COBRA-SFS: Theory Manual*. PNNL-32245, Pacific Northwest Laboratory, Richland, Washington.

Rogers JT and NE Todreas. 1968. *Coolant Mixing in Reactor Fuel Rod Assemblies--Single Phase Coolants*. In ***Heat Transfer in Rod Assemblies*****,** ASME, pp. 1-56.

Rogers JT and RG Rosehart. 1972. *Mixing by Turbulent Interchange in Fuel Assemblies: Correlations and Inferences*. American Society of Mechanical Engineers (ASME) Paper No. 72-HT-53.

Rowe DS. 1973. *COBRA-IIIC: A Digital Computer Program for Steady-State and Transient Thermal-Hydraulic Analysis of Rod Bundle Nuclear Fuel Elements*. BNWL-1695, Battelle Northwest Laboratory, Richland, Washington.

Siegel R. and J.R. Howell. 1981. **Thermal Radiation Heat Transfer**. MacGraw-Hill Book Company, New York, New York.

Slattery JC. 1972. *Momentum, Energy, and Mass Transfer in Continua*. McGraw-Hill, Inc., New York.

Stewart CW, CL Wheeler, RJ Cena, CA McMonagle, JM Cuta, and DS Trent. 1977. *COBRA-IV: The Model and the Method*. BNWL-2214, Battelle Northwest Laboratory, Richland, Washington.

Suffield SR, Richmond DJ, Fort JA, 2019. Modeling of Dry Cask Simulator. PNNL-XXXX Pacific Northwest National Laboratory, Richland, Washington.

Tong LS. 1968. *Pressure Drop Performance of a Rod Assembly*. In ***Heat Transfer in Rod Assemblies***, American Society of Mechanical Engineers (ASME), pp. 57-69.

USDOE Office of Civilian Radioactive Waste Management, Washington, DC (1987). Characteristics of spent fuel, high-level waste, and other radioactive wastes which may require long-term isolation: Appendix 2A, Physical descriptions of LWR [Light-Water Reactor] fuel assemblies (DOE/RW--0184-Vol3). United States

Wheeler CL, RA McCann, NJ Lombardo, DR Rector, and TE Michener. 1986. *HYDRA and COBRA-SFS Temperature Calculations for CASTOR-1C, REA-2023, CASTOR-V/21, and TN-24P Casks*. In ***Proceedings Third International Spent Fuel Storage Technology Symposium and Workshop***, Vol. 1, S77-S98, CONF-960417, National Technical Information Service, Springfield, Virginia.

Wiles LE, NJ Lombardo, CM Heeb, UP Jenquin, TE Michener, CL Wheeler, JM Creer, and RA McCann. 1986. *BWR Spent Fuel Storage Cask Performance Test, Volume II: Pre- and Post-Test Decay Heat, Heat Transfer, and Shielding Analyses*. PNL-5777, Vol. 2, Pacific Northwest Laboratory, Richland, Washington.

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354
1-888-375-PNNL (7665)

www.pnnl.gov