ORNL/TM-2021/1885

A Verification and Validation Approach for COMSOL Multiphysics to Support the High Flux Isotope Reactor (HFIR)

Trevor K. Howard
Prashant K. Jain

July 2021

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE US DEPARTMENT OF ENERGY

ORNL/TM-2021/1885

Nuclear Energy and Fuel Cycle Division
Research Reactors Division

A Verification and Validation Approach for COMSOL Multiphysics to Support the High Flux Isotope Reactor (HFIR)

Trevor K. Howard
Prashant K. Jain

July 2021

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831-6283
managed by
UT-BATTELLE, LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

ACRONYMS

AIAA	American Institute of Aeronautics and Astronautics
ANS	American Nuclear Society
ASME	American Society of Mechanical Engineers
ATR	Advanced Test Reactor
DoD	US Department of Defense
HEU	highly enriched uranium
HFIR	High Flux Isotope Reactor
HPRR	high-performance research reactor
INL	Idaho National Laboratory
LEU	low-enriched uranium
MMS	method of manufactured solutions
NACA	National Advisory Committee for Aeronautics
ORNL	Oak Ridge National Laboratory
SQA	software quality assurance
SSHTC	steady-state heat transfer code
V&V	verification and validation

ABSTRACT

Over the last several decades, many reactors have successfully been converted from highly enriched uranium (HEU) to low-enriched uranium (LEU) fuels in the United States in support of its global nonproliferation objectives. Of the reactors slated for conversion, five high-performance research reactors (HPRRs) remain. The High-Flux Isotope Reactor (HFIR) at Oak Ridge National Laboratory (ORNL) is one of the five. Conversion of HFIR requires the qualification of new fuel. To aid in the conversion process and improve safety margins, COMSOL Multiphysics was chosen to support and eventually supplement the steady-state heat-transfer code (SSHTC).

The COMSOL code must undergo verification and validation to be used as a supplementary tool for safety basis calculations, an essential process for using any software to assess nuclear systems. However, physics-based solvers have a unique challenge in that the validation is much more rigorous, and the domain is only applicable to the general scope of the problem.

This document details the verification and validation of COMSOL for HFIR analysis. This report addresses the needs of two different types of users: (1) current and future users of COMSOL for HFIR, and (2) individuals interested in assessing the scope of the validation of COMSOL for HFIR.

This report includes six sections, the first of which details the process of verification and validation. Section 1 provides an overview of key concepts from the American Society of Mechanical Engineers (ASME) Standards for Verification and Validation (V&V) 10, 20, and 40 and illustrates how these concepts are used to validate any multiphysics software. Section 2 covers the approach needed to validate multiphysics software for a given HFIR analysis. Sections 3 through 5 address the key concepts highlighted in Section 2. The final section provides concluding remarks demonstrating the validation study's scope and limitations and describes how these may be expanded and improved.

The purpose of this report is three-fold:

(1) It outlines the process and steps required for validation of any commercial multiphysics tool and the steps the user must take,

(2) It highlights the efforts and limitations of the validation problems simulated with COMSOL and their corresponding experiments,

(3) It supports the validity of COMSOL as a safety basis tool for use in HFIR.

As a final note, this is a living document that will be updated regularly in the future as more knowledge is gained, as the code is tested on more problems, and as results become available. Procedures may change, and better recommendations may be made as the code is further implemented.

1. VERIFICATION AND VALIDATION

1.1 IMPORTANT CONCEPTS IN VERIFICATION AND VALIDATION

Validation is a multi-step process. Several publications detail and standardize the verification and validation process for different software. In this report, the verification and validation process was chosen to reflect ASME Standards V&V 10, 20, and 40 [2]–[4] because they are based on previous standards and recommendations from the American Institute of Aeronautics and Astronautics (AIAA) and the US Department of Defense (DoD) [2].

1.1.1 Key Definitions

The following definitions, adapted from ASME V&V 10 report [2], are provided below to maintain consistency.

Accuracy: the degree to which the result of a measurement, calculation, or specification conforms to the correct value or a standard. Accuracy is a combination of *trueness* and *precision*.

Calibration (simulation): the process of adjusting physical modeling parameters in the computational model to improve agreement with experimental data.

Calculation: the process of selecting model inputs and running a simulation to determine a previously unknown parameter of interest.

Code: the computer implementation of algorithms developed to facilitate the formulation and to approximate the solution of a class of problems.

Completeness: the ability of a validation domain problem to capture all representative physical phenomena.

Model: the conceptual, mathematical, and numerical representation of the physical phenomena needed to represent specific real-world conditions and scenarios, including the geometric representation, governing equations, boundary, and initial conditions, loadings, constitutive models and related material parameters, special and temporal approximations, and numerical solution algorithms.

Representation: the degree to which the validation experiment is representative of the desired model application.

Problem, application: the problem for which there is a desired solution. In the context of this report, it is any safety basis or HFIR-related calculation.

Problem, validation: the problem used to validate the code. For this problem, a solution already exists (typically, experimental results).

Trueness: the difference between the true value and the experimental or simulation value: typically, that of the validation problem value and that of the simulation.

Validation: the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model.

Verification: the process of determining that a computational model accurately represents the underlying mathematical model and its solution.

1.2 VERIFICATION

Verification, while often listed as separate from validation, is better stated as a precursor or subset of validation. Verification is the process of determining whether the computational model is accurately representative of the mathematical model. A model that fails verification would not be valid. This is an important distinction because it allows verification to be lumped into validation as long as the context is provided regarding the nature of the validation. Typically, verification problems include nonexperimental comparisons such as analytic solutions, the method of manufactured solutions, and code-to-code comparisons. The primary advantage of distinguishing verification problems from validation problems is that verification solutions are already exactly known, so they provide a high degree of precision when finding the solution. If a code has been verified, then a failure of validation can be traced to inherent problems in the modeling choices, not a failure of the code. This report takes a potentially controversial stance and describes verification problems as validation problems. In other words, any comparative solution is considered validation as long as accuracy and representation are addressed. This is the only instance in which verification is incorporated into validation.

Outside of verification problems, the ASME V&V standards identify two types of verification: code verification and calculation verification. The former depends on the software developer, and the latter depends on the software user. Here, verification is viewed outside the context of validation. Software verification is the process of ensuring that changes to the software do not invalidate a previous validation. Solution verification ensures that the user has appropriately derived a solution while using the software. These two forms of verification are not only important steps in the validation process, as a solution is not considered valid unless both the software and simulation have been confirmed, but they also have an additional purpose after a tool has undergone validation.

1.2.1 Code Verification

Code verification is the act of verifying the underlying code functions as intended. Code verification activities are performed by the developer instead of the user. Code verification includes the numerical verification of the solver and software quality assurance (SQA).

SQA is the complete process of documentation and testing, which demonstrates that changes to the code do not impact the solution. To this end, the goal of SQA is to ensure that the software functions correctly and as intended. If the code is being developed or modified, SQA has an even greater role since changes must be tracked all throughout the process.

A numerical verification of the code shows that the underlying mathematical models are being implemented appropriately. This is achieved through a verification process in which the model's behavior and end solution are explicitly known and can be compared directly to the model results.

Code verification is primarily the role of the software developer as opposed to the end-user. In the context of this report, it is not imperative to run code verification to the rigor required since the code was developed out of house. The verification for this code falls within the framework of validation and SQA. An important exception applies when code is used to modify inputs based on the solution or, as can be done in COMSOL when differential equations are created by the user to be implemented in the solution.

1.2.2 Calculation Verification

Calculation verification is the act of ensuring that all model inputs were chosen appropriately and that the solution was fully converged or sufficiently converged to approximate the final desired solution. For input verification this is often accomplished through a formal check-and-review process. Grid and temporal convergence are often described as *using a grid convergence index*, as suggested within ASME V&V 10 and 20 [2], [3]. However, ASME V&V 40 [4] allows for the user to determine the convergence method. Whatever the methodology chosen, there must be a means to assess whether the model is converged. In some cases, this assessment could be accomplished without multiple trials, especially if the problem is similar to a previously converged study. In such cases, the mesh can be appropriately scaled based on changes to the inputs and geometry.

Beyond calculation verification, some convergence studies may be used to determine the asymptotic solution when the mesh is fully resolved. Perhaps the oldest but most robust approach is that of Richardson extrapolation, as suggested in ASME V&V 20 [3]. However, this approach has its own issues, especially in more complex geometries. Estimation of a fully converged solution is not a requirement for calculation verification; however, if a rigorous verification is performed, then the final estimation of the solution must be calculated.

1.3 VALIDATION

Validation is extremely important for codes, especially when the codes are used to make safety decisions. Even though no software can ever be 100% validated for the application problem, validation should be viewed as a continuing, graded process, so the level of validation depends on the importance and impact of the study. Instead of explicitly dictated requirements, the weighted approach to validation is the approach used for ASME V&V 40 [4], which is the standard for verification and validation for medical isotopes.

At the highest level, validation is a comparison of physics to a model. Typically, this is done through experiments. A validation experiment is conducted, and the results are compared to that of the model results. If the model results agree with the experiment, then the model is typically said to be validated. The reality is slightly more complicated than this because the uncertainty of the experiment must be considered. This process is reasonable for determining the accuracy of the model for the validation problem, but sometimes even this is not true. Model designs often have simplifications to make the problem reasonable to solve. Even if the simplifications are negligible, the validation experiment may be oversimplified and may not fully represent the application problem conditions. Most problems closest to the application problems were not designed as validation problems, so there is usually a larger uncertainty in the experiment. This may result in a model that is valid for the validation experiment but invalid for the application problem. If the validation problem is both accurate and representative of the application problem, then the complexity of multiphysics is still an issue. An application may be valid for individual physics models, yet the application model is missing key physics or is invalid when the multiple physics are included.

The combined uncertainties make the process of assigning an objective validation quantity daunting. However, the challenge of validation is not insurmountable. Various aspects impact the validity of a study. This report divides software validation into three different domains—representation, completeness, and accuracy—all of which are discussed in the following sections. At its core, validation is the process of demonstrating that a model is suitable for determining a solution. The level of validation required depends on the goal of the model and its alignment with the validation metrics of representation, completeness, and accuracy.

1.4 REPRESENTATION

Representation is the degree to which the validation problem epitomizes the application problem. Any problem may be used to validate a code, but ultimately, the validation problem must sufficiently represent the application problem: that is, the validation problem must be valid. This gives rise to the concepts of the *validation domain* and the *application domain*. The validation domain includes the cases in which the code has been validated, and the application domain is the region for which the code is intended to be used.

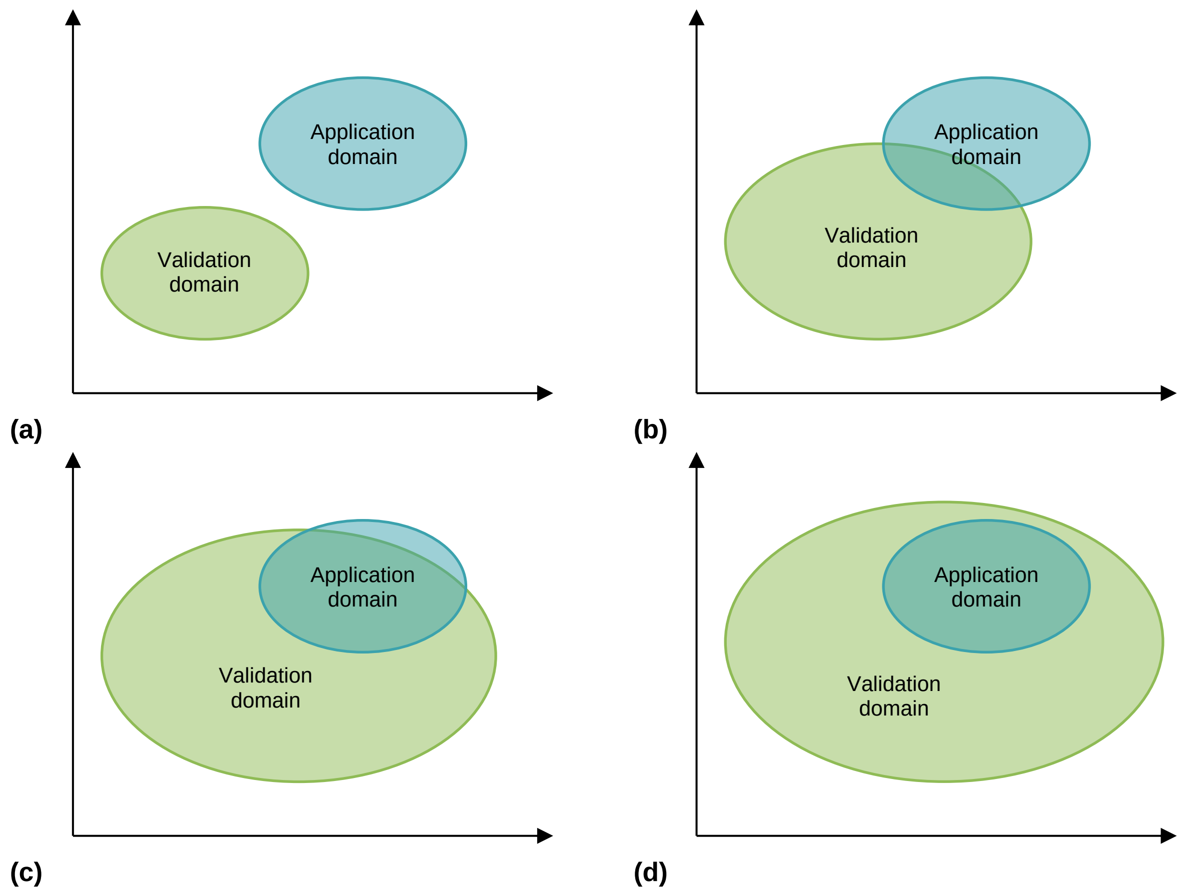

Figure 1.1. Validation domain vs. application domain regimes.

Figure 1.1 illustrates four validation regimes: (a) the application of the code is outside the known validation of the code, (b) aspects of the code are validated, but the application exists outside the validation domain, (c) the code has been sufficiently validated for the application, and (d) the application is a validation problem to better assess the validity of the problem in application space.

To objectively determine whether the problem is representative, the individual aspects can be examined. In this sense, representation can be thought of on the basis of a viewpoint hierarchy. The validation problem must first represent the application problem in the global sense, and then it must capture important local phenomena. Finally, the problem must account for as-built design changes that may generate discrepancies between the as-built application model and the as-built application design. Using this hierarchy, an approximate metric for representation is presented in Table 1.1.

Table 1.1: Representation metrics for validation problems

Level	Metric	Global	Local	Design
-	The validation problem has no physical basis in relation to the application problem	×	×	×
1	The validation problem includes the necessary physics but is not very similar to the application problem	×	×	×
2	The validation problem is similar to the application problem but has differences that may result in different global phenomena	?	×	×
3	The validation problem is representative of the application problem in a global sense, but it differs in ways that may result in different local phenomena. Unscaled variation of the design means design-specific phenomena are lost	✓	?	×
4	The validation problem is a scalable or representative version of the application problem. Concessions made could impact as-built effects that may otherwise be revealed in validation testing.	✓	✓	?
5	The validation problem is fully representative of the application problem.	✓	✓	✓
+	The validation problem is the application problem.	✓	✓	✓

It is important to note that the level in the hierarchy is based on the definitions of what is considered a local, global, or design effect. In its most rigorous sense, only a prototypic validation experiment or co-simulation can meet the needs of a level 5 validation. However, depending on the desired parameter, anything could be defined as a level 5 validation.

One important aspect of representation that must be considered is that of the boundary conditions and their sensitivity. In most cases, boundary conditions are design phenomena, and scaling is appropriate; however, boundary conditions may drive the validity of the solution, as well.

1.5 COMPLETENESS

Multiphysics problems further complicate the problem. In some application problems, multiple physics are needed to provide valid results. This represents the concept of *completeness*. For a fully integrated system, all aspects must be modeled. Completeness and representation are similar, as they both address how similar the validation problem is when compared to the application problem. The distinction is that *completeness* examines whether a physics model or component is needed, whereas representation addresses whether model simplifications and changes are appropriate.

An example is illustrated by the process for determining pressure drop through a heated channel. *Representation* addresses whether the inputs, dimensions, and fluid properties are sufficient to validate the application problem, whereas *completeness* analyzes whether a heating model or temperature-dependent properties are needed. The metrics for completeness are presented in

Table 1.2 below.

Table 1.2. Completeness metrics for validation problems

Level	Metric	Major impacts	Substantial impacts	Minor impacts
-	The validation problem captures none of the dominant physics or components	✓	✓	✓
1	The validation captures some of the major physics and components, but the missing physics have a major impact on the solution	✓	✓	✓
2	The validation problem captures the dominant physics and components of the application, but situations may result in the potential for major impacts affecting the application problem	?	✓	✓
3	The validation problem covers all major impacts to the solution from the physics and components, but minor impacts could become substantial under certain conditions	✗	?	✓
4	The validation problem captures the key physics, but some missing physics or components may have a minor impact on the validation problem	✗	✗	?
5	The validation problem captures all the important physics and components that contribute to the desired application problem	✗	✗	✗
+	The validation problem includes all physics of the problem	✗	✗	✗

The completeness metric provides an assessment of whether the validation is occurring for a subset of the multiphysics or for the full application problem. In some cases, having a lower completeness value is necessary to assess an individual physics model, providing confidence in a complete model's representativeness and accuracy.

1.6 ACCURACY

Of the three validation metrics, *accuracy* is the most quantifiable, as it is a direct comparison between the validation experiment and the simulation results. Accuracy in the context of this report is not whether or not the solution is correct, but it is rather the combination of correctness and precision. A validation model may have a low accuracy level if the validation experiment has a large uncertainty; low accuracy does not only result from a significantly different model and comparison values.

For the validation problem, in order to assess accuracy, the uncertainty of the validation model and solution must be determined. There are various methods with which to determine the uncertainty in the validation model and the validation solution, depending on the approach used to verify the model and obtain a solution.

The accuracy of the validation depends on the accuracy of the application model. The application model's accuracy is based on the numerical verification, the uncertainty of the inputs and sensitivity, and the desired range in which the application solution value should fall. The validation accuracy can be viewed as an additional term of uncertainty that contributes to the overall application accuracy. A metric for accuracy is provided in Table 1.3 below.

Table 1.3. Accuracy metrics for validation problems

Level	Metric
-	The model solution is unproven or is inaccurate
1	The accuracy of the validation problem drives the accuracy of the application solution
2	The accuracy of the validation problem provides a major contribution to the application accuracy
3	The accuracy of the validation problem provides a substantial contribution to the application accuracy
4	The accuracy of the validation problem provides a minor contribution to the application accuracy
5	The accuracy of the validation problem is insignificant compared to the application accuracy
+	The accuracy of the validation problem is negligible

The accuracy of the validation is only as important as the degree of accuracy needed. Therefore, the validation accuracy is only partially dependent on the validation problem and requires an application problem to determine whether the accuracy of the validation is sufficient.

2. VERIFICATION AND VALIDATION OF MULTIPHYSICS SOFTWARE FOR HFIR APPLICATIONS

2.1 VERIFICATION

As mentioned in the previous section, verification is predominately a subset of the validation process. However, unlike validation, a verification must be performed for each individual analysis problem. The main objective is solution verification because each solution to each problem must be verified prior to analysis. Software verification, on the other hand, is only needed when configuration changes occur. The process of verification is outlined in Figure 2.1 for cases in which a new application problem is presented.

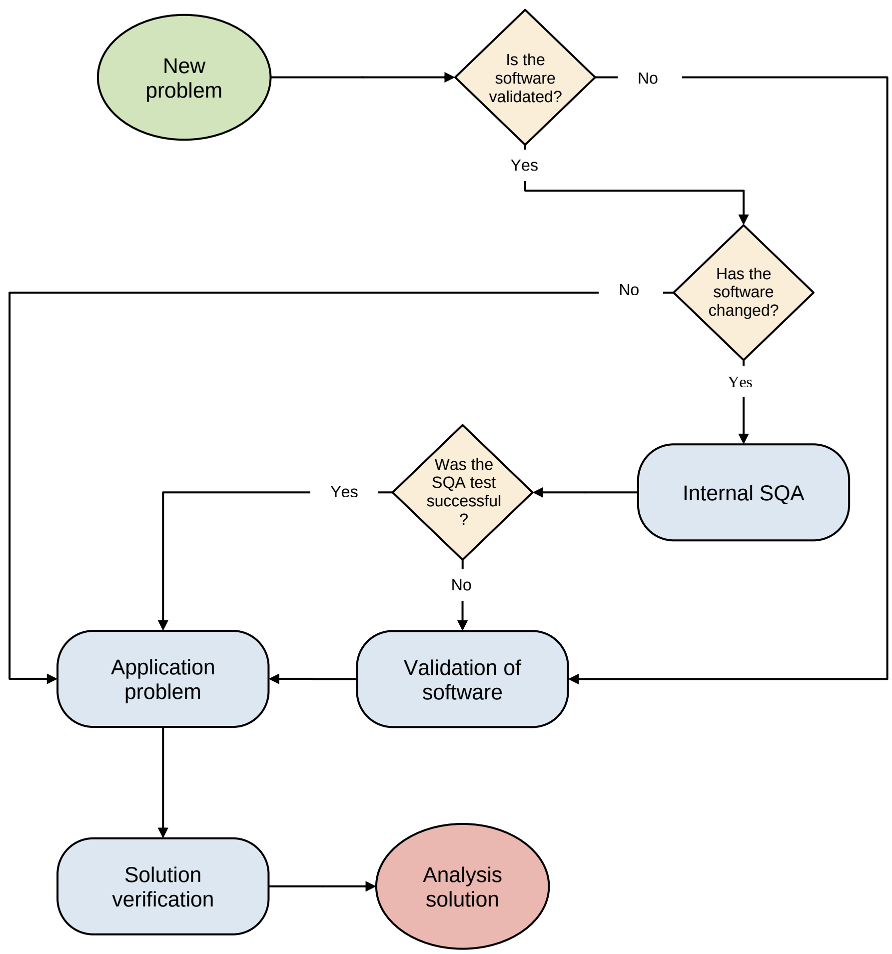

Figure 2.1. Verification and validation process for using a multiphysics software for HFIR applications.

The verification process covers the problem throughout its cycle, from the start through the achievement of a final solution. The first question is to ask whether the software is valid for the new application. If not, then another validation of the code for the application problem must occur. If the software is valid, then the next step is to determine whether the software has changed since the validation. If it has, then SQA must be performed. If the SQA determines that the software is still valid—that no substantial changes to the code occurred—then the application problem may be solved using the model. If the software is not valid, then validation should be performed again. Once the application problem is solved, the solution must be verified before it can finally be accepted as the analysis solution.

2.1.1 Verification in Validation

As mentioned above, there were two types of verification: software verification and solution verification. The former is not an important aspect of this report because any off-the-shelf code usually has its own verification and SQA process. However, a user may or may not be privy to these processes, and even if the user knows the verification processes, they may not be sufficient for accomplishing true software verification. To use the code to derive a solution, software verification is addressed through validation. A full numeric assessment of COMSOL may be made, and it has been for a few problems, but it is not necessary because the code was not developed in-house, and the validation problems are used to assess whether the code is suitable to use for an application such as HFIR.

As a part of the validation process, solution verification is necessary to determine the accuracy of the validation. Therefore, grid independence of the numerical solution is ascertained for each validation problem.

2.1.2 Verification Post-Validation

It is important to note that the verification process does not end once a verification and validation report has been finalized. Instead, a verification must be performed for every solution. As mentioned in the previous section, there are various methods for solution verification, and the following three goals must be achieved:

1. The validity of the validation problem for the application must be verified
2. The model inputs must be verified
3. The solution needs to be shown to be grid-independent

Even though the software has been subjected to a validation process, the solution obtained by the model may not be accurate or valid. Every new application problem should be examined to determine whether the changes that were made would impact the validity of the solution. The next step is to verify that the constructed model contains the required inputs, such as the geometry dimensions, material properties, physics choices, and mesh generation techniques. Once the model construction has been verified, the solution must also be verified. Typically, a grid convergence study is conducted, but in many cases, especially if the mesh has been used before the inputs were changed, a grid convergence can be shown to be independent of a previously obtained solution.

2.2 VALIDATION

Validation is performed to demonstrate that the code is valid for the domain of the application. The key to validation is finding problems that are as representative and complete as possible. While accuracy is desired, it is not determined until after the solution is obtained, so initially, it is better to find problems that are as complete and representative as possible. In Section 3, validation problems are summarized to show the relevance of problems that were chosen to be simulated in COMSOL to meet as many of the validation requirements as possible for HFIR core applications.

2.2.1 Validation Challenges with Multiphysics Approaches

A common theme throughout a coupled multiphysics analysis is the characterization of the involvement of various physics components. In addition to well-established, well-understood thermal-hydraulics considerations, fluid-thermal-structural interactions can also occur and may be highly nonlinear. For the HFIR geometry, the key physics that must be modeled are neutronics, solid and fluid heat transfer, fluid dynamics, and structural mechanics.

A diagram of the multiphysics coupling is shown in Figure 2.2.

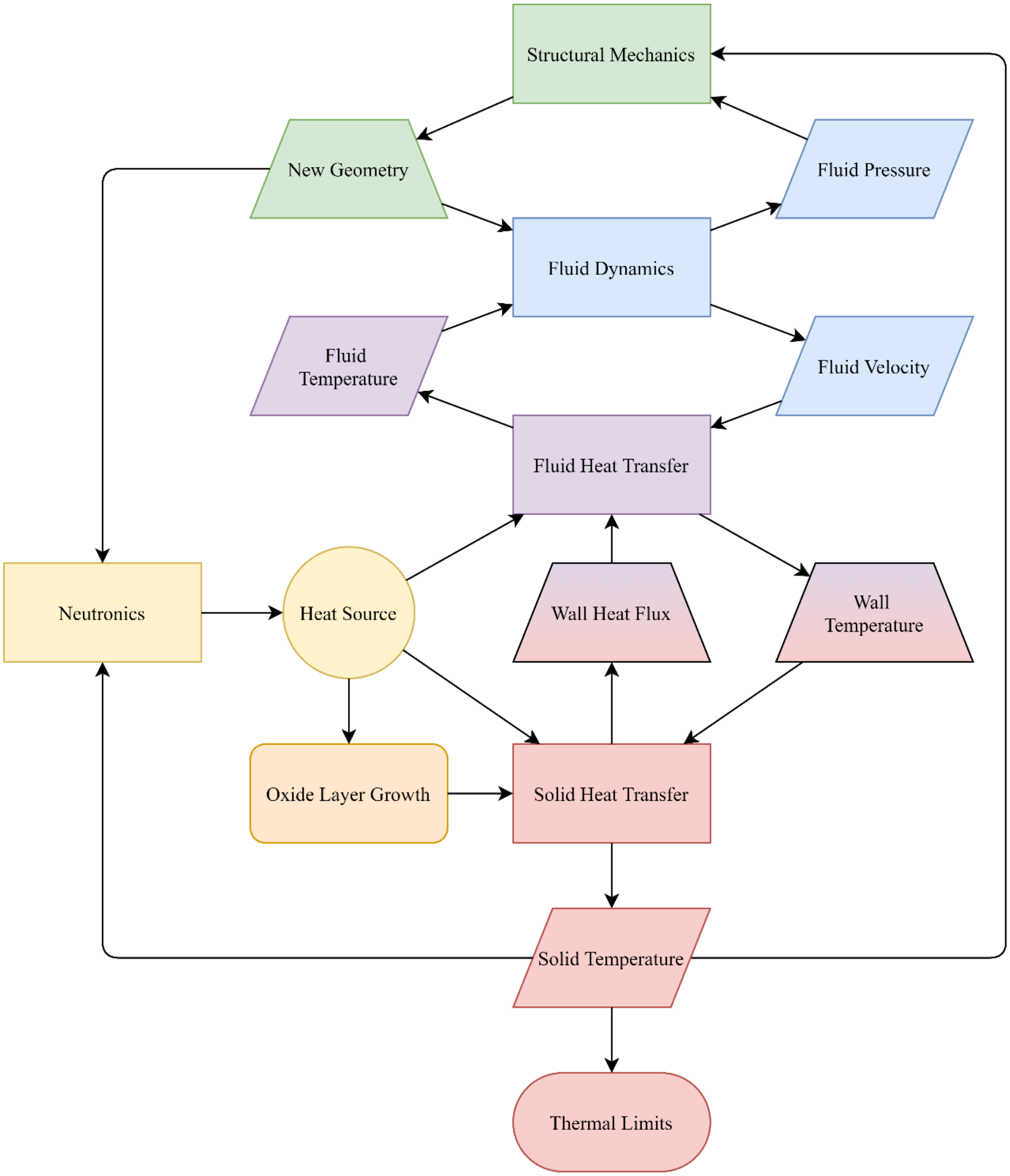

Figure 2.2. Simplified thermal oriented multiphysics layout for HFIR core applications.

The diagram in Figure 2.2 demonstrates how difficult it can be to determine completeness since every aspect of the multiphysics problem is codependent. To help justify the degree of representation and completeness for a problem, it is useful to create a hierarchy of multiphysics to help determine which physics allows individual application problems to be complete and representative for the final problem.

A sample of the hierarchical validation requirements used for HFIR is presented in Figure 2.3.

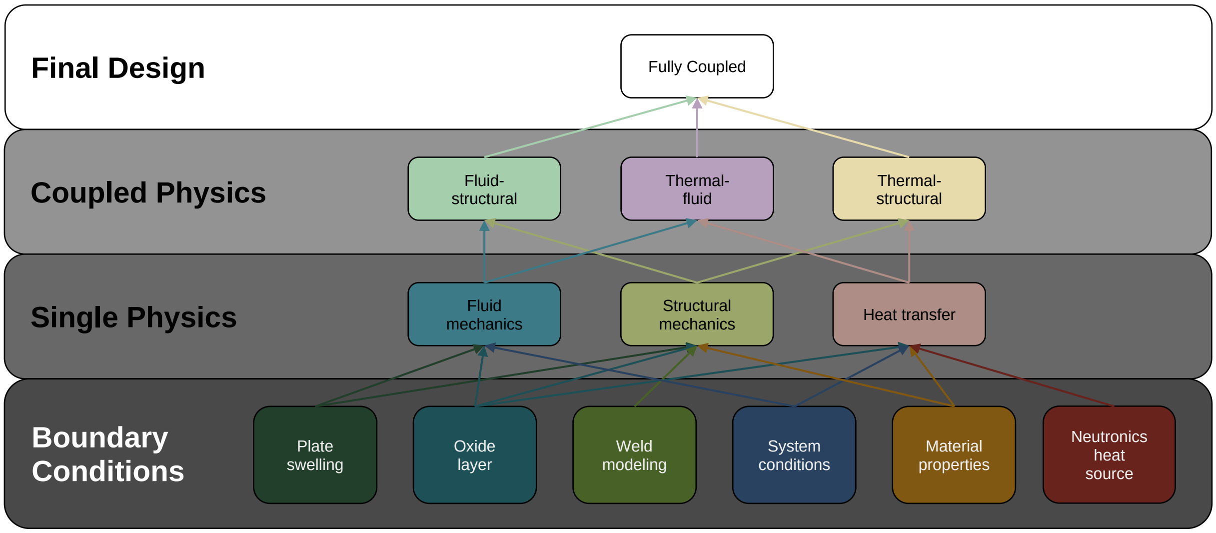

Figure 2.3. Hierarchy of validation requirements.

The boundary conditions are at the bottom of the hierarchy. In some cases, the boundary conditions or inputs are determined outside the current scope of this report. For now, the neutronics and material properties are obtained from outside sources. The boundary conditions are very significant because they are often simplified when the next level is obtained. If it can be shown that certain physics and representations of boundary conditions can be simplified, then the validation for the single physics is not limited by the boundary conditions in completeness or representation.

The next tier includes the single physics validation problems, which separate the individual physics from the fully coupled problems. There are many reasons for validating single physics. First, it determines a representative mesh for the problem and physics, which is useful to prevent over-meshing when a multiphysics problem is attempted. Second, isolating the individual physics ensures that any challenges in validation related solely to the model based on the physics are addressed. Finally, validating single physics provides greater confidence in the fully coupled solution, especially in the absence of complete or representative validation problems.

Moving up the tiers increases the potential for problem completeness. With single physics validation as a base, the hierarchy moves up to the coupled physics problems. For HFIR, the three dominant physics being modeled at this time are fluid mechanics, structural mechanics, and heat transfer. These are coupled together at the coupled physics tier. When physics are coupled together, there is a small potential that the sum of the two physics models will be invalidated by the coupling of the required physics. Various aspects of the software may cause this, but a common reason is that coupling the physics may result in changes to the initial mesh. This is a notable issue with fluid-structure problems in which plate deflections alter the mesh. Another primary reason for coupled physics to be less valid is that properties may change, possibly resulting in the need to decrease time steps or stability parameters. The greatest challenge occurs when all physics must be coupled.

The fully coupled model is above the coupled physics problem. Even though the fully coupled model is at the top of the pyramid, it may have less use. Fully coupled problems are very difficult to validate due to their intricacies. Furthermore, there is a desire to reduce the amount or complexity of the equations being solved. Depending on the information desired, single physics equations can be removed by treating them as boundary conditions in a manner similar to how neutronics is simplified to a heat source term. This approach is valid only when the additional physics do not contribute dynamically or when the contributions are reasonably predicted by a simplified model. For example, a fully coupled system can be modeled as a thermal-structural system by applying a force for the fluid in the structural domain and a heat transfer coefficient for the thermal domain. This approach is only valid insofar as the forces and heat transfer equations are valid.

The number of tiers in the hierarchy may be changed, depending on the application problem, and components can be included, as well as physics. HFIR's geometry and conditions lend the reactor to the hierarchy presented in Figure 2.3, but there is no explicit reason this hierarchy must be followed. It merely provides the framework for which validation problems are chosen.

2.2.2 Validation Problems Solutions

To validate an application problem, a validation solution is required. In this work, the validation solution is the result being used to compare against the model. It is commonplace to use an experiment as the validation solution, as required by most definitions of *validation*. However, in some cases, there is no complete, representative validation experiment with the appropriate level of accuracy. So other solutions—analytic solutions, method of manufactured solutions (MMS), or a code-to-code comparison—can be used. It is important to note that these cases ultimately sacrifice accuracy and certain aspects of representation in exchange for better representation and completeness. Analytical solutions and MMS provide very accurate results, but this is because they draw on the underlying equations that were used to develop the model. Code-to-code comparison will result in the same errors as the code being validated if the same models are used. All of these are typically considered to be verification approaches as opposed to validation approaches, but they have different limitations. Analytic or empirical solutions are typically derived from observations and experiments, so the validation accuracy may be fully assessed if the additional uncertainty is accounted for. Code-to-code comparison can be used to build a complete model. If the code is valid and uses a different solution methodology and mesh, then it provides further confidence in the model being validated. Differences between the two methods can be used to assess a metric-based accuracy, as well. MMS is truly verification because the solution is purely mathematical, but MMS provides solutions that can match expected results and be used to determine what degree of meshing is needed to accurately predict the solution.

Once the various validation solutions are established, the next step is to determine the type of problem that must be solved. The goal is to choose validation problems based on a single HFIR plate and channel. An example of the typical safety basis problem is presented in Figure 2.4.

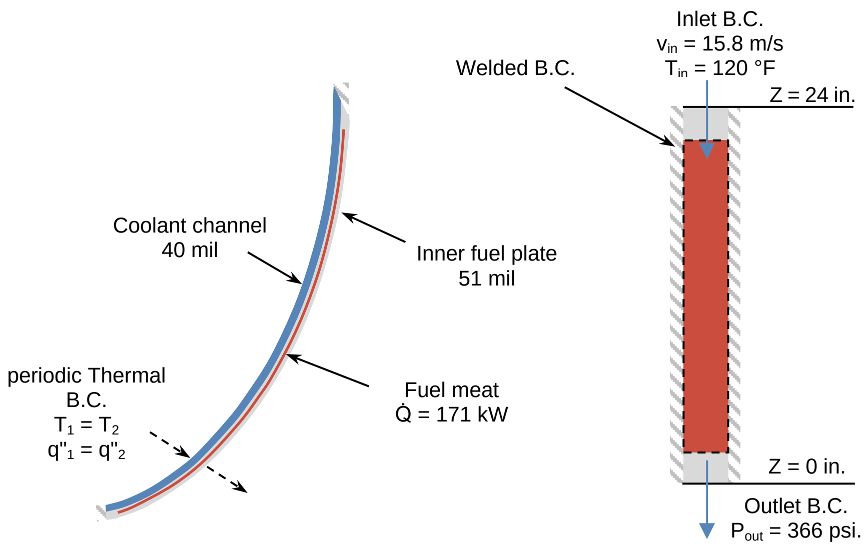

Figure 2.4. Single plate target validation problem.

In Figure 2.4, a single HFIR plate is the target geometry for the validation problem. The goal is to use validation problems that represent the HFIR geometry as best as possible. Completeness is obtained through the hierarchy presented earlier in Figure 2.3. Furthermore, validation experiments should be used when it is reasonable to do so. Unfortunately, the best validation experiments for the geometry were performed in the early days of reactor experiments and had a large degree of uncertainty in parameters. However, they are still used where appropriate.

A list of all of the comparison problems is presented in Table 2.1, and descriptions of the problems are discussed further in Section 3.

Table 2.1. List of validation problems

Section	Physics	Validation comparison
3.1.1	HFIR system	HFIR plant data [1]
3.1.2	Fuel swelling	Advanced Test Reactor (ATR) data [5]
3.1.3	Oxidation	American Nuclear Society (SNS) data [6]
3.1.4	Structural attachment	HFIR plate pull test data
3.2.1	Heat transfer	Analytic solution of heat transfer [1][7]
3.2.2	Fluid mechanics	National Advisory Committee for Aeronautics (NACA) experiments [8]
3.2.3	Structural mechanics	Analytic solution of plate deflection [1]
3.3.1	Fluid-thermal	STAR-CCM+ comparison [1], Gambill & Bundy [9]
3.3.2	Fluid-structural	Smissaert experiments [10]
3.3.3	Thermal-structural	Cheverton-Kelley [11]

2.2.3 Level of Validation

Validation is not necessarily a pass-fail check, as can be the case with verification. Instead, validation is on a spectrum, so whether a code is validated is determined by the parties of interest. However, a suggested approach for the calculation of the validation (V) is to take the inverse of the following:

$$V = 3(A^{-1} + C^{-1} + R^{-1})^{-1} ,$$

where accuracy (A), completeness (C), and representation (R) metrics are combined via inverse addition. This weights the validation toward the lowest values while still giving credence to the validation based on the other values. Alternatively, using the lowest metric value is the weakest link approach, thus ensuring that validation is only as good as the worst aspect of the validation.

The level of validation is at the user's discretion, but the metrics are chosen so that they were on par with one another. A value of 3 is set at the minimum target based on the conditions. Since the criteria are dependent on the application problem, the user decides whether the criteria are satisfied. A full summary of the validation criteria and the expected value are presented in Table 2.2.

Table 2.2. Validation metric

Level	Validation description	Validation acceptance
-	The model is unproven or fails to meet validation standards	Unproven/flawed (unacceptable)
1	The tool has been shown to be capable of predicting the application problem solution, but the validation problem is not valid for the application	The current validation is unacceptable, as it uncertain whether the validation problem matches the application model
2	The model is likely valid for the application, but there is a degree of uncertainty surrounding the validation problem's relationship to the application problem	The model does not meet the requirements of validation, and further validation must be conducted
3	The model is valid for the application, but there are potential uncertainties that should be addressed if the application is extended	The model meets the minimum requirements of validation, and further validation should be considered.
4	The model is valid for the application, but uncertainties indicate that some areas may be further explored to improve the validation	The mode meets the requirements of validation, and further validation may be warranted
5	The validation problem is completely valid for the application problem, and all aspects have been considered	The model has been fully validated for the application problem, and further validation is not needed
+	The model has been compared to and exactly matches the application (validation study)	Over validated (acceptable)

3. VALIDATION PROBLEM EXPLANATIONS

3.1 BOUNDARY CONDITIONS

The first series of problems examined here is the assessment of the validity of boundary conditions used in COMSOL simulations. These problems are implemented to ensure that the boundary conditions used are appropriate for the system. There are three main purposes for validating the boundary conditions: (1) to increase the level of the model's completeness by including additional physics that would otherwise be modeled as simple inputs, (2) to provide evidence that the boundary condition assumptions being used are representative of the true solution, and (3) to determine coefficients for empirical equations for cases in which the boundary conditions are being modeled as boundary conditions. These three purposes are detailed in the following sections.

3.1.1 System Data Input Modeling: Plant Data

To validate system conditions, HFIR plant data were used in conjunction with a system model developed in COMSOL. Plant data could have fallen under the material properties conditions, but the eventual goal is to incorporate the system-level model as boundary conditions into the COMSOL design and safety basis models. This will be useful when the reactor conditions impact the system's state conditions. Consequently, it is necessary to validate this as a boundary condition, as well as the system model. Validation of system boundary conditions is the only validation study that explicitly uses in-core data.

3.1.2 Fuel Swelling

Fuel swelling is an important phenomenon that impacts the structural integrity, geometry, and thermal resistance of the fuel. The fuel swelling experiment was performed at Idaho National Laboratory (INL) using plate L1P04A from the RERTR-9A irradiation campaign at INL's Advanced Test Reactor (ATR) [5]. The test plate was rectangular and was similar in size to the HFIR plate dimensions. In this effort, the greatest concern with representation is the square nature of the plate and the fact that it uses U-Mo as a fuel foil.

3.1.3 Oxide Growth

Oxide growth is typically assumed for most analyses, but there are instances in which the oxide growth may need to be modeled dynamically. There are several challenges associated with oxide layers. Oxide layer growth was determined experimentally and modeled with an empirical correlation. For the design of the Advanced Neutron Source, the data for the oxide growth were calculated with HFIR fluxes for aluminum at ORNL[6]. The experimental data provide a correlation for oxide growth with corresponding experimental data—the biggest concern when modeling oxide growth is the non-negligible uncertainty in oxide layer growth rates. In heat transfer problems, the oxide layer is typically modeled as an additional thermal resistance and is occasionally modeled as a change in the geometry.

3.1.4 Plate Attachment

Plate attachment is one of the more uncertain aspects of the HFIR core. Plate attachment is currently modeled to be in discrete fixed areas. For most analyses, this is sufficient, but under conditions in which full attachment does not exist, a fixed boundary condition is inappropriate. To mitigate this fact, the plate attachment must be modeled using nonlinear plasticity physics. As a validation model, HFIR plate pull test data were used to demonstrate the conditions under which the plate attachment would or would not be valid. The experiments were performed in conjunction with scanning to determine a level of plate attachment. The plate pull tests demonstrate the conditions under which a fixed boundary condition is invalid and when plasticity needs to be modeled.

3.2 SINGLE PHYSICS V&V

The second series of problems is the single physics problems. A series of problems was chosen which best represented the challenges associated with validation of HFIR core physics. In most cases, the greatest challenge was to find the experiments that could validate the extreme conditions and high aspect ratios of the HFIR channels.

3.2.1 Validation of Heat Transfer

Solid heat transfer is well understood, so instead of a validation experiment, MMS and analytic solutions were used to demonstrate that heat transfer for the solid domain would be appropriate. The later studies incorporating multiphysics would use experimental validation solutions. The heat transfer problems used include an applied source with isothermal boundary conditions, a laminar flow boundary condition, and an MMS problem in which the temperature profile was designed to mimic what is expected in the HFIR fuel.

3.2.2 Validation of Fluid Mechanics

For coolant flow modeling, not many problems were available which examined high flow and high aspect ratio channels. Experimental results that fit this criterion were obtained from NACA experiments [8]. However, the shape of the channel was not an involute; the high aspect ratios provided for a reasonable demonstration of which turbulence models and parameters can be used for fluid flow.

3.2.3 Validation of Structural Mechanics

Like heat transfer, the area of structural mechanics is well defined. Both beam and plate deflections have well known solutions. For this reason, the analytic beam deflections matching the length and material properties of the HFIR plate were used. Representation, in this case, is a little more challenging since the plate is not an involute, but this issue is addressed again under thermal-structural validation.

3.3 MULTIPHYSICS

For the multiphysics validation, each of the individual physics was coupled, and the most representative problems were chosen. In these cases, experimental validation for each was the goal because there was a large potential for errors or other inaccuracies to appear. For each case, a nuclear fuel plate geometry was used, and in two of the three validation problems, the validation experiment was explicitly designed to represent the HFIR core.

3.3.1 Fluid Heat Transfer Validation

The fluid-heat transfer has two associated validation problems. The first was a direct code-to-code comparison with Star-CCM+. Since this was not an experimental problem, an additional experiment was found from the Gambill and Bundy [9] fluid heat transfer experiments, which were explicitly designed to determine the heat transfer coefficients for the HFIR fuel plates. Two problems were used to allow for the large degree of uncertainty in the Gambill and Bundy experiment, which makes it a challenging validation problem. To supplement the uncertainty, a code-to-code comparison was included.

3.3.2 Fluid-Structure Validation

Fluid-structure validations are likely the weakest of the three validations. These experiments are difficult to conduct, so there were not any for HFIR with sufficient data to validate. Fluid-structure modeling of the HFIR plates had previously been conducted [12]–[14], and the simulation was validated by Curtis [15] using experimental data from Smissaert [10]. While the experimental plates were not involute, they were still high aspect ratio plates that measured the amount of the plate's deflection. The most significant limitation of this validation is that secondary involute effects on deflection are not fully demonstrated.

3.3.3 Thermal-Structural Validation

The thermal-structural validation is the strongest multiphysics validation. The Cheverton-Kelley experiments [11] were designed to analyze the thermal deflection effects of the HFIR plate during operation; this served as a basis for all of the experimental designs. Jain et al. [16] modeled the Cheverton-Kelley experiments using COMSOL. Like the other studies conducted around this time, the experiments were not designed explicitly as validation studies, and the uncertainty in the study may be greater than is desired for the application.

4. SUMMARY

4.1 VALIDATION OF COMSOL MULTIPHYSICS

This report details the problems selected and the reasons they were chosen to demonstrate the validity of COMSOL for use as a safety basis tool. The validation aspects of the problems are discussed, and validation examples are included. Ultimately, COMSOL is demonstrated as being a valid tool for an array of problems related to HFIR, but validation is dependent upon the specific application.

4.1.1 Summary of Validation Simulations

For each of the physics and boundary conditions involved, COMSOL has been shown to run as intended. While validation is dependent upon the application problem, the general application problems can be surmised based on previous experience and uses of COMSOL. To this end, each problem was examined, and a degree of validation was determined. In all cases, the validation was determined to be sufficient, but the cases with better validation problems provided a better validation value. Estimation of these values is seen in Table 4.1.

Table 4.1: Estimation of COMSOL validation metrics from the presented problems

Section	Physics	Validation estimation*			
		Accuracy	Completeness	Representation	Validation
3.1.1	HFIR system	4	5	5+	4.6
3.1.2	Fuel swelling	5	5	4	4.6
3.1.3	Oxidation	3	5	4	3.8
3.1.4	Structural attachment	5	5	5	5
3.2.1	Heat transfer	4	5	4	4.3
3.2.2	Fluid mechanics	4	5	3	3.8
3.2.3	Structural mechanics	4	5	3	3.8
3.3.1	Fluid thermal	3	5	5	4.1
3.3.2	Fluid structural	3	5	3	3.5
3.3.3	Thermal-structural	4	5	5	4.6

Based on Table 4.1, it can be surmised that the structural attachment problem provided the best validation because it was explicitly designed to assess the weld attachment conditions of the HFIR core. The recent time frame of the experiment, combined with the goal of demonstrating the as-built conditions, indicates that the validation problem was superior. Conversely, the weakest validation problem was that of the fluid-structural problem, which was an early experiment that used very long, flat plates. Therefore, the estimated accuracy and representation were reduced as a result.

4.1.2 COMSOL Validation Database

In addition to the selected problems, COMSOL provides its own validation problems. These validation problems are not detailed because they have a low representation score, but the problems are high accuracy, and they lend additional credence to the use of the individual physics models used by COMSOL.

* These are estimations of the level of validation for expected use in HFIR; these are *not* the actual values, as values must be addressed according to the application problem.

COMSOL’s relevant validation problems are presented in Table 4.2.

Table 4.2. COMSOL validation problems

Problem	Physics
Bracket	Structural mechanics
Steady-state 2D heat	Heat transfer
Axisymmetric transient heat transfer	Heat transfer
Buoyancy flow of free fluids	Fluid-thermal
Flow around an inclined NACA 0012 airfoil	Fluid mechanics
Airflow over an Ahmed body	Fluid mechanics
Unsteady 3D flow past a cylinder	Fluid mechanics
Turbulent flow over a backward-facing step	Fluid mechanics
Non-isothermal laminar flow in a circular tube	Fluid-thermal
Heat conduction in a finite slab	Heat transfer
Non-isothermal turbulent flow over a flat plate	Fluid-thermal
Sheet metal forming	Structural & attachment
Spherical cap with a central point load	Structural

For the most part, the validation problems add confidence to the tool, but they do not necessarily enhance its validity for HFIR-related problems.

4.1.3 Other Evidence in Support of Validation

In addition to the COMSOL simulation database provided, several publications describe work using COMSOL, but these are not detailed in the current iteration of this report. Several studies have demonstrated COMSOL’s successes and shortcomings.

An important consideration that has yet to be addressed is the number of users for a given software. Most nuclear codes have limited user bases due to the nature of the codes and the industry. However, a larger user base makes it more likely that problems will be found for given problems/models in given situations. Once the limitations and issues are found, then the software developer can begin updating the code to address them.

4.2 IMPORTANCE OF THE USER IN THE VALIDITY OF VALIDATION

Even though a code is never 100% validated, it may be valid for a particular application problem. The validity of the code is dependent upon the user. This document provides a method for determining whether a validation study is appropriate for the problem, and it presents problems demonstrating the validity of COMSOL. However, the user of the code must make the determination as to whether the code is valid for a given application.

Theoretically, a code could be fully validated for every application, but for now, it still relies on the user to provide the appropriate inputs, mesh, and physics for the problem. If these things are not done, then even a valid code may produce inaccurate results. The solution must be verified for any application. Validation merely demonstrates the ability of the code to accurately solve a given application problem.

4.3 POTENTIAL AREAS FOR IMPROVEMENT

The greatest limitations of validating COMSOL for HFIR lie not in the code itself but, instead, in the availability of full validation quality experiments. The bulk of experiments best representing HFIR are not great validation experiments. They are likely at the very limit of what can be deemed as acceptable validation problems. To improve validation efforts, better experiments should be sought or performed to improve the validation domain of COMSOL or any other tool. A good validation experiment details the experiment as much as possible and provides an extensive amount of data to help build a good validation model.

If all the validation solutions were improved, then there would still be limits in the scope of the validation performed. The problems were chosen to cover the most validation for HFIR application problems. One notable blind spot is that of fluid-structure vibration. There are very few experiments in the literature for the vibration of high-aspect-ratio plates in a fluid. Vibration characteristics are not at the forefront of design and safety basis problems, but they are a notable blind spot for which a valid tool would be useful for analysis.

As stated above, validation is never 100% complete, but instead, it falls within a spectrum. One area that can be significantly improved is the process of solution verification and validation. An improvement to the solution validation efforts would be to use a standard for verifying the solution, followed by an independent review process. The method for doing this is arbitrary, as long as it provides a means to demonstrate grid independence.

It should be noted that *validation verification* is a new term. Various validation metrics are provided throughout the report, but this still requires an application problem, and the user must determine whether the validation problem appropriately demonstrates validation of the code for the application problem. To this end, it is necessary to know where simplifying assumptions can be made and where those assumptions will invalidate a validation study. To verify that the validation metrics are appropriately chosen, an improvement would be to objectively determine the impacts through scaling and estimation or to conduct a sensitivity study prior to simulating the application problem.

5. REFERENCES

[1] T. K. Howard, P. K. Jain, E. L. Popov, M. W. Crowell, and F. G. Curtis, *Verification and Validation of COMSOL Multiphysics for HFIR*, unpublished, 2020.

[2] ASME, "Guide for Verification and Validation in Computational Solid Mechanics," *Am. Soc. Mech. Eng.*, 2006, doi: 10.1007/s00366-007-0072-z.

[3] ASME PTC Committee, "Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer: ASME V&V 20," *Am. Soc. Mech. Eng.*, 2009, doi: 10.1016/S0160-4120(97)89943-6.

[4] ASME, "Assessing Credibility of Computational Modeling and Simulation Results through Verification and Validation: Application to Medical Devices," *ASME V&V 40-2018*, 2018.

[5] Y. S. Kim, G. L. Hofman, J. S. Cheon, A. B. Robinson, and D. M. Wachs, "Fission induced Swelling and Creep of U-Mo Alloy Fuel," *J. Nucl. Mater.*, 2013, doi: 10.1016/j.jnucmat.2013.01.346.

[6] R. E. Pawel, G. L. Yoder, C. D. West, and B. H. Montgomery, *The Development of a Preliminary Correlation of Data on Oxide Growth on 6061 Aluminum under ANS Thermal-Hydraulic Conditions*, ORNL/TM-11517, 1990.

[7] T. K. Howard, P. Jain, and E. Popov, "Verification and Validation of COMSOL for Heat Transfer in Thin Rectangular Channels Using NACA Test Results [1]," in *18th International Topical Meeting on Nuclear Reactor Thermal Hydraulics, NURETH 2019*, 2019.

[8] J. Laufer, *Investigation of Turbulent Flow in a Two-Dimensional Channel*, National Advisory Committee for Aeronautics, NACH TR 1053, 1950.

[9] W. R. Gambill and R. D. Bundy, *HFIR Heat-Transfer Studies of Turbulent Water Flow in Thin Rectangular Channels*, ORNL 3079, 1961.

[10] G. E. Smissaert, "Static and Dynamic Hydroelastic Instabilities in MTR-Type Fuel Elements, Part I. Introduction and Experimental Investigation," *Nucl. Eng. Des.*, 1968, doi: 10.1016/0029-5493(68)90103-9.

[11] R. D. Cheverton and W. H. Kelley, *Experimental Investigation of HFIR Fuel Plate Deflections Induced by Temperature and Pressure Differentials*, ORNL TM 2325, 1968.

[12] F. G. Curtis, K. Ekici, and J. D. Freels, "Fluid-Structure Interaction Modeling of High-Aspect Ratio Nuclear Fuel Plates Using COMSOL," *Proc. 2013 COMSOL Conf.*, 2013.

[13] F. G. Curtis, J. D. Freels, and K. Ekici, "Predicting Large Deflections of Multiplate Fuel Elements Using a Monolithic FSI Approach," *Nucl. Sci. Eng.*, 2018, doi: 10.1080/00295639.2017.1379304.

[14] F. G. Curtis, J. D. Freels, and K. Ekici, "Deflection Predictions of Involute-Shaped Fuel Plates Using a Fully-Coupled Numerical Approach," *Ann. Nucl. Energy*, 2019, doi: 10.1016/j.anucene.2019.02.001.

[15] F. G. Curtis, *Fluid-Structure Interaction of Involute Fuel Plates in the High Flux Isotope Reactor Using a Fully-Coupled Numerical Approach*," University of Tennessee Knoxville, 2018.

[16] P. K. Jain, J. D. Freels, and D. H. Cook, *3D COMSOL Simulations for Thermal Deflection of HFIR Fuel Plate in the* "*Cheverton-Kelley*" *Experiments*, ORNL/TM-2012/138, 2012.