Argonne Reactor Code Software Verification and Validation Plan for the Versatile Test Reactor

Nuclear Science and Engineering Division

About Argonne National Laboratory

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

DOCUMENT AVAILABILITY

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

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

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

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

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

Disclaimer

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

Argonne Reactor Code Software Verification and Validation Plan for the Versatile Test Reactor

M. A. Smith, G. Aliberti, and F. Heidet
Nuclear Science and Engineering Division
Argonne National Laboratory

December 15, 2018

ABSTRACT

This Argonne Reactor Code (ARC) software has taken a pivotal role in the design work of the Versatile Test Reactor. The ARC software itself consists of several independent or semi-independent pieces of software that are used in concert to carry out specific analysis and design objectives. In this report, we discuss the validation that is needed for the ARC software overall and discuss available, usable experimental data and how it relates to the VTR project. We also discuss the verification that is required for the ARC software with respect to its usage in the VTR project and give some concrete examples as to what it must accomplish. A list of goals for the verification and validation work are presented as the general procedure that will be implemented for every individual piece of software in ARC.

The ARC software has been continuously developed for over 50 years and many of the features used for the VTR project have thus been rigorously tested for accuracy during that timeframe. The ARC software is presently under source code control and undergoes nightly regression testing of not only the individual pieces, but the interconnected aspects that are common to most analysis projects. Its greatest weakness, from a software verification and validation perspective, is both a lack of documentation and outdated documentation. As an example, much of the previous validation work will require contemporary effort to re-evaluate given the updated base data libraries and modifications to the software that have occurred since. This requires new documentation to be produced to support the usage of the software in the VTR project.

The bulk of the ARC software is focused on steady state neutronics and TH behavior of a reactor system. In this regard, we have identified the ZPPR, FFTF, and EBR-II programs as the best candidates for sources of validation data for the VTR project. The ZPPR-15 series in the ZPPR program was chosen as it is the best representation of the VTR fuel form and reactor core layout of all of the ZPPR measurements. Additional support for it comes from the fact that most of the data has already been processed into usable ARC models for VTR validation. The FFTF and EBR-II reactors were both functional reactors that provide reactor scale measurements. FFTF is clearly a closer match to VTR and its startup measurements would be a very valuable validation basis. EBR-II has invaluable burnup related measurements that will play a key role in the ARC software validation work.

Each piece of ARC software will require a report detailing the software verification. The validation work will be logically split into the results of modeling each identified experimental resource (ZPPR, EBR-II, FFTF). The intended timeline for the VTR project is to have the ARC software verified and validated by the end of fiscal year 2020. This is incumbent on sufficient funding allocated to V&V activities and no change in programmatic priorities.

TABLE OF CONTENTS

ABSTRACT	ii
1 Introduction	1
2 Verification and Validation Plan Structure	2
3 Software Verification	4
4 Software Validation	5
4.1 Identification of the ZPPR-15 series of experiments	5
4.2 Identification of the EBR-II validation data.....	6
4.3 Identification of the FFTF validation data	7
4.4 Identification of Reactivity Coefficients to be Validated	8
5 Representativity Methodology	9
6 Verification and Validation Problem Setup and Execution.....	10
Appendix A. ARC Software Summary	10
References	12

LIST OF FIGURES

Figure 1. ARC Software Connections for Fast Spectrum Reactor Analysis.....	1
Figure 2. Stationary Half Loading Diagrams of Four ZPPR-15 Phases	6

LIST OF TABLES

Table 1. Example Engineering Outputs and Software Verification Methodology 3

Table 2. VTR Reactivity Coefficients and Methodology behind Them 9

Table 3. List of Components in the ARC Software Package and Their Purpose..... 12

1 Introduction

This document discusses the intended software verification and validation plan for the Argonne Reactor Code (ARC) software package [1-13] as part of the Versatile Test Reactor (VTR) program. The ARC software package consists of steady state neutronics and thermal hydraulics modeling capabilities which are being used by the VTR program to develop most of the VTR reactor design details which will be part of the licensing application. In Figure 1, the connection to the SAS4A code, which is not an ARC software component, is shown since SAS4A is used to perform the safety analyses for the VTR. Figure 1 shows the current ARC software components (orange boxes) and the connections between them when used for fast reactor analysis or design projects like VTR. Most of the ARC software in Figure 1 is used in the VTR project with the exception of NUBOW, VARI3D, and RCT but it should be clear that ARC is a complex set of connected software.

Figure 1. ARC Software Connections for Fast Spectrum Reactor Analysis

In this document, we focus on verification and validation needs associated with the ARC modeling software that is used in the VTR design activities and thus will be an integral part of the VTR safety analysis report. While the ARC software does have some dynamics capability, the

assessment of the dynamics behavior (*i.e.* transients) falls under the safety analysis component (SAS4A) of the VTR program and thus those aspects are not covered in this document.

With unlimited time and resources, DOE would carry out zero-power mockup experiments for the VTR to establish validation cases and reduce the design uncertainties. Given current lack of such experimental capabilities, this would significantly delay the present VTR project timeline. As an alternative, we can rely upon previous experiments and operational reactor data to meet our validation needs. In this regard, we have a substantial database of experimental measurements on zero power critical mockups, ZPPR [18] along with historical data from the operating fast spectrum reactors EBR-II [19] and FFTF [20]. While none of these systems are perfectly identical to the proposed VTR, they provide sufficient basis to rigorously test the ARC software. In this report, we also detail the representativity methodology which will be used to qualitatively assess how well a particular measurement in the experiment will compare to a similar measurement in the VTR.

The Zero Power Plutonium Reactor experiments carried out at Argonne National Laboratory (ANL) during the 1980s for the ZPPR-15 program have been identified as a good starting point for the VTR validation efforts. This choice accounts for several factors: 1) the availability and quality of the data, 2) existing staff that are already familiar with the experimental machine and measurements, 3) most of the ZPPR-15 loadings of interest have already been processed into ARC models, and 4) a full uncertainty quantification has already been done for several loadings of ZPPR-15. The FFTF startup physics experiments are equally ideal validation data but only a limited amount of information is presently available to us. As part of the VTR project, we will obtain the data for and build FFTF models to supplement the validation data available in ZPPR-15. The amount of FFTF startup physics experiment data recovered will be based upon the VTR project timeline and the validation needs not met with ZPPR-15. The EBR-II reactor is less ideal than the previous two systems, but the availability of measured fuel depletion data makes it an invaluable part of the validation effort. The EBR-II data is already processed, ARC models are constructed and can be executed today. It is important to note that the startup tests of the VTR itself will serve as the ultimate validation basis of the ARC software as they become available.

2 Verification and Validation Plan Structure

Software verification and validation begins by identifying the key features or capabilities of the software that are being used in the design process. In this regard, we must identify the software engineering outputs that are used to establish the safe and economical operation of the VTR (*i.e.* feeding into the Preliminary/Final Safety Analysis Report). From there, software verification will require testing the methodology used in the software to ensure that the outputs it produces are consistent with the provided user inputs, and their stated meaning and use in the VTR project. As an example, the reported peak fast fluence is used to constrain the operational lifetime of each VTR fuel assembly. The software verification process will require that a test problem be created (consistent with, but not explicitly a VTR model) and used to show that the software produces the correct fast fluence in the output, given correct inputs. In this regard, the test problem poses a solvable mathematical problem with a known reference solution (obtained by either numerical solution or a hand calculation) thereby avoiding the uncertainties associated with physical experiments.

Table 1 shows an example list of such engineering outputs that need software verification. We also identify that a reference solution is needed for each test problem to complete the proof of accuracy. While we suggest the MCNP software in Table 1, we are not limited to that software to establish a reference solution and can use other software as needed and appropriate. We, of course favor existing software that have been previously verified.

Table 1. Example Engineering Outputs and Software Verification Methodology

Engineering Output	Testing Methodology	Reference Solution
Criticality	Several semi-analytic benchmark problems (small to large) that demonstrate the accuracy of the output	MCNP is used in multi-group mode to obtain reference solutions to non-trivial problems
Reaction Rate Distribution		
Discharge Fuel Burnup		
Discharge Peak Fast Fluence		
Control Rod Worth		
Reactivity Coefficients used in Safety Analysis		

Software validation consists in the assessment of the predictive accuracy of the software for the VTR, based on its accuracy for real experimental measurements. As an example, if we have an experimental measurement of reactivity worth in FFTF, the ARC software might have a bias in the prediction of that worth, which can be determined through uncertainty quantification. By using the representativity methodology, we can estimate what kind of bias will be present in the VTR reactor for a similar measurement. In this particular example, we can also make use of other neutronics software packages to further assess the altered bias in the VTR reactor. Overall, the purpose of the validation work is to help assess the predictive accuracy of the software on the VTR and, by extension, help assess the initial uncertainties on operational characteristics.

With the preceding descriptions, we can envision the verification and validation plan to consist of the following high-level steps:

- 1) Identify a piece of software being used in the design process of the VTR.
- 2) Verify that the documentation (manual, user guide) is up to date on the software.
- 3) Identify the key inputs and outputs for that software being used by the VTR project.
- 4) Create or identify verification problems that can demonstrate each identified output is consistent with the stated definition in the software manual and with the way the VTR project is using that output.
- 5) Identify validation problems based upon measured experiments that can be used to assess the predictive accuracy of the software output for the VTR project.

- 6) Setup and document the analysis effort and software result for each validation case. Perform a representativity calculation and/or code-to-code comparison as needed to assess the reliability of translating the computed bias from the experiments to the VTR.

3 Software Verification

As stated, software verification is primarily focused on proving that the software produces the stated output consistent with the embedded methodology. For VTR, we will approach software verification in the modern way. To begin, the VTR core design team have to identify the key software components, features and capabilities that are being used in the design process. This identification process will define which input and output options from each piece of ARC software are important and thus need to be checked and verified for implementation accuracy.

The ARC software has been continuously developed for over 50 years, exclusively by ANL and thus most of the software components, features and capabilities have been rigorously tested for accuracy in the past. Currently, the software is part of an automatic regression system that checks the integrity of the software, compilers and operating system. The verification process has evolved over time to require multiple test problems be executed for each piece of ARC software which check most of the capabilities. The existing ARC software verification test suites were created to detect problems with the software as it progresses through time and computing technology. As the software functionalities were updated and bugs resolved, additional test problems were added. Although this was done over a 50 year period, due to the funding being virtually non-existent over the last 25 years, documentation detailing this work is scattered and lacking.

Upon review, the lack of documentation and an associated reference output makes many of the existing test cases unacceptable for the VTR project as the existing solution produced by the code has no basis in accuracy other than it is consistent with the previous result. As part of the VTR project, most of the verification work will have to be redone. When possible, previous code-to-code comparison documents and reference solutions will be used. In some cases, alternate software (*i.e.* MCNP) will have to be used to establish an updated reference solution. As necessary, all of the engineering outputs will be checked to ensure they produce the stated results and all of this work will be documented appropriately.

As part of the verification requirements, the manuals must consistently state what the software currently produces. For REBUS and SUPERENERGY, the existing manuals are inaccurate in their description of the current input and output as the software was continuously updated for the last 40 years but the manuals are dated to the early 1980s. As part of the VTR project, we will have to update the manuals of many ARC components along with the verification test suite.

From Figure 1, we can already identify the ARC software components that will need a software verification plan. Given a defined list of inputs and outputs to check for each piece of software, we will begin by updating the manuals, and defining test problems and reference solutions that prove the accuracy of the implemented methodology for each feature and capability. We expect to add new tests to the existing verification test suite and to replace obsolete verification tests as necessary.

The outcome of this work will be a formal document describing the software verification status of each ARC software component used by VTR which will be consistently updated as the software progresses in the VTR project or in the rare cases where the source code has to be modified.

4 Software Validation

The primary purpose of validation work for VTR on the ARC software is to assess the software accuracy on predicting the core criticality state and the power and temperature distribution at all points in time during a prescribed fuel cycle operation. The accuracy must also be assessed for the software's ability to calculate the transmutation of the fuel under the proposed fuel cycle operation (fuel burnup) and compute several quantities relevant for materials irradiation limits. For transient safety analysis, the validation work must demonstrate the software can predict the behavior of the system due to small compositional and geometrical perturbations at each time point.

As discussed in Section 1, the ARC software validation work will primarily focus on using data from experiments which have existed and share commonalities with the VTR design. This relies on the assumption that the VTR core design is defined to a sufficient level of details allowing to identify representative systems. Based on the reference VTR core design [18], systems that could be used to perform the ARC software validation have been identified. Currently, only US-based systems are being considered to ensure we can more easily access the required data, and these systems are introduced in the following three sub-sections.

4.1 Identification of the ZPPR-15 series of experiments

In typical operating fast spectrum reactor plants, the reactivity feedbacks that dictate the stability of the reactor during transients are typically small. Given this, having highly accurate measurements of small perturbations and being able to predict them with the modeling software is key to the accuracy of the software on modeling the transients. The ZPPR-15 set of experiments are the best U.S. example we have of a critical mockup of the VTR reactor with clean experimental data on such small reactivity measurements. The ZPPR-15 data is presently maintained by both INL and ANL in both paper and electronic forms. All of the ZPPR-15 loading details and much of the experimental measurements have already been processed by ANL staff and are available as ARC executable benchmark problems.

The ZPPR-15 experiments were conducted from April 1985 through July 1986 under the Integral Fast Reactor (IFR) Benchmark Physics Test Program [14]. The focus of the IFR program work was to provide experimental support for core designs using metallic alloy fuel. The basis for the critical assembly was a 330 MWe sodium-cooled fast reactor. To simulate a reactor, the ZPPR experimenters filled drawers with plates to create a desired core composition and then loaded those drawers into an array of matrix tubes. The system could become critical when two adjacent arrays of matrix tubes were moved together. The ZPPR-15 experiments included diverse reactor physics measurements such as criticality, void worth, region-wise spectral indices, reaction rates, control rod worth, and several reactivity coefficients.

The ZPPR-15 experiments were performed in four phases, referred to as "phase A" through "phase D". Each phase was marked by a particular composition of the reference assembly, with the last three being representations of the anticipated IFR fuel cycle. Phase A was a clean physics assembly containing only plutonium, depleted uranium, stainless steel and sodium without zirconium.

Phases B through D were all designed to simulate zirconium alloyed metal fuel. Phase B focused on Pu based fuel, while phase C was mixed Pu and U fuel and phase D was primarily uranium based fuel. For the VTR work, measurements obtained from all four phases are relevant. Figure 2 shows an illustrative drawer loading diagram for each of the four phases of ZPPR-15, where each loading corresponds to specific drawer loading details referenced by experimenters in their notebooks.

Many of the ZPPR-15 experiments were previously analyzed for a variety of projects. The ZPPR-15 experiments were reconstructed into MCNP models. These models are the most geometrically accurate representation of the experiment that can be constructed today given the information available on the experiment. The MCNP code generally provides the best answer for core criticality but MCNP can be very inefficient in producing accurate answers for some of the reactor physics measurements in ZPPR. Consequently, as part of the previous work, deterministic models using the ARC software were developed for many of the ZPPR-15 experiments.

Figure 2. Stationary Half Loading Diagrams of Four ZPPR-15 Phases

4.2 Identification of the EBR-II validation data

The EBR-II reactor was a sodium-cooled fast reactor that started operation in 1964. The intention of the reactor was to demonstrate reprocessing and power plant operations on the same site. The initial physics startup tests for EBR-II are lost, however, in the last phase of its life, very important

and valuable experiments were performed which provide very good validation cases. These included passive safety related testing and fuel cycle depletion data, the data for which has been processed into existing ARC and SAS4A models. Of particular interest for the ARC software are the fuel cycle depletion cases which can be used as excellent validation problems for the VTR.

The Physics Analytical Database (PADB) was first created as part of the Integral Fast Reactor (IFR) fuel cycle demonstration program in EBR-II and its adjoining Fuel Cycle Facility (FCF). The PADB contained all the required data (measured and calculated) to model the metallic fuel (Zr-alloyed Uranium is the target data in this work) loaded into the Experimental Breeder Reactor II (EBR-II), the fuel irradiation during the reactor operation, and the fuel discharged from EBR-II.

The PADB was created and the data was collected before 1994, after which EBR-II was permanently shut down. With the exception of a small amount of data at the end of the operational period, a considerable amount of fuel cycle related data was stored in the PADB. The data in the PADB was later processed by staff in four different divisions at Argonne and stored in the file systems of each division according to the type of data of interest. In 2014, the PADB system data was recovered at ANL for use as validation data for the ARC software. The PADB contains the EBR-II core loading patterns, subassembly dimensions, control rod positions, and other data necessary to reconstruct the EBR-II operational details for the IFR part of its operational lifetime.

The fuel cycle measurements of interest are based upon Isotope Dilution Mass Spectroscopic (IDMS) analysis of the ^{139}La and ^{148}Nd fission products. The measurement uncertainty is $\pm 4\%$ due primarily to the way the fission products can diffuse during operation and the measurement accuracy itself. There is an additional uncertainty on the power level of EBR-II of 4% and thus historically we have assigned a 6% uncertainty on the burnup predictions, which for 20% fuel burnup, are very good. These measurements were historically stored in the chemistry division component of the PADB but previous work by ANL staff has already coalesced this data into a usable set of test cases.

4.3 Identification of the FFTF validation data

The Fast Flux Test Facility (FFTF) reactor is located on the Hanford site in eastern Washington state and was constructed in 1978 as part of the breeder reactor program. Its first criticality was achieved in 1980 with full power operations from 1982 to 1993. Most of the startup test data is reported to be available although not in a processed form. Although FFTF was designed to use oxide fuel, the startup physics tests are still invaluable as they can further provide validation of criticality at different operating temperatures for an operating power reactor along with control rod and other whole core reactivity measurements. At present, the accuracy on any measured data is unclear and thus the first step should be to identify the physics measurements that are inadequately specified by ZPPR-15 and EBR-II. From there, an assessment of the measurement uncertainty for a proposed FFTF data source must be identified before any work should be done to try and reconstruct the geometrical loading details.

As an example of foreseeable troubles with FFTF data, the measurement error is anticipated to be rather small yet the real uncertainty will come from the accuracy of the geometric and material information. Consistent with other real reactors, this data is expected to be far worse due to the higher fabrication uncertainties and potential loading uncertainties than that observed in base experimental machines like ZPPR. In addition to this basic level of uncertainty, the startup tests at

operating temperatures will involve approximate modeling of the core restraint system impact on the reactor. Such measurement details are notably non-existent in the literature and although they will result in some additional uncertainty in the models we construct, they can provide valuable insight into the error that can result from translating ARC software performance on ZPPR to a real reactor like EBR-II, FFTF, or VTR.

4.4 Identification of Reactivity Coefficients to be Validated

For the safety analysis of fast reactor systems with metal fuel, Table 2 shows the list of reactivity coefficients typically used. The same list is anticipated to be applicable to the safety analysis for the VTR. Most of these reactivity coefficients are not easy to evaluate experimentally in an operating reactor, in particular the sodium density, fuel density, and structure density coefficients. These are next to impossible to assess in ZPPR because they involve the core-wide physical insertion or removal of materials which was not possible in ZPPR due to the limited inventory of plates. The radial expansion coefficient (related to the core restraint system) is also difficult to assess in ZPPR as it requires a change in the geometry (assembly pitch) that is not possible.

The kinetics parameters Λ and β would seem to be straightforward, but only β and its coalesced representation was measured in ZPPR. While these measurements were standard for every loading, the reporting of the data was not done and we are fortunate that we have reported measurements of β for ZPPR-15.

From Table 2, an additional aspect worth discussion is the first order perturbation methodology. The first order approximation is a mathematical approximation to exact perturbation theory and is only valid for “small” perturbations. All experimental measurements are exact perturbation measurements. In this regard, we cannot “validate” a first order approximation to the exact perturbation, but can only verify that it is consistent with the exact perturbation for small reactivity worths. This will be handled in the software verification stage. In this regard, most of the validation work with the reactivity coefficients is proving that the modeling methodology and input data is consistent with the measurement and by extension the proposed behavior of the VTR system.

Table 2. VTR Reactivity Coefficients and Methodology behind Them

Reactivity Coefficient	Perturbation type	Methodology
Doppler	First Order	Temperature of fuel and alloying isotopes are all increased substantially in the perturbed state
Sodium Density	First Order	The density of the flowing sodium inside the assembly duct is changed by ~10%
Fuel Density	First Order	The density of the fuel and alloying isotopes are all changed by ~10%
Structure Density	First Order	The density of the duct, cladding, and wire wrap are all changed by ~10%
Radial Expansion	Exact	The geometry is uniformly increased in the radial direction and the compositions are all adjusted to preserve mass of structure and fuel isotopes. Sodium backfill is assumed
Unswelled and Swelled Fuel Axial Expansion	Exact	The unswelled case assumes the fuel geometrically expands and pushes some bond sodium into the plenum region. The swelled case assumes the cladding is also forcibly expanded with the fuel.
Control rod worth	Exact	The “S curve” control rod worth is computed for a given loading.
Λ (prompt neutron lifetime)	N/A	Standard procedure
β (delay neutron fraction)	N/A	Standard procedure

5 Representativity Methodology

Integral experiments play an essential role in the reduction of design uncertainties related to reactor neutronics calculations. The purpose of any validation problem is to use integral experiment measurements to provide an estimate of the modeling error associated with each quantity of interest for a given piece of software. However, the experiments used for validation do not constitute an exact mockup of the VTR reactor itself. In this context, a “representativity” calculation [16] can give a quantitative and synthetic measure by which to judge the relevance of the chosen experiment and the desired validation use in VTR.

As an example, modeling control rod worths in ZPPR-15 will provide a modeling bias for control rod worth in the ZPPR-15 based upon the modeling approach used and cross section data. We can further determine an uncertainty with that bias by identifying all of the measurement and geometry uncertainties. Our desire is to use that information to determine the bias in the control rod worth for VTR for which we will use the representativity methodology to quantitatively provide an estimate of comparison. Translating the uncertainty will require including assumptions about fabrication details on the VTR and thus we know that the uncertainty on that bias will be higher than that observed in ZPPR, but tractable given the uncertainty assessment done in ZPPR.

When possible, MCNP calculations [15] will be used to provide an alternative comparison point to the ARC software result to further assess the inherent modeling errors associated with the approximations made in the ARC software and inexact nuclear data. The measurement uncertainty of the entire experiment will be evaluated for a select critical configuration of each reactor system to get a sense of how reliable each validation problem is. Due to the nature of available experiments used for validation, we need to be careful when using a local measurement as a direct validation of a characteristic across the entire VTR core. If we do, this will show up as a fundamental bias in the representativity calculation when we attempt to compare the similarity of the two different phenomena in the two different machines. Thus we will have to work hard to carry out our validation efforts to isolate the mismatch of the physical experiment performed from the desired VTR characteristic.

6 Verification and Validation Problem Setup and Execution

The initial execution of the V&V work for the ARC software will produce a documented assessment of the predictive accuracy of the software for the VTR project. As is common with software, the desire to add additional functionality or update the software will necessitate the update of the V&V documentation. Any other issue such as bugs or modeling mistakes in the V&V test cases will require additional updates to the V&V documentation. Consequently, all of the previously stated verification and validation problems will be put in the source code control repository and will be tracked alongside with the software. This will be done in accordance of the VTR SQA plan aiming at meeting NQA-1 requirements [17].

From Figure 1, for each identified validation case we will need to execute a sequence of ARC software such as MC²-3, REBUS, and PERSENT, along with some intermediate utility programs. The complete output from each code will be tracked using an automated checking procedure and pre-determined tolerances in order to identify any errors that arise due to changes to either the software, the ARC software models or the computer configuration. To setup this infrastructure for all of the validation cases of ZPPR-15, EBR-II and FFTF is a significant undertaking, but will make the V&V process more practical to maintain going forward. All additional validation cases for the ARC software that will be identified with time will be setup similarly.

Appendix A. ARC Software Summary

The neutronics component of the ARC software [1-13] is presently based upon ENDF/B-VII.0 and uses a multi-group formalism. The ETOE software [2] is used to process the ENDF/B data and provide the cross section data in an alternative representation for use in the MC²-3 software [2]. MC²-3 generally is applied to a much simplified geometric representation of the reactor system being modeled. The MC²-3 software produces multi-group cross sections for both neutron and gamma transport along with delay neutron data. For ARC, the MC²-3 produces multi-group cross sections specifically targeting a homogenized assembly equivalence theory model that allows the follow-on software to accurately predict the average reaction rates of the homogenized assembly with typically small errors on the minimum and maximum reaction rates within each assembly as discussed in the MC²-3 manual.

The multi-group neutron cross section library is provided to either the DIF3D software [3-6] for steady-state eigenvalue based neutronics calculations or the REBUS software [7] component for

fuel cycle analysis work on a given reactor operating under a specific fuel management strategy. The REBUS software uses the DIF3D software at each time step of the fuel cycle to obtain the steady state neutron flux solution and has a very specific methodology about how the flux solution changes over the time step to produce the correct fuel transmutation. The REBUS software provides a snapshot of the reactor state at each time point which can be used to define the input for GAMSOR software [8]. GAMSOR is a wrapper around the DIF3D software to allow the user to compute the coupled neutron-gamma transport calculation and produce both neutron and gamma heating throughout the modeled reactor system. The GAMSOR output is passed to the steady-state thermal hydraulics SUPERENERGY software which is a derivative product of reference 9 (referred to historically as SE2ANL and more recently SUPERENERGY-ANL). SUPERENERGY has built-in assumptions about where the neutron and gamma heating is deposited based upon experience with ducted assembly based fast reactors.

The reactor details at each time point in REBUS can be extracted and passed to the perturbation and sensitivity software PERSENT [10]. In PERSENT, the DIF3D code is used to obtain the eigenvalue change between different reactor configurations (reactivity coefficients). PERSENT is primarily used to compute reactivity coefficients for use in the safety analysis calculations of SAS4A. It additionally processes the delay neutron data and computes the kinetic parameters Λ and β . The sensitivity functionality in PERSENT includes an uncertainty quantification and representativity calculation capability. These functionalities are wrapped around the reactivity coefficient or kinetic parameter calculations that PERSENT can perform and are deemed essential to assess the quality of each validation problem for the ARC software in the VTR project.

The following additional ARC software will be used for some of the validation work:

- Most of the fuel burnup measurements usable as validation cases were done on individual pins of unique assemblies and we will require the RCT software [11] to carry out that validation work. The REBUS code is primarily built around a homogenized fuel assembly model while the RCT code is an auxiliary fuel cycle analysis code that, with REBUS, can be used to track pin-by-pin depletion details of fuel assemblies in a given reactor model.
- The DPT software [12] may be used by the VTR project. DPT is to generate sensitivity coefficients for the fuel cycle scheme used in REBUS. Those sensitivity coefficients can be used in a representatively calculation in PERSENT to estimate how well the experimentally measured burnup matches an equivalent calculation in the VTR reactor itself. Because of its present status, we may resort to alternative approaches to obtain the sensitivity coefficients.
- The NUBOW-3D software [13] may be used in designing the core constraint system of the VTR. In NUBOW-3D, the user provides the core constraint system details (dimensions and geometry) and each assembly's irradiation and temperature distribution history along with reactivity coefficient distributions. For each steady state time point in a fuel cycle analysis, NUBOW-3D estimates the change in reactivity feedback based upon the deformed core configuration resulting from prescribed power ramps for the core restraint accounting for creep and swelling.

Table 3 summarizes the ARC software components and their purpose. In each case, the documentation will need to be updated to address missing details on those features used by the VTR project.

Table 3. List of Components in the ARC Software Package and Their Purpose

Software Name	Purpose	Software Name	Purpose
MC ² -3	Generate an Equivalence Theory Based Broad Group Multigroup Cross Section Library	ETOE	Process ENDF/B data into MC ² -3 libraries
DIF3D	Steady state neutron/gamma Flux Distribution	GAMSOR	Coupled neutron-gamma Flux Solution
REBUS	Fuel Cycle Analysis	RCT	Fuel Pin Level Fuel Cycle Analysis Capability
SUPERENERGY (SE2ANL)	Steady State Thermal-hydraulics Capability	NUBOW	Thermal Mechanical Behavior of Irradiated System
PERSENT	Reactivity Coefficient and Sensitivity Analysis	DPT	Sensitivity Analysis (UQ) for Fuel Cycle Analysis

References

1. L. C. Just, H. Henryson II, A. S. Kennedy, S. D. Sparck, B. J. Toppel and P. M. Walker, "The System Aspects and Interface Data Sets of the Argonne Reactor Computation (ARC) System," ANL-7711, Argonne National Laboratory (1971).
2. C. H. Lee and W. S. Yang, "MC²-3: Multigroup Cross Section Generation Code for Fast Reactor Analysis," ANL/NE-11-41 Rev. 1, Argonne National Laboratory, January, 2012.
3. K. L. Derstine, "DIF3D: A Code to Solve One-, Two-, and Three-Dimensional Finite-Difference Diffusion Theory Problems," ANL-82-64, Argonne National Laboratory (1984).
4. R. D. Lawrence, "The DIF3D Nodal Neutronics Option for Two- and Three-Dimensional Diffusion Theory Calculations in Hexagonal Geometry," ANL-83-1, Argonne National Laboratory, March 1983.
5. Palmiotti, G., Lewis, E. E., Carrico, C. B., "VARIANT: VARIational Anisotropic Nodal Transport for Multidimensional Cartesian and Hexagonal Geometry Calculation," Argonne National Laboratory ANL-95/40, 1995.
6. M. A. Smith, E. E. Lewis, E. R. Shemon, "DIF3D-VARIANT 11.0, A Decade of Updates," ANL/NE-14/1 (2014).
7. B. J. Toppel, "A User's Guide to the REBUS-3 Fuel Cycle Analysis Capability," ANL-83-2, Argonne National Laboratory (1983).

8. M. A. Smith, C. H. Lee, and R. N. Hill, "GAMSOR: Gamma Source Preparation and DIF3D Flux Solution," ANL/NE-16/50 Rev. 1.0, June 28, 2017.
9. K. L. Basehore, N. E. Todreas, "SUPERENERGY-2: A Multi-Assembly Steady-State Computer Code for LMFBR Core Thermal-Hydraulic Analysis," PNL-3379 (1980).
10. M. A. Smith, , "VARI3D & PERSENT: Perturbation and Sensitivity Analysis," ANL/NE-13/8 Rev. 2 (2018).
11. W. S. Yang, "RCT - A Code to Reconstruct Pin Power and Burnup Characteristics from Nodal Calculations in Hexagonal Geometry," ANL-IFR-192, April 1993.
12. Won Sik Yang, "Sensitivity Theory for Reactor Burnup Analysis based on Depletion Perturbation Theory (DPT)," PhD Thesis Purdue University, May 1989.
13. J. Grudzinski, T. Moran, C. Grandy, "Supplement to the NUBOW-3D Manual," ANL/NE-15/9 Rev. 1 (2015).
14. W. H. Hannum, Ed., "The Technology of the Integral Fast Reactor and its Associated Fuel Cycle," Progress in Nuclear Energy, **31**, No 1 and 2, Special Issue 1997.
15. J. F. Briesmeister, et al., "MCNP – A General Monte Carlo N-Particle Transport Code," LA-13709-M, Los Alamos National Laboratory (2000).
16. G. Aliberti, G. Palmiotti, M. Salvatores, "Representativity Studies for Sodium- and Gas-Cooled Reactors", ANS Meeting, Albuquerque (NM), USA, November 2006.
17. American Society of Mechanical Engineers, "Quality Assurance Requirements for Nuclear Facility Applications," ASME NQA-1-2008, 2008.
18. H. F. McFarlane, S. B. Brumbach, S. G. Carpenter, and P. J. Collins, 'Benchmark Physics Tests in the Metallic-Fueled Assembly ZPPR-15," *Nucl. Sci. Eng.*, **101**, pp. 137 – 152 (1989).
19. R. W. Benedict and S. P. Henslee, 'The EBR-H Spent-Fuel Demonstration Project" Trans. Am Nucl. Soc., **77**, p 75-76 (1997).
20. J. D. Bess, et. al, "Evaluation of the Initial Isothermal Physics Measurements at the Fast Flux Test Facility, a Prototypic Liquid Metal Fast Breeder Reactor." INL/EXT-09-16524, Idaho National Laboratory, March 2010.

Nuclear Science and Engineering Division

Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439-4842

www.anl.gov

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