DISCLAIMER

ANL/NSE-25/67 Rev. 1

Fiscal Year 2025 Software Quality Assurance Activities for the ARC Software

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**

ANL/NSE-25/67 Rev. 1

Fiscal Year 2025 Software Quality Assurance Activities for the ARC Software

---

prepared by

Micheal A. Smith, Kalin R. Kiesling, Nicholas Wozniak, Ting Fei, Zhaopeng Zhong

Nuclear Science and Engineering Division
Argonne National Laboratory

September 30, 2025

EXECUTIVE SUMMARY

The continued goal of the ARC SQA project in the Advanced Reactor Technologies program of DOE is to resolve the QA gaps for the ARC software that limit, or prevent, commercialization of the software for industry users. This project started in earnest in fiscal year 2023 which saw the entire code system moved from a SVN repository to a GitLab repository and an associated software quality assurance plan (SQAP) developed and ratified. Most of the QA gaps in the ARC software were identified in collaboration with industry partners and work begin in fiscal year 2023 and continued through 2024 and 2025. The continuous integration testing was extended to RCT, DASSH, and SE2ANL. Minor changes were required to the original continuous integration methodology to make this happen. When full confidence in the methodology is complete, a report will be created to detail the automated regression testing methodology and minor reports will be created to detail the tolerance settings that have been applied to the output for each ARC code.

The primary documentation that is missing includes user manuals, user guides, software verification reports, and code coverage assessments. The DASSH, SE2ANL, and SE2RCT manuals were completed this fiscal year. A review of the SE2ANL software identified that it is unrealistic to include updated correlations or different geometry models and it was scheduled for deprecation in favor of DASSH. The SE2ANL manual is essential for SE2RCT as they are similar but quite different in purpose. The only piece of software missing a manual consistent with the source code is NUBOW-3D which is a focus of the coming year. The code coverage report for DIF3D was updated and code coverage reports were created for REBUS, RCT, PERSENT, GAMSRC, and DASSH. Minor coverage issues were identified for all of these pieces of software which did not prevent the work done to transition them to the OneAPI compiler. Because SE2ANL was scheduled for deprecation, it was not transitioned, but it was successfully tested with the OneAPI compiler. This leaves SE2RCT and NUBOW-3D as the only pieces of software not transitioned to OneAPI and further work is required to get SE2RCT to work properly. The SE2RCT software transition will begin early next year while the NUBOW-3D software requires a manual before it can begin.

Software verification work has been completed for DIF3D, REBUS, GAMSOR, GAMSRC, VARPOW, EvaluateFlux, and SUMMAR. The PERSENT software verification work was completed this year which was somewhat delayed because of unexpected bugs in the software. The PERSENT manual was updated to detail some of the issues and discuss the bowing reactivity worth feature added in the previous fiscal year. The RCT, DASSH, SE2RCT, and NUBOW-3D software are the only maintained pieces of software without verification reports. The software verification work for DASSH will be a focus in the upcoming fiscal year and it is hoped that some of the test cases created can serve as verification tests for SE2RCT. The NUBOW-3D work will begin when the manual and requirements report are completed.

Only minor industry partner software development funds were provided this year. The DASSH software was updated to handle general axial geometry for each assembly and the NUBOW-3D software was updated to incorporate a new input format and better output. Overall progress on resolving the QA gaps has been good this year.

TABLE OF CONTENTS

Executive Summary 2

Table of Contents 1

List of Figures 2

List of Tables 3

1. Introduction 4

2. ARC Software Quality Assurance Work 6

   2.1 Status of the restarted QA Program for ARC 7

   2.2 QA and Commercialization Assessment of the ARC Software 8

3. Industry Specific Developments 26

   3.1.1 DASSH upgrades. 26

   3.1.2 NUBOW-3D Input and Output Modifications 28

   3.1.3 PyARC modifications to include DASSH and NUBOW-3D 32

4. Conclusions 33

References 34

LIST OF FIGURES

Figure 2.1. ARC Fuel Cycle Analysis Software Connections 6
Figure 2.2. ARC Software Connections for Fast Spectrum Reactor Analysis 7
Figure 3.1. Example DASSH Axial Geometry Variation Possible in the New Version. 26
Figure 3.2. Example DASSH-F Temperature Profile Interpolation at Axial Boundary. 27
Figure 3.3. Example of NUBOW-3D New Structured Input. 29
Figure 3.4. Example of NUBOW-3D Radial Displacement Visualization in the Axial View. 30
Figure 3.5. Example of NUBOW-3D Visualization of Load Pad Contact Force Values at the ACLP (left) and TLP (right) at Each Duct Face.. 31

LIST OF TABLES

Table 2.1. ARC Software Identified for Commercialization for Fast Spectrum Reactor Analysis 9
Table 2.2. Basic Documentation Required for Software Commercialization 10
Table 2.3. DIF3D Software Documentation Status 12
Table 2.4. Outstanding DIF3D Software Updates 13
Table 2.5. REBUS Software Documentation Status 14
Table 2.6. Outstanding REBUS Software Updates 15
Table 2.7. PERSENT Software Documentation Status... 16
Table 2.8. Outstanding PERSENT Software Updates 17
Table 2.9. GAMSOR/GAMSRC Software Documentation Status 20
Table 2.10. Outstanding GAMSOR/GAMSRC Software Updates 20
Table 2.11. SE2ANL Software Documentation Status.. 20
Table 2.12. Outstanding SE2ANL Software Updates.. 21
Table 2.13. SE2RCT Software Documentation Status 21
Table 2.14. Outstanding SE2RCT Software Updates 22
Table 2.15. DASSH Software Documentation Status 22
Table 2.16. Outstanding DASSH Software Updates 23
Table 2.17. RCT Software Documentation Status 23
Table 2.18. Outstanding RCT Software Updates... 24
Table 2.19. NUBOW-3D Software Documentation Status 25
Table 2.20. Outstanding NUBOW-3D Software Updates.. 25

1. Introduction

The Argonne Reactor Code (ARC) software suite [1-18] has been developed by Argonne researchers for fast reactor design and analysis since the 1970s. With the ARC software suite, a user can quickly build a model of a proposed or existing fast spectrum reactor and carry out fuel cycle, nominal thermal analysis and flow requirements, and assess, as is appropriate, whether the core design and constraint system yield an acceptable mechanical behavior. For transient reactor analysis with SAS4A [19], the ARC software suite can be used to generate reactivity coefficients and kinetics parameters at any modeled fuel cycle time point which forms part of the input to SAS4A.

The ARC suite was regularly being developed until the 1990s and followed a software QA program which was an appropriate standard for the time. In the 1990s, the DOE funding to fast reactor research and development was all but eliminated and the ARC software was put into maintenance mode. In the early 2000s, the software quality assurance (SQA) program for ARC was still in place to define an official version, but by 2005 it all but was abandoned as there were insufficient staff to fill the work roles.

Since 2005, there has been a considerable increase in research and design work on fast spectrum reactors. The ARC software as a whole has since been exported to many universities and commercial companies and ANL support has been given to the various projects over the years [20-24]. Further, MC^2-3, PERSENT, and DASSH were all developed after 2005 without any adherence to a software standard. In recent time, the DOE VTR project [23] paid for verification work to be done on the ARC software as part of the goal of making it NQA-1 complaint. The VTR project was not considered the appropriate pathway to fund and maintain a SQA program for the ARC software and while software developments (DASSH) were made and several manuals were updated and software verification work was carried out, the ARC software is not NQA-1 compliant.

More recently the Advanced Reactor Development Program (ARDP [24,25]) has funded the creation of manuals for some ARC utility programs and funded additional software verification work on DIF3D [6, 7], MC^2-3 [2-5], and PERSENT [9] for the purpose of commercial grade dedication. Additional funding through the technology commercialization fund [26] motived work on DASSH, PERSENT, and NUBOW-3D all of which was connected in PyARC [27]. Because of the VTR and ARDP projects, software verification work was completed on MC^2-3 [28-31], DIF3D [32], EvaluateFlux [33], GAMSOR [34], VARPOW [35], REBUS [36], and PERSENT [37]. These reports discuss the inputs for and outputs from the codes that are covered by the verification work and link various analytic, code-to-code, and hand calculation-based verification work presented in the report with verification test problems provided with the software. This is a key part of the commercial grade dedication work and constitutes the bulk of the cost to get the ARC software to commercial grade. The SE2ANL, DASSH, RCT, SE2RCT, and NUBOW-3D codes are the ones that still need software verification to be completed at this time.

With the support of the advanced reactor technology program in DOE [38], the software quality assurance program on ARC was rebuilt in 2023 [39] based upon GitLab [40,41]. As inferred from the preceding discussion, the ARC software suite is a valuable asset as a fast reactor design and analysis tool set that has been reasonably well verified and validated with various fast reactor benchmark problems and experiments over several decades. Some or all of the ARC software suite has been utilized for designing the IFR [21], PGSFR [22], VTR [23], and Natrium [24] reactors and we can expect it to continue to be used for advanced fast reactor design and/or confirmatory calculation purposes in the future. Due to increased interest by commercial companies and regulatory bodies, it is becoming more important to make the ARC software suite complete and ready-to-use in terms of its SQA pedigree and commercial grade dedication needs. This report discusses the achievements made in fiscal year 2025 under the SQA program on the ARC software and dealing with outstanding identified QA gaps. It also summarizes the development work done on the ARC software as part of this project or collaborative efforts with industry partners to improve the ARC software.

2. ARC Software Quality Assurance Work

The ARC software suite consists of many individual software products usable on, and in some cases specifically tailored for, fast reactor design and analysis. As the software suite was developed with small computational resources compared with today, most of the analysis capability is not on explicit geometry modeling. Instead, the software focuses on providing an analysis capability based upon a homogenized geometry treatment or simplified geometry definition on an assembly basis. With this in mind, the main software of note for the current external stakeholders includes the multi-group cross section library generation codes (ETOE2 [4] and MC^2-3 [2-3]), neutron and gamma transport calculation (GAMSOR [10]), fuel cycle analysis calculation with REBUS [8], pin power reconstruction and depletion analysis (RCT [11]), steady state thermal analysis (SE2ANL [17], SE2RCT [16], and DASSH [13,14]), perturbation and sensitivity analysis (PERSENT [9]) and a mechanics code to assess the impact of structural forces and irradiation induced creep during nominal reactor operations (NUBOW-3D [15]). Figure 2.1 shows the ARC software component (orange boxes) workflow and their connections for typical fuel cycle analysis work.

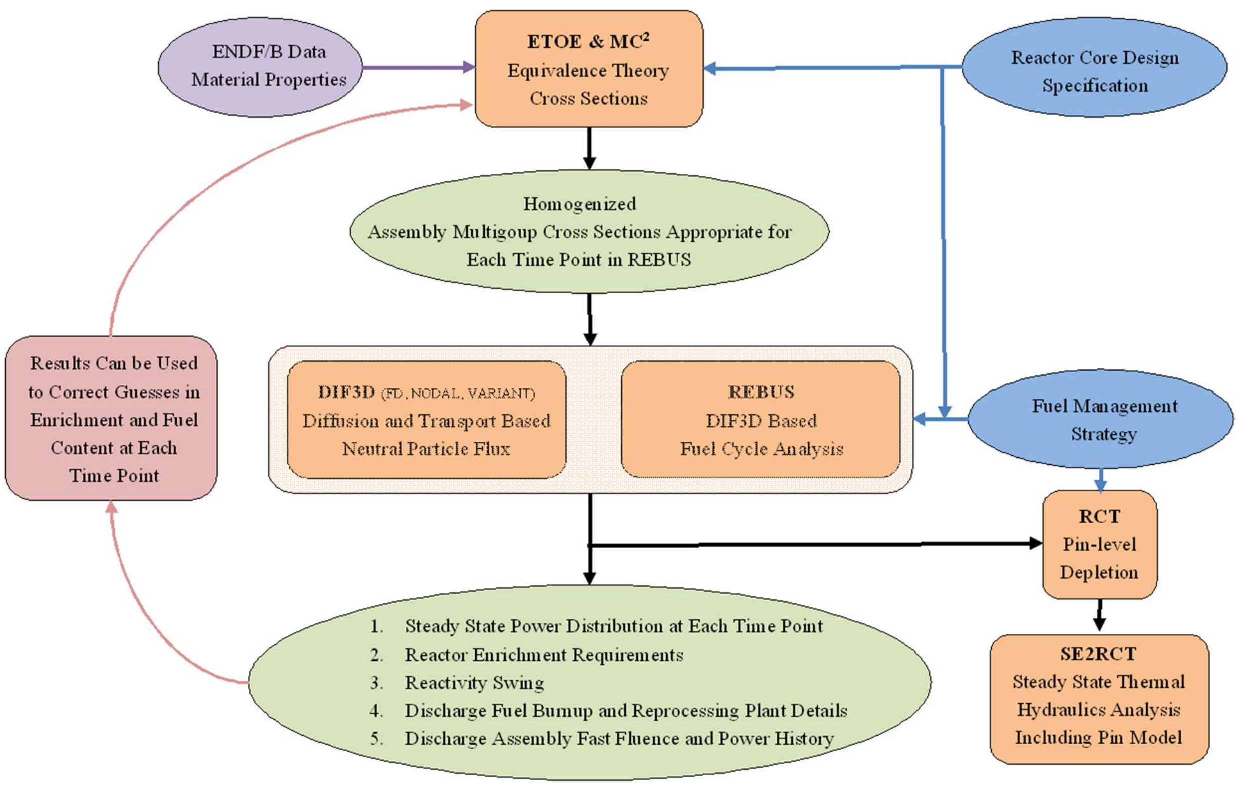

Figure 2.1. ARC Fuel Cycle Analysis Software Connections

Figure 2.2 shows the ARC software components and workflow to perform the follow-on steady state thermal analysis, assess the transient reactor performance with SAS4A [19], and assess the mechanical behavior of the system (NUBOW-3D). Within this SQA Program is also support (in collaboration with NEAMS [42]) for the development of the PyARC [27] code to support the needs of industry. PyARC is a workflow management system for the ARC codes that automates the input creation, execution, and postprocessing of the ARC codes to enhance usability, efficiency, and code-to-code comparison. Some of the ARC codes development work detailed in section [Industry Specific Developments] was done in coordination with PyARC development or motivated by PyARC usage.

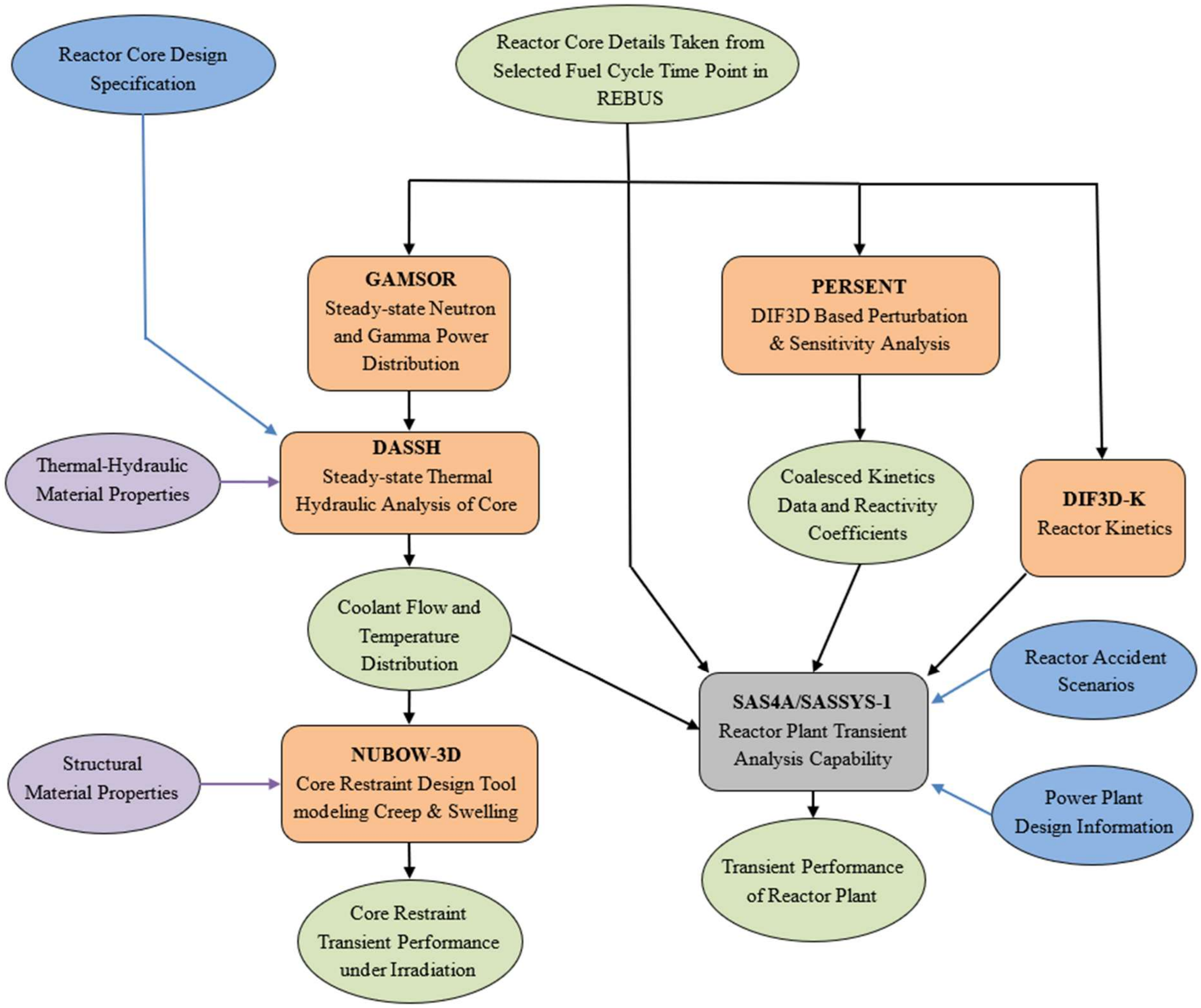

Figure 2.2. ARC Software Connections for Fast Spectrum Reactor Analysis

2.1 Status of the restarted QA Program for ARC

As stated earlier, the ARC suite was consistently being developed until the early 1990s and had a SQA program in place up to around 2005. With funding from NEAMS [42], the ARC software was pulled from the SCCS repository on SunOS in 2009 and put into a SVN repository on Linux. The NEAMS program funded the development of an automated nightly regression testing package for UNIC and via that program the existing regression test suite for ARC was reorganized and included. There is thus digital traceability of the ARC software changes made since 2009, including nightly software verification reports, although there are scant paper records to document the various release versions or software changes. Some of the ARC software such as SE2RCT [16], SE2ANL [17], RCT [11], and NUBOW-3D [15] were not added to this repository until later as they were simply not being used or had only a single user and maintainer. The EBR-II Fuels Irradiation & Physics Database (FIPD) project [20] under ART required a more rigorous QA be applied to the RCT and SE2RCT (also known as SE2P) codes and thus they were added to the repository and the nightly regression testing in 2017, but the funding was cut before the NQA-1 program could be put in place. The current effort on FIPD starting in 2020 was focused on restarting the QA program for FIPD and thus performing the necessary verification work on RCT and SE2RCT.

The fiscal year 2023 focused on building the QA program for ARC started from the end point of the existing development progress. In 2023, a new SQAP [39] was created around the GitLab software control system and it was implemented for all of the ARC software. Considerable time was spent in fiscal year 2024 to update the automated regression testing. In a previous program, a automated regression or continuous integration (CI) capability was put in place for ARC which would compile and run the verification test suite for each ARC code each night. That infrastructure used an execution and output checking methodology that was tailored for each ARC code which was not done consistently nor documented. With the move to GitLab and the new SQAP, this process was generalized in fiscal year 2023 but only applied to the DIF3D software and its utility programs. In fiscal year 2024, time was spent modifying that CI version to make a single output checking methodology viable for all of the ARC software. This will result in a single document to describe the CI methodology for the ARC software and minor documentation to detail the tolerance settings applied to the output for each ARC code which will be created when we are confident that it is robust across all ARC codes. The previous SVN based methodology would have required a separate document for each ARC code to detail the output checking methodology and the tolerance settings for the output from each code. In the current fiscal year, the CI capability was extended to all of the ARC codes. In addition, many QA gaps were resolved by creating the necessary reports and software manuals that were previously not available.

As shown in Figure 2.1 and Figure 2.2, there are several different software components in the ARC software and up to this point, VARI3D, DIF3D-K, and DPT are not presently included in the QA program. This is primarily because the usage of these pieces of software is virtually non-existent today and no path forward for commercialization has been identified. As this is a QA program, one of the important tasks is to identify the commercialization needs of each piece of software and identify the QA gaps for those pieces of software.

2.2 QA and Commercialization Assessment of the ARC Software

Table 2.1 lists the primary pieces of the ARC software and their function where most of these appear in the workflow of Figure 2.1 and Figure 2.2. This section will review the past discussion on the QA assessment and give updates on the accomplished work in the current fiscal year. From the descriptions in Table 2.1, it should be clear that the ARC software is used to do steady state and transient analysis of a fast reactor. Steady state characterization is required to ensure the reactor, and its associated systems, behave as expected during normal operation. This analysis step is also important to the definition of plant conditions prior to the onset of transients. Assessments of the neutronic performance of the core, progression of the fuel cycle and fuel performance, and status of the heat transport systems must be performed. Additionally, performance of the plant and supporting systems during transients must be characterized to predict the degree of satisfaction of predetermined safety metrics (e.g., peak clad temperature, margin to sodium boiling, etc.). Accordingly, the fuel performance, including fission gas behavior and fuel/clad motion, status of heat transport systems, structural response, and source term must be analyzed.

Phenomena unique to SFRs, including sodium-water interactions, sodium fire, and inherent reactivity feedback, must also be assessed. Not all of these areas are covered by the ARC software suite, but the pieces that have been identified with outstanding commercialization needs are listed in Table 2.1.

Table 2.1. ARC Software Identified for Commercialization for Fast Spectrum Reactor Analysis

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 Neutral Particle Flux Distribution	GAMSOR GAMSRC	Coupled neutron-gamma Flux Solution
REBUS	Fuel Cycle Analysis	RCT	Fuel Pin Level Fuel Cycle Analysis
SE2ANL	DIF3D-FD Based Steady State Thermal-hydraulics Analysis	DASSH DASSH-F	DIF3D-VARIANT Based Steady State Core Thermal Analysis
SE2RCT	DIF3D-Nodal Based Steady State Assembly Thermal-hydraulics Analysis	NUBOW-3D	Thermal Mechanical Analysis of Core Restraint System with Irradiation Creep and Swelling
PERSENT	DIF3D-VARIANT Based Reactivity Coefficient and Sensitivity Analysis	VARI3D	DIF3D-FD Based Reactivity Coefficient and Sensitivity Analysis
DIF3D-K	DIF3D-Nodal Based Reactor Kinetics Capability that can be Incorporated into SAS4A for Dynamics Analysis	SAS4A	Liquid Metal Coolant Reactor Safety Analysis

VARI3D has been deprecated by the availability of PERSENT and will not be discussed further in this document. SAS4A is not considered part of ARC and will not be covered in this document. MC^2-3 and ETOE are primarily funded from NEAMS today and, while a SQAP was created for it and a code coverage assessment was done as part of this project, they will not be covered further in this document as these codes are hosted in their own GitLab repository noting that extensive verification work has been completed on the software [28-31].

As stated earlier, the VTR and ARDP projects invested a considerable amount of funding to push DIF3D and MC^2-3 forward as part of a commercialization effort. Using experience from that work, the basic documentation (and work) needed to support software commercialization is identified in Table 2.2.

Table 2.2. Basic Documentation Required for Software Commercialization

User manual	Details the equations being solved by the software in question and the underlying methodology. Input and Output are described as is necessary to facilitate the discussion of what the software does. Material property information is considered part of the input and at least one source should be mentioned.
User guide	Details how to construct inputs and covers what the outputs are. One or more example problems are displayed and the inputs/outputs for them are provided with the software distribution. A detailed dissection of a given example problem is provided to show how to properly use the software and interpret the output.
Software design report	Generally a part of new software developments where it lays out the goals of the software and to some degree the computing requirements.
(Commercial) software requirement report	States which features of the software are being used by industry. In particular, the inputs and outputs being used by the commercial entity must be identified. To satisfy the licensing requirements of the software, any identified inputs and outputs should be verified to work which is documented in a software verification report and reproducible by a distributable set of test input and output that the end commercial user can execute to verify a dedicated executable.
Software verification report	Details a set of test problems which prove that the stated equations being solved by the software are being solved in the manner specified in the user manual for the software. The test problems will also prove that the input provided to the software is being handled properly and that the output from the software is consistent with the user manual.
Code coverage report	For a given set of test problems that are maintained with the software for regular testing, the code coverage report details what subroutines in the software are covered by the test problems and which are not. Any QA gaps with respect to software coverage should be identified in this report noting that they may not be important for a given software requirement report and the associated software verification report.

This list of documents is not all inclusive in that there is potential that a commercial entity would request additional documentation not listed here (such as detailed source code change history). Each of the documents listed in Table 2.2 are intended to live with the software itself such that as source code changes (bugs, performance enhancement, and feature additions) are made, the documentation is consistently updated and new revisions of the software and documents are released together. These documents, when available, are maintained in the same GitLab source code control repository that the ARC software is stored in. In the tables that follow, a set of tasks to complete the commercialization work on each piece of software identified in Table 2.1 are displayed. An importance is assigned to each task as is a status of work being done to resolve the outstanding issues as part of this project. The parts that were worked on this fiscal year are highlighted with a **bold font**.

Table 2.3 provides the documentation status for the DIF3D software. The only change made this year was to update the code coverage report of DIF3D [43]. This is required as the code coverage report must be consistent with the recently modified version of DIF3D that included threaded capabilities. This ensure that the OneAPI conversion done this year would be fully tested by the existing test suite. The remaining documents are deemed low priority at this time where an existing report [44] serves as the user guide for DIF3D-VARIANT.

Table 2.4 provides a list of known software maintainability and usability issues for DIF3D where again the font is bolded for work done in the present fiscal year. Note that because many parts of the ARC software still contain Fortran 66, and some of that source coding has been deprecated by modern compilers, updates to the software for portability and maintainability are required for future use of the software. As seen, only the OneAPI compatibility was achieved in the current year.

Table 2.5 provides the documentation status for the REBUS software. As seen, the verification report was completed [36] and the code coverage report was also created [45]. Table 2.6 provides a list of known software maintainability and usability issues for REBUS where the only accomplishment was getting OneAPI support.

Table 2.7 provides the documentation status for the PERSENT software. As seen, the manual [9] was updated to detail the bowing reactivity worth capability added to PERSENT in the past year. The software verification report [37] was completed which took longer than expected due to the presence of serious bugs in the N-2N scattering treatment and the 60 and 120 degree periodic geometry sensitivity calculations. The code coverage report [46] was also completed. It is noted that there are relatively few issues to resolve compared with DIF3D and REBUS where Table 2.8 provides the list of known software maintainability and usability issues for PERSENT. One issue resolved this year is the OneAPI conversion which identified a very minor bug in the DIF3D-VARIANT software which was present since its creation. This delayed the OneAPI conversion by about a week as the bug was identified and resolved.

Table 2.3. DIF3D Software Documentation Status

Description	Importance	Status
The user manual consists of multiple reports covering different parts of the DIF3D input and the DIF3D solvers themselves. Additional documentation on BPOINTER and the file I/O are available to help new software developers understand how it works.	Medium	Complete
There is no user guide and it is wise to create one as there are many peculiarities of running older software like DIF3D that are not discussed in the current manuals.	**Low**	Not started
There is no software design report.	Low	Not started
The existing software requirement report is from VTR.	High	Complete
The verification report is up to date with the latest revision of the software.	High	Complete
The code coverage report is up to date with the latest revision of the software.	High	**Updated** Complete

Table 2.4. Outstanding DIF3D Software Updates

Description	Importance	Status
DIF3D needs to support OneAPI compilation where the present production executables are created with the 2013 Intel compiler.	**High**	**Complete**
The Fortran 66 features of BPOINTER and the file I/O system should be updated or replaced. Modernization of the memory management is difficult and time consuming. Changing the file management in DIF3D requires simultaneous changes to all of the ARC software that uses the same libraries.	High	Not started
The special inhomogeneous fixed source solver in DIF3D that is used by PERSENT is applied externally to DIF3D by PERSENT. It would be more efficient to have it applied internally to DIF3D to eliminate the performance issues that plague using the one in PERSENT.	Low	Not started
The fixed source solution algorithm, used by GAMSOR, has no convergence criteria check. One should be added so that the user does not have to guess how many iterations are needed.	Medium	Not started
DIF3D-VARIANT has a rather ineffective outer iteration acceleration algorithm. Coarse mesh rebalance was implemented previously but it was not working on hexagonal geometries and disabled in favor of Tchebychev.	Low	Not started
The peaking calculation in DIF3D-VARIANT is using the DIF3D-Nodal one. It would be wise to replace this with one that is consistent with DIF3D-VARIANT to avoid needing to maintain the VARPEAK utility program.	Low	Not started

Table 2.5. REBUS Software Documentation Status

Description	Importance	Status
The user manual was updated as part of the VTR project.	Low	Complete
There is no user guide and it is wise to create one as there are many peculiarities of running older software like REBUS that are not discussed in the current manual.	Medium	Not started
There is no software design report.	Low	Not started
The existing software requirement report is from VTR.	High	Complete
The existing software verification report is from VTR but requires referencing to the new test problems to meet commercial user requirements.	**High**	**Complete**
The code coverage report is not present	**High**	**Complete**

Table 2.6. Outstanding REBUS Software Updates

Description	Importance	Status
REBUS needs to support OneAPI compilation where the present production executables are created with the 2013 Intel compiler.	**High**	**Complete**
The Fortran 66 features of BPOINTER and the file I/O system should be updated or replaced. Modernization of the memory management is difficult and time consuming. Changing the file management in REBUS requires simultaneous changes to all of the ARC software that uses the same libraries.	High	Not started
The burnup in MWD/MT and atom % are not consistent with the text book definitions. It should be replaced so that the code uses the equations in the textbook and produces what the user can produce with a simple hand calculation.	Medium	Not started
There is an initialization problem when using card type 37 in A.BURN	Low	Not started
The binary files stored on the STACK file do not correspond to the final converged result in the output, but the preceding iteration. The final ones need to be stored.	Medium	Not started
The Modify utility program allows the DIF3D input to be reconstructed at any REBUS time point but it is not a perfect translation. PyARC similarly builds the DIF3D models by importing the ASCII output from the REBUS deck which means it must have REBUS dump the atom density output at each time point. Is there a better way to facilitate this?	Low	Not started

Table 2.7. PERSENT Software Documentation Status

Description	Importance	Status
The user manual (and guide) are up to date with the current release version.	Low	**Updated Complete**
There is no software design report.	Low	Not started
The existing software requirement report is from VTR.	High	Complete
The existing software verification report was started under VTR but is not complete.	**High**	**Complete**
The code coverage report is not present.	**High**	**Complete**

Table 2.8. Outstanding PERSENT Software Updates

Description	Importance	Status
PERSENT needs to support OneAPI compilation where the present production executables are created with the 2013 Intel compiler.	**High**	**Complete**
The existing PERSENT code is entirely serial. To keep consistent with existing computing resources, it would be wise to thread some aspects of the PERSENT source code that presently limit the usability of higher fidelity (more meshes, more energy groups, higher space-angle approximations, etc…) models.	Medium	Not started
The existing sensitivity coefficients for Beta are done for the total beta and there is an outstanding request for the isotopic breakdown.	Low	Not started
The UQ routine has not been tested with any other COMMARA like dataset. There is one available and PERSENT should be modified or verified that PERSENT can work with it.	Low	Not started
The interface to the DIF3D-VARIANT basis functions is rather crude because the basis functions were built to operate with DIF3D and not be a generic application.	Low	Not started
The special inhomogeneous fixed source solver is presently applied external to DIF3D and should be part of DIF3D.	Low	Not started
The convergence criteria on the homogeneous eigenvalue problem needs to be better managed when the inhomogeneous problem is solved as insufficient convergence in the homogeneous one will lead to divergence or slow convergence in the inhomogeneous one	Low	Not started

Table 2.9 provides the documentation status for the GAMSOR and GAMSRC software. GAMSOR is not being used by industry partners in favor of GAMSRC. This is because GAMSOR was a modified version of DIF3D and GAMSRC is an independent utility version that can be executed cleanly without interferring with DIF3D. Because these two pieces of software are used to carry out the same coupled neutron/photon transport calculation, the documentation and verification work done cover both pieces of software. The verification of GAMSRC [34] was completed in the previous fiscal year and the code coverage report [47] was completed this fiscal year leaving only the software design report as missing. This report is not deemed essential for commercial grade dedication. In the next fiscal year, the goal is to deprecate GAMSOR and officially remove it from the list of maintained software.

Table 2.10 provides a list of known software maintainability and usability issues for GAMSOR/GAMSRC. In earlier versions of this table, the VARPOW utility program was cited as missing documentation. VARPOW is used to interface the ARC codes with DASSH and the regression testing is included as part of the GAMSOR/GAMSRC regression testing. The verification work on VARPOW was completed in the previous year [35] and because VARPOW is called as a subroutine in DASSH, the code coverage for it is assessed as part of DASSH instead as its own document. VARPOW will be considered a part of DASSH going forward noting that testing is done in the GAMSOR/GAMSRC repository and the DASSH repositories. As seen in Table 2.10, the only remaining issue to address with GAMSRC is the OneAPI conversion which was completed this year. OneAPI conversion of GAMSOR was not attempted as the goal is to deprecate it next year.

Table 2.11 provides the documentation status for the SE2ANL software while Table 2.12 provides a list of known software maintainability and usability issues associated with it. As seen, there was considerable documentation that was identified to be missing, in particular the user manual. That manual was created this fiscal year and after assessing the capabilities of SE2ANL against that of DASSH, we are choosing to deprecate SE2ANL in the next fiscal year and remove it from the list of ARC software that we maintain. This is primarily because of the geometric shortcomings that are inherent to SE2ANL which are either resolved, or easier to resolve, with DASSH. While industry users have relied upon the SE2ANL capability up to this point, we have begun transitioning their work to using DASSH [14] going forward and PyARC was successfully upgraded to support the most recent version of DASSH detailed later in this report. The OneAPI compiler was tested and verified to work on SE2ANL with the existing test suite. Because the code coverage report was not completed beforehand we cannot be fully certain that SE2ANL is OneAPI complaint at this time.

Table 2.13 provides the documentation status for the SE2RCT software while Table 2.14 provides a list of known software maintainability and usability issues associated with it. As was the case with SE2ANL, there is considerable documentation that was identified to be missing. The SE2RCT manual was created this fiscal year and unlike the SE2ANL software we cannot deprecate it as it is an essential component of FIPD. The code coverage report and OneAPI conversion was not completed this year because without the manual, it was not clear what the requirements report would be. The goal for the next fiscal year is to develop the requirements document and code coverage report. Because the verification work for DASSH is scheduled for the next fiscal year, it is possible that some of that work can be used for the verification of SE2RCT. However, it is worth noting that there is a significant desire to make use of DASSH as a replacement for SE2RCT which will be assessed in the coming fiscal year.

Table 2.15 provides the documentation status for the DASSH software while Table 2.16 provides a list of known software usability issues. The original version of DASSH was written in python [13] while an updated version was written in Fortran [14]. Because of the performance issues and missing functionality, commercialization of the python version is not being pursued and all work has ceased on it. The manual for the Fortran version was completed this year [14] and as part of that work, quite a bit of source code modifications were made to produce output amenable to software verification. Additional work was done to fix the geometry constraints that only allowed a single axially heterogeneous model which is detailed later in the report. As seen from the two tables, there is some missing documentation needed for commercialization of the software which is scheduled to be completed next year. The expectation is that the remaining major issues identified in Table 2.16 will also be resolved in the next year.

Table 2.17 provides the documentation status for the RCT software while Table 2.18 provides a list of known software maintainability and usability issues. The RCT software is not a target for commercialization but is a vital part of the FIPD project and software verification work needs to be provided for it to be in compliance with NQA-1. No significant work was completed on RCT this year and it is one of the major ARC pieces of software preventing the transition to a newer compiler in the ARC system. As was done for SE2ANL, the OneAPI compiler was used on RCT and it successfully compiled and ran all test problems. With the recently completed code coverage report, some additional work will be needed to ensure all capabilities are working when the OneAPI compiler is fully adopted for RCT.

Table 2.19 provides the documentation status for the NUBOW-3D software while Table 2.20 provides a list of known software maintainability and usability issues. The NUBOW-3D software is a target for commercialization but requires significant work before it can achieve this goal. Quite a bit of work was done on upgrading the user input interface of NUBOW-3D this year as part of the current software quality assurance project. In addition, an ongoing TCF collaboration [26] with Westinghouse Electric Corporation was successful in connecting NUBOW-3D to DIF3D-VARIANT, DASSH-F, and PERSENT as will be discussed in a later section of this report. The existing manuals will have to be updated as a consequence of the new input interface which will be a goal in the coming year as indicated in the table. Since all of the NUBOW-3D work will be done consistent with the SQAP governing the ARC software, the SQA aspects will be covered by this project.

Table 2.9. GAMSOR/GAMSRC Software Documentation Status

Description	Importance	Status
The user manual (and guide) are up to date with the current release version.	Low	Complete
There is no software design report.	Low	Not started
The existing software requirement report is from VTR.	High	Complete
The verification report is up to date with the latest revision of the software.	High	Complete
The code coverage report is not present.	**High**	**Complete**

Table 2.10. Outstanding GAMSOR/GAMSRC Software Updates

Description	Importance	Status
GAMSOR**/GAMSRC needs to support OneAPI compilation where the present production executables are created with the 2013 Intel compiler.**	**High**	**Complete**

Table 2.11. SE2ANL Software Documentation Status

Description	Importance	Status
There is no user manual or guide.	**High**	**Complete**
There is no software design report.	**Not needed**	Not started
There is no software requirement report.	**Not needed**	Not started
There is no software verification report.	**Not needed**	Not started
The code coverage report is not present.	**Not needed**	Not started

Table 2.12. Outstanding SE2ANL Software Updates

Description	Importance	Status
SE2ANL/SE2RCT needs to support OneAPI compilation where the present production executables are created with the 2013 Intel compiler.	**High**	**Complete**
The Fortran 66 features should be updated or replaced.	**Low**	Not started
The input is not checked for errors but relies upon floating point exception traps.	**Low**	Not started
The correlations are not checked to ensure they are being used properly.	**Low**	Not started
The output is organized in different ways with the same numbering (i.e. assembly 1 means different actual assemblies in different parts of the output).	**Low**	Not started
The input and output are in British units. Some of the output is in SI units and it is preferred that it all be in SI.	**Low**	Not started
The flow split, mixing, and friction factor correlations should be updated.	**Low**	Not Started

Table 2.13. SE2RCT Software Documentation Status

Description	Importance	Status
There is no user manual or guide.	**High**	**Complete**
There is no software design report.	**Not needed**	Not started
There is no software requirement report.	High	**In Progress**
There is no software verification report.	High	Not started
The code coverage report is not present.	High	Not started

Table 2.14. Outstanding SE2RCT Software Updates

Description	Importance	Status
SE2RCT needs to support OneAPI compilation where the present production executables are created with the 2013 Intel compiler.	High	Not started
The Fortran 66 features should be updated or replaced.	High	Not started
The input is not checked for errors but relies upon floating point exception traps.	High	Not started
The correlations are not checked to ensure they are being used properly.	Medium	Not started
The output is organized in different ways with the same numbering (i.e. assembly 1 means different actual assemblies in different parts of the output).	Medium	Not started
The input and output are in British units. Some of the output is in SI units and it is preferred that it all be in SI.	Low	Not started
The flow split, mixing, and friction factor correlations should be updated.	High	Not Started
The RCT power shape is not actually used as defined, but recast into the existing SUPERENERGY-2 form of a single radial profile combined with a single axial profile.	Medium	Not Started

Table 2.15. DASSH Software Documentation Status

Description	Importance	Status
There is no user manual or guide for DASSH.	**High**	**Complete**
There is no software design report.	Low	Not started
There is no software requirement report.	High	Not started
There is no software verification report.	High	Not started
The code coverage report is not present.	**High**	**Complete**

Table 2.16. Outstanding DASSH Software Updates

Description	Importance	Status
DASSH needs to support OneAPI compilation where the present production executables are created with the 2013 Intel compiler.	**High**	**Complete**
The implemented pin geometry model is not consistent with some assembly designs that consist of solid blocks with coolant holes drilled through them.	High	Not Started
The automatic orifice selection routine does not function as desired	Medium	Not Started

Table 2.17. RCT Software Documentation Status

Description	Importance	Status
There are two user manuals for RCT and neither function as a user guide. Neither manual has a clear description of how the input translates to the output generated by RCT.	Medium	Not Started
There is no user guide.	**Low**	Not Started
There is no software design report.	Not needed	Not Started
There is a software requirement report but it has not been reviewed or approved.	High	Not Started
There is no software verification report.	High	Not Started
The code coverage report is not present.	**High**	**Complete**

Table 2.18. Outstanding RCT Software Updates

Description	Importance	Status
RCT needs to support OneAPI compilation where the present production executables are created with the 2013 Intel compiler.	**High**	**Complete**
The Fortran 66 features should be updated or replaced.	High	Not started
Only part of the input is checked for accuracy and invalid input can easily be provided which results in invalid output	Medium	Not started
The pin follow depletion approach is not combined with the spatial depletion model to preserve the average depletion results from REBUS. This allows the pin depletion results to drift with time and the consequence of this approximation should be inspected and documented as the EBR-II validation basis is somewhat dependent upon this being accurate.	Medium	Not started
The terminal output is lengthy and none of it is actually used by any follow-on program. FIPD exclusively pulls the binary interface files in their work and it would seem useful to actually have the terminal output produce pin depletion details summarized by assembly.	Medium	Not started
RCT uses the REBUS restart file RFILES to avoid requiring the depletion chain details as input and the mapping between ISOTXS isotopes to depleting isotopes. The entire use is focused on EBR-II modeling almost making it exclusively limited to this reactor.	Low	Not started

Table 2.19. NUBOW-3D Software Documentation Status

Description	Importance	Status
There is a non-distributable user manual for NUBOW-3D and a supplemental manual. A new manual needs to be created that is consistent with the updated input interface.	High	Not Started
There is no user guide.	Low	Not started
There is no software design report.	Low	Not started
There is no software requirement report.	High	Not started
There is no software verification report.	High	Not started
The code coverage report is not present.	High	Not started

Table 2.20. Outstanding NUBOW-3D Software Updates

Description	Importance	Status
NUBOW-3D needs to support OneAPI compilation where the present production executables are created with the 2015 or 2019 Intel compilers.	High	Not Started
The Fortran 66 features should be updated or replaced.	High	In Progress
Requests have been made to store the NUBOW-3D results such that they can be use as initial conditions for a follow-on analysis (i.e. cycle to cycle behavior).	Medium	Not Started

3. Industry Specific Developments

As the ARC software is being used by industry partners, there have been funded tasks from those industry partners to accomplish specific capabilities of the ARC software and PyARC. The details in this section covers work that was done using that funding along with funding from the current ARC SQA project funded by ART to support the QA of that work.

3.1.1 DASSH upgrades.

The Fortran version of DASSH [14] was quickly put together as the python software developer of DASSH [13] was leaving the lab. Because of the short development time, the Fortran version was made inferior to the python version with respect to its geometry handling capability along with some other features that were simply not included. The geometry aspect was resolved this year by allowing the user to specify different pin geometry (or porous body medium) models for each axial segment of each assembly. In Figure 3.1, the pin temperature profile is shown for each axial plane in a test problem with seven total hexagonal assemblies and 9 axial planes in the DIF3D model. As seen, the pin geometry model is varied from a single pin geometry up to a maximum of 6 rings of pins.

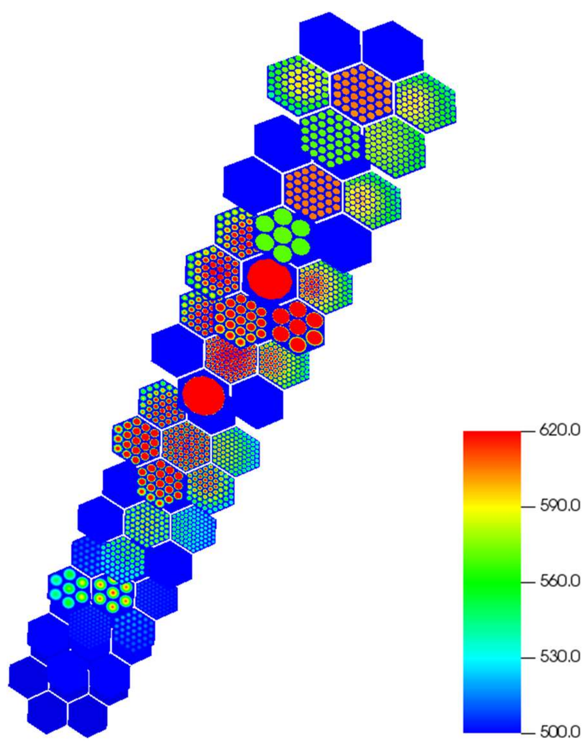

Figure 3.1. Example DASSH Axial Geometry Variation Possible in the New Version.

To allow this, a new routine had to be added to DASSH to do radial temperature interpolation. Interpolation schemes are setup between each axial plane for the temperature distribution in four regions: 1) coolant internal to the inner duct, 2) bypass gap coolant between ducts, 3) inner duct wall, and 4) outer duct wall. If no interpolation is needed because the pin geometry is the same, then a direct copy of the temperature solution is simply made. For single ducted assemblies, only two of these interpolation schemes are needed.

A second order basis is used for the interpolation to preserve the temperature profile and the average temperature for each region between planes. For flowing coolant regions, the subchannel temperature is weighted with the subchannel flow rate noting that the flow rate for each subchannel is known given the pin geometry and evaluated flow split correlation. For single pin models, the second order interpolation must be reduced to 1st order to avoid erroneous solutions. Figure 3.2 shows the effect of the radial interpolation near the 140 cm axial plane in this test case. The left hand picture shows the pin temperatures while the right hand picture shows the subchannel coolant temperature. Several of the assemblies were transitioned from porous body medium regions (i.e. no pins) to regions with pins at 140 cm. The sub-channel coolant temperature on the right shows how the solution coarsens or refines when the pin geometry is increased or decreased between the axial planes.

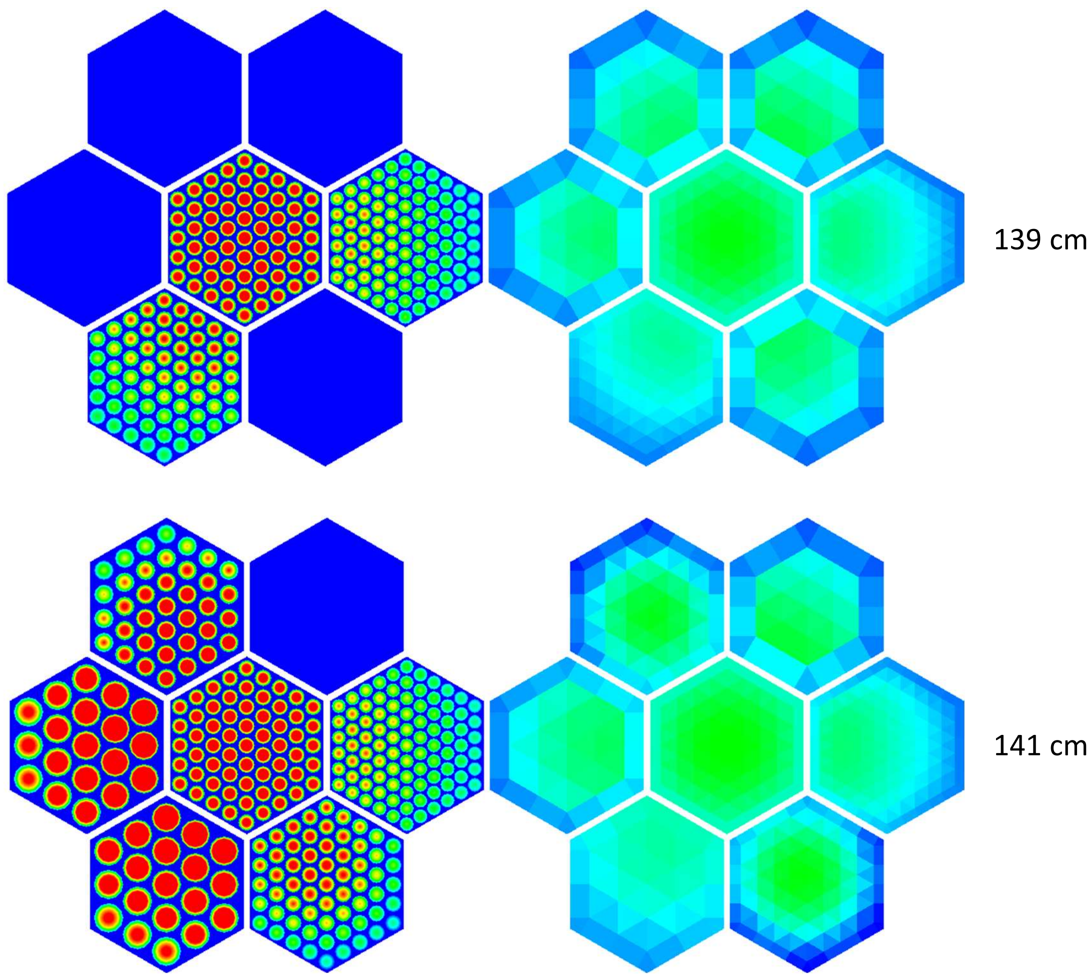

Figure 3.2. Example DASSH-F Temperature Profile Interpolation at Axial Boundary.

As seen, the shape of the temperature distribution is mostly preserved, but there are notable changes in the temperature distribution that indicate the coolant becomes colder along the duct wall surfaces in the upper axial plane relative to the lower axial plane temperature solution. This can be most clearly seen in the lower right assembly. This behavior an artifact of the quadratic fitting and the changing subchannel flow rates near the duct wall border between the two axial regions as the axial geometry changes. Correction of this behavior is a focus of future work, noting that this is not the observed behavior for real problems and is only observable here because of the characteristics of the problem being solved.

3.1.2 NUBOW-3D Input and Output Modifications.

The TCF project [26] with Westinghouse and ART fast reactor program [38] supported making changes and improvements to NUBOW-3D and work progressed on creating a user-friendly experience running analysis. The major improvements included 1) input file consolidation and reorganization to allow readability, removal of redundant and unnecessary inputs, and integration with PyARC, 2) output file consolidation and reorganization to allow for visualization of results and remove unnecessary outputs, and 3) general code cleanup and bug fixes to support the aforementioned improvements.

The input reorganization focused on reformatting the input file structure to consolidate the necessary inputs into the fewest number of required parameters from the user, to simplify the burden on the user and remove redundant defined values that can cause runtime issues or erroneous analysis results if not changed at the same time. The other main consideration was reducing the number of input files as low as possible, by putting all the main inputs required for defining the mesh geometry, defining the contact points and behavior, and setting simulation runs and convergence criteria into one file, instead of four. An example of the new input structure is shown with a portion of one of the benchmark inputs tests in Figure 3.3. The original input file structure used the NAMELIST feature to define groups of variable inputs using ambiguous internal variable names, while the new structure relies on keyword input syntax with more descriptive naming of parameters for users. Originally, multiple sets of input axial meshes were required to be defined for geometry input definition, temperature input assignment, fast flux input assignment, and strain step solution calculation. The geometry and strain axial mesh inputs were unified into a single input, AXIAL_MESH, while the temperature and fast flux assignment meshes are now included as headers in INPUT_TEMPS and INPUT_FASTFLUX files, respectively (which are outputs provided by DASSH).

```
INPUT_NAME       ABTR_Tutorial_Sample21
INPUT_GEOM       nubow_adjacency.inp
TIME_DAYS        0.0              240.0            480.0
INPUT_TEMPS      nubow_temp_0.inp nubow_temp_1.inp nubow_temp_2.inp
INPUT_FASTFLUX   nubow_flux_0.inp nubow_flux_1.inp nubow_flux_2.inp

ASSEMBLY_PITCH  5.7472e+00
FLUX_USAGE       3
PREFIX_VTK       viz

FAB_TEMP         4.4033e+02 ! The fabrication temperature (F)
INLET_ZERO       6.6200e+02 ! The inlet temperature (F) at hot zero power
INLET_TEMP       6.6200e+02 ! The inlet temperature (F) at nominal operation
GRID_TEMP        6.6200e+02 ! The grid temperature (F) at nominal operation
INLET_BREAK      6.6200e+02 ! The grid material type
MEAN_DUCT_TEMP   6.6200e+02 ! The grid plate stiffness
GRID_MAT         8          ! The mean duct temperature (F)
GRID_STIFFNESS   1.0000e+05 ! The inlet temperature (F) at full flow break point
BREAK_POINT      1.0        ! The full flow break point
PRESSURE_INLET   5.7000e+01 ! The pressure (psi) at the inlet
PRESSURE_OUTLET 2.9008e+01 ! The pressure (psi) at the upper most load point
COMPL_PRESSURE   2.0000e-02 ! The compliance value for duct dialiation due to pressure
COMPL_CONTACT    6.0000e-02 ! The compliance value for duct dialiation due to contact forces

AXIAL_MESH 0.0000e+00 1.0000e+01 1.4252e+01 1.8504e+01 2.2756e+01 2.7008e+01 3.1260e+01 3.5512e+01
AXIAL_MESH 3.9764e+01 4.4016e+01 4.8268e+01 5.2520e+01 5.6772e+01 6.1024e+01 6.2992e+01 6.4961e+01
AXIAL_MESH 6.8898e+01 7.2835e+01 7.6772e+01 8.0709e+01 8.4646e+01 8.8583e+01 9.0551e+01 9.2520e+01
AXIAL_MESH 9.7441e+01 1.0236e+02

EVAL_FORCE 0.0000e+00 2.5263e+01 5.0526e+01 7.5789e+01 1.0105e+02 1.2632e+02 1.5158e+02 1.7684e+02
EVAL_FORCE 2.0211e+02 2.2737e+02 2.5263e+02 2.7789e+02 3.0316e+02 3.2842e+02 3.5368e+02 3.7895e+02
EVAL_FORCE 4.0421e+02 4.2947e+02 4.5474e+02 4.8000e+02
```

Figure 3.3. Example of NUBOW-3D New Structured Input.

In addition to the simplification of the user-defined inputs, NUBOW-3D was officially updated to support directly importing temperature, fast flux, and duct assembly adjacency information from DASSH, which also required some work in DASSH to output the required formatted files. All of the existing software verification tests were updated and reformatted to follow the new input structure to be used for testing modifications to NUBOW-3D. A software verification report detailing these tests will be prepared after the manual and requirements report are completed.

The output reorganization focused on reformatting and reorganizing the output files to reduce the number of files created and present them in a way that is more readable to a general user. For visualization, the solution data and geometry grid were cast into a finite element framework via the VTK file format. Originally, the outputs were formatted in a way that would be useful for a developer, who was familiar with the source code for the output subroutines, to use personalized scripts/functions in either Matlab or Python to create specific plots or manage the data for analysis of deformation or contact force results. These scripts were not distributed with executables of NUBOW-3D. To simplify that, the PREFIX_VTK input has NUBOW-3D create the VTK files necessary for visualizing the deformation or load pad contact force results. The deformation values in X and Y directions for all ducts are compiled, the radial values are calculated with positive values indicating outward deformation (expansion) and negative values indicating inward deformation (compaction). An example plot of the radial deformation magnitude is shown in Figure 3.4 where the ducts are distorted (bowed) in a greatly exaggerated manner for visualization purposes.

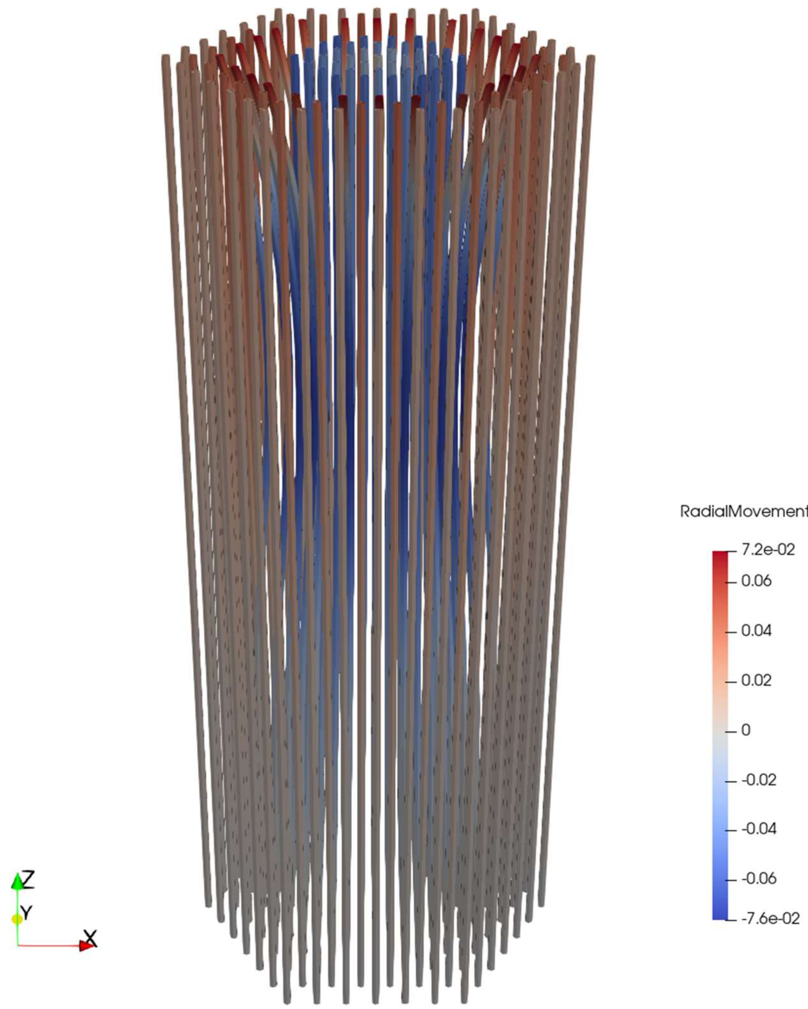

Figure 3.4. Example of NUBOW-3D Radial Displacement Visualization in the Axial View.

In addition to the deformation, the contact forces at the load pads (and nozzle contact points) are collected for each duct face and saved into another VTK file. Figure 3.5 shows an example of a clipped portion of the contact force plots at the Above Core Load Pad (ACLP) and the Top Load Pad (TLP) for an example problem. These plots allow a user to quickly visualize the contact behavior across the core to qualitatively determine the applicability of the existing core restraint system design and to see if any quick changes could be made before detailed analysis of the results.

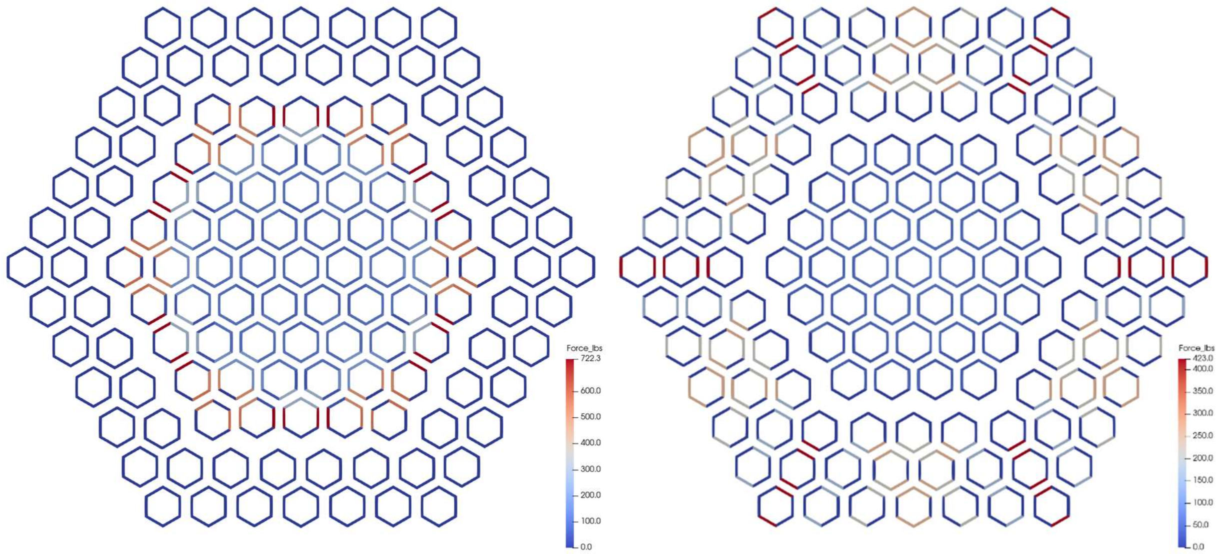

Figure 3.5. Example of NUBOW-3D Visualization of Load Pad Contact Force Values at the ACLP (left) and TLP (right) at Each Duct Face.

To support the above improvements, new subroutines and functions were added to import and organize the inputs following the new structure. Existing subroutines were also cleaned up and updated from Fortran 77 (or Fortran 66) syntax to support Fortran 90 free format syntax for compilation consistency across platforms and remove unsupported and undesirable features in previous Fortran version, such as Assigned GOTO, Computed GO TO, and Arithmetic IF statements. Much of the required inputs are now internally error-checked to ensure the variables are properly, and reasonably defined, with upper and lower limits defined based on the known or assumed geometry. The subroutines relating to the post-processed calculation of the reactivity change due to bowing were removed, due to potential inadequacy of the existing methodology and unclear source of the required inputs, in favor of formatting the deflection values and passing them to PERSENT for the reactivity change calculation.

Bug fixes were also made to eliminate issues caused by previous upgrades to the code over the years, and minor improvements to some of the existing subroutines. One bug was found in the STRAIN step calculation due to a previously unknown redundant input (defined in multiple input files originally) which caused some variables to remain undefined, resulting in issues of undefined or zero value variables being used in calculations during the irradiation strain steps. This bug was fixed by forcing the values to be defined when necessary for strain step calculation, or skipping the calculation if they are not required to be defined. It was also discovered that the Windows and Linux compiler versions caused discrepancies in the irradiation creep and swelling strain results because a different behavior of 32-bit and 64-bit executables. This was corrected and tested to ensure large variations in the results were not occurring between the different compiled versions on Windows and Linux.

3.1.3 PyARC modifications to include DASSH and NUBOW-3D

Through joint funding from the ART fast reactor program [38] and a Technology Commercialization Fund (project 21-24979) [26], PyARC was upgraded to replace the original support for the python version of DASSH with the Fortran version. NUBOW-3D was also integrated to support the DASSH-to-NUBOW-3D workflow. Both of these developments are described in detail in [50] and only a brief description is included here. Both of these new upgrades are available in the v3.0.0 release of PyARC.

Following the decision to support the commercialization of the Fortran version of DASSH rather than the python version, PyARC was updated to support input generation, execution, and post processing of the DASSH software. Through this upgrade, there were major changes to the required user-input, geometry limitations, and available output. It is important to note that this PyARC upgrade was completed prior to the DASSH upgrade for looser geometric restrictions noted earlier. As such, the DASSH integration was modified for more robust geometry checking and automatic geometry modification (where possible) in the preprocessing stage to support the limited geometric scope of the earlier version. As a part of this upgrade, the user-input requirements for DASSH in PyARC were significantly simplified to reduce user-burden by automatically inferring the geometric definition as needed by DASSH from the PyARC geometry input. Due to the nature of the analysis performed by DASSH and the geometric constraints, not all geometry information can be automatically translated requiring some additional details to be provided by the user. However, the new input requirements are far more simplified and reduce opportunities for human error in the input, which is a core capability that PyARC seeks to provide. Lastly, the available post-processed output was greatly reduced in this new upgrade as well due to time restrictions for this developmental work and the fact that there was no pressing need for such output to be made immediately available through PyARC.

Following the aforementioned upgrade for DASSH in PyARC, the NUBOW-3D code was integrated as a new feature to support core bowing analysis. This connection was integrated such that as much information as possible is reused from the DASSH input to generate the necessary NUBOW-3D input. The DASSH integration was updated to support the DASSH generated input for NUBOW-3D. This integration was completed for preprocessing input and executing the code, but the postprocessing stage for this integration has not yet been completed as it depends on the output from NUBOW-3D which has not yet been finalized.

4. Conclusions

The ARC software quality assurance program was stopped in ~2005 due to a lack of funding and thus staffing to fulfill the roles. The ARC software quality assurance program was restarted in fiscal year 2023 and as seen from this report, the progress achieved is strongly industry focused. Considerable work was done this year to create missing manuals and carry out commercial grade dedication software verification work that can be used by industry.

Some detail was also provided on adding capabilities to the ARC software this year. In particular, DASSH was modified to allow general axial geometry specifications and improvements in the porous body medium were made. The NUBOW-3D software was greatly improved by fixing bugs in the source code and creating a formal input that is user friendly. The NUBOW-3D output was also updated to make more user readable output along with visualization capabilities via the VTK file output options.

The focus on resolving the QA gaps in the ARC software cited in this report will continue in the coming year as more missing documentation is created. At present only a fraction of the originally missing documentation is still missing. One aspect to resolve next year is the software verification work for DASSH along with the software verification work and code coverage report for SE2RCT. Almost all of the ARC software was successfully transitioned to using the OneAPI compiler this year with the exception being NUBOW-3D and SE2RCT which cannot be transitioned until up to date manuals and a code coverage report are prepared. These tasks are obviously the priority for the coming year. Additional work will be focused on software portability issues and it is unclear how long this will take to fully complete as there are many difference pieces of software in ARC that were written by different people using different programming standards. In the coming years, the remaining QA gaps identified in this report will be addressed which should yield an ARC software product that can better facilitate the commercialization needs of industry.

5. Acknowledgements

This work was funded by U.S. Department of Energy via the Office of Nuclear Energy's Advanced Reactor Technologies (ART) Fast Reactor Program (FRP). Argonne National Laboratory's work was supported by the U.S. Department of Energy under contracts DE-AC02-06CH11357 and DOE DENE0009054.

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," Nucl. Sci. Eng., 187, 268 (2017).
3. C. H. Lee and W. S. Yang, "Development of Multigroup Cross Section Generation Code MC2-3 for Fast Reactor Analysis," Proc. of Int. Conf. on Fast Reactors and Related Fuel Cycles (FR09), Kyoto, Japan, Dec. 7-11 (2009).
4. C. H. Lee, H. Park, and Z. Zhong, "Recent Updates in ETOE2 and MC2-3," ANL/NSE-21/75, Argonne National Laboratory (2021).
5. H. Park, W. S. Yang, and C. H. Lee, "User Manual for MC2-3 Gamma Library Generation," ANL/NSE-18/12 Rev.1, Argonne National Laboratory (2019).
6. 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).
7. M. A. Smith, E. E. Lewis, E. R. Shemon, "DIF3D-VARIANT 12.0, Updates and New Features," ANL/NE-14/1 Rev. 1, Argonne National Laboratory (2025).
8. B. J. Toppel, "A User's Guide to the REBUS-3 Fuel Cycle Analysis Capability," ANL-83-2, Argonne National Laboratory (1983).
9. M. A. Smith, C. Adams, W. S. Yang, and E. E. Lewis, "VARI3D & PERSENT: Perturbation and Sensitivity Analysis," ANL/NE-13/8 Rev. 5, Argonne National Laboratory (2024).
10. M. A. Smith, C. H. Lee, and R. N. Hill, "GAMSOR: Gamma Source Preparation and DIF3D Flux Solution, Rev. 2.0," ANL/NE-16/50, Rev. 2, Argonne National Laboratory (2022).
11. W. S. Yang and M. A. Smith, "RCT: REBUS Based Pin Power Reconstruction Using the DIF3D-Nodal and DIF3D-VARIANT Options, ANL/NE-14/15, Argonne National Laboratory (2014).
12. K. L. Basehore, N. E. Todreas, "SUPERENERGY-2: A Multi-Assembly Steady-State Computer Code for LMFBR Core Thermal-Hydraulic Analysis," PNL-3379, Pacific Northwest National Laboratory (1980).
13. Milos Atz, Micheal A. Smith, Florent Heidet, "Ducted Assembly Steady State Heat Transfer Software (DASSH) - Theory Manual", ANL/NSE-21/33, Argonne National Laboratory, (2021).

14. M. A. Smith, “DASSH-F: Subchannel Based Thermal Analysis,” ANL/NSE-24/82, Argonne National Laboratory (2024).

15. J. Grudzinski, T. Moran, C. Grandy, "Supplement to the NUBOW-3D Manual," ANL/NE-15/9 Rev. 1, Argonne National Laboratory (2015).

16. Z. Zhong, M. A. Smith, “User Manual for the Thermal Hydraulic Code SE2RCT,” ANL/NSE-25/68, Argonne National Laboratory (2025).

17. Z. Zhong, M. A. Smith, “Theory and User Manual for the Thermal Hydraulic Code SE2-ANL,” ANL/NSE-25/44, Argonne National Laboratory (2025).

18. M. A. Smith, “EvaluateFlux, VARPOW, VARPEAK, DIF3DtoVTK Utility Programs for DIF3D,” ANL/NSE-22/55, Argonne National Laboratory (2022).

19. D. J. O’Grady, et. al., “The SAS4A/SASSYS-1 Version 5.7 Safety Analysis Code System,” ANL/NSE-SAS/5.7, Argonne National Laboratory, Lemont IL, June (2023).

20. A. M. Yacout, A. Oaks, W. Mohamed, K. Mo, “FIPD: EBR-II Fuels Irradiation & Physics Database,” ANL-ART-124, Argonne National Laboratory (2017).

21. “Integral Fast Reactor,” https://en.wikipedia.org/wiki/Integral_fast_reactor, retrieved (2023).

22. Prototype Generation-IV Sodium-cooled Fast Reactor, https://www.ans.org/news/article-1849/pgsfr-an-advanced-fuel-cycle-and-power-solution-for-korea/ retrieved (2023).

23. “Versatile Test Reactor,” https://gain.inl.gov/SitePages/VirtualTestReactor.aspx, retrieved July (2019).

24. “TerraPower Natrium,” https://www.terrapower.com/doe-natrium-demonstration-award, retrieved (2023).

25. “Advanced Reactor Demonstration Program Summary,” https://www.energy.gov/ne/advanced-reactor-demonstration-program, retrieved on September 2025.

26. N. Stauff and M. A. Smith, “Enhancement of PyARC for Westinghouse Electric Company’s Lead Fast Reactor Design,” TCF-21-24979, ANL/NSE-25/57, Argonne National Laboratory, (2025).

27. Nicolas E. Stauff, Taek K. Kim, Robert A. Lefebvre, Brandon R. Langley, Bradley T. Rearden, “Integration of the Argonne Reactor Computation codes into the NEAMS Workbench,” ANS Summer meeting, Philadelphia, PA, USA, June 17-21 (2018).

28. C. H. Lee, H. Park, and Z. Zhong, "Verification of the ENDF/B-VII.1 based MC2-3 Library," ANS/NSE-24/48 Rev.1, Argonne National Laboratory (2025).

29. Z. Zhong and C. H. Lee, “Improvement and Verification of the ENDF/B-VII.0 based MC2-3 Library,” ANL/NSE-23/43 Rev.3, Argonne National Laboratory (2024).

30. H. Park, W. S. Yang, M. Smith and C. H. Lee. “Verification and Validation Tests of the Gamma Library of the ARC Software Package (Rev.1),” ANL/NSE-19/22 Rev.1, Argonne National Laboratory (2022).

31. C. H. Lee and Z. Zhong, "Verification of the MC2-3 Software for Fast Reactor Analysis," ANL/NSE-22/36, Argonne National Laboratory (2022).

32. A. G. Nelson, M. A. Smith, “Verification of the DIF3D Software to Support Fast Reactor Analysis,” ANL/NSE-20/3 Rev. 3, Argonne National Laboratory (2024).

33. Z. Zhong, "Verification of the EvaluateFlux Utility Program," ANL/NSE-24/34 Rev. 1, Argonne National Laboratory (2024).
34. Z. Zhong, M. A. Smith, K. Ramey, "Gamma Source Verification for GAMSRC and GAMSOR," ANL/NSE-24/15 Rev. 1, Argonne National Laboratory (2024).
35. Z. Zhong, "Software Verification of VARPOW," ANL/NSE-24/33 Rev. 1, Argonne National Laboratory (2024).
36. M. A. Smith, A. G. Nelson, "Verification of the REBUS Software," ANL/NSE-25/39, Argonne National Laboratory (2025).
37. M. A. Smith, "Verification of the PERSENT Software," ANL/NSE-24/83 Rev. 1, Argonne National Laboratory (2025).
38. "Advanced Reactor Technologies," https://www.energy.gov/ne/advanced-reactor-technologies retrieved in September (2025).
39. Kalin Kiesling and M. A. Smith, "Software Quality Assurance Plan," ANL/NSE-23/13, Argonne National Laboratory (2023).
40. "GitLab," https://about.gitlab.com/ retrieved (2023).
41. "Git," https://git-scm.com/ retrieved (2023).
42. "Nuclear Energy Advanced Modeling and Simulation," https://neams.inl.gov/ retrieved (2023).
43. Zhaopeng Zhong, Micheal A. Smith, and Chang-ho Lee, "Code Coverage Status of ARC Code – DIF3D," ANL/NSE-23/65 Rev. 1, Argonne National Laboratory (2025).
44. M. A. Smith, "DIF3D-VARIANT Usage Guide," ANL/NSE-22/33, Argonne National Laboratory (2022).
45. M. A. Smith and Z. Zhong, "Code Coverage Status of the ARC Code REBUS," ANL/NSE-25/36, Argonne National Laboratory (2025).
46. K. R. Kiesling and M. A. Smith, "Code Coverage Status of the ARC Code PERSENT," ANL/NSE-25/50, Argonne National Laboratory (2025).
47. M. A. Smith, "Code Coverage Status of the ARC Code GAMSRC," ANL/NSE-25/49, Argonne National Laboratory (2025).
48. M. A. Smith and Z. Zhong, "Code Coverage Status of the ARC Code RCT," ANL/NSE-25/41, Argonne National Laboratory (2025).
49. M. A. Smith, "Code Coverage Status of the ARC Code DASSH-F," ANL/NSE-25/48, Argonne National Laboratory (2025).
50. K. R. Kiesling, "PyARC Status Report: New Integrations and Upgrades to the Fast Reactor Analysis Workflow Management Tool," ANL/NSE-25/62, Argonne National Laboratory (2025).

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

www.anl.gov

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