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 any of their employees, 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 authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. Reference herein to any social initiative (including but not limited to Diversity, Equity, and Inclusion (DEI); Community Benefits Plans (CBP); Justice 40; etc.) is made by the Author independent of any current requirement by the United States Government and does not constitute or imply endorsement, recommendation, or support by the United States Government or any agency thereof.

ANL/NSE-25/50

Code Coverage Status of the ARC Code PERSENT

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.

ANL/NSE-25/50

Code Coverage Status of the ARC Code PERSENT

prepared by

Kalin R. Kiesling and Micheal A. Smith
Nuclear Science and Engineering Division, Argonne National Laboratory

August 30, 2025

ABSTRACT

The Argonne Reactor Code (ARC) software system supports users in their fast reactor design goals by providing neutronic, thermal-hydraulic, and structural analysis capabilities. PERSENT fulfills the role of generating reactivity coefficients for a given time point of a REBUS calculation usable in a point kinetics based safety analysis capability. PERSENT also provides a sensitivity coefficient capability on eigenvalue, reactivity worth, and several other key coefficients that are used in the follow-on safety analysis. Given a co-variance matrix, PERSENT can carry out the uncertainty quantification to indicate the amount of error in the reactivity coefficients derived from the errors in the cross section measurements.

With continued improvement of computational resources, many of the geometry modeling capabilities in DIF3D that were primarily used in low order schemes are not really needed anymore. Today, the diffusion and transport capabilities of DIF3D-VARIANT are primarily used in the reactor design process with some scattered usage of DIF3D-FD and DIF3D-Nodal. PERSENT is part of the ARC code system and is built around DIF3D-VARIANT and the flux solution it provides.

The purpose of the present work is to identify a set of test problems for PERSENT and assess the code coverage of PERSENT for those test problems. PERSENT treats the DIF3D executable as an external executable and thus the code coverage considerations only need to focus on the PERSENT source code and only a fraction of the connected modules in the existing ARC software library. The goal is to document what parts of the existing PERSENT code are touched by the set of test problems and which are not. Because the verification work done on PERSENT was focused on the most common uses of PERSENT for fast reactor analysis, the code coverage assessment of those capabilities is the highest priority. This will ensure that nothing is being missed by the existing verification test problems that users of PERSENT rely upon.

The code coverage analysis of PERSENT was performed with the Code Coverage Tool of the Intel Fortran compiler which requires modifications to the compilation of PERSENT. The detailed coverage tables are given for each submodule of PERSENT. Most of the uncovered parts/files could be easily ignored because they are either for error message and debugging output or not needed by PERSENT today. Only a few uncovered parts of PERSENT deserve extending the verification test suite.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

The Argonne Reactor Code (ARC) software package consists of 13 primary and many secondary pieces of software that are connected through interface files. The ARC software suite supports users in their fast reactor design goals by providing neutronic, thermal-hydraulic, and structural analysis capabilities. Over the past few decades, the ARC software suite has been applied to numerous fast and thermal spectrum reactor analysis projects. For those cases with experimental measurements, ARC performed exceptionally well, yielding results of consistent accuracy to those produced by the Monte Carlo method at a fraction of the cost. At present, the ARC software suite is primarily used for the analysis of advanced reactors. The typical design process workflow for the fuel cycle analysis is shown in Figure 1.1, with the cited codes named in orange boxes, and as can be seen, REBUS [1]-[3] and DIF3D [1][4]-[9] are central to the entire ARC system.

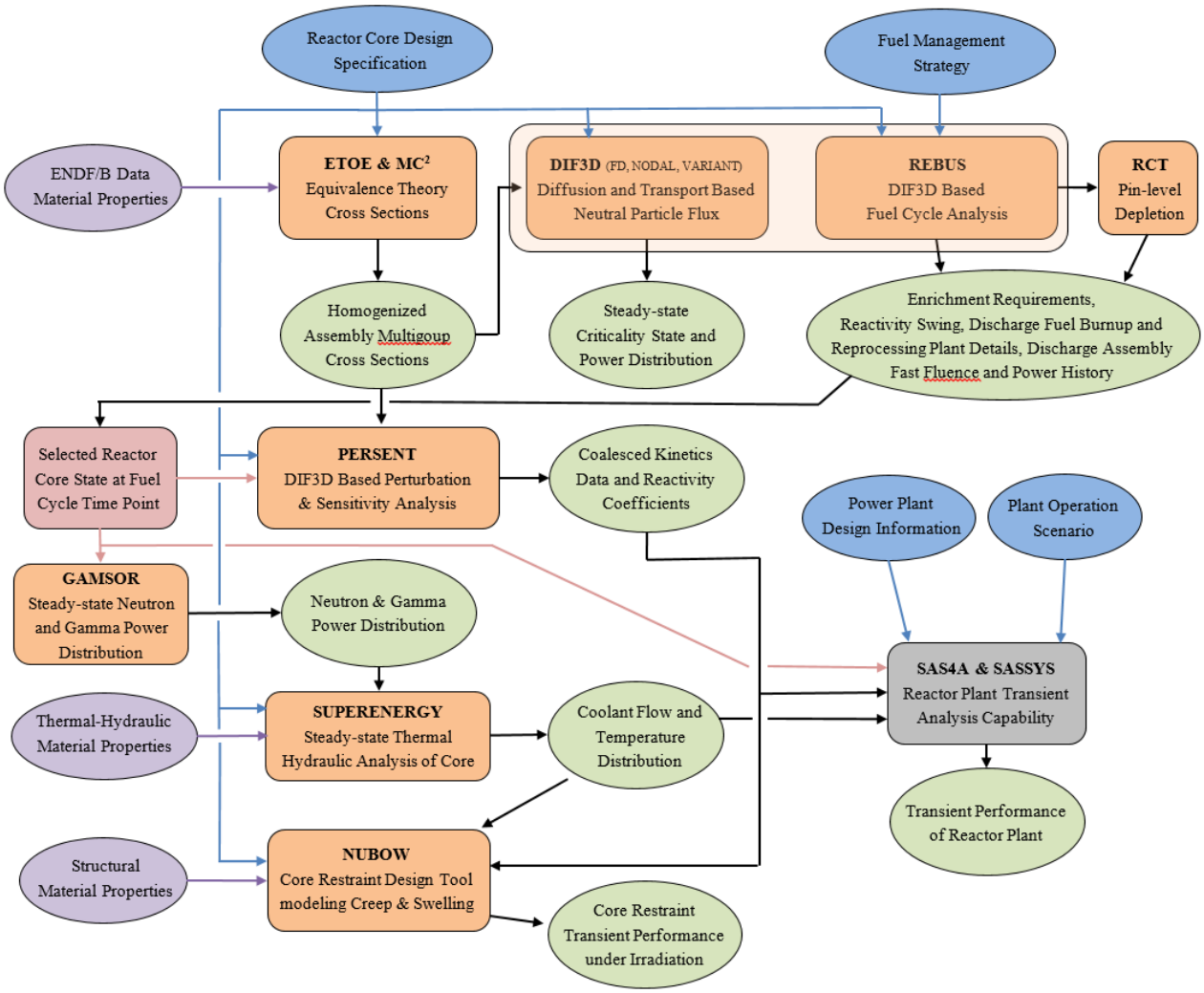

Figure 1.1 ARC Software Connections for Fast Spectrum Fuel Cycle Analysis

PERSENT [10] is used at each time point of a REBUS calculation to produce reactivity coefficients. Given a co-variance matrix, PERSENT can be used to generate cross section measurement related uncertainties for each reactivity coefficient based upon the linear sensitivity methodology. The PERSENT software requirements for anticipated commercial uses is detailed in [11] and the software verification work is detailed in [12]. This work is similar to that done for DIF3D in [13] and the code coverage assessment for PERSENT will follow that done for DIF3D in [14]. Also of relevance is the code coverage assessment of DASSH-F [15].

The ARC Software Quality Assurance (SQA) Program [16] is aimed to provide the controls and processes necessary to enable software improvement while meeting user and program sponsor requirements. The SQA Plan (SQAP) delineates the SQA Program framework for the ARC software by describing the Program activities, organization, and documentation, and by clearly defining the interconnection of all program items. The analysis presented in this report is part of the SQAP for each piece of the ARC software.

Much of the ARC software has its origins in the 1960s with many components added or upgraded over the following 60 years of continuous development. PERSENT is built around the VARIANT capability of the DIF3D code which was added in ~1994 and thus is much more modern having started its development in 2011. PERSENT is geometrically limited to the models that can be constructed with DIF3D-VARIANT. Today, the diffusion and transport capabilities of DIF3D-VARIANT are primarily used in the reactor design process with some scattered usage of DIF3D-FD and DIF3D-Nodal. Because DIF3D is treated as an external executable, much of the verification work of PERSENT is reliant upon the DIF3D executable [13].

The purpose of the present work is to identify a set of test problems for PERSENT and document the code coverage of PERSENT for those test problems. Because PERSENT is distributed with a regression test suite and many are included as part of the verification work [12], the set of test problems for the code coverage assessment will be the test suite. In the following sections, the parts of the existing PERSENT coding which are or are not touched by the set of test problems are documented. The code coverage analysis was performed using the Code Coverage Tool of the Intel Fortran compiler (gprof and codecov). To use this capability, the compilation of PERSENT must be modified to create a special PERSENT executable which is different from the production executable. The existing regression test suite will be used to verify that the new executable is correct, in addition to serving as the code coverage assessment test cases. A detailed code coverage report is generated by the Intel Code Coverage Tool, which shows the coverage of subroutines/functions, blocks within each subroutine/function, and the frequency of execution of each subroutine/function. With this code coverage report, it is easy to figure out which subroutines/functions are fully covered, fully uncovered, or partially covered in PERSENT. A review of the PERSENT source files that are not fully covered was performed and some indication is reported here for the lack of coverage and its importance for the PERSENT code usage. One unique behavior of the version of gprof used is that it does not provide call information for all subroutines within a file. In the tables that follow,

simple estimates are provided for those routines that were not detailed by gprof and are highlighted for clarity.

2. PERSENT Overview

PERSENT was developed by Argonne starting in 2011 and first released in 2013. The main motivation behind constructing PERSENT was to fix two outstanding problems: reactivity coefficients where transport was important and existing sensitivity analysis being done using R-Z transport. Any reactivity coefficient that involves changes in the leakage were known to have errors due to the use of diffusion theory. In the past, two-dimensional transport calculations (R-Z) were used to "correct" the 3D diffusion theory perturbation results. Most of the sensitivity analysis done before PERSENT was focused on using 2D R-Z transport calculations to generate the sensitivity coefficients. This is because relatively few 3D capabilities were available. For several reactivity coefficients, such as control rods or sample worths, using R-Z was known to introduce considerable errors in the sensitivity analysis work. The PERSENT code resolved these problems by building upon the DIF3D-VARIANT solver geometric capabilities.

At its core, DIF3D-VARIANT is a rigorous treatment of the diffusion equation. The ability of DIF3D-VARIANT to apply low order P_N transport to full core reactor problems in a reasonable amount of time, eliminates the need to do comparative R-Z transport calculations. At this time, PERSENT is one of the few deterministic 3D transport capability available today for generating sensitivity coefficents.

The present PERSENT capability can generate reactivity coefficients and the point kinetics constants β and Λ needed for a follow-on safety analysis modeling capability. PERSENT can also generate linear sensitivity coefficients for the eigenvalue and the various reactivity coefficients that it can generate. Finally, PERSENT can carry out an uncertainty quantification assessment of each reactivity coefficient given a co-variance matrix. This provides the user the uncertainty on each reactivity coefficient due to the cross section measurement errors. These three capabilities were verified in recent work [12].

3. PERSENT Coverage Assessment

While PERSENT treats the DIF3D executable as an external executable, this does not mean it is entirely independent of the DIF3D source code. All of the modules used in PERSENT are listed in Table 3.1 and it should be clear that it links to many parts of the DIF3D source code repository.

Table 3.1 Modules used by PERSENT

Modules	Directory locations
REBUS (driver)	source
Common use	dif3d/arccommon/source
Fortran 90 modules	dif3d/arcmodules/source
VARIANT basis	dif3d/variantbasis/source
DIF3D to VTK	dif3d/dif3dtovtk/source
LAPACK	dif3d/lapack/source
LINPACK	linpack/source

When PERSENT is compiled, many of the source files in each module (directory) are compiled and linked as part of the executable to avoid complicated compiling procedures. However, it should be noted that not all of the files in each directory are actually used by PERSENT. In particular, some subroutines are only called by DIF3D and are not usable in PERSENT. For LINPACK and LAPACK, they are generic linear algebra packages with which PERSENT only interacts a small fraction of. This code coverage analysis is thus an important reference if a streamlined code compilation is desired. PERSENT is distributed with 36 test cases with the first 19 created by the developers as part of the development and the last 17 created as part of the verification work [12].

3.1 Primary subroutines and driver of the PERSENT code

PERSENT is unique in the ARC system as the makefile generates two executables: persent.x and dif3d_ifs.x. The first is of course the executable of interest for commercial uses while dif3d_ifs.x is a special use executable that allows the user to solve the special inhomogeneous fixed source problems external to the main PERSENT path. To date, this extra executable has only been used by the developers of PERSENT. While there is a single test case in the PERSENT test suite for dif3d_ifs.x, it will yield full source code coverage and this aspect is deemed of little importance to users.

The driver subroutines of PERSENT are stored in the directory *source* with the bulk of the PERSENT source code stored in *source/library* and *source/structures*. The code coverage of the driver files is shown in PERSENT Source Code Coverage Details

This section contains the detailed tables indicating the coverage status of each subroutine in each source file within the PERSENT repository. Detailed assessments of these tables are in section 3.1. For each table, the bolded text in the “Uncovered Portion Description” indicates which parts need additional coverage.

Table 3.1 while the covered or partially covered subroutines in *source/library* are detailed in Table 3.2. For the uncovered or partially covered subroutines, a description of the uncovered parts is also given in the table. Also included is the number of calls by the test package showing the usage frequency of each source file. Table 3.3 details the uncovered subroutines in *source/library*. Table 3.4 provides the code coverage details of all files in the *source/structures* directory noting that the uncovered files only correspond to routines that are either not used or not reachable without passing through debug or fatal error branching in PERSENT.

3.1.1 Driver Routines

Starting with the driver routine coverage in PERSENT Source Code Coverage Details

This section contains the detailed tables indicating the coverage status of each subroutine in each source file within the PERSENT repository. Detailed assessments of these tables are in section 3.1. For each table, the bolded text in the "Uncovered Portion Description" indicates which parts need additional coverage.

Table 3.1, one can see that both the driver routines are partially covered. Since both driver routines have warning and error handling sections and those are the parts that are uncovered, further source code coverage is not needed. It should be noted that the dif3d_ifs.x executable is only used twice while persent.x is used 106 times.

3.1.2 General Missing Coverage

Continuing with the files in *source/library* and *source/structures*, all of the files have the file extension ".F90" which is omitted for brevity. There are several partially covered source code files in Table 3.2 and Table 3.4 and only the most important ones will be discussed here. Overall, the following features of PERSENT were identified to not be fully covered by the existing test suite:

1. Fatal error branching, various warning branches on user inputs or problem setup, and debugging output branching. These issues can be ignored for code coverage aspects.
2. DIF3D-VARIANT can export a flux vector of lower space-angle order than that used for solving the problem (e.g. exports 4^{th} order instead of 6^{th} order spatial basis). The operator in PERSENT is inherently inconsistent when this occurs as either the flux solution obtained is not consistent with the lower order operator or it is not defined for the higher order operator. This means that the reactivity or sensitivity coefficients in PERSENT will have a built-in error. To handle it, branches are inserted in PERSENT to use the array lengths associated with the NHFLUX file rather than the operator, and those extra branches are not being tested. They were also not verified but they are known to have worked in the past, and although there is no reason to believe they are not functioning today, an additional verification test is suggested to improve the code coverage. However, it is not required as it would invariably be a reduced accuracy result that was not verified.
3. The file wide χ treatment in DIF3D and χ matrix treatments were implemented in PERSENT, but none of the current tests cover those branches. Isotopic χ is the most accurate

approach for code to experiment calculations and it is entirely unwise and impractical to define a single χ usable for use in all enrichment regions of a given model. The PERSENT_Modify_DIF3D_Input subroutine prevents the user from being able to even reach those branches as it imposes all user inputs to use isotopic χ data. If that isotopic data is not present on ISOTXS, DIF3D will abort and thus PERSENT will abort. The χ matrix feature is not implemented in DIF3D and thus is not testable in PERSENT today. No additional code coverage is needed for either of these two aspects and it is recommended that both sections be eliminated from the PERSENT source code.

4. The PERSENT_M03 branching refers to the "NS_FIRST_ORDER" option for perturbation theory. The name is for non-standard first order approach for computing the perturbation worth. The manual clearly indicates that the user should use FIRST_ORDER_PT and the NS_FIRST_ORDER option is really intended for developers of PERSENT. No additional code coverage is needed for this feature.
5. The reaction rate and reaction rate ratio sensitivity capabilities do not test all reactions possible in ISOTXS. It is possible to construct a test case where all reactions are involved in a single calculation, but it is not clear what additional confidence this would give as it would simply touch all of the branches and could give an incorrect answer. In that regard, it is suggested that the verification work done in [12] be extended to include full checking of all cross section options in both reaction rate and reaction rate ratio. Given that work, the test problem can be defined and trusted to correctly test all of the source code.

The missing coverage for fatal errors, warnings, or debugging outputs are ignored for this code coverage assessment. Of these 5 features, only the 5th one is important as failure of the branches to work properly would lead to incorrect results for the user. However, this should be observable in the output as the displayed reaction rate from PERSENT would not be consistent with the one generated by the SUMMAR module of DIF3D (or via a hand calculation).

3.1.3 PERSENT_GeometryMaps

In Table 3.2, the first file of interest is PERSENT_GeometryMaps. As detailed, a branch associated with the NRASS=1 option of GEODST is not covered (NRASS meaning the regional assignment). When a user provides input to DIF3D, the default setup of GEODST will have NRASS=0. In fact, it is very difficult to get DIF3D to generate a GEODST file with NRASS=1. The basic requirement is that the user provide the zone assignment information for every axial mesh and that no adjacent axial assignment or axial mesh heights are identical in the domain. For clarity, the following example is provided:

> If the user defines axial plane boundaries of 10, 20, 30, 40, etc…, DIF3D will setup NRASS=0 if any assignment information between each consecutive planes is the same. It is very common for users to define an axial reflector and shield region with coarse dimensions (e.g. 0.0 to 65.0 and 65.0 to 120.0) and then using simple integer indicators (via card type 9 of A.NIP3) to further split these into axial mesh sizes between 8 and 20 cm. This input option of DIF3D allows the

user to define a large input with a very simple and compact input specification and DIF3D will generate GEODST with NRASS=0. To get a NRASS=1 outcome from DIF3D, the user must specify unique axial mesh heights assuming that the material to mesh assignment between axial planes is invariably going to be the same somewhere. This is not typical for real reactor models and while this section of code was tested during the PERSENT development, testing it today is of little importance.

3.1.4 PERSENT_IFS_Setup

The next subroutine of interest in Table 3.2 is PERSENT_IFS_Setup. This subroutine is only called in the dif3d_ifs.x executable and the missing code coverage aspects are not of great importance at this time. Although the dif3d_ifs.x capability was used during the early part of the PERSENT development and tested at that time, further testing and verification of the capability would be advisable if a commercial entity expresses interest in defining a workflow that uses it.

3.1.5 PERSENT_Input_Read

The PERSENT_Input_Read subroutine is one of the most important aspects to address in Table 3.2 and as can be seen, there are multiple branches that are not covered by the existing test suite. This subroutine handles all of the input processing of PERSENT. The DLAYXS_INPUT is used to point PERSENT to a file other than a DLAYXS located in the current working directory. Although this input option has been known to work, adding or updating an additional test case to use this flag would be minimal effort and advisable. The COMMARA_IGNOREMISSINGDATA input option is only used when the user provides an ISOTXS file that has more isotopes than the COMMARA file. This is generally a bad idea and PERSENT issues many error messages when there is missing COMMARA data and aborts with a fatal error. The flag is effectively there to allow the user to have PERSENT ignore these fatal errors and generate an answer even though the data is missing. Because most lumped fission product isotopes will not have co-variance data, almost every realistic PERSENT usage will encounter this problem and thus this option is known to work. Consequently, there is little motivation to add new testing for it although it would be advisable.

The MIFS_FILE is defunct and should be removed from the PERSENT source code as that feature of the PERSENT reaction rate sensitivity capability was removed. The PERSENT_S_SolveIFS subroutine has an associated orthogonalization branch that is also unreachable today as a result. The ISOTOPE_LIST is used in PERSENT for a variety of purposes and the most relevant case that lacks code coverage here is the reaction rate sensitivity option. The reason for the lack of code coverage is that the test suite only contains a single isotope in each ISOTOPE_LIST which is typical case for reaction rate ratio sensitivity options and the adjust_xs perturbation option (rarely used). In reaction rate sensitivity problems, there can be multiple regions and thus PERSENT will intersect the ISOTOPE_LIST and ZONE_LIST when computing the reaction rate of interest and thus having further code coverage would verify that logic.

The ADJUST_DENSITY input branch code coverage issue occurs because the existing test suite only applies the density change to a single zone. Because this perturbation option is not used by

anyone but the developers, no additional code coverage is needed. The NEW_ZONE input branch code coverage issue for resize is important, but difficult to carry out as the current implementation requires the user to provide over 100 zone perturbations. The uncovered branch is really just a test of the resize option of ASSIGNMENT_Define and clearly over the years this aspect has been tested through PERSENT as most realistic models have over 100 zones and the coolant density perturbation will impact all zones. As is the case with many other features of the ARC codes, specific testing of the *dif3d/arcmodules/source* need to be added and thus no additional source code coverage in PERSENT for this feature.

The lack of complete coverage of the connected models for ADJUST_XS, ADJUST_DENSITY, REACTION_RATE, REACTION_RATIO, and POWER_FRACTION are all due to the test suite only containing a single line of input for each of these options. The ADJUST_XS and ADJUST_DENSITY cases can be ignored as they are typically just developer test options. The REACTION_RATE input is again the only one of great interest and has been known to work in the past. A single test case that artificially touches these branches is advisable.

The missing coverage of the BOWING_WORTH input branch is of little importance. To trigger this section the user must provide the same line of input twice or two lines of input that point to the same PROBLEM_NAME. The source code would simply replace the previous results with whatever the latest inputs are and thus there are no real concerns that this would not work as expected. For the PLATE_MOVE_WORTH input, this was added as a developer option and it is very unlikely that any commercial user will make use of it. Given there is an available test problem, it is suggested to include that test case to improve the code coverage.

The USE_SHIFT option is by default set to YES. To disable it causes PERSENT to take considerably longer to converge the special inhomogeneous fixed source problems. It is typically only disabled when the developers are debugging a problem where convergence is not reached. In almost all cases the convergence issue has always been traced to insufficient convergence specified by the user in the forward and adjoint DIF3D flux solution problems. No additional coverage is suggested as it very unlikely the user will disable it and is primarily for developmental purposes.

The USE_TOTALNU flag is used to allow PERSENT to compute sensitivities to ν without having the DLAYXS files (USE_TOTALNU=YES). This output will not be consistent with the co-variance matrix and thus it is not wise to have PERSENT generate sensitivities for ν without being able to separate the prompt and delay aspects. It is advisable to include a test case that verifies the branching in the PERSENT code is tested adequately even though users are unlikely to use it.

The last uncovered aspect of PERSENT_Input_Read is a branch for GAMMA_FORWARD_FILE. This file can be provided by the user after using the dif3d_ifs.x executable to obtain it. The current single test case of dif3d_ifs.x only uses the GAMMA_ADJOINT_FILE input. Because this is unlikely to be used, no additional testing is recommended.

3.1.6 PERSENT_Modify_DIF3D

The uncovered branches in PERSENT_Modify_DIF3D_Input are of little importance. The one for "DATASET=A.SUMMAR" input is not very important as the branch for "UNFORM=A.SUMMAR" input is tested along with all of the aspects associated with it. In both cases, PERSENT is setup to simply copy the input as provided by the user. The "MODIFY" block input directive of DIF3D is rarely used today but has a similar conclusion to the preceding one as PERSENT will just assume whatever is given is correct. It should be noted that PERSENT traps many of the errors of an improperly setup DIF3D input after DIF3D is executed. The lack of code coverage on a missing HMG4C input is not relevant as PERSENT will simply write the one it wants to be there. Whatever the user provided is similarly replaced with what the PERSENT developers wants them to use. There are various other integer flags in the inputs that PERSENT will correct to what it wants and it should be no surprise that the inputs in the test suite are mostly setup to use what the PERSENT developers want the users to use. No additional testing of this subroutine is suggested at this time.

3.1.7 BowWorth

The partially covered subroutines including BowWorth in the name in Table 3.2 are associated with the bowing reactivity worth capability recently added to PERSENT. Only a few specific subroutines are important to discuss here. The PERSENT_P_BowWorth_DeformF missing code coverage deals with 2^{nd} order finite element information coming from the mechanics code. During the development, the quadratic feature was tested to ensure it worked, but the existing mechanics codes that can provide information to PERSENT only work with linear (VpE=2) elements and thus no additional code coverage is needed at this time. Similarly, the ability to run the bowing worth using a 2D geometry model was part of the original development work but is an unrealistic use case. There are several branches in the BowWorth group of subroutines that specifically deal with the 2D and quadratic FE options which can be ignored. The subroutines P_BowWorth_OddR and P_BowWorth_Op indicate branch testing of the DIF3D-VARIANT transport capability is missing. This occurs because the only test case in the test suite is for diffusion theory. However, the code-to-code comparison work detailed in the manual [10] clearly demonstrates that the transport and anisotropic scattering features work. It is recommended to add a test case that uses transport with at least P_3 anisotropic scattering to deal with the missing code coverage in the BowWorth group of subroutines. Because the plate worth input is missing, the best option would be to fabricate a Cartesian geometry case that uses transport.

3.1.8 PERSENT_S_Perturb_COMPXS

From Table 3.2, the code coverage summary of subroutine PERSENT_S_Perturb_COMPXS indicates that the n2n scattering reaction is not touched for the P_1 data branch. This is the expected result because the n2n scattering data does not have angular distribution information. The PERSENT_S_SolveIFS subroutine is not fully covered. This subroutine controls the inhomogeneous fixed source solver and to get full coverage would require setting up some difficult

problems as the various branches only get activated when convergence issues are observed. Clearly in the simple fast running test cases, those aspects do not occur and this is why no additional coverage is suggested at this point.

3.1.9 PERSENT_SetupCOMMARALink

The subroutine PERSENT_SetupCOMMARALink is used to map the ISOTXS isotope names to those found in the COMMARA co-variance data file. The missed branch occurs when the COMMARA dependent isotope name occurs before the isotope name in the COMMARA data file. This could have been tested with rearrangement of the data in the uncertainty quantification capability in [12]. It is possible that this could occur but there is no reason to believe the existing coding would not work as expected as the isotope name would simply appear later in the list of unique COMMARA isotope names.

3.1.10 Miscellaneous Subroutines in source/library

The remaining subroutines in Table 3.2 that are partially covered should fall under the five identified features of PERSENT listed in section 3.1.2. These features are not fully covered and only the last item listed needs additional coverage, but it inherently requires that the work be checked as part of the software verification.

Looking at the uncovered subroutines in the directory *source/library* detailed in Table 3.3, most of the subroutines are not used or reachable from the existing source code. Many of the unused subroutines should be removed from the repository at the earliest convenience.

3.1.11 Subroutines in source/structures

Continuing with the subroutines in the directory *source/structures* detailed in Table 3.4, most of the missing code coverage falls into the fatal error, warning messages, or debugging outputs. The only subroutine of interest to add code coverage to is PERSENT_Problem_Copy, which, similar to the NEW_ZONE input discussed earlier, requires the user to provide more than 100 lines of input to invoke. In this case it would be wiser to add unit testing as was found to be needed for the *arcmodules* source of DIF3D. The uncovered files in Table 3.4 are mostly associated with printing the data structures included in each module. These are reachable only if the user input is invalid. As a result, these routines are primarily used for development of PERSENT and additional code coverage is not needed.

3.2 PERSENT Source Code Coverage Details

This section contains the detailed tables indicating the coverage status of each subroutine in each source file within the PERSENT repository. Detailed assessments of these tables are in section 3.1. For each table, the bolded text in the "Uncovered Portion Description" indicates which parts need additional coverage.

Table 3.1 Covered or Partially Covered Driver PERSENT Files in the Directory *source*

File Name	Submodule or Function	# of calls by test package	Coverage Status	Uncovered Portion Description	Further Coverage Needed?
dif3d_ifs.F90	(MAIN)	2	Partially	Warning and/or error handling	No
persent.F90	(MAIN)	106	Partially	Warning and/or error handling	No
	PERSENT_SillyFunction	17	Fully		

Table 3.2 Covered or Partially Covered Driver PERSENT Files in the Directory *source/library*

File Name	Submodule or Function	# of calls test suite	Coverage Status	Uncovered Portion Description	Further Coverage?
PERSENT_CheckMaterials	PERSENT_CheckMaterials	53	Partially	Warning and/or error branches	No
PERSENT_ComputeUQ	PERSENT_ComputeUQ	19	Partially	warning and/or error branches, reformatting imaginary numbers	No
PERSENT_CopyISOTXS	PERSENT_CopyISOTXS	367	Partially	warning and/or error handling	No
PERSENT_ExportAsVTK	PERSENT_ExportAsVTK	38	Partially	warning and/or error branches, branch when denominator sum is 0	No
PERSENT_Fill_OddRemoval	PERSENT_Fill_OddRemoval	213	Partially	warning and/or error branches	No
PERSENT_GeometryMaps	PERSENT_GeometryMaps	53	Partially	NRASS=1 and print statement branches	No
PERSENT_IdentifyIsotopes	PERSENT_IdentifyIsotopes	52	Fully		No
PERSENT_IFS_Setup	PERSENT_IFS_Setup	2	Partially	forward gamma flux branch, diffusion theory branch, warning and/or error branches	No
PERSENT_Input_Read	PERSENT_Input_Read	106	Partially	warning and/or error branches. Inputs not tested:	Yes

				• **DLAYXS_INPUT** • COMMARA_IGNOREMISSINGDATA • MIFS_FILE • **PLATE_MOVE_WORTH** • **USE_SHIFT** • **USE_TOTALNU** • GAMMA_FORWARD_FILE Branches within inputs not reached • isotop_list already in list • adjust_density problem exists • new_zone resize • resize for adjust_xs, adjust_density, reaction_rate, reaction_ratio, and power_fraction • duplicate bowing_worth input	
PERSENT_Modify_DIF3D_Input	Modify_DIF3D_Input	126	Partially	warning and/or error branches Missing branches for • DATASET=A.SUMMAR, • MODIFY • HMG4C not present • Various input integer cases	No
PERSENT_Node_BalanceEdits	PERSENT_Node_BalanceEdits	77617	Fully		No
PERSENT_Node_EvenRemoval	PERSENT_Node_EvenRemoval	77617	Partially	Unreachable adjoint case	No
PERSENT_Node_Fission	PERSENT_Node_Fission	>1.1E6	Fully		No
PERSENT_Node_InnerProduct	PERSENT_Node_InnerProduct	74789	Fully		No
PERSENT_Node_Lambda	PERSENT_Node_Lambda	32343	Fully		No
PERSENT_Node_OddRemoval	PERSENT_Node_OddRemoval	77617	Partially	reduced dimension flux vector branching and debugging branches	No
PERSENT_Node_S_EvenRemoval	PERSENT_Node_S_EvenRemoval	74789	Partially	Unreachable adjoint branch	No
PERSENT_Node_S_Fission	PERSENT_Node_S_Fission	19789	Partially	Unreachable adjoint branch	No
PERSENT_Node_S_OddRemoval	PERSENT_Node_S_OddRemoval	74789	Partially	reduced dimension flux vector branching and debugging branches	No

PERSENT_P_AdjustDensity	PERSENT_P_AdjustDensity	8	Partially	print to file branch for sensitivity usage, branching for subzones in input, branch for user provided NHFLUX	No
PERSENT_P_AdjustProblem	PERSENT_P_AdjustProblem	9	Fully		No
	PERSENT_P_AdjustProblem_ TwoOperatorApproach	9	Partially	warning and/or error branches, make input only branch	No
PERSENT_P_AdjustXS	PERSENT_P_AdjustXS	10	Partially	warning and/or error branches, file chi and matrix chi branches, PERSENT_M03 branching	No
PERSENT_P_AdjustZone	PERSENT_P_AdjustZone	104	Partially	warning and/or error branches, print to file branch for sensitivity usage, branches for subzones in input and branching for PERSENT_M03.	No
PERSENT_P_BowWorth_BalEdit	PERSENT_P_BowWorth_BalEdit	39520	Partially	reduced dimension flux vector branching	No
PERSENT_P_BowWorth_CartGP	PERSENT_P_BowWorth_CartGP	5	Partially	warning and/or error branching, 2D geometry debug branch	No
PERSENT_P_BowWorth_DeformF	PERSENT_P_BowWorth_DeformF	60120	Partially	warning and/or error branches, 2D geometry debug branch, quadratic order finite element input	No
PERSENT_P_BowWorth_EvenR	PERSENT_P_BowWorth_EvenR	30060	Fully		No
PERSENT_P_BowWorth_Fission	PERSENT_P_BowWorth_Fission	39520	Partially	Aconst=0.0 branch where duct and bypass gap material are identical	No
PERSENT_P_BowWorth_HexGP	PERSENT_P_BowWorth_HexGP	6	Partially	warning and/or error branches, 2D geometry debug branch	No
PERSENT_P_BowWorth_OddR	PERSENT_P_BowWorth_OddR	30060	Partially	reduced dimension flux vector branching, Aconst<>0 branch where duct and bypass gap material are identical	No
PERSENT_P_BowWorth_Op	PERSENT_P_BowWorth_Op	11	Partially	warning and/or error branching, **non-diffusion branch**, file χ and χ matrix branching, cross section correction branching	Yes
PERSENT_P_BowWorth	PERSENT_P_BowWorth	11	Partially	warning and/or error branches, 2D geometry debug branch,	Yes

				NHSTRAIN coordinate adjustment, Cartesian PLATE_MOVE_WORTH	
PERSENT_P_CheckInput	PERSENT_P_CheckInput	53	Partially	warning and/or error branching, user convenience input branching	No
PERSENT_P_Currents	PERSENT_P_Currents	9	Partially	warning and/or error branches, unreachable base operator branch	No
PERSENT_P_LambdaBeta	PERSENT_P_LambdaBeta	17	Partially	warning and/or error branches	No
PERSENT_P_Operator	PERSENT_P_Operator	18	Partially	warning and/or error branches, file χ and χ matrix branching	No
PERSENT_P_PrintTable	PERSENT_P_PrintTable	298	Partially	Special print outputs that are not frequently used	No
PERSENT_ProcessISOTXS	PERSENT_ProcessISOTXS	106	Partially	warning and/or error branches and file χ	No
PERSENT_Product_Beta	PERSENT_Product_Beta	2930	Partially	warning and/or error branches	No
PERSENT_Product_Fission	PERSENT_Product_Fission	629	Partially	warning and/or error branches and file χ	No
PERSENT_Product_Lambda	PERSENT_Product_Lambda	22	Partially	warning and/or error branches and file χ	No
PERSENT_Product_Perturbation	PERSENT_Product_Perturbation	121	Partially	warning and/or error branches, first order perturbation operation for small user perturbation. file χ and χ matrix branching, PERSENT_M03 branching	No
PERSENT_RegionMaps	PERSENT_RegionMaps	62	Partially	warning and/or error branches	No
PERSENT_RunDIF3D	PERSENT_RunDIF3D	600	Partially	warning and/or error branches	No
	PERSENT_RunDIF3D_Ignore	748	Partially	warning and/or error branches	No
PERSENT_S_and_P	PERSENT_S_and_P	53	Partially	warning, error, and/or info print branches, unreachable sensitivity branch for no DIF3D execution	No
PERSENT_S_Beta	PERSENT_S_Beta	5	Partially	warning and/or error branches, print by family branch	No
PERSENT_S_CheckInput	PERSENT_S_CheckInput	106	Partially	warning and/or error branching	No
PERSENT_S_Diff	PERSENT_S_Diff	5	Partially	warning and/or error branching	No
PERSENT_S_Eigenvalue	PERSENT_S_Eigenvalue	41	Partially	warning and/or error branching	No
PERSENT_S_ExportFile	PERSENT_S_ExportFile	23	Partially	warning and/or error branching	No

PERSENT_S_GetReactionRateXS	PERSENT_S_GetReactionRateXS	3558	Partially	total/transport, nu, total, elastic, inelastic, n2n, scattering branching	No
PERSENT_S_ImportFile	PERSENT_S_ImportFile	5	Partially	warning and/or error branching	No
PERSENT_S_Lambda	PERSENT_S_Lambda	5	Partially	warning and/or error branching	No
PERSENT_S_LinearOperatorChange	PERSENT_S_LinearOperatorChange	112	Partially	warning and/or error branching, file χ and χ matrix branching	No
PERSENT_S_Perturb_COMPXS	PERSENT_S_Perturb_COMPXS	>24E6	Partially	P_1 n2n scattering branch for computing sensitivity vector contribution	No
PERSENT_S_PrintTable	PERSENT_S_PrintTable	1217	Fully		No
PERSENT_S_Product_Beta_im	PERSENT_S_Product_Beta_im	2808	Partially	warning and/or error branching	No
PERSENT_S_Product_Beta	PERSENT_S_Product_Beta	10	Partially	warning and/or error branching, file χ and χ matrix branching	No
PERSENT_S_Product_Lambda	PERSENT_S_Product_Lambda	10	Partially	warning and/or error branching, file χ and χ matrix branching	No
PERSENT_S_PromptNu	PERSENT_S_PromptNu	72	Fully		No
PERSENT_S_ReactionRate	PERSENT_S_ReactionRate	5	Partially	warning and/or error branching, branching for inputs with subzones, **branching for all selections of reaction rate**.	Yes
PERSENT_S_ReactionRateRatio	PERSENT_S_ReactionRateRatio	17	Partially	warning and/or error branching, branching for inputs with subzones, **branching for all selections of reaction rate**.	Yes
PERSENT_S_ReactivityWorth	PERSENT_S_ReactivityWorth	12	Partially	warning and/or error branching	No
PERSENT_S_RemoveFMC	PERSENT_S_RemoveFMC	601	Fully		No
PERSENT_S_SetupFixedS	PERSENT_S_SetupFixedS	32	Fully		No
PERSENT_S_SolveIFS	PERSENT_S_SolveIFS	86	Partially	Unreachable orthogonalization branch (MIFS), various branches for problems with convergence issues	No
PERSENT_S_SpecialChiNorm	PERSENT_S_SpecialChiNorm	71	Fully		No
PERSENT_SetupCOMMARALink	PERSENT_SetupCOMMARALink	2810778	Partially	warning and/or error branching, dependent isotope name occurs later in data file rather than earlier	No

PERSENT_VHGEOM	PERSENT_VHGEOM	54	Partially	warning, error, and/or info print branching, Triangular-Z geometry, branch for lower order Z	No
PERSENT_XS_CheckInput	XS_A_CheckInput	73	Partially	Error branching	No
	XS_CheckInput	13	Partially	Error branching	No
	XS_P_CheckInput	33	Partially	Error branching	No

Table 3.3 Uncovered Files in the Directory *source/library*

File Name	Submodule or Function	Description	Further Coverage?
PERSENT_CopyISOTXS	PERSENT_Print8bitwords	Prints error or warning message for different compilers and platforms associated with external DIF3D execution	No
PERSENT_Node_InvVelocity	PERSENT_Node_InvVelocity	Applies the 1/v * flux for DIF3D-VARIANT (**not used**)	No
PERSENT_P_AdjustProblem	PERSENT_P_AdjustProblem_ OneOperatorApproach	Alternative one operator approach for perturbation (**not used**)	No
PERSENT_P_BowWorth_OpB	PERSENT_P_BowWorth_OpB	Alternative to PERSENT_P_BowWorth_Op that does not work (**not used**)	No
PERSENT_P_BowWorth_OddRB	PERSENT_P_BowWorth_OddRB	Applies the odd-parity removal term for PERSENT_P_BowingWorth_OpB (**not used**)	No
PERSENT_Scale_OddRemoval	PERSENT_Scale_OddRemoval	Intended to be used in first order perturbation theory (**not used**)	No

Table 3.4 Coverage Status of Files in the Directory *source/structures*

File Name	Submodule or Function	# of calls test suite	Coverage Status	Uncovered Portion Description	Further Coverage?
PERSENT_CommonBlock	PERSENT_CommonBlock_Special	54	Fully		No
PERSENT_Problem	PERSENT_Problem_Copy	1566	Partially	Branching for target storage already defined	No
	PERSENT_Problem_Define	488	Partially	Unreachable branch and branching for resizing structure	No
	PERSENT_Problem_Define_Array	3620	Partially	Branching for resizing array and fatal error branches	No
	PERSENT_Problem_SetUnit	53	Fully		No
	PERSENT_Problem_Void	2349	Partially	Fatal error branching	No
	PERSENT_Problem_Print		**Uncovered**	Only reachable in debug branching or fatal error branching	No
	PERSENT_Problem_	129	Fully		No

	XS_IdentifyName				
PERSENT_ProblemSpecs	PERSENT_ProblemSpecs_Define	488	Partially	Fatal error branching	No
	PERSENT_ProblemSpecs_SetUnit		**Uncovered**	Used to control output path for module	No
	PERSENT_ProblemSpecs_Void	356	Partially	Fatal error branching	No
	PERSENT_ProblemSpecs_Copy		**Uncovered**	Not reachable	No
	PERSENT_ProblemSpecs_ IdentifyProblemName	620	Fully		No
	PERSENT_ProblemSpecs_Print		**Uncovered**	Only reachable in debug branching or fatal error branching	No
PERSENT_General_Input	PERSENT_General_Input_Print		**Uncovered**	Prints the data in the structure	No
	PERSENT_General_Input_SetUnit		**Uncovered**	Sets the output unit for the module	No

4. DIF3D Coverage Update

The preceding PERSENT subroutine coverage is the primary focus on this work. Because much of the remaining source code is linked to the DIF3D repo, some aspects of the DIF3D code coverage for those connected components is reiterated here [14].

4.1 Source code in the directory dif3d/arccommon/source

The directory *dif3d/arccommon/source* in the PERSENT distribution contains some common high-level functions used to replace large sections of the original DIF3D code. For PERSENT, only the QuickSort functions are used as seen below in **Error! Reference source not found.**. These subroutines are not used in DIF3D or REBUS and because they are library functions, the detailed code coverage is not given here as these functions need testing external to PERSENT (as all uncovered parts are not reachable with PERSENT).

Table 4.1 Code Coverage Details of Files in the Directory *dif3d/arccommon/source*

File Name	Function or Subroutine	# of calls test suite
Cholesky_dchdc_mod.f90	Cholesky_dchdc	0
	Cholesky_dchdc_mod	0
	Cholesky_dchdc_mod_GetX	0
Convert_COMPXS_ISOTXS.F90	Convert_COMPXS_ISOTXS	0
HMG4C_Response_Zone.F90	HMG4C_Response_Zone	0
HMG4C_XS.F90	HMG4C_XS	0
HMG4C_XS_Full.F90	HMG4C_XS_Full	0
HMG4C_XS_Zone.F90	HMG4C_XS_Zone	0
MCNP_Export_mgACE.F90	MCNP_Export_mgACD_laguer	0
	MCNP_Export_mgACD_zroots	0
	MCNP_Export_mgACE	0
	MCNP_Export_mgACE_PN	0
Proliferate_Zones.F90	Proliferate_Zones	0
QuickSort_R64_MD.F90	QuickSort_R64_MD_Driver	9532
	QuickSort_R64_MD_Segment	9518
	QuickSort_R64_MD_SetUnit	0
	QuickSort_Sort_R64_MD	14
	QuickSort_Sort_R64_MD_NI	0
VARPOW_Generate.F90	VARPOW_Generate	0

4.2 Source code in the directory dif3d/arcmodules/source

The source code in the directory *dif3d/arcmodules/source* of the PERSENT distribution primarily provide self-contained memory allocation and reading/writing of the binary interface files created in the ARC suite of codes. Because the approach taken to create them is somewhat consistent with modern object coding, each file can contain both the data structure and many subroutines/functions that operate on that data structure. The various modules and subroutines in the directory were developed to support many parts of the ARC software suite and very few of these are used in DIF3D or REBUS as they were developed after that software was created. Not all of the files in the directory are presently compiled as part of PERSENT but the code coverage summary of all files is provided here in Table 4.2 as if PERSENT owns them. The number of calls by the DASSH-F test suite is included here [15] as it also heavily uses the source code in this directory.

From Table 4.2, one can see that the components ALIAS, ASSIGNMENT, Basic, COMMARA, COMPXS, DIF3D, DLAYXS, GEODST, ISOTXS, LABELS, MATERIALS, NDXSRF, NE_FreeForm, NE_Kind, NE_VTKexport, NHFLUX, NHSTRAING, NonZeroStorage and ZNATDN are all used in PERSENT. There are several other components in this directory that are compiled and usable by PERSENT and some not usable because they simply appear in other parts of the source code (DIF3D, DASSH, etc. …). For DASSH-F, GNU monitor tool (gprof) was used to assess the number of subroutine calls, but there were some cases where this failed to report the number of calls to a subroutine and thus some values are estimates. For PERSENT, the actual subroutine call numbers were obtained using Intel's codecov tool to avoid this missing information.

With respect to code coverage, these are common use modules, and it is difficult to argue that PERSENT or DASSH-F are fully responsible for the code coverage on them. A quick survey of the missing code coverage parts shows that the bulk of them are fatal error, debug output, and invalid input branches which would not be active for test cases that are expected to work. There is no real lack of code coverage concerns with these modules for PERSENT, noting that the various sub-zone branching aspects not covered in DASSH-F are fully covered by the PERSENT inputs.

All other code coverage aspects on this source coding is effectively deemed of little importance to the functionality of PERSENT. It is recommended that a set of unit tests be fabricated for the arcmodules source code to completely negate the importance of code coverage for PERSENT and other similarly setup codes.

Table 4.2 Code Coverage Details of Files in the Directory *dif3d/arcmodules/source* Relative to DASSH

File Name	Subroutine	# calls PERSENT	# calls by DASSH	Coverage Assessment	Further Coverage
ALIAS.F90	ALIAS_Define	87		Missing fatal error branches	No
	ALIAS_IdentifyDestination				
	ALIAS_IdentifyTarget	18		Fully covered	No
	ALIAS_Print				
	ALIAS_SetUnit				
	ALIAS_Void	144		Missing fatal error branches	No
Append_DLAYXS_ISOTXS.F90	Append_DLAYXS_ISOTXS				
ARC_Append_NDXSRF.F90	ARC_Append_NDXSRF				
ARC_Build_MCNP.F90	ARC_Build_MCNP				
	ARC_Build_MCNP DeformDoNotUse				
ARC_CopyFile.F90	ARC_CopyFile	196		Missing fatal error branches	No
ARC_EasyAtomDensity.F90	ARC_EasyAtomDensity	100		Fully covered	No
ARC_EasyDLAYXS.F90	ARC_EasyDLAYXS	17		Fully covered	No
ARC_EasyISOTXS.F90	ARC_EasyISOTXS	72		Chi matrix and odd ISOTXS setup	No
ARC_ExportASCII.F90	ARC_ExportASCII				
	ARC_ExportASCII_type07				
	ARC_ExportASCII_type14				
ARC_SystemCommand.F90	ARC_Print8bitwords				
ARC_Simplify_NDXSRF.F90	ARC_Simplify_NDXSRF				
ARC_SystemCommand.F90	ARC_SystemCommand				
ASSIGNMENT.F90	ASSIGNMENT_Copy				
	ASSIGNMENT_Define	137	6	Missing fatal error branches	No
	ASSIGNMENT_IdentifyMaterial	492	932	Fully covered	No
	ASSIGNMENT_Print				
	ASSIGNMENT_SetUnit			Fully covered	No
	ASSIGNMENT_Void	184	289	Missing fatal error branches	No
Basic_Abort.F90	Basic_Abort				
Basic_AbortInvalidInt.F90	Basic_AbortInvalidInt				
Basic_AbortWithMessage.F90	Basic_AbortWithMessage				
Basic_CheckError.F90	Basic_CheckError		18,752		
ChannelInput.F90	ChannelInput_AppendRegion				
	ChannelInput_Copy				
	ChannelInput_Define				

	ChannelInput_Print				
	ChannelInput_SetUnit				
	ChannelInput_Void				
COMMARA.F90	COMMARA_AssignPrintInfo				
	COMMARA_Copy				
	COMMARA_Define	10		Missing fatal error branches	No
COMPXS_IMPORT.F90	COMMARA_Import	5		Missing fatal error branches	No
COMMARA.F90	COMMARA_Print				
	COMMARA_Void	10		Missing fatal error branches	No
COMPXS.F90	COMPXS_ADDSCALE	24		Missing chi matrix branching	No
	COMPXS_ASSIGNPRINTINFO	54		Fully covered	No
	COMPXS_BALANCEXS				
	COMPXS_COPY	42		Previously defined structure branching	No
	COMPXS_DEFINE	237		Missing fatal error branches	No
COMPXS_EXPORT.F90	COMPXS_EXPORT	42		Missing fatal error branches, file chi branch	
COMPXS.F90	COMPXS_FullShift_For_IHS				
COMPXS.F90	COMPXS_IMPORT	195		Missing fatal error branches, file chi branch	
COMPXS_Mass.F90	COMPXS_Mass				
COMPXS_Mass_by_HABSID.F90	COMPXS_Mass_by_HABSID	2554		Poorly defined input branch	No
COMPXS_Modify_For_IHS.F90	COMPXS_Modify_For_IHS	42		File chi branch	No
	COMPXS_Modify_For_IHS_fails				
COMPXS.F90	COMPXS_PRINT				
COMPXS_Modify_For_IHS.F90	COMPXS_Test_GS_solve				
COMPXS.F90	COMPXS_UPDATEPROPERTIES	42		Missing chi matrix branch	No
	COMPXS_VOID	246		Missing fatal error branches	No
	COMPXS_ZERO	237		Missing branch when data includes delay data	No
DIF3D.F90	DIF3D_ALLOCATE	603		Missing fatal error branches	No
	DIF3D_ASSIGNPRINTINFO	53		Fully covered	No
	DIF3D_DEFINE			Not used	No
DIF3D_EXPORT.F90	DIF3D_EXPORT				
DIF3D.F90	DIF3D_FLUXBYPASS_OFF				
	DIF3D_FLUXBYPASS_ON				

DIF3D_IMPORT.F90	DIF3D_IMPORT	603		Missing debug and fatal error branches	No
DIF3D_MakeADIF3D.F90	DIF3D_MakeADIF3D	674		Missing fatal error branches	No
DIF3D.F90	DIF3D_PRINT				
	DIF3D_RESTOREPRINTING				
	DIF3D_TURNOFFPRINTING				
	DIF3D_VOID	612		Fully covered	No
	DIF3D_VOID_DIF3DTYPE	612		Missing fatal error branches	No
DLAYXS.F90	DLAYXS_Copy				
DLAYXS.F90	DLAYXS_Define	17		Missing fatal error branches	No
DLAYXS_Export.F90	DLAYXS_Export			ASCII file branch	No
DLAYXS_Export_ASCII.F90	DLAYXS_Export_ASCII				
DLAYXS_Export_BINARY.F90	DLAYXS_Export_Binary				
DLAYXS_Import.F90	DLAYXS_Import	17		ASCII file branch	No
DLAYXS_Import_ASCII.F90	DLAYXS_Import_ASCII				
DLAYXS_Import_BINARY.F90	DLAYXS_Import_Binary	17		Missing fatal error branches	No
DLAYXS_Map_to_ISOTXS.F90	DLAYXS_Map_to_ISOTXS	17		Missing fatal error branches	No
DLAYXS.F90	DLAYXS_Print				
	DLAYXS_SetUnit				
	DLAYXS_Void	17		Missing fatal error branches	No
FIXSRC.F90	FIXSRC_Define				
FIXSRC_EXPORT.F90	FIXSRC_Export				
FIXSRC_Export_ASCII.F90	FIXSRC_Export_ASCII				
FIXSRC_Export_Binary.F90	FIXSRC_Export_Binary				
FIXSRC_IMPORT.F90	FIXSRC_Import				
FIXSRC_Import_ASCII.F90	FIXSRC_Import_ASCII				
FIXSRC_Import_Binary.F90	FIXSRC_Import_Binary				
FIXSRC.F90	FIXSRC_Print				
	FIXSRC_SetUnit				
	FIXSRC_Void				
GEODST.F90	GEODST_AddMeshes				
	GEODST_ASSIGNPRINTINFO	54	289	Fully covered	No
	GEODST_ComputeAverageThickness				
	GEODST_ComputeCentroids				
	GEODST_Copy	59	602	Missing NRASS=0 related branch	No
	GEODST_DEFINE	167	1,204	Missing fatal error branching and debug printing	No

GEODST_EXPORT.F90	GEODST_EXPORT	285		ASCII file branch	No
GEODST_EXPORT_ASCII.F90	GEODST_EXPORT_ASCII				
GEODST_EXPORT_BINARY.F90	GEODST_EXPORT_BINARY	285		Missing debug and fatal error branches	No
GEODST_IMPORT.F90	GEODST_IMPORT	74	301	Missing ASCII file branch	No
GEODST_IMPORT_ASCII.F90	GEODST_IMPORT_ASCII				
GEODST_EXPORT.F90	GEODST_IMPORT_BINARY	74	301	Missing fatal error branches, debug print branches, NRASS=1 branch	No
GEODST.F90	GEODST_LocatePoint				
	GEODST_MeshVolumes		301	Missing non-hexagonal branches	No
	GEODST_NODAL_factors		301		
	GEODST_NRASS1	34	602	Fully covered	No
	GEODST_PRINT				
	GEODST_RegionsOnBoundary				
	GEODST_RingPositionIJ	1	8,988	Missing fatal error branching and invalid input branch	No
	GEODST_VOID	167	903	Missing fatal error branching	No
ISOTXS_Close.F90	ISOTXS_Close	127		Missing fatal error branches and file chi	
ISOTXS_Collect_Info.F90	ISOTXS_Collect_Info				
ISOTXS_Create.F90	ISOTXS_Create	63		Missing fatal error branches	No
ISOTXS_GetIsotope.F90	ISOTXS_GetIsotope	33594		Missing fatal error branches	No
ISOTXS_Interface.F90	ISOTXS_Interface_Copy				
	ISOTXS_Interface_Define	181		Missing debug and fatal error branches	No
	ISOTXS_Interface_IdentifyIsotope	34281		Fully covered	No
	ISOTXS_Interface_Print				
	ISOTXS_Interface_SetUnit				
	ISOTXS_Interface_Void	191		Missing fatal error branches	No
ISOTXS_Isotope.F90	ISOTXS_Isotope_CalculateNOP	677976		Fully covered	No
	ISOTXS_Isotope_Collapse				
	ISOTXS_Isotope_ComputeBanding	6186		Zero band size branch	No
	ISOTXS_Isotope_ComputeMAXORD	39780		Fully covered	No
	ISOTXS_Isotope_Copy				
	ISOTXS_Isotope_Copy_AlterNSCMAX				

	ISOTXS_Isotope_Define	79560		Missing fatal error branches	No
	ISOTXS_Isotope_Merge				
	ISOTXS_Isotope_N2NFactor	13500		Missing production based n2n scattering matrix	No
	ISOTXS_Isotope_NegativeDefinite				
	ISOTXS_Isotope_Print				
	ISOTXS_Isotope_SetUnit				
	ISOTXS_Isotope_Void	119330		Missing fatal error branches	No
ISOTXS_Modify_Isotope.F90	ISOTXS_Modify_Isotope	19		A variety of branches for different reactions	No
ISOTXS_Modify_Library.F90	ISOTXS_Modify_Library	10		Missing debug or error branches	No
ISOTXS_Open.F90	ISOTXS_Open	118		Missing fatal error branches and file chi branch	No
ISOTXS_PutIsotope.F90	ISOTXS_PutIsotope	6186		Missing fatal error branches	No
ISOTXS_ReadIsotope.F90	ISOTXS_ReadIsotope	39780		Chi matrix branch	No
ISOTXS_Rewind.F90	ISOTXS_Rewind	145		Missing fatal error branches	No
ISOTXS_Unique_HABSID.F90	ISOTXS_Unique_HABSID	53		Fully covered	No
ISOTXS_WriteIsotope.F90	ISOTXS_WriteIsotope	12372		Chi matrix branch	No
LABELS.F90	LABELS_AssignPrintInfo	54	289	Fully covered	No
	LABELS_Copy				
	LABELS_Define	74	301	Missing fatal error branches, debug print branches	No
LABELS_EXPORT.F90	LABELS_EXPORT	285		Missing fatal error branches	No
LABELS.F90	LABELS_IdentifyArea				
	LABELS_IdentifyRegion		940		
	LABELS_IdentifyZone	882		Fully covered	No
LABELS_IMPORT.F90	LABELS_IMPORT	74	301	Missing fatal error branches, debug print branches	No
LABELS_Integrate.F90	LABELS_Integrate				
	LABELS_IntegrateMD				
	LABELS_PRINT				
LABELS_Integrate.F90	LABELS_Void	74	301	Missing fatal error branch	No
MATERIALS.F90	MATERIALS_Append				
	MATERIALS_CheckProperties				
	MATERIALS_Copy	1080	20	Missing structure already defined branch	No

	MATERIALS_Define	1597	26	Missing fatal error branches	No
	MATERIALS_EliminateDuplicates	201		Fully covered	No
	MATERIALS_Normalize				
	MATERIALS_Print				
	MATERIALS_SetUnit		289		
	MATERIALS_Void	874	10	Missing fatal error branches	No
NDXSRF.F90	NDXSRF_ASSIGNPRINTINFO	54	289	Fully covered	No
	NDXSRF_Copy	157			
	NDXSRF_DEFINE	284	301	Missing fatal error branches, debug print branches, subzone setup of structure	No
NDXSRF_EXPORT.F90	NDXSRF_EXPORT	389			
NDXSRF_IMPORT.F90	NDXSRF_IMPORT	74	301	Missing fatal error branches, debug print branches, subzone setup of structure	No
NDXSRF.F90	NDXSRF_PRINT				
NDXSRF_ReadAssignIsotope.F90	NDXSRF_ReadAssignIsotope		301		
NDXSRF.F90	NDXSRF_VOID	441	301	Missing fatal error branches, debug print branches	No
NE_FreeForm.F90	NE_FreeForm_ADDWORD	118776	>14,000	Missing fatal error branches, bad input cases, and special case string handling	No
	NE_FreeForm_FINDWORD	19125	14,731	Missing fatal error branches, bad input cases, and special case string handling	No
	NE_FreeForm_CLEAR	20743		Fully covered	No
	NE_FreeForm_Decomposition	20743	23,891	Bad Input branching	No
	NE_FreeForm_PRINT				
	NE_FreeForm_SetUnit				
NE_Kind.F90	NE_Kind_AddUnderscores	2750		Fully covered	No
	NE_Kind_AlphabetInteger				
	NE_Kind_DIF3DCharacterFix	774	3,990,104	Fully covered	No
	NE_Kind_FREELOGICALUNIT	4606	2,425	Missing fatal error branch and size limit	No
	NE_Kind_GetFreeLogicalUnit	4688	3,032	Missing fatal error branches	No
	NE_Kind_IntegerAlphabet				
	NE_Kind_IntToLogical				
	NE_Kind_LogicalToInt				

	NE_Kind_SetUnit				
	NE_Kind_StandardCharacterFix	8668	39,014	Fully covered	No
	NE_Kind_UpperCaseString	485702	38,113	Fully covered	No
	NE_Kind_YesOrNo				
NE_LMA_Disk.F90	NE_LMA_Disk_AssignPrintInfo				
	NE_LMA_Disk_Define				
	NE_LMA_Disk_Print				
	NE_LMA_Disk_Void				
NE_LMA_Partition.F90	NE_LMA_Partition_AssignPrintInfo				
	NE_LMA_Partition_Copy				
	NE_LMA_Partition_Define				
	NE_LMA_Partition_Print				
	NE_LMA_Partition_Void				
NE_LMA_R64.F90	NE_LMA_R64_AssignPrintInfo				
NE_LMA_R64_Create.F90	NE_LMA_R64_Create				
NE_LMA_R64.F90	NE_LMA_R64_Define				
NE_LMA_R64_Disk_Load.F90	NE_LMA_R64_Disk_Load				
NE_LMA_R64_Disk_Open.F90	NE_LMA_R64_Disk_Open				
NE_LMA_R64_Disk_Store.F90	NE_LMA_R64_Disk_Store				
NE_LMA_R64_Free.F90	NE_LMA_R64_Free				
NE_LMA_R64_Get.F90	NE_LMA_R64_Get				
NE_LMA_R64.F90	NE_LMA_R64_Print				
	NE_LMA_R64_Void				
NE_LMA_Resident.F90	NE_LMA_Resident_AssignPrintInfo				
	NE_LMA_Resident_Compress				
	NE_LMA_Resident_Define				
	NE_LMA_Resident_Find				
	NE_LMA_Resident_Print				
	NE_LMA_Resident_Void				
NE_VTKexport.F90	NE_VTKexport_ALLOCATE	60	33	Missing fatal error branches	No
	NE_VTKexport_ALLOCATE_Block	158866	>33	Missing fatal error branches, debug output branches	No
	NE_VTKexport_Define	38	33	Fully covered	No
	NE_VTKexport_Define_Block	158866	12,593	Fully covered	No
	NE_VTKexport_DefineMapping	476598	37,779	Missing unknown element type branch	No
	NE_VTKexport_EXPORT	60	33	Missing fatal error branches	No

	NE_VTKexport_EXPORT_ASCII_VTK	60	>33	Missing fatal error branches, parallel output branch	No
	NE_VTKexport_PRINT				
	NE_VTKexport_Print_Block				
	NE_VTKexport_Print_VTKdata				
	NE_VTKexport_Void	38	33	Missing optional block input case	No
	NE_VTKexport_Void_VTKdata	38	>33	Missing fatal error branch	No
	NE_VTKexport_Void_block_VTKdata	38	33	Missing fatal error branch	No
NHFLUX_ActiveNodeAdjacency.F90	NHFLUX_ActiveNodeAdjacency		289	Missing fatal error branches, debug print branches	No
NHFLUX.F90	NHFLUX_ASSIGNPRINTINFO	107	289	Fully covered	No
	NHFLUX_DEFINE	622	903	Missing fatal error branches, debug print branches	No
NHFLUX_EvaluateFlux.F90	NHFLUX_EvaluateFlux				
NHFLUX_EXPORT.F90	NHFLUX_EXPORT	373		Missing debug and fatal error branches	No
NHFLUX.F90	NHFLUX_FindActiveNode				
NHFLUX_GeometryMaps.F90	NHFLUX_GeometryMaps		301		
NHFLUX_IMPORT.F90	NHFLUX_IMPORT	590	602	Missing fatal error branches, debug print branches, adjoint flux branch, previously defined structure	No
NHFLUX.F90	NHFLUX_INTEL_IDEAL_OPEN	53		Missing very large block read	No
	NHFLUX_PRINT				
NHFLUX_RingSectorPosition.F90	NHFLUX_RingSectorPosition		289		
NHFLUX_SetupVARSRC.F90	NHFLUX_SetupVARSRC				
NHFLUX.F90	NHFLUX_VOID	643	903	Missing fatal error branches	No
NHSTRAIN.F90	NHSTRAIN_ASSIGNPRINTINFO				
	NHSTRAIN_DEFINE	426		Missing fatal error branches	No
NHSTRAIN_Export_to_MCNP_Hex.F90	NHSTRAIN_EtoMCNP_Hex_Wedges	296256		Fully covered	No
NHSTRAIN.F90	NHSTRAIN_EvalPoints	325168		Branch for quadratic element basis	No
NHSTRAIN_Export.F90	NHSTRAIN_Export				
NHSTRAIN_Export_to_MCNP.F90	NHSTRAIN_Export_to_MCNP	22		Missing fatal error branches	No
NHSTRAIN_Export_to_MCNP_Cart.F90	NHSTRAIN_Export_to_MCNP_Cart	10		Missing fatal error branches and branches for specific geometry setup	No

NHSTRAIN_Export_to_MCNP_Hex.F90	NHSTRAIN_Export_to_MCNP_Hex	12		Missing fatal error branches and branches for specific geometry setup	No
NHSTRAIN_Import.F90	NHSTRAIN_Import	13		Missing debug and fatal error branches	No
NHSTRAIN.F90	NHSTRAIN_PRINT				
	NHSTRAIN_SPLIT	22		Missing fatal error branches	No
	NHSTRAIN_VOID	426		Missing fatal error branches	No
NonZero_StoreMatrix.f90	NonZero_StoreMatrix	1998		Fully covered	No
NonZeroStorage_D.f90	NonZeroStorage_D_Allocate	1837		Missing debug and fatal error branches	No
	NonZeroStorage_D_CopyFromNZRWS	5	8,324	Missing fatal error branches	No
	NonZeroStorage_D_CopyFromNZS				
	NonZeroStorage_D_Define	1837		Fully covered	No
	NonZeroStorage_D_Print				
	NonZeroStorage_D_Print_Type				
	NonZeroStorage_D_SetUnit				
	NonZeroStorage_D_Void		8,324		
	NonZeroStorage_D_Void_Type				
NonZeroRowWiseStorage_D.f90	NZRowWiseStorage_D_Define	5	8,324	Fully covered	No
	NZRowWiseStorage_D_Fill	625470	56,498	Missing fatal error branches, duplicated column storage	No
	NZRowWiseStorage_D_Print				
	NZRowWiseStorage_D_Reset				
	NZRowWiseStorage_D_SetUnit				
	NZRowWiseStorage_D_Void	5	8,324	Fully covered	No
	NZRWS_D_Allocate_RowWiseData	625470		Missing fatal error branches	No
	NZRWS_D_Allocate_Type	5		Missing fatal error branches	No
	NZRWS_D_Print_Row				
	NZRWS_D_Print_Type				
	NZRWS_D_Void_RowWiseData	1377025		Missing fatal error branches	No
	NZRWS_D_Void_Type	5		Missing fatal error branches	No
PMATRX.F90	PMATRX_ASSIGNPRINTINFO				
	PMATRX_DEFINE		289		
PMATRX_EXPORT.F90	PMATRX_EXPORT_BINARY				
PMATRX_IMPORT.F90	PMATRX_IMPORT_BINARY		289		
PMATRX.F90	PMATRX_PRINT				

	PMATRX_VOID				
	PWDINT_ASSIGNPRINTINFO				
PWDINT.F90	PWDINT_DEFINE				
PWDINT_EXPORT.F90	PWDINT_EXPORT				
PWDINT_EXPORT_ASCII.F90	PWDINT_EXPORT_ASCII				
PMATRX_EXPORT_BINARY.F90	PWDINT_EXPORT_BINARY				
PWDINT_IMPORT.F90	PWDINT_IMPORT				
PWDINT_IMPORT_ASCII.F90	PWDINT_IMPORT_ASCII				
PMATRX_IMPORT_BINARY.F90	PWDINT_IMPORT_BINARY				
PWDINT.F90	PWDINT_PRINT				
	PWDINT_TotalPower				
	PWDINT_VOID				
RCTDEN.F90	RCTDEN_ASSIGNPRINTINFO				
	RCTDEN_DEFINE				
RCTDEN_EXPORT.F90	RCTDEN_EXPORT				
RCTDEN_EXPORT_ASCII.F90	RCTDEN_EXPORT_ASCII				
RCTDEN_EXPORT_BINARY.F90	RCTDEN_EXPORT_BINARY				
RCTDEN_IMPORT.F90	RCTDEN_IMPORT				
RCTDEN_IMPORT_ASCII.F90	RCTDEN_IMPORT_ASCII				
RCTDEN_IMPORT_BINARY.F90	RCTDEN_IMPORT_BINARY				
RCTDEN.F90	RCTDEN_PRINT				
	RCTDEN_VOID				
RCTFLX.F90	RCTFLX_ASSIGNPRINTINFO				
	RCTFLX_DEFINE				
RCTFLX_EXPORT.F90	RCTFLX_EXPORT				
RCTFLX_EXPORT_ASCII.F90	RCTFLX_EXPORT_ASCII				
RCTFLX_EXPORT_BINARY.F90	RCTFLX_EXPORT_BINARY				
RCTFLX_IMPORT.F90	RCTFLX_IMPORT				
RCTFLX_IMPORT_ASCII.F90	RCTFLX_IMPORT_ASCII				
RCTFLX_IMPORT_BINARY.F90	RCTFLX_IMPORT_BINARY				
RCTFLX.F90	RCTFLX_PRINT				
	RCTFLX_VOID				
RCTPWD.F90	RCTPWD_ASSIGNPRINTINFO				
	RCTPWD_DEFINE				
RCTPWD_EXPORT.F90	RCTPWD_EXPORT				
RCTPWD_EXPORT_ASCII.F90	RCTPWD_EXPORT_ASCII				
RCTPWD_EXPORT_BINARY.F90	RCTPWD_EXPORT_BINARY				
RCTPWD_IMPORT.F90	RCTPWD_IMPORT				

RCTPWD_IMPORT_ASCII.F90	RCTPWD_IMPORT_ASCII				
RCTPWD_IMPORT_BINARY.F90	RCTPWD_IMPORT_BINARY				
RCTPWD.F90	RCTPWD_PRINT				
	RCTPWD_VOID				
REBUS_T50_ReadInput.F90	REBUS_T50_ReadInput				
REBUS_T50.F90	REBUST50_Define				
	REBUST50_IdentifyMixing				
	REBUST50_Print				
	REBUST50_ReturnCount				
	REBUST50_SetUnit				
	REBUST50_Void				
RMFLUX.F90	RMFLUX_ASSIGNPRINTINFO				
	RMFLUX_DEFINE				
RMFLUX_EXPORT.F90	RMFLUX_EXPORT				
RMFLUX_EXPORT_BINARY.F90	RMFLUX_EXPORT_BINARY				
RMFLUX_IMPORT.F90	RMFLUX_IMPORT				
RMFLUX_IMPORT_BINARY.F90	RMFLUX_IMPORT_BINARY				
RMFLUX.F90	RMFLUX_PRINT				
	RMFLUX_VOID				
RTFLUX.F90	RTFLUX_ASSIGNPRINTINFO				
	RTFLUX_DEFINE				
RTFLUX_EXPORT.F90	RTFLUX_EXPORT				
RTFLUX_EXPORT_ASCII.F90	RTFLUX_EXPORT_ASCII				
RTFLUX_EXPORT_BINARY.F90	RTFLUX_EXPORT_BINARY				
RTFLUX_IMPORT.F90	RTFLUX_IMPORT				
RTFLUX_IMPORT_ASCII.F90	RTFLUX_IMPORT_ASCII				
RTFLUX_IMPORT_BINARY.F90	RTFLUX_IMPORT_BINARY				
RTFLUX.F90	RTFLUX_PRINT				
	RTFLUX_VOID				
RZFLUX.F90	RZFLUX_ASSIGNPRINTINFO				
	RZFLUX_DEFINE				
RZFLUX_EXPORT.F90	RZFLUX_EXPORT				
RZFLUX_IMPORT.F90	RZFLUX_IMPORT				
RZFLUX.F90	RZFLUX_PRINT				
	RZFLUX_VOID				
RZMFLX.F90	RZMFLX_ASSIGNPRINTINFO				
	RZMFLX_DEFINE				
RZMFLX_EXPORT.F90	RZMFLX_EXPORT				

RZMFLX_EXPORT_BINARY.F90	RZMFLX_EXPORT_BINARY				
RZMFLX_IMPORT.F90	RZMFLX_IMPORT				
RZMFLX_IMPORT_BINARY.F90	RZMFLX_IMPORT_BINARY				
RZMFLX.F90	RZMFLX_PRINT				
	RZMFLX_VOID				
SFEDIT.F90	SFEDIT_ASSIGNPRINTINFO				
	SFEDIT_DEFINE				
SFEDIT_EXPORT.F90	SFEDIT_EXPORT				
SFEDIT_IMPORT.F90	SFEDIT_IMPORT				
SFEDIT.F90	SFEDIT_PRINT				
	SFEDIT_VOID				
ZNATDN.F90	ZNATDN_ASSIGNPRINTINFO	54	289	Fully covered	No
	ZNATDN_Copy	112		Fully covered	No
	ZNATDN_DEFINE	186	301	Missing fatal error branches, debug print branches	No
ZNATDN_EXPORT.F90	ZNATDN_EXPORT	397		Missing debug and fatal error branches	No
ZNATDN_IMPORT.F90	ZNATDN_IMPORT	74	301	Missing fatal error branches, branch for previously defined structure	No
ZNATDN.F90	ZNATDN_MergeWithFactors				
	ZNATDN_PRINT				
	ZNATDN_VOID	298	301	Missing fatal error branches	No

4.3 Source code in the directory dif3d/variantbasis/source

The directory *dif3d/variantbasis/source* in the PERSENT distribution stores source files that construct the spatial and angular basis functions and matrices for the DIF3D-VARIANT solver. The original DIF3D-VARIANT used hard-wired spatial and angular matrices and the updates from DIF3D 9.0 to 10.0 introduced these routines to allow higher order space-angle approximations to be investigated. They were moved external to the *dif3d/source* directory as they needed to be accessed by other parts of the ARC software system such as PERSENT, DASSH, and EvaluateFlux. They are almost entirely programmed in Fortran 77 but the extensions were changed to Fortran 90 because of problems with the GNU Fortran compiler and the underlying Fortran 90 functionality being used. The original design of the software was to perform the basis and matrix computation work for Cartesian, hexagonal, and triangular-Z geometries and thus only a few subroutines are called from DIF3D with different input switches to select the angular, spatial, and geometry options. The code coverage summary of the source files in this directory is listed in Table 4.3.

In Table 4.3, one can see that the majority of the subroutines are called by the PERSENT tests. The primary exceptions are the SP_N capability of DIF3D-VARIANT which is not usable in PERSENT. The GetSpaceBasis subroutine is the pathway that PERSENT uses to obtain the space-angle matrices which is not used in DIF3D. Because DIF3D is the primary use of the source code in this directory, one should focus on improving the code coverage there rather than in PERSENT, except for the unique parts that PERSENT uses. Examples of this are the subroutines BreakDown_Input_ID.F90 and GetSpaceBasis.F90 which are effectively fully covered. The remaining uncovered routines that are still not covered can be considered developer subroutines which are not presently usable by DIF3D-VARIANT or any other part of ARC and that they do not need coverage.

Table 4.3 Code Coverage Details of Files in the Directory *dif3d/variantbasis/source* Relative to DIF3D

File Name	Function or Subroutine	# of calls by DIF3D tests	# of calls by PERSENT tests
BreakDown_Input_ID.F90	BreakDown_Input_ID	0	0
Build_Hmatrix.f90	Build_Hmatrix	1010	576
CalculateGamma.f90	CALCULATEGAMMA	172	0
CheckSpatialApprox.f90	CHECKSPATIALAPPROX	82	0
CheckSurfaceRank.f90	CHECKSURFACERANK	82	0
Common_GetSurfacesPerNode.f90	GETSURFACESPERNODE	82	54
Common_LinearAlgebra.f90	MAS_CxV	0	0
	MAS_Mt	199	122
	MAS_MtxM	554	0
	MAS_MxM	601	180
	MAS_MxMt	1076	650
	MAS_MxV	11482	8319
	MAS_VCOPY	1260	0
	MAS_VcrossV	32750	22757
	MAS_VdotV	1057184	882353
	MAS_VpCxV	2946180	2329816
	VECTORMULTIPLY3	0	0
	VECTORMULTIPLY4	40229	9240
	VECTORMULTIPLY5	76	0
Common_SimpleFunctions.f90	CHECKWITHIN	0	0
	DCLEANMATRIX	16211	10807
	DCOUNTZEROS	0	0
	DFILLVECTOR	0	0
	DGETMAX	0	0

	DGETMIN	0	0
	DISPLAYMATRIX	0	0
	DMIKESORT	0	0
	DNORMALIZE	0	0
	DREMOVEZEROS	490	264
	DSIMPLESUM	0	0
	ICLEANMATRIX	688	468
	IGETMAX	1013	419
	IGETMIN	26	5
	MAS_XtoN	1143597	954336
	OBTAINFACTORIAL	687176	84264
ConstructSpaceAngleMatrices.f90	ConstructSpaceAngleMatrices	82	54
GQ00_GetSnQuadPoints.f90	GETSNQUADPOINTS	426	54
GQ00_GrabSnQuad.f90	GRABSNQUAD	426	54
GQ01_OneDChebyInt.f90	ONEDCHEBY	381	54
GQ01_OneDGaussInt.f90	ONEDGAUSS	426	65
GetSpaceBasis.f90	GetSpaceBasis	0	54
Pn01_DefineEven_RMA.f90	DEFINEEVEN_RMA	0	0
Pn01_DefineOdd_RMA.f90	DEFINEODD_RMA	100	0
Pn01_Obtain_H_V_E.f90	OBTAIN_H_V_E	73	54
Pn01_Obtain_K_Vacuum.f90	OBTAIN_K_VACUUM	0	0
Pn01_Obtain_L_Vacuum.f90	OBTAIN_L_VACUUM	154	0
Pn01_Obtain_PnE.f90	OBTAIN_PnE	154	0
Pn02_DefineReductionMatrix.f90	DEFINEREDUCTIONMATRIX	327	54
	WonSikConstant	0	0
Pn02_IndexYLMTerms.f90	INDEXYLMTERMS	327	54
Pn02_IntegrateEmatrices.f90	INTEGRATE_EMATRICES	254	54
	INTEGRATE_EMATRIX	456	180

Pn02_IntegrateHmatrices.f90	INTEGRATE_HMATRICES	9	0
	INTEGRATE_HMATRIX	9	0
Pn02_IntegrateIdentities.f90	INTEGRATE_IDENTITIES	0	0
Pn02_IntegrateVacuum.f90	INTEGRATE_VACUUM	172	0
Pn02_IntegrateVmatrices.f90	INTEGRATE_VMATRICES	73	54
	INTEGRATE_VMATRIX	209	142
Pn02_ObtainHmatrices.f90	OBTAIN_HMATRICES	73	54
Ritz01_Obtain_P_U_D_W.f90	OBTAIN_P_U_D_W	82	54
Ritz02_ObtainSurfaceDMatrix.f90	OBTAINSURFACEDMATRIX	606	360
Ritz02_ObtainSurfaceWMatrix.f90	OBTAINSURFACEWMATRIX	0	0
Ritz02_ObtainVolumeFMatrix.f90	OBTAINVOLUMEFMATRIX	0	0
Ritz02_ObtainVolumePMatrix.f90	OBTAINVOLUMEPMATRIX	668	386
Ritz02_ObtainVolumePaltMatrix.f90	OBTAINVOLUMEPaltMATRIX	0	0
Ritz02_ObtainVolumeUMatrix.f90	OBTAINVOLUMEUMATRIX	232	142
Ritz03_GetDSurfaceIntegrands.f90	GETdSURFACEINTEGRANDS	606	360
Ritz03_GetPVolumeIntegrands.f90	GETpVOLUMEINTEGRANDS	0	0
Ritz03_GetUVolumeIntegrands.f90	GETuVOLUMEINTEGRANDS	232	142
Ritz03_GetWSurfaceIntegrands.f90	GETwSURFACEINTEGRANDS	0	0
RitzOS00_GetNumberRitzTerms.f90	GETNUMBERRITZTERMS	1016	522
	RET_GETNUMBERRITZTERMS	0	0
RitzOS00_GetSurfaceExpansionOrders.f90	GETSURFACEEXPANSIONORDERS	1212	720
RitzOS00_RitzOrthogonalSet.f90	RITZORTHOGONALSET	688	468
RitzOS01_GetPolyExponents.f90	GETPOLYEXPONENTS	688	468
RitzOS01_GetVolumeIntegrands.f90	GETVOLUMEINTEGRANDS	688	468
RitzOS02_Int_CartesianNode.f90	INT_CARTESIANNODE	39698	153724
RitzOS02_Int_EquiTriZNode.f90	INT_EQUITRIZNODE	0	0
RitzOS02_Int_HexZNode.f90	INT_HEXZNODE	726447	541628
RitzOS03_GetRitzPolyCoef.f90	GETRITZPOLYCOEF	688	468

RitzOS03_HexSurfaceW.f90	HEXSURFACEW	37444	23600
RitzOS03_SimpleEquiTriSurf.f90	SIMPLEEQUITRISURF	0	0
RitzOS03_SimpleHexSurf.f90	SIMPLEHEXSURF	668715	505250
RitzOS03_SimpleResult.f90	SIMPLERESULT	804602	940222
Ritz_PeriodicSpace.f90	Ritz_PeriodicSpace	109	0
SPn00_GetSPnTerms.f90	GETSPNTERMS	151	0
	RET_GETSPNTERMS	0	0
SPn01_ObtainSPn_K_Vacuum.f90	OBTAINSPn_K_VACUUM	0	0
SPn01_ObtainSPn_L_Vacuum.f90	OBTAINSPn_L_VACUUM	18	0
SPn01_Obtain_H_E.f90	OBTAIN_H_E	9	0
SPn01_Obtain_SPnE.f90	OBTAIN_SPnE	18	0
SPn02_EvaluateSPn.f90	EVALUATE_SPN	45	0
SPn03_EvaluatePn.f90	EVALUATEPN	182	0
YLM00_AngularSurfPerNode.f90	ANGULARSURFPERNODE	73	54
YLM00_GetEvenParityTerms.f90	GETEVENPARITYTERMS	740	192
	RET_GETEVENPARITYTERMS	36	0
YLM00_GetOddParityTerms.f90	GETODDPARITYTERMS	667	246
	RET_GETODDPARITYTERMS	36	0
YLM00_Obtain_YLM_Mu.f90	OBTAIN_YLM_MU	381	54
YLM01_AngularRotations.f90	ANGULARROTATIONS	381	54
YLM01_EvaluateYlm.f90	EVALUATEYLM	381	54
YLM02_DefineSurfaceMapping.f90	DEFINESURFACEMAPPING	381	54
YLM02_EvaluateYLMSurface.f90	EVALUATEYLM_SURFACE	583	180
YLM03_EvaluateAssociatedPn.f90	EVALUATEASSOCIATEDPN	21716	3108
YLM03_FormYLM3d.f90	FORM_YLM3D	21716	3108
YLM04_GetColumnP.f90	GETCOLUMNP	92392	11720

4.4 Source code in the directory dif3d/lapack/source

The files in directory *dif3d/lapack/source* in the PERSENT distribution were not created by ANL and while it contains some useful capabilities for linear algebra manipulations, only some parts are used by PERSENT. Because the LAPACK setups have multiple matrix operation features controlled by input to each subroutine, none of the files in Table 4.4 are fully covered. The lack of coverage can be completely ignored for LAPACK and it should really be tested externally for the ARC software.

Table 4.4 Code Coverage Details of Files in the Directory *dif3d/lapack/source* Relative to DIF3D

File Name	Function or Subroutine	# of calls by DIF3D tests	# of calls by PERSENT tests
blktr1.f	blktr1	0	0
blktri.f	BLKTRI	0	0
bsrh.f	bsrh	0	0
compb.f	compb	0	0
cprod.f	cprod	0	0
cprodp.f	cprodp	0	0
dbdsqr.f	DBDSQR	0	0
dgebak.f	DGEBAK	0	0
dgebal.f	DGEBAL	0	0
dgebd2.f	DGEBD2	0	0
dgebrd.f	DGEBRD	0	0
dgeev.f	DGEEV	0	0
dgehd2.f	DGEHD2	0	0
dgehrd.f	DGEHRD	0	0
dgelq2.f	DGELQ2	0	0
dgelqf.f	DGELQF	0	0
dgemv.f	DGEMV	2134676	15968
dgeqpf.f	DGEQPF	0	0
dgeqr2.f	DGEQR2	0	0
dgeqrf.f	DGEQRF	0	0
dger.f	DGER	378008	15968
dgerq2.f	DGERQ2	0	0
dgesvd.f	DGESVD	0	0
dgetf2.f	DGETF2	23551	3295
dgetrf.f	DGETRF	0	3295
dgetri.f	DGETRI	23540	3295
dggsvd.f	DGGSVD	0	0
dggsvp.f	DGGSVP	0	0

dhseqr.f	DHSEQR	0	0
dlabad.f	DLABAD	0	0
dlabrd.f	DLABRD	0	0
dlacpy.f	DLACPY	0	0
dladiv.f	DLADIV	0	0
dlags2.f	DLAGS2	0	0
dlahqr.f	DLAHQR	0	0
dlahrd.f	DLAHRD	0	0
dlaln2.f	DLALN2	0	0
dlamc1.f	DLAMC1	0	0
dlamc2.f	DLAMC2	0	0
dlamc3.f	DLAMC3	0	0
dlamc4.f	DLAMC4	0	0
dlamc5.f	DLAMC5	0	0
dlamch.f	DLAMCH	0	0
dlange.f	DLANGE	0	0
dlanhs.f	DLANHS	0	0
dlanv2.f	DLANV2	0	0
dlapll.f	DLAPLL	0	0
dlapmt.f	DLAPMT	0	0
dlapy2.f	DLAPY2	0	0
dlarf.f	DLARF	0	0
dlarfb.f	DLARFB	0	0
dlarfg.f	DLARFG	0	0
dlarft.f	DLARFT	0	0
dlarfx.f	DLARFX	0	0
dlartg.f	DLARTG	0	0
dlas2.f	DLAS2	0	0
dlascl.f	DLASCL	0	0
dlaset.f	DLASET	0	0
dlasr.f	DLASR	0	0
dlassq.f	DLASSQ	0	0
dlasv2.f	DLASV2	0	0
dlaswp.f	DLASWP	0	0
dlauu2.f	DLAUU2	37540	0
dlauum.f	DLAUUM	34909	0
dlazro.f	DLAZRO	0	0
dnrm2.f	dnrm2	0	0
dorg2r.f	DORG2R	0	0

dorgbr.f	DORGBR	0	0
dorghr.f	DORGHR	0	0
dorgl2.f	DORGL2	0	0
dorglq.f	DORGLQ	0	0
dorgqr.f	DORGQR	0	0
dorm2r.f	DORM2R	0	0
dormbr.f	DORMBR	0	0
dorml2.f	DORML2	0	0
dormlq.f	DORMLQ	0	0
dormqr.f	DORMQR	0	0
dormr2.f	DORMR2	0	0
dpotf2.f	DPOTF2	34967	0
dpotrf.f	DPOTRF	0	0
dpotri.f	DPOTRI	34932	0
drot.f	drot	0	0
dsyrk.f	DSYRK	2630	0
dtgsja.f	DTGSJA	0	0
dtrevc.f	DTREVC	0	0
dtrmm.f	DTRMM	7913	0
dtrmv.f	DTRMV	1285387	19263
dtrsm.f	DTRSM	3381	0
dtrti2.f	DTRTI2	61324	3295
dtrtri.f	DTRTRI	58462	3295
epmach.f	epmach	0	0
ilaenv.f	ILAENV	116857	9885
indxa.f	indxa	0	0
indxb.f	indxb	0	0
indxc.f	indxc	0	0
lsame.f	LSAME	23202599	288945
ppadd.f	ppadd	0	0
ppsgf.f	ppsgf	0	0
ppspf.f	ppspf	0	0
prod.f	prod	0	0
prodp.f	prodp	0	0
psgf.f	psgf	0	0
sgemv.f	SGEMV	0	0
sger.f	SGER	0	0
sgetf2.f	SGETF2	0	0
sgetrf.f	SGETRF	0	0

sgetri.f	SGETRI	0	0
slaswp.f	SLASWP	0	0
slauu2.f	SLAUU2	0	0
slauum.f	SLAUUM	0	0
spotf2.f	SPOTF2	0	0
spotri.f	SPOTRI	0	0
ssyrk.f	SSYRK	0	0
store.f	store	0	0
strmm.f	STRMM	0	0
strmv.f	STRMV	0	0
strsm.f	STRSM	0	0
strti2.f	STRTI2	0	0
strtri.f	STRTRI	0	0
tevls.f	tevls	0	0
xerbla.f	XERBLA	0	0

4.5 Source code in the directory dif3d/linpack/source

The files in directory *dif3d/linpack/source* in the PERSENT distribution were collaboratively developed and some work originated at ANL. Much like LAPACK, LINPACK contains some useful matrix manipulation capabilities but virtually none are used in PERSENT. Because LAPACK calls the LINPACK routines, much of the coverage of LINPACK is due entirely to the usage of LAPACK in PERSENT. The lack of coverage can be completely ignored for LINPACK and it should really be tested externally for the ARC software.

Table 4.5 Code Coverage Details of Files in the Directory *dif3d/linpack/source* Relative to DIF3D

File Name	Function or Subroutine	# of calls by DIF3D tests	# of calls by PERSENT tests
blas1.F90	linpack_dasum	0	0
	linpack_daxpy	0	946773360
	linpack_dcopy	0	0
	linpack_ddot	0	0
	linpack_dmach	0	0
	linpack_dnrm2	0	0
	linpack_drot	0	0
	linpack_drotg	0	0
	linpack_drotm	0	0
	linpack_drotmg	0	0
	linpack_dscal	0	18548640

	linpack_dswap	0	0
	linpack_idamax	0	18548640
	linpack_lsame	0	0
	linpack_xerbla	0	0
daxpy.f	daxpy	0	0
dcopy.f	dcopy	705769	0
ddot.f	ddot	1803279	0
dgedi.f	dgedi	0	0
dgefa.f	dgefa	0	0
dgemm.f	DGEMM	3402	0
dpodi.f	dpodi	0	0
dpofa.f	dpofa	0	0
dscal.f	dscal	2584821	35231
dswap.f	dswap	4483	0
dsymm.f	DSYMM	0	0
idamax.f	idamax	401563	19263
isamax.f	isamax	0	0
linpack.F90	linpack_dchdc	0	0
	linpack_dchdd	0	0
	linpack_dchex	0	0
	linpack_dchud	0	0
	linpack_dgbco	0	0
	linpack_dgbdi	0	0
	linpack_dgbfa	0	0
	linpack_dgbsl	0	0
	linpack_dgeco	0	0
	linpack_dgedi	0	0
	linpack_dgefa	0	215280
	linpack_dgesl	0	645840
	linpack_dgtsl	0	0
	linpack_dpbco	0	0
	linpack_dpbdi	0	0
	linpack_dpbfa	0	0
	linpack_dpbsl	0	0
	linpack_dpoco	0	0
	linpack_dpodi	0	0
	linpack_dpofa	0	0
	linpack_dposl	0	0
	linpack_dppco	0	0

	linpack_dppdi	0	0
	linpack_dppfa	0	0
	linpack_dppsl	0	0
	linpack_dptsl	0	0
	linpack_dqrdc	0	0
	linpack_dqrsl	0	0
	linpack_dsico	0	0
	linpack_dsidi	0	0
	linpack_dsifa	0	0
	linpack_dsisl	0	0
	linpack_dspco	0	0
	linpack_dspdi	0	0
	linpack_dspfa	0	0
	linpack_dspsl	0	0
	linpack_dsvdc	0	0
	linpack_dtrco	0	0
	linpack_dtrdi	0	0
	linpack_dtrsl	0	0
saxpy.f	saxpy	0	0
scopy.f	scopy	0	0
sdot.f	sdot	0	0
sgedi.f	sgedi	0	0
sgefa.f	sgefa	0	0
sgemm.f	SGEMM	0	0
spodi.f	spodi	0	0
spofa.f	spofa	0	0
sscal.f	sscal	0	0
sswap.f	sswap	0	0
ssymm.f	SSYMM	0	0

4.6 Source code in the directory dif3d/dif3dtovtk/source

The files in directory *dif3d/dif3dtovtk/source* of the PERSENT distribution were developed as a post processing visualization capability for DIF3D. Because the various subroutines that translate the GEODST geometry into the finite element visualization file (VTK) are written as subroutines, they are callable by any program such as PERSENT. No details on the code coverage are given, but it should be clear that the hexagonal and x-y-z geometry options valid with DIF3D-VARIANT are used in PERSENT and the other geometry options are not. No further coverage tests are needed

for these files and they should be tested external to the PERSENT software like the *dif3d/arcmodules* source files.

Table 4.6 Relative Coverage Details of Files in the Directory *dif3d/dif3dtovtk/source* to DIF3D

File Name	Function or Subroutine	# of calls by DIF3D tests	# of calls by PERSENT tests
Create_VTK_Cell_Based.F90	Create_VTK_Cell_Based	0	38
Create_VTK_Cell_Based_Hex.F90	Create_VTK_Cell_Based_HEX	0	14
Create_VTK_Cell_Based_ThetaRZ.F90	Create_VTK_Cell_Based_ThetaRZ	0	0
Create_VTK_Cell_Based_TriZ.F90	Create_VTK_Cell_Based_TriZ	0	0
Create_VTK_Cell_Based_XYZ.F90	Create_VTK_Cell_Based_XYZ	0	24

5. Summary and Conclusions

In the preceding analysis the coverage report for each submodule of PERSENT was given. The submodule refers to separable libraries (LINPACK, LAPACK, etc.) that are included either partially or fully as part of PERSENT even though the development of the source code was not specific to PERSENT. For the PERSENT specific source code, most of the subroutines were fully or partially covered to the point where additional testing is not necessary. As part of this work, a considerable number of unused subroutines was identified in PERSENT and it is recommended that those subroutines be removed from the PERSENT source code. There was also considerable branching put in place for different χ treatments which should also be removed from the source code.

Five features were identified in section 3.1.2 that dominate the reason for lack of full coverage on many of the subroutines in PERSENT. Of those five features, only the lack of coverage on all possible reaction types in the reaction rate and reaction rate ratio sensitivity subroutines was identified as needing more coverage. To address some of the other coverage issues, it is recommended to include a transport case for the bowing reactivity worth capability for a Cartesian geometry and include unit testing on the structures in the PERSENT code.

REFERENCES

[1] R. P. Hosteny, "The ARC System Fuel Cycle Analysis Capability, REBUS-2," ANL-7721, Argonne National Laboratory (1978).

[2] B. J. Toppel, "A User's Guide to the REBUS-3 Fuel Cycle Analysis Capability," ANL-83-2, Argonne National Laboratory (1983).

[3] W. S. Yang, M. A. Smith, "Theory Manual for the Fuel Cycle Analysis Code REBUS," ANL/NE-19/21, Argonne National Laboratory (2020).

[4] 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).

[5] 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).

[6] R. D. Lawrence, "Progress in Nodal Methods for the Solution of the Neutron Diffusion and Transport Equations," *Prog. Nucl. Energy*, **17**, 271 (1986).

[7] M. R. Wagner, "Three-Dimensional Nodal Diffusion and Transport Theory for Hexagonal-Z Geometry," *Nucl. Sci. Eng*., **103**, 377-391 (1989).

[8] G. Palmiotti, E. E. Lewis, and C. B. Carrico, "VARIANT: VARIational Anisotropic Nodal Transport for Multidimensional Cartesian and Hexagonal Geometry Calculation," ANL-95/40, Argonne National Laboratory (1995).

[9] M. A. Smith, E. E. Lewis, and E. R. Shemon, "DIF3D-VARIANT 12.0, A Decade of Updates," ANL/NE-14/1 Rev. 1, Argonne National Laboratory (2025).

[10] M. A. Smith, C. Adams, W. S. Yang, and E. E. Lewis, "VARI3D & PERSENT: Perturbation and Sensitivity Analysis," ANL/NE-13/8 Rev. 6, Argonne National Laboratory (2025).

[11] M. A. Smith, "Requirements Description of the PERSENT Software," ANL/NSE-24/80, Argonne National Laboratory (2024).

[12] M. A. Smith, "Verification of the PERSENT Software," ANL/NSE-24/83 Rev. 1, Argonne National Laboratory (2024).

[13] A. Nelson and M. A. Smith, "Verification of the DIF3D Software to Support Fast Reactor Analysis," ANL/NSE-20/3, Argonne National Laboratory (2020).

[14] Z. Zhong, M. A. Smith, C. H. Lee, "Code Coverage Status of the ARC Code DIF3D," ANL/NSE-23/65 Rev. 1, Argonne National Laboratory (2025).

[15] M. A. Smith, "Code Coverage Status of the ARC Code DASSH-F," ANL/NSE-25/48, Argonne National Laboratory (2025).

[16] K. Kiesling and M. A. Smith, "Software Quality Assurance Plan," ANL/NSE-23/13, Argonne National Laboratory (2023).

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