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-25/48

Code Coverage Status of the ARC Code DASSH-F

Nuclear Science and Engineering Division

ANL/NSE-25/48

Code Coverage Status of the ARC Code DASSH-F

prepared by

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. DASSH-F serves as a steady state thermal hydraulic capability within the ARC system and replaces the SE2-ANL software that preceded it.

This document identifies the set of test problems used to assess the code coverage for DASSH-F. The goal is to document what parts of the existing DASSH-F code are touched by the set of test problems and which are not. Because the verification work remains to be done on DASSH-F, one can assume that most of these issues will be resolved as part of that work.

The code coverage analysis of DASSH-F was performed with the Code Coverage Tool of the Intel Fortran compiler which requires modifications to the compilation of DASSH-F. The code coverage tables are given for each submodule of DASSH-F. Because DASSH-F links to modules in DIF3D, some details on coverage changes to the DIF3D linked files is provided. As will be seen, most of the uncovered parts/files can be ignored because they are either for error message and debugging output or obviously not needed by DASSH-F today. Seven features of the DASSH-F code were identified to not be covered by the existing testing suite and thus additional verification test problems are suggested to fully cover these sections.

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 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. DASSH-F [10], referred from this point as DASSH, is called for each time point of the REBUS fuel cycle analysis work and serves as the steady state thermal analysis capability in the ARC system. DASSH is the replacement for SUPERENERGY-2 [11] which performs the same functionalities as DASSH. Both codes are built to take power details from a GAMSOR [12] calculation (neutron and gamma) but DASSH can also work only with the neutron power distribution from DIF3D.

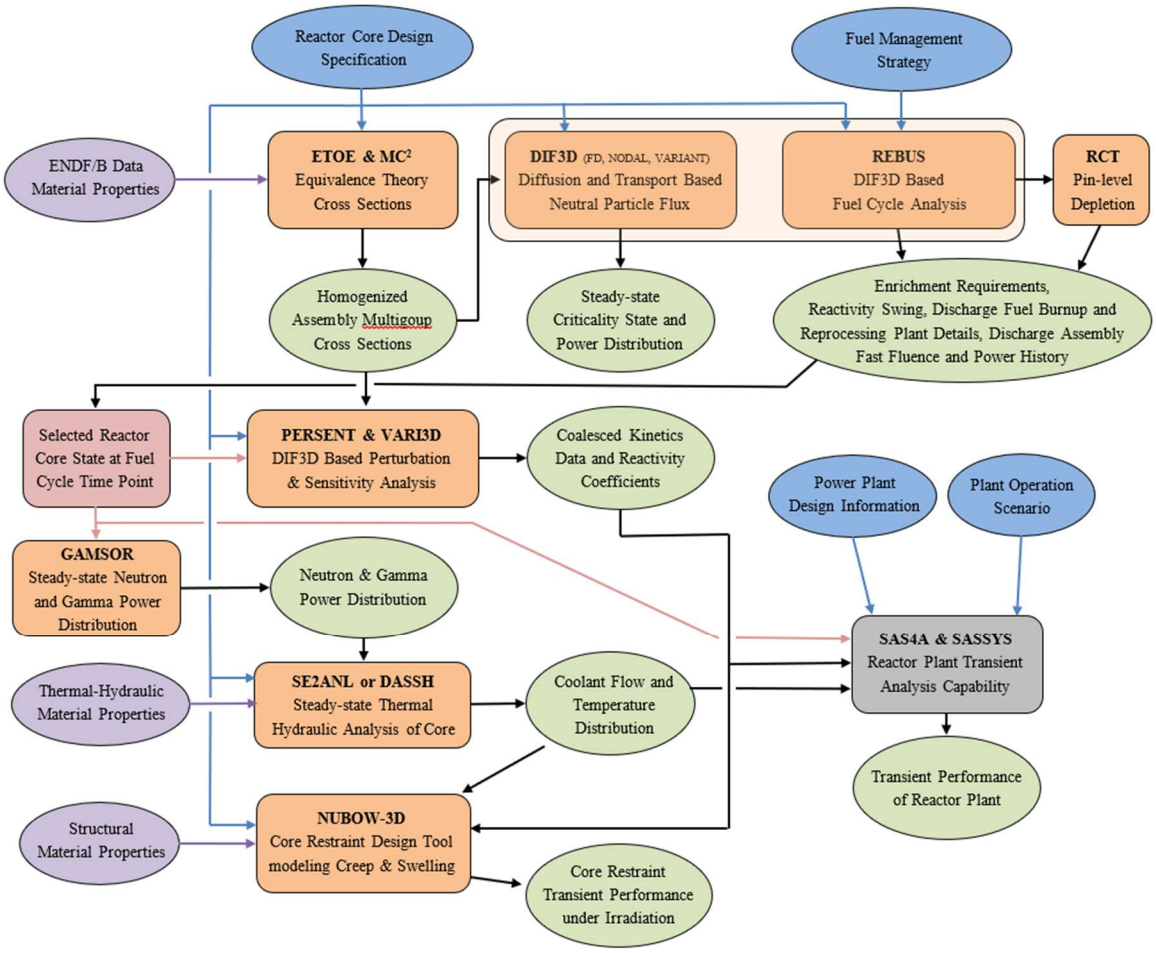

Figure 1.1 ARC Software Connections for Fast Spectrum Fuel Cycle Analysis

The ARC Software Quality Assurance (SQA) Program [13] is aimed to provide the controls and processes necessary to enable software improvement while meeting user and program sponsor requirements. The Software QA 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. REBUS is built around the DIF3D code and thus is geometrically limited to the models that can be constructed with DIF3D. 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. The DASSH software is exclusively connected to the DIF3D-VARIANT capability and uses the flux solution to reconstruct the radial and axial pin power shape. DASSH is also geometrically limited to hexagonal geometries. The code coverage assessment on DASSH will thus primarily consider the DASSH specific source code coverage and give details on the DIF3D components as secondary importance. The purpose of the present work is to identify a set of test problems for DASSH and document the code coverage of DASSH for those test problems. DASSH is distributed with a regression test suite and those are used for the code coverage assessment here.

In the following sections, the parts of the existing DASSH coding which are, or are not, touched by the set of test problems are documented. The coverage of the DIF3D components is included in a follow up section. 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 DASSH must be modified to create a special DASSH 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 DASSH. A review of the DASSH 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 DASSH 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. DASSH-F Overview

The DASSH thermal analysis code is designed to rapidly allow a reactor design engineer to obtain flow rates requirements that satisfy peak temperature constraints in the domain. The advantage of using DASSH over a hand calculation is that it has a more rigorous treatment of the pin power distribution and coolant heat transfer within an assembly and between assemblies. The advantage of using DASSH over a conventional 3D subchannel code or a computational fluid dynamics code (CFD) is that it can obtain the desired solution in a matter of minutes in serial with minor computer memory needs.

The DASSH methodology is virtually identical to SUPERENERGY-2 with additional functionalities taken from follow on work to SUPERENERGY-2 done at ANL in the 1980s. DASSH today is an integral component of the Argonne fast reactor analysis suite for reactor design work. DASSH obtains the power distribution from a coupled neutron-gamma heating calculation in DIF3D/GAMSOR at each time point of a companion fuel cycle analysis calculation with REBUS. The domain in DASSH assumes a hexagonal grid typical for fast reactors with much of the geometry information taken from the DIF3D model. DASSH assumes the assemblies that are loaded into each grid position are ducted to minimize crossflow between adjacent assemblies and thereby constrain the coolant flow unique to each assembly.

The DASSH output provides tables of evaluated material properties and key coolant and pin temperature results. DASSH can create Python scripts that generate domain summary pictures and DASSH can also generate assembly temperature maps and VTK output files which allow the DASSH solution to be visualized. As the primary purpose of the DASSH software is to compute the coolant and fuel pin temperature distribution for a given model of a reactor, much of the output focus is giving the user quick summary tables needed to assess the performance of a given orifice flow specification. The present version of DASSH has a crude orifice search capability and an efficient orifice flow search capability. The flow search tries to meet user specified constraints for 1) peak 2-sigma clad midwall temperature, 2) peak coolant temperature, and 3) desired bulk outlet temperature. All of the preceding features lead to considerable branching in the DASSH software and those branches and features that are not covered fully are detailed here.

3. DASSH Coverage Assessment

The driver of DASSH is a single subroutine in the directory */source* of the DASSH distribution. The bulk of the source code lies in the directory *source/library* and the structures unique to DASSH are stored in *source/Structures*. All the modules used in DASSH are listed in Table 3.1.

Table 3.1 Modules used by DASSH

Modules	Directory locations
DASSH driver	source
DASSH subroutines	source/library
DASSH structures	source/Structures
Fortran 90 modules	gamsor/dif3d/arcmodules/source
VARIANT basis	gamsor/dif3d/variantbasis/source
DIF3D to VTK	gamsor/dif3d/dif3dtovtk/source
LINPACK	gamsor/dif3d/linpack/source
Auxiliary ARC routines	gamsor/dif3d/arccommon/source

When DASSH is compiled many source files in each module (directory) are compiled and linked as part of the executable to avoid complicated compiling procedures but they are not necessarily used. One exception is LINPACK which DASSH only compiles the Fortran 90 versions in linpack.F90 and blas.F90. DASSH is distributed with 11 test cases. The first 5 test cases are designed to test out many of the different pin geometry settings in DASSH each of which contains more than 30 inputs each and were created as part of the development work to debug mistakes made in the source code. Tests 7 and 8 have a similar purpose but are for 120- and 60-degree periodic geometries. Tests 6, 9, 10, and 11 were added to test out the orifice search features and those also have more than one DASSH input. In total, there are 289 individual DASSH input files which should be sufficient to touch most of the DASSH software.

3.1 Primary subroutines of the DASSH code

The primary subroutines of the DASSH code are stored in the directories *source*, *source/library*, and *source/Structures*. The code coverage of all of the files in *source* and *source/library* are shown in Table 3.1.1 and Table 3.1.2. The code coverage details of all of the files in source/Structures are shown in Table 3.1.3 For the uncovered or partially-covered subroutines, a description of the uncovered parts is also given in the table. For the covered source files, the number of calls by the test package are also listed thus detailing the usage frequency of each source file. The only subroutine that gmon failed to report coverage on was dasshf_radialinterp_define which can be seen in Table 3.1.3.

Starting with the code coverage of the driver and primary subroutines in Table 3.1.1, one can see that many subroutines have partial coverage and most are denoted as needing no additional coverage. Many of these subroutines are missing coverage because of fatal error branches that are not reached with the test cases and a variety of debug output messages and warning messages on

bad inputs. For those files that are denoted as needing additional coverage, the feature that is not covered that is recommended to be covered is in bold font. The driver has the first example and one finds that the non-GAMSOR power input path is not covered and it is recommended to add a test case that covers this aspect. This is not the intended usage of DASSH, but since it is present, it would seem wise to verify that it is working properly.

The major concerns for code coverage are:

1) The three-sigma hot channel factor capability. This aspect is encountered in the dasshf_read_input and dasshf_solve_pindetails subroutines. This feature is known to work but additional testing needs to be put into place to ensure it is consistently maintained.
2) The ring_inlet_temp related branching is not tested. This aspect is encountered in the dasshf_read_input and dasshf_setup_assemblyij subroutines. It is not clear if this is being used or if it works.
3) Dummy pin related branching. The most important usage is in dasshf_solve_fuelpin which is not reached with the current test suite. This feature is of greatest importance for experiment assemblies and is not realistic for regular design work.
4) The duct gap flow fraction input feature. This is encountered in dasshf_read_input and dasshf_solvetempfield and is important as the existing method for determining the flow in the duct bypass gap is really a design adjustable parameter.
5) Clockwise wire wrap geometry setup (as opposed to counter-clockwise). It is not clear from the manual whether this capability is working and clearly testing is not presently included in the verification test suite.
6) The subzone setup for ZNATDN and NDXSRF is not presently tested in DASSH. Because there is specific coding in DASSH related to it, it should be tested.
7) The IsotopeMap file for choosing which isotopes are part of fuel, clad, and coolant is not tested. This is expected to be a key setup for advanced users and should be tested.

Beyond these seven major issues, the rest of the issues are relatively minor. As an example, many of the override inputs for manually setting the flow split, mixing, and swirl velocity are important features to have but unlikely to be used by a user. The define_hcf_direct falls under the same category as unlikely to be used. All of these types of issues will be addressed when software verification of DASSH is completed.

Continuing with the uncovered subroutines in source/library in Table 3.1.2, one can see that the subroutine dasshf_coolant_na is recommended to be removed as it is unused. Many of these uncovered subroutines are only used for debugging outputs and thus they can be ignored. The only ones of real concern are the D9 and SS316 material property subroutines. Inspection of the source code makes it very unlikely that they will cause a fatal error and if the material property evaluations are incorrect then the DASSH output can provide the user sufficient detail on that aspect. The fact

that the user can provide their own table of material properties should negate the importance of these subroutines with time.

The files in *source/Structures* detailed in Table 3.1.3 are found to have many cases of no coverage. All of the files are listed as No or Maybe. The Maybe designation was included as for good programming practice, these subroutines should be used although they clearly are not necessary for the software to function properly. Most of the remaining uncovered subroutines with a No designation are done because they are either presently unused features of the module or only used for debugging output which obviously does not occur with the current test suite.

All of the preceding lack of code coverage aspects will be dealt with when the software verification work for DASSH is completed in the next fiscal year. This report will be revised when that new verification test suite exists.

Table 3.1.1 Coverage or Partially Covered Files in the Directory *source* and *source/library*

File Name	Subroutine	# of calls by the test package	Coverage Status	Uncovered portion description	Further Coverage needed?
dassh	main	289	Partially	Fatal error branches, Debug sections, **Non-GAMSOR power input path**, 3D VTK output, write time step output to individual files,	Yes
DASSHF_ByPassGap_SCAdj	dasshf_bypassgap_scadj	2,772	Partially	Debug sections, **Case where adjacent assemblies have same number of meshes but different sizes**	Yes
DASSHF_Correlation_CTD	dasshf_correlation_ctd	55,550	Partially	No wire-wrap input setup and single pin assembly case	No
DASSHF_Correlation_KC	dasshf_correlation_kc	55,550	Fully		No
DASSHF_Correlation_MIT	dasshf_correlation_mit	55,550	Partially	No wire-wrap input setup	No
DASSHF_Correlation_NOV	dasshf_correlation_nov	55,550	Partially	No wire-wrap input setup	No
DASSHF_Corr_Spacer_CD	dasshf_corr_spacer_cd	24	Partially	Bad result warning message	No
DASSHF_Corr_Spacer_REMHE	dasshf_corr_spacer_remhe	4	Partially	Bad result warning message and out of bounds issue on Reynolds number	No
DASSHF_GetBondProp	dasshf_getbondprop	19,551	Partially	All branches for bond material that is not sodium including **user table**	Yes
DASSHF_GetCoolantProp	dasshf_getcoolantprop	115,587	Partially	All branches for coolant type other than sodium and user table	No
DASSHF_GetFuelProp	dasshf_getfuelprop	781,953	Partially	Branches for SS316 and D9	No
DASSHF_GetStructureProp	dasshf_getstructureprop	387,032	Partially	Branches for SS316 and D9	No
DASSHF_HeatTransferCoef	dasshf_heattransfercoef	156,680	Fully		No
DASSHF_Hex_Fill_PinAdj	dasshf_hex_fill_pinadj	37,845	Partially	Error branching	No
DASSHF_Hex_Fill_SCAdj	dasshf_hex_fill_scadj	37,845	Partially	Fatal error branches, Debug branching, and various sub-channel adjacency setups which are unreachable	No
DASSHF_ImportPropertyTable	dasshf_importpropertytable	8	Partially	Error branching and branching for specific named property tables	No

DASSHF_Output_AsciiArt	dasshf_output_asciiart	495	Partially	Error branching, Debug branching, **Auxiliary table output, branching for temperatures > 999**	Yes
	dasshf_output_aal	650,658	Partially	**Branching for temperatures > 999**	Yes
DASSHF_Output_Geom	dasshf_output_geom	289	Fully		No
DASSHF_Output_oldNUBOW	dasshf_output_oldnubow	4	Partially	Error branching, bad input branching	No
DASSHF_OutputPP_VTK_2D	dasshf_outputpp_vtk_2d	1	Partially	60 and 120 periodic geometry setups	No
DASSHF_Output_Print2D	dasshf_output_print2d_fwi	14	Fully		No
	dasshf_output_print2d_iwi	5	Fully		No
DASSHF_Output_SCTempTable	dasshf_output_sctemptable	359	Partial	Exclusion branching for printing only the driver assembly	No
DASSHF_Output_TimePeaks	dasshf_output_timepeaks	37	Partial	Three sigma output branching	No
DASSHF_Output_VTK_2D	dasshf_output_vtk_2d	11	Partial	Branch for missing .vtk in filename	No
DASSHF_PythonOut	dasshf_pythonout_header	6	Fully		No
	dasshf_pythonout_realarray	201	Partially	Branch for 60 and 120 periodic geometry setups	No
DASSHF_Read_Input	dasshf_read_input	289	Partially	Fatal error branching, input error branching, various alternative input order branching, branching for other coolant, clad, fuel types, branching for inputs time_output**,** override_dz, pin_power_only, display_iso_map, isotope_map, temp_map_table, vtk_3d_assembly, zone_list, **duct_gap_fract**, **corr_flow_manual**, **corr_frict_manual**, **corr_eps_manual**, **use_three_sigma**, **ring_inlet_temp define_hcf_direct**, **hcf_method**,	Yes
DASSHF_Setup_AsciiArt_Assem	dasshf_setup_asciiart_assem	5,893	Fully		No
DASSHF_Setup_AsciiArt	dasshf_setup_asciiart	289	Partially	Debug output, table output resize	No
DASSHF_Setup_AssemblyIJ	dasshf_setup_assemblyij	289	Partially	Fatal error branching, **ring_inlet_temp branch**, region list usage on assign_flow input	Yes

DASSHF_Setup_AssemblyPower	dasshf_setup_assemblypower	301	Partially	Fatal error branching, debug branching, bad input branching	No
DASSHF_Setup_CheckAssemblies	dasshf_setup_checkassemblies	289	Partially	Fatal error branching, branching for bad input setup, specific pin heat transfer coefficient selection	No
DASSHF_Setup_CheckFiles	dasshf_setup_checkfiles	289	Partially	Fatal error branching, different spatial order in NHFLUX and GHFLUX	No
DASSHF_Setup_GeometryConst	dasshf_setup_geometryconst	2,772	Partially	Fatal error and debug branching	No
DASSHF_Setup_Oriface	dasshf_setup_oriface	289	Partially	Fatal error and debug branching	No
DASSHF_Setup_Power	dasshf_setup_power	289	Partially	Fatal error, warning, and debug branching, isotope_map related branching,	No
DASSHF_Setup_Properties	dasshf_setup_properties	289	Partially	Fatal error branching and unused named property types	No
DASSHF_Setup_SubChannelMap	dasshf_setup_subchannelmap	289	Partially	Case where number of pins is the same but pitch is different	No
DASSHF_Setup_TempField	dasshf_setup_tempfield	289	Partially	Branch for enforcing minimal axial mesh size	No
DASSHF_Solve_FuelPin_Evalk	dasshf_solve_fuelpin_evalk	>161E6	Fully		No
DASSHF_Solve_FuelPin	dasshf_solve_fuelpin	18,130	Partially	Fatal error and debug branching, **dummy pin branching**	Yes
DASSHF_Solve_PinDetails	dasshf_solve_pindetails	359	Partially	Fatal error branching, **dummy pin branching**, vertical hot channel factor branch, **three-sigma flag** branching, 3D VTK output branch	Yes
DASSHF_SolveTempField	dasshf_solvetempfield	359	Partially	Fatal error and debug branching, large output table header handles, **duct gap flow fraction related branch**, axial mesh size constraint correction, print driver assembly only branching	Yes
DASSHF_SolveTempField_GetFixedC	dasshf_solvetempfield_getfixedc	3,294	Partially	Fatal error and debug branching, **clockwise swirl setup**	Yes
DASSHF_SolveTempField_H3D	dasshf_solvetempfield_h3d	3,294	Partially	Fatal error and debug branching and array overrun branching	No
DASSHF_SolveTempField_InitField	dasshf_solvetempfield_initfield	359	Fully		No
DASSHF_SolveTempField_Interpolate	dasshf_solvetempfield_interpolate	2,935	Partially	Warning messages for potential interpolation errors	No

DASSHF_SolveTempField_MatProp	dasshf_solvetempfield_matprop	359	Partially	Large header table handling	No
DASSHF_SolveTempField_PBMSolveFF	dasshf_solvetempfield_pbmsolveff	1,609	Partially	Fatal error branching	No
DASSHF_SolveTempField_PowerNorm	dasshf_solvetempfield_powernorm	359	Partially	Fatal error and debug branching	No
DASSHF_SolveTempField_THinfo	dasshf_solvetempfield_thinfo	359	Partially	Fatal error branching, low Reynolds number PBM friction factor branch, **manual flow split, mixing, and swirl velocity input branches**, large header table handling	Yes
DASSHF_SolveTempField_Zlimit	dasshf_solvetempfield_zlimit	359	Partially	One pin assembly case for low flow assemblies, axial mesh size correction	No
DASSHF_Update_FlowRate	dasshf_update_flowrate	47	Partially	Large header table handling, branching for alternate output results, convergence related branching	No
DASSHF_x_B4C	dasshf_x_b4c	2,322	Fully		No
DASSHF_x_HT9	dasshf_x_ht9	437,164	Fully		No
DASSHF_x_Metal	dasshf_x_metal	725,523	Partially	Fatal error and warning branching	No
DASSHF_x_Na	dasshf_x_na	134,914	Fully		No

Table 3.1.2 Uncovered Files in the Directory *source/library*

File Name	Subroutine	Description	Further Coverage needed?
DASSHF_Coolant_Na	dasshf_coolant_na	Unused alternative to dasshf_x_na, remove	No
DASSHF_Output_Print2D	dasshf_output_print2d	Debug printing	No
	dasshf_output_print2d_wi	Debug printing	No
	dasshf_output_print2d_index	Debug printing	No
	dasshf_output_print2d_wiw	Debug printing	No
DASSHF_Output_VTK_3D	dasshf_output_vtk_3d	3D single assembly plotting capability	No
DASSHF_x_D9	dasshf_x_d9	D9 structural steel material properties	No
DASSHF_x_SS316	dasshf_x_ss316	SS316 structural steel material properties	No

Table 3.1.3 Coverage Status of the Files in the Directory *source/Structures*

File Name	Subroutine	# of calls by the test package	Coverage Status	Uncovered portion description	Further Coverage needed?
DASSHF_AsciiArt	dasshf_asciiart_assignprintinfo	289	Fully		No
	dasshf_asciiart_define	2,772	Partially	Fatal error branching	No
	dasshf_asciiart_void	2,772	Partially	Fatal error branching	No
	dasshf_asciiart_print	0		Debug printing only	No
DASSH_AssemblyPower	dasshf_assemblypower_assignprintinfo	291	Fully		No
	dasshf_assemblypower_define	37,845	Partially	Fatal error branching	No
	dasshf_assemblypower_void	39,843	Partially	Fatal error branching	No
	dasshf_assemblypower_print	0		Debug printing only	No
	dasshf_assemblypower_print2	63	Partially	Warning branching	No
DASSHF_ByPassAdjacency	dasshf_bypassadj_assignprintinfo	289	Fully		No
	dasshf_bypassadj_define	2,772	Partially	Fatal error branching	No
	dasshf_bypassadj_copy	0		Unused feature	No
	dasshf_bypassadj_void	2,772	Partially	Fatal error branching	No
	dasshf_bypassadj_print	0		Debug printing only	No
DASSHF_GeometryConst	dasshf_geometryconst_assignprintinfo	289	Fully		No
	dasshf_geometryconst_define	2,772	Partially	Fatal error branching	No
	dasshf_geometryconst_void	2,772	Partially	Fatal error branching	No
	dasshf_geometryconst_print	0		Debug printing only	No
DASSHF_Input_Assembly	dasshf_input_assembly_assignprintinfo	0		Unused feature	Maybe
	dasshf_input_assembly_define	4,705	Partially	Fatal error branching	No
	dasshf_input_assembly_copy	2,306		Unused feature	No
	dasshf_input_assembly_void	2,318	Partially	Fatal error branching	No
	dasshf_input_assembly_print	0		Debug printing only	No
DASSHF_Input_Assign	dasshf_input_assign_assignprintinfo	289	Fully		No
	dasshf_input_assign_define	2,289	Partially	Fatal error branching	No
	dasshf_input_assign_copy	0		Unused feature	No
	dasshf_input_assign_void	0	Partially	Unused feature	Maybe
	dasshf_input_assign_print	0		Debug printing only	No
DASSHF_Input	dasshf_input_assignprintinfo	289	Fully		No

	dasshf_input_define	1,383	Partially	Fatal error branching	No
	dasshf_input_void	0		Unused feature	Maybe
	dasshf_input_print	0		Debug printing only	No
DASSHF_PinAdjacency	dasshf_pinadj_assignprintinfo	289	Fully	Fully	No
	dasshf_pinadj_define	37,845	Partially	Fatal error branching	No
	dasshf_pinadj_copy	0		Unused feature	No
	dasshf_pinadj_void	39,843	Partially	Fatal error branching	No
	dasshf_pinadj_print	0		Debug printing only	No
DASSHF_PinTemp	dasshf_pintemp_assignprintinfo	289	Fully		No
	dasshf_pintemp_define	57,159	Partially	Fatal error branching	No
	dasshf_pintemp_copy	0		Unused feature	No
	dasshf_pintemp_void	39,843	Partially	Fatal error branching	No
	dasshf_pintemp_print	0		Debug printing only	No
DASSHF_RadialInterp	dasshf_radialinterp_assignprintinfo	0		Unused feature	Maybe
	dasshf_radialinterp_define	>100,000	Partially	Fatal error branching	No
	dasshf_radialinterp_void	159,372	Partially	Fatal error branching	No
	dasshf_radialinterp_print	0		Debug printing only	No
	dasshf_radialinterp_buildm	1,680	Fully		No
DASSHF_SCAdjacency	dasshf_scadj_assignprintinfo	289	Fully		No
	dasshf_scadj_define	63,085	Partially	Fatal error branching	No
	dasshf_scadj_copy	0		Unused feature	No
	dasshf_scadj_void	65,083	Partially	Fatal error branching	No
	dasshf_scadj_print	0		Debug printing only	No
DASSHF_TableData	dasshf_tabledata_assignprintinfo	289	Fully		No
	dasshf_tabledata_define	608	Partially	Fatal error branching	No
	dasshf_tabledata_copy	0		Unused feature	No
	dasshf_tabledata_void	0		Unused feature	Maybe
	dasshf_tabledata_print	0		Debug printing only	No
	dasshf_tabledata_ordertable	8	Partially	Table not already in order case	Yes
	dasshf_tabledata_interpolate	4,200	Fully		No
DASSHF_TempSolution	dasshf_tempsolution_assignprintinfo	289	Fully		No
	dasshf_tempsolution_define	2,772	Partially	Fatal error branching	No
	dasshf_tempsolution_copy	0		Unused feature	No

	dasshf_tempsolution_void	2,772	Partially	Fatal error branching	No
	dasshf_tempsolution_print	0		Debug printing only	No
DASSHF_WallTempInterp	dasshf_walltempinterp_assignprintinfo	289	Fully		No
	dasshf_walltempinterp_define	4,162	Partially	Fatal error branching	No
	dasshf_walltempinterp_copy	0		Unused feature	No
	dasshf_walltempinterp_void	39,843	Partially	Fatal error branching	No
	dasshf_walltempinterp_print	0		Debug printing only	No

4. DIF3D Coverage Update

The preceding DASSH subroutine coverage is the primary focus on this work. Because some parts of DIF3D were not fully covered in the DIF3D code coverage report [14], some aspects of the DIF3D code coverage is reiterated here. Of particular importance are the inclusion of source files in the DASSH distribution directory *gamsor/dif3d/arcmodules/source* and *gamsor/dif3d/arccommon/source*.

4.1 Source code in the directory gamsor/dif3d/arcmodules/source

The Fortran coding in the directory *gamsor/dif3d/arcmodules/source* of the DASSH 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 DASSH but the code coverage summary of all files is provided here in Table 4.1.1.

From Table 4.1.1, one can see that the components ASSIGNMENT, Basic, GEODST, LABELS, MATERIALS, NDXSRF, NE_FreeForm, NE_Kind, NE_VTKexport, NHFLUX, NonZeroStorage, PMATRX, and ZNATDN are all used in DASSH. There are several other components in this directory that are compiled and usable by DASSH and some not usable because they simply appear in other parts of the source code (DIF3D). There were many cases where gmon failed to report the number of calls to a subroutine and thus the values provided are estimated which are highlighted in the table.

With respect to code coverage, these are common use modules, and it is difficult to argue that DASSH is 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. The lack of code coverage that are relevant for DASSH include:

1) The sub-zone branching in the NDXSRF module that is covered in PERSENT.
2) The lack of coverage on the ISOTXS module occurs because the alternative power setup in DASSH is not used.
3) The lack of coverage in the NDXSRF_ReadAssignIsotope which was also highlighted earlier as a DASSH user can provide the very input that this subroutine is responsible for reading.

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

Table 4.1.1 Coverage Assessment of Files in the Directory *gamsor/dif3d/arcmodules/source*

File Name	Subroutine	# calls by test suite	Coverage Assessment	Further Coverage
ALIAS.F90	ALIAS_Define			
	ALIAS_IdentifyDestination			
	ALIAS_IdentifyTarget			
	ALIAS_Print			
	ALIAS_SetUnit			
	ALIAS_Void			
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			
ARC_EasyAtomDensity.F90	ARC_EasyAtomDensity			
ARC_EasyDLAYXS.F90	ARC_EasyDLAYXS			
ARC_EasyISOTXS.F90	ARC_EasyISOTXS			
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	6	Missing fatal error branches	No
	ASSIGNMENT_IdentifyMaterial	932	Fully covered	No
	ASSIGNMENT_Print			
	ASSIGNMENT_SetUnit	289	Fully covered	No
	ASSIGNMENT_Void			
Basic_Abort.F90	Basic_Abort			
Basic_AbortInvalidInt.F90	Basic_AbortInvalidInt			
Basic_AbortWithMessage.F90	Basic_AbortWithMessage			
Basic_CheckError.F90	Basic_CheckError	18,752	Missing fatal error branch	No
ChannelInput.F90	ChannelInput_AppendRegion			
	ChannelInput_Copy			
	ChannelInput_Define			
	ChannelInput_Print			
	ChannelInput_SetUnit			

	ChannelInput_Void			
COMMARA.F90	COMMARA_AssignPrintInfo			
	COMMARA_Copy			
	COMMARA_Define			
COMPXS_IMPORT.F90	COMMARA_Import			
COMMARA.F90	COMMARA_Print			
	COMMARA_Void			
COMPXS.F90	COMPXS_ADDSCALE			
	COMPXS_ASSIGNPRINTINFO			
	COMPXS_BALANCEXS			
	COMPXS_COPY			
	COMPXS_DEFINE			
COMPXS_EXPORT.F90	COMPXS_EXPORT			
COMPXS.F90	COMPXS_FullShift_For_IHS			
COMPXS.F90	COMPXS_IMPORT			
COMPXS_Mass.F90	COMPXS_Mass			
COMPXS_Mass_by_HABSID.F90	COMPXS_Mass_by_HABSID			
COMPXS_Modify_For_IHS.F90	COMPXS_Modify_For_IHS			
	COMPXS_Modify_For_IHS_fails			
COMPXS.F90	COMPXS_PRINT			
COMPXS_Modify_For_IHS.F90	COMPXS_Test_GS_solve			
COMPXS.F90	COMPXS_UPDATEPROPERTIES			
	COMPXS_VOID			
	COMPXS_ZERO			
DIF3D.F90	DIF3D_ALLOCATE			
	DIF3D_ASSIGNPRINTINFO			
	DIF3D_DEFINE			
DIF3D_EXPORT.F90	DIF3D_EXPORT			
DIF3D.F90	DIF3D_FLUXBYPASS_OFF			
	DIF3D_FLUXBYPASS_ON			
DIF3D_IMPORT.F90	DIF3D_IMPORT			
DIF3D_MakeADIF3D.F90	DIF3D_MakeADIF3D			
DIF3D.F90	DIF3D_PRINT			
	DIF3D_RESTOREPRINTING			
	DIF3D_TURNOFFPRINTING			
	DIF3D_VOID			
	DIF3D_VOID_DIF3DTYPE			
DLAYXS.F90	DLAYXS_Copy			

DLAYXS.F90	DLAYXS_Define			
DLAYXS_Export.F90	DLAYXS_Export			
DLAYXS_Export_ASCII.F90	DLAYXS_Export_ASCII			
DLAYXS_Export_BINARY.F90	DLAYXS_Export_Binary			
DLAYXS_Import.F90	DLAYXS_Import			
DLAYXS_Import_ASCII.F90	DLAYXS_Import_ASCII			
DLAYXS_Import_BINARY.F90	DLAYXS_Import_Binary			
DLAYXS_Map_to_ISOTXS.F90	DLAYXS_Map_to_ISOTXS			
DLAYXS.F90	DLAYXS_Print			
	DLAYXS_SetUnit			
	DLAYXS_Void			
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	289	Fully covered	No
	GEODST_ComputeAverageThickness			
	GEODST_ComputeCentroids			
	GEODST_Copy	602	Missing NRASS=0 related branch	No
	GEODST_DEFINE	1,204	Missing fatal error branching and debug printing	No
GEODST_EXPORT.F90	GEODST_EXPORT			
GEODST_EXPORT_ASCII.F90	GEODST_EXPORT_ASCII			
GEODST_EXPORT_BINARY.F90	GEODST_EXPORT_BINARY			
GEODST_IMPORT.F90	GEODST_IMPORT	301	Missing ASCII file branch	No
GEODST_IMPORT_ASCII.F90	GEODST_IMPORT_ASCII			
GEODST_EXPORT.F90	GEODST_IMPORT_BINARY	301	Missing fatal error branches, debug print branches, non-hexagonal branches, NRASS=1 branch	No
GEODST.F90	GEODST_LocatePoint			

	GEODST_MeshVolumes	301	Missing non-hexagonal branches	No
	GEODST_NODAL_factors	301	Missing fatal error branch	No
	GEODST_NRASS1	602	Missing non-hexagonal branches	No
	GEODST_PRINT			
	GEODST_RegionsOnBoundary			
	GEODST_RingPositionIJ	8,988	Missing fatal error branching and invalid input branch	No
	GEODST_VOID	903	Missing fatal error branching	No
ISOTXS_Close.F90	ISOTXS_Close			
ISOTXS_Collect_Info.F90	ISOTXS_Collect_Info			
ISOTXS_Create.F90	ISOTXS_Create			
ISOTXS_GetIsotope.F90	ISOTXS_GetIsotope			
ISOTXS_Interface.F90	ISOTXS_Interface_Copy			
	ISOTXS_Interface_Define			
	ISOTXS_Interface_IdentifyIsotope			
	ISOTXS_Interface_Print			
	ISOTXS_Interface_SetUnit			
	ISOTXS_Interface_Void			
ISOTXS_Isotope.F90	ISOTXS_Isotope_CalculateNOP			
	ISOTXS_Isotope_Collapse			
	ISOTXS_Isotope_ComputeBanding			
	ISOTXS_Isotope_ComputeMAXORD			
	ISOTXS_Isotope_Copy			
	ISOTXS_Isotope_Copy_AlterNSCMAX			
	ISOTXS_Isotope_Define			
	ISOTXS_Isotope_Merge			
	ISOTXS_Isotope_N2NFactor			
	ISOTXS_Isotope_NegativeDefinite			
	ISOTXS_Isotope_Print			
	ISOTXS_Isotope_SetUnit			
	ISOTXS_Isotope_Void			
ISOTXS_Modify_Isotope.F90	ISOTXS_Modify_Isotope			
ISOTXS_Modify_Library.F90	ISOTXS_Modify_Library			
ISOTXS_Open.F90	ISOTXS_Open			
ISOTXS_PutIsotope.F90	ISOTXS_PutIsotope			
ISOTXS_ReadIsotope.F90	ISOTXS_ReadIsotope			

ISOTXS_Rewind.F90	ISOTXS_Rewind			
ISOTXS_Unique_HABSID.F90	ISOTXS_Unique_HABSID			
ISOTXS_WriteIsotope.F90	ISOTXS_WriteIsotope			
LABELS.F90	LABELS_AssignPrintInfo	289	Fully covered	No
	LABELS_Copy			
	LABELS_Define	301	Missing fatal error branches, debug print branches	No
LABELS_EXPORT.F90	LABELS_EXPORT			
LABELS.F90	LABELS_IdentifyArea			
	LABELS_IdentifyRegion	940	Fully covered	No
	LABELS_IdentifyZone			
LABELS_IMPORT.F90	LABELS_IMPORT	301	Missing fatal error branches, debug print branches	No
LABELS_Integrate.F90	LABELS_Integrate			
	LABELS_IntegrateMD			
	LABELS_PRINT			
LABELS_Integrate.F90	LABELS_Void	301	Missing fatal error branch	No
MATERIALS.F90	MATERIALS_Append			
	MATERIALS_CheckProperties			
	MATERIALS_Copy	20	Missing structure already defined branch	No
	MATERIALS_Define	26	Missing fatal error branches	No
	MATERIALS_EliminateDuplicates			
	MATERIALS_Normalize			
	MATERIALS_Print			
	MATERIALS_SetUnit	289	Fully covered	No
	MATERIALS_Void	10	Missing fatal error branches	No
NDXSRF.F90	NDXSRF_ASSIGNPRINTINFO	289	Fully covered	No
	NDXSRF_Copy			
	NDXSRF_DEFINE	301	Missing fatal error branches, debug print branches, subzone setup of structure	No
NDXSRF_EXPORT.F90	NDXSRF_EXPORT			
NDXSRF_IMPORT.F90	NDXSRF_IMPORT	301	Missing fatal error branches, debug print branches, subzone setup of structure	No
NDXSRF.F90	NDXSRF_PRINT			
NDXSRF_ReadAssignIsotope.F90	NDXSRF_ReadAssignIsotope	301	Effectively uncovered	Yes

NDXSRF.F90	NDXSRF_VOID	301	Missing fatal error branches, debug print branches	No
NE_FreeForm.F90	NE_FreeForm_ADDWORD	>14,000	Missing fatal error branches, bad input cases, and special case string handling	No
	NE_FreeForm_FINDWORD	14,731	Missing fatal error branches, bad input cases, and special case string handling	No
	NE_FreeForm_CLEAR		Fully covered	No
	NE_FreeForm_Decomposition	23,891	Bad Input branching	No
	NE_FreeForm_PRINT			
	NE_FreeForm_SetUnit			
NE_Kind.F90	NE_Kind_AddUnderscores			
	NE_Kind_AlphabetInteger			
	NE_Kind_DIF3DCharacterFix	3,990,104	Fully covered	No
	NE_Kind_FREELOGICALUNIT	2,425	Fully covered	No
	NE_Kind_GetFreeLogicalUnit	3,032	Missing fatal error branches	No
	NE_Kind_IntegerAlphabet			
	NE_Kind_IntToLogical			
	NE_Kind_LogicalToInt			
	NE_Kind_SetUnit			
	NE_Kind_StandardCharacterFix	39,014	Fully covered	No
	NE_Kind_UpperCaseString	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	33	Missing fatal error branches	No
	NE_VTKexport_ALLOCATE_Block	>33	Missing fatal error branches, debug output branches	No
	NE_VTKexport_Define	33	Fully covered	No
	NE_VTKexport_Define_Block	12,593	Fully covered	No
	NE_VTKexport_DefineMapping	37,779	Missing unknown element type branch	No
	NE_VTKexport_EXPORT	33	Missing fatal error branches	No
	NE_VTKexport_EXPORT_ASCII_VTK	>33	Missing fatal error branches, parallel output branch	No
	NE_VTKexport_PRINT			
	NE_VTKexport_Print_Block			
	NE_VTKexport_Print_VTKdata			
	NE_VTKexport_Void	33	Missing optional block input case	No
	NE_VTKexport_Void_VTKdata	>33	Missing fatal error branch	No
	NE_VTKexport_Void_block_VTKdata	33	Missing fatal error branch	No
NHFLUX_ActiveNodeAdjacency.F90	NHFLUX_ActiveNodeAdjacency	289	Missing fatal error branches, debug print branches	No
NHFLUX.F90	NHFLUX_ASSIGNPRINTINFO	289	Fully covered	No
	NHFLUX_DEFINE	903	Missing fatal error branches, debug print branches	No
NHFLUX_EvaluateFlux.F90	NHFLUX_EvaluateFlux			
NHFLUX_EXPORT.F90	NHFLUX_EXPORT			
NHFLUX.F90	NHFLUX_FindActiveNode			
NHFLUX_GeometryMaps.F90	NHFLUX_GeometryMaps	301	Missing fatal error branches	No
NHFLUX_IMPORT.F90	NHFLUX_IMPORT	602	Missing fatal error branches, debug print branches, adjoint	No

			flux branch, previously defined structure	
NHFLUX.F90	NHFLUX_INTEL_IDEAL_OPEN			
	NHFLUX_PRINT			
NHFLUX_RingSectorPosition.F90	NHFLUX_RingSectorPosition	289	Missing fatal error branches	No
NHFLUX_SetupVARSRC.F90	NHFLUX_SetupVARSRC			
NHFLUX.F90	NHFLUX_VOID	903	Missing fatal error branches	No
NHSTRAIN.F90	NHSTRAIN_ASSIGNPRINTINFO			
	NHSTRAIN_DEFINE			
NHSTRAIN_Export_to_MCNP_Hex.F90	NHSTRAIN_EtoMCNP_Hex_Wedges			
NHSTRAIN.F90	NHSTRAIN_EvalPoints			
NHSTRAIN_Export.F90	NHSTRAIN_Export			
NHSTRAIN_Export_to_MCNP.F90	NHSTRAIN_Export_to_MCNP			
NHSTRAIN_Export_to_MCNP_Cart.F90	NHSTRAIN_Export_to_MCNP_Cart			
NHSTRAIN_Export_to_MCNP_Hex.F90	NHSTRAIN_Export_to_MCNP_Hex			
NHSTRAIN_Import.F90	NHSTRAIN_Import			
NHSTRAIN.F90	NHSTRAIN_PRINT			
	NHSTRAIN_SPLIT			
	NHSTRAIN_VOID			
NonZero_StoreMatrix.f90	NonZero_StoreMatrix			
NonZeroStorage_D.f90	NonZeroStorage_D_Allocate		Missing fatal error branches, debug print branches	No
	NonZeroStorage_D_CopyFromNZRWS	8,324	Missing fatal error branches	No
	NonZeroStorage_D_CopyFromNZS			
	NonZeroStorage_D_Define		Fully covered	No
	NonZeroStorage_D_Print			
	NonZeroStorage_D_Print_Type			
	NonZeroStorage_D_SetUnit			
	NonZeroStorage_D_Void	8,324	Fully covered	No
	NonZeroStorage_D_Void_Type		Missing fatal error branches	No
NonZeroRowWiseStorage_D.f90	NZRowWiseStorage_D_Define	8,324	Fully covered	No
	NZRowWiseStorage_D_Fill	56,498	Missing fatal error branches, duplicated column storage	No
	NZRowWiseStorage_D_Print			
	NZRowWiseStorage_D_Reset			
	NZRowWiseStorage_D_SetUnit			
	NZRowWiseStorage_D_Void	8,324	Fully covered	No
	NZRWS_D_Allocate_RowWiseData		Missing fatal error branches	No

	NZRWS_D_Allocate_Type		Missing fatal error branches	No
	NZRWS_D_Print_Row			
	NZRWS_D_Print_Type			
	NZRWS_D_Void_RowWiseData		Missing fatal error branches	No
	NZRWS_D_Void_Type		Missing fatal error branches	No
PMATRX.F90	PMATRX_ASSIGNPRINTINFO			
	PMATRX_DEFINE	289	Missing fatal error branches, debug print branches, branching for composition region input	No
PMATRX_EXPORT.F90	PMATRX_EXPORT_BINARY			
PMATRX_IMPORT.F90	PMATRX_IMPORT_BINARY	289	Missing fatal error branches, debug print branches	No
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	289	Fully covered	No
	ZNATDN_Copy			
	ZNATDN_DEFINE	301	Missing fatal error branches, debug print branches	No
ZNATDN_EXPORT.F90	ZNATDN_EXPORT			
ZNATDN_IMPORT.F90	ZNATDN_IMPORT	301	Missing fatal error branches, branch for previously defined structure	No
ZNATDN.F90	ZNATDN_MergeWithFactors			
	ZNATDN_PRINT			
	ZNATDN_VOID	301	Missing fatal error branches	No

4.2 Source code in the directory gamsor/dif3d/variantbasis/source

The directory *gamsor/dif3d/variantbasis/source* in the DASSH 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 4.2.1 compared against the DIF3D code coverage.

Because DIF3D is the primary use of this module, one should focus on improving the code coverage there rather than in DASSH, except for the unique parts that DASSH uses. One such example of this are the subroutines BreakDown_Input_ID.F90 and GetSpaceBasis.F90 which are not used by DIF3D. Unlike DIF3D, DASSH only needs the spatial basis and not the various matrices which is why most of the subroutines in this module have no coverage. The uncovered or partially covered routines listed here do not need more coverage with respect to its usage in DASSH.

Table 4.2.1 Relative Coverage Details of Files in the Directory *gamsor/dif3d/variantbasis/source* to DIF3D

File Name	Subroutine	# of calls by the DIF3D tests	# of calls by the DASSH tests
BreakDown_Input_ID	BreakDown_Input_ID	0	1,180
Build_Hmatrix.f90	Build_Hmatrix	443	
CalculateGamma.f90	CALCULATEGAMMA	94	
CheckSpatialApprox.f90	CHECKSPATIALAPPROX	39	
CheckSurfaceRank.f90	CHECKSURFACERANK	39	
Common_GetSurfacesPerNode.f90	GETSURFACESPERNODE	39	
Common_LinearAlgebra.f90	MAS_MxV	6,772	
	MAS_VpCxV	88,126	2,014,908
	MAS_CxV	0	
	MAS_Mt	70	
	MAS_MxMt	431	
	MAS_VcrossV	431	
	MAS_VdotV	658,018	2,014,908
	VECTORMULTIPLY3	0	
	VECTORMULTIPLY4	23,858	
	VECTORMULTIPLY5	76	
	MAS_MxM	273	
	MAS_MtxM	262	
	MAS_VCOPY	756	
Common_SimpleFunctions.f90	DCLEANMATRIX	8,838	1,457
	ICLEANMATRIX	301	578
	DISPLAYMATRIX	0	
	MAS_XtoN	755,074	1,085,676
	OBTAINFACTORIAL	398,472	
	DGETMIN	0	
	DGETMAX	0	
	IGETMIN	20	
	IGETMAX	464	
	DNORMALIZE	0	
	CHECKWITHIN	0	
	DSIMPLESUM	0	
	DREMOVEZEROS	189	
	DCOUNTZEROS	0	
	DFILLVECTOR	0	
	DMIKESORT	0	

ConstructSpaceAngleMatrices.f90	ConstructSpaceAngleMatrices	39	
GetSpaceBasis.f90	GetSpaceBasis		578
GQ00_GetSnQuadPoints.f90	GETSNQUADPOINTS	227	
GQ00_GrabSnQuad.f90	GRABSNQUAD	227	
GQ01_OneDChebyInt.f90	ONEDCHEBY	182	
GQ01_OneDGaussInt.f90	ONEDGAUSS	227	
Pn01_DefineEven_RMA.f90	DEFINEEVEN_RMA	0	
Pn01_DefineOdd_RMA.f90	DEFINEODD_RMA	43	
Pn01_Obtain_H_V_E.f90	OBTAIN_H_V_E	30	
Pn01_Obtain_K_Vacuum.f90	OBTAIN_K_VACUUM	0	
Pn01_Obtain_L_Vacuum.f90	OBTAIN_L_VACUUM	76	
Pn01_Obtain_PnE.f90	OBTAIN_PnE	76	
Pn02_DefineReductionMatrix.f90	DEFINEREDUCTIONMATRIX	149	
	WonSikConstant	0	
Pn02_IndexYLMTerms.f90	INDEXYLMTERMS	149	
Pn02_IntegrateEmatrices.f90	INTEGRATE_EMATRICES	133	
	INTEGRATE_EMATRIX	>266	
Pn02_IntegrateHmatrices.f90	INTEGRATE_HMATRICES	9	
	INTEGRATE_HMATRIX	>18	
Pn02_IntegrateIdentities.f90	INTEGRATE_IDENTITIES	0	
Pn02_IntegrateVacuum.f90	INTEGRATE_VACUUM	94	
Pn02_IntegrateVmatrices.f90	INTEGRATE_VMATRICES	30	
	INTEGRATE_VMATRIX	>60	
Pn02_ObtainHmatrices.f90	OBTAIN_HMATRICES	30	
Ritz01_Obtain_P_U_D_W.f90	OBTAIN_P_U_D_W	39	
Ritz02_ObtainSurfaceDMatrix.f90	OBTAINSURFACEDMATRIX	262	
Ritz02_ObtainSurfaceWMatrix.f90	OBTAINSURFACEWMATRIX	0	
Ritz02_ObtainSurfaceFMatrix.f90	OBTAINSURFACEFMATRIX	0	
Ritz02_ObtainVolumePaltMatrix.f90	OBTAINVOLUMEPaltMATRIX	0	
Ritz02_ObtainVolumePMatrix.f90	OBTAINVOLUMEPMATRIX	281	
Ritz02_ObtainVolumeUMatrix.f90	OBTAINVOLUMEUMATRIX	103	
Ritz03_GetDSurfaceIntegrands.f90	GETdSURFACEINTEGRANDS	262	
Ritz03_GetPVolumeIntegrands.f90	GETpVOLUMEINTEGRANDS	0	
Ritz03_GetUVolumeIntegrands.f90	GETuVOLUMEINTEGRANDS	103	
Ritz03_GetWSurfaceIntegrands.f90	GETwSURFACEINTEGRANDS	0	
RitzOS00_GetNumberRitzTerms.f90	GETNUMBERRITZTERMS	457	578
	RET_GETNUMBERRITZTERMS	0	578
RitzOS00_GetSurfaceExpansionOrders.f90	GETSURFACEEXPANSIONORDERS	524	

RitzOS00_RitzOrthogonalSet.f90	RITZORTHOGONALSET	301	578
RitzOS01_GetPolyExponents.f90	GETPOLYEXPONENTS	301	578
RitzOS01_GetVolumeIntegrands.f90	GETVOLUMEINTEGRANDS	301	879
RitzOS02_Int_CartesianNode.f90	INT_CARTESIANNODE	39,698	8,428
RitzOS02_Int_EquiTriZNode.f90	INT_EQUITRIZNODE	0	
RitzOS02_Int_HexZNode.f90	INT_HEXZNODE	459,811	2,063,460
RitzOS03_EquiTriSurfaceW.f90	EQUITRISURFACEW	0	
RitzOS03_GetRitzPolyCoef.f90	GETRITZPOLYCOEF	301	578
RitzOS03_HexSurfaceW.f90	HEXSURFACEW	24,460	
RitzOS03_SimpleEquiTriSurf.f90	SIMPLEEQUITRISURF	0	
RitzOS03_SimpleHexSurf.f90	SIMPLEHEXSURF	422,113	2,063,460
RitzOS03_SimpleResult.f90	SIMPLERESULT	537,594	2,071,888
Ritz_PeriodicSpace.f90	Ritz_PeriodicSpace	52	
SPn00_GetSPnTerms.f90	GETSPNTERMS	151	
	RET_GETSPNTERMS	0	
SPn01_ObtainSPn_K_Vacuum.f90	OBTAINSPn_K_VACUUM	0	
SPn01_ObtainSPn_L_Vacuum.f90	OBTAINSPn_L_VACUUM	18	
SPn01_Obtain_H_E.f90	OBTAIN_H_E	9	
SPn01_Obtain_SPnE.f90	OBTAIN_SPnE	18	
SPn02_EvaluateSPn.f90	EVALUATE_SPN	45	
SPn03_EvaluatePn.f90	EVALUATEPN	182	
YLM00_AngularSurfPerNode.f90	ANGULARSURFPERNODE	30	
YLM00_GetEvenParityTerms.f90	GETEVENPARITYTERMS	327	
	RET_GETEVENPARITYTERMS	10	
YLM00_GetOddParityTerms.f90	GETODDPARITYTERMS	297	
	RET_GETODDPARITYTERMS	10	
YLM00_Obtain_YLM_Mu.f90	OBTAIN_YLM_MU	182	
YLM01_AngularRotations.f90	ANGULARROTATIONS	182	
YLM01_EvaluateYlm.f90	EVALUATEYLM	182	
YLM02_DefineSurfaceMapping.f90	DEFINESURFACEMAPPING	182	
YLM02_EvaluateYLMSurface.f90	EVALUATEYLM_SURFACE	255	
YLM03_EvaluateAssociatedPn.f90	EVALUATEASSOCIATEDPN	10,932	
YLM03_FormYLM3d.f90	FORM_YLM3D	10,932	
YLM04_GetColumnP.f90	GETCOLUMNP	50,728	

4.3 Source code in the directory gamsor/dif3d/arccommon/source

The directory *gamsor/dif3d/arccommon/source* in the DASSH distribution contains some common high-level functions used to replace large sections of the original DIF3D code. For DASSH, the VARPOW subroutine and QuickSort functions are the only parts compiled and linked. These subroutines are not used in DIF3D or REBUS and thus the code coverage here is done assuming that DASSH owns these subroutines with the code coverage summary shown in Table 4.3.1.

As can be seen, the QuickSort functions are not used by DASSH and are only present because some other linked source code requires them to be compiled and linked. The VARPOW subroutine is used and the uncovered sub-zone branch is the most important to update. Thus when the DASSH specific source code coverage for a problem with subzones is addressed, the code coverage assessment here should also be updated. The various other fuel, coolant, and structure type is no longer relevant as the preferred approach is to use the isotope map file detailed earlier to be uncovered as part of the NDXSRF related files in *gamsor/dif3d/arcmodules/source*.

Table 4.3.1 Coverage Status of Files in the Directory *gamsor/dif3d/arccommon/source*

File Name	Subroutine	# of calls by the test package	Coverage Status	Uncovered portion description	Further Coverage needed?
QuickSort_R64_MD.F90	QuickSort_R64_MD_SetUnit	0		Unused	Yes
	QuickSort_Sort_R64_MD	0		Unused	Yes
	QuickSort_Sort_R64_MD_NI	0		Unused	No
	QuickSort_R64_MD_Driver	0		Unused	No
VARPOW_Generate.F90	VARPOW_Generate	301	Partially	Fatal error branching, alternative fuel and coolant isotope selection setups, print out of the isotope map, **subzone branch for atom densities**, zero energy bound in ISOTXS, seven component power output	Yes

4.4 Source code in the directory gamsor/dif3d/linpack/source

The files in directory *gamsor/dif3d/linpack/source* in the DASSH distribution was collaboratively developed and some work originated at ANL. LINPACK contains some useful matrix manipulation capabilities but virtually none are used in DIF3D. For DASSH, only the functions linpack.F90 and blas.F90 are compiled and thus only those subroutines are listed in the code coverage assessment in Table 4.4.1. Because neither DIF3D or REBUS use these, the details on the code coverage are unique to DASSH. As seen, most of the lack of code coverage deals with erroneous inputs to the subroutines or input options that are not used by DASSH. The remaining lack of coverage on the other subroutines can be ignored as as they are not reachable in (used by) DASSH.

Table 4.4.1 Coverage Status of Files in the Directory *gamsor/dif3d/linpack/source*

File Name	Subroutine	# of calls by the test package	Coverage Status	Uncovered portion description
blas1.F90	linpack_daxpy	1,036,071,836	Partially	Unequal increments
	linpack_dcopy			
	linpack_drot			
	linpack_drotg			
	linpack_drotm			
	linpack_drotmg			
	linpack_dscal	215,763,904	Partially	Unequal increments
	linpack_dswap			
	linpack_xerbla			
linpack.F90	linpack_dchdc			
	linpack_dchdd			
	linpack_dchex			
	linpack_dchud			
	linpack_dgbco			
	linpack_dgbdi			
	linpack_dgbfa			
	linpack_dgbsl			
	linpack_dgeco			
	linpack_dgedi			
	linpack_dgefa	53,312,084	Partially	Error branching
	linpack_dgesl	53,320,484	Partially	Transpose solve feature
	linpack_dgtsl			
	linpack_dpbco			
	linpack_dpbdi			
	linpack_dpbfa			
	linpack_dpbsl			
	linpack_dpoco			
	linpack_dpodi			
	linpack_dpofa			

	linpack_dposl			
	linpack_dppco			
	linpack_dppdi			
	linpack_dppfa			
	linpack_dppsl			
	linpack_dptsl			
	linpack_dqrdc			
	linpack_dqrsl			
	linpack_dsico			
	linpack_dsidi			
	linpack_dsifa			
	linpack_dsisl			
	linpack_dspco			
	linpack_dspdi			
	linpack_dspfa			
	linpack_dspsl			
	linpack_dsvdc			
	linpack_dtrco			
	linpack_dtrdi			
	linpack_dtrsl			

4.5 Source code in the directory gamsor/dif3d/dif3dtovtk/source

The files in directory *gamsor/dif3d/dif3dtovtk/source* of the DASSH distribution were developed to provide a 3D visualization capability of the DIF3D code. It primarily contains routines which translate the GEODST file to a finite element mesh viewable with compatible visualization packages. For DASSH, these routines are presently compiled and linked but are not actually used in the DASSH source code. This is because DASSH has custom mesh generators appropriate for its geometry which were detailed in the code coverage section shown earlier. Table 4.5.1 provides the list of subroutines that are contained in the source directory and details the purpose of each subroutine. Because these subroutines are not called by DASSH, the lack of coverage can be ignored at this time and these subroutines should be removed from the makefile.

Table 4.5.1 Uncovered Files in the Directory *gamsor/dif3d/dif3dtovtk/source*

File Name	Subroutine	Description
Create_VTK_Cell_Based.F90	Create_VTK_Cell_Based	Main call path to translate GEODST to a VTK file
Create_VTK_Cell_Based_HEX.F90	Create_VTK_Cell_Based_HEX	Hexagonal 2D and 3D VTK file generator
Create_VTK_Cell_Based_ThetaRZ.F90	Create_VTK_Cell_Based_ThetaRZ	Theta-RZ VTK file generator
Create_VTK_Cell_Based_TriZ.F90	Create_VTK_Cell_Based_TriZ	Triangular-Z VTK file generator
Create_VTK_Cell_Based_XYZ.F90	Create_VTK_Cell_Based_XYZ	Cartesian XY and XYZ file generator

5. Summary and Conclusions

In the preceding analysis the coverage report for each submodule of DASSH was given. The submodule refers to separable libraries (LINPACK, LAPACK, etc.) that are included either partially or fully as part of DASSH even though the development of the source code was not specific to DASSH.

For the DASSH specific source code, most of the subroutines were fully or partially covered to the point where additional testing is not necessary. However, seven features were identified that should be addressed for code coverage which include the 3-sigma hot channel factor capability, the ring_inlet_temp input, the dummy pin related branching in the fuel pin treatment, the duct gap flow fraction input, the wire wrap orientation input, the subzone treatments, and the isotope map input file capability. These features are clearly not being tested presently and while the manual clearly indicates that some of them are working (and thus desirable), additional routine testing is wise.

With regard to the uncovered subroutines of the DASSH code, there are several material property features that are not presently tested. Much like some of the other minor code coverage issues mentioned above, these would be fully resolved as part of the software verification work.

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, "DASSH-F: Subchannel Based Thermal Analysis," ANL/NSE-24/82, Argonne National Laboratory (2024).

[11] K. L. Basehore, N. E. Todreas, "SUPERENERGY-2: A Multiassembly, Steady-state Computer Code for LMFBR Core Thermal-Hydraulic Analysis," PNL-3379, COO-2245-57TR, August (1980).

[12] M. A. Smith, C. H. Lee, and R. N. Hill, "GAMSOR: Gamma Source Preparation and DIF3D Flux Solution," ANL/NE-16/50 revision 1, June 28 (2017).

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

[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).

Nuclear Science and Engineering Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439-4842

www.anl.gov