Status of the SAS4A/SASSYS-1 Safety Analysis Code

Nuclear Engineering Division

About Argonne National Laboratory

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

DOCUMENT AVAILABILITY

Online Access: U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via DOE's SciTech Connect (<http://www.osti.gov/scitech/>)

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

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

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

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

Disclaimer

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

Status of the SAS4A/SASSYS-1 Safety Analysis Code

prepared by
T. H. Fanning and A. J. Brunett

Nuclear Engineering Division
Argonne National Laboratory

June 30, 2017

ABSTRACT

SAS4A/SASSYS-1 is a simulation tool used to perform deterministic analysis of anticipated events as well as design basis and beyond design basis accidents for advanced reactors. This report summarizes the current status of SAS4A/SASSYS-1, including the implementation of software quality assurance procedures for configuration management, new licenses, current users, and a gap assessment of the test suite. The existing copyright has been extended for an additional five years, which was essential so that Argonne can continue developing, distributing, and supporting the software for both domestic use and international collaborations. SAS4A/SASSYS-1 Version 5.2 was completed in March 2017 and released to users in May.

The importance of SAS4A/SASSYS-1 maintenance reflects its relevance to a number of U.S. Department of Energy programs as well as domestic and international collaborations. External collaborations have also produced improvements in the SAS4A/SASSYS-1 code, and these are noted in this report. SAS4A/SASSYS-1 has a growing user base that continues to strengthen the promotion of advanced non-LWR reactor concepts. Additional users will help solidify DOE's leadership role in fast reactor safety both domestically and in international collaborations.

TABLE OF CONTENTS

Abstract	i
Table of Contents	iii
List of Figures	iv
List of Tables	iv
1 Introduction.....	1
2 Current Status	3
2.1 Release of Version 5.2.....	3
2.2 New License Agreements.....	3
3 Software Licensing	4
3.1 Current Users.....	4
3.2 Extension of Copyright.....	4
3.3 Revision of Licensing Terms.....	4
4 Software Quality Assurance	6
4.1 Configuration Management	6
4.2 Implementation of Procedures	8
4.3 Automated Testing.....	9
5 Test Suite Gap Assessment.....	9
5.1 Implementation	9
5.2 Test Suite Description	10
5.3 Gap Assessment.....	10
6 Related Work.....	12
7 Summary	12
8 References	13

LIST OF FIGURES

Figure 1: Ticket Workflow in the Trac Environment for SAS4A/SASSYS-1..... 7

LIST OF TABLES

Table 1: SAS License Agreements Established during FY2017.....	3
Table 2: Licensed Users for SAS4A/SASSYS-1 and Mini SAS.....	5
Table 5.1: Test Suite Coverage Summary.....	11
Table 5.2: Fractional Function and Block Coverage per Module.....	11

1 Introduction

SAS4A/SASSYS-1 is a simulation tool used to perform deterministic analysis of anticipated events as well as design basis and beyond design basis accidents for advanced liquid-metal-cooled nuclear reactors. [1] With its origin as SAS1A in the late 1960s, the SAS series of codes has been under continuous use and development for nearly fifty years and represents a critical investment in safety analysis capabilities for the U.S. Department of Energy. [2,3] Version 5.2 of SAS4A/SASSYS-1 was completed in March 2017.

Continued maintenance and improvement of the SAS4A/SASSYS-1 code system is motivated by the relevance of its simulation capability to a number of U.S. Department of Energy programs as well as domestic and international collaborations. Active programs and collaborations that currently use SAS4A/SASSYS-1 include the following:

- FFTF Benchmark: The U.S. DOE is preparing benchmark specifications for the Passive Safety Tests (PST) carried out at the Fast Flux Test Facility between 1984 and 1986. The most prominent tests were the loss of flow without scram (LOFWOS). In collaboration with PNNL, Argonne is assessing the benchmark specifications and preparing SAS4A/SASSYS-1 models for verification and validation purposes.
- EBR-II IAEA Benchmark: The U.S. DOE supported a high-profile Coordinated Research Project with the International Atomic Energy Agency based on the Shutdown Heat Removal Tests conducted at EBR-II. Both protected (SHRT-17) and unprotected (SHRT-45R) loss-of-flow tests were part of the benchmark activity.
- The U.S. DOE is identifying and addressing gaps for the licensing of SFRs. One such gap is the regulatory acceptability of SFR analysis tools. Initial efforts are underway to establish more rigorous software quality assurance for SAS4A/SASSYS-1.
- The U.S. Nuclear Regulatory Commission has licensed SAS4A/SASSYS-1 to evaluate its use for advanced non-LWR license applications, particularly for liquid metal cooled reactors.
- TerraPower, LLC has licensed the SAS4A/SASSYS-1 source code to perform safety analysis studies for their TWR concept. TerraPower also funds code development activities that improve the modeling capabilities of SAS4A/SASSYS-1.
- GE-Hitachi Nuclear Energy has licensed Mini SAS to support pre-application license evaluations of the Advanced Reactor Concepts ARC-100 design and interactions with the Canadian Nuclear Safety Commission.
- Westinghouse Electric Company has licensed Mini SAS to support safety analysis for their lead-cooled fast reactor conceptual design and has initiated the process for acquiring a full license for SAS4A/SASSYS-1.
- Oklo, Inc. is investigating available tools for fast reactor analysis and has acquired a license for Mini SAS to evaluate its applicability to their design.
- The University of California at Berkeley is using Mini SAS to evaluate safety benefits that might be achieved with autonomous reactivity control devices.
- Kansas State University and the University of Wisconsin are preparing experiments that can improve the modeling of thermal stratification in sodium-cooled fast reactors, while their partners, the University of Illinois and Virginia Commonwealth

University, respectively, will develop improved models that could be incorporated into SAS4A/SASSYS-1.

- CEA Bilateral Collaboration: An implementation agreement has been established between the U.S. DOE and the Commissariat à l'énergie atomique et aux énergies alternatives of France for cooperation in low carbon energy technologies. One purpose of the agreement is to evaluate the safety performance of the ASTRID reactor design. DOE participates in this collaboration using the SAS4A/SASSYS-1 safety analysis code.
- JAEA Bilateral Collaboration: The Civil Nuclear Energy Research and Development Working Group (CNWG) was established by the U.S.-Japan Bilateral Commission on Civil Nuclear Cooperation in 2012 to enhance coordination of joint civil nuclear research and development efforts. The Japan Atomic Energy Agency and Argonne are collaborating under the CNWG to improve the oxide fuel severe accident modeling capabilities in SAS4A/SASSYS-1.
- NRA: The Nuclear Regulation Authority of Japan has acquired a license for SAS4A/SASSYS-1. The NRA plans to use SAS4A/SASSYS-1 to support the SFR licensing evaluations in Japan.
- KAERI PG-SFR: The Korean Atomic Energy Research Institute acquired a license for SAS4A/SASSYS-1 to perform safety analysis and model development for the "Prototype Generation-IV Sodium Fast Reactor". KAERI is supporting metallic fuel severe accident model developments that will be incorporated into a future version of SAS4A/SASSYS-1.
- KINS: The Korea Institute of Nuclear Safety is an independent regulatory expert organization that supports the Nuclear Safety and Security Commission (NSSC) in Korea. KINS acquired a license for SAS4A/SASSYS-1 to support the regulatory obligations over the PG-SFR project.
- CIAE Bilateral Collaboration: The DOE-NE Office of International Nuclear Energy Policy and Cooperation has established the U.S.-China Bilateral Civil Nuclear Energy Cooperative Action Plan (BCNECAP) with the China Institute of Atomic Energy. Joint activities under the action plan include model development and safety analyses of the China Experimental Fast Reactor using SAS4A/SASSYS-1.
- The Royal Institute of Technology (KTH: Kungliga Tekniska Högskolan) in Stockholm Sweden has a license for SAS4A/SASSYS-1 to perform natural circulation design performance studies of their ELECTRA lead-cooled fast reactor concept.
- The Center of Technology and Engineering for Nuclear Projects (CITON) in Romania has a license for Mini SAS to perform analysis for the Falcon consortium to support the ALFRED (LFR) demonstrator project.

This report provides an overview of the current status of SAS4A/SASSYS-1, licensing activities to make it more accessible to users, implementation of new software quality assurance procedures, and results of code coverage testing. Information on technical updates to the code will be summarized in a future milestone report. This report satisfies the deliverable for the Level 3 milestone M3AT-17AN1702031, "SAS4A/SASSYS-1 Code Maintenance".

2 Current Status

2.1 Release of Version 5.2

Version 5.2 of SAS4A/SASSYS-1 was completed in March 2017 and announced to users in mid-May following a software quality assurance review and clearance of the updated manual. [1] Fifteen organizations have requested and obtained license keys for the new version. The 5.2 release introduces an extension to the Control System module that provides access to an extensive set of core and core channel state variables such as fuel, cladding, coolant, and structure temperatures; coolant flow rates and pressures; and several other parameters. [4] Development of this modeling update was funded by TerraPower, LLC.

The release of 5.2 also introduces a number of internal code changes and bug fixes. Internally, nearly all obsolete code constructs have been removed to bring the code into compliance with Fortran 2003 standards. This effort included the rewrite of 4141 loops and the elimination of 254 obsolete computed GOTO and arithmetic IF statements. In addition, compiler options for Windows builds have been significantly revised to produce identical results when compared with macOS and Linux builds.

2.2 New License Agreements

During FY2017, six new license agreements were established for Mini SAS and SAS4A/SASSYS-1. These are summarized in Table 1. Based on new as well as existing licensing agreements, a trend has been observed that reveals that the existing licensing structure creates unintended barriers for domestic use. This topic is covered in more detail in Section 3.3, and a new licensing arrangement is being established to eliminate these barriers.

Table 1: SAS License Agreements Established during FY2017.

Organization	Code	Purpose
Japan Atomic Energy Agency	SAS4A/SASSYS-1	CNWG Bilateral
U.S. Nuclear Regulatory Commission	SAS4A/SASSYS-1	Government Use
GE-Hitachi Nuclear Energy	Mini SAS	ARC-100 Support
Westinghouse Electric Company	Mini SAS	LFR Concept
Ulsan National Institute of Science and Technology (UNIST)	SAS4A/SASSYS-1	Academic Use/ PG-SFR
CITON — Center of Technology and Engineering for Nuclear Projects	Mini SAS	EU ALFRED Project

3 Software Licensing

3.1 Current Users

Globally, there are twenty-four licensed users for SAS4A/SASSYS-1 or Mini SAS (see Table 2 on the next page). The limited version, Mini SAS, is popular because it is offered at no cost. However, it does not include the severe accident modeling capabilities found in the full version and its fidelity is limited to only five core channels. This is often adequate for many studies. In fact, Mini SAS is currently being used in three NEUP projects.

3.2 Extension of Copyright

Argonne National Laboratory asserted copyright over SAS4A/SASSYS-1 in May 2012. In 2015, the copyright was extended to include Mini SAS. The Department of Energy originally authorized copyright assertion for a period of five years. In May 2017, Argonne received approval from DOE to extend the copyright for an additional five years, through May 2022. This extension was essential so that Argonne can continue developing, distributing, and supporting SAS4A/SASSYS-1 for both domestic use and international collaborations.

3.3 Revision of Licensing Terms

An evaluation of existing license agreements shows that a trend has developed. In general, licensees can be grouped into one of four categories, although exceptions do exist.

Category	Code (typical)
Domestic Vendors	Mini SAS
Domestic Academic Institutions	Mini SAS
International Bilateral Collaborations	SAS4A/SASSYS-1 (with source)
International Academic Institutions	SAS4A/SASSYS-1 (executable)

The original licensing structure for SAS4A/SASSYS-1 was created to provide a preference for domestic industry. Instead, special bilateral agreements that supersede the license structure have led to full source code access for international organizations such as the Japan Atomic Energy Agency and the Korea Atomic Energy Research Institute. While these agreements are extremely valuable to support U.S. DOE collaborations in fast reactor safety, they put domestic vendors at a disadvantage.

Both established companies and new start-up ventures face significant challenges in the advanced reactor marketplace. After extensive discussions with domestic vendors, a new “limited commercial” license category is being established that will provide *full source-code access* at negligible cost for domestic users. The license can be used during the entire phase of pre-application research and development, including design certification reviews. The free (Mini SAS) and academic (SAS4A/SASSYS-1) licenses will continue to be offered, but do not include source code access. If a design advances to a formal application with the NRC, then the traditional commercial license will apply.

Table 2: Licensed Users for SAS4A/SASSYS-1 and Mini SAS.¹

Organization	Code	Version	Purpose
Pacific Northwest National Laboratory	SAS4A/SASSYS-1	5.2	FFTF Benchmark
China Institute of Atomic Energy	SAS4A/SASSYS-1	5.2	BCNECAP/CEFR
Japan Atomic Energy Agency	SAS4A/SASSYS-1	5.2	CNWG Bilateral
Korea Atomic Energy Research Institute	SAS4A/SASSYS-1	5.2	PG-SFR
Kansas State University	Mini SAS	5.1	NEUP Project
Massachusetts Institute of Technology	Mini SAS	5.1	NEUP Project
North Carolina State University	Mini SAS	5.2	Uncertainty Quantification
Purdue University	Mini SAS	5.2	Academic Use
Texas A&M University	Mini SAS	5.1	Academic Use
University of California, Berkeley	Mini SAS	5.1	NEUP Project
University of Illinois	Mini SAS	5.2	NEUP Project
University of Michigan	Mini SAS	5.1	Academic Use
Virginia Commonwealth University	Mini SAS	5.2	NEUP Project
Royal Institute of Technology	SAS4A/SASSYS-1	5.2	LFR Concepts
Ulsan National Institute of Science and Technology (UNIST)	SAS4A/SASSYS-1 (prev. Mini SAS)	5.2	Academic Use/ PG-SFR
Qvist Atomenergi AB	Mini SAS	5.1	NEUP Project
GE-Hitachi Nuclear Energy	Mini SAS	5.2	ARC-100 Support
Oklo, Inc.	Mini SAS	5.1	GAIN Voucher
TerraPower, LLC	SAS4A/SASSYS-1	5.2	TWR Concept
Westinghouse Electric Company	Mini SAS	5.2	LFR Concept
CITON — Center of Technology and Engineering for Nuclear Projects	Mini SAS	5.2	EU ALFRED Project
Korea Institute of Nuclear Safety	SAS4A/SASSYS-1	5.1	Licensing Review
Nuclear Regulatory Authority (Japan)	SAS4A/SASSYS-1	5.1	Licensing Review
U.S. Nuclear Regulatory Commission	SAS4A/SASSYS-1	5.2	Government Use

¹ Shaded cells indicate new licenses and upgrades in FY2017.

4 Software Quality Assurance

A fundamental goal of software quality assurance (SQA) is to help ensure the development of high-quality software. For very small projects, requirements specifications, code design, implementation, testing, and release may all be carried out by a single person. More formal SQA programs are often based on approved standards from recognized organizations such as the American Society of Mechanical Engineers (ASME) [5,6] and the Institute of Electrical and Electronics Engineers (IEEE). Formal programs focus on documenting the software development activities according to approved procedures to ensure traceability and accountability. The development of an SQA Program for SAS4A/SASSYS-1 will be the subject of a future milestone. Preliminary work is documented in Reference 7.

SAS4A/SASSYS-1 has been under some form of software quality assurance over its entire lifetime. During much of its development (1970-1990) a large team of engineers was involved in code improvements. Full-time staff members were tasked with code management, recordkeeping, code integration, and verification testing. With the elimination of advanced reactor funding in 1994, the development team eventually collapsed to only one or two people. Around the same time, computing hardware underwent significant transformations. Mainframes were replaced with workstations, and eventually personal computers. Earlier code versions and many electronic records did not survive the transition.

Despite this upheaval, SAS4A/SASSYS-1 Version 2.0 and its successors survived along with a handful of validation test cases. However, prior records of SAS development (SAS1A[8], SAS2A[9], SAS3A[10], SAS3D, and SAS4A[11,12]) are unavailable other than through published reports and related citations. Although a wealth of information exists (e.g. experimental data and validation reports), it is not in an easily traceable form.

Although the *development* of the SAS4A/SASSYS-1 SQA Program is the subject of a different work scope, the *implementation* is the direct responsibility of the code maintainers. In the following subsections, initial implementation efforts are described from the viewpoints of source code configuration management, implementing procedures, and automated testing.

4.1 Configuration Management

Source code repositories are essential elements of a software configuration management (SCM) system. Migration of the SAS4A/SASSYS-1 source code into a SCM repository began nine years ago with the first commit of SAS4A/SASSYS-1 Version 2.0, which was originally developed approximately thirty years ago. Over the next few years, the history of SAS4A/SASSYS-1 through versions 2.1, 3.0, 3.1, and 3.1.2 was reconstructed in the Subversion repository. In parallel to these efforts, significant updates were incorporated into the code that culminated with the release of Version 5.0 in late 2012. [13]

With this transition, changes to the source code are now tracked in considerable detail including time stamp, author, rationale, affected files, and lines changed. In addition, only authorized users can modify the repository and only the code manager can create official releases known as *tags*.

Figure 1: Ticket Workflow in the Trac Environment for SAS4A/SASSYS-1.

Along with the Subversion repository, a Trac *environment* is used to maintain records on code releases, code or documentation issues, and build configurations. Until recently, the Trac environment was not extensively used other than to browse code revision histories. With the gradual introduction of SQA practices, Trac is more heavily used for its powerful *ticket* system. For example, tickets are used to maintain QA records for new code developments and issue reporting. Through these records, traceability is substantially improved.

Trac can be configured to route tickets through a set of stages known as a *workflow*. In conjunction with the development of the SQA Plan and related procedures, a Trac workflow was developed and is shown in Figure 1. Four key review stages are introduced that are not part of the default workflow in Trac but that are critical for software development:

- **Requirements Review** ensures that a Software Requirements Specification (SRS) is written that clearly identifies the purpose and requirements of the proposed software development activity.
- **Design Review** ensures that a Software Design Description (SDD) is written that is consistent with the SRS and that contains sufficient detail to define the proposed implementation.
- **Technical Review** ensures that the actual software and documentation implementations are consistent with the SDD and that testing verifies that the requirements defined in the SRS are satisfied.
- **QA Review** ensures traceability by confirming that all relevant SQA Plan requirements and procedures were followed during the development activity.

The Trac workflow does not enforce any particular sequence among the new stages. This is intentional so that a reviewer can reject a ticket and set it to a different stage, if necessary. It is important to maintain flexibility in order to minimize the burden of SQA procedures and encourage their adoption.

4.2 Implementation of Procedures

As described above, the Trac workflow provides review stages for the different types of tickets that may be submitted into the system. The particular path through the workflow is documented in formal procedures that are part of the SQA implementation. Over time, the formal procedures will evolve to meet evolving SQA goals. As of this writing, four procedures have been drafted and are being adopted as part of SAS4A/SASSYS-1 maintenance. These include the following:

- **Software Development and Modification** establishes the process for developing and implementing new functional capabilities or modifying or removing existing functional capabilities within the SAS4A/SASSYS-1 safety analysis software.
- **Problem Reporting and Corrective Action** establishes the process for reporting problems associated with the SAS4A/SASSYS-1 safety analysis software (including documentation) and for taking corrective actions to resolve those errors.
- **Software Testing** establishes the process for evaluating whether the SAS4A/SASSYS-1 software adequately performs all intended functions, properly handles abnormal conditions and credible failures, does not perform adverse unintended functions, and does not degrade the host system.
- **Version Release** establishes the process for the formal release of a new version of the SAS4A/SASSYS-1 safety analysis software to authorized users.

Of the above procedures, only the first, *Software Development and Modification*, requires that all four review stages in the workflow be approved. The *Problem Reporting and Corrective Action* procedure is invoked when an issue is identified that conflicts with expected code behavior defined in existing SRS or SDD documentation. Therefore, only the *Technical Review* and *QA Review* stages need to be approved when the issue is resolved.

Previous SAS4A/SASSYS-1 maintenance efforts often met the *spirit* of the above procedures, but not the rigor and formality required for a more robust SQA program. In the past, big fixes were usually only noted in commit messages and no ticket would be created. Furthermore, an independent review of the correction was not conducted, and testing would be done with limited or no documentation. While the end result may be the same, the lack of formality impairs traceability, which is a critical element of SQA.

4.3 Automated Testing

Current regression testing is extremely strict: successive code updates are expected to produce bit-identical results across all supported platforms. With revitalization of SAS4A/SASSYS-1 maintenance, a larger test suite is emerging to support validation and regression testing. Section 5 summarizes how well the current test suite covers the extensive code base of SAS4A/SASSYS-1.

Although regression testing is still done by hand, a tool to automate much of the repetitive work is being developed so that tests can be run nightly on an automated testing server. Currently, the automated testing code is also being modified to support interactive testing, which is crucial for technical reviews of code changes under SQA procedures.

5 Test Suite Gap Assessment

In an effort to quantify the need for improved coverage of the SAS4A/SASSYS-1 test suite, a gap analysis has been performed that identifies and prioritizes the need for additional test problems. Test suite coverage is an important component of acceptance and regression testing, and an extensive test suite will help ensure continued code integrity. While continuous test suite improvements are expected throughout the lifetime of the SAS4A/SASSYS-1 project, gaps relevant to regularly-exercised models need to be addressed in the near term.

To accomplish this task, Intel's code coverage tool was utilized to assess the file and line coverage of the current test suite. The tool is an integrated component of the Intel Fortran Composer XE compiler that generates web- or text-based reports of dynamic profile statistics during the use of an application. Partial, no, or full source coverage is reported for basic blocks, functions, and files. This information can be viewed by the developer such that line-by-line coverage can be assessed in detail.

5.1 Implementation

Generation of the code coverage report entailed the following steps:

1. Compile the instrumented executable. Standard source is utilized, but the make process is altered by specifying additional compiler options, including enabling source profile generation (via the *-prof-gen* option) and disabling optimization to ensure no source is omitted from the application.
2. Utilize the instrumented executable for a specified workload, in this case, the test suite described in Section 5.2. Execution of the test suite was accomplished via use of a simple shell script that automatically generates unique directories for each test case and executes SAS4A/SASSYS-1 in that directory. Profile information cataloging the

blocks and files utilized during the simulation is generated for each test case during this step and stored in a unique directory.

3. Merge the profile information generated during Step 2 into a single analysis profile using the compiler's *profmerge* tool.
4. Generate the coverage report from the single analysis profile generated in Step 3 using the *codecov* tool. This process creates HTML-based documentation that can be browsed per source file to assess line-by-line coverage in detail.

At this point, the coverage report was reviewed for gaps in the test suite. Two main categories of coverage are identified in the report for each source file: partial and none. For files with partial coverage, the fractional coverage of functions (subroutines) and basic blocks is identified per file. Partially covered source files are marked per line or block to reflect one of the following five conditions: covered, uncovered basic block, uncovered function, partially-covered code, or unknown (typically a comment, include directive, or variable declaration). For files with no coverage, only the total number of functions and blocks in that file are provided.

5.2 Test Suite Description

The current test suite is comprised of 95 unique test cases which range from simple steady state tests to validation of a selection of severe accident models representing phenomena that occurred during the TREAT M-series tests. Of these tests, 83 were developed as part of the verification and validation (V&V) test suite (funded by TerraPower, LLC) which exercises a broad range of basic modeling capabilities for generic reactor characteristics, including:

- Simple steady state,
- Simple transients,
- Alternative treatment of material properties,
- Point kinetics and reactivity feedback,
- Heat rejection system thermal hydraulics, and
- Control system signals.

The remaining test cases include variants of an unprotected loss of flow transient in the ABTR; protected and unprotected transient overpower simulations in EBR-II for a nonspecific core loading; and a series of FPIN validation tests based on the TREAT M5, M6, and M7 tests.

5.3 Gap Assessment

The process outlined in Section 5.1 was applied to the test suite described in Section 5.2 to determine the coverage of the test suite. Table 3 provides a summary of the fractional coverage of all source files, functions, and blocks. The table indicates that while more than half the source files are utilized in at least one test case, approximately half of the functions are being exercised, and less than a quarter of the basic blocks are being utilized, meaning there are significant portions of the code without coverage. Function and block coverage are not equivalent in SAS4A/SASSYS-1 due to the modular structure of the code. Additional

details on test suite coverage per module and a listing of prioritized gaps are described in the remainder of this section.

Results for test suite coverage per module are provided in Table 4 for functions and blocks. Shading in this table indicates ranges of block coverage: red cells indicate block coverage of 0-25%; orange cells indicate block coverage of 26-50%; yellow cells indicate block coverage of 51-75%; and green cells indicate 76-100% block coverage. Modules with the lowest coverage per block include the balance-of-plant (BOP), DEFORM-4, and PLUTO2. Modules with the highest coverage include the severe accident modules DEFORM-5 and PINACLE.

While the results in Table 4 are useful to assist with identifying gaps in the test suite, it is important to consider other factors when prioritizing these gaps. Factors that can affect gap priority include input/model complexity or frequency of module use. For this exercise, gaps are prioritized based on their need for main logic path control or modeling of typical transient behavior, meaning gaps in the MAIN and PRIMAR-4 modules are considered to be of highest priority.

Table 3: Test Suite Coverage Summary

Metric	Covered/Total	Percent Coverage
Files	338/548	62%
Functions	508/1,018	50%
Blocks	32,133/146,277	23%

Table 4: Fractional Function and Block Coverage per Module

Module	Description	Function	Block
BOIL	Two-phase coolant thermal hydraulics	0.81	0.35
BOP	Balance-of-plant systems and components	0.00	0.00
CNTL	Reactor and plant control systems	0.53	0.33
DATA	Data management	0.47	0.39
DEFORM-4	Oxide fuel/cladding mechanics	0.21	0.10
DEFORM-5	Metallic fuel/cladding mechanics	1.00	0.76
FPIN2	Pre-cladding-failure fuel relocation	0.51	0.34
LEVITATE	Post-cladding-failure fuel relocation	0.85	0.44
MAIN	Main program, logic path control, etc.	0.63	0.41
PLUTO2	Post-cladding-failure fuel/coolant interaction	0.38	0.18
PRIMAR-4	Coolant loops thermal hydraulics	0.77	0.43
PINACLE	Molten fuel relocation	0.91	0.68
TSCL	Coolant thermal hydraulics	0.73	0.38

Key gaps in MAIN include limited or no coverage in the subroutines that are responsible for calling alternative logic flow paths, such as CHCHFL, REINIT, SSCLM1, or SSNULL. These gaps can be resolved by developing additional tests that include a restart calculation, null transient, and subassembly-to-subassembly heat transfer calculations. In the case of gaps of PRIMAR-4, a significant portion of these gaps can be closed with additional tests that include alternative compressible volumes or alternative heat exchanger models. Examples include addition of an annular element, check valve, pipe rupture source/sink, compressible liquid volume without gas, stratified volume, vertical inlet/output plenum, detailed steam generator model, a simplified IHX model using a normalized temperature drop, and alternative pump types.

6 Related Work

Two areas of international collaboration will have important impacts on future SAS4A/SASSYS-1 code maintenance activities. Investments in code development by the Korea Atomic Energy Research Institute and the Japan Atomic Energy Agency have led to improvements in fuel failure and accident models for metal and oxide fuel, respectively. Argonne has access to these code developments and will be working to integrate modeling improvements into SAS4A/SASSYS-1.

Expectations were that KAERI would support the integration of the metal fuel modeling improvements, which are of particular interest to the U.S. With the recent policy shift in Korea away from nuclear energy, it is not clear if this support will materialize.

Integration of the oxide fuel modeling improvements is being pursued under the Civil Nuclear Energy Research and Development Working Group (CNWG). At the time of this writing, the very first technical exchanges between Argonne and JAEA are taking place.

7 Summary

Over the last few years, the SAS4A/SASSYS-1 safety analysis code has undergone significant revisions to modernize code structure and data management. [14,15,16,17] With the continued importance of advanced non-LWR reactors, there is also growing interest in applying SAS4A/SASSYS-1 to a number of SFR and LFR concepts. In total, there are twenty-four license agreements in place, with six of them established in the current fiscal year. The safety analysis code continues to be of significant importance to DOE sponsored activities as well as to domestic and international collaborations. SAS4A/SASSYS-1 Version 5.2 was completed in March 2017 and released in May.

Code maintenance activities are gradually shifting into a more formal software quality assurance program that is being developed under the Regulatory Technology Development Plan. As part of this process, source code and documentation are maintained within a software configuration management system and formal code changes are tracked in an electronic ticket system according to preliminary SQA procedures. An evaluation of the current test suite reveals that there are important gaps that need to be addressed. Adherence to an SQA program will help ensure that software development and documentation activities are carried out in a way that ensures traceability and accountability.

8 References

1. T. H. Fanning, A. Brunett, and T. Sumner, eds., *The SAS4A/SASSYS-1 Safety Analysis Code System*, ANL/NE-16/19, Nuclear Engineering Division, Argonne National Laboratory, March 31, 2017.
2. M. Denman et al., "Sodium Fast Reactor Safety and Licensing Research Plan – Volume I," SAND2012-4260, Sandia National Laboratories, 2012.
3. Idaho National Laboratory, "Advanced Reactor Technology – Reactor Technology Development Plan (RTDP)," INL/EXT-14-32837, 2015.
4. T. H. Fanning, A. J. Brunett, and G. Zhang, *SAS4A/SASSYS-1 Code Improvements for FY2016*, ANL-ART-75, Argonne National Laboratory, September 30, 2016.
5. American Society of Mechanical Engineers, *Quality Assurance Requirements for Nuclear Facility Applications*, ASME NQA-1-2008, 2008.
6. American Society of Mechanical Engineers, *Addenda to ASME NQA-1-2008: Quality Assurance Requirements for Nuclear Facility Applications*, NQA-1a-2009, 2009.
7. A. J. Brunett, L. L. Briggs, T. H. Fanning, *Status of SFR Codes and Methods QA Implementation*, ANL-ART-83, Argonne National Laboratory, January 31, 2017.
8. J. C. Carter et al., *SAS1A, A Computer Code for the Analysis of Fast-Reactor Power and Flow Transients*, ANL-7607, Argonne National Laboratory, 1970.
9. F. E. Dunn et al., *The SAS2A LMFBR Accident-Analysis Computer Code*, ANL-8138, Argonne National Laboratory, 1974.
10. M. G. Stevenson et al., "Current Status and Experimental Basis of the SAS LMFBR Accident Analysis Code System," *Proc. of the Fast Reactor Safety Meeting*, pp. 1303–1321, Beverly Hills, CA, 1974.
11. G. Birgersson, et al., *The SAS4A LMFBR Accident Analysis Code System*, ANL/RAS 83-38, Revision 2, Volumes 1 and 2, Argonne National Laboratory, February, 1988.
12. F. E. Dunn, et al., *The SASSYS-1 LMFBR Systems Analysis Code*, ANL/RAS 84-14, Revision 1, Argonne National Laboratory, March, 1987.
13. T. H. Fanning, ed., *The SAS4A/SASSYS-1 Safety Analysis Code System*, ANL/NE-12/4, Nuclear Engineering Division, Argonne National Laboratory, January 31, 2012.
14. T. H. Fanning, F. E. Dunn, D. Grabaskas, T. Sumner, and J. W. Thomas, *SAS4A/SASSYS-1 Modernization*, ANL-ARC-265, Argonne National Laboratory, September 20, 2013.
15. T. H. Fanning, A. Brunett, T. Sumner, and N. Stauff, *SAS4A/SASSYS-1 Modernization and Extension*, ANL-ARC-299, Argonne National Laboratory, September 26, 2014.
16. T.H. Fanning, A. Brunett, T. Sumner, R. Hu, *SAS4A/SASSYS-1 Modernization and Extension*, ANL-ART-30, Argonne, IL, September 30, 2015.
17. T. H. Fanning, A. J. Brunett, and G. Zhang, *SAS4A/SASSYS-1 Code Improvements for FY2016*, ANL-ART-75, Argonne, IL, September 30, 2016.

Nuclear 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