Pacific Northwest
NATIONAL LABORATORY

PNNL-29778

FAST-1.0 Software Release Document

Developed Under NQA-1-2017

March 2020

Ken Geelhood
David Colameco
Michelle Bales
James Corson
Lucas Kyriazidis

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
email: orders@ntis.gov <https://www.ntis.gov/about>
Online ordering: http://www.ntis.gov

PNNL-29778

FAST-1.0 Software Release Document

Developed Under NQA-1-2017

March 2020

Ken Geelhood
David Colameco
Michelle Bales
James Corson
Lucas Kyriazidis

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Project Summary and Document Characteristics

Project Name	FRAPCON and FRAPTRAN Fuel Performance Code Development and Assessment
Project No.	66419 Task 60-16-09
Product Management Office No. / Organization	PM053/ Nuclear Science and Legacy Waste

Approvals

Role	Name	Signature	Date
Project Manager	Tara O'Neil	Tara K O'Neil Digitally signed by Tara K O'Neil Date: 2020.03.24 16:38:59 -07'00'	
Lead Software Developer	Ken Geelhood	Kenneth J (Ken) Geelhood Digitally signed by Kenneth J (Ken) Geelhood Date: 2020.03.24 15:33:07 -07'00'	
Code Custodian	David Colameco	David V Colameco Digitally signed by David V Colameco Date: 2020.03.24 15:01:40 -07'00'	

Revision History

Revision	Date	Comments
0	March 2020	Original Release

Introduction

The purpose of this document is to detail the release of FAST-1.0 and the documents that accompany the development and testing of FAST-1.0. These design requirements were developed by the U.S. Nuclear Regulatory Commission (NRC) in the Statement of Work (SOW) for NRC Agreement Number NRC-HQ-25-14-D-001 and are transcribed to this document as part of the Software Quality Assurance Plan (SQAP) for the FAST Code System, PNNL-28767. FAST 1.0 represents the merger of the FRAPCON and FRAPTRAN codes and as such FAST-1.0 performs steady state and transient fuel performance calculations described below.

FAST-1.0 calculates the steady state response of light-water reactor fuel rods during long-term burnup. The code calculates temperature, pressure, and deformation of a fuel rod as functions of time-dependent fuel rod power and coolant boundary conditions. The phenomena modeled by the code include: 1) heat conduction through fuel and cladding to the coolant; 2) cladding elastic and plastic deformation; 3) fuel-cladding mechanical interaction; 4) fission gas release from the fuel and rod internal pressure; and 5) cladding oxidation. FAST is used to perform independent calculations for regulatory evaluations of fuel performance under normal operation and anticipated operational occurrences (AOOs).

FAST-1.0 also calculates the transient performance of light-water reactor fuel rods during AOOs. FAST-1.0 does not yet have full capabilities to model more rapid and severe transients that occur during hypothetical accidents such as loss of coolant accidents (LOCAs), anticipated transients without SCRAM (ATWS), and reactivity-initiated accidents (RIA). FAST calculates the temperature and deformation history of a fuel rod as a function of time-dependent fuel rod power and coolant boundary conditions. The phenomena modeled by FAST include: 1) heat conduction; 2) heat transfer from cladding to coolant; 3) elastic-plastic fuel and cladding deformation; 4) cladding oxidation; 5) fission gas release; and 6) fuel rod gas pressure.

FAST code assessment, development and maintenance drive a significant portion of the NRC fuel research activities and the tools are used in a substantial number of regulatory products. Given the centrality of the FAST code to the effectiveness of fuel research, it is critical to assess, develop and maintain this tool. The overall objectives are characterized by four main themes:

- Ensuring FAST maintains state-of-the-art features, material properties and fuel performance models.
- Making FAST easier to use and more reliable.
- Developing new capabilities required to perform more sophisticated analysis of in-reactor transient fuel response as well as analysis related to spent fuel storage.
- Supporting an active and engaged peer community through the FAST User Group.

Acronyms and Abbreviations

AOO	Anticipated Operational Occurrences
ASME	American Society of Mechanical Engineers
ATWS	Anticipated Transient Without Scram
CIP	Cabri International Project
CM	Configuration Management
CMMP	Configuration Management & Maintenance Plan
COR	Contracting Officer's Representative
DOE	United States Department of Energy
EBR-II	Experimental Breeder Reactor II
FGR	Fission Gas Release
FUMAC	Fuel Modeling in Accident Conditions
LOCA	Loss of Coolant Accident
LWR	Light Water Reactor
NRC	United States Nuclear Regulatory Commission
NQA-1	Nuclear Quality Assurance – 1
PNNL	Pacific Northwest National Laboratory
POC	Point of Contact
QA	Quality Assurance
QAP	Quality Assurance Plan
RIA	Reactivity Initiated Accident
SCIP	Studsvik Cladding Integrity Program
SFR	Sodium Fast Reactor
SMR	Small Modular Reactor
SNAP	Symbolic Nuclear Analysis Program
SNL	Sandia National Laboratory
SOW	Statement of Work
SQA	Software Quality Assurance
SQAP	Software Quality Assurance Plan

Definitions

This Section provides definitions specific to the software project.

Assessment	A review, evaluation, inspection, test, check, surveillance, or audit to determine and document whether items, processes, systems, or services meet specified requirements and perform effectively. (NQA-1-2017)
Acceptance Testing	The process of exercising or evaluating a system or system component by manual or automated means to ensure that it satisfies the specific requirements and to identify differences between expected and actual results in the operating environment. (NQA-1-2017)
Configuration Item	A collection of hardware or software elements treated as unit for the purpose of configuration control. (NQA-1-2017)
Configuration Management (software)	The process of identifying and defining the configuration items in a system (i.e. software and hardware), controlling the release and change of those items throughout the system's life cycle, and recording and reporting the status of configuration items and change requests. (NQA-1-2017)
Baseline	A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for use and further development, and that can be changed only by using an approved control process. (NQA-1-2017)
Error	A condition deviating from an established baseline, including deviations from the current approved computer program and its baseline requirements. (NQA-1-2017)
Confluence	Confluence is an easy to use web-based tool that is utilized for electronically documenting software in a wiki format. Documents can also be controlled. It offers the ability to document, collaborate, and share.
Graded Approach	The process of ensuring that the level of analysis, documentation, and actions used to comply with a requirement is commensurate with: a) the relative importance to safety, safeguards, and security b) the magnitude of any hazard involved c) the life-cycle stage of a facility or item d) the programmatic mission of a facility e) the particular characteristics of a facility or item f) the relative importance of radiological and nonradiological hazards g) any other relevant factors (NQA-1-2017)
HDI	A web search engine that houses PNNL's Lab-level requirements and procedures and considerations for conducting work. The content is delivered via graphical workflows (step-by-step flowcharts with steps for each activity), through narrative work controls (listing of requirements and considerations for managing specific risks and hazards), or in forms or exhibits (linked documents that include greater detail).
Independent	(Independent Reviews or Independent Testing) Person sufficiently independent with respect to the material/product they are reviewing/testing; they did not perform the work they are reviewing or testing. Staff also possess enough subject matter expertise to adequately review/test/evaluate.
Operating Environment	A collection of software, firmware, and hardware elements that provide for the execution of computer programs. (NQA-1-2017)

Software Design Verification	The process of determining if the product of the software design activity fulfills the software design requirements. (NQA-1-2017)
Software Engineering	(a) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (b) The study of approaches in (a) (NQA-1-2017)
Test Case	A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement. (NQA-1-2017)
Test Plan (Procedure)	A document that describes the approach to be followed for testing a system or component. Typical contents identify items to be tested, tasks to be performed, and responsibilities for the testing activities. (NQA-1-2017)
Verification	Mathematical proof of the correctness of algorithms, by confirming that code subroutines and functions produce the expected numerical output.
Validation	The process of evaluating software to determine whether it satisfies specified requirements, by comparing code predictions to experimental data.
Unit test	Process or code developed to test the numeric accuracy and functionality of new or modified subroutines and functions.
Unit test suite	Set of unit tests created while developing and maintaining FAST.
Verification test suite	Set of input files that exercise all the code options, used to verify that code changes do not negatively impact code performance, and that results are as expected.
Validation test suite	Set of input files used to validate the codes' predictions against experimental measurements, to quantify the accuracy, bias, and uncertainty of code predictions.

Contents

Project Summary and Document Characteristics ii
Introduction iii
Acronyms and Abbreviations iv
Definitions v
Contents vii
1.0 Summary of Release 1
2.0 FAST-1.0 Documentation 3
3.0 FAST-1.0 Executable Builds 5
4.0 FAST-1.0 Acceptance Testing 6
5.0 References 7

Tables

Table 1 – FAST-1.0 Executable Identification 1
Table 2 – FAST-1.0 User Delivery File Identification 1
Table 3 – FAST-1.0 User Delivery File Structure 2
Table 4 – FAST-1.0 Zip files for NRC Delivery 2
Table 5 – FAST-1.0 Build File Listing Identification 2
Table 6 – FAST Project Quality Assurance Documents 3
Table 7 – FAST-1.0 Release Related Documents 4
Table 8 – FAST-1.0 Build Computer Architecture 5
Table 9 – FAST Build Compilers Used 5
Table 10 – FAST-1.0 Build Test Suite Log File 6
Table 11 – FAST-1.0 Tested Operating Systems 6

1.0 Summary of Release

The following software executables listed in Table 1 were built in accordance with the FAST-1.0 Software Design Requirements, PNNL-28868 under NQA-1-2017 through the Software Quality Assurance Plan for the FAST Code System, PNNL-28767.

Table 1 – FAST-1.0 Executable Identification

Executable	OS	Hash
FAST.exe	Windows 10	MD5 5b7ac46985d22069133326e3fe994126 SHA-1 f412f6490721c1259b0d5dd76969c9ac5db34feb
FAST	Linux	MD5 f450ffac9f581d39303d52f4318ffab0 SHA1 e46e1cac223c1f8d10f0a1a60cd8f257f0b8afd2

The executables in Table 1 and their user installation and verification files are being published to the FAST website in the following zip files listed in Table 2. The zip and tgz files contain identical contents.

Table 2 – FAST-1.0 User Delivery File Identification

File	OS	Hash
FAST.zip	Windows/Linux	MD5 bf31043e0551fa9b7b5d637871dd6519 SHA-1 ac94f12352b78ce4bb62e586eb98fa92b3ab299d
FAST.tgz	Windows/Linux	MD5 8d2fd36281c318e9dae31ad14d1c9d7b SHA1 4b05b1d845b926051cd7bb79f5b3514f6bf8bdfa
md5sum_listing	Windows/Linux	MD5 70a9076c591690e4777190c2e0c70252 SHA1 7308ba768debab5ef98a381f99f6d499696ebefe
sha1sum_listing	Windows/Linux	MD5 05777e22a053cd8b4ca9f150592b027e SHA1 1d0fdd2a827b3ad803595e12a1c40cc873a89af1

The Linux command `tar -xzvf FAST.tgz` will decompress the folder. Instructions for users to download the code and perform self-service licensing are contained in the document FAST-1.0 User Installation and Verification Guide, PNNL-29775.

The contents of FAST.zip or FAST.tgz are designed to allow the user to copy a single set of files to a target machine and perform all testing outlined in PNNL-29775. Instructions are also included to inform the user to update the Python script verification_diff.py to point to the location of the executable if the user installs the executable in an alternate location.

Currently, the Sandia National Laboratory CALU licensing file must be placed in the directory where the FAST inputs will be run. The implementation of CALU in FAST will be reviewed and if possible updated to allow the user to install the license file in the same folder as the executable for user convenience.

The FAST.zip and FAST.tgz files have the same contents described in Table 3 below:

Table 3 – FAST-1.0 User Delivery File Structure

Folder	Description
Convenience_Scripts	Python 3 conversion script
Installation_Verification	Installation Verification Tests
Linux_Executable	Linux Executable
Linux_FAST_Licensing	Linux Licensing software
Windows_Executable	Windows Executable
Windows_FAST_Licensing	Windows Licensing software

The identifiers listed in Table 4 are for the zip files of the GitHub repository without a build and the Linux and Windows builds. These files will be sent via Pacific Northwest National Laboratory's (PNNL's) File Transfer System. Documents, including this one, associated with FAST-1.0 will be sent in a separate zip transmittal.

Table 4 – FAST-1.0 Zip files for NRC Delivery

File	Hash
FAST-1.0_GitHub_No_Build.zip	MD5 17ce8490d376053f8d55524a172c3f7b SHA-1 bda7852abe08ce7cc66bd71b1f5341bb74154ee3
FAST-1.0_WE30748_Windows.zip	MD5 1abd8c2f3e4dfdaf9e2329a0ec35af27 SHA1 b2259e2a08e6632e1fc7022b96782152c859247c
FAST-1.0_WE36879_Linux.zip	MD5 12bd5d8d30184b97477c0d3a43fe035d SHA1 f3c85aa8ce09f40d98bcd744072ac736022ce88f

The identifiers of the builds and GitHub repository files are included in Table 5 below:

Table 5 – FAST-1.0 Build File Listing Identification

File	Hash
FAST-1.0_Linux_Build_md5sum_listing	MD5 b9bdb877b71aedfa3f9f0c78255e3d5c SHA-1 7d5c02d07cb8571a11ff3f0339d3c3b3b0731fa6
FAST-1.0_Linux_Build_sha1sum_listing	MD5 e1edf23f359f5aade8038bf980056bfe SHA1 b62c0605fe27747d34f3d4e3eac3674850798795
FAST-1.0_Windows_Build_md5sum_listing	MD5 28269ddd3c01a48c4505150f2f7b9d67 SHA1 54db293217896457bded03d7c3f5e1108bdbe5b4
FAST-1.0_Windows_Build_sha1sum_listing	MD5 b64e3316ea81853e8ef72f28f88dc8cd SHA1 3bbcb96e9334e84fdcfe145e8ee9232e05de0745

2.0 FAST-1.0 Documentation

FAST-1.0 was developed under NQA-1-2017 within the Software Quality Assurance Plan (SQAP) for the FAST Code System, PNNL-28766. NQA-1-2017 results in a better-quality product delivered to our sponsors and users' group. This has also resulted in more documentation than previous releases of FRAPCON and FRAPTRAN to address the additional requirements set forth in NQA-1-2017. This section describes the documentation and how it fits together in the FAST-1.0 release.

The following documents set up the Quality Assurance (QA) program at the project level. The project QA program falls under the umbrella of the PNNL Nuclear Quality Assurance Program for the Laboratory which is in-and-of-itself related to the PNNL Laboratory wide quality assurance program. This FAST-1.0 Software Release Document will focus on the project level documents listed in Table 6 below.

Table 6 – FAST Project Quality Assurance Documents

Document	Number	Comments
Quality Assurance Requirements for Nuclear Facility Applications	ASME NQA-1-2017	Sets quality assurance requirements used by this project for software in Section 2.0.
Software Quality Assurance Plan for the FAST Code System	PNNL-28767	Overarching quality assurance plan describing the process and requirements that the FAST-1.0 development process undertakes.
Configuration Management and Maintenance Plan for the FAST Code System	PNNL-28765	Plan for tracking maintenance activities to provide traceability to project baselines.
Software Assessment Testing Plan for the FAST Code System	PNNL-28766	Plan for testing each baseline and setting requirements for acceptability of testing.

The following documents are released with each version of the FAST code. These documents listed in Table 7 below are set as requirements in the Plans listed in Table 6 above.

Table 7 – FAST-1.0 Release Related Documents

Document	Number	Comments
FAST-1.0 Software Design Requirements Document	PNNL-28868	Software requirements transcribed from the NRC statement of work.
FAST-1.0 Software Design Verification Document	PNNL-29772	Document addressing how the Software Design Requirements were met in FAST-1.0. Also includes Code Design information.
FAST-1.0: A Computer Code for Thermal-Mechanical Nuclear Fuel Analysis under Steady-state and Transients	PNNL-29720	Document describing the FAST-1.0 code and the methodologies contained within it for solving fuel performance characteristics.
FAST-1.0 Integral Assessment	PNNL-29727	Assessment testing results and discussion. Includes comparisons to measured test data.
MatLib-1.0: Nuclear Material Properties Library	PNNL-29728	Document describing the material properties that FAST-1.0 relies upon for thermal and mechanical properties.
FAST-1.0 Software Acceptance Testing Report	PNNL-29769	Document outlining the 418 tests performed on Linux and Windows to determine acceptability of the FAST-1.0 code.
FAST-1.0 User Installation and Verification Guide	PNNL-29775	Document instructing users how to obtain FAST-1.0, install and register the code, and perform installation tests.
FAST-1.0 Software Release Document	PNNL-29778	This document details the FAST-1.0 Release.

3.0 FAST-1.0 Executable Builds

FAST-1.0 was built on Red Hat Enterprise Linux 7.7 Workstation and Windows 10 Enterprise with the following computer architecture listed in Table 8:

Table 8 – FAST-1.0 Build Computer Architecture

File	Hash
Windows 10 Enterprise	Dell Precision Tower 7910 with Intel® Xeon® CPU E5-2609 v3 @1.90GHz (2 Processors)
Red Hat Enterprise Linux 7.7	Dell Precision Tower 7920 with Intel® Xeon® Gold 5120 CPU @2.20GHz (2 Processors)

The compilers listed in Table 9, below, successfully built the executable listed in Table 1.

Table 9 – FAST Build Compilers Used

Compiler	Operating System	Comments
Intel® Visual Fortran Compiler 18.0.5.274	Windows	Intel Fortran Compiler
Microsoft® C/C++ Optimizing Compiler Version 19.00.24215.1 for x64	Windows	Microsoft C Compiler
Microsoft® C/C++ Optimizing Compiler Version 19.00.24215.1 for x64	Windows	Microsoft C++ Compiler
CMake 3.15.3	Windows	CMake Build Software Version
Intel® Fortran 2018.5.274	Linux	Intel Fortran Compiler
gcc Version 8.3.0	Linux	GCC C Compiler
g++ Version 8.3.0	Linux	GCC C++ Compiler
CMake 3.15.3	Linux	CMake Build Software Version

4.0 FAST-1.0 Acceptance Testing

The FAST-1.0 Code system underwent 418 acceptance tests on both the Linux and Windows machines they were compiled upon. The test logs in Table 10, below, are included in their respective Linux and Windows zip files listed in Table 4:

Table 10 – FAST-1.0 Build Test Suite Log File

File	OS	Hash
LastTest.log	Linux	MD5 5584038495d8e4af43f23ecd5b228433 SHA-1 c7d7f46a3daa0c946ea0ca726e9178879de5728d
LastTest.log	Windows	MD5 0ef3eabf495920094ab654da5ef45dbc SHA1 a61112b4a8288cdaff5ce7e44a606d6366700d97

All 418 software acceptance tests passed on Linux and Windows. Testing is discussed in the FAST-1.0: Integral Assessments, PNNL-29727 and the FAST-1.0 Software Acceptance Testing Report, PNNL-29769.

The following environments were successfully tested upon with the User Software Installation and Verification instructions for the verification testing:

Table 11 – FAST-1.0 Tested Operating Systems

Operating System	Version	Comments
Windows 10 Enterprise	1803	OS Build 17134.1304
Windows 10 Enterprise	1909	OS Build 18363.418
Windows Server 2016 Standard	1607	OS Build 14393.693
Windows Server 2019 Standard	1809	OS Build 17763.737
Debian Buster	10.0	Linux Kernel 4.19.0-6
Fedora	31.0	Linux Kernel 5.3.7-301
openSUSE Leap	15.1	Linux Kernel 4.12.14-lp152
Red Hat Enterprise Linux	8.1	Linux Kernel 4.18.0-147
Red Hat Enterprise Linux	7.7	Linux Kernel 3.10.0-1062
Ubuntu LTS	18.04.1	Linux Kernel 4.15.0-72

5.0 References

The following documents were utilized to develop and/or are referenced in this document:

- 10 CFR, Title 10 Code of Federal Regulations, United State Government, 2018.
- ASME NQA-1-2017, Quality Assurance Requirements for Nuclear Facility Applications, January 18, 2018.
- DOE G 414.1-4, Safety Software Guide for use with 10 CFR 830 Subpart A, Quality Assurance Requirements, and DOE O 414.1C, Quality Assurance, November 2010.
- NRC NUREG/BR-0167 Software Quality Assurance Program and Guidelines, February 1993.
- PNNL-28765, Configuration Management and Maintenance Plan for the FAST Code System, Revision 0, June 2019.
- PNNL-28766, Software Assessment Testing Plan for the FAST Code System, Revision 0, June 2019.
- PNNL-28767, Software Quality Assurance Plan for the FAST Code System, Revision 0, June 2019.
- PNNL-28868, FAST-1.0 Software Design Requirements, Revision 0, July 2019.
- PNNL-29720, FAST-1.0 A Computer Code for Thermal-Mechanical Nuclear Fuel Analysis under Steady State and Transients, Revision 0, March 2020.
- PNNL-29727, FAST-1.0: Integral Assessment, Revision 0, March 2020.
- PNNL-29728, MatLib-1.0: Nuclear Material Properties Library, Revision 0, March 2020.
- PNNL-29769, FAST-1.0 Software Acceptance Testing Report, Revision 0, March 2020.
- PNNL-29772, FAST-1.0 Software Design Verification Document, Revision 0, March 2020.
- PNNL-29775, FAST-1.0 User Installation and Verification Document, Revision 0, March 2020.
- Project Plan for the Enterprise-Wide Agreement (EPR 66421 and 66419), PNNL, February 2018.
- Statement of Work (SOW) FRAPCON and FRAPTRAN Fuel Performance Code Development and Assessment, NRC Task Order NRC-HQ-60-16-T-0009.

PNNL-29778

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354
1-888-375-PNNL (7665)

www.pnnl.gov