XLPR GROUP REPORT

COMPUTATIONAL GROUP

PROBABILISTIC FRACTURE MECHANICS CODE

DISCLAIMER

THIS PUBLICATION WAS PREPARED AS AN ACCOUNT OF WORK JOINTLY SPONSORED BY THE ELECTRIC POWER RESEARCH INSTITUTE (EPRI) AND AN AGENCY OF THE U.S. GOVERNMENT. NEITHER EPRI NOR THE U.S. GOVERNMENT NOR ANY AGENCY THEREOF, NOR ANY EMPLOYEE OF ANY OF THE FOREGOING, MAKES ANY WARRANTY, EXPRESSED OR IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR ANY THIRD PARTY'S USE, OR THE RESULTS OF SUCH USE, OF ANY INFORMATION, APPARATUS, PRODUCT, OR PROCESS DISCLOSED IN THIS PUBLICATION, OR REPRESENTS THAT ITS USE BY SUCH THIRD PARTY COMPLIES WITH APPLICABLE LAW.

THIS PUBLICATION DOES NOT CONTAIN OR IMPLY LEGALLY BINDING REQUIREMENTS. NOR DOES THIS PUBLICATION ESTABLISH OR MODIFY ANY REGULATORY GUIDANCE OR POSITIONS OF THE U.S. NUCLEAR REGULATORY COMMISSION AND IS NOT BINDING ON THE COMMISSION.

xLPR Group Report Computational Group	
xLPR-GR-FW Version 1.0	
Author Signature	**Date**
Michael Starr Computational Group Lead	01/02/2020
Reviewer Signatures	**Date**
Marjorie Erickson Marjorie Erickson, PEAI xLPR Models Group Lead	12/30/2019
Approver Signatures	**Date**
Matthew Homiack NRC xLPR Code Development Lead	12/31/2019
Craig Harrington EPRI xLPR Code Development Lead	01/06/2020

THE TECHNICAL CONTENTS OF THIS DOCUMENT WERE NOT PREPARED IN ACCORDANCE WITH THE XLPR SOFTWARE QUALITY ASSURANCE PLAN.

Revision History

Version Number	Description of Changes	Issue Date
1.0	Initial Issue	1/6/2020
	The U.S. Nuclear Regulatory Commission Office of Nuclear Regulatory Research’s and the Electric Power Research Institute’s xLPR Project Contacts approved an administrative update in 2021 to support public release of this document without incrementing the version number or issue date. The administrative updates included: (a) title changed from “Computational Framework Development, Testing, and Analysis” to “xLPR Group Report—Computational Group” throughout the document, (b) cover and title pages updated accordingly, (c) disclaimer statement added, and (d) statement that the document was not prepared in accordance with the xLPR Software Quality Assurance Plan added.	

EXECUTIVE SUMMARY

xLPR Version 2.0 (V2.0) is a probabilistic fracture mechanics code designed to calculate the probability of rupture of welds in nuclear power plant piping. The xLPR V2.0 Code incorporates a set of deterministic models that represent the full range of physical phenomena necessary to evaluate both fatigue and primary water stress-corrosion cracking (PWSCC) degradation modes from crack initiation through rupture. These models are each implemented in a modular form and linked together by the xLPR V2.0 Framework, which contains the logic for code execution, exercises the individual modules as required, performs bookkeeping functions, and displays results.

The purpose of this report is to document the technical basis for development, verification, and validation of the Framework for its intended use to determine the probability of failure for nuclear power plant piping components. This includes descriptions of the functional components of the Framework, their development, verification, validation, and additional activities that support the proper functionality of the xLPR V2.0 Code. The Framework is described with varying degrees of detail throughout this report. High-level summaries of the Framework and its functional components are provided along with additional details to provide an understanding of the computational structure and functionality of the xLPR V2.0 Code. Information regarding the relevant boundaries and constraints of the Framework is also provided.

The Framework serves three main purposes:

1. provides an interface between the user and the global structure of the Code for both inputs and outputs, including post-processing of calculated results

2. propagates uncertainty by defining the number and order of realizations and appropriate values sampled from uncertain input parameters and by performing the user-selected strategy for sampling and uncertainty characterization

3. integrates the overall deterministic model by linking all the sub-models together in a functional structure with all inputs necessary throughout a model run, coordinates all data flow including necessary bookkeeping functions, and generates required outputs

At a basic level, the Framework consists of three components:

1. an Inputs Interface in the form of an Inputs Workbook and a preprocessing program in the form of a Microsoft Excel spreadsheet Add-in

2. a probabilistic simulation engine, landing platform, deterministic model, and outputs interface in the form of a GoldSim model file

3. a suite of Dynamic Link Library files that incorporate the generally deterministic physics models necessary to evaluate both fatigue and PWSCC degradation

The Framework Inputs Interface allows the user to define various options used to control the simulation input values to the deterministic model. These inputs determine the components of the deterministic model that will be exercised during the simulation, such as crack orientation, mechanisms for crack initiation and growth, type of mitigation, and performance of inspections. The inputs also control the way that uncertainty is propagated through the deterministic model via various sampling options including sample size and sampling scheme. Numerous problem-

specific, physically descriptive values and material properties used throughout the deterministic model calculations are also defined in the Inputs Interface.

A systematic approach to characterize and propagate uncertainty forms the backbone of the xLPR Code's probabilistic structure. The Framework considers and separates uncertainty into two types: (1) aleatory uncertainty (which represents inherent randomness), and (2) epistemic uncertainty (which represents lack-of-knowledge). In probabilistic analyses aiming to characterize the likelihood of an adverse event (e.g., pipe rupture), aleatory uncertainty represents the risk or frequency of occurrence, whereas epistemic uncertainty represents the confidence about that frequency. Uncertainty is propagated through the deterministic model using sampling-based Monte Carlo methods. Epistemic and aleatory uncertainties are separated using a two-loop sampling structure. Advanced sampling options beyond Simple Random Sampling include Latin Hypercube Sampling, Discrete Probability Distribution sampling, and Importance Sampling.

The deterministic model of the Framework implements the overall algorithm that calculates crack initiation and subsequent crack behavior over time and is linked to the probabilistic structure and the Inputs Interface through a "landing platform" that controls all data flows. Crack initiation and growth due to fatigue and PWSCC are considered in the deterministic model. The growing cracks are compared to a critical crack size that would result in failure due to loss of structural stability. Leak rates due to through-wall cracks (TWCs) are also calculated. Dynamic Link Libraries are used to connect the Framework to the computational modules that calculate crack initiation, growth, stability, detection, and evaluation.

The Framework was developed using configuration management tools in accordance with the quality assurance requirements and protocols established for xLPR V2.0 Code development. Verification testing was split into unit and integration tests that included both static and dynamic tests of the Framework's functions. The verification testing effort demonstrated that the Framework is structured and performs according to its requirements and design specifications. Validation testing, also termed acceptance testing, was performed to demonstrate that the overall modeling approach produces probability of rupture outputs that reasonably reflect what is known and expected to occur. Three types of validation tests were completed to accomplish this goal: (1) service tests, (2) model behavior tests, and (3) benchmark tests. The conclusion from the acceptance testing is that the Framework is suitable for its intended use and ready for applications and maintenance. Like the Framework, all the modules were developed under similar configuration management controls and are separately documented, verified, and validated.

Additionally, the readiness of the xLPR V2.0 Code was further vetted through a scenario analysis effort that was undertaken after verification and validation testing had concluded. The scenario analyses, which were primarily intended to provide guidance on sampling options for a range of specific scenarios of interest, also exercised the various options, components, and sampling schemes available in the xLPR V2.0 Code. As part of this effort, methods for analyzing and interpreting Code outputs were developed and exercised. As a demonstration of Code readiness, the scenario analyses succeeded in applying xLPR in a manner consistent with its intended use, and the results obtained were consistent with expectations.

TABLE OF CONTENTS

LIST OF FIGURES

ACRONYMS AND ABBREVIATIONS

CDF	Cumulative Distribution Function
CGR	Crack Growth Rate
COA	Crack Opening Area
COD	Crack Opening Displacement
CS	Crack Stability
CTM	Crack Transition Module
DLL	Dynamic Link Library
DM	Dissimilar Metal
DPD	Discrete Probability Distribution
EFPY	Effective Full Power Year
EPRI	Electric Power Research Institute
FW	Framework
GDC	General Design Criteria
ID	Identification Number
iHL	Inner Half-Length
ISI	In-Service Inspection
LB	Large Break
LBB	Leak Before Break
LEAPOR	Leak Analysis of Piping – Oak Ridge
LHS	Latin Hypercube Sampling
LOCA	Loss-Of-Coolant Accident
LRD	Leak Rate Detection
MB	Medium Break
MOU	Memorandum of Understanding
MSC	Multiple Surface Crack
MSIP®	Mechanical Stress Improvement Process ®
MTS	Number of time intervals for crack initiation models
NRC	Nuclear Regulatory Commission
oHL	Outer Half-Length
PCF	Plant Capacity Factor
PNR	Probability of Non-Repair

POD	Probability of Detection
POR	Probability of Repair
PRAISE	Piping Reliability Analysis Including Seismic Events
PROMETHEUS	PRObabilistic METHods for Evaluating and Understanding Structures
PWR	Pressurized Water Reactor
PWSCC	Primary Water Stress Corrosion Cracking
QA	Quality Assurance
RO	Ramberg-Osgood
SB	Small Break
SC	Surface Crack
SCC	Stress Corrosion Cracking
SCMP	Software Configuration Management Plan
SDD	Software Design Description
SIF	Stress Intensity Factor
SM	Similar Metal
SNL	Sandia National Laboratories
SRD	Software Requirements Description
SRS	Simple Random Sampling
SSE	Safe Shutdown Earthquake
STP	Software Test Plan
STRR	Software Test Result Report
SVVP	Software Verification and Validation Plan
TIFFANY	Thermal Stress Intensity Factors For ANY Coolant History
TRC	Transitioning Crack
TW	Through Wall
TWC	Through Wall Crack
US	United States
V2.0	Version 2.0
V&V	Verification and Validation
WRS	Weld Residual Stress
xLPR	eXtremely Low Probability of Rupture

1. INTRODUCTION

The purpose of this report is to document the technical basis for the development, verification, and validation of the Extremely Low Probability of Rupture (xLPR) Code Version 2.0 (V2.0) computational framework (Framework) for its intended use to determine the probability of failure for nuclear power plant piping components. This includes an introduction of the functional components of the Framework and a description of the development, validation and verification, and additional activities that support the proper functionality of the xLPR V2.0 Code. This report is organized as follows:

- Section 1 introduces the Framework and includes information regarding the background and regulatory purpose that motivated code development, an introduction to the overall structure and scope of the Framework, a description of the structure and objectives of the Computational Group responsible for development of the Framework, and a description of the requirements and design of the Framework.

- Section 2, 3, and 4 provide a detailed description of each of the functional components of the Framework, including the inputs interface, the GoldSim© model file, and the module Dynamic Link Libraries (DLLs), respectively. These sections also describe the interfaces between all these components.

- Section 5 describes the testing used to verify and validate the Framework.

- Section 6 provides an overview of the scenario analyses, which exercised the various options, components, and sampling schemes available in the xLPR V2.0 Code.

- Section 7 describes code limitations and provides future recommendations.

- Section 8 presents a summary of the report and provides broad conclusions regarding the use of the xLPR V2.0 Code.

1.1 Background and Regulatory Purpose

Part 50 of Title 10 of the *Code of Federal Regulations*, Appendix A, General Design Criterion (GDC) 4 states, in part, that the dynamic effects associated with postulated fluid system pipe ruptures may be excluded from the design basis when analyses reviewed and approved by the Nuclear Regulatory Commission (NRC) demonstrate that the probability of fluid system piping rupture is extremely low under conditions consistent with the design basis. Section 3.6.3 of NUREG-0800, "Standard Review Plan for the Review of Safety Analysis Reports for Nuclear Power Plants: LWR Edition," describes procedures that have been used to-date to assess the consistency of deterministic leak-before-break (LBB) analyses with the requirements of GDC 4. Currently, NUREG-0800 Section 3.6.3 precludes piping systems with active degradation mechanisms, such as primary water stress corrosion cracking (PWSCC), which is currently occurring in systems that have previously been approved for leak-before-break. Even though these piping systems have been shown to be consistent with GDC 4 through qualitative arguments, no tool existed to quantitatively assess consistency.

From the NRC staff's perspective, a modular, probabilistic fracture mechanics code capable of determining the probability of failure for nuclear power plant piping components was needed. The need for such a modular code was strongly driven by the desire to quantitatively assess piping systems previously approved for LBB. The NRC Office of Nuclear Regulatory Research entered into a cooperative program with the Electric Power Research Institute, Inc. (EPRI) to define, design, and develop the xLPR Code to meet these needs. Under this program, NRC and EPRI set out to build a code that was comprehensive with respect to known challenges, vetted with respect to scientific adequacy of models and inputs, flexible enough to permit analysis of a variety of in-service situations, and adaptable to accommodate evolving and improving knowledge. The final Code was intended to be structured in a modular fashion so that, as additional situations arise, additions or modifications could be easily incorporated without extensive code restructuring. Based on the language in GDC 4, this program and Code was named "xLPR."

An initial version of the xLPR Code, V1.0, was developed through a pilot study to determine the feasibility of the planned approach, effectiveness of the management structure, and the appropriate computational framework [1], [2]. The pilot study considered only a limited scope focused on PWSCC in a pressurizer surge nozzle dissimilar metal weld. The pilot study demonstrated that it was feasible to develop a modular-based computer code for the calculation of rupture probabilities for piping systems previously approved for LBB. Furthermore, although NRC and EPRI determined that the organizational structure used to develop xLPR V1.0 was adequate; the two organizations also identified important organizational changes that would improve the efficiency of further code development. Finally, substantial knowledge and experience were gained through the development of two parallel computational codes, one based on a commercially licensed simulation software and the other based an open-source programming language. The conclusions of the pilot study were used to inform development of xLPR V2.0. This next version has a broader scope to more comprehensively evaluate piping rupture probabilities, including but not exclusively for piping systems previously approved for LBB.

1.2 Framework Introduction

xLPR Version 2.0 (V2.0) is a probabilistic fracture mechanics code designed to calculate the probability of rupture of welds in nuclear power plant piping. The xLPR V2.0 Code incorporates a set of deterministic models that represent the full range of physical phenomena necessary to evaluate both fatigue and PWSCC degradation modes from crack initiation through rupture. These models are each implemented in a modular form and linked together by the Framework, which contains the logic for code execution, exercises the individual modules as required, performs bookkeeping functions, and displays results.

The Framework serves three main purposes:

1. provides an interface between the user and the global structure of the Code for both inputs and outputs, including post-processing of calculated results
2. propagates uncertainty by defining the number and order of realizations and appropriate values sampled from uncertain input parameters and by performing the user-selected strategy for sampling and uncertainty characterization
3. integrates the overall deterministic model by linking all the sub-models together in a functional structure with all inputs necessary throughout a model run, coordinates all data flow including necessary bookkeeping functions, and generates required outputs

At a basic level, the Framework consists of three components:

1. an Inputs Interface consisting of an Inputs Workbook and a preprocessing program in the form of a Microsoft Excel spreadsheet Add-in
2. a probabilistic simulation engine, landing platform, deterministic model, and outputs interface in the form of a GoldSim model file
3. a suite of Dynamic Link Library files that incorporate the generally deterministic physics models necessary to evaluate both fatigue and PWSCC degradation

A commercial off-the-shelf probabilistic simulation software GoldSim©, developed by the GoldSim Technology Group [3], was selected to develop the Framework for xLPR V2.0 due to its range of available features, general compatibility with the goals of the project, ease of use and code maintenance, and ability to easily incorporate externally coded modules as DLLs. GoldSim© provides a highly graphical, object-oriented programming environment that displays the program logic and linkage between code elements in a readily understood manner.

The Framework Inputs Interface allows the user to define various options used to control the xLPR simulation input values that will be the inputs to the deterministic model. These inputs determine the components of the deterministic model that will be exercised during the simulation, such as crack orientation, mechanisms for crack initiation and growth, type of mitigation, and performance of inspections. The inputs also control the way that uncertainty is propagated through the deterministic model via various sampling options including sample size and sampling scheme. Numerous problem-specific, physically descriptive values and material properties used throughout the deterministic model calculations are also defined in the Inputs Interface.

A systematic approach to characterize and propagate uncertainty forms the backbone of the xLPR Code's probabilistic structure. The Framework considers and separates uncertainty into two types: (1) aleatory uncertainty (which represents inherent randomness), and (2) epistemic uncertainty (which represents lack-of-knowledge). In probabilistic analyses aiming to characterize the likelihood of an adverse event (e.g., pipe rupture), aleatory uncertainty represents the risk or frequency of occurrence, whereas epistemic uncertainty represents the confidence about that frequency. Uncertainty is propagated through the deterministic model using sampling-based Monte Carlo methods. Epistemic and aleatory uncertainties are separated using a two-loop sampling structure. For each outer loop, a single sample of the epistemic parameters is selected and held constant while within the inner loop, aleatory parameters are sampled the desired number of times. The results of each epistemic realization (e.g., probability of crack, leak, and rupture) represent an average over all the aleatory samples for that epistemic realization. Advanced sampling options beyond simple random sampling (SRS) include Latin Hypercube Sampling (LHS), Discrete Probability Distribution (DPD) sampling, and Importance Sampling.

The deterministic model of the Framework implements the overall algorithm that calculates crack initiation and subsequent crack behavior over time and is linked to the probabilistic structure and the Inputs Interface through a "landing platform" that controls all data flows. Crack initiation and growth due to fatigue and PWSCC are considered in the deterministic model. The growing cracks are compared to a critical crack size that would result in failure due to loss of structural stability. Leak rates due to through-wall cracks (TWCs) are also calculated. DLLs are used to connect the Framework to the computational modules that calculate crack initiation, growth, stability, detection, and evaluation.

Detailed descriptions of each of the components of the Framework are provided in Sections 2, 3, and 4.

1.3 Computational Group Structure and Objectives

Under the NRC-EPRI developmental organization, the Computational Group was responsible for developing the Framework for the xLPR V2.0 Code. The Computational Group was led by Sandia National Laboratories (SNL) under contract to the NRC, and included mathematicians, engineers, and materials scientists from NRC and EPRI and their contractors. To facilitate transfer of knowledge and integration of the Framework, many of the members of the Computational Group were also members of the other code-development groups, such as the Models Group and Inputs Group. The Computational Group members not only integrated and developed the Framework, but they also performed the verification and validation (V&V) testing established in accordance with the project quality assurance (QA) requirements [5].

Development of the Framework followed the configuration management protocols as required by the Software Configuration Management Plan (SCMP) [6] and the quality assurance requirements and protocols defined by the Software Quality Assurance Plan (SQAP) [5]. The SCMP defines deliverables considered as “the configuration items” to be controlled and the configuration management tools and practices used by the developers. The three software configuration management tools used throughout the developmental effort were Subversion, SharePoint, and JIRA. Subversion is a document repository and revision control tool that was used to track changes performed to configuration items [6]. SharePoint was used for the automated workflow approval of configuration item baselines as described in the SCMP [6]. JIRA is an issue tracking system that was used for problem reporting, corrective actions, and the initiation of changes to baselined configuration items [6].

1.4 Software Requirements

The functional requirements for the Framework are given in the Software Requirements Description (SRD) [7]. This section summarizes these requirements.

The specific requirements of the Framework are defined based on the functions and interfaces desired. The Framework’s functions are described by a series of Unified Modeling Language activity diagrams given in Section 2.1 of the SRD. The diagrams start at the top-level of the workflow and progress down into the detailed workflow. The top level, which is outside the time loop though which crack histories are modeled through time (Section 3.3.3), is referred to as Level 1. The next level contains the time loop and is referred to as Level 2. Level 3 captures any sublevels within the time loop. Detailed requirements for the Code and for interfaces within the time loop are also given in Section 2.1 of the SRD.

Section 2.2 of the SRD defines the requirements for the interfaces between the GoldSim model file component of the Framework and other parts of the model. These interfaces are:

- inputs set and graphical user interface
- GoldSim-DLL interfaces
- DLL-module interfaces
- GoldSim-output files interface

Software attributes, stated in Section 2.3 of the SRD, include that the Framework be developed using the GoldSim software, that the input database be developed using Microsoft Excel, and that the DLLs be compiled in a manner such that they can be called by the Framework. Sections 2.4 and 2.5 of the SRD describe the Framework design constraints and performance requirements, respectively. Requirements for the User Manual are given in Section 2.6 of the SRD. These requirements cover all components of the Framework architecture, inputs, outputs, and protocols for running the Framework that are to be included in the User Manual. The acceptance criteria for software performance are listed in tabular form in Section 2.7 of the SRD. These criteria cover the Framework itself and each module.

The xLPR V2.0 Code was designed to meet each of the requirements stated in the SRD. This design is described in detail in the Software Design Description (SDD) [8] and summarized in Section 1.5. A suite of tests was developed to confirm that the Framework's design met each requirement. These tests are described in the Framework Software Test Plans (STPs) [9], [10], [11].

1.5 Software Design

The computational sequences, data structures, and logic used to meet all the requirements in the SRD [7] are described in the SDD [8]. The SDD details the integration of the various components of the Framework (e.g., Inputs Set, DLLs, GoldSim model file, and GoldSim output files), implementation of the sampling strategies for uncertainty characterization, and initial post-processing of results.

Section 1.2 of the SDD describes the scope of the SDD, including the requirements from the SRD for the Framework itself, and regarding Framework interaction with the individual modules. The modules are listed below and information regarding their functions is given in Section 4:

- Axial Crack Opening Displacement (AxCOD) Module
- Circumferential Crack Opening Displacement (CrCOD) Module
- Crack Coalescence (Coalescence) Module
- Crack Transition Module (CTM)
- PWSCC and Fatigue Crack Growth Rate (CGR) Module
- PWSCC Crack Initiation (CI-SCC) Module
- Fatigue Crack Initiation (CIF) Module
- Axial Crack Stability (AxCS) Modules
- Circumferential Surface Crack Stability (SC_fail) Module
- Circumferential Through-Wall Crack Stability (TWC_fail) Module
- In-Service Inspection (ISI) Module
- Leak Analysis of Piping – Oak Ridge (LEAPOR) Module

- Thermal Stress Intensity Factors for ANY Coolant History (TIFFANY) Module
- K Calculator for Circumferential and Axial Part-Through-Wall Cracks (KPW) Module
- K Calculator for Circumferential and Axial Through-Wall Cracks (KTW) Module

Section 2.1 of the SDD describes the components of the Framework, the GoldSim software, sampling options, modeling options, use of GoldSim objects in the Framework, and design constraints (adherence to requirements, software versions, etc.).

The bulk of the technical implementation of the Framework is detailed in Section 3 of the SDD. This includes:

- descriptions of the user options and the sampling structure
- operations performed within the Framework and the sequence of calculations
- treatment of stress requirements, loads, weld residual stresses (WRS), spatial variability, correlation, pre-processing, mitigation, crack initiation, stress intensity factor solutions, and crack growth
- crack coalescence, transition, stability, and opening displacement
- leak rate calculations
- in-service inspections
- the import of inputs into the GoldSim model file
- a description of the results outputted
- error logging methodology and error codes for all modules

Each of the components of the Framework are described in detail in the sections that follow in this report.

2. INPUTS INTERFACE

The Inputs Interface provides the inputs selected by the user for use in calculations performed in the GoldSim model file. The Inputs Interface consists of the Inputs Workbook and the Preprocessor. Each of these components are described in detail in Sections 2.1 and 2.2, respectively. Figure 2-1 shows the components of the Inputs Interface and their relationships to one another.

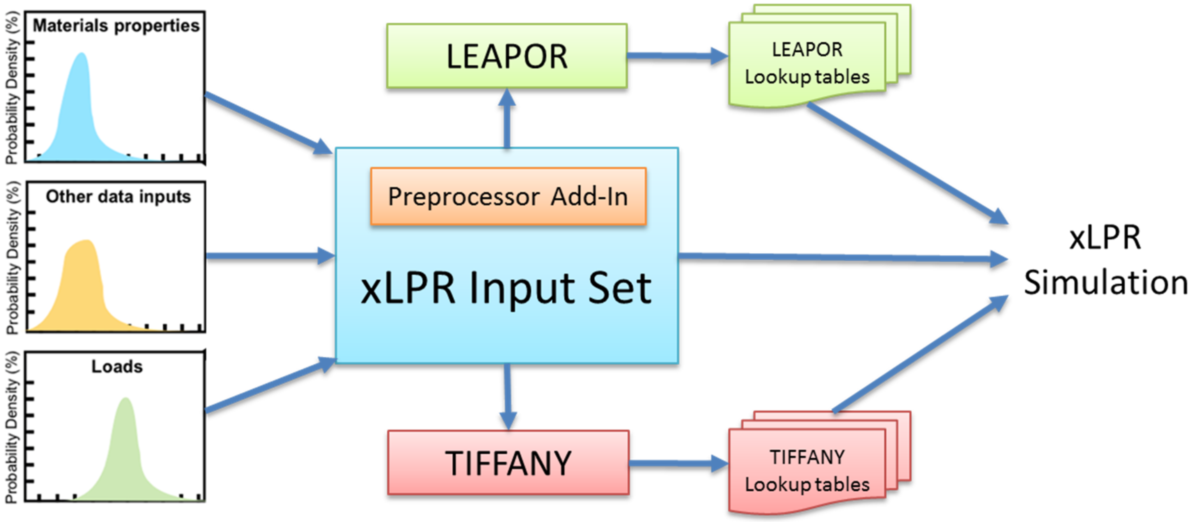


Figure 2-1. xLPR Framework Inputs Interface with the Inputs Workbook shown in blue, the Preprocessor Shown in orange, and the LEAPOR and TIFFANY modules shown in green and red, respectively.

The Sim Editor is an additional element that is available to users to access and edit the Inputs Workbook via a Graphical User Interface, which aids in searching for and populating inputs in each of the worksheets [13]. The Sim Editor was not developed in accordance with quality assurance requirements applicable to the Framework and is thus not described in detail in this report. Pertinent information regarding the interface between the Sim Editor and the Inputs Workbook is provided where appropriate.

2.1 Inputs Workbook

The Inputs Workbook is a Microsoft Excel file that contains nearly all the inputs needed to run a simulation. The only additional inputs needed are several sampling options that must be set in the GoldSim model file from the Global Settings Dashboard, as described in Section 3.1.

The Inputs Workbook contains 13 worksheets. Each of these worksheets are designated by the type of inputs they contain. The worksheets show the input names, identification numbers called "Global IDs," and descriptions, and are organized to assist the user with entry of the various options, values, and distributions as needed. The Inputs Workbook also contains built-in tools and indicators to facilitate data entry. For example, color-coding indicates important information about cell content, and the first column in each worksheet generally indicates the acceptable ranges for each input.

Most of the inputs defined in the Inputs Workbook and read by the GoldSim model file are assigned a four-digit Global ID number. The Global ID is used to track each input in the GoldSim model file and helps to distinguish inputs that may have the same description (e.g., “Yield Strength”) but may pertain to multiple materials, modules, etc. Global IDs are not defined for inputs that are treated as vectors, such as the WRS profiles and transient uncertainty multipliers. In the remainder of this section, Global IDs are referenced to aid in the identification of particular inputs or groups of inputs.

The following subsections describe the structure of the Inputs Workbook. Each of the thirteen worksheets are described in separate subsections or are grouped together where appropriate. For a complete description of the Inputs Workbook and for instructions on how to use the Inputs Workbook and Microsoft Excel Preprocessor, the reader is referred to the User Manual [14].

2.1.1 User Options Worksheet

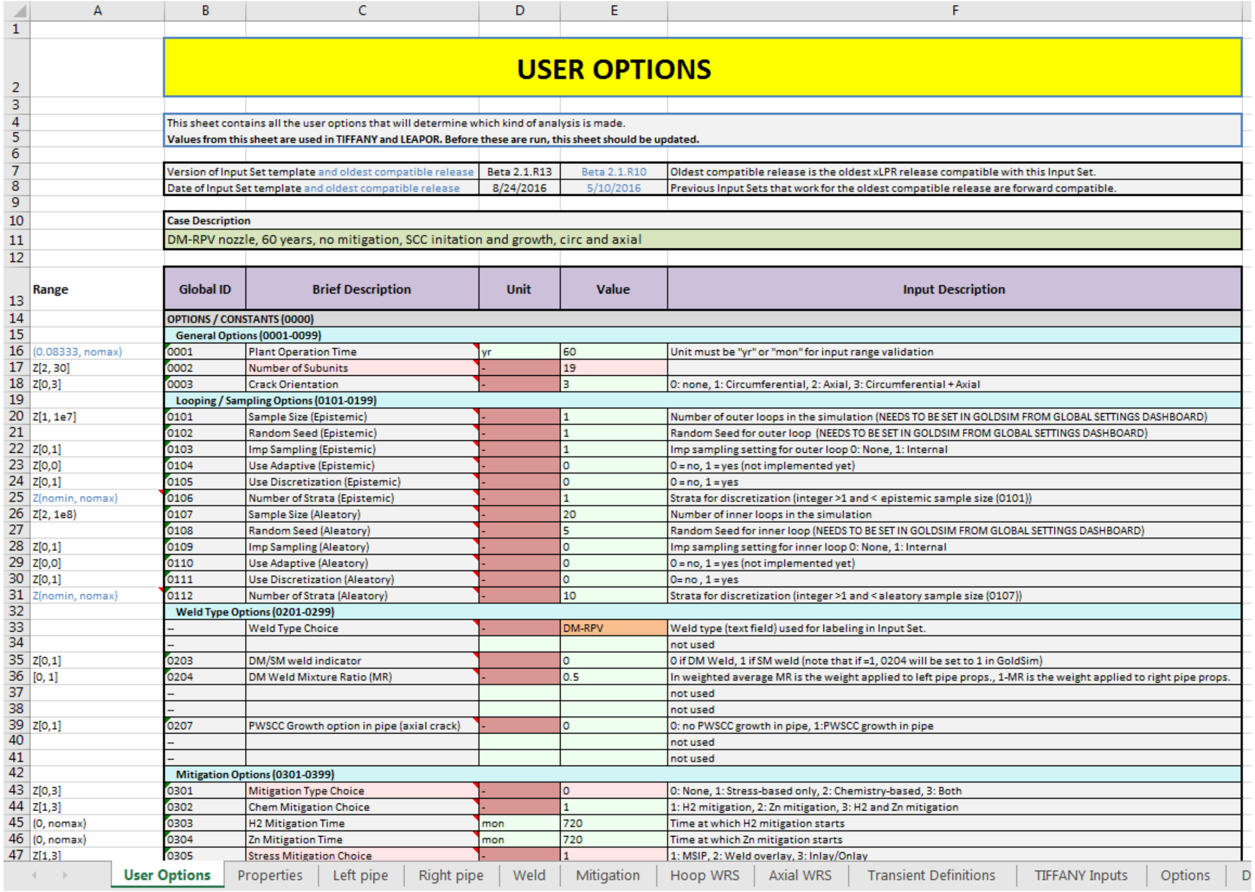
The User Options worksheet contains all the options and attributes that define the scope of the simulation. It includes the 0000 to 0900-series Global IDs.

Unlike other inputs in the Inputs Workbook, the inputs entered on the User Options worksheet are not subject to uncertainty. Values for these inputs are provided in Column E and a brief description is provided in Column F. They include flags for selecting various options and constant parameter values. Several inputs on this worksheet require units, which are set in Column D. Color shading of cells provides additional information regarding whether the input (a) can be changed by the user, (b) requires specific formatting, (c) is used by the Microsoft Excel Preprocessor modules, (d) is not read by the GoldSim model file, or (e) is out of the range of validity defined in Column A.

The categories of inputs on this worksheet are:

- General Options
- Looping/Sampling Options
- Weld Type Options
- Mitigation Options
- Load/Stress Options
- Crack Modeling Options
- Inspection/Leak Detection Options
- LEAPOR Inputs and Options
- COD/Stability Options
- Loss-of-Coolant Accident (LOCA) Options
- Error Handling Options
- Leak Rate Thresholds
- LHS Options

The following subsections describe the inputs under each of these categories in more detail. The top portion of the User Options worksheet is shown in Figure 2-2.


USER OPTIONS

This sheet contains all the user options that will determine which kind of analysis is made.
Values from this sheet are used in TIFFANY and LEAPOR. Before these are run, this sheet should be updated.

Version of Input Set template and oldest compatible release	Beta 2.1.R13	Beta 2.1.R10	Oldest compatible release is the oldest xLPR release compatible with this Input Set.
Date of Input Set template and oldest compatible release	8/24/2016	5/10/2016	Previous Input Sets that work for the oldest compatible release are forward compatible.

Case Description
DM-RPV nozzle, 60 years, no mitigation, SCC initation and growth, circ and axial

Range	Global ID	Brief Description	Unit	Value	Input Description
	OPTIONS / CONSTANTS (0000)				
	General Options (0001-0099)				
(0.08333, nomax)	0001	Plant Operation Time	yr	60	Unit must be "yr" or "mon" for input range validation
Z[2, 30]	0002	Number of Subunits	-	19	
Z[0,3]	0003	Crack Orientation	-	3	0: none, 1: Circumferential, 2: Axial, 3: Circumferential + Axial
	Looping / Sampling Options (0101-0199)				
Z[1, 1e7]	0101	Sample Size (Epistemic)	-	1	Number of outer loops in the simulation (NEEDS TO BE SET IN GOLDSIM FROM GLOBAL SETTINGS DASHBOARD)
	0102	Random Seed (Epistemic)	-	1	Random Seed for outer loop (NEEDS TO BE SET IN GOLDSIM FROM GLOBAL SETTINGS DASHBOARD)
Z[0,1]	0103	Imp Sampling (Epistemic)	-	1	Imp sampling setting for outer loop 0: None, 1: Internal
Z[0,0]	0104	Use Adaptive (Epistemic)	-	0	0 = no, 1 = yes (not implemented yet)
Z[0,1]	0105	Use Discretization (Epistemic)	-	0	0 = no, 1 = yes
Z(nomin, nomax)	0106	Number of Strata (Epistemic)	-	1	Strata for discretization (integer >1 and < epistemic sample size (0101))
Z[2, 1e8)	0107	Sample Size (Aleatory)	-	20	Number of inner loops in the simulation
	0108	Random Seed (Aleatory)	-	5	Random Seed for inner loop (NEEDS TO BE SET IN GOLDSIM FROM GLOBAL SETTINGS DASHBOARD)
Z[0,1]	0109	Imp Sampling (Aleatory)	-	0	Imp sampling setting for inner loop 0: None, 1: Internal
Z[0,0]	0110	Use Adaptive (Aleatory)	-	0	0 = no, 1 = yes (not implemented yet)
Z[0,1]	0111	Use Discretization (Aleatory)	-	0	0= no , 1 = yes
Z(nomin, nomax)	0112	Number of Strata (Aleatory)	-	10	Strata for discretization (integer >1 and < aleatory sample size (0107))
	Weld Type Options (0201-0299)				
	--	Weld Type Choice	-	DM-RPV	Weld type (text field) used for labeling in Input Set.
	--				not used
Z[0,1]	0203	DM/SM weld indicator		0	0 if DM Weld, 1 if SM weld (note that if =1, 0204 will be set to 1 in GoldSim)
[0, 1]	0204	DM Weld Mixture Ratio (MR)	-	0.5	In weighted average MR is the weight applied to left pipe props., 1-MR is the weight applied to right pipe props.
	--				not used
	--				not used
Z[0,1]	0207	PWSCC Growth option in pipe (axial crack)	-	0	0: no PWSCC growth in pipe, 1:PWSCC growth in pipe
	--				not used
	--				not used
	Mitigation Options (0301-0399)				
Z[0,3]	0301	Mitigation Type Choice	-	0	0: None, 1: Stress-based only, 2: Chemistry-based, 3: Both
Z[1,3]	0302	Chem Mitigation Choice	-	1	1: H2 mitigation, 2: Zn mitigation, 3: H2 and Zn mitigation
(0, nomax)	0303	H2 Mitigation Time	mon	720	Time at which H2 mitigation starts
(0, nomax)	0304	Zn Mitigation Time	mon	720	Time at which Zn mitigation starts
Z[1,3]	0305	Stress Mitigation Choice	-	1	1: MSIP, 2: Weld overlay, 3: Inlay/Onlay

Figure 2-2. User Options worksheet of the Inputs Workbook.

2.1.1.1 General Options

The General Options section of the User Options worksheet covers the 0000-series Global IDs. The plant operation time (Global ID 0001) is the calendar time of the simulation. Certain modules, such as the crack initiation module, use effective full-power years (EFPY), which is defined in the Properties worksheet as described in Section 2.1.2. The ratio of EFPY to plant operation time cannot exceed unity.

The number of subunits (Global ID 0002) is the number of equally-sized subunits or segments around the circumference of the pipe into which the weld is discretized to enable multiple crack initiation sites and for spatially varying parameter considerations. This number must be an integer from 2 to 30. The number of subunits was studied as described in the PWSCC Crack Initiation Calibration report [15] and as a result of that effort is set to a default value of 19. If the number of subunits is changed by the user, the corresponding parameters for PWSCC initiation may need to be recalibrated. Figure 2-3 shows the component circumferential segmentation.

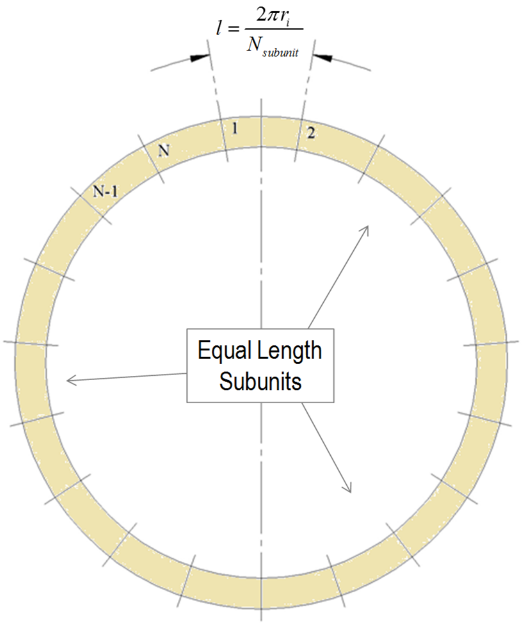


Figure 2-3. Component circumferential segmentation [23].

The crack orientation (Global ID 0003) options are axial, circumferential, both axial and circumferential, and none. Use of the “none” option facilitates checking the validity of an inputs set without activating any of the time loop calculations.

Figure 2-4 shows a layout of the simplified pipe geometry modeled in xLPR.

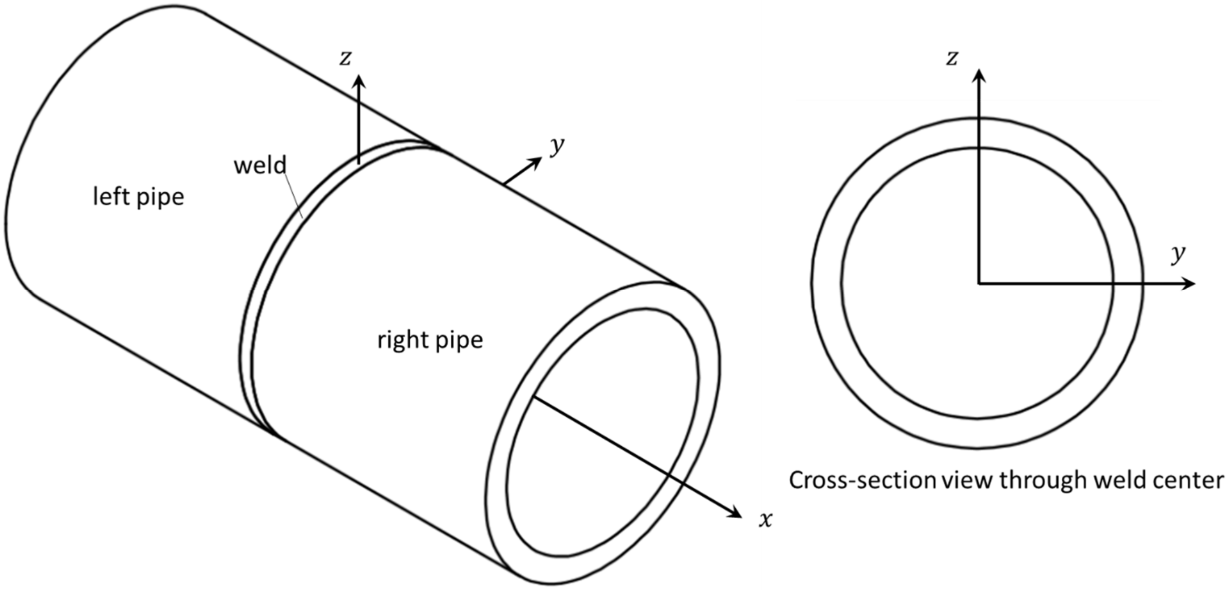


Figure 2-4. Pipe and weld geometry and orientation.

2.1.1.2 Looping / Sampling Options

The Looping/Sampling Options section of the User Options worksheet covers the 0100-series Global IDs. Each input that is set to uncertain in the Inputs Workbook must be categorized as either epistemic (sampled in the outer loop) or aleatory (sampled in the inner loop). The number of realizations and random seeds are defined independently for each loop. The random seed selected for each loop determines the random sequence that will be used to sample the uncertain inputs. Setting the random seed allows the probabilistic simulation to be repeated so that the same results can be obtained given the same set of inputs. Options to turn on Importance Sampling, adaptive sampling, and discretization are also defined in this section of the worksheet. Three of the inputs controlling sampling must be entered directly into the GoldSim model file simulation settings via the Global Settings Dashboard. These three inputs are: (1) epistemic sample size, (2) epistemic random seed, and (3) aleatory random seed. Detailed instructions for entering these inputs are provided in Appendix B of the User Manual [14].

2.1.1.3 Weld Type Options

The Weld Type Options section of the User Options worksheet covers the 0200-series Global IDs. The Weld Type Choice, which has no Global ID, is a text field to describe the weld including whether it is a dissimilar metal (DM) or similar metal (SM) weld. This is only a labeling input and is not used by the Framework; however, it is used by the Sim Editor when accessing and using the weld type database.

The DM/SM weld indicator (Global ID 0203) indicates whether the weld is DM or SM. For a DM weld, the mixture ratio (Global ID 0204) is needed to define the properties used in the crack stability (AxCS, SC-Fail, and TWC-Fail) and COD (CrCOD and AxCOD) modules. The mixture ratio (also referred to as a mixture percentage) is used to create mixed properties by applying a weighted average of the Left Pipe and Right Pipe material properties, including the yield strength, ultimate strength, and calculated Ramberg-Osgood parameters. For further information, see Appendix B of the User Manual [14].

The PWSCC growth option (Global ID 0207) is used to define whether PWSCC growth is allowed in the Left Pipe and Right Pipe for axial cracks.

2.1.1.4 Mitigation Options

The Mitigation Options section of the User Options worksheet covers the 0300-series Global IDs. The mitigation options define which mitigations, if any, occur during the simulation and when they occur. The mitigation material choice, which has no Global ID, names the material used for overlay or inlay. This name is not used in GoldSim model file, but it is used to label different materials when using the Sim Editor.

2.1.1.5 Load / Stress Options

The Load/Stress Options section of the User Options worksheet covers the 0400-series Global IDs. The user has the option of entering normal operating stresses or loads for up to three operating periods during a simulation. The options in this section are used to define these periods and to select whether loads or stresses are used.

Up to 20 transients may be defined for the simulation. Flags in the Load/Stress Options section of the worksheet indicate which of the 20 transients are included in the simulation and whether they are Type I, Type I and II, or Type III. The type selections for each transient all share the overall Global ID 0411 and are distinguished from one another with an added decimal (e.g., Global ID 0411.2 is the transient type selection for the second transient). In addition, a displacement-controlled correction factor (Global ID 0410) is defined here for simulations involving fatigue and for stress calculations.

2.1.1.6 Crack Modeling Options

Crack modeling options include the Crack Initiation Options, Crack Growth Options, and Crack Coalescence Options sections of the User Options worksheet. These sections cover the 0500-, 0600-, and 0700-series Global IDs, respectively. The crack initiation type and model, depending on the initiation type selection, are specified in this section. The crack growth type and related logic options are also specified here. The crack coalescence options allow the user to control the behavior of the crack coalescence module.

2.1.1.7 Inspection/Leak Detection Options

The Inspection/Leak Detections section of the User Options worksheet covers Global IDs 0801 to 0821. Inspection and leak detection options include pre-mitigation and post-mitigation options that define:

- how and when inspections are performed
- how sizing uncertainty is handled
- the detectable leak rate
- how to calculate the as-regressed probability of detection (POD)
- how to calculate the probability of non-repair (PNR)

The ligament flags are used by the ISI module for calculating POD and determine the sets of units in the Inspection/Leak Detection Properties section of the Properties worksheet as described in Section 2.1.2.5.

2.1.1.8 LEAPOR Inputs

The LEAPOR Inputs section of the User Options worksheet covers Global IDs 0851 to 0866. This section provides LEAPOR inputs that are used nowhere else in the Framework. They include the crack morphology parameters for PWSCC and fatigue. For some of these inputs, specific units are required by LEAPOR and thus cannot be changed. The LEAPOR inputs also include the temperature and pressure ranges used to create the look-up tables.

2.1.1.9 COD / Stability Options

The COD/Stability Options section of the User Options worksheet covers Global IDs 0901 to 0911. The surface crack method (Global ID 0901) specifies the shape used in surface crack stability calculations performed by the SC-Fail Module. This is currently the only module that allows for multiple surface crack profiles, thus, constant depth is the only recommended crack profile at this time. The remaining options are tolerances for reducing the number of calls to the various crack stability modules to speed-up calculation times. Stability call reduction is described in greater detail in Section 3.3.3.9.6.

2.1.1.10 LOCA Options

The LOCA Options section of the User Options worksheet covers Global IDs 0921 to 0926. The LOCA definitions can be based on either leak rates or crack-opening areas (COAs). The options here define the typical small break LOCA (SBLOCA), medium break LOCA (MBLOCA), and large break LOCA (LBLOCA).

2.1.1.11 Error Handling

The Error Handling section of the User Options worksheet covers Global IDs 0931 and 0932. Global ID 0931 is an important user control option that, if set to 0, will allow the Code to proceed with calculations even if a module sends a fatal error to the Framework. By default, this input is set to 1, meaning that calculations will stop if there is a fatal error. Greater detail regarding error codes can be found in Section 3.4.4. Global ID 0932 allows the user to determine if uncertainty will be applied to leak rate calculations less than or equal to 10 gpm.

2.1.1.12 Leak Rate Thresholds

The Leak Rate Thresholds section of the User Options worksheet covers Global IDs 0941 and 0942. These inputs are used to control the lower and upper bounds for the leak rate jump indicator, respectively. This indicator records instances in which the leak rate jumps from below the lower bound to above the upper bound in one timestep.

2.1.1.13 LHS Options

The options in the LHS Options section of the User Options worksheet have no Global IDs. These options indicate whether LHS is specified for the epistemic and aleatory loops and must be entered directly using the GoldSim interface. These options are included in the User Options worksheet to allow the user to record all the simulation settings for the purposes of reproducibility and traceability.

2.1.2 Properties Worksheet

The Properties worksheet generally defines the input variables associated with the piping system and not the material. It includes the 1000-, 3000-, 4000-, 5000-, and 9000-series Global IDs.

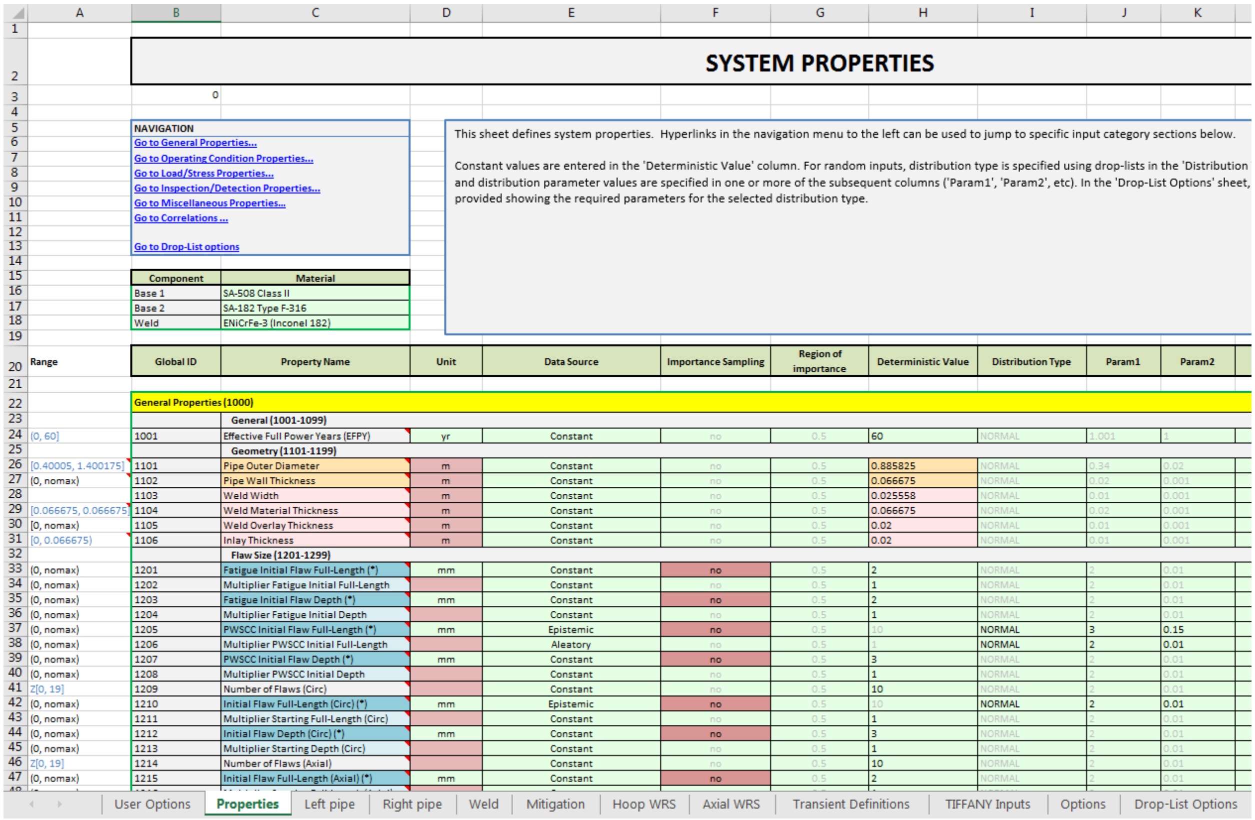
The variables in the Properties worksheet can be defined as either constants or as distributions to be sampled in either the epistemic or aleatory loops, depending on the Data Source selected in Column E. If constant, the deterministic value in Column H is used; if aleatory or epistemic, Columns I and beyond are used to define the distribution type and the parameters of the distribution. Column D is used to set the units.

As in the User Options worksheet, color shading of cells provides additional information to the user. For example, LEAPOR and TIFFANY inputs are shaded in yellow and pink, respectively, and orange for both. Only the deterministic values are shaded here because LEAPOR and TIFFANY only use deterministic values, regardless of the Data Source selection. Additional color-coding shows inputs that cannot be changed by the user, inputs that are spatially varying, inputs that are multipliers to be used for Importance Sampling on spatially varying properties, and general inputs that can be changed by the user.

The categories of inputs on this worksheet are:

- Component Materials (names only)
- General Properties
- Operating Conditions/Environmental Properties
- Load/Stress Properties
- Inspection/Leak Detection Properties
- Miscellaneous Properties
- Correlations
- Stress Calculation Tool

The following subsections describe the inputs under each of these categories in more detail. Figure 2-5 shows an example of the top part of the Properties worksheet.



SYSTEM PROPERTIES

0

NAVIGATION
Go to General Properties...
Go to Operating Condition Properties...
Go to Load/Stress Properties...
Go to Inspection/Detection Properties...
Go to Miscellaneous Properties...
Go to Correlations ...

Go to Drop-List options

This sheet defines system properties. Hyperlinks in the navigation menu to the left can be used to jump to specific input category sections below.

Constant values are entered in the 'Deterministic Value' column. For random inputs, distribution type is specified using drop-lists in the 'Distribution
and distribution parameter values are specified in one or more of the subsequent columns ('Param1', 'Param2', etc). In the 'Drop-List Options' sheet,
provided showing the required parameters for the selected distribution type.

Component	Material
Base 1	SA-508 Class II
Base 2	SA-182 Type F-316
Weld	ENiCrFe-3 (Inconel 182)

Range	Global ID	Property Name	Unit	Data Source	Importance Sampling	Region of importance	Deterministic Value	Distribution Type	Param1	Param2
	General Properties (1000)									
		General (1001-1099)								
(0, 60]	1001	Effective Full Power Years (EFPY)	yr	Constant	no	0.5	60	NORMAL	1.001	1
		Geometry (1101-1199)								
[0.40005, 1.400175]	1101	Pipe Outer Diameter	m	Constant	no	0.5	0.885825	NORMAL	0.34	0.02
(0, nomax)	1102	Pipe Wall Thickness	m	Constant	no	0.5	0.066675	NORMAL	0.02	0.001
	1103	Weld Width	m	Constant	no	0.5	0.025558	NORMAL	0.01	0.001
[0.066675, 0.066675]	1104	Weld Material Thickness	m	Constant	no	0.5	0.066675	NORMAL	0.02	0.001
[0, nomax)	1105	Weld Overlay Thickness	m	Constant	no	0.5	0.02	NORMAL	0.01	0.001
[0, 0.066675)	1106	Inlay Thickness	m	Constant	no	0.5	0.02	NORMAL	0.01	0.001
		Flaw Size (1201-1299)								
(0, nomax)	1201	Fatigue Initial Flaw Full-Length (*)	mm	Constant	no	0.5	2	NORMAL	2	0.01
(0, nomax)	1202	Multiplier Fatigue Initial Full-Length		Constant	no	0.5	1	NORMAL	2	0.01
(0, nomax)	1203	Fatigue Initial Flaw Depth (*)	mm	Constant	no	0.5	2	NORMAL	2	0.01
(0, nomax)	1204	Multiplier Fatigue Initial Depth		Constant	no	0.5	1	NORMAL	2	0.01
(0, nomax)	1205	PWSCC Initial Flaw Full-Length (*)	mm	Epistemic	no	0.5	10	NORMAL	3	0.15
(0, nomax)	1206	Multiplier PWSCC Initial Full-Length		Aleatory	no	0.5	1	NORMAL	2	0.01
(0, nomax)	1207	PWSCC Initial Flaw Depth (*)	mm	Constant	no	0.5	3	NORMAL	2	0.01
(0, nomax)	1208	Multiplier PWSCC Initial Depth		Constant	no	0.5	1	NORMAL	2	0.01
Z[0, 19]	1209	Number of Flaws (Circ)		Constant	no	0.5	10	NORMAL	2	0.01
(0, nomax)	1210	Initial Flaw Full-Length (Circ) (*)	mm	Epistemic	no	0.5	10	NORMAL	2	0.01
(0, nomax)	1211	Multiplier Starting Full-Length (Circ)		Constant	no	0.5	1	NORMAL	2	0.01
(0, nomax)	1212	Initial Flaw Depth (Circ) (*)	mm	Constant	no	0.5	3	NORMAL	2	0.01
(0, nomax)	1213	Multiplier Starting Depth (Circ)		Constant	no	0.5	1	NORMAL	2	0.01
Z[0, 19]	1214	Number of Flaws (Axial)		Constant	no	0.5	10	NORMAL	2	0.01
(0, nomax)	1215	Initial Flaw Full-Length (Axial) (*)	mm	Constant	no	0.5	2	NORMAL	2	0.01

Figure 2-5. Properties worksheet of the Inputs Workbook.

2.1.2.1 Component Materials

The names of the materials for the Left Pipe, Right Pipe, and Weld components of the system are specified in the upper left corner of the Properties worksheet. These names are not used in the simulation; however, they are used to describe the overall system. Names are needed for these materials when the Sim Editor is used to modify the Inputs Workbook or work with the weld type and materials databases.

2.1.2.2 General Properties

The General Properties section of the Properties worksheet covers the 1000-series Global IDs. The first input parameter in this section is EFPY. This is the equivalent number of years of full power operation during the plant operation period, which is defined in Section 2.1.1.1.

The remainder of the inputs in the General Properties section are parameters that describe the geometry of the components (i.e., weld, pipes, overlay, and inlay) and the size of either pre-existing cracks or cracks initiated by PWSCC or fatigue. Each dimension of the initiated crack is spatially variable around the circumference of the weld, and each has a corresponding multiplier for use in Importance Sampling.

2.1.2.3 Operating Conditions / Environmental Properties

The Operating Conditions/Environmental Properties section of the Properties worksheet covers the 3000-series Global IDs. This section contains the inputs for fluid flow rate through the pipe, operating pressure and temperature, and the concentrations of hydrogen, zinc, and dissolved oxygen. Normal operating temperature, pressure, and dissolved oxygen concentration may be defined for up to three operating periods. Initial and mitigation concentrations are defined for hydrogen and zinc. The three time periods are defined in the User Options worksheet as discussed in Section 2.1.1.4.

2.1.2.4 Load / Stress Properties

The Load/Stress Properties section of the Properties worksheet covers the 4000-series Global IDs. The inputs in this section include:

- earthquake properties, such as stresses and probability of occurrence
- loads and stresses for each of the three possible normal operating periods
- additional axial and circumferential surface stresses at the inside diameter of the weld
- WRS distribution and sampling options

Whether load inputs or stress inputs are used depends on the options selected in the User Options worksheet, as discussed in Section 2.1.1.5.

The WRS distribution and sampling options are covered in Global IDs 4350, 4351, 4352, and 4353. These inputs allow the user to specify the WRS data source as either constant, epistemic, or aleatory and allow importance sampling to be applied to the WRS inputs on the Hoop WRS and Axial WRS worksheets. A normal distribution is required for these inputs. Importance Sampling affects only the WRS value at the first point on the inside diameter. This allows for Importance Sampling to be applied to the WRS profile at its most critical point without reducing the impact of Importance Sampling by applying it to too many variables (i.e., all 26 WRS points). This also allows for the point-to-point correlation algorithm used to construct the sampled WRS profiles to be used with Importance Sampling.

2.1.2.5 Inspection / Leak Detection Properties

The Inspection/Leak Detection Properties section of the Properties worksheet covers the 5000-series Global IDs. These inputs are the set of variables used in the ISI module that may be represented by uncertainty distributions. They include factors applied to the nominal POD, threshold values used in approximations, and sizing and slope terms in the inspection routines.

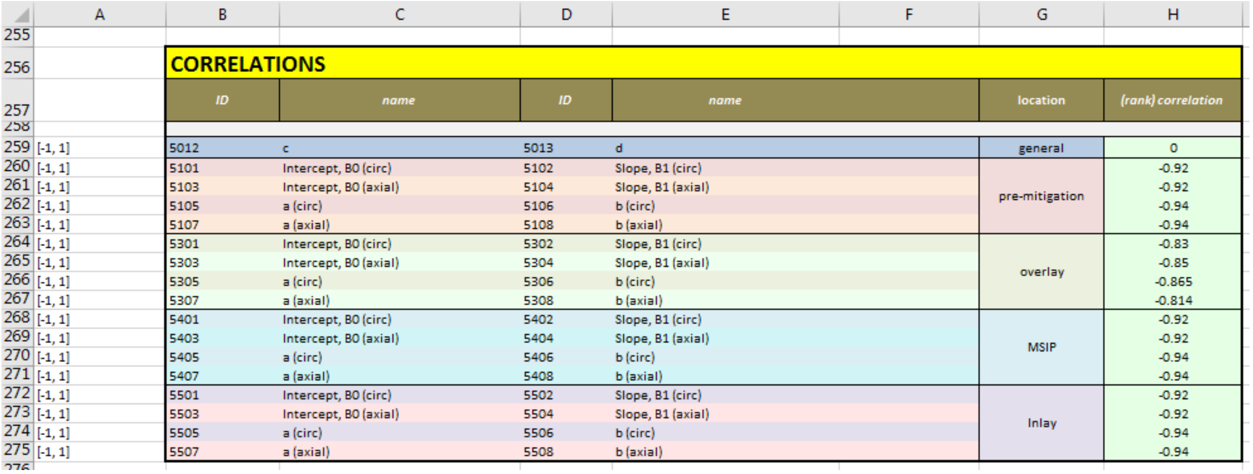
Separate inputs are needed for pre-mitigation, post-overlay, post-Mechanical Stress Improvement Process (MSIP®), and post-inlay. Many of these inputs and their units depend on the ligament options selected in the User Options worksheet, as discussed in Section 2.1.1.7.

2.1.2.6 Miscellaneous Properties

The Miscellaneous Properties section of the Properties worksheet covers the 9000-series Global IDs. This section contains one input for fatigue growth, two inputs for coalescence distance tolerances, and seventeen undefined inputs reserved for possible future use. The units for the surface crack distance rule modifier (Global ID 9002) are determined by a coalescence option (Surface Distance Rule, Global ID 0702) selected in the User Options worksheet, as discussed in Section 2.1.1.6.

2.1.2.7 Correlations

Rank correlations for some inputs on the Properties worksheet may be specified in the correlations table at the bottom. This table is shown in Figure 2-6. Input sampling in the Framework is designed to enforce the rank correlations entered in this table. More information regarding the correlations implemented in the Framework can be found in Section 3.1.3.6.



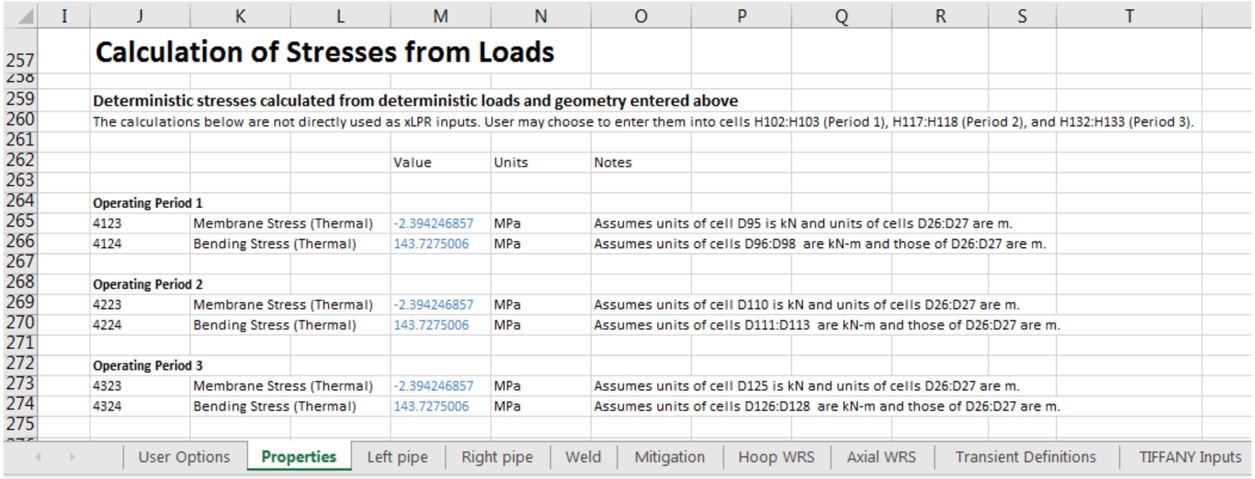
CORRELATIONS

	ID	name	ID	name	location	(rank) correlation
[-1, 1]	5012	c	5013	d	general	0
[-1, 1]	5101	Intercept, B0 (circ)	5102	Slope, B1 (circ)	pre-mitigation	-0.92
[-1, 1]	5103	Intercept, B0 (axial)	5104	Slope, B1 (axial)		-0.92
[-1, 1]	5105	a (circ)	5106	b (circ)		-0.94
[-1, 1]	5107	a (axial)	5108	b (axial)		-0.94
[-1, 1]	5301	Intercept, B0 (circ)	5302	Slope, B1 (circ)	overlay	-0.83
[-1, 1]	5303	Intercept, B0 (axial)	5304	Slope, B1 (axial)		-0.85
[-1, 1]	5305	a (circ)	5306	b (circ)		-0.865
[-1, 1]	5307	a (axial)	5308	b (axial)		-0.814
[-1, 1]	5401	Intercept, B0 (circ)	5402	Slope, B1 (circ)	MSIP	-0.92
[-1, 1]	5403	Intercept, B0 (axial)	5404	Slope, B1 (axial)		-0.92
[-1, 1]	5405	a (circ)	5406	b (circ)		-0.94
[-1, 1]	5407	a (axial)	5408	b (axial)		-0.94
[-1, 1]	5501	Intercept, B0 (circ)	5502	Slope, B1 (circ)	Inlay	-0.92
[-1, 1]	5503	Intercept, B0 (axial)	5504	Slope, B1 (axial)		-0.92
[-1, 1]	5505	a (circ)	5506	b (circ)		-0.94
[-1, 1]	5507	a (axial)	5508	b (axial)		-0.94

Figure 2-6. Correlations table in Properties worksheet of the Inputs Workbook.

2.1.2.8 Stress Calculation Tool

The Stress Calculation Tool in the lower right corner of the Properties worksheet can be used to calculate thermal membrane and bending stresses from the deterministic values entered for load inputs, pipe diameter, and wall thickness. This tool, shown in Figure 2-7, assumes that the length inputs are in meters, forces are in kN and moments are in kN-m. This tool is especially useful if TIFFANY is used and only loads are known because TIFFANY requires stress inputs in cells H102 and H103. The stresses calculated here may be manually copied to column H for Global IDs 4123, 4124, 4223, 4224, 4323, and 4324.



Calculation of Stresses from Loads

Deterministic stresses calculated from deterministic loads and geometry entered above

The calculations below are not directly used as xLPR inputs. User may choose to enter them into cells H102:H103 (Period 1), H117:H118 (Period 2), and H132:H133 (Period 3).

		Value	Units	Notes
Operating Period 1				
4123	Membrane Stress (Thermal)	-2.394246857	MPa	Assumes units of cell D95 is kN and units of cells D26:D27 are m.
4124	Bending Stress (Thermal)	143.7275006	MPa	Assumes units of cells D96:D98 are kN-m and those of D26:D27 are m.
Operating Period 2				
4223	Membrane Stress (Thermal)	-2.394246857	MPa	Assumes units of cell D110 is kN and units of cells D26:D27 are m.
4224	Bending Stress (Thermal)	143.7275006	MPa	Assumes units of cells D111:D113 are kN-m and those of D26:D27 are m.
Operating Period 3				
4323	Membrane Stress (Thermal)	-2.394246857	MPa	Assumes units of cell D125 is kN and units of cells D26:D27 are m.
4324	Bending Stress (Thermal)	143.7275006	MPa	Assumes units of cells D126:D128 are kN-m and those of D26:D27 are m.

Figure 2-7. Stress calculation tool in the Properties worksheet of the Inputs Workbook.

2.1.3 Left Pipe and Right Pipe Worksheets

The next four worksheets in the Inputs Workbook are:

- Left Pipe
- Right Pipe
- Weld
- Mitigation

Each of these worksheets provide the material properties for their respective components and all are identically organized with the same input descriptions and locations. The one exception is that the Global ID numbers are different on each worksheet. Maintaining an identical organization for all four of these worksheets facilitates the interchange of materials between components and the maintenance of a materials database for use by the Sim Editor.

The Left Pipe worksheet includes the 2100- and 2200-series Global IDs. The Right Pipe worksheet includes the 2300- and 2400-series Global IDs. Only a limited set of the input fields in these two worksheets are used and interaction with the Framework is different from the Weld and Mitigation worksheets. Although cracks may grow by fatigue and PWSCC (when Global ID 0207 is set to 1) into the base metals, TIFFANY does not use any of the TIFFANY parameters (Global IDs 2111 to 2120 and Global IDs 2311 to 2320, respectively). In addition, cracks do not initiate in the base metals, so the crack initiation property inputs (Global IDs 2121 to 2155 and Global IDs 2321 to 2355) are also never used. The one exception in the crack initiation properties section is the Sulfur Content (Global IDs 2127 and 2327), which is used for fatigue crack growth calculations. The Threshold SIF Scaling Factor (Global IDs 2180 and 2380) is not used for any material and the Reference Temperature (Global IDs 2197 and 2397) is not used for the Left and Right Pipe materials and is instead read from the Weld worksheet.

The full set of inputs for all four material worksheets is summarized in the Weld worksheet section that follows.

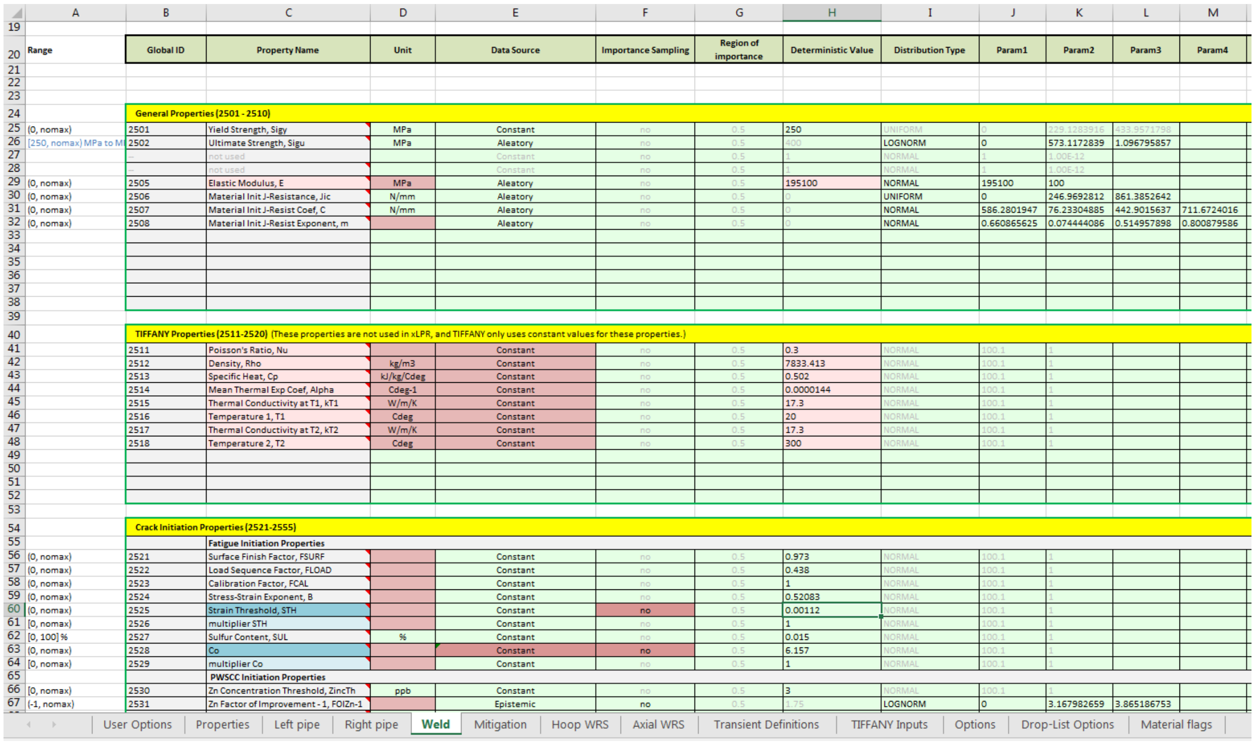
2.1.4 Weld Worksheet

The Weld worksheet includes the 2500- and 2600-series Global IDs. All inputs are active in the Weld worksheet except for the Threshold SIF Scaling Factor (Global ID 2580). As in the Properties worksheet, variables in the Weld worksheet can be defined as either constants or as distributions. Also, color shading of cells provides additional information to the user, such as which inputs are used by TIFFANY, which inputs cannot be changed by the user, which inputs are spatially varying, which inputs are multipliers to be used for Importance Sampling on spatially varying properties, and which are general inputs that can be changed by the user.

The categories of inputs on this worksheet are:

- General Properties
- TIFFANY Properties
- Crack Initiation Properties
- Crack Growth Properties
- Miscellaneous Properties
- Correlations
- Environmental Factor Table

The following subsections describe the inputs under each of these categories in more detail. The top portion of the Weld worksheet is shown in Figure 2-8.



Range	Global ID	Property Name	Unit	Data Source	Importance Sampling	Region of importance	Deterministic Value	Distribution Type	Param1	Param2	Param3	Param4
	General Properties (2501 - 2510)											
(0, nomax)	2501	Yield Strength, Sigy	MPa	Constant	no	0.5	250	UNIFORM	0	229.1283916	433.9571798	
[250, nomax) MPa to MI	2502	Ultimate Strength, Sigu	MPa	Aleatory	no	0.5	400	LOGNORM	0	573.1172839	1.096795857	
	--	not used		Constant	no	0.5	1	NORMAL	1	1.00E-12		
	--	not used		Constant	no	0.5	1	NORMAL	1	1.00E-12		
(0, nomax)	2505	Elastic Modulus, E	MPa	Aleatory	no	0.5	195100	NORMAL	195100	100		
(0, nomax)	2506	Material Init J-Resistance, Jic	N/mm	Aleatory	no	0.5	0	UNIFORM	0	246.9692812	861.3852642	
(0, nomax)	2507	Material Init J-Resist Coef, C	N/mm	Aleatory	no	0.5	0	NORMAL	586.2801947	76.23304885	442.9015637	711.6724016
(0, nomax)	2508	Material Init J-Resist Exponent, m		Aleatory	no	0.5	0	NORMAL	0.660865625	0.074444086	0.514957898	0.800879586
	TIFFANY Properties (2511-2520) (These properties are not used in xLPR, and TIFFANY only uses constant values for these properties.)											
	2511	Poisson's Ratio, Nu		Constant	no	0.5	0.3	NORMAL	100.1	1		
	2512	Density, Rho	kg/m3	Constant	no	0.5	7833.413	NORMAL	100.1	1		
	2513	Specific Heat, Cp	kJ/kg/Cdeg	Constant	no	0.5	0.502	NORMAL	100.1	1		
	2514	Mean Thermal Exp Coef, Alpha	Cdeg-1	Constant	no	0.5	0.0000144	NORMAL	100.1	1		
	2515	Thermal Conductivity at T1, kT1	W/m/K	Constant	no	0.5	17.3	NORMAL	100.1	1		
	2516	Temperature 1, T1	Cdeg	Constant	no	0.5	20	NORMAL	100.1	1		
	2517	Thermal Conductivity at T2, kT2	W/m/K	Constant	no	0.5	17.3	NORMAL	100.1	1		
	2518	Temperature 2, T2	Cdeg	Constant	no	0.5	300	NORMAL	100.1	1		
	Crack Initiation Properties (2521-2555)											
		Fatigue Initiation Properties										
(0, nomax)	2521	Surface Finish Factor, FSURF		Constant	no	0.5	0.973	NORMAL	100.1	1		
(0, nomax)	2522	Load Sequence Factor, FLOAD		Constant	no	0.5	0.438	NORMAL	100.1	1		
(0, nomax)	2523	Calibration Factor, FCAL		Constant	no	0.5	1	NORMAL	100.1	1		
(0, nomax)	2524	Stress-Strain Exponent, B		Constant	no	0.5	0.52083	NORMAL	100.1	1		
(0, nomax)	2525	Strain Threshold, STH		Constant	no	0.5	0.00112	NORMAL	100.1	1		
[0, nomax)	2526	multiplier STH		Constant	no	0.5	1	NORMAL	100.1	1		
[0, 100] %	2527	Sulfur Content, SUL	%	Constant	no	0.5	0.015	NORMAL	100.1	1		
(0, nomax)	2528	Co		Constant	no	0.5	6.157	NORMAL	100.1	1		
[0, nomax)	2529	multiplier Co		Constant	no	0.5	1	NORMAL	100.1	1		
		PWSCC Initiation Properties										
[0, nomax)	2530	Zn Concentration Threshold, ZincTh	ppb	Constant	no	0.5	3	NORMAL	100.1	1		
(-1, nomax)	2531	Zn Factor of Improvement - 1, FOIZn-1		Epistemic	no	0.5	1.75	LOGNORM	0	3.167982659	3.865186753	

Figure 2-8. Weld worksheet of the Inputs Workbook.

2.1.4.1 General Properties

The General Properties section of the Weld worksheet covers Global IDs 2501 to 2508. These material properties include the Yield Strength, Ultimate Strength, Elastic Modulus, and fracture toughness parameters.

2.1.4.2 TIFFANY Properties

The TIFFANY Properties section of the Weld worksheet covers Global IDs 2511 to 2518. These material property inputs are only used in the TIFFANY module. Because TIFFANY is a deterministic preprocessor, it only reads the values in the deterministic value column.

2.1.4.3 Crack Initiation Properties

The Crack Initiation Properties section of the Weld worksheet covers Global IDs 2521 to 2554. Crack initiation modeling is highly material-dependent for both PWSCC and fatigue. Material flag options selected in the Miscellaneous Properties section of the Weld worksheet (discussed in Section 2.1.4.5) and the PWSCC initiation method choice in the User Options worksheet (discussed in Section 2.1.1.6) determine which of the models are used. These inputs cover all material-specific inputs to the crack initiation models.

2.1.4.4 Crack Growth Properties

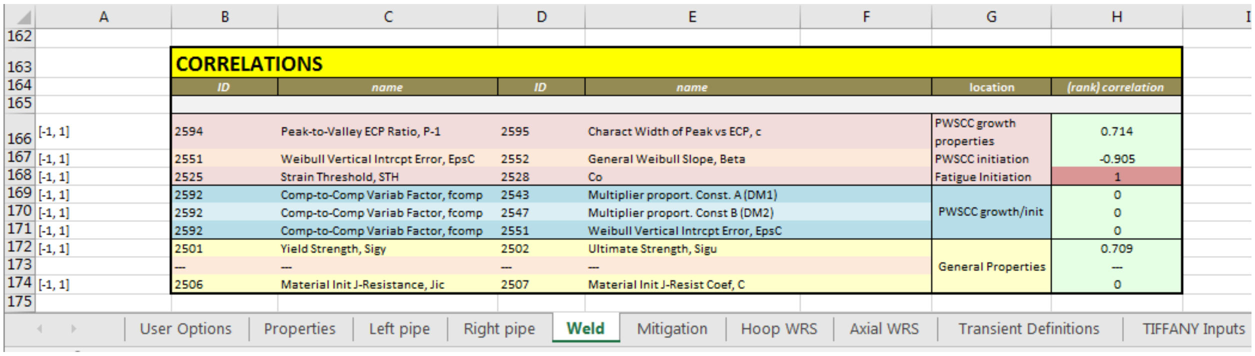
The Crack Growth Properties section of the Weld worksheet covers Global IDs 2560 to 2597. As in the case of crack initiation modeling, crack growth modeling is also highly material-dependent. Material flag options selected in the Miscellaneous Properties section of the Weld worksheet (as discussed in Section 2.1.4.5) and the crack growth options in the User Options worksheet (as discussed in Section 2.1.1.6) determine which crack growth models are used. These inputs cover all material-specific inputs to the various crack growth models.

2.1.4.5 Miscellaneous Properties

The Miscellaneous Properties section of the Weld worksheet covers the 2600-series Global IDs. This section contains deterministic crack initiation and growth material flags, temperature coefficients for adjusting the material strength properties to room temperature, and four undefined inputs reserved for possible future use. Information regarding the material flags is provided in the Material Flags worksheet, which is described in Section 2.1.11.

2.1.4.6 Correlations

Rank correlations for some of the material properties may be specified in the correlations table at the bottom of the Weld worksheet. This table is shown in Figure 2-9. Input sampling in the Framework is designed to enforce the rank correlations entered in this table. More information regarding the correlations implemented in the Framework can be found in Section 3.1.3.6.



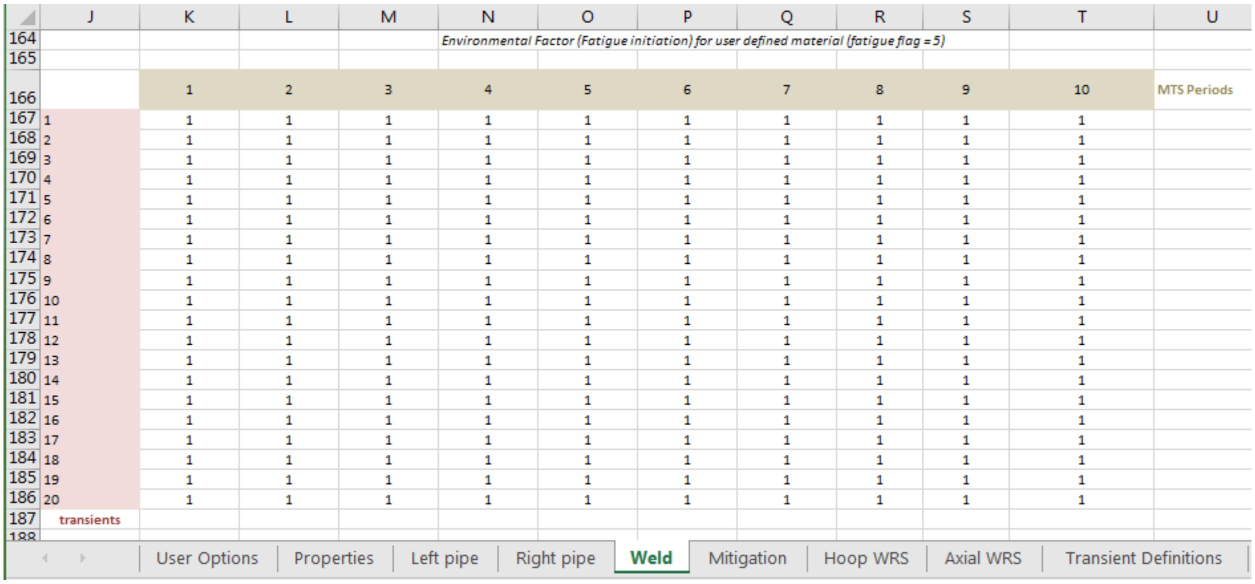
CORRELATIONS

	ID	name	ID	name	location	(rank) correlation
[-1, 1]	2594	Peak-to-Valley ECP Ratio, P-1	2595	Charact Width of Peak vs ECP, c	PWSCC growth properties	0.714
[-1, 1]	2551	Weibull Vertical Intrcpt Error, EpsC	2552	General Weibull Slope, Beta	PWSCC initiation	-0.905
[-1, 1]	2525	Strain Threshold, STH	2528	Co	Fatigue Initiation	1
[-1, 1]	2592	Comp-to-Comp Variab Factor, fcomp	2543	Multiplier proport. Const. A (DM1)	PWSCC growth/init	0
[-1, 1]	2592	Comp-to-Comp Variab Factor, fcomp	2547	Multiplier proport. Const B (DM2)		0
[-1, 1]	2592	Comp-to-Comp Variab Factor, fcomp	2551	Weibull Vertical Intrcpt Error, EpsC		0
[-1, 1]	2501	Yield Strength, Sigy	2502	Ultimate Strength, Sigu	General Properties	0.709
	---	---	---	---		---
[-1, 1]	2506	Material Init J-Resistance, Jic	2507	Material Init J-Resist Coef, C		0

Figure 2-9. Correlations table in the Properties worksheet the Inputs Workbook.

2.1.4.7 Environmental Factor Table

The Environmental Factor Table in the bottom left corner of the Weld worksheet is used when fatigue crack initiation is simulated and the fatigue crack initiation material flag (Global ID 2602) is set to “5,” indicating that a custom material is used. Figure 2-10 shows this table with its default values. This table allows the environmental factor in the fatigue crack initiation model to be customized for specific transients and operating conditions.



Environmental Factor (Fatigue initiation) for user defined material (fatigue flag = 5)

	1	2	3	4	5	6	7	8	9	10	MTS Periods
1	1	1	1	1	1	1	1	1	1	1	
2	1	1	1	1	1	1	1	1	1	1	
3	1	1	1	1	1	1	1	1	1	1	
4	1	1	1	1	1	1	1	1	1	1	
5	1	1	1	1	1	1	1	1	1	1	
6	1	1	1	1	1	1	1	1	1	1	
7	1	1	1	1	1	1	1	1	1	1	
8	1	1	1	1	1	1	1	1	1	1	
9	1	1	1	1	1	1	1	1	1	1	
10	1	1	1	1	1	1	1	1	1	1	
11	1	1	1	1	1	1	1	1	1	1	
12	1	1	1	1	1	1	1	1	1	1	
13	1	1	1	1	1	1	1	1	1	1	
14	1	1	1	1	1	1	1	1	1	1	
15	1	1	1	1	1	1	1	1	1	1	
16	1	1	1	1	1	1	1	1	1	1	
17	1	1	1	1	1	1	1	1	1	1	
18	1	1	1	1	1	1	1	1	1	1	
19	1	1	1	1	1	1	1	1	1	1	
20	1	1	1	1	1	1	1	1	1	1	
transients											

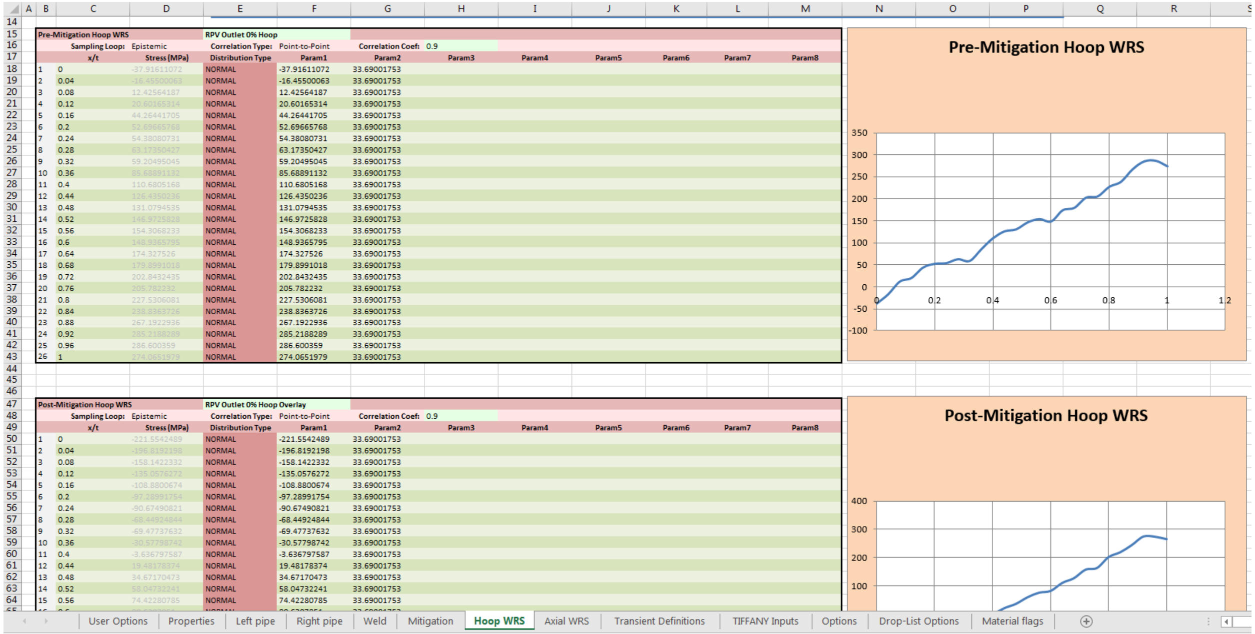
Figure 2-10. Environmental factor table in Properties worksheet of the Inputs Workbook.

2.1.5 Mitigation Worksheet

The Mitigation worksheet includes the 2700- and 2800-series Global IDs. It is the same as the Weld worksheet, except that all the inputs apply to the post-mitigation weld material (used only for inlay and overlay), and the Threshold SIF Scaling Factor (Global ID 2780) and Reference Temperature (Global ID 2797) inputs are not active. The inputs on the Mitigation worksheet are used when mitigation options are selected on the User Options worksheet, as described in Section 2.1.1.4.

2.1.6 Hoop and Axial Weld Residual Stress Worksheets

The Hoop and Axial WRS worksheets in the Inputs Workbook are used to define the WRS profiles. These two worksheets have the same organization and structure. The inputs on these worksheets are imported to the GoldSim model file as vectors that describe the WRS profile and are thus not assigned Global IDs, which are generally used to reference individual inputs only. Each worksheet provides WRS values at 26 points from the inside diameter of the pipe to the outside diameter, both before and after mitigation. The Hoop WRS worksheet is shown in Figure 2-11.



Pre-Mitigation Hoop WRS | RPV Outlet 0% Hoop

Sampling Loop: Epistemic | Correlation Type: Point-to-Point | Correlation Coef: 0.9

	x/t	Stress (MPa)	Distribution Type	Param1	Param2	Param3	Param4	Param5	Param6	Param7	Param8
1	0	-37.91611072	NORMAL	-37.91611072	33.69001753						
2	0.04	-16.45500063	NORMAL	-16.45500063	33.69001753						
3	0.08	12.42564187	NORMAL	12.42564187	33.69001753						
4	0.12	20.60165314	NORMAL	20.60165314	33.69001753						
5	0.16	44.26441705	NORMAL	44.26441705	33.69001753						
6	0.2	52.69665768	NORMAL	52.69665768	33.69001753						
7	0.24	54.38080731	NORMAL	54.38080731	33.69001753						
8	0.28	63.17350427	NORMAL	63.17350427	33.69001753						
9	0.32	59.20495045	NORMAL	59.20495045	33.69001753						
10	0.36	85.68891132	NORMAL	85.68891132	33.69001753						
11	0.4	110.6805168	NORMAL	110.6805168	33.69001753						
12	0.44	126.4350236	NORMAL	126.4350236	33.69001753						
13	0.48	131.0794535	NORMAL	131.0794535	33.69001753						
14	0.52	146.9725828	NORMAL	146.9725828	33.69001753						
15	0.56	154.3068233	NORMAL	154.3068233	33.69001753						
16	0.6	148.9365795	NORMAL	148.9365795	33.69001753						
17	0.64	174.327526	NORMAL	174.327526	33.69001753						
18	0.68	179.8991018	NORMAL	179.8991018	33.69001753						
19	0.72	202.8432435	NORMAL	202.8432435	33.69001753						
20	0.76	205.782232	NORMAL	205.782232	33.69001753						
21	0.8	227.5306081	NORMAL	227.5306081	33.69001753						
22	0.84	238.8363726	NORMAL	238.8363726	33.69001753						
23	0.88	267.1922936	NORMAL	267.1922936	33.69001753						
24	0.92	285.2188289	NORMAL	285.2188289	33.69001753						
25	0.96	286.600359	NORMAL	286.600359	33.69001753						
26	1	274.0651979	NORMAL	274.0651979	33.69001753						

Post-Mitigation Hoop WRS | RPV Outlet 0% Hoop Overlay

Sampling Loop: Epistemic | Correlation Type: Point-to-Point | Correlation Coef: 0.9

	x/t	Stress (MPa)	Distribution Type	Param1	Param2	Param3	Param4	Param5	Param6	Param7	Param8
1	0	-221.5542489	NORMAL	-221.5542489	33.69001753						
2	0.04	-196.8192198	NORMAL	-196.8192198	33.69001753						
3	0.08	-158.1422332	NORMAL	-158.1422332	33.69001753						
4	0.12	-135.0576272	NORMAL	-135.0576272	33.69001753						
5	0.16	-108.8800674	NORMAL	-108.8800674	33.69001753						
6	0.2	-97.28991754	NORMAL	-97.28991754	33.69001753						
7	0.24	-90.67490821	NORMAL	-90.67490821	33.69001753						
8	0.28	-68.44924844	NORMAL	-68.44924844	33.69001753						
9	0.32	-69.47737632	NORMAL	-69.47737632	33.69001753						
10	0.36	-30.57798742	NORMAL	-30.57798742	33.69001753						
11	0.4	-3.636797587	NORMAL	-3.636797587	33.69001753						
12	0.44	19.48178374	NORMAL	19.48178374	33.69001753						
13	0.48	34.67170473	NORMAL	34.67170473	33.69001753						
14	0.52	58.04732241	NORMAL	58.04732241	33.69001753						
15	0.56	74.42280785	NORMAL	74.42280785	33.69001753						

Figure 2-11. Hoop WRS worksheet of the Inputs Workbook.

In both worksheets, the first table defines the WRS profile for the pre-mitigation conditions and the second table defines the WRS profile for the post-mitigation conditions. The table headings include the WRS profile name, an indicator of the sampling loop (i.e., constant, epistemic, or aleatory) carried over from the Properties worksheet, and the coefficient for the point-to-point correlation used when uncertainty is applied. The profile name is used to label the WRS profile or to identify the profile in the database for use by the Sim Editor. The sampling loop is defined in the Properties worksheet (Global IDs 4350, 4351, 4352, and 4353), as discussed in Section 2.1.2.4. The point-to-point correlation used to apply uncertainty to the WRS profile employs a correlation coefficient to control the degree of the correlation applied from point-to-point. This coefficient can be changed by the user and is read by the GoldSim model file. The correlation coefficient is applied when the stress values are specified as uncertain and acts to correlate sampled adjacent stress values via a correlation matrix to ensure that the WRS values sampled represent a physically reasonable and smooth profile. A value of 0.9 or greater is typically recommended.

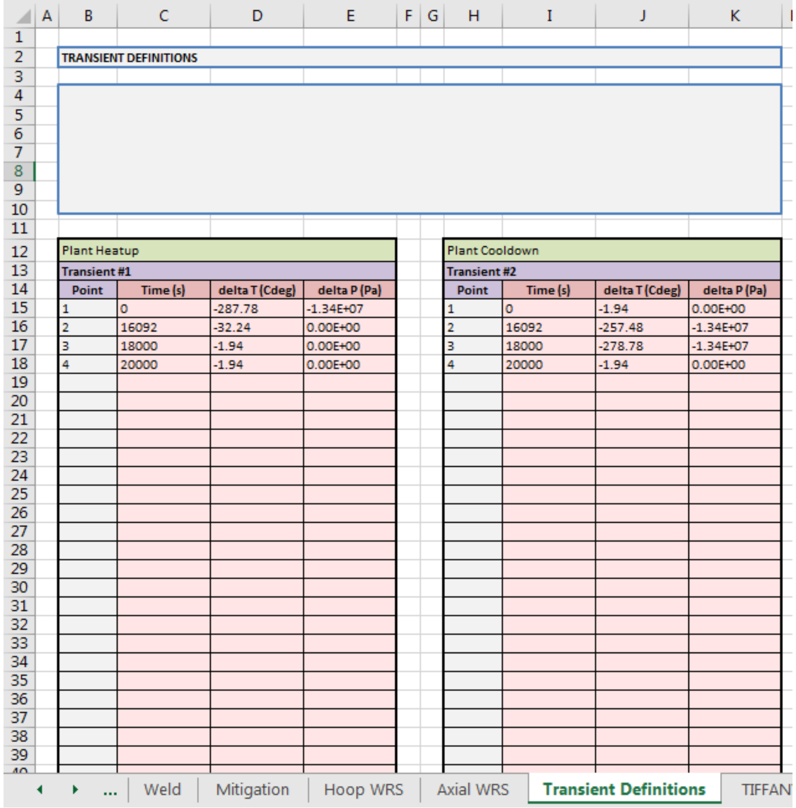
The location of the WRS data points is defined by the normalized distance from the inside diameter of the weld to the outside diameter. The notation describing this location is “x/t”. If the sampling loop is set to constant, the deterministic stress values in Column D are used. If the sampling loop is set to aleatory or epistemic, the means and standard deviations for the normal distribution parameters for each data point are used from Columns F and G, respectively. These parameters are read in by the GoldSim model file and are used in the point-to-point correlation algorithm to generate the sampled WRS profiles.

To aid the user in visualizing the WRS profile, a plot of the deterministic stress values vs. location is shown to the right of each table of inputs.

2.1.7 Transient Definitions Worksheet

Transient types are selected on the User Options worksheet as described in Section 2.1.1.5. This selection determines which transient inputs will be used by the TIFFANY module. Up to 20 transients may be defined as one of: none, Type I, Type I and II, or Type III). The inputs defined on this worksheet are not assigned Global IDs.

The Transient Definitions worksheet contains 20 tables for defining the 20 possible Type I thermal transients. These tables specify the changes in temperature and pressure relative to normal operating conditions as a function of time. The TIFFANY module uses these inputs to calculate stress intensity factors, maximum and minimum stresses, and rise time. Examples of two tables on this worksheet are shown in Figure 2-12.



TRANSIENT DEFINITIONS

Plant Heatup			
Transient #1			
Point	Time (s)	delta T (Cdeg)	delta P (Pa)
1	0	-287.78	-1.34E+07
2	16092	-32.24	0.00E+00
3	18000	-1.94	0.00E+00
4	20000	-1.94	0.00E+00

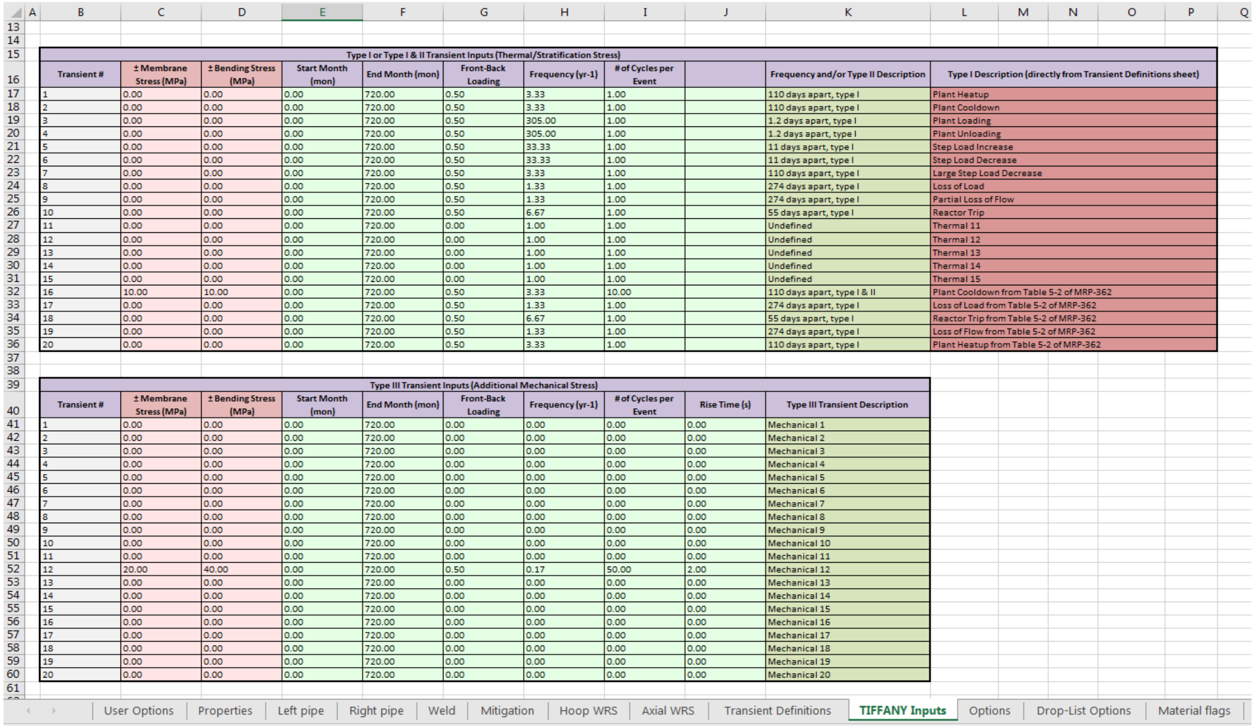
Plant Cooldown			
Transient #2			
Point	Time (s)	delta T (Cdeg)	delta P (Pa)
1	0	-1.94	0.00E+00
2	16092	-257.48	-1.34E+07
3	18000	-278.78	-1.34E+07
4	20000	-1.94	0.00E+00

Figure 2-12. Transient Definitions worksheet in the Inputs Workbook.

Each Type 1 transient is defined using a minimum of 2 and up to 50 points that include the time, change in temperature, and change in pressure from the constant values for normal operating temperature (Global ID 3101) and pressure (Global ID 3102) for Operating Period One. If uncertainty is applied to the normal operating temperature or pressure for Operating Period One, the user should also set the constant value to the mean or median value of the distribution, since only this value is used by the TIFFANY preprocessor. Type I frequencies and other scheduling parameters are input in the TIFFANY Inputs worksheet because Type I and Type II frequencies must align as these transients occur together. Greater detail regarding the Transient Definitions worksheet can be found in the User Manual [14].

2.1.8 TIFFANY Inputs Worksheet

As stated in Section 2.1.7, up to 20 transients may be defined on the User Options worksheet. The TIFFANY Inputs worksheet defines all the frequency and scheduling parameters for these transients. It also defines the loads for the Type II portion of the Type I and II transients and the Type III transients. In addition, the worksheet defines the transient uncertainty multipliers for the stress intensity factor solution values. The top portion of the TIFFANY Inputs worksheet is shown in Figure 2-13. The inputs defined on this worksheet are not assigned Global IDs.



Type I or Type I & II Transient Inputs (Thermal/Stratification Stress)

Transient #	± Membrane Stress (MPa)	± Bending Stress (MPa)	Start Month (mon)	End Month (mon)	Front-Back Loading	Frequency (yr-1)	# of Cycles per Event		Frequency and/or Type II Description	Type I Description (directly from Transient Definitions sheet)
1	0.00	0.00	0.00	720.00	0.50	3.33	1.00		110 days apart, type I	Plant Heatup
2	0.00	0.00	0.00	720.00	0.50	3.33	1.00		110 days apart, type I	Plant Cooldown
3	0.00	0.00	0.00	720.00	0.50	305.00	1.00		1.2 days apart, type I	Plant Loading
4	0.00	0.00	0.00	720.00	0.50	305.00	1.00		1.2 days apart, type I	Plant Unloading
5	0.00	0.00	0.00	720.00	0.50	33.33	1.00		11 days apart, type I	Step Load Increase
6	0.00	0.00	0.00	720.00	0.50	33.33	1.00		11 days apart, type I	Step Load Decrease
7	0.00	0.00	0.00	720.00	0.50	3.33	1.00		110 days apart, type I	Large Step Load Decrease
8	0.00	0.00	0.00	720.00	0.50	1.33	1.00		274 days apart, type I	Loss of Load
9	0.00	0.00	0.00	720.00	0.50	1.33	1.00		274 days apart, type I	Partial Loss of Flow
10	0.00	0.00	0.00	720.00	0.50	6.67	1.00		55 days apart, type I	Reactor Trip
11	0.00	0.00	0.00	720.00	0.00	1.00	1.00		Undefined	Thermal 11
12	0.00	0.00	0.00	720.00	0.00	1.00	1.00		Undefined	Thermal 12
13	0.00	0.00	0.00	720.00	0.00	1.00	1.00		Undefined	Thermal 13
14	0.00	0.00	0.00	720.00	0.00	1.00	1.00		Undefined	Thermal 14
15	0.00	0.00	0.00	720.00	0.00	1.00	1.00		Undefined	Thermal 15
16	10.00	10.00	0.00	720.00	0.50	3.33	10.00		110 days apart, type I & II	Plant Cooldown from Table 5-2 of MRP-362
17	0.00	0.00	0.00	720.00	0.50	1.33	1.00		274 days apart, type I	Loss of Load from Table 5-2 of MRP-362
18	0.00	0.00	0.00	720.00	0.50	6.67	1.00		55 days apart, type I	Reactor Trip from Table 5-2 of MRP-362
19	0.00	0.00	0.00	720.00	0.50	1.33	1.00		274 days apart, type I	Loss of Flow from Table 5-2 of MRP-362
20	0.00	0.00	0.00	720.00	0.50	3.33	1.00		110 days apart, type I	Plant Heatup from Table 5-2 of MRP-362

Type III Transient Inputs (Additional Mechanical Stress)

Transient #	± Membrane Stress (MPa)	± Bending Stress (MPa)	Start Month (mon)	End Month (mon)	Front-Back Loading	Frequency (yr-1)	# of Cycles per Event	Rise Time (s)	Type III Transient Description
1	0.00	0.00	0.00	720.00	0.00	0.00	0.00	0.00	Mechanical 1
2	0.00	0.00	0.00	720.00	0.00	0.00	0.00	0.00	Mechanical 2
3	0.00	0.00	0.00	720.00	0.00	0.00	0.00	0.00	Mechanical 3
4	0.00	0.00	0.00	720.00	0.00	0.00	0.00	0.00	Mechanical 4
5	0.00	0.00	0.00	720.00	0.00	0.00	0.00	0.00	Mechanical 5
6	0.00	0.00	0.00	720.00	0.00	0.00	0.00	0.00	Mechanical 6
7	0.00	0.00	0.00	720.00	0.00	0.00	0.00	0.00	Mechanical 7
8	0.00	0.00	0.00	720.00	0.00	0.00	0.00	0.00	Mechanical 8
9	0.00	0.00	0.00	720.00	0.00	0.00	0.00	0.00	Mechanical 9
10	0.00	0.00	0.00	720.00	0.00	0.00	0.00	0.00	Mechanical 10
11	0.00	0.00	0.00	720.00	0.00	0.00	0.00	0.00	Mechanical 11
12	20.00	40.00	0.00	720.00	0.50	0.17	50.00	2.00	Mechanical 12
13	0.00	0.00	0.00	720.00	0.00	0.00	0.00	0.00	Mechanical 13
14	0.00	0.00	0.00	720.00	0.00	0.00	0.00	0.00	Mechanical 14
15	0.00	0.00	0.00	720.00	0.00	0.00	0.00	0.00	Mechanical 15
16	0.00	0.00	0.00	720.00	0.00	0.00	0.00	0.00	Mechanical 16
17	0.00	0.00	0.00	720.00	0.00	0.00	0.00	0.00	Mechanical 17
18	0.00	0.00	0.00	720.00	0.00	0.00	0.00	0.00	Mechanical 18
19	0.00	0.00	0.00	720.00	0.00	0.00	0.00	0.00	Mechanical 19
20	0.00	0.00	0.00	720.00	0.00	0.00	0.00	0.00	Mechanical 20

Figure 2-13. TIFFANY Inputs worksheet the Inputs Workbook.

The transient frequency and scheduling parameters include the start month, end month, front-back loading, frequency, and number of cycles per transient event. The front-back loading parameter shifts the scheduled time of the transient within the period defined by the transient frequency. The frequency defines how many times the transient occurs within one year. The number of cycles per event defines how many cycles the transient has each time the transient occurs. For Type III transients, the rise time must also be defined. For Type I or Type I and II transients, the rise time is calculated by the TIFFANY module. The rise time specifies how quickly the strain rate reaches a predefined value for fatigue initiation and growth. Since Type III transients are meant to represent instantaneous mechanical loads, a rise time of one second or less is appropriate.

Additional columns to the right of the inputs are included for the user's information only and are not used in the Framework calculations. In Column K, the user may describe the transient in more detail. For Type I or Type I and II transients, Column L carries over the name of the Type I transient as defined by the user in the tables of the Transient Definitions worksheet.

The transient uncertainty multipliers are defined in the bottom portion of the TIFFANY Inputs worksheet. The purpose of these multipliers is to allow uncertainty to be applied to the stress intensity factors calculated by the TIFFANY module, which is a completely deterministic calculation made prior to running the probabilistic simulation. An uncertainty multiplier may be defined for each transient as either a constant or as a distribution. The multiplier, whether sampled or constant, is multiplied in the GoldSim model file by the stress intensity factor values selected from the TIFFANY-generated look-up tables prior to calling of the CGR module.

2.1.9 Options Worksheet

The Options worksheet is read-only and cannot be changed by the user. It is used to communicate the user inputs defined on the User Options, Properties, Left Pipe, Right Pipe, Weld, Mitigation, Hoop WRS, Axial WRS, Transient Definitions, and TIFFANY Inputs worksheets to the GoldSim model file. The functions on this worksheet transform the user inputs for the sampling loop selection, Importance Sampling selection, and Importance Sampling target probability level into numerical values that can be read-in as vectors rather than as individual cells. These vectors are then evaluated as required in the GoldSim model file.

2.1.10 Drop-List Options Worksheet

The Drop-List Options worksheet is read-only and cannot be changed by the user. It is provided for the user's reference. This worksheet lists all the options for each of the drop-lists used throughout the Inputs Workbook. Drop-lists are used for the WRS Correlation Type, Data Source, Importance Sampling, Spreadsheet Name, Time Units, and Distribution Type inputs. This worksheet also lists the order and definition of the probability distribution parameter entries expected by the GoldSim model file. This parameterization must be used when specifying probability distributions for uncertain inputs throughout the Inputs Workbook so that the probability distributions are sampled correctly.

2.1.11 Materials Flags Worksheet

The Material Flags worksheet is read-only and cannot be changed by the user. It is provided for the user's reference. This worksheet lists the material type flags used by the CI-SCC, CIF, and CGR modules that the user must input on the Left Pipe, Right Pipe, Weld, and Mitigation worksheets, as applicable. This worksheet also includes some recommendations with respect to material parameters so that the inputs remain physically reasonable.

2.2 Preprocessor

The Preprocessor is implemented as a Microsoft Excel Add-In called "xLPR_Preprocessor.xll". This Add-In is used to run the TIFFANY and LEAPOR module DLLs. The TIFFANY module generates look-up tables of stress intensity factors, rise times, and maximum and minimum stresses for fatigue crack initiation and fatigue crack growth. The LEAPOR module generates leak rate look-up tables as a function of COD, crack length, and pipe thickness. The calculations performed by the TIFFANY and LEAPOR modules are deterministic. As such, application of uncertainty to these results is done within the GoldSim model file.

The Preprocessor Add-In is invoked by double clicking the .xll file in Windows and enabling the Add-In for the current session when prompted by the security warning. A new tab called "xLPR Preprocessing" will appear in the Microsoft Excel ribbon within the Inputs Workbook, allowing the preprocessor to be run by the user. This tab is shown in Figure 2-14 below.

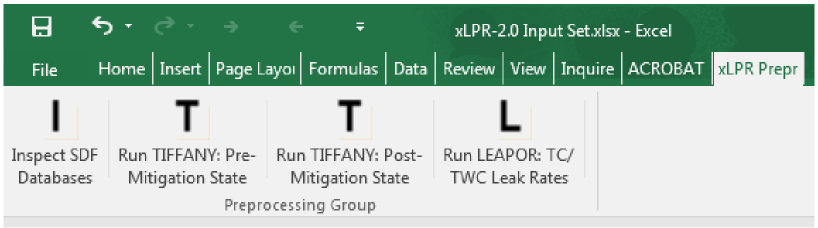


Figure 2-14. Preprocessor ribbon shown in the Inputs Workbook.

The xLPR Preprocessing tab in the Inputs Workbook presents five options:

(1). As a result of running TIFFANY or LEAPOR, results are generated as both text files to be read-in by the GoldSim model file and as an .sdf binary file. The "Inspect SDF Databases" option allows the user to compare two .sdf files, check the inputs used to generate the results, and plot the leak rates from a previous LEAPOR run.

(2). The "Run TIFFANY Pre-Mitigation State" option runs the TIFFANY module based on the pre-mitigation properties and pipe geometry defined in the Inputs Workbook. An .sdf file for use in database inspection and a set of .txt files that are read by the GoldSim model file are generated as a result.

(3). The "Run TIFFANY Post-Mitigation State" option runs the TIFFANY module based on the post-mitigation properties and pipe geometry defined in the Inputs Workbook. An .sdf file for use in database inspection and a set of .txt files that are read by the GoldSim model file are generated as a result.

(4). The "Run LEAPOR TC/TWC" option runs the LEAPOR module based on the inputs defined in the Inputs Workbook. An .sdf file for use in database inspection and a set of .txt files that are read by the GoldSim model file are generated as a result.

(5). The GoldSim model reads the text files generated as a result of running options 2 through 4 above to create a set of look-up tables that are accessed when needed within a realization. The Preprocessor places the text files in local subdirectories named "TIFFANY" and "LEAPOR." If these subdirectories do not already exist, the Preprocessor will create them.

2.2.1 TIFFANY Module

The TIFFANY module generates look-up tables of stress intensity factors, rise times, and maximum and minimum stresses that are used by the Framework for calculations related to fatigue crack initiation and fatigue crack growth. The TIFFANY module is run by the Preprocessor once for pre-mitigation conditions and once for post-mitigation conditions.

The TIFFANY module is only required to be run when fatigue crack initiation, fatigue growth, or both are modeled, as selected by the user on the User Options worksheet. The TIFFANY module receives all its inputs from the Inputs Workbook. Color-coding in the Inputs Workbook points the user to the inputs required by the TIFFANY module. For normal operating stresses or loads, which may be defined as distributions on the Properties worksheet, the TIFFANY module uses the deterministic values as inputs. Transient definitions and stresses are input as deterministic values on the Transient Definitions and TIFFANY Inputs worksheets.

Three different transient combinations are available for TIFFANY module calculations: Type I, Type I and II, and Type III. Up to 20 transients of any of these combinations may be used to simulate fatigue loads occurring during plant operation. The transient types and the TIFFANY module are described in greater detail in the User Manual [14]. Additional information regarding the use of TIFFANY module outputs in the Framework calculations is provided in the description of the Deterministic Model given in Section 3.3.

2.2.2 LEAPOR Module

The LEAPOR module is used to generate leak rate look-up tables as a function of COD, crack length, and pipe thickness. It calculates the leak rate from TWCs and transitioning cracks (TRCs). The module is run by the Preprocessor as discussed in Section 2.2. The LEAPOR module receives all its inputs from the Inputs Workbook. As with the TIFFANY module, color coding is used within the Inputs Workbook to help the user identify the inputs required by LEAPOR.

3. GOLDSIM MODEL FILE

The xLPR V2.0 GoldSim file is a GoldSim 'model' that provides a landing platform that links the Inputs Interface with the probabilistic structure that is used to propagate uncertainty through the deterministic model and provides results using outputs and postprocessing capabilities. The GoldSim development environment uses 'elements' as the basic building blocks of a model. Elements of various types are used for the xLPR V2.0 model depending on the required functionality. A description of each of the main elements used in the xLPR V2.0 GoldSim file is provided in this section. Greater detail regarding each of these elements is found in the GoldSim User Manual [16].

In the GoldSim environment, the GoldSim model file is editable. Graphical user interfaces created using 'dashboard elements' direct the user to important functions including setting up a simulation, which is accomplished using the Global Settings Dashboard, viewing results through the Results Dashboards, and monitoring errors using the Error Tracking Dashboards. Results and errors are available for both axial and circumferential cracks. However, these dashboards are not the only way that a user can interact with this portion of the Framework. Users can also use the built-in GoldSim functions to navigate through the entire calculation process contained within the GoldSim model file, which can be notionally compared to the Windows file explorer environment. Although the Inputs Interface defines nearly all the inputs needed to run a simulation, some additional inputs must be set within the GoldSim model file, and they can be accessed using the Global Settings Dashboard. These inputs are related to the probabilistic structure and are described in Section 3.1.

The GoldSim model file utilizes 'submodel elements' to create a nested-loop structure to separate epistemic and aleatory uncertainties. Submodel elements are used to embed a GoldSim model within a larger GoldSim model. The two models then interact via a defined input and output interface. In the xLPR V2.0 context, the epistemic loop is located at the top level of the GoldSim model file known as the Model Root. At this level, simulation information defined by the user is imported from the Inputs Interface and epistemic sampling calculations are made. This information is then passed to a submodel element that defines the aleatory loop, which is nested within the epistemic loop. The landing platform and the deterministic model exist inside the aleatory submodel element. Greater detail regarding the implementation of epistemic and aleatory uncertainty using the probabilistic structure is given in Section 3.1.

Both the epistemic and aleatory levels of the GoldSim model file are organized using 'container' elements. These elements are used to hierarchically organize the remaining elements that serve as the building blocks for the calculations performed by the Framework. Containers can use conditionality to determine when the calculations they contain are made based on logical expressions.

Several elements are used to read and store information related to simulation inputs and intermediate values used for the Framework calculations. 'Spreadsheet' elements are used to read in information from the Inputs Interface. 'Data' elements are used to store scalar, vector, or matrix data for inputs and intermediate values. 'Lookup Table' elements are used to read the look-up tables generated during execution of the Preprocessor. 'Stochastic' elements are used to define uncertain data as probability distributions. 'Previous value' elements are used to store the value of inputs at the previous model update as needed for some calculations.

GoldSim stock and function elements are also used to perform calculations. 'Expression' elements are used to define mathematical or logical expressions that perform functions and calculations used throughout the GoldSim model file. ‘Script’ elements implement more detailed calculations in the Framework using a procedural programing language environment. ‘Integrator’ elements are used to integrate values through the time history of the simulation. ‘External’ elements are used to interface with the module DLLs. ‘Selector’ elements use logical expressions to select a value for use in subsequent calculations.

Event elements are used to implement the effects of discrete events that occur throughout the simulation. They include ‘triggered events’ elements to generate signals based on certain conditions, ‘information delay’ elements to delay signals as needed, ‘status’ elements to generate a logical condition in response to a defined condition, ‘discrete change’ elements to generate a discrete signal in response to a defined condition, and ‘interrupt’ elements to interrupt a simulation when a specified condition occurs.

Finally, result elements are used to save and display simulation results. ‘Time history result’ elements display simulation results saved over the time history, while ‘array result’ elements display vectors and matrices of results.

The GoldSim model file consists of four major components: a probabilistic structure, which is described in Section 3.1, that is used to propagate uncertainty through a deterministic model, which is described in Section 3.3. These two components are linked together with the Inputs Interface using the landing platform, which is described in Section 3.2. The final component is the simulation results displayed using outputs and postprocessing capabilities as described in Section 3.4.

3.1 Probabilistic Structure

The probabilistic structure of the Framework is located within the GoldSim model file. This probabilistic structure uses sampling capabilities native to GoldSim that are controlled by user inputs in the Inputs Workbook and by additional options that are set using the Global Settings Dashboard in the GoldSim model file. These additional options include the Epistemic Sample Size (Global ID 0101), Epistemic Random Seed (Global ID 0102), Aleatory Random Seed (Global ID 0108), and the selection of LHS for the epistemic and aleatory loops. The Framework’s probabilistic structure is used to propagate uncertainty through the Framework’s deterministic model.

The sampling algorithm for the Framework is structured into two nested loops: an outer loop that corresponds to epistemic uncertainty and an inner loop that corresponds to aleatory uncertainty. Aleatory uncertainty is classified as natural, random variation in the input considered and is deemed to be irreducible. Epistemic uncertainty is classified as lack of knowledge about the behavior of an input, which could theoretically be reduced with more information. In probabilistic analyses aiming to characterize the likelihood of an adverse event (e.g., pipe rupture), aleatory uncertainty represents the risk of an adverse event occurring, whereas epistemic uncertainty represents the confidence about that risk.

The sampling algorithm propagates epistemic and aleatory uncertainties. For each realization and for both sampling loops, GoldSim samples a vector of values from a uniform distribution between 0 and 1, whose size matches the total number of possible random input variables. The number of samples (realizations) for each epistemic and aleatory loop is selected by the user. These uniform samples are taken using either SRS, LHS, or DPD sampling depending on the options that the user selects.

Each input is thus associated with two uniform samples, one for the epistemic loop and one for the aleatory loop, though only one of these samples is used based on the user's classification of each input as either epistemic or aleatory. The uniform samples represent the quantile values that are used to sample the selected distributions for each input. The user selection in the Input Set determines which of these quantiles will be used to generate the input value for each realization. If an input is set to epistemic, the epistemic quantile value sampled will be used to generate the input value for each unique epistemic realization and all its associated aleatory realizations. If an input is set to aleatory, the aleatory quantile value sampled for each aleatory realization will be used to generate the input value.

The sampled quantile values are further modified based on the selection of additional sampling inputs or correlations, including the use of DPD sampling or Importance Sampling. Once the quantile values have been modified to implement all the sampling options, the quantiles are applied to the defined probability distributions to obtain the input values for each realization. These values are then passed to the deterministic model via the landing platform.

3.1.1 Uncertainty Characterization

xLPR V2.0 models the formation and growth of cracks to determine the probability of rupture for piping welds susceptible to PWSCC or fatigue-induced cracking. It is understood, however, that the probability of rupture depends on many uncertain factors that make the probability of rupture difficult to accurately predict. The Framework accounts for many of these uncertainties to produce probability of rupture estimates (as well as estimates for other significant metrics related to the problem of interest) that include a characterization of uncertainty.

To account for uncertainty, the Framework combines deterministic modules with a random sampling structure. The modules calculate aspects of the different physical processes, such as crack initiation, crack growth, and crack coalescence, that are required to predict rupture. Inputs to each module include information about the weld, materials, operating conditions, etc., and also appropriate model calibration parameters. There are two types of uncertainty associated with each module that must be considered: model uncertainty and parameter uncertainty. Model uncertainty refers to the uncertainty induced in the quantity being estimated by the specific mathematical model used to represent the physical process of interest. When two or more competing models exist, it may not be clear which model is best, if any. Different models will provide different results. Parameter uncertainty refers to uncertainty in the specific values of fitting or calibration parameters that exist in each model. Though the model form may remain unchanged, simulations that vary the value of the model parameters lead to different results, reflecting uncertainty induced by parameter selection.

Uncertainty is primarily accounted for in xLPR V2.0 through implementation of a sampling strategy. Each deterministic model depends upon specific inputs, some of which describe model fitting to specific datasets (model parameters) and some of which describe the geometry, material, and environmental variables accounted for by the model. Uncertainty is implemented in the Framework by sampling input values randomly from probability distributions defined for each uncertain input and running each sampled set of values through the deterministic models to generate a single output. Each set of sampled input values leads to a different model result that is considered equally valid based on the current state-of-knowledge. A different random set of input values is generated for each realization such that the compilation of realizations results in a distribution of outputs that describes the expected mean tendency and uncertainty on that output.

With respect to model uncertainty, each module was developed by experts to be as accurate as possible while maintaining reasonable computational costs. Because of this constraint, model uncertainty that cannot be accounted for by sampling from input probability distributions was predominantly beyond the scope of the xLPR V2.0 developmental effort. One exception is the PWSCC initiation method. Three initiation models are implemented in xLPR V2.0 and the user may specify which is used in the analysis. This facilitates examination of the model uncertainty by allowing for the comparison of simulations using the three different models.

The implementation of uncertainty in the Framework begins with the user-specified sampling options and input distributions. These specifications are made in the Inputs Workbook and the Global Controls Dashboard. Sampling options include the sample sizes to be used, the amount of discretization for spatially varying inputs, and sampling methods (e.g., SRS, LHS, Importance Sampling, and DPD). Other user options are selected specifically for each input. For inputs that can be uncertain, the user must select one of three values for uncertainty implementation: constant (deterministic or no uncertainty), epistemic, or aleatory.

If the user specifies an input as constant, no sampling or uncertainty is applied by the Framework. If the user specifies that an input has epistemic or aleatory uncertainty, the user must also specify a probability distribution. The Framework imports these specifications by importing the values from the Inputs Workbook into the appropriate location within the GoldSim model file. The imported values are assigned to elements to be used in selecting the final values for each input. A sampled quantile is evaluated using the user-specified distribution to obtain a sample for each input. This is done for each uncertain input for each realization.

Conceptually, the user specifies a probability distribution for each uncertain input and the Framework samples from that distribution for each simulation. Computationally, however, the sampling is performed by randomly sampling quantiles from a uniform distribution that are transformed into samples from the specified probability distributions. This is a standard practice for randomly sampling from distributions that are not uniform and allows other options to be applied directly to the quantile samples.

The distinction between epistemic and aleatory uncertainty is implemented using nested sampling loops. This concept is illustrated in Figure 3-1. To consider the implications of separating aleatory and epistemic uncertainty, consider estimating the probability of circumferential crack rupture, for example. Without separating uncertainties, the Code could be run n times to calculate the number of ruptures, x, and an estimate for the probability of circumferential crack rupture would be x/n. By separating uncertainty, there are n_a aleatory samples for a single epistemic realization. For $i = 1, \dots, n_e$, the number of ruptures x_i in the n_a aleatory samples gives an estimated probability of rupture for each epistemic realization i. In other words, for epistemic realization i, the estimated probability of rupture is x_i/n_a. Separation of uncertainty allows the user to assess how these rupture estimates change due to epistemic uncertainty, though the best-estimate of the probability of rupture over both aleatory and epistemic uncertainties can also be obtained by averaging the n_e different probability estimates.

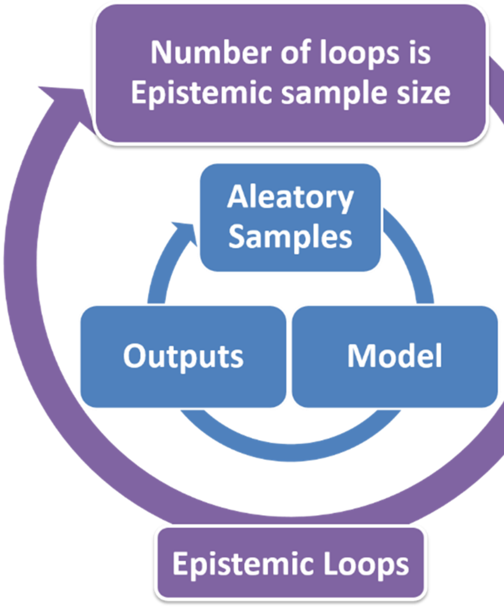


Figure 3-1. Illustration of the nested sampling loop structure implemented in the Framework. For each epistemic sample, the aleatory loop is run many times according to the number of aleatory samples selected by the user.

The Framework estimates the occurrence of rupture, occurrence of crack, and other quantities of interest for each aleatory realization as a function of time and uses these estimates to calculate the probability of the event occurring at or before each time step throughout the simulation. The cumulative distribution functions (CDFs) from each epistemic realization are presented individually as the output of the simulation and averaged to provide an overall estimate of the CDF.

The CDF results are contained in the result elements at the epistemic level and can be used to communicate the uncertainty in the probability of rupture, for example. The average CDF itself is the estimated probability of rupture at or before that time at each time step. The population of CDFs, one from each epistemic realization, represent the uncertainty around this estimate that is introduced by uncertainty in the inputs. An example of this type of result taken from the Scenario Analysis Report [17] is shown in Figure 3-2. In this figure, the probability of occurrence for circumferential crack rupture is shown for each epistemic realization over time. Summary statistics, including the mean and quantiles of interest, are also displayed. These statistics characterize the uncertainty in the estimate of the probability of rupture.

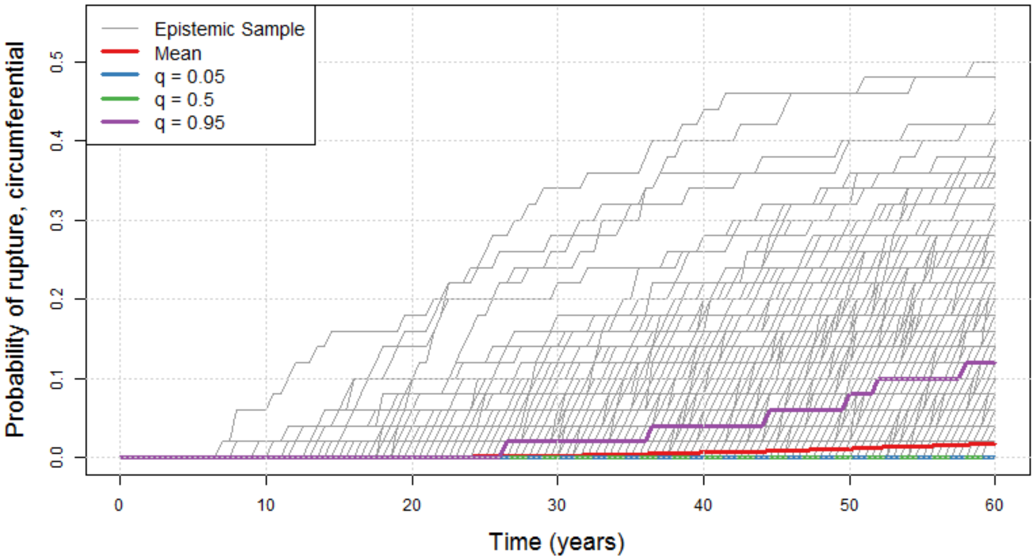


Figure 3-2. Probability of occurrence of circumferential rupture for each epistemic realization (grey), the mean (red), and the 5th (blue), 50th (green), and 95th (purple) percentiles for xLPR Scenario Analysis, Scenario 3

Proper interpretation of these results requires that the input samples are obtained using random sampling. Without randomness, the samples could be drawn in a biased way to lead to specific results. The Framework implements multiple random sampling options. These options, which are detailed in Section 3.1.3, are SRS, LHS, Importance Sampling, and DPD sampling. Selection of the sampling method is problem-specific and can affect the uncertainty in the estimate of the probability of rupture or other output metrics as discussed in the next section.

3.1.2 Uncertainty Treatment Biases

The most basic sampling method implemented by the Framework is SRS. It uses uniform sampling of quantiles in which each sample has equal probability. Additional sampling options are implemented to allow more focused random sampling to support convergence of results, meaning that the estimate of the quantity of interest, for instance probability of rupture, can be made with greater precision for the same number or fewer samples. With SRS, convergence can be slow, meaning that many realizations may be required to reach a result that is representative and statistically stable. SRS may also fail to sample sufficiently in the tails of the probability distributions, which can lead to simulation results that do not include extreme but possible input combinations. For example, a probability distribution on operating pressure is likely centered around the design pressure. Very high pressures that could lead to rupture, though possible, may never be sampled using SRS because the sample size is simply too small to ensure samples of high operating pressures. In this case, SRS would fail to include potentially important crack behaviors and subsequent ruptures in estimates of uncertainty.

An additional concern when using sampling methods is obtaining a collection of samples that is representative of the desired distribution. This is demonstrated in the plots in Figure 3-3., which show the histograms of uniform random samples taken from the same distribution with increasing sample sizes. Though the samples are all drawn from the same distribution, the first plot does not appear uniform. The second plot looks more uniform, though potentially bimodal, and the final plot is much more representative of a uniform distribution.

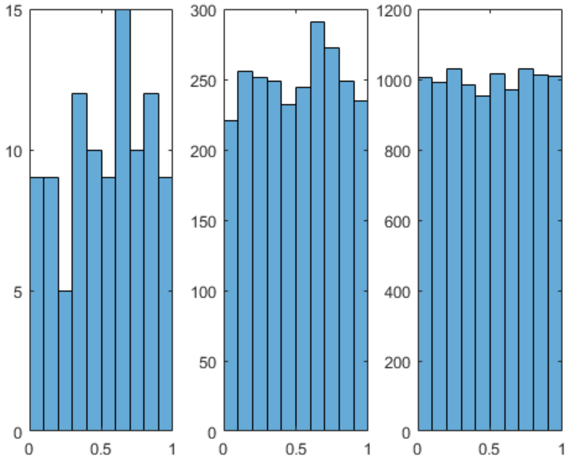


Figure 3-3. Notional example showing convergence to a uniform distribution with increasing sample size.

Additional sampling options address these issues by encouraging faster convergence than results obtained using SRS and forcing the Framework to draw samples in the extreme regions of the probability distributions.

The first and most straightforward sampling method that can be used to improve on SRS is LHS. SRS applied to a uniform distribution will eventually, with a large enough number of samples, result in samples that are uniformly distributed. LHS uses a stratified sampling algorithm to ensure that samples are uniformly distributed over the range of the input regardless of the sample size. For example, if the sample size is 10 and the distribution is uniform between 0 and 1, LHS will draw one sample randomly between 0 and 0.1, one sample between 0.1 and 0.2, etc. Regarding uncertainty, this sampling procedure leads to samples that are representative of the specified distribution with smaller sample sizes and hence leads to faster convergence of results. LHS is a global sampling option in GoldSim, meaning that it can be used for all epistemic and aleatory samples, but it cannot be used on a subset of the uncertain inputs. LHS may be applied to the epistemic loop, aleatory loop, or both as described in Section 3.1.3.

To obtain samples in regions of the input space that are of interest but are not highly probable, the Framework includes the ability to use Importance Sampling. While the use of LHS is relatively straightforward, the use of Importance Sampling requires much more analysis from the user. In general, the Framework obtains samples from distributions by generating a uniformly distributed random sample between 0 and 1 and applying a transformation to obtain a sample from the desired distribution. Importance Sampling differs from this method in that it samples from a distribution that is not uniform between 0 and 1. For example, half of the samples would be drawn from the interval from 0 to 0.9, and the other half would be drawn from the interval from 0.9 to 1. These samples are then transformed, just as they would be if they were sampled uniformly, to obtain samples from the desired probability distribution. In this example, if the target distribution were normal, half of the resulting samples would be above the 90th percentile of the normal distribution (i.e., concentrated in the upper tail).

The Importance Sampling distribution implemented in the Framework uses a user-specified quantile for each input selected for Importance Sampling. If Importance Sampling is applied, half of the samples are taken within an interval about this selected quantile. The width of the

Importance Sampling interval is a function of the number of variables selected for Importance Sampling. This implementation ensures that if a user selects too many variables for Importance Sampling (generally greater than 5), the Importance Sampling will be rendered effectively insignificant to avoid skewing the simulation results. Greater detail regarding this feature can be found in the SDD [8].

In addition to LHS and SRS, the Framework can also perform DPD sampling, which is a stratified sampling method similar to LHS. There are two primary differences between LHS and DPD. First, DPD uses a user-specified number of strata to stratify the range of the input for sampling, whereas LHS uses the number of samples. Second, after partitioning the sample space, LHS samples randomly within each stratum and DPD uses the conditional mean of the stratum. As a result, LHS samples are on average more uniformly distributed than SRS, whereas DPD results in samples that are always uniformly distributed but take on fewer unique values. This can be useful when the simulation sample size is limited, however, it means that replicate simulations cannot be used to improve estimates in the tails of the distribution. Such improvements require the use of Importance Sampling in addition to DPD.

As stated above, advanced sampling options are applied to allow more focused random sampling to support convergence of results, meaning that the estimate of the quantity of interest can be made with greater precision for the same number or fewer samples. In some situations, large numbers of samples may still be needed to estimate very rare quantities of interest, even when advanced sampling options are applied. The memory limitations of the GoldSim model file may prohibit such calculations. In these cases, realizations can be run in different simulations with different random seeds for the epistemic and aleatory loops and then combined to achieve a large enough sample size.

3.1.3 Sampling Technique Implementation

The following subsections discuss how the various sampling techniques are implemented within the Framework.

3.1.3.1 Simple Random Sampling

SRS is implemented in the Framework using the stochastic elements in GoldSim. SRS is used to sample quantile values from a uniform distribution between 0 and 1 such that each sample has the same probability as any other sample in the set. GoldSim accomplishes this by employing a system of random seeds and a linear congruential generator. This system is summarized below and described in greater detail in the GoldSim User Manual [16]. SRS is used by default when LHS is not selected in the Global Settings Dashboard for the epistemic and aleatory loops, or when DPD is not selected in the Inputs Workbook.

Simple random samples are generated using the model, run, realization, element, and combined random seeds. The model seed was generated randomly based on the system clock when the Framework was created and remains the same. The epistemic and aleatory random run seeds are set by the user in the Simulation Settings menu, which can be opened from the Global Settings Dashboard. The random seeds can be changed at the beginning of any simulation but remain the same for all realizations within that simulation. Like the model seed, an element seed is generated for each stochastic element randomly based on the system clock when the stochastic element was initially created and remains the same thereafter. Hence, for any simulation performed with xLPR V2.0, the model, run, and element seeds are all constant.

At the beginning of the simulation, the realization seed is generated by combining the model seed and run seed. This realization seed is then used again at the beginning of the subsequent realization to generate a new realization seed. At the beginning of the realization, the realization seed is combined with the element seed to create the combined seed, which is used by the random number generator for sampling and to create a new combined seed for the next random number that is generated. Hence, the realization seed changes for each realization and the combined seed changes every time a random sample is taken. Greater detail regarding the combination of and use of random seeds by GoldSim can be found in the GoldSim User Manual [16].

For the user, it is important to note that the model and element seeds are fixed in the Framework and, due to the dependencies between seeds, the run seed alone allows for exact repetition of simulations. Once the run seed is fixed by the user for the epistemic and aleatory loops (or when the Framework default seeds are used), the realization seed that is generated will always be the same, and hence the combined seed will always be the same. This means that a simulation with precisely the same inputs will give the same results any time it is run using the same epistemic and aleatory run seeds. When the run seeds are changed, the realization seed and the combined seed also change, resulting in different simulation results.

3.1.3.2 Latin Hypercube Sampling

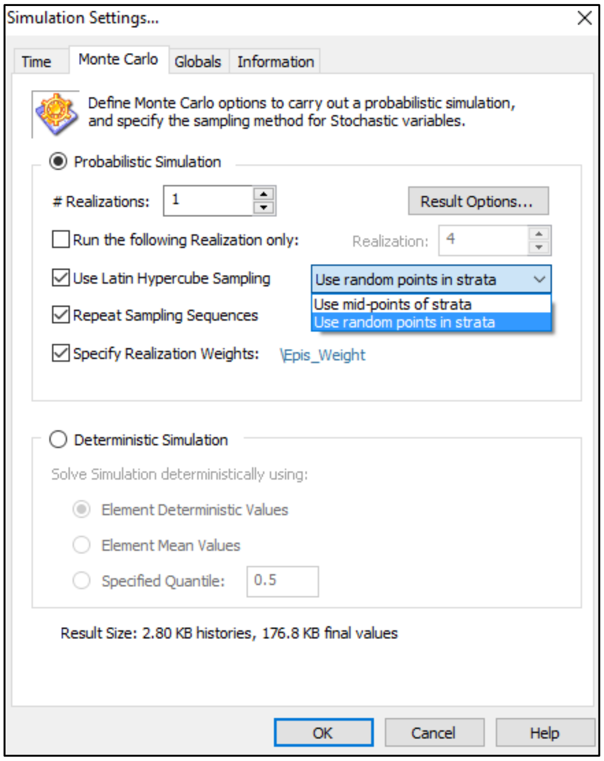
LHS is a stratified sampling technique that forces random samples to be drawn in such a way that each stratum is equally likely and contains one sample, though the sample is randomly located within the stratum. With SRS, if n-samples are drawn from a uniform distribution between 0 and 1, there may be more samples between 0 and 0.5 than between 0.5 and 1. SRS can only guarantee the same number of samples in each equally probable stratum if the sample size is infinite. In many cases, however, this is not problematic as the sample size is large enough such that the distribution is sufficiently uniform for the given application. When a small sample size is desired, or when the quantity being estimated in a simulation is very small, enforcing a uniform distribution with LHS can be more efficient than SRS.

The user may select LHS for the epistemic and aleatory loops from the Global Settings Dashboard. When the “Use Latin Hypercube Sampling” box is checked in the Simulation Settings window, LHS is applied instead of SRS. The steps of the internal GoldSim LHS calculations are:

- Divide the interval [0,1] into equally-sized strata
- Randomly reorder the strata
- Apply SRS to sample one value from each stratum

This implementation includes two randomizing steps: (1) the reordering of the strata and (2) the random sampling within each stratum. There is an additional option, shown in Figure 3-4, to either apply SRS within the strata or to use the midpoints of the strata. If the midpoints are used, the resulting samples will be more uniform but less random.

Regarding GoldSim implementation, LHS is further applied to subdivisions of the total sample when the sample size is even. Hence, if the sample size is 100, then there are two subsets of 50 strata each, and within each subset there are 2 subsets of 25 strata. LHS is conducted this way to support convergence testing.



Define Monte Carlo options to carry out a probabilistic simulation, and specify the sampling method for Stochastic variables.

● Probabilistic Simulation

Realizations: 1

☐ Run the following Realization only: Realization: 4

☑ Use Latin Hypercube Sampling: Use random points in strata

- Use mid-points of strata
- Use random points in strata

☑ Repeat Sampling Sequences

☑ Specify Realization Weights: \Epis_Weight

○ Deterministic Simulation

Solve Simulation deterministically using:

● Element Deterministic Values

○ Element Mean Values

○ Specified Quantile: 0.5

Result Size: 2.80 KB histories, 176.8 KB final values

Figure 3-4. Menu for LHS options in Simulation Settings.

3.1.3.3 Discrete Probability Distribution (DPD) Sampling

The quantile value sampled by the stochastic elements using SRS or LHS can be further modified based on the selection of additional sampling inputs, including the use of DPD sampling and Importance Sampling. It is possible to apply DPD on one of the loops, either the epistemic or aleatory loop, or both loops.

The order in which these changes are performed within the Framework's probabilistic structure is important. Beginning with Importance Sampling and then trying to discretize would lead to the suppression of any importance measures. As a result, the Framework checks first whether DPD is requested by the user and, if so, discretizes the domain into as many strata or levels as selected by the user. The discretization is applied to all variables on the aleatory or epistemic loop where DPD is applied.

Regardless of whether Importance Sampling is used, in DPD each sampled value is positioned into an initially equiprobable stratum based on the number of strata selected by the user. For instance, if a quantile value of 0.5632 is sampled and 50 strata are considered, the stratum including this value is greater than or equal to 0.56 and less than or equal to 0.58 (step=1/50=0.02). If 20 strata are considered, the stratum including this value is greater than or equal to 0.55 and less than or equal to 0.6 (step=1/20=0.05). When DPD is selected, it is applied to all the variables in the sampling loop (i.e., all epistemic variables if it is selected for epistemic uncertainty and all aleatory variables if it is selected for aleatory uncertainty).

Once each stratum has been defined, the algorithm estimates the representative discrete value to use as the sampled value of the input, which corresponds to the conditional expected mean for the distribution truncated by the stratum boundaries. The conditional mean is calculated on the distribution and not on the quantile. Consequently, the nature of the distribution needs to be known because the conditional mean is dependent on the type of distribution and its uncertainty and is usually different from the midpoint for non-symmetric distributions.

3.1.3.4 Importance Sampling

The quantile value sampled by the stochastic elements using SRS or LHS at the beginning of the sampling process is modified based on the user's decision to apply Importance Sampling. It is possible to apply Importance Sampling on one of the loops, either the epistemic or aleatory loops, or both loops for most inputs, though it is applied by the Framework after DPD if both are used. Implementation of Importance Sampling in the Framework occurs in five parts. For clarity, the structure that accomplishes this implementation is described below according to each part, not necessarily in the order of calculation or location in the Framework.

(1). Determine which inputs are sampled using Importance Sampling. The Importance Sampling option is set in Global IDs 0103 and 0109 on the User Options worksheet for epistemic and aleatory inputs, respectively. The inputs for which Importance Sampling is applied are specified on subsequent worksheets.

(2). Specify the target probability for Importance Sampling. The target probabilities are specified by the user for each input for which Importance Sampling is selected in the Inputs Workbook.

(3). Determine the width of the Importance Sampling interval around the target probability (quantile). The region defined for Importance Sampling depends on the number of inputs for which Importance Sampling is applied. The Framework first determines the number of inputs in each loop that have been selected for Importance Sampling. It then calculates the Importance Sampling widths for epistemic and aleatory inputs using the number of inputs selected for Importance Sampling.

(4). Define the Importance Sampling quantile. To calculate the sampled values for an input that is selected for Importance Sampling, the Framework uses the user-specified Importance Sampling quantile and the calculated Importance Sampling width. The Framework samples the input such that half of the samples lie outside the region specified for Importance Sampling and the other half lie within the region. The Importance Sampling region is determined by the width of the Importance Sampling interval and the target probability.

When both Importance Sampling and DPD are used, Importance Sampling is applied after DPD. The implementation is the same as described above except that the Importance Sampling quantile is calculated using the DPD-transformed quantiles rather than the initial sampled quantiles.

(5). Calculate the Importance Sampling weight. The sampled quantile value calculated under the Importance Sampling is used to produce an Importance Sampling weight. Because Importance Sampling is used to obtain a greater number of samples within a specific region of the input space, the estimate of the probability of an event will be incorrect if the samples are used directly as if they were drawn using SRS. Weights are used to correct for the bias introduced by Importance Sampling when the final results are calculated.

3.1.3.5 Spatial Variability

Spatial variability describes inherent heterogeneity in a material or in conditions that affect crack occurrence and evolution at a specific location. A classical way to represent spatial variability is through the use of a probabilistic framework in which spatially varying inputs are represented with distributions supported by observation and measurements.

In xLPR V2.0, several inputs are required to be spatially variable to capture "within weld variability". Without such representation, cracks would occur at the same time within each segment of the weld and grow at the same rate. This leads to another important observation: spatially variable distributions are dependent on the discretization of the problem. As an example, if a weld is divided into 1,000 segments instead of 10, one would expect that the likelihood of a crack occurring in each segment will be reduced accordingly by a factor of 100.

The implementation of spatial variability requires a different sampling structure as compared to the sampling structure implemented for non-spatially varying parameters because the values are sampled several times per realization. A set of commonly considered spatially varying inputs have been identified by the experts for each module and no other inputs may be selected by the user for spatial variability. Each of the spatially varying inputs has been split into two components. The first represents the uncertainty or weld-to-weld variability and is sampled only once per realization. The second represents the within-weld variability and is sampled several times per realization generally corresponding to the number of subunits. The two variables are multiplied within the Framework to generate a different value in each subunit that captures both uncertainty and variability. This separation is necessary to allow the user to apply Importance Sampling on the spatially varying inputs. It allows Importance Sampling to be applied on the uncertainty component, which is sampled only once per realization, instead of on the spatially varying component. If Importance Sampling were applied to the spatially varying component, the sample region defined for Importance Sampling would become so small that it would no longer be representative.

The inputs for which spatial variability is included are identified in the Inputs Workbook using color-coding. A detailed description of the spatially varying inputs can be found in Appendix F of the User Manual [14]. The user can specify whether spatially varying inputs are treated as uncertain and, if so, what distributions to apply and whether the inputs are treated as epistemic or aleatory. Importance Sampling cannot be selected for the spatially varying component of the input because it is implemented on the uncertainty multiplier component instead.

3.1.3.6 Correlations

Most uncertain inputs used by the Framework are assumed to be statistically independent from one another, meaning that changing the value of one input does not impact the likelihood of another input. However, a subset of Framework inputs is statistically dependent, meaning that these inputs must be sampled such that this dependency is preserved. This dependence is accomplished using correlations. Certain pairs of parameters have been pre-determined to allow correlation. The strengths of pairwise correlations implemented in the Framework are specified by the user in the Inputs Workbook.

As previously described, uncertainty in the Framework is propagated via a sample of uniform distributions between 0 and 1 whose results are used to define quantile values for the input probability distributions. Correlation is applied to these quantile values. This is equivalent to applying a rank correlation between variables. That is, if the sampled quantile value for the first variable in a correlation is q_{x_1} and the sampled quantile value for the second variable in the correlation is q_{x_2}, then the correlation ρ is applied to the quantile values as given in Equation 3-1 below:

Equation 3-1. Calculation of sample quantiles for correlated inputs.

$$\begin{cases} q_{x_1} = q_{x_1} \\ q_{x_2} = \rho q_{x_1} + q_{x_2}\sqrt{1-\rho^2} \end{cases}$$

This quantile value is used to calculate the sampled value for the input that will be used in deterministic model calculations. Correlation is implemented in the Framework as further described in the SDD [8].

3.2 Landing Platform

The landing platform is the link between the Inputs Interface, the probabilistic structure, and the deterministic model. It assembles all the inputs and organizes them according to the modules that require them for calculations. The landing platform also collects the sampled input values generated by the probabilistic structure and groups them according to their use in subsequent calculations. The deterministic model uses these inputs as it calls the various modules.

The landing platform serves as the interface between the various modules. The modules were developed with different units for shared parameters that must be converted as inputs are passed between modules. Inputs must also be manipulated in intermediate calculation steps between modules (e.g., forces and moments converted to stresses or vice versa). The landing platform organizes and performs these functions according to the requirements of each module.

The landing platform is located within the aleatory submodel element. This location allows the landing platform to update module inputs for each aleatory realization according to the sampled values passed by the probabilistic structure. The landing platform uses meaningful names for elements that are used to graphically represent each input as opposed to using the Global ID numbers. This makes navigation of the GoldSim model file more meaningful. For example, the user can see that the Yield Strength (Global ID 2501) affects the stability calculation modules as would be expected, rather than tracing “p2501” through the Framework.

The landing platform thus serves as a central hub integrating the model inputs, sampling scheme, and deterministic model by referencing and feeding inputs to the appropriate models, providing the necessary logic for correctly assigning the sampled inputs, and establishing a structure to enable the correct flow of data throughout the model.

The specific functions implemented in the landing platform are difficult to divest from their purpose in supporting the deterministic model calculations, which highlights the interlinkage between these two components of the GoldSim model file. Details regarding these functions and calculations are thus left to the description of the deterministic model in Section 3.3.

3.3 Deterministic Model

The deterministic model is the set of calculations performed for a single realization. Deterministic calculations start with the TIFFANY and LEAPOR preprocessors described in Section 2.2. The remaining deterministic calculations are performed, requested, and managed by the GoldSim model file. The GoldSim model file performs additional preprocessing, calls the various modules as needed, and tracks the system variables throughout the simulation. The general flow of the deterministic model calculations is:

- pre-time loop calculations
- crack initiation
- time loop

Sections 3.3.1 to 3.3.3 discuss these topics, respectively. In these sections, references to specific inputs include a reference to the associated Global ID number, when applicable. Post-processing calculations are generally needed for probabilistic analyses, and they are described in Section 3.4.

3.3.1 Pre-Time Loop Calculations

Before the Framework can begin simulating crack initiation and growth, there are many calculations it must perform:

- space and time discretization
- operating conditions and time periods
- geometry calculations
- material property calculations
- circumferential spatial variability
- stress calculations
- load limits
- crack properties
- additional pre-time loop calculations for DLLs

These calculations are described in detail in the following subsections.

3.3.1.1 Space and Time Discretization

The Framework discretizes space around the circumference of the weld into equally-sized subunits. An operational constraint of xLPR V2.0 is that only one crack of each orientation type may initiate in each subunit. The number of subunits, or Nsub, is defined by the user and may range from 2 to 30 (Global ID 0002). These subunits are arranged clockwise such that Subunit 1 is located at top dead center of the weld and all the subsequent ones are numbered in order.

Each subunit spans 2*π/Nsub radians. The Framework defines a vector of length Nsub, Phi_subunit[Nsub], that defines the center of each subunit. For a specific subunit, n, within the vector, the center is calculated as follows:

$$\text{Phi\_subunit}[n] = 2*\pi*(n-1)/\text{Nsub}$$

The Framework discretizes time by a constant time step length called TimeStep. The time step is typically one month but may be changed by the user via the Global Settings Dashboard. It remains constant during a simulation. The length of the simulation is SimDuration and is in calendar time. It is the plant operating period entered by the user (Global ID 0001).

Crack initiation times and crack growth rates are calculated in effective full-power years (EFPY). The Framework converts these calculations to calendar time by dividing by the plant capacity factor, PCF. EFPY is provided by the user (Global ID 1001). The Framework calculates PCF by dividing the EFPY by SimDuration.

Discrete time periods within SimDuration are defined to accommodate different operating conditions and pre- and post-mitigation periods. These time periods are described in the next section.

3.3.1.2 Operating Conditions and Time Periods

The Framework initially defines up to three operating periods (Global IDs 0402 and 0405). These periods correspond with constant operating conditions. The operating conditions that remain constant during these periods are:

- operating temperature (Global IDs 3102, 3202, and 3302)
- operating pressure (Global IDs 3101, 3201, and 3301)
- dissolved oxygen (Global IDs 3103, 3203, and 3303)
- deadweight and thermal expansion loads and stresses (Global IDs 4101 through 4324)

Mitigations (Global IDs 0301 through 0306) may be defined for any time during the simulation and may occur within any part of an operating period. For physical mitigation, changes in normal operating stresses occur, new materials may be involved in the cases of inlay and overlay, and the weld thickness may change if there is an overlay. For chemical mitigation, the concentrations of zinc (Global IDs 3003 and 3005), hydrogen (Global IDs 3002 and 3004), or both may change.

Many of the physical processes modeled by xLPR V2.0 are simulated in the time loop, which is described in Section 3.3.3. For these processes, changes to conditions due to mitigations are accommodated by calculating the new conditions after the mitigation occurs. The Framework does this by including conditional statements in elements based on the modeled elapsed time at that point in the simulation, ETime. Elements that provide temporally variable conditions at ETime are referred to as dynamic elements and are typically indicated by a "_dt" extension in their names.

Crack initiation calculations are performed prior to the time loop. Consequently, the crack initiation modules require further division of the time periods when mitigations are applied so that the Framework provides all discrete changes in important crack initiation conditions over time to these modules. The discrete time periods for the crack initiation modules are called MTS periods. The number and duration of each MTS period is determined by the Framework based on the number of operating periods, type of mitigation, and mitigation timing. Once the number and duration of each MTS period are determined, the Framework determines the stresses for each MTS period as described in Section 3.3.2.1.

For fatigue crack initiation, the Framework defines a maximum of four MTS periods: three normal operating periods plus an additional period should physical mitigation divide one of the normal operating periods. For PWSCC crack initiation, the Framework defines a maximum of five MTS periods: three normal operating periods plus a potential division due to physical mitigation and another potential division due to zinc mitigation.

An additional type of time period is the TIFFANY time period. When fatigue is considered, the entire simulation has either one or two TIFFANY time periods. Two TIFFANY time periods are needed when an inlay or overlay occurs.

3.3.1.3 Geometry Calculations

The Framework performs geometry calculations using the following user inputs:

- pipe outer diameter, Outer_Diameter (Global ID 1101)
- pipe wall thickness, Pipe_Thickness (Global ID 1102)
- weld thickness, Weld_Thickness (Global ID 1104)
- weld overlay thickness, WOL_Thickness (Global ID 1105)
- weld inlay thickness, Inlay_Thickness (Global ID 1106)

Using the constant or sampled values for these inputs, the Framework calculates the following standard parameters as follows:

- pipe outer radius, Outer_Radius = Outer_Diameter/2
- pipe inner diameter, Inner_Diameter = Outer_Diameter – 2*Pipe_Thickness
- pipe inner radius, Inner_Radius = Inner_Diameter/2
- pipe average radius, Average_Radius = 0.5*(Inner_Radius + Outer_Radius)
- pipe cross-sectional area, Cross_sectional_area = π*(Outer_Radius^2 - Inner_Radius^2)
- maximum circumferential crack inner half-length, Max_iHL = π*Inner_Radius
- maximum circumferential crack outer half-length, Max_oHL = π*Outer_Radius
- pipe moment of inertia, Moment_of_Inertia = π*(Outer_Radius^4 - Inner_Radius^4)/4

With overlay mitigation, the Framework recalculates several of these parameters as follows:

- pipe wall thickness with weld overlay, Pipe_Thickness_WOL = Pipe_Thickness + WOL_Thickness
- pipe outer diameter with weld overlay, Outer_Diameter_WOL = Outer_Diameter + 2*WOL_Thickness
- pipe outer radius with weld overlay, Outer_Radius_WOL = Outer_Diameter_WOL/2
- pipe average radius with weld overlay, Average_Radius_WOL = 0.5*(Inner_Radius + Outer_Radius_WOL)
- pipe cross-sectional area with weld overlay, Cross_sectional_area_WOL = π*(Outer_Radius_WOL^2 - Inner_Radius^2)
- maximum circumferential crack outer half-length with weld overlay, Max_oHL_WOL = π*Outer_Radius_WOL
- Pipe moment of inertia with weld overlay, Moment_of_Inertia_WOL = π*(Outer_Radius_WOL^4 - Inner_Radius^4)/4

The Framework tracks these parameters during the time loop by including conditional logic based on ETime. For example, the pipe wall thickness at ETime, Pipe_Thickness_dt, is calculated as follows:

> Pipe_Thickness_dt = Pipe_Thickness (if before overlay) or Pipe_Thickness + Overlay_Thickness (if after overlay)

Several geometry parameters that may change during a simulation are needed for the crack initiation modules prior to execution of the time loop. These parameters are vectors of length MTS and include:

- Pipe_Thickness_MTS[MTS]
- Inner_Radius_MTS[MTS]
- Outer_Radius_MTS[MTS]
- Moment_of_Inertia_MTS[MTS]
- Cross_sectional_area_MTS[MTS]

For these parameters, the Framework prepares separate vectors by MTS period for fatigue crack initiation and PWSCC initiation as needed. The Framework also performs geometry calculations for cracks. Those calculations are described in Section 3.3.1.8.

3.3.1.4 Material Property Calculations

Important material property inputs include those for strength and toughness. They include:

- yield strength, yield_strength (Global IDs 2101, 2301, 2501, and 2701)
- ultimate strength, ultimate_strength (Global IDs 2102, 2302, 2502, and 2702)
- elastic modulus, elasticity (Global IDs 2105, 2305, 2505, and 2705)
- material initiation J-resistance Jic parameter, Resist_Jic (Global IDs 2106, 2306, 2506, and 2706)
- material initiation J-resistance coefficient, Resist_C (Global IDs 2107, 2307, 2507, and 2707)
- material initiation J-resistance exponent, Resist_m (Global IDs 2108, 2308, 2508, and 2708)

These inputs are placed into 4-element vectors where the given material property is contained in element 1 for Left Pipe, element 2 for Right Pipe, element 3 for Weld, and element 4 for the Mitigation material.

xLPR V2.0 uses the Ramberg-Osgood model [18] to describe the relationship between stress and strain. The Framework calculates the Ramberg-Osgood parameters based on the material strength properties for Left Pipe and Right Pipe. It performs the calculations in separate script elements for Left Pipe and Right Pipe after first ensuring that the yield strength does not exceed the ultimate strength. The Ramberg-Osgood parameters are calculated as follows [7]:

- Ramberg-Osgood alpha parameter, RO_alpha = 0.002*elasticity/yield_strength
- Ramberg-Osgood epsilon naught parameter, RO_eps0 = yield_strength/elasticity
- Ramberg-Osgood hardening exponent, RO_n = exp((1.05487525 MPa + 0.003054*yield_strength - 0.000572243*ultimate_strength) *1/1MPa)

A warning is issued if the Ramberg-Osgood hardening exponent is outside the range of 2.5 to 9.0.

The Framework calculates mixed strength material properties for Left Pipe and Right Pipe as a function of the user-entered dissimilar metal mixture ratio, DM_mixture_ratio (Global ID 0204). For yield strength and ultimate strength, the calculations are as follows:

- mixed material yield strength, Mixed_yield_strength = DM_mixture_ratio*yield_strength[1] + (1 - DM_mixture_ratio)*yield_strength[2]
- mixed material ultimate strength, Mixed_ultimate_strength = DM_mixture_ratio*ultimate_strength[1] + (1 - DM_mixture_ratio)*ultimate_strength[2]

Here, the brackets note which element of the material property vector is used in the calculation. Mixed-strength Ramberg-Osgood parameters are calculated in a similar manner.

3.3.1.5 Circumferential Spatial Variability

Circumferential spatial variability is the variation of properties across the individual subsections of the weld. The deterministic model includes this variability by populating spatial distributions for each realization. Spatial variability may be defined by the user for most spatially varying inputs, including:

- initial lengths and depths of circumferential and axial cracks (Global IDs 1201, 1205, 1210, and 1215, 1203, 1207, 1212, and 1217)
- fatigue crack initiation parameters (Global IDs 2525, 2528, 2725, and 2728)
- PWSCC initiation proportionality constants (Global IDs 2542, 2546, 2742, and 2746)
- fatigue crack growth scaling and environmental factors (Global IDs 2160, 2163, 2165, 2167, 2170, 2174, 2181, 2360, 2363, 2365, 2367, 2370, 2374, 2381, 2560, 2563, 2565, 2567, 2570, 2574, 2581, 2760, 2763, 2765, 2767, 2770, 2774, and 2781)
- PWSCC growth within-component variability factors (Global IDs 2193, 2393, 2593, and 2793)

Due to spatial variability, these parameters are implemented within the Framework as vectors of dimension Nsub.

Spatial variability for leak rate is built into the Framework as specified in the Leak Rate Module SDD [19]. Spatial variability is implemented using a multiplier on the standard deviation of the leak rate sampled for each leaking crack for the realization. The multiplier has a fixed normal distribution with a mean of 0 and standard deviation of 1. Leak rate uncertainty is addressed in Section 3.3.3.11.

3.3.1.6 Stress Calculations

The Framework performs several stress calculations prior to and during the time loop. Each of these calculations is described in detail in the xLPR Stress Requirements technical report [12]. Footnotes in that report explain how the stress calculations are implemented within the Framework. A general overview of the stress calculations is presented here. Stresses implemented in the Framework include:

- weld residual stresses
- normal thermal membrane and bending stresses
- dead weight membrane and bending stresses
- additional stresses at the inner diameter of the pipe due to grinding or other surface treatments (Global IDs 4005 and 4006)
- membrane stress due to internal pressure
- crack face pressure
- seismic stresses (Global IDs 4002, 4003, and 4004)
- stresses due to Type I, Type I and II, and Type III transients

Many of these stresses are constant for the entire simulation. Some change discretely due to transients or changes in conditions, and some change continuously as a function of crack growth. This section addresses those stress calculations performed prior to the time loop.

Because the Framework performs all crack initiation calculations prior to executing the time loop, the crack initiation modules only receive the pre-processed stress calculations. Each of the stresses listed above is included, except for seismic stresses. Fatigue stresses are included for fatigue crack initiation by adding the TIFFANY outputs of minimum and maximum transient change in stress to the total stresses calculated for PWSCC initiation.

The PWSCC initiation stresses are calculated for each subunit, as described in Section 3.3.1.1, and for each MTS period, as described in Section 3.3.1.2. The Framework generates a matrix of stress calculations of dimensions Nsub by MTS for each crack initiation module.

For fatigue crack initiation, the Framework calculates the maximum and minimum circumferential and axial initiation stresses for each fatigue MTS period, denoted MTSf, by adding the maximum and minimum transient stresses to the normal operating stresses. The normal operating stresses are calculated exactly as they are for PWSCC, except that they have a matrix of dimensions Nsub by Ntrans and there is one maximum and one minimum stress matrix for each MTSf period. Ntrans is the number of transients, up to 20. The Framework uses the TIFFANY look-up table outputs to calculate the maximum and minimum transient stresses.

For modules within the time loop that require dynamically updated stress inputs, the Framework calculates the stresses based on updated conditions and crack properties prior to calling these modules. Examples of these calculations are provided in Section 3.3.3.

3.3.1.7 Load Limits

The Framework calculates load limits to test whether the piping system can sustain normal operating loads, normal operating loads plus maximum transient loads, and normal operating loads plus seismic loads. Prior to executing the time loop, only normal operating loads and normal operating loads plus seismic loads are evaluated. If the normal operating loads exceed the limits of the piping system, an error is issued, and the realization is stopped. If normal operating loads plus seismic loads exceed the limits of the piping system, a warning is issued, and a seismic instability is recorded.

Load limit calculations are also performed at the beginning of each time step. Here, the total loads, which include transient loads but exclude seismic loads, are calculated and evaluated against the piping system limits in each time step immediately prior to executing single TWC stability calculations. Those calculations are described further in Section 3.3.3.9.5.

3.3.1.8 Crack Properties

The Framework tracks certain crack properties during the time loop. These properties are used to characterize each potential circumferential and axial crack in the simulation. Because only one crack of each orientation type can occur in each subunit, the crack properties are tracked in vectors of dimension Nsub. The Framework uses integrator state variable elements to track these vectors because the values may be updated by discrete changes and triggers.

One of the primary crack property vectors, Crack_type, tracks the crack type. Here a “0” value indicates no crack. A “-1” value indicates an SC. This is the starting status when a crack initiates and represents a crack on the inner surface of the weld with a depth less than the weld thickness. A “-2” value indicates a TRC. This is a crack that extends through the weld thickness, but whose lengths on the inner and outer surfaces are different. This status serves as a transition between an SC and a TWC. A “-3” value indicates a TWC. This is a crack through the weld thickness with identical lengths (in radians) on the inner and outer surfaces. A positive value from 1 to n (the number of pipe subunits) indicates that the circumferential crack coalesced with another circumferential crack. The history of the subsumed crack is not tracked further because the subsuming crack maintains this information. For example, if a crack in subunit “2” is subsumed by a crack in subunit “3”, the crack type for the first crack changes from “-1” to “3” while the second crack remains type “-1”. A value of “200” indicates a rupture occurred during the realization.

The other primary crack property vectors track position and dimensions. The vector, Position, is used to track the crack position. It is a value in radians and varies between 0 and 2π. This is the position of the crack center, where 0 is located at top dead center of the pipe. For a circumferential crack, the radial position may change with coalescence but the axial position is constant. For an axial crack, the radial position does not change; however, the axial center position may move in the case of dissimilar materials. All cracks are assumed to initiate at the axial center of the weld width. A different vector, Crack_depth, tracks the non-dimensional crack depth. Here, the depth of the crack measured radially outward from the inside surface of the pipe is normalized by the thickness of the pipe plus the thickness of an overlay, if present. The crack depth is only tracked for SCs. In the case of TRCs and TWCs, the crack depth is set to 1 and does not change thereafter. Another vector, iHL, tracks the non-dimensional crack inner half-length. Here, half the length of a crack as measured on the inside of the pipe is normalized by π for circumferential cracks or by the half-width of the weld for axial cracks. The value can be greater than 1 for axial cracks that grow into the adjacent base materials by fatigue crack growth or by PWSCC growth when the user activates this option (Global ID 0207). A similar vector, oHL, separately tracks the non-dimensional crack outer half-length.

To distinguish between the crack orientations, the Framework uses “_ac” and “_cc” extensions in the vector names to represent axial and circumferential crack properties, respectively. Initially, the Framework defines the crack property vectors such that the subunit number corresponds directly with the vector element number. However, once initiation times are determined for each subunit, the Framework reorders the values in the vectors according to the order of initiation time as described in Section 3.3.2.3.

In addition to the primary crack properties, the Framework tracks other parameters related to cracks. These parameters include:

- number of cracks, Ncracks, which is the number of active cracks in the current time step
- number of initial flaws, Nflaws, which is the number of pre-existing cracks prescribed by the user when using the initial flaw density option (Global ID 0501)
- number of cracks initiated, Ninit
- non-dimensional axial deviation, total_axial_deviation, which is the distance between the center of an axial crack and the axial center of the weld normalized by the weld width
- dimensional counterparts to several of the non-dimensional crack properties, which are indicated by the extension “_m” for meters or “_rad” for radians (e.g., Crack_depth_m[Nsub], iHL_m[Nsub], iHL_rad[Nsub], oHL_m[Nsub], and oHL_rad[Nsub])
- crack areas, crack_area[Nsub]

- fraction of cross-sectional area cracked due to all circumferential cracks, f_cracked_cc
- occurrence of crack, Occurrence_crack, which is true when one or more cracks exist
- occurrence of leak, Occurrence_leak, which is true if one or more cracks are leaking
- occurrence of rupture, Occurrence_rupture, which is true if a rupture occurs

3.3.1.9 Additional Pre-Time Loop Calculations

The Framework performs several additional calculations prior to executing the time loop.

For the PWSCC initiation module, the Framework converts the operating temperature from Celsius to Kelvin. It also converts yield strength, ultimate strength, and elastic modulus properties for weld and mitigation materials (Global IDs 2501, 2502, 2505, 2701, 2702, and 2705) from operating to room temperature based on the specified conversion factors (Global IDs 2604, 2605, 2606, 2804, 2805, and 2806).

For the fatigue crack initiation module, the Framework calculates the number of cycles per year for each transient.

For the crack growth rate module, the Framework sets the power law constant to zero if the user selects the option to not allow PWSCC growth into the base materials in the case of axial cracks (Global ID 0207).

For the leak rate calculations, the Framework checks whether the operating pressure is outside the bounds of the LEAPOR look-up tables. If so, the Framework issues a warning and continues the simulation. The Framework also uses a stochastic input element to introduce an uncertainty factor into the leak rate calculations. The uncertainty factor is sampled from a normal distribution with a mean of 0 and a standard deviation of 1. A different uncertainty is sampled for axial and circumferential cracks. The uncertainty factors are applied within the time loop as discussed in Section 3.3.3.11.

For the multiple surface crack stability (MSC) calculations, the Framework determines the MSC spatial step size as follows:

MSC_stepsize = 2*π*Outside_Radius_dt/10000

3.3.2 Crack Initiation

The timing and locations of cracks initiating during the plant operating period are determined prior to the Framework time loop. For PWSCC and fatigue crack initiation, the timing and locations are determined by calling the corresponding crack initiation modules as described in Section 3.3.2.1. For the initial flaw density option, the number of pre-existing cracks is prescribed by the user and the locations are determined by the Framework, except for the first crack which is always located at top dead center as described in Section 3.3.2.2.

Crack initiation is determined such that no more than one axial and one circumferential crack may occur in each subunit. The Framework calls the crack initiation modules independently for each combination of initiation process and crack orientation.

Once the timing and locations of cracks are determined, the Framework prepares the crack property vectors discussed in Section 3.3.1.8 by placing the cracks in the first elements of the vectors in the order of initiation. This placement minimizes the number of rows included in the processing loops and expedites the Framework calculations.

Sections 3.3.2.1, 3.3.2.2, and 3.3.2.3 describe how the Framework manages and performs the various crack initiation calculations.

3.3.2.1 PWSCC and Fatigue Crack Initiation

When PWSCC initiation, fatigue crack initiation, or both is selected (Global ID 0501), the Framework calls the appropriate DLLs with the input parameters as determined in the Landing Platform. Inputs to these modules include stresses, loads, material properties, operating temperature, and material-specific initiation model parameters. Several of these inputs are provided in vectors or matrices discretized either by subunit as described in Section 3.3.1.1, by MTS period as described in Section 3.3.1.2, or both. Discretization by subunit is needed to account for spatial variability in parameter inputs and for differences in stresses around the circumference. Discretization by MTS period is needed to account for mitigations and changes to the operating conditions over the plant operating time. The full sets of inputs for the crack initiation modules are described in the PWSCC Crack Initiation SDD [21] and the Fatigue Crack Initiation SDD [22].

The deterministic outputs returned by the crack initiation modules are two vectors, one for initiation time, TinitEFPY_dr[Nsub], and the other for location, LocInit_dr[Nsub]. The elements in these vectors are ordered by aximuthal position as described in Section 3.3.1.1. Crack initiation time is returned in EFPY and the Framework converts it to calendar time by dividing by the PCF as described in Section 3.3.1.1.

When both PWSCC and fatigue crack initiation processes are selected for a simulation, the Framework calls each module independently before processing any potential conflicts. Conflicts arise when both PWSCC and fatigue crack initiations are predicted within the plant operating period for the same crack orientation in the same subunit. When this happens, the Framework selects the earlier of the two cracks. The Framework records the initiation process, either PWSCC or fatigue, associated with each crack in the vector, Mech_dr[Nsub].

The final task performed by the Framework for crack initiation prior to execution of the time loop is to reorder the initiation times and locations. This task is described further in Section 3.3.2.3.

Additional crack initiation calculations must be performed by the Framework when the simulation involves an inlay because an inlay creates a new surface for initiation. The Framework calculates crack initiation for inlay in parallel with the initiation calculations for the original, pre-mitigated weld. Thus, prior to executing the time loop, two complete sets of crack initiation times and locations are available, one for the weld material and one for the inlay material. For the inlay, the Framework calls the PWSCC and fatigue crack initiation modules, as applicable, just like it does for the weld material with two exceptions: (1) it uses the material-specific inputs from the Mitigation worksheet, and (2) it uses a conditional statement to set the stress inputs to zero for the pre-inlay MTS periods. This approach ensures a pristine inlay when inlay mitigation occurs. Exactly as it handles crack initiation in the weld, the Framework converts initiation times to calendar times and resolves conflicts when both PWSCC and fatigue crack initiations occur in the same subunit. The Framework also reorders the outputs by initiation time as described in Section 3.3.2.3 and stores them in memory for later use in the time loop as described in Section 3.3.3.2.

3.3.2.2 Initial Flaw Density

The initial flaw density option (Global ID 0501) allows the user to specify one or more pre-existing cracks at the start of a simulation. The user specifies the number of initial circumferential and axial cracks in Global IDs 1209 and 1214, respectively. No additional cracks occur during the simulation. Regardless of the number of cracks specified, the Framework always locates the first at top dead center of the weld, which is defined as 0 radians. The remaining cracks are randomly located at other subunits such that no more than one crack of a given orientation occurs in a given subunit. The Framework orders cracks in ascending azimuthal position in the two vectors, LocInit and Tinit. LocInit stores each crack's initial position and Tinit stores the initiation time. These vectors are used in the time loop as described in Section 3.3.3.2.

3.3.2.3 Reordering Crack Initiation Vectors

In the time loop, the Framework performs numerous crack-related calculations for each realization. To reduce computational time, cracks in the crack property vectors described in Section 3.3.1.8 are ordered by initiation time so that the Framework need only loop through rows of existing cracks. For the initial flaw density option, reordering is more straightforward as described in Section 3.3.2.2.

For PWSCC and fatigue crack initiation, the Framework reorders the initiation times, locations, and initiating mechanisms by initiation time. For circumferential cracks, it also maps the reordered values to the corresponding subunits using the vector, IndexTrans[Nsub], which is needed by the coalescence module. For initial flaw density, the values for IndexTrans[NSub] are set to the row number because by design the initial flaw vectors are ordered by both time and position.

3.3.3 Time Loop

When the calculations described in Sections 3.3.1 and 3.3.2 are complete, the Framework executes the time loop. The general flow of operations during the time loop is:

- update system properties and conditions
- initiate any new cracks
- call stress intensity factor DLLs
- update transient stress inputs
- call crack growth DLL
- update crack type
- call crack coalescence DLL
- call crack transition DLL
- call crack stability DLLs
- call COD DLLs
- calculate leak rate using look-up tables
- call ISI DLL

Sections 3.3.3.1 through 3.3.3.12 describe these operations in detail.

3.3.3.1 Update System Properties and Conditions

At the beginning of each time step, the Framework updates the following system properties and conditions as needed. The first task is to update the general, time-dependent system properties and conditions. More specifically, the Framework:

- adjusts the operating temperature, pressure, and dissolved oxygen as needed when a new operating period begins
- changes the hydrogen concentration to the mitigation concentration when the elapsed time exceeds the time of hydrogen mitigation
- updates parameters to account for material thicknesses that may change upon physical mitigation
- uses new axial and hoop WRS profiles if a physical mitigation occurs
- updates crack properties if a weld overlay occurs to include new crack types and new crack depths and to set the outer crack half-lengths to zero
- updates crack properties if an inlay occurs to include setting all the crack properties and number of cracks to zero
- performs new calculations for each transient to include the time between events, number of events, and number of cycles

Stresses that are independent of crack properties are recalculated at the beginning of each time step regardless of the operation of active time transient. As described in [12], the calculations include:

- hoop WRS
- crack face pressure
- PWSCC circumferential SC and TWC growth stresses
- total maximum hoop stress
- SC and TWC stability hoop stresses
- certain membrane and bending stresses

3.3.3.2 Initiate New Cracks

The first crack process that the Framework simulates in a time step is initiating new cracks depending on the crack initiation times calculated as discussed in Section 3.3.2. For each crack orientation, there is a vector of initiation times for the weld and another for the inlay, if simulated. If inlay is simulated and the elapsed time exceeds or equals the physical mitigation time (Global ID 0306), the Framework refers to the vector of initiation times for the inlay.

The initial crack depth depends on several factors as stated in the crack initiation SDDs [21], [22]. The Framework sets the nondimensional depth to 0.95 if the initial depth (Global IDs 1203, 1207, 1212, or 1217, depending on the mechanism) results in a value greater than 0.95. The Framework also issues a warning to the user.

The Framework calculates the initial non-dimensional crack half-length as directed in the crack initiation module SRDs [23], [24]. If the initial depth is greater than the half-length, the initial half-length is set equal to the initial depth. Also, if the initial non-dimensional half-length exceeds 1, then it is set to 1.

3.3.3.3 Stress Intensity Factors

The first DLLs called by the Framework in the time loop are for the stress intensity factor solution modules. For circumferential cracks, several of the stress inputs to these DLLs are a function of the crack properties, which may change due to crack property updating at the beginning of the time step. To ensure that these inputs are current, the Framework recalculates them immediately prior to calling the stress intensity factor solution DLLs. As described in [12], these stresses are:

- membrane stresses for PWSCC growth of circumferential SCs and TWCs
- bending stresses for PWSCC growth of circumferential SCs and TWCs

With these inputs, the Framework calls the part-wall and through-wall stress intensity factor calculation DLLs as needed depending on the types of cracks that exist at this point in the time step. These DLLs were compiled with a wrapper code that allows a single call per time step for all cracks such that the entire crack property vectors of dimension [Nsub] are entered for each call and the outputs generated are of the same dimension.

The normal operating stress intensity factor values for SCs are determined by the KPW calculation DLL. However, the values for TWCs require further calculation [12]. The Framework calculates the normal operating stress intensity factor values for TRCs using correction factors from the crack transition module determined at the previous time step as described in Section 3.3.3.8. These correction factors reduce the effective stress intensity factor values. For TWCs, the Framework sets the correction factors to one so that the stress intensities are not affected.

3.3.3.4 Transient Stress Intensity Factors and Other TIFFANY Outputs

When fatigue is modeled for crack initiation and growth, transient stress intensity factors are calculated by the TIFFANY preprocessor. Because inlay and overlay cannot occur in the same simulation, a maximum of two TIFFANY periods are simulated, one for pre-mitigation and one for post-inlay or post-overlay.

As explained in Section 2.2.1, the TIFFANY pre-processor produces a set of text files for each TIFFANY period. The Framework reads the outputs in these text files and places them in look-up table elements in the Landing Platform. The look-up table elements in the GoldSim model file are denoted with a "_LUT" extension to their name. For example, the look-up table for the maximum through-wall membrane stress intensity factor is Kmax_TWmem_LUT. The look-up table is three-dimensional and consists of the inner half-length, c, the subunit number of the crack, crackseg, and the transient, trans. The Framework interpolates a single value from the look-up table using the function Kmax_TWmem_LUT(c, crackseg, trans).

3.3.3.5 Crack Growth

Before the Framework can calculate all the various parameters needed by the crack growth module, it must determine the materials where the tips of the cracks are located so that the correct material flags can be provided. Materials 1, 2, 3, and 4 are for Left Pipe, Right Pipe, Weld, and

Mitigation materials, respectively. For axial cracks, the tips are assumed to reach the base metals at the same time.

The Framework also calculates the precise locations of the crack tips in terms of radians for circumferential cracks and a non-dimensional distance for axial cracks. Precise locations are needed for circumferential cracks for interpolating stress intensity factors from TIFFANY look-up tables and for averaging the within-component variability. The Framework uses these positions to determine the subunit of the crack tip.

For circumferential PWSCC growth, the Framework accounts for within-component variability using a factor, fflaw (Global IDs 2193, 2393, 2593, and 2793), as described in Section 3.3.1.5. The value of this factor depends on the material and crack tip position. The Framework determines this factor for the crack growth DLL using the left and right crack tip subunit numbers as determined from the crack tip positions.

For fatigue crack growth, the Framework accounts for spatial variability in a number of crack growth DLL inputs (Global IDs 2160, 2163, 2165, 2167, 2170, 2174, 2381, 2360, 2363, 2365, 2367, 2370, 2374, 2381, 2560, 2563, 2565, 2567, 2570, 2574, 2581, 2760, 2763, 2765, 2767, 2770, 2774, and 2781). As in the case of PWSCC growth, the Framework uses the calculated left and right tip subunit numbers to ensure that appropriate values are used.

The total stress intensity factors used as input to the crack growth DLL are combinations of both transient and normal operating values. When called, the crack growth DLL loops through all cracks and transients. As described in Section 3.3.2.3, the cracks are ordered by initiation time so that the DLL only needs to loop through rows in which cracks have initiated. Transients are not reordered and may be sporadically active within a set of time steps.

The deterministic outputs of the crack growth DLL are:

- crack depth growth rate
- crack length growth rate at the inner diameter
- crack length growth rate at the outer diameter
- rate of change of the center of an axial crack

These growth rate outputs are in meters per EFPY. The Framework converts them to dimensionless growth rates in calendar time by multiplying by the plant capacity factor and dividing by the reference length. The resulting dimensionless growth rates are then multiplied by the time step length to calculate the dimensionless growth during the time step for each crack depth and half-length. If the user elects to not allow PWSCC growth of axial cracks in the base metals (Global ID 0207), the Framework calculates corrections to the axial crack growth rates.

3.3.3.6 Crack Type Updating

With growth, an SC may become a TRC and a TRC may become a TWC. Immediately after the crack depths and half-lengths are updated based on crack growth, the Framework checks whether the crack types have changed and, if so, updates them too. For circumferential cracks, the Framework resets the non-dimensional half-lengths to one if they become greater than one. For circumferential cracks that become TRCs, the Framework sets the outer half-length to 25 percent of the inner half-length or to the ratio of the pipe thickness to the outer radius, whichever is smaller, as specified in the Crack Transition Module SRD [20]. For axial cracks that become

TRCs, the Framework sets the outer half-length to 25 percent of the inner half-length or the pipe thickness, whichever is smaller.

3.3.3.7 Crack Coalescence

At this point in the time step, if there are multiple circumferential cracks, the Framework calls the coalescence module DLL to determine whether they coalesce. The various inputs required by the coalescence DLL are documented in the Coalescence SDD [25]. They include crack type, crack depth, crack position, inner and outer crack half-lengths, inner pipe diameter, pipe thickness, IndexTrans[Nsub] (Section 3.3.2.3), and selected user options for the coalescence calculations such as the minimum distance between two cracks before coalescence occurs. The Framework gathers the current values for these inputs and provides them to the module.

The deterministic outputs of the coalescence DLL are the following vectors with dimensions equal to the number of subunits:

- crack type
- crack position
- crack inner half-length
- crack depth
- crack outer half-length

If no coalescence occurs, these outputs are the same as the inputs to the module and the Framework makes no changes to the crack property vectors. However, if the Framework detects that the coalescence DLL changed any of the crack types, it replaces all crack property vector values with the values returned by the coalescence DLL.

The coalescence DLL creates TRCs from coalescing TRCs, an SC coalescing with a TRC, and a TRC coalescing with a TWC. It also combines the outer half-lengths of coalescing TRCs and TWCs. However, the DLL does not enforce the Crack Transition Module SRD requirement RTC-16 concerning the threshold inner to outer half-length ratio of 1.05 [20]. Instead, the Framework implements this requirement immediately after calling the coalescence DLL. If this ratio is below the threshold value, the Framework turns the TRC into a TWC.

3.3.3.8 Crack Transition

If there is at least one TRC, the Framework calls the crack transition DLL. The crack transition DLL determines correction factors for stress intensity factors and correction factors for CODs of TRCs and TWCs. For TWCs, the DLL wrapper sets the correction factors to one so that no correction is made.

The Framework calls the crack transition DLL with the most current crack properties and other inputs. The deterministic outputs of the crack transition DLL are the following correction factors:

- bending stress correction factor for inner diameter, G1i_bending_cc[Nsub]
- bending stress correction factor for outer diameter, G1o_bending_cc[Nsub]
- tension correction factor for inner diameter, G2i_tension_cc[Nsub]
- tension correction factor for outer diameter, G2o_tension_cc[Nsub]
- bending stress COD correction factor for inner diameter, H1i_bending_cc[Nsub]
- bending stress COD correction factor for outer diameter, H1o_bending_cc[Nsub]
- tension COD correction factor for inner diameter, H2i_tension_cc[Nsub]

- tension COD correction factor for outer diameter, H2o_tension_cc[Nsub]
- hoop stress correction factor for inner diameter, G1i_hoop_ac[Nsub]
- hoop stress correction factor for outer diameter, G1o_hoop_ac[Nsub]
- hoop stress COD correction factor for inner diameter, H1i_hoop_ac[Nsub]
- hoop stress COD correction factor for outer diameter, H1o_hoop_ac[Nsub]

The correction factors that start with H are used in the current time step when calculating the COD for determining the leak rate as described in Section 3.3.3.10. Those that start with G are used in the subsequent time step when calculating stress intensity factors as described in Section 3.3.3.3.

3.3.3.9 Crack Stability

The Framework begins to call the crack stability DLLs in the first time-step that a crack is initiated. For circumferential cracks, these DLLs are the SC stability DLLs for single and multiple surface cracks (MSCs) and the TWC stability DLL for TWCs. For axial cracks, they are the SC stability DLL for single SCs and the TWC stability DLL for TWCs.

Sections 3.3.3.9.1 and 3.3.3.9.2 discuss the crack stability workflow and calculation of the relevant loads and stresses, respectively. Sections 3.3.3.9.3 through 3.3.3.9.5 discuss the individual DLLs. Section 3.3.3.9.6 describes the Framework's stability call reduction feature.

3.3.3.9.1 Stability Work Flow

The logic and workflow for calling the stability DLLs is shown in Figure 3-5. Each box in the flow diagram containing a DLL is either bypassed or visited no more than once per time step when one or more cracks are present. After a crack occurs, calls to the stability module DLLs may be bypassed by the Framework's call reduction coding and user settings as described in Section 3.3.3.9.6.

Prior to calling the stability module DLLs, the Framework evaluates at each time step whether the loads applied during the time step exceed the strength of the un-cracked weld. If the normal operating loads plus the seismic loads exceed the strength of the weld, the Framework records a seismic instability as described in Section 3.3.1.7. If the normal operating loads plus the maximum transient load (i.e., non-seismic conditions) exceed the strength of the weld, a rupture is recorded and the Framework ends the simulation as described in Section 3.3.3.9.2.

The remainder of the workflow evaluates the stability of the cracked weld. These evaluations involve the following DLLs:

- SC stability – Failure in the case of an SC results in an SC becoming a TWC (i.e., the crack type is set to -3, the normalized crack depth is set to 1, and the outer diameter crack length is set equal to the current inner diameter crack length).

- TWC stability – Failure in the case of a TWC or a TRC results in a pipe failure.

- MSC stability – Failure in the case of either a single SC or the full set of existing cracks, regardless of crack type, results in a pipe failure.

The general workflow for circumferential cracks is complicated by the need to evaluate stability under both seismic and non-seismic conditions. Because loads under seismic conditions are greater than those under non-seismic conditions, the stability module DLLs are initially called using the loads under seismic conditions. If there is no failure under seismic conditions, the Framework does not repeat the call with loads under non-seismic conditions. If there is a failure under seismic conditions, the Framework notes the timing of the failure (seismic instability) and calls the DLL a second time but uses loads under non-seismic conditions. For time steps following a seismic instability, the Framework will only call the stability module DLLs using non-seismic loads. Seismic conditions are not evaluated for axial cracks.

The call order of the stability module DLLs for circumferential cracks is: MSC stability, SC stability, and then TWC stability. For axial cracks, it is: SC stability and then TWC stability. The MSC stability DLL does not apply to axial cracks.

In the first stage of the workflow for stability regarding circumferential cracks, the Framework calls the MSC stability DLL if there are any circumferential SCs or if there are multiple circumferential cracks regardless of crack type. If failure is determined to occur by the MSC stability DLL under seismic conditions, a seismic MSC instability is recorded. If failure is determined to occur by the MSC stability DLL under normal operating loads plus the maximum transient load (non-seismic conditions), a rupture is recorded, and the Framework ends the realization.

In the next stage of the workflow, which is the first stage for axial cracks, the Framework calls the single SC stability DLL if there are any SCs. For each circumferential SC, seismic instability is recorded if failure is determined to occur by the single SC stability DLL under seismic conditions. If a seismic instability is recorded for a circumferential SC, the Framework also calls the single TWC stability DLL for seismic conditions, treating the seismically unstable SC as a TWC to determine if there is a seismic rupture. For each circumferential or axial SC, if failure is determined to occur by the single SC stability DLL under normal operating loads plus the maximum transient load (non-seismic conditions), the SC is converted directly into an idealized TWC, bypassing the crack transition process.

In the last stage of the workflow, the Framework calls the single TWC stability DLL if there is at least one TRC or TWC, including any new TWCs created by an unstable SC as described in the previous stage of the workflow. Seismic instability of a circumferential TRC or TWC is recorded if failure is determined to occur by the single TWC stability DLL under seismic conditions. If failure is determined to occur by the single TWC stability DLL under normal operating loads plus the maximum transient load (non-seismic conditions), a rupture is recorded and the Framework ends the realization.

,

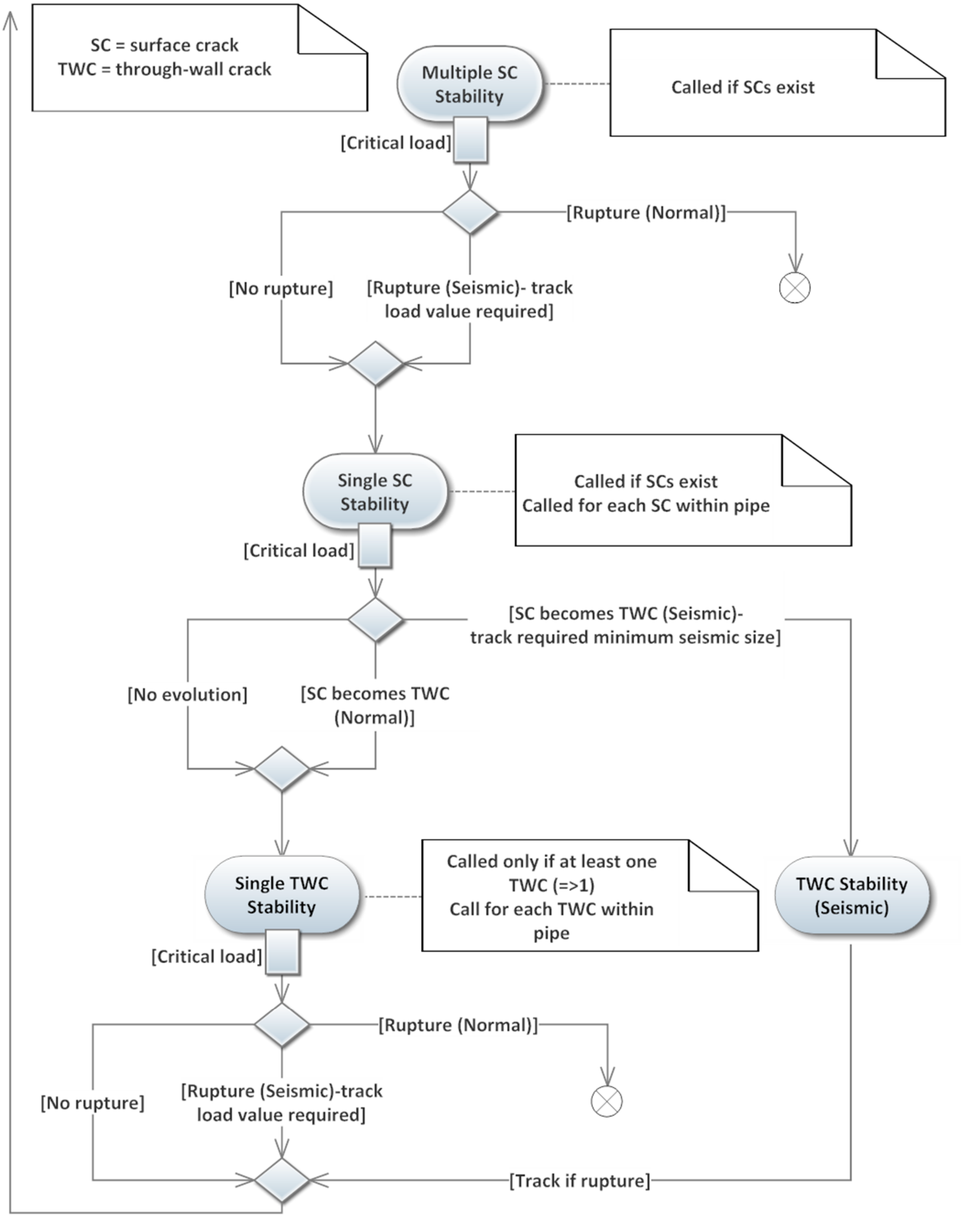


Figure 3-5. Crack Stability Framework implementation.

3.3.3.9.2 Loads and Stresses

To prepare for calls to the crack stability DLLs, the Framework recalculates stresses applicable to circumferential crack stability as a function of the newly updated crack properties. Stresses applicable to axial crack stability are not a function of crack properties and are described in Section 3.3.1.6. The newly updated crack properties, such as crack area and fraction of the cross-sectional area cracked (as described in Section 3.3.1.8), are used to calculate the following axial stresses as described in [12]:

- membrane stress accounting for transients
- bending stress accounting for transients
- membrane and bending stresses due to normal operating loads plus the maximum transient load for SC stability
- membrane and bending stresses due to seismic conditions for SC stability
- membrane and bending stresses due to normal operating loads plus the maximum transient load for TWC stability
- membrane and bending stresses for TWC stability, seismic conditions, and circumferential cracks

After the above stresses and loads are calculated, the Framework calculates the axial loads due to the normal operating loads plus the maximum transient load. If the load limit is exceeded, the Framework issues an error, notes it in the Run Log, and skips the remainder of the realization.

3.3.3.9.3 Multiple Surface Crack Stability

The purpose of the MSC stability module DLL is to determine whether a rupture occurs due to one or more circumferential cracks. It assesses the stability due to all circumferential cracks subject to combined tension and bending loads. In addition to SCs, the Framework includes TRCs and TWCs as input to the module by temporarily converting them to “deep” SCs (i.e., an SC with a non-dimensional crack depth of 0.9999 with an inner half-length equal to that of the TRC or TWC or 0.99π, whichever is smaller to prevent a DLL input error). This temporary conversion is made only for the MSC stability calculation and is not passed to the crack property variables tracked by the Framework.

In the MSC stability calculation, the ultimate moment-carrying capacity of MSCs and each individual SC are compared against the current applied loading based on the pipe and crack geometry, material properties, and loads. The result of this comparison is returned through a flag that indicates whether a failure is predicted. The MSC stability DLL also provides the bending moment ratio, which is a measure of stability that signals rupture when it reaches a value of one or greater.

The Framework calls the MSC stability module DLL using seismic conditions until the DLL determines that there is a seismic instability. As soon as the DLL determines a seismic rupture, the Framework records it and calls the MSC stability DLL again in the same time step using the normal operating loads plus the maximum transient load. For the remaining time steps, the Framework calls the MSC stability module DLL using these loads. If the DLL signals a rupture, the Framework records it, sets the crack types for each subsection to “200,” and ends the realization. The seismic rupture recorded by the Framework is used in post-processing calculations to determine the contribution of a seismic event to the probability of rupture. The MSC stability DLL issues a run error whenever loading, crack size, or both put the neutral axis out of range. When this happens, the Framework effects a rupture.

Of note, the MSC stability DLL assumes that 0 radians is located at bottom dead center of the weld. Because the Framework defines top dead center as 0 radians, it recalculates the crack positions for the DLL by subtracting by π.

3.3.3.9.4 Single Surface Crack Stability

The single SC stability module DLL checks whether an SC is unstable and whether it should become a TWC. Framework calls to this DLL occur separately at each time step for axial and circumferential cracks when SCs are present, except when the number of calls is reduced as described in Section 3.3.3.9.6. The single SC stability DLL uses the constant depth surface crack analysis based on the net-section collapse methodology [26]. The outputs of the single SC stability module DLL include bending moment ratios and a flag for each SC that indicates whether it is stable. For axial cracks, the critical ratio is called the SC ratio or TWC ratio.

The single SC stability module DLL is called using seismic conditions until the DLL determines that at least one of the SCs is seismically unstable. Here, the load inputs are the same as those to the MSC stability module DLL. As soon as the single SC stability module DLL determines that an SC is seismically unstable, the Framework calls the single TWC stability module DLL with the same seismic loads to determine whether a seismic SC instability also causes a rupture. If it does, the Framework records a seismic rupture and continues. From this point on, the Framework calls the single SC stability module DLL using the normal operating loads plus the maximum transient load. If the DLL signals an SC failure, the Framework turns the SC into a TWC. For axial cracks, seismic loads are not considered. The single SC stability module DLL issues a run error whenever loading, crack size, or both put the neutral axis out of range. When this happens, the Framework effects a rupture.

Like the MSC stability module DLL, the single SC stability module DLL assumes that 0 radians is located at bottom dead center of the weld. Therefore, the Framework recalculates the crack positions for the DLL by subtracting by π.

3.3.3.9.5 Single Through-Wall Crack Stability

The single TWC stability module DLL determines whether there is a rupture due to a single TWC or TRC. Framework calls to this module occur separately at each time step for axial and circumferential cracks whenever a TWC or TRC is present, or when an SC fails, except when the number of calls is reduced as described in Section 3.3.3.9.6. The outputs of the single TWC stability module DLL include a bending moment ratio and a stability flag for each TRC and TWC. For axial cracks, the critical ratio is called the SC ratio or TWC ratio.

The Framework initially calls the single TWC stability module DLL using seismic conditions. An average of the inner and outer crack half-lengths is used for the half-length input. As soon as the DLL determines that a TWC or TRC is seismically unstable, the Framework records a seismic rupture and continues. From this point on, the Framework calls the single TWC stability module DLL using the normal operating loads plus the maximum transient load. If the DLL signals a rupture, the Framework records it, sets the crack types for each subsection to “200,” and ends the realization. As with the single SC stability module DLL, seismic conditions are not considered for axial cracks.

The single TWC stability module DLL for circumferential cracks can have a run error if the load in the absence of cracks is slightly less than the load limit. The Framework treats this condition as a rupture (i.e., rupture occurs if a TWC stability module DLL run error occurs and the loads are at least 90 percent of the load limit).

3.3.3.9.6 Stability Call Reduction

Crack stability calculations are typically not necessary at every time step. Calling the stability module DLLs at every time step slows down a realization that has a crack. To speed up simulation times, a method was developed for skipping calls to the stability module DLLs for reasonable periods of elapsed time. The method is implemented for seismic conditions for circumferential cracks and for normal operating loads plus the maximum transient load for axial cracks.

The logic implemented uses the rate of change in the critical ratio, $\frac{dR}{dt}$, to predict when the crack stability module DLLs should be rerun. The critical ratio (i.e., the bending moment ratios for circumferential cracks and SC ratios and TWC ratios for axial cracks calculated by the crack stability module DLLs) is the ratio of the actual load to the critical load. The Framework calculates the rate of change of the critical ratio using Equation 3-2:

Equation 3-2. Rate of change in the critical ratio.

$$\frac{dR}{dt} = \frac{R_i - R_{i-1}}{T_i - T_{i-1}}$$

R_i and T_i are the critical ratio and elapsed time of the critical ratio calculation from the most recent crack stability module DLL execution. R_{i-1} and T_{i-1} are the values from the previous crack stability module DLL execution.

At least one of the following five conditions is needed for the Framework to call a crack stability module DLL in a given time step:

(1). the number or location of cracks change, as affected by new initiations or coalescence, either during the current time step or the previous time step

(2). the normal operating loads including transients change during the current time step or the previous time step

(3). the material properties change, as in the case of inlay or overlay mitigation

(4). the maximum time between calls as specified by the user has been reached

(5). the critical ratio is predicted to move to the next critical ratio to test, which can be no more than a value half way to 1 from its most recently calculated value, either during the current time step or the previous time step.

The first four conditions call the crack stability module DLL at successive time steps so that a rate of change in the critical ratio can be calculated. The next time at which the stability module would be called according to the fifth condition is given in Equation 3-3:

Equation 3-3. Time at which stability module will be called next, according to the fifth condition.

$$T_{i+1} = T_i + \frac{R_{i+1} - R_i}{\frac{dR}{dt}} D_{i+1}$$

where R_{i+1} is given by Equation 3-4:

Equation 3-4. Halfway point between R_i and 1 used to calculate time of next stability call.

$$R_{i+1} = \frac{1 + R_i}{2}$$

and D_{i+1} is given by Equation 3-5:

Equation 3-5. Empirical dampening factor used to calculation time of next stability call.

$$D_{i+1} = (1 - R_i)^x$$

R_{i+1} is the halfway point between R_i and 1, D_{i+1} is an empirical dampening factor, and x is the dampening factor exponent. The dampening factor may be needed to reduce the time between stability tests as R_{i+1} approaches one because the critical ratio can increase nonlinearly with time. A value of zero for this exponent effectively eliminates the dampening factor, values greater than zero reduce the time to the next call, and values less than zero should not be used. Additionally, the rate of change of the critical ratio is not allowed to decrease within each realization unless a physical mitigation has occurred within the last time step. This ratcheting of the rate of change of the critical ratio is needed due to transients that may or may not be present every time step.

The elapsed time until the next stability test is set at T_{i+1} unless it is less than the time step length or more than the maximum time allowed between stability module executions. If it is less than the time step length, it is reset to the time step length. If it is more than the maximum time allowed between stability tests, it is reset to the maximum time allowed. Both the maximum time between calls and the dampening factor exponent are controlled by user inputs (Global IDs 0901 through 0911). Preliminary investigations have found some instances where the evolution of the critical ratio is highly non-linear. Depending on the maximum time between calls and the dampening factor exponent selected by the user, this may lead to a situation where the crack stability module DLLs are not called during a time step when a rupture would have occurred. To aid the user in identifying such a situation, the Framework sends a warning to the Run Log whenever it finds that the crack stability was not evaluated in the time step immediately before failure. The warning tells the user which crack stability module DLL and crack orientation prompted the warning, and it suggests that the user reduce the time between tests, increase the dampening factor exponent, or both. In the extreme, the user can set the maximum time between calls to zero to have the stability modules called at every time step.

An example of stability call reduction is shown in Figure 3-6. In this example, the maximum time between calls is two years, the dampening factor exponent is zero, and the time step is one month. Reduced calls are indicated by the diamond symbols.

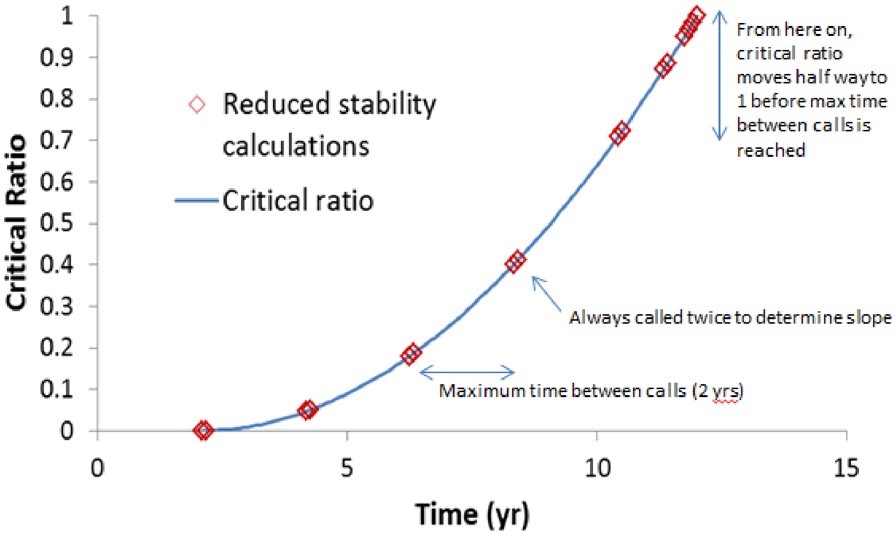


Figure 3-6. Stability Call Reduction Example

3.3.3.10 Crack Opening Displacement

At this point in the time step, the Framework calls the COD module DLLs if there are any TRCs or TWCs. For circumferential cracks, the Framework makes separate calls to the circumferential COD module DLL: one for membrane stress and one for bending stress. The Framework enters updated loads as prescribed in the Stress Requirements technical report [12]. To prevent a run error for circumferential cracks, the Framework ensures that the inner half-length in radians is not π by limiting its value to 0.99π.

Full lists of the inputs to the COD module DLLs are documented in the Circumferential TWC Combined Tension and Bending COD SDD [27] and the Axial TWC COD SDD [28]. They include crack type, inner crack half-length, bending moments, axial membrane force, mixed toughness parameters, pressure, inner pipe radius, and pipe thickness. The Framework assembles the current values for these inputs and provides them to the COD module DLLs.

The deterministic outputs of the COD module DLLs are the:

- membrane and bending COD for circumferential cracks at both the inner and outer diameter locations
- inner and outer diameter COD for axial cracks

The Framework uses these outputs and the correction factors from the crack transition module DLL to calculate the actual inner diameter and outer diameter CODs and crack opening areas (COAs).

3.3.3.11 Leak Rate

The Framework interpolates leak rates for TRCs and TWCs from the look-up tables generated by the LEAPOR preprocessor. There are different look-up tables for PWSCC-initiated and fatigue-initiated cracks, and the Framework chooses the correct look-up table depending on how the crack initiated. The look-up table variables are the COD, crack length, and pipe thickness.

The leak rate table interpolations require the Framework to calculate the fractional binomial coefficient, wetted perimeter, and hydraulic diameter. The first set of interpolations uses the crack half-length and COD at the inner diameter to calculate the leak rate for an idealized TWC. The Framework also fixes the lower bound value for the inner diameter crack length to 0.01 mm. The second set of interpolations uses the crack half-length and COD at the outer diameter to calculate the leak rate when the hydraulic diameter is greater than 30. Interpolations based on the outer diameter crack dimensions are performed in the same manner as those for the inner diameter. The Framework calculates the actual leak rate as an average of the inner and outer diameter values. The leak rate for an axial crack is determined using the axial crack COD and half-length, and the leak rate for a circumferential crack is determined using the circumferential crack COD and half-length.

The Framework can include uncertainty in leak rates less than 10 gpm if the user opts to include it (Global ID 0932). Uncertainty is included by assigning distributions based on covariances at 280 Celsius and 340 Celsius as described in Section 3.8.4 of the Leak Rate Module SDD [19]. The uncertainty factors for each potential crack are sampled pre-time loop, as discussed in Section 3.3.1.9. The leak rates for each through-wall crack with uncertainty are calculated within the time loop as a function of the sampled uncertainty factors, the covariances at 280 Celsius and 340 Celsius, operating temperature, and the actual leak rates described in the previous paragraph.

The final leak rates, with or without uncertainty, are used in post-processing to determine the total leak rate and the probabilities of exceeding the thresholds for small break, medium break, and large break LOCAs, as described in Section 3.4.

3.3.3.12 In-Service Inspections

After leak rates for each TWC and TRC are calculated, the Framework calls the in-service inspection (ISI) module DLL if an inspection is scheduled in that time step. The ISI module DLL determines the probability of detection (POD) and probability of repair (POR) for each crack. POD is the probability of detecting a crack using ultrasonic testing; POR is the probability of repairing the crack based on the sizing by ultrasonic testing.

The Framework determines whether to conduct an inspection based on the user inputs. Inspection timing before and after physical mitigation may be entered as either a frequency or as a fixed set of up to 10 points in time (Global IDs 0811, 0812, 0813, 0808 and 0809). To prepare inputs for the ISI module DLL, the Framework selects the appropriate input parameters based on the elapsed time. It then calls the ISI module DLL separately for inspection and evaluation. The inspection routine calculates the POD and the evaluation routine calculates the POR.

With the values for POD and POR, the Framework proceeds to calculate the overall probability of non-repair (PNR). If the user opts to calculate this probability assuming POD is independent of time (i.e., option 1 for Global ID 0821), the POD at the time of the first inspection is $D_1 = POD_1$ and the POR is POR_1. The probability of repairing the pipe is then $R_1 = D_1 \times POR_1$ and the PNR for the crack is $1 - R_1$. The probability of non-detection in the first inspection is $1 - D_1$, and the probability that the crack has been detected in the first inspection and not repaired is $D_1 \times (1 - POR_1)$. During the second inspection, the Framework assumes that the crack has not yet been detected, so the overall POD at the second inspection is POD_2 plus the probability that it was detected previously but not repaired, so that it can be detected again. As a result, $D_2 = POD_2 \times (1 - D_1) + D_1 \times (1 - POR_1)$. Accordingly, the POR at the second inspection is $R_2 =$

$D_2 \times POR_2$. Note that these equations are valid for the first inspection because $D_2 = 0 + D_1 \times (1 - 0) = D_1$.

For dependency over time (i.e., option 0 for Global ID 0821), a crack detected in the first inspection cannot be detected in subsequent inspections. In this case, during the second inspection, detection of the crack can only be linked to the fact that it was or was not previously detected. Therefore, $D_2 = D_1 \times (1 - POR_1)$. The POR is the same equation, $R_2 = D_2 \times POR_2$, as for the PNR for the crack, $1 - R_2$.

The Framework calculates the PNR for each crack as one minus the POR for the crack. The overall PNR is then calculated differently depending on the option for the number of cracks detected (Global ID 0820). If this option is set such that only the deepest crack is detected, the Framework calculates the overall PNR as one minus the largest POR among all the cracks. If this option is set such that all cracks are detected independently, the overall PNR is calculated as the product of the POR values for all the cracks.

3.4 Outputs and Post-processing

This section describes the outputs and the postprocessing algorithm used to generate them. The available outputs are specified in the SRD [7]. They include general outputs such as the total probability of rupture, orientation-specific outputs such as the probability of occurrence of a circumferential crack, and crack-specific outputs such as the depth of the first crack for each realization. These outputs are calculated using indicator function time histories for each realization, which indicate with a value of 0 that an event (i.e., crack, leak, or rupture) has not occurred and a value of 1 that an event has happened. The time histories are averaged over the aleatory realizations for each epistemic realization and stored in result elements at the epistemic level within the GoldSim model file. The results from each epistemic realization are available to the user through the two Results Dashboards. The user has the option to view the results from individual epistemic realizations or statistics of the results (e.g., mean, median, 5th percentile, 95th percentile, or custom).

Many outputs under specific conditions, such as in-service inspection and leak rate detection, are calculated using a postprocessing algorithm within the Framework that applies assumptions to the nominal crack, leak, and rupture histories calculated in the deterministic model at the end of each aleatory realization. This allows for many different conditions to be considered without the need for multiple simulations, a process that would quickly become computationally prohibitive.

The results available in the Results Dashboards use the GoldSim Time History Result elements. These types of elements allow the user to display time histories, which can be presented as individual time histories, a set of time histories, or as probabilities and statistics of the time histories. Chapter 8 of the GoldSim User's Guide gives more information on viewing results and how the probabilities and statistics are calculated [16]. Because xLPR V2.0 uses a dual loop sampling structure, the user options in the individual time history results can be selected to present either (a) an average over the aleatory realizations for a specific epistemic realization time history, or (b) the specific aleatory realizations for a specific epistemic realization time history depending on the output.

The remainder of this section groups and describes the outputs calculated by the Framework. Common outputs are described in Section 3.4.1, crack orientation-specific outputs are detailed in Section 3.4.2, and crack-specific outputs are given in Section 3.4.3. Error tracking outputs are described in Section 3.4.4.

3.4.1 Common Outputs

Outputs common to both circumferential and axial cracks are provided by default.

The detection effect results show indicator functions for the following:

- occurrence of a crack
- occurrence of a leak
- occurrence of a rupture
- seismic contribution to rupture
- occurrence of rupture due to a safe shutdown earthquake (SSE)
- occurrence of rupture with ISI
- occurrence of rupture with leak rate detection (LRD)
- occurrence of rupture with LRD and ISI
- occurrence of SC ruptures
- occurrence of SC ruptures with ISI
- seismic contribution to SC ruptures
- occurrence of SC ruptures due to SSE

In addition to the above outputs, there are also some useful debugging indicators:

- pipe system without cracks cannot sustain the normal operating loads
- pipe system without cracks cannot sustain the transient loads
- pipe system without cracks cannot sustain the normal operating plus seismic loads

Typically, these indicators would be non-zero if the loads input by the user are too great for the weld geometry and material. If this occurs, the user should revise the loads, geometry, or material property inputs and re-run the simulation.

The LOCA results show indicator functions for LOCA events. The LOCA events may be defined in the User Options spreadsheet in terms of exceeding either a threshold leak rate or a threshold COA. Three levels of LOCA event, small break (SB), medium break (MB), and large break (LB), may be defined. The following LOCA indicators functions are output:

- SB LOCA by leak rate
- SB LOCA by COA
- MB LOCA by leak rate
- MB LOCA by COA
- LB LOCA by leak rate
- LB LOCA by COA
- SB LOCA by leak rate with ISI
- SB LOCA by leak rate with LRD
- SB LOCA by leak rate with LRD and ISI
- SB LOCA by COA with ISI
- SB LOCA by COA with LRD
- SB LOCA by COA with LRD and ISI MB LOCA by leak rate
- MB LOCA by leak rate with ISI
- MB LOCA by leak rate with LRD
- MB LOCA by leak rate with LRD and ISI

- MB LOCA by COA with ISI
- MB LOCA by COA with LRD
- MB LOCA by COA with LRD and ISI MB LOCA by leak rate
- LB LOCA by leak rate with ISI
- LB LOCA by leak rate with LRD
- LB LOCA by leak rate with LRD and ISI
- LB LOCA by COA with ISI
- LB LOCA by COA with LRD
- LB LOCA by COA with LRD and ISI MB LOCA by leak rate

The total leak rate output provides the sum of leak rates from all active circumferential and axial cracks.

The PNR total output provides the combined probability of not repairing a weld for both circumferential and axial cracks. As with the crack-specific PNR, these indicators start at one until an inspection has occurred. As with the circumferential and axial specific PNR, depending on the user input (Global ID 0820), the result for multiple flaws is either the PNR for the deepest flaw only or the product of the PNRs for all flaws.

The leak rate jump indicator tracks the occurrence of the leak rate “jumping” from below a user-defined lower bound value (Global ID 0941) to above a user-defined upper bound value (Global ID 0942) within one time step. If a rupture occurs during a time step, the leak rate for this indicator is assumed to always be greater than the user-defined upper bound. The specific outputs are:

- leak rate jump
- leak rate jump with ISI
- leak rate jump with LRD
- leak rate jump with ISI and LRD

3.4.2 Crack Orientation Specific Outputs

By default, the following crack orientation-specific results are stored and made available to the user.

- indicator functions
- fractional area damaged for circumferential cracks only
- total cracked area for axial cracks only
- number of cracks
- PNR

The indicator function plots provide indicators for the occurrence of rupture, the occurrence of leak, and the occurrence of a crack either due to circumferential cracks only or due to axial cracks only. Within one realization, the occurrence calculation is either 0 for no occurrence or 1 for occurrence. However, since the results here are the average of the aleatory realizations for each epistemic realization, they represent probabilities of occurrence.

The fractional area damaged plot shows the fraction of area damaged by all the circumferential cracks. It is the sum of the total crack area assuming idealized crack shapes divided by the total weld cross-sectional area.[1]

The total cracked area plot shows the total area damaged by all the axial cracks. The total crack area is calculated as the sum of all the axial cracks assuming idealized crack shapes.[1]

The number of crack plots show the total number of active circumferential and axial cracks at any time in the simulation. The total number of circumferential cracks in any simulation may increase over time due to initiations that occur at various times in the operating period and may decrease due to the coalescence of cracks as they grow closer. Axial cracks cannot coalesce, so they only increase in number with time.

The PNR plots provide the PNR due to inspections where circumferential or axial cracks may have been detected. The values always start as 1 since the weld will never be repaired prior to inspection. After inspection, the PNR for a single flaw is the product of the POD determined by the ISI module's inspection routine and 1 minus the POR returned by the ISI module's evaluation routine. If more than one flaw is present, then the result is either the value for the deepest flaw only or the product of all the PNR for all flaws, depending on whether the user selected only the deepest crack to be detected or to detect all the cracks independently (Global ID 0820).

3.4.3 Crack-Specific Outputs

Crack-specific results for both circumferential and axial cracks can be accessed through the respective Results Dashboards. By default, the Results Dashboards contain crack-specific results for the first five cracks. The Framework numbers the cracks sequentially according to their initiation times. As a result, crack numbers one through five will have relatively longer histories from initiation to rupture and are thus assumed to be of interest.

The default crack-specific outputs for the first five axial and circumferential cracks are:

- crack type
- crack position
- leak rate
- inner half length
- crack depth
- outer half length
- stress intensity factors

The crack type, position, inner half-length, depth, and outer half-length display the primary crack property results discussed in Section 3.3.1.8.

The stress intensity factors at the crack tips due to the normal operating loads are also provided. For TWCs, values are available at the inner and outer surface tips. For SCs, values are available at the 0° crack tip location, which corresponds to the deepest point of the crack, and at the 90° crack tip location, which corresponds to the inside surface tip. Due to symmetry assumptions, the

---

[1] The use of inlay mitigation removes all existing crack areas, even for cracks that had a depth greater than the inlay depth.

same stress intensity factor is used to grow the inner and outer crack tips for TRCs and TWCs, and a single stress intensity factor is used to grow the inner crack tip for SCs.

The volumetric leak rate for each crack is also reported for the first five axial and circumferential cracks. This output is calculated as described in the Leak Rate Module SDD [19].

3.4.4 Error Tracking Outputs

Error tracking is provided through the Error Tracking Dashboards. Errors are tracked for axial and circumferential cracks separately. Error codes are returned from each of the modules and are generally classified as a 100-, 200-, or 300-level errors as follows.

Typically, a 100-level error is reported by a module when it receives an input value that is outside the range of validity. Generally, these errors are fatal and are shown in the Error Tracking Dashboard as a “Stop” sign. The user should resolve this type of error by stopping the simulation and checking the associated input(s). In most cases, the inputs to a module can be traced directly back to an input in the Inputs Workbook, unless it is derived from a combination of inputs.

Typically, a 200-level error is associated with a module run time error. Generally, these errors are fatal and are shown in the Error Tracking Dashboard as a “Stop” sign. The user should resolve this type of error by stopping the simulation and determining the cause. These errors are harder to debug, although it is common for them to be the result of some extreme input value sent to the module, though the value may still be within the allowable range.

Typically, a 300-level error produces a warning and is shown in the Error Tracking Dashboard as a yellow “Warning” sign. This type of error may impact the validity of the results. The user should record the warning but may continue the simulation. A common example of this type of warning is when a combination of inputs is outside the validated range of the module.

By default, if the Framework encounters a 100- or 200-level error during the simulation, then the simulation will be interrupted with a message to alert the user. There are exceptions, however, as some modules, such as the COD modules, were developed with warnings, not fatal errors, logged as 100- or 200-level errors. While not recommended, the user has the option to turn this warning off using a flag in the Inputs Workbook (Global ID0931).

The Error Tracking Dashboard provides a list of error descriptions for each module that can be used to determine the source of an error code. A time history is also available to allow the user to determine when the error occurred during the simulation. Modules that were not called and thus cannot have experienced an error are shown using a grey square. If a module experienced unique error codes for different cracks, a flag will be shown next to the error indicator.

The Error Tracking Dashboard can be found in the Result Options layer of the Global Control Dashboard. The circumferential Error Tracking Dashboard is shown in Figure 3-7.

Figure 3-7. Error Tracking Dashboard for circumferential cracks.

Further details on error messages, troubleshooting, and reporting can be found in Chapter 5 of the User Manual [14].

4. DYNAMIC LINK LIBRARIES

xLPR V2.0 was designed with a modular structure to facilitate incorporation of future enhancements. This modularity was accomplished using DLL files (a shared library concept in the Microsoft Windows operating system) and the capability of the GoldSim software to couple with external functions implemented as DLL elements. The external functions or modules, which in this case are subroutines developed by the Models Group in Fortran, are linked into the GoldSim model file during the simulation as part of the deterministic model. Linking these modules requires a "wrapper" around the module, and the wrapper and module are then compiled together into a single DLL file. In addition to the module-specific DLLs, the Framework utilizes several other DLLs for utility purposes to implement software licensing features.

The Framework's module-specific DLLs are listed below along with a short description of the functions they perform and a reference to each module's respective report.

- Axial Through-Wall Crack Opening Displacement (AxCOD) Module DLL – Calculates the COD for axial TRCs and TWCs [29].

- Circumferential Through-Wall Crack Combined Tension and Bending COD (CrCOD) Module DLL – Calculates the COD for circumferential TRCs and TWCs [29].

- Fatigue Crack Initiation (CIF) Module DLL – Calculates the number of fatigue crack initiations, initiation time, and initiation location [30].

- Crack Coalescence Module DLL – Calculates whether two adjacent circumferential cracks coalesce [31].

- PWSCC and Fatigue Crack Growth Rate (CGR) Module DLL – Calculates crack growth due to PWSCC and fatigue [31].

- PWSCC Crack Initiation (CI-SCC) Module DLL – Calculates the number of PWSCC initiations, initiation time, and initiation location [30].

- In-Service Inspection (ISI) Module DLL – Calculates the probability of detection (POD) and probability of repair (POR) for simulated ultrasonic inspections [32].

- K Calculator for Circumferential and Axial Part-Wall Cracks (KPW) Module DLL – Calculates stress intensity factors for axial and circumferential SCs at the deepest point and at the interior surface point [33].

- K Calculator for Circumferential and Axial Through-Wall Cracks (KTW) Module DLL – Calculates stress intensity factors for axial and circumferential TWCs [33].

- Leak Rate (LEAPOR) Module DLL – Calculates leak rates from TWCs. This DLL is accessed by the Preprocessor to generate a set of look-up tables that are read into the GoldSim model file [34].

- Thermal Stress Intensity Factors for Any Coolant History (TIFFANY) Module DLL – Calculates the cyclic stress intensity factor values for axial and circumferential SCs and

TWCs due to fatigue loads. This DLL is accessed by the Preprocessor to generate a set of look-up tables that are read into the GoldSim model file [35].

- Axial Crack Stability (AxCS) Module DLL for SCs – Calculates the stability of a single axial SC subject to pressure loads [36].
- Circumferential Surface Crack Stability (SC-Fail) Module DLL for multiple SCs – Calculates the stability of multiple circumferential SCs subject to combined pressure, tension, and bending loads [36].
- Circumferential Surface Crack Stability (SC-Fail) Module DLL for single SCs – Calculates the stability of a single circumferential SC in a pipe subject to combined pressure, tension, and bending loads [36].
- Axial Crack Stability (AxCS) Module DLL for TWCs – Calculates the stability of axial TWCs subject to pressure loads [36].
- Circumferential Through-Wall Crack Stability (TWC-Fail) Module DLL – Calculates the stability of circumferential TWCs subject to combined pressure, tension, and bending loads [36].
- Crack Transition Module (CTM) DLL –Calculates correction factors that are applied to the idealized stress intensity factor solutions and COD values to simulate the transition from a semi-elliptical SC to an idealized TWC [37].

The Framework’s utility DLLs are listed below along with a short description of their functions.

- calu_nl DLL – Called by the other module DLLs during their initialization to check for a valid xLPR V2.0 software license. This DLL must be present along with a valid license in order to run a simulation.
- msvcp110 DLL – Microsoft Visual C++ redistributable DLL called by the calu_nl DLL for the license check.
- msvcr110 DLL – Microsoft Visual C++ redistributable DLL called by the calu_nl DLL for the license check.
- sdf DLL – Simple database format DLL accessed by the Preprocessor to inspect existing preprocessor results.

Detailed information on the individual modules listed above can be found in the referenced reports. Information regarding the versions of each DLL and the required structure for running them with the Framework can be found in the User Manual [14].

5. TESTING AND RESULTS

This section describes the testing that was performed to verify and validate the Framework. Verification testing ensured that the Framework complies with its documented requirements. Validation testing demonstrated that the software will fulfill its intended use. V&V testing efforts included unit, integration, and acceptance testing, as described in the following sections. The tests were performed on “Revision 12” (Beta_v2.1_R12) of the Framework. Subsequent changes were made to the Code in “Revision 13” (Beta_v2.1_R13) and in “Version 2” (xLPR_v2.0). Greater detail regarding these changes is given in the SVVR [46]. Due to project constraints and the expected low technical risk of these changes, the V&V team did not re-execute formal Framework testing for these subsequent releases. While it was determined that the risk of the changes referenced above was sufficiently low to forgo additional testing prior to release, it was recommended that regression testing be performed on the Framework prior to use for production analyses (e.g., analyses used to inform plant or regulatory decision-making).

5.1 Verification Testing

The Framework Software Test Plans (STPs) [9], [10] and Software Test Results Reports (STRRs) [38], [39] respectively describe the verification test cases used for assessing the performance and functionality of the Framework and discuss the results obtained from those test cases. Verification testing consisted of two parts: unit testing and integration testing. Unit testing was conducted to assess functionality of the Framework itself, and integration testing was conducted to assess functionality of the modules and Framework together.

Unit and integration testing consisted of both static and dynamic tests. Static tests involved the tester manually inspecting the source code to verify that a specific requirement was met. The static tests were generally intended to verify that the intent of the developer matched the functional requirements. They were also intended to identify programming errors or any general issues with coding style. Dynamic tests involved running the Code with a pre-defined set of inputs and verifying that the software met the specific performance requirement by comparing the test results to a pre-defined, expected result. The dynamic tests were primarily intended to directly verify correct behavior of the software in cases where there was a significant risk of an error that would not have been easily identified through a static inspection [10].

A Software Test Log for each of the tests described in the STPs was used to document the results from executing the test procedure. These logs are included as attachments to the STRRs. Anomalies encountered during testing were either fixed and verified through subsequent testing, or they were accepted and documented as known errors in the applicable STRR and User Manual [14].

5.1.1 Unit Testing

Framework Unit Testing is described in detail in the Unit STP [10] and the details of each test are provided in the Unit STRR [38]. A summary of this information is provided below.

The Unit STP includes focused test cases that were designed to assess one or more requirements included in the SRD [7] or design elements included in the SDD [8].

The Software Verification and Validation Plan (SVVP) [40] identifies the methods and criteria that were used for the Unit STP to verify the following [10]:

- conformance to all applicable software requirements as a stand-alone unit
- performance at boundaries (e.g., data and interfaces)
- traceability of source code elements
- compliance with applicable coding standards
- performance consistent with author intent
- critical characteristics of commercial off-the-shelf software on which the Framework or module depends (e.g., Microsoft Excel)

The Unit STP describes the test cases that were performed [10]. Each section of testing lists the Framework requirements along with descriptions of those requirements and provides the acceptance criteria used to determine whether the requirement had been met. The focus for each section of testing is listed below. For Level 1 and Level 2 testing, static and dynamic tests were addressed separately.

- Level 1 testing verified the functionality of the graphical user interface and Inputs Workbook, preprocessors, epistemic and aleatory sampling, calls to the crack initiation modules, invocation of the time loop, and post-processing

- Level 2 testing verified the requirements for the operating time loop

- DLL interface testing verified the correct passage of information between the GoldSim model file and the module DLLs

- Framework output testing verified the requirements for the set of ASCII text files and corresponding plots produced by the GoldSim model file that contain the specified information for all realizations executed

- Software attributes testing verified the requirements for Framework software attributes including the software used for implementation of components

The unit test results are described in the Unit STRR [38]. Unit testing resulted in five rounds of testing: one round for each beta version of the Framework starting with "Revision 8" and ending with "Revision 12." Starting in the third round, only test cases that had previously failed or were impacted by changes to the Framework were tested.

Only one error remained unresolved after completion of the unit testing. This error, recorded as xLPR-699, occurs in the calculation of the effective bending moment when loads are specified in terms of forces and moments rather than as stresses. As stated in the Unit Testing STRR, this issue was decided to be too large to fix at the time and was accepted as a known error [38]. The effect has been investigated and recommended user actions to avoid the error are detailed in the User Manual [14].

5.1.2 Integration Testing

The Integration STP [9] defines the set of test cases used to determine whether the Framework and incorporated modules encompassed all of the functional requirements related to module integration as specified in the SRD [7]. Like unit testing, both static and dynamic tests were performed.

The SVVP [40] identifies the methods and criteria used for the Integration STP. The tests in the Integration STP were applied to verify the following [9]:

- compliance with critical software requirements when the Framework and modules are integrated together as a single system
- compliance of the interfaces between the Framework and the modules to software requirements and design elements
- performance of the integrated system at boundaries (e.g., data and interfaces).

The results of the test cases for static and dynamic integration testing are described in the Integration STRR [39]. As with unit testing, integration testing resulted in five rounds of tests, and starting in the third round, only test cases that had previously failed or were impacted by changes to the Framework were tested. The fifth round of integration testing executed test cases that were revised to include expanded unit checking in response to an issue found in the fourth round of testing.

Only one error remained unresolved after completion of the integration testing. This error, recorded as xLPR-805, is due to a discrepancy in the implementation of the Weibull model for PWSCC initiation as compared to the technical basis documented in the report on PWSCC initiation model parameter development [15]. Though the error is detectable, the low magnitude of the discrepancy allows the Weibull model to be used without any correction or additional user action. Accordingly, it was accepted as a known error and is documented in the User Manual [14].

5.2 Validation Testing

The purpose of Acceptance Testing was to validate the Framework. The Acceptance STP [11] provides test case descriptions that include case-specific inputs, a description of the test case goal, specifications for the testing procedure, and the procedure for validating the results of the test case [11]. The Acceptance STP was designed to address two goals:

(1). Validation of the overall modeling approach for generic applications
(2). Validation of the output probability of rupture, primarily for LBB applications

The Acceptance STP defines three types of tests to accomplish these goals:

(1). Service Tests compared the results of the Code to in-service experience, where information about in-service cracking was available or where detailed deterministic calculations had been made to predic crack growth and leakage.

(2). Model Behavior Tests compared the overall behavior of the Code with expert understanding of the expected system behavior

(3). Benchmark Tests evaluated the Code against comparable codes, such as PROMETHEUS [41] and PRAISE [42].

The Acceptance STRR [43] documents the detailed results for each of the test cases described in [11]. The test cases were categorized based on their outcome as follows:

- Pass: The test performed as expected, and the results are in line with the description in the Acceptance STP.

- Miss: The outcome was not as initially anticipated, but the results could be explained with further study and are not linked to any issue in the Code itself.

- Fail: The test revealed an issue requiring the Code to be corrected.

The limited amount of field data made direct comparisons between Code results and field data difficult. Based on these difficulties, alternative approaches to traditional validation testing were undertaken with comparisons to field data made where possible. The Benchmark Tests and Model Behavior Tests provide confidence in the results in those cases where direct comparisons to field data were not possible.

The conclusion from the acceptance testing results is that xLPR V2.0 is considered suitable for its intended use and ready for the applications and maintenance phase. The following subsections describe the sets of tests defined for each validation test type and summarize the Acceptance Testing results that, in combination, led to this conclusion.

5.2.1 Service Tests

The Service Tests compared xLPR V2.0 results against data from two plants in the United States, two Swedish plants, and one Japanese plant. The service experiences chosen for comparison included:

- a leaking axial crack and several circumferential cracks found at Virgil C. Summer Nuclear Station, Unit 1

- five identified cracks in steam generator dissimilar metal welds at North Anna Power Station, Unit 1

- several axial cracks found in the reactor pressure vessel nozzle to safe end welds at Ringhals, Units 3 and 4

- one TWC and multiple axial cracks in safety and relief valve nozzles at Tsuruga, Unit 2.

The data pertaining to the service experiences used for the Service Tests is detailed in the Acceptance STP [11]. Several individual test cases were created for each Service Test to address different conditions or to consider different metrics for evaluation.

Generic test case inputs were developed for the reactor pressure vessel nozzle, steam generator nozzle, and the pressurizer safety relief nozzle. Due to the limited availability of plant-specific data, the plant-specific test cases relied on the generic test case inputs and plant-specific inputs were applied when known. Both generic and field event cases, inputs, and expected results are described in detail in the Acceptance STP [11].

The full set of Service Tests validated xLPR V2.0 with respect to time to axial crack depth, growth, and leakage; circumferential crack growth, depth, and leakage; and flaw formation. The results and analyses from these tests are detailed in the Acceptance STRR [43]. Of the sixteen tests performed, two were categorized as misses and fourteen were categorized as passes. The two missed tests were investigated and the deviations from the expected results were attributed to the WRS profiles used in the tests. The tests are considered misses because the results are consistent with expected behavior given the WRS profiles.

5.2.2 Model Behavior Tests

In addition to the Service Tests, acceptance testing also focused on validating behavior related to the implementation of the Framework. Because there have been no observed cases of ruptures in the field, explicit validation of the rupture results from xLPR V2.0 against field data was not possible [11]. Similarly, the full range of intended use of xLPR V2.0 is not covered by field data. To validate the Code despite these limitations, the Framework Acceptance STP defined Model Behavior Tests to establish that, in the absence of relevant data, the Code behaved as expected according to the opinion of subject matter experts. To ensure transparency, the judgement of these experts is fully documented in the Acceptance STRR [43].

The Acceptance STP defined five Model Behavior Tests to demonstrate xLPR V2.0 behavior when:

(1). elevated probabilities of SC rupture are expected
(2). the frequency of ISI is increased
(3). inputs are varied to increase the probability of break-before-leak-detection
(4). Importance Sampling is applied to improve convergence when calculating outputs with a very low probability of occurrence
(5). mitigation options are changed

Detailed results and analyses from the Model Behavior Tests are documented in the Acceptance STRR [43]. Of the five Model Behavior Tests performed, one was categorized as a miss and four were categorized as passes. The results of the missed test showed that the probability of a break-before-leak was higher for a case with elevated bending stress than for the base case, which was the opposite of the expected result. However, after further investigation that involved calculating the critical bending moment required for rupture of a circumferential SC at a fixed length and depth, the subject matter experts determined that the behavior of the Code was reasonable due to the interplay between bending stress, initiation, and rupture.

5.2.3 Benchmark Tests

Benchmark Tests were defined using the Service Test inputs to examine the performance of xLPR V2.0 as compared to alternate codes. Comparisons were made with the PROMETHEUS code, the beyond-PRAISE code, and field data. PROMETHEUS and beyond-PRAISE aim at solving the same problem as xLPR V2.0 and precursors to both codes have been used and accepted in regulatory applications. More information on these codes and their use in the Benchmark Tests is given in the Acceptance STP [11]. Advanced finite element analyses were also used. The Benchmark Tests were used to validate behavior regarding axial crack growth and leakage and circumferential crack growth by applying the alternate codes as detailed in the Acceptance STP [11].

The outcomes of the Benchmark Tests are summarized along with the Service Test outcomes in the Framework Acceptance STRR [43]. All Benchmark Tests were classified as passes. There were small discernable variations in the results; however, these variations were expected and deemed insignificant.

6. SCENARIO ANALYSES

After conclusion of the verification and validation testing, the readiness of xLPR V2.0 was further evaluated by applying the Code to analyze 11 different scenarios. This effort, originally scoped to provide guidance on sampling options for specific scenarios of interest, exercised the various options, components, and sampling schemes available in xLPR V2.0. It also served to develop and exercise methods for analyzing and interpreting output from the Code. [17].

Scenario Analyses were performed on a comprehensive range of scenarios covering the essential features and options available in xLPR V2.0 using input sets developed by the Inputs Group [45]. Each scenario examined the behavior of a reactor pressure vessel outlet weld over the course of a 60-year simulation using different crack initiation, crack growth, flaw orientation, and mitigation options. The defining features for each scenario are:

- Scenario 1 considers initial flaws and fatigue growth of circumferential cracks only
- Scenario 2 considers PWSCC initiation and growth of circumferential cracks only
- Scenario 3 considers PWSCC initiation and growth of both circumferential and axial cracks
- Scenario 4 considers PWSCC initiation and growth of both circumferential and axial cracks with MSIP® mitigation occurring at year 20
- Scenario 5 considers PWSCC initiation and growth of both circumferential and axial cracks with MSIP® mitigation occurring at year 40
- Scenario 6 considers PWSCC initiation and growth of both circumferential and axial cracks with zinc mitigation occurring at year 20
- Scenario 7 considers PWSCC initiation and growth of both circumferential and axial cracks with hydrogen mitigation at year 20
- Scenario 8 considers PWSCC initiation and growth of both circumferential and axial cracks with zinc and hydrogen mitigation occurring at year 20
- Scenario 9 considers PWSCC initiation and growth of both circumferential and axial cracks with inlay mitigation occurring at year 40
- Scenario 10 considers PWSCC and fatigue initiation and growth of both circumferential and axial cracks with MSIP®, zinc, and hydrogen mitigation occurring at year 40
- Scenario 11 considers fatigue initiation and growth of both circumferential and axial cracks

Scenarios 1 and 11 include only fatigue crack growth and experienced very different results than the other scenarios. Scenarios 2 through 10 are defined by PWSCC initiation and growth. Scenario 3 is used as a comparative base case for Scenarios 2 and 4 through 10.

A procedure was followed for each scenario that iterated sampling options to analyze the impact of the sampling method and epistemic sample size on results relevant to that scenario. Multiple epistemic sample sizes were used with SRS and LHS. Sensitivity analyses using rank regression were then applied to determine the most effective uncertain inputs for Importance Sampling. Importance Sampling was then applied accordingly, and the results compared to the results form SRS and LHS [17].

Minor code additions were required to generate the data needed to conduct the scenario analyses. These additions were made using the GoldSim version control features and did not impact the calculations or results generated by the Code. New elements were added to export all the sampled inputs for each realization at the aleatory level so that the sampled inputs could be analyzed along with the outputs of interest using sensitivity analysis techniques. In addition, outputs of interest that were not available, such as maximum crack properties at each timestep, were added so that they could be included in the analyses. Finally, elements were included to extract maximum crack properties taken as the maximum value of crack depth, inner half-length, and outer half-length over all subunits at each timestep.

Example results from the Scenario Analysis report are plotted in Figure 6-1. This plot shows approximations of the mean probability of rupture from circumferential cracks for different epistemic sample sizes and sampling methods for Scenario 3.

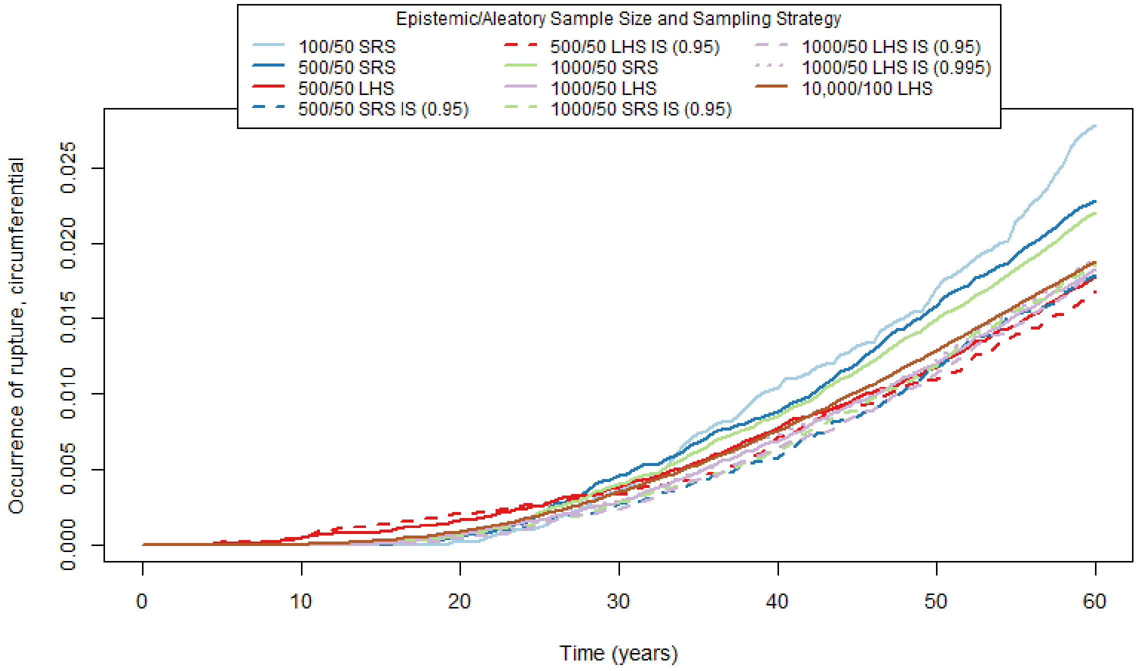


Figure 6-1. Approximation of rupture probability using iterations on the sampling options [17].

The results from Scenario 3 were compared to results from those scenarios that included mitigation to determine whether application of the various mitigation types performed as expected. The "occurrence of rupture" results for select scenarios are shown in Figure 6-2, Figure 6-3, and Figure 6-4. A more comprehensive comparison, including comparisons of results for additional outputs of interest, such as occurrence of crack and occurrence of leak, can be found in the Scenario Analysis Report [17].

The impact of implementing MSIP® mitigation at 20 years as defined for Scenario 4 is shown in Figure 6-2. Here, MSIP® mitigation was found to be particularly effective at limiting crack growth because the crack properties saw marginal increases from year 20 to year 60. The same impact was found for Scenario 5, but it was shifted in time due to its application at year 40.

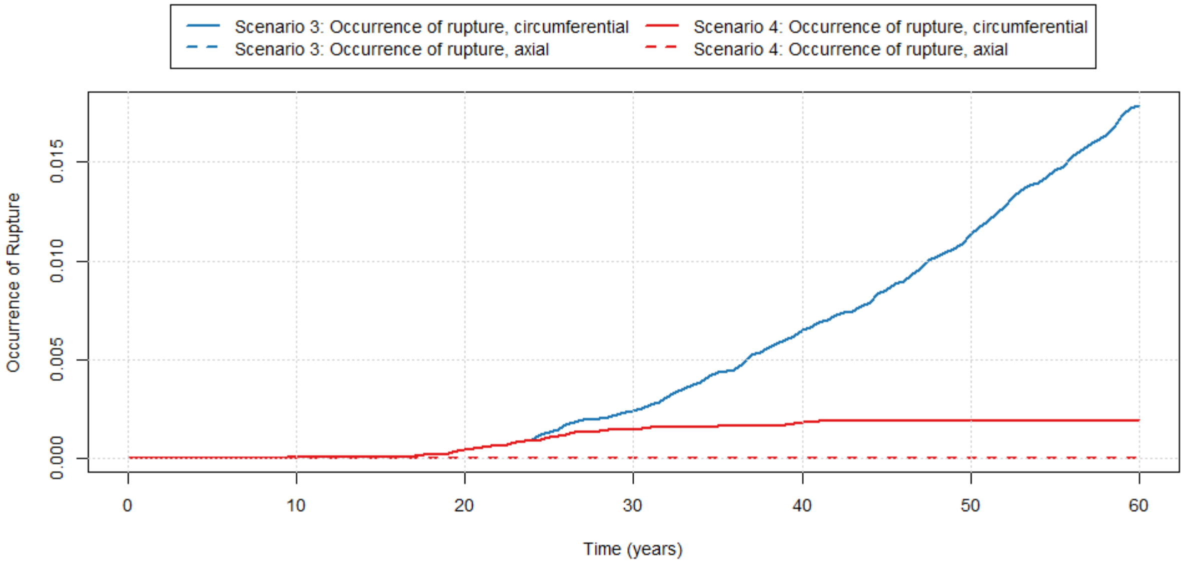


Figure 6-2. Mean probability of axial (dashed lines) and circumferential (solid lines) rupture occurrence for Scenario 3 (blue lines) and Scenario 4 showing the effect of MSIP mitigation application at 20 years (red lines) [17].

The impact of implementing zinc and hydrogen mitigation at 20 years as defined for Scenario 8 is shown for the occurrence of rupture in Figure 6-3. Zinc mitigation was applied separately in Scenario 6 and was found to slow crack initiation, resulting in fewer cracks or the delayed occurrence of cracks as expected. Hydrogen mitigation was applied separately in Scenario 7 and was found to slow crack growth as expected, leading to fewer leaks and ruptures. These two mitigation types were applied together in Scenario 8 and, as expected, decreased both crack dimensions and the total leak rate.

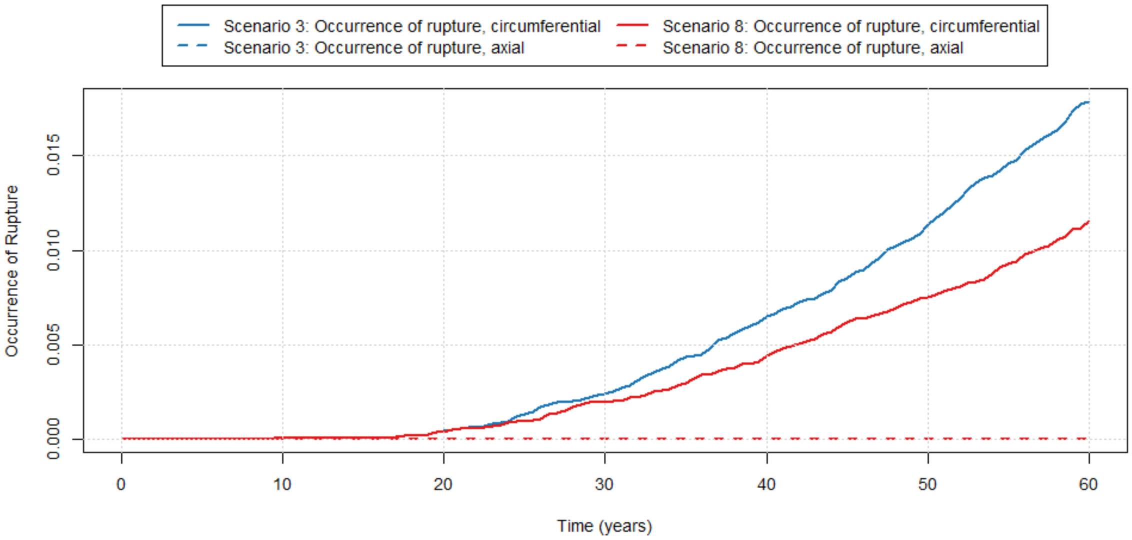


Figure 6-3. **Mean probability of axial (dashed lines) and circumferential (solid lines) occurrence of rupture for Scenario 3 (blue lines) and Scenario 8 showing the effect of application of hydrogen and zinc mitigation at 20 years (red lines) [17].**

Inlay mitigation was studied under Scenario 9. The impact of the inlay is shown for the occurrence of crack rupture in Figure 6-4. Under the inputs defined for this scenario, inlay mitigation was found to result in higher probabilities of occurrence of axial cracks, circumferential cracks, and axial crack leaks at year 60 when compared to Scenario 3, as shown in Figure 6-5 and Figure 6-6. Closer inspection of the inputs suggested that the cause of this increase is due to the tensile inner diameter post-mitigation WRS. However, the resulting WRS values do not cause occurrences of circumferential leaks or ruptures to be greater than those found in Scenario 3.

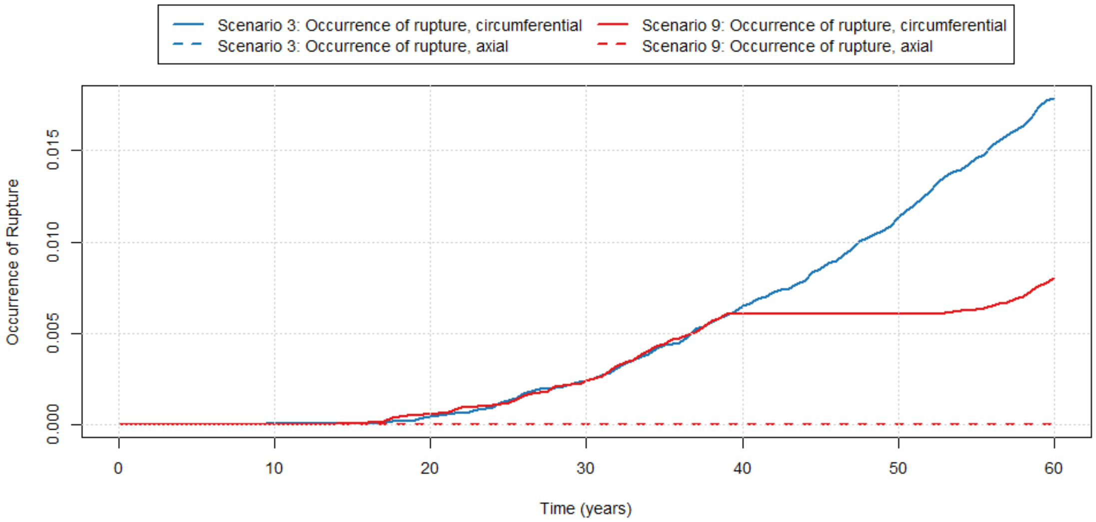


Figure 6-4. **Mean probability of axial (dashed lines) and circumferential (solid lines) rupture occurrence for Scenario 3 (blue lines) and Scenario 9 showing the effect of inlay mitigation applied at 40 years (red lines) [17].**

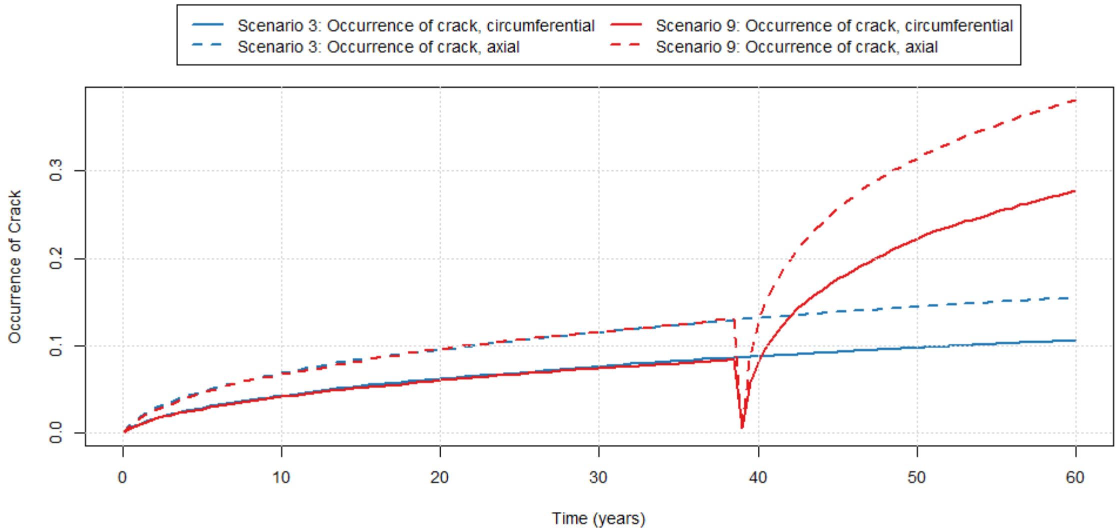


Figure 6-5. Mean probability of axial (dashed lines) and circumferential (solid lines) crack occurrence for Scenario 3 (blue lines) and Scenario 9 showing the effect of inlay mitigation applied at 40 years (red lines) [17].

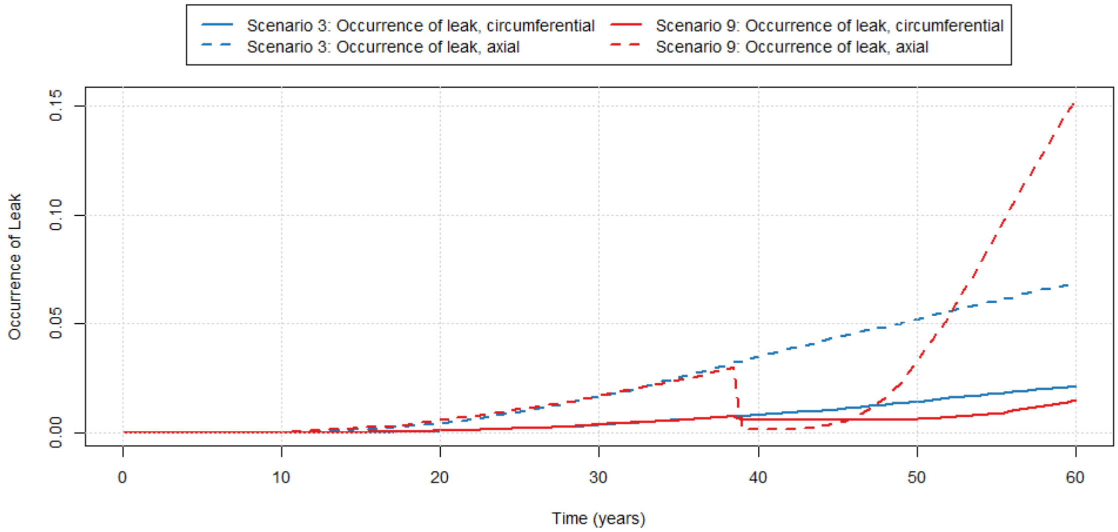


Figure 6-6. Mean probability of axial (dashed lines) and circumferential (solid lines) leak occurrence for Scenario 3 (blue lines) and Scenario 9 showing the effect of inlay mitigation applied at 40 years (red lines) [17].

Scenario 10 examined the impact of implementing zinc, hydrogen, and MSIP® mitigation at 20 years. The results from this scenario intuitively mirror the results from the other scenarios where these mitigation types were considered separately. The inclusion of fatigue crack initiation and growth was not found to have a strong impact on the Scenario 10 results.

Fatigue growth was studied without the inclusion of PWSCC initiation and growth in Scenarios 1 and 11. The results of Scenario 1, which assumed an initial flaw distribution, show very little crack growth and low variability across all realizations, even for simulations using Importance Sampling.

Using the Scenario 1 inputs, the same mean crack properties were found regardless of the selected sampling options. Scenario 11 examined both fatigue crack initiation and growth. The original transients defined for this scenario did not result in any cracks, so a hypothetical transient was also considered. Leaks and ruptures were not observed for Scenario 11, and crack occurrence was found to be much lower than the results found in the scenarios that included PWSCC.

In addition to the individual scenario analyses, separate analyses were conducted of aleatory sampling uncertainty, Importance Sampling methods, comparison of LHS and SRS, random seed selection, and methods for bounding low probability events.

The scenario analyses demonstrated the systematic use of sampling options available in xLPR V2.0. It showed that increasing epistemic sample size decreased uncertainty as expected. It also demonstrated methods for quantifying and decreasing uncertainty using the sampling options. Furthermore, by demonstrating the effectiveness of LHS for reducing uncertainty in the results, it confirmed that the functionality of the sampling options is consistent with the intent to include them in xLPR V2.0 [17].

As a demonstration of Code readiness, the scenario analyses also succeeded in applying the Code extensively in a manner consistent with its intended use. The analyses support confidence in the functionality of the implemented sampling options and provide a detailed procedure for determining the most suitable sampling options for a given problem. The various scenarios also explored a multitude of options available in the Code. Although the conclusions drawn from the scenario analyses are constrained by the veracity of the inputs used, in each case, xLPR V2.0 results were consistent with expectations, which further supports the conclusion that the Code is ready for application.

7. RECOMMENDATIONS

General limitations in the Framework stem from the GoldSim software used to implement the probabilistic structure and to connect the module DLLs in the deterministic model. GoldSim was selected to take advantage of its many built-in probabilistic tools and the ability to visualize the complex structure of the Framework using a graphical environment. As such, GoldSim allows some features of the Framework to be implemented with ease, such as simple random sampling, interfaces with Microsoft Excel, visualization of results, and calculation of statistics. However, as the complexity of the Framework grew during development, the use of GoldSim also presented some challenges to implementation, which necessitated either inefficient or elaborate workarounds.

One such challenge resulted from the need to separate aleatory and epistemic uncertainties. GoldSim does not have a native structure for implementing nested sampling loops. Implementation required the use of a submodel element, which is described in more detail in Section 3. This element requires an interface to communicate values from the aleatory loop to the epistemic loop. This imposes a structure on the Framework that is somewhat inflexible and requires some sampling options to be defined in the Inputs Workbook, while others must be defined directly in the GoldSim model file. The use of this submodel element also imposes difficulties in the calculation of results. Native GoldSim results elements are not designed with the representation of both aleatory and epistemic uncertainties in mind and are therefore unable to save and export both types of results concurrently. In addition, any outputs of interest calculated in the aleatory loop must be passed to the epistemic loop to be averaged at this level. This implementation means that only a subset of possible results is currently available to the user, and any additional outputs of interest would have to be added to the interface in a modified version of the GoldSim model file. The Computational Group recommends that potential additional outputs be collected and assessed for inclusion in the GoldSim model file.

Similarly, GoldSim is only capable of representing time using a constant time step. This simplifies implementation, but it also prevents the use of adaptive time stepping. The default one-month time step is often sufficient, but for events that progress rapidly, such as leaks progressing to ruptures, a smaller time step might be desirable once the cracks initiate or when leaks occur. Without adaptive time stepping, the user would need to re-run the entire simulation with a smaller time step, which is inefficient and can quickly become computationally infeasible due to the amount of data that is saved.

In addition to imposing certain structures in the Framework, there are features in GoldSim that do not function as anticipated. Notably, GoldSim contains an algorithm for Importance Sampling that greatly simplifies implementation; however, it severely limits the possible Importance Sampling distributions. This prevents Importance Sampling from being used to its full potential in overcoming sample size limitations to efficiently estimate extremely rare events. The Computational Group recommends discussions with GoldSim to see if this Importance Sampling implementation could be updated in a future release. In addition, the Computational Group recommends the development of user strategies for implementing importance distributions via the Inputs Workbook and post-processing related results accordingly outside of the GoldSim model.

The allocation of system memory in the GoldSim architecture limits the number of samples that can be used in a single simulation. Results are only available at the end of the entire simulation after all epistemic and aleatory realizations have been run. This means that memory must be allocated to store results over the entire course of the simulation, rather than continuously writing results to a separate file where memory can be reallocated as needed. If a prohibitive number of samples are used, memory limitations can cause simulations to terminate without warning, sometimes after many hours of execution. This issue is particularly impactful when calculating extremely rare events because when Importance Sampling and LHS are insufficient to approximate a low probability, it may be necessary to combine the results from multiple simulations using other software tools. The Computational Group recommends discussions with GoldSim to see if this memory limitation can be overcome in changes implemented in a future GoldSim release.

As described in Section 6, new elements must be added to the GoldSim model file to export sampled inputs to be analyzed with the outputs of interest using sensitivity analysis techniques. It is anticipated that future uses of xLPR V2.0 will necessarily include sensitivity analyses, and the need to make code modifications prior to performing sensitivity analyses represents a limitation of the Framework. In addition, the best implementation of sensitivity analysis outputs uses GoldSim spreadsheet elements that write the sampled inputs to a Microsoft Excel file. This prohibits the use of parallel processing during large simulations, making these simulations computationally expensive due to both linear processing and file writing time. Accordingly, the Computational Group recommends the development of a solution to save and export the values of sampled outputs. This solution should involve discussions with GoldSim to determine if values could be saved efficiently during parallel processing.

Finally, the GoldSim architecture and the inputs interface prohibited V&V testing from being fully automated. Input files were instead defined for each test and tests were run using many copies of the GoldSim model file. This produced many files that needed to be saved for traceability purposes and limited the ability for tests to be quickly rerun. More information regarding this limitation is described in the Software Verification and Validation Report (SVVR) [46]. The Computational Group recommends development of automated regression testing using the input files developed during the V&V testing for use in maintenance of the Code.

8. SUMMARY

xLPR V2.0 was developed to perform robust analyses of pipe rupture probabilities and other event-specific probabilities relevant to leak-before-break problems. To this end, the Framework combines a system of rigorously vetted modules that have been independently verified and validated into a cohesive model capable of estimating crack, leak, and rupture probabilities. The complex structure of the Framework passes information between modules and accomplishes all the checking, preprocessing, sampling, calculating, and post-processing steps necessary to model the entire problem from crack initiation to crack growth to leaks and ruptures. Each component of the Framework was developed and implemented under a rigorous quality assurance program and the modules and the Framework were tested extensively.

Verification testing of the Framework was split into Unit and Integration tests consisting of both static and dynamic tests. The verification testing effort found that the Framework is structured and performs according to its requirements and design documentation, although two minor errors that do not impact the overall results were discovered and have been documented. Validation testing of the Framework, also termed Acceptance Testing, was performed to validate the overall modeling approach for generic applications and to validate the output of probability of rupture, primarily for leak-before-break applications. Three types of validation tests were completed to accomplish these goals: service tests, module behavior tests, and benchmark tests. The conclusion from Acceptance Testing is that the Framework is considered suitable for its intended use and ready for the applications and maintenance phase.

In addition, the readiness of xLPR V2.0 was further vetted through the Scenario Analysis effort after completion of the Verification and Validation testing. This effort, scoped to provide guidance on sampling options for specific scenarios of interest, also exercised the various options, components, and sampling schemes available in xLPR V2.0 and developed and exercised methods for analyzing and interpreting the Code's outputs. As a demonstration of the Code's readiness, the scenario analyses succeeded in applying the Code extensively in a manner consistent with its intended use. The results obtained were consistent with expectations, further supporting the conclusion that xLPR V2.0 is ready for applications.

REFERENCES

[1] NUREG-2110, *xLPR Pilot Study Report*, U.S. Nuclear Regulatory Commission, Washington, DC, May 2012.

[2] MRP-135. Materials Reliability Program: xLPR Pilot Study Report. EPRI, Palo Alto, CA, and U.S. NRC, Washington, DC: 2011.

[3] Rudland, D., Harrington, C., Dingreville, R., Development of the extremely low probability of rupture (xLPR) Version 2.0 code, PVP2015-45134, Proceedings of the 2015 ASME Pressure Vessels and Piping Conference, 2015.

[4] GoldSim 11.1.5, GoldSim Technology Group LLC, 1998-2016.

[5] xLPR-SQAP, xLPR Software Quality Assurance Plan, Version 5.0, September 2014.

[6] xLPR-SCMP, xLPR Software Configuration Management Plan, Version 4.0, February 2014.

[7] xLPR-SRD-FW, xLPR Software Requirements Description for xLPR Framework, Version 4.0, June 2016.

[8] xLPR-SDD-FW, xLPR Software Design Description for xLPR Framework, Version 3.0, August 2016.

[9] xLPR-STP-FW-Integration, xLPR Software Test Plan for xLPR Framework Integration Testing, Version 3.0, August 2016.

[10] xLPR-STP-FW-Unit, xLPR Software Test Plan for xLPR Framework Unit Testing, Version 3.0, August 2016.

[11] xLPR-STP-FW-Acceptance, xLPR Software Test Plan for Framework Acceptance, Version 2.0, September 2016.

[12] xLPR-TR-Stress, xLPR Version 2.0 Stress Requirements, Version 1.1, March 2016.

[13] Mariner, P.E. 2017. xLPR Sim Editor 1.0 User's Guide. SAND2017-3166. Sandia National Laboratories, Albuquerque, NM.

[14] xLPR-UM V1.1, xLPR User Manual for xLPR V2.0 Framework and Modules, Version 1.1, December 2017.

[15] xLPR-TR-CI-SCC-Calibration, PWSCC Initiation Model Parameter Development, Confirmatory Analyses, and Validation for xLPR Version 2.0, Version 2.0, July 2017.

[16] GoldSim User's Guide, Version 11.1; GoldSim Technology Group: Issaquah, WA, May 2014.

[17] Eckert-Gallup, A., Lewis, J., Martin, N., Hund, L., Clark, A., Brooks, D., Mariner, P., xLPR Scenario Analysis Report, Sandia Internal Report, SAND2017-2854, March 2017.

[18] Ramberg, W., Osgood, W. R. Description of stress–strain curves by three parameters. Technical Note No. 902, National Advisory Committee For Aeronautics, Washington DC, 1943.

[19] xLPR-SDD-LRM, xLPR Software Design Description for a Leakage Rate Module, Version 3.2, November 2016.

[20] xLPR-SRD-CTM, xLPR Software Requirements Description for the Crack Transition Module, Version 2.2, June 2016.

[21] xLPR-SDD-CI-SCC, xLPR Software Design Description for PWSCC Crack Initiation, Version 3.1, March 2014.

[22] xLPR-SDD-CIF, xLPR Software Design Description for Fatigue Crack Initiation, Version 2.0, March 2014.

[23] xLPR-SRD-CI-SCC, xLPR Software Requirements Description for Crack Initiation – PWSCC, Version 3.4, December 2015.

[24] xLPR-SRD-CIF, xLPR Software Requirements Description for Crack Initiation – Fatigue, Version 3.3, June 2015.

[25] xLPR-SDD-Coalescence, xLPR Software Design Description for the Crack Coalescence Module, Version 3.0, April 2016.

[26] Dingreville, R., Eckert-Gallup, A., Sallaberry, C., Uncertainty analysis for the net-section-collapse failure criterion of circumferentially cracked cylinders for multiple arbitrary-shaped circumferential cracks, International Journal of Pressure Vessels, 2014. and Piping 123, 30-45, 2014.

[27] xLPR-SDD-CrCOD, xLPR Software Design Description for a Circumferential Through-Wall Crack Combined Tension and Bending COD Module, Version 3.1, January 2016.

[28] xLPR-SDD-AxCOD, xLPR Software Design Description for Axial Through-Wall Crack Opening Displacement Module, Version 1.2, January 2017.

[29] xLPR-MSGR-COD, Summary of the xLPR Version 2.0 Crack Opening Displacement (COD) Modules, Version 1.0, April 2016.

[30] xLPR-MSGR-CI, PWSCC and Fatigue Crack Initiation Module Development, Version 1.0, August 2016.

[31] xLPR-MSGR-CGR, PWSCC and Fatigue Crack Growth and Coalescence Module Development, Version 1.0, June 2016.

[32] xLPR-MSGR-ISI, In-Service Inspection (ISI) Module Development, Version 1.0, June 2016.

[33] xLPR-MSGR-KSol, Surface and Through-Wall Crack Stress Intensity Factor Module Development, Version 1.0, June 2016.

[34] xLPR-MSGR-LRM, Leak Rate Module Development (LEAPOR), Version 1.0, June 2016.

[35] xLPR-MSGR-TIFF, Cyclic Stress Intensity Factors Due to Operating Transients Module Development (TIFFANY), Version 1.0, January 2016.

[36] xLPR-MSGR-Stability, Axial and Circumferential Crack Stability Module Development, Version 1.0, June 2016.

[37] xLPR-MSGR-CTM, Surface-to-Through-Wall Crack Transition Module Development, Version 1.0, April 2016.

[38] xLPR-STRR-FW-Unit, xLPR Software Test Results Report for xLPR Framework Unit Testing, Version 1.0, July 2016.

[39] xLPR-STRR-FW-Integration, xLPR Software Test Results Report for xLPR Framework Integration Testing, Version 1.0, July 2016.

[40] xLPR-SVVP, xLPR Software Verification and Validation Plan, Version 2.0, May 2015.

[41] PROMETHEUS, The Theory, Verification, and Benchmarking of PROMETHEUS, December 2015.

[42] Harris, D.O., Dedhia, D., WinPRAISE 07, Expanded PRAISE Code in Windows, January 2007.

[43] xLPR-STRR-FW-Acceptance, xLPR Software Test Results Report for Framework Acceptance, Version 1.0, September 2016.

[44] xLPR-QAA, xLPR Version 2.0 Quality Assurance Approach, Version 1.0, April 2012.

[45] xLPR-GR-IG, xLPR Inputs Group Report, Version 1.0, December 2017.

[46] xLPR-SVVR, xLPR Software Verification and Validation Report, Version 1.0, March 2018.