xLPR Version 1.0 Report

Technical Basis and Pilot Study Problem Results

xLPR Computational Group

February 2011

U.S. Nuclear Regulatory Commission
Office of Nuclear Regulatory Research
Washington, DC 20555-0001

U.S. NRC-RES Project Manager
David L. Rudland

Electric Power Research Institute
3420 Hillview Avenue
Palo Alto, CA 94303

EPRI Project Manager
Craig Harington

Executive Summary

10 CFR Part 50, Appendix A, General Design Criteria (GDC) 4 states, in part, that the dynamic effects associated with postulated reactor coolant 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. The NRC Standard Review Plan (SRP) 3.6.3 describes Leak-Before-Break (LBB) deterministic assessment procedures that have been used to date demonstrate compliance with the GDC-4 requirement. Currently, SRP 3.6.3 does not allow for assessment of piping systems with active degradation mechanisms, such as Primary Water Stress Corrosion Cracking (PWSCC), which is currently occurring in systems that have been granted LBB exemptions. Even though the piping systems experiencing PWSCC have been shown to be compliant with the regulations through qualitative arguments, no tool currently exists to quantitatively assess compliance to this criterion.

From the NRC staff perspective, a long term goal is to develop a modular based, probabilistic fracture mechanics tool capable of determining the probability of failure for Reactor Coolant System (RCS) components. The need for this modular-based code is strongly driven by the need to quantitatively assess the LBB-approved piping system's compliance with GDC-4 on an interval (time) basis. To meet this need, the NRC Office of Nuclear Regulatory Research (RES) has entered into a cooperative program with the Electric Power Research Institute (EPRI) though an addendum to the Memorandum of Understanding to define, design and develop this code, which will be 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 such as to accommodate evolving and improving knowledge. The code is structured in a modular fashion so that as additional situations arise, additions or modifications can be easily incorporated without code restructuring. Based on the terminology of GDC-4, this program and code is entitled Extremely Low Probability of Rupture (xLPR).

Under the MOU addendum, a project management organizational structure with balanced NRC/industry representation was developed consisting of four topical technical task groups coordinated by an overarching Project Integration Board (PIB). A diverse team of experts in the various technical specialties reflected in this complex analytical project was assembled under funding from both the NRC and EPRI to develop the detailed analytical methodologies, identify the necessary input data and mathematical models, and assemble them into a functioning set of computational tools. The technical task groups and highlights of their areas of responsibility are:

- *Computational*: integration of the computational elements (models) into a robust, fully developed, tested, and verified computational tool.
- *Models*: selection, documentation, and coding of the mathematical model building blocks.
- *Inputs*: identification, collection, and presentation of the data required for the models and the sample problem as input tables and distributions.
- *Acceptance Criteria*: formulation of probabilistic acceptance criteria for assessing the code results.

The development of a sophisticated probabilistic software tool that meets quality assurance (QA) and technical requirements is a technically and programmatically challenging task. The management structure, the probabilistic framework, and data handling are just a few of the issues that need to be addressed early on in the software development effort. In order to meet this need, a pilot study was conducted. This pilot study is a proof-of-concept effort to develop an initial assessment tool for dissimilar metal (DM) pressurizer surge nozzle welds, for which a considerable amount of publicly available information exists. The xLPR pilot study objective is to demonstrate the feasibility of the proposed developmental process, framework, and model for a probabilistic code to address degradation mechanisms in piping system safety assessments and to determine the computational framework structure that is the most appropriate to meet the longer term program goal to develop a modular based, probabilistic fracture mechanics tool capable of determining the probability of failure for RCS components. This report documents the technical basis for the pilot study, describes the xLPR Version 1.0 code, and demonstrates its capabilities through a series of example problems. A separate report documents the final results and accomplishments of the xLPR pilot study.

To assess the capabilities of computational framework architecture, two unique framework codes were developed in the pilot study to investigate the advantages and disadvantages of the different approaches used by each code. GoldSim was used to develop a commercial software version of the xLPR Model, and an open source software version, SIAM-PFM, was also developed.

As part of the pilot study, the two xLPR codes were developed following a strict configuration management (CM) structure. The xLPR CM structure consists of a systematic approach applied to both the developed software and models to ensure some of the basic fundamentals of a QA program are met, including: 1) Access Control; 2) Version Control; 3) Verification (e.g., Checking); and 4) Traceability (e.g., Documentation). The xLPR CM approach included documentation of each step in the process. The CM process was implemented as detailed in a series of Guidance Documents which outline the specific steps for each of four key components of the xLPR pilot program: 1) Module Development; 2) Framework Development; 3) Model Parameters and Inputs for the pilot study test case; and 4) xLPR Model Production Runs and Uncertainty/Sensitivity Analyses for the pilot study test case. Even though a strict CM plan was followed, that CM plan was not linked to a formalized QA process, which will occur in later versions of the code.

The flow of the xLPR Version 1.0 code is centered on a time-based history of events where PWSCC initiates flaws that grow until failure. The technical basis for this behavior was developed within the Models group by subject matter experts, who, together with the Computational group members, coded, compiled, and verified, using the pre-defined CM process, the modules needed for this purpose. These modules included loads with weld residual stress, crack initiation, crack growth, crack coalescence, crack stability, crack opening displacement, leakage, inspection and mitigation. Both the commercial and open source applications utilized these self-contained modules which were linked together to create two xLPR framework codes that control the time flow of the analyses and properly account for and propagate the problem uncertainties.

The xLPR framework for calculating the probability of primary system pipe rupture evokes a systematic approach to uncertainty characterization and the propagation of probability distributions. In order to better understand the effects of uncertainty on the distribution of the output parameters desired, the uncertainty is classified as aleatory (random or irreducible) or epistemic (lack of knowledge or reducible). The uncertainty in the input is propagated through the model using sampling-based methods. The appropriate way to propagate uncertainty is ultimately dependent on the computational constraints as well as the nature of the inputs and outputs under consideration. Within the xLPR pilot study, several sampling methods were evaluated, such as simple random sampling, Latin hypercube sampling, and discrete probability distribution space sampling. In addition, importance sampling was used in order to estimate the low probability events.

A sample problem statement was developed to demonstrate the feasibility of conducting analyses to assess the probability of rupture in pressurizer surge nozzle DM welds, and to compare the results from the two frameworks developed in this effort. This problem statement consisted of two deterministic analyses, a probabilistic base case, and a series of sensitivity analyses to demonstrate the features of the Version 1.0 code as well as to provide a proof of concept for the overall xLPR methodology. These runs focused on the demonstration of the calculation of probability of rupture with and without mitigation, inspection and leak detection, while accounting for epistemic and aleatory uncertainties.

Not unexpectedly, the probabilistic base case, which contained high weld residual stress with no inspection, mitigation or leak detection, produced relatively high mean probabilities of rupture using both the Goldsim and SIAM frameworks. In fact, the two codes gave approximately the same values, with the difference being attributed to the handling of the crack initiation model. Further analyses suggest that the calculated results are stable and highly driven by the epistemic uncertainty. Due to the high crack growth rate and assumptions made on the crack behavior, the benefit of leak detection and inspection was not as large as expected. Parameter sensitivity studies conducted on the base case suggested that the weld residual stress and the crack initiation parameters (all characterized as epistemic) were controlling the uncertainty in the probability of rupture.

The sensitivity studies conducted investigated the effects of stress mitigation, chemical mitigation, crack initiation model, and weld residual stress on the probability of rupture as compared to the base case. The conclusions from these sensitivity studies indicate:

- For stress mitigation, the effects are seen in both the crack initiation and growth models. The mitigation was applied at 10, 20 or 40 years. The results from these analyses demonstrate that the effective application of the assumed stress-based mitigation could cause the probability of rupture to no longer increase with operating time after its application.
- Two chemical mitigation cases changed the hydrogen content in the water from the base case value of 25cc/kg to 50cc/kg and 80cc/kg. As expected additional hydrogen decreases the probability of rupture at sixty years by about 50% for 50cc/kg, but has little additional affect for 80 cc/kg.
- Leak detection over the range of 1 to 10 GPM reduced the rupture probability at 60 years by a factor of ten.
- In-service inspection (ISI) every 2 years reduced the 60-year rupture probability by a factor of 70, while ISI every 10 years only reduced it by a factor of 2.
- Changing the crack initiation model had very little impact on the overall rupture probability. This was expected since each of the models is empirically based and calibrated to the same service history.
- Changing the crack initiation parameters from epistemic to aleatory had a large effect on the probability of rupture distribution. While the mean value stays the same, when the uncertainty in initiation time is characterized as epistemic, 50% of the time there is a 0% chance of any future rupture. When it’s characterized as aleatory, 50% of the time there is a 35% chance of any future rupture.
- Changing the weld residual stress had a large impact on the rupture probability. Changing the residual stress from a surge nozzle geometry without a safe end weld to

one with a safe end weld decreased the rupture probabilities at 60 years by two orders of magnitude.

In many of the cases with low residual stress and/or inspection and leak detection, the number of samples taken was not sufficient to produce stable results. Therefore, importance sampling was needed. In all cases considered, the weld residual stress and an initiation parameter (B1) were importance sampled. The results indicate that with inspection and leak detection, probabilities down to 10^{-6} at 60 years can be calculated with reasonable confidence. However, when mitigation is added, probabilities down to 10^{-9} at 60 years are calculated, but the confidence in the mean values is very poor. Additional realizations (predictions of rupture) are required to increase the confidence in these results.

Over the course of this investigation, the xLPR project team developed an appreciation for the complexity of this problem, and the structure needed for successful completion of a comprehensive PFM code. Through the process, many important lessons were learned. These lessons include not only technical lessons from the module and framework development and implementation, but also from the organization and program management viewpoint, including:

- Three very important organizational structure aspects are required for program success:
    - Dedicated team members, whose qualifications cover the important aspects of the group responsibility. It is advantageous if team members understand the basic computational process to aid in the incorporation of elements into the overall model.
    - An enthusiastic team and group leadership.
    - An efficient communication process within and among the teams.
- Configuration management is only a small portion of quality assurance. Establishing a program QA plan and controls for xLPR is the essential first step in the continuing development process. Each organization involved in the development of the code needs to develop a plan that identifies the process used to comply with the program QA requirements. The xLPR program needs to have a transparent and traceable CM system that will cover the xLPR code lifecycle.
- A well-written, unambiguous software requirements document needs to be developed and followed for future xLPR versions.
- From the models standpoint, certain assumptions were made due to the limited scope of the pilot study. Some models-based limitations and lessons learned include:
    - Manufacturing defects and fatigue initiation and growth were ignored in the pilot study. Both should be included in future version since their omission may lead to non-conservative rupture probabilities.
    - The load module needs to be updated to include a more realistic weld residual stress model and transient definitions.
    - Considering only circumferential cracks may over-predict the rupture probabilities. The addition of axial cracks may reduce the rupture probabilities due to their higher leakage probabilities which would lead to early repair / mitigation.
    - Assuming idealized flaw shapes and simplistic transitions from a surface crack to a through-wall crack may cause an overestimate of the leak rate.
    - More realistic surface crack stability, inspection and mitigation models are required for making best-estimate predictions of their effects.
- Focus should be placed on efficient data storage, data handling and post-processing to improve the running of the code.

- Importance sampling is necessary for the calculation of the probability of rupture in piping systems. Emphasis needs to be placed on processes and procedures for identifying the variables that need to be importance sampled. Adaptive sampling or other reliability methods need to be considered.
- The classification of uncertainty is very important to understanding the overall uncertainty in the probability of rupture. Uncertainty is very important and needs involvement at all levels of development of a complex system. Knowing which variables control the rupture and what part of the uncertainty in those variables is epistemic and can be reduced will not only inform the regulators, but will also help direct future research in this area. In other words, xLPR can be used to prioritize research efforts and degradation management strategies to quantitatively improve safety.

Finally, the complete xLPR pilot study effort, which includes not only the code development efforts, but the management structure, the pilot statement problem, and the detailed analysis of the results demonstrate that **it is feasible** to develop a modular-based computer code for the determination of probability of rupture for LBB approved piping systems.

Forward

The present document is one of a series of reports summarizing the results of the NRC xLPR pilot study project. The complete list of report titles and the sponsoring organizations are as follows:

xLPR Pilot Study Final Report: Project Summary, Outcomes, and Recommendations	NRC NUREG and EPRI
Assessment of Capabilities of Extremely Low Probability of Rupture (xLPR) Software – GoldSim and SIAM Version 1.0	Nuclear Regulatory Center for Nuclear Waste Regulatory Analysis (CNWRA)
xLPR Version 1.0 Report, Technical Basis and Pilot Study Problem Results	xLPR Computational Group
xLPR Framework (GoldSim) Model User's Guide	Sandia National Laboratory
Structural Integrity Assessments Modular-Probabilistic Fracture Mechanics (SIAM-PFM): User's Guide for xLPR	Oak Ridge National Laboratory
Development, Analysis, and Evaluation of a Commercial Software Framework for the Study of Extremely Low Probability of Rupture (xLPR) Events at Nuclear Power Plants	Sandia National Laboratory
SIAM-xLPR Version 1.0 Framework Report	Oak Ridge National Laboratory
Materials Reliability Program: Models and Inputs Developed for Use in the xLPR Pilot Study (MRP-302)	xLPR Models and Input Groups

Acknowledgements

The development of the Version 1.0 xLPR code was a group effort that spanned a variety of experts across many fields of expertise from the U.S. NRC, EPRI, and their contractors. The success of the program emphasizes the dedication of the xLPR team, the strength of its leadership, and the generous support from both the NRC and EPRI. There are many people to thank, including members from the computational, models, inputs and acceptance groups, as well as the Program Integration Board (PIB). Every person on this team provided valuable contributions, and their efforts were sincerely appreciated.

Computational Group
David Rudland – U.S. NRC
Bruce Bishop – Westinghouse
Nathan Palm – Westinghouse
Patrick Mattie – Sandia National Laboratories
Cedric Sallaberry – Sandia National Laboratories
Don Kalinich – Sandia National Laboratories
Jon Helton – Sandia National Laboratories
Hilda Klasky – Oak Ridge National Laboratory
Paul Williams – Oak Ridge National Laboratory
Robert Kurth – Emc^2
Scott Sanborn – Pacific Northwest National Laboratory
David Harris – Structural Integrity Associates
Dilip Dedhia – Structural Integrity Associates
Anitha Gubbi – Structural Integrity Associates

Inputs Group
Eric Focht – U.S. NRC
Mark Kirk – U.S. NRC
Guy DeBoo – Exelon
Paul Scott – Battelle
Ashok Nana – AREVA NP Inc.
John Broussard – Dominion Engineering
Nathan Palm – Westinghouse
Pat Heasler – Pacific Northwest National Laboratory
Gery Wilkowski – Emc^2

Acceptance Group
Mark Kirk – U.S. NRC
Glenn White – Dominion Engineering Inc.
Aladar Csontos – U.S. NRC
Robert Hardies – U.S. NRC
David Rudland – U.S. NRC
Bruce Bishop – Westinghouse
Robert Tregoning – U.S. NRC

Models Group
Marjorie Erickson – PEAI
Gary Stevens – U.S. NRC
Howard Rathbun – U.S. NRC
David Rudland – U.S. NRC
John Broussard – Dominion Engineering
Glenn White – Dominion Engineering
Do-Jun Shim – Emc^2
Gery Wilkowski – Emc^2
Bud Brust – Emc^2
Cliff Lange – Structural Integrity Associates
Dave Harris – Structural Integrity Associates
Steve Fyfitch – AREVA NP Inc.
Ashok Nana – AREVA NP Inc.
Rick Olson – Battelle
Darrell Paul – Battelle
Lee Fredette – Battelle
Craig Harrington – EPRI
Gabriel Ilevbare – EPRI
Frank Ammirato – EPRI
Patrick Heasler – Pacific Northwest National Laboratory
Bruce Bishop - Westinghouse

Program Integration Board
Craig Harrington – EPRI
Aladar Csontos – U.S. NRC
Robert Hardies – U.S. NRC
Denny Weakland - Ironwood Consulting
David Rudland – U.S. NRC
Bruce Bishop – Westinghouse
Eric Focht – U.S. NRC
Guy DeBoo – Exelon
Marjorie Erickson – PEAI
Gary Stevens – U.S. NRC
Howard Rathbun – U.S. NRC
Mark Kirk – U.S. NRC
Glenn White – Dominion Engineering Inc.

Nomenclature

Symbols

N_E	Number of epistemic realizations
N_A	Number of aleatory realizations
A	Cross-sectional area of the pipe
a	Crack depth
$\dot{a}$	Crack growth rate
α	Crack growth power-law coefficient
B_1	Crack initiation parameter, heat-to-heat material variability
β	Stress intensity factor exponent
β_1	POD model coefficient
β_2	POD model coefficient
c	Half crack length
c_w	Characteristic width of crack growth rate peak versus electrochemical potential
D_o	Outer diameter
D_i	Inner diameter
d_1	First crack depth for coalescence
d_2	Second crack depth for coalescence
$\Delta ECP_{Ni/NiO}$	Electrochemical potential difference at the current hydrogen concentration to the Ni/NiO transition at temperature
$f_{cracked}$	Crack area divided by pipe cross-sectional area.
F_{DW}	Dead weight axial load
F_{NTE}	Normal thermal (including stratification) axial load
F_P	Axial load due to pressure
F_x	Axial load
f_{weld}	Weld factor
f_{ww}	Within weld factor
ϕ	Azimuthal location of crack
G_n	Influence function coefficient
H_2	Hydrogen concentration
I	Moment of inertia
J	J-integral fracture mechanics parameter
K	Stress intensity factor
K_{Ith}	Crack-tip stress intensity factor threshold
ℓ	Segment length
M_x	Moment about the x-direction
M_y	Moment about the y-direction
M_z	Moment about the z-direction
M_{eff}	Effective moment
P	Pressure
P_R	Peak-to-valley ratio for effect of electrochemical potential on crack growth rate
Q_g	Thermal activation energy for crack growth
R	Universal gas constant
R_i	Inner Radius
R_o	Outer Radius
R^2	Coefficient of determination

s	Distance between cracks for coalescence
σ_B	Maximum bending stress
$\sigma_{B\text{-}L}$	Location specific bending stress
σ_{CS}	Total stress used for crack stability
σ_{oDW}	Axial stress due to dead weight
σ_{oNTE}	Axial stress due to normal thermal expansion
σ_{oP}	Axial stress due to pressure
σ_{WRS}	Weld residual stress
σ_{nWRS}	Weld residual stress curve fit coefficients
t	Wall thickness
T	Temperature
T_{abs}	Absolute temperature at location of crack
T_{ref}	Absolute reference temperature
x	Distance from ID
X_c	Distance from ID weld residual stress crosses zero

Acronyms

CCDF	Complementary cumulative distribution function
CDF	Cumulative distribution function
CM	Configuration management
CNWRA	Nuclear Regulatory Center for Nuclear Waste Regulatory Analysis
COA	Crack Opening Area
COD	Crack Opening Displacement
DLL	Dynamically linked library
DM	Dissimilar metal
DPD	Discrete Probability Distribution
DW	Dead weight
EMC^2	Engineering Mechanics Corporation of Columbus
EPRI	Electric Power Research Institute
GDC	General Design Criteria
GE	General Electric
GPM	Gallons per minute
GRC	Golden rule clustering
GUI	Graphical User Interface
ID	Inner diameter
IDE	Integrated Development Environment
ISI	In-service Inspection
LBB	Leak-Before-Break
LHS	Latin Hypercube Sampling
LOCA	Loss of coolant accident
MC	Monte Carlo
MOU	Memorandum of Understanding
MRP	Materials Reliability Program
MTS	Number of Intervals in the Operating History
NPS	Nominal Pipe Size
NRC	U. S. Nuclear Regulatory Commission
NTE	Normal thermal expansion
NURBIM	Nuclear Risk Based Inspection Methodology
OD	Outer diameter

OOOS	Object Oriented Open Source
ORNL	Oak Ridge National Laboratory
PEAI	Phoenix Engineering Associates, Inc.
PDF	Probability density function
PIB	Project integration board
PND	Probability of non-detection
PNNL	Pacific Northwest National Laboratory
POD	Probability of detection
PRA	Probabilistic Risk Assessment
PWR	Pressurized water reactor
PWSCC	Primary water stress-corrosion cracking
QA	Quality assurance
RES	Office of Nuclear Regulatory Research
RCS	Reactor coolant system
SC	Surface crack
SCM	Software Configuration Management
SI	International system of units
SIA	Structural Integrity Associates
SIAM	Structural Integrity Assessment Modular
SIAM-PFM	Structural Integrity Assessment Modular – Probabilistic Fracture Mechanics
SNL	Sandia National Laboratories
SQUIRT	Seepage Quantification of Upsets in Reactor Tubes
SRP	Standard Review Plan
SRRC	Standardized Rank Regression Coefficient
SSE	Safe Shutdown Earthquake
STP	Standard temperature and pressure
TWC	Through-wall crack
V&V	Verification and validation
WRS	Weld residual stress
xLPR	Extremely low probability of rupture

Table of Contents

1 Introduction

1.1 Background

10 CFR Part 50, Appendix A, General Design Criteria (GDC) 4 states, in part, that the dynamic effects associated with postulated Reactor Coolant System (RCS) 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. Licensees have typically demonstrated compliance with this probabilistic criterion through deterministic and highly conservative analyses. The NRC Standard Review Plan (SRP) 3.6.3 describes Leak-Before-Break (LBB) deterministic assessment procedures that have been used to demonstrate compliance with the GDC-4 requirement. Currently, SRP 3.6.3 does not allow for assessment of piping systems with active degradation mechanisms, such as Primary Water Stress Corrosion Cracking (PWSCC), which is currently occurring in systems that have been granted LBB exemptions. Even though the piping systems experiencing PWSCC have been shown to be compliant with the regulations through qualitative arguments, no tool currently exists to quantitatively assess this compliance.

Given recent advances in probabilistic methodologies, the NRC staff and industry believe that performing a probabilistic analysis of primary system piping that fully addresses and quantifies uncertainties and directly demonstrates compliance with GDC 4 is more appropriate. The NRC and industry expect that a robust probabilistic software tool, developed cooperatively, will facilitate meeting this goal, and result in improvement in licensing, regulatory decision-making and design, and will be mutually beneficial. Based on the terminology of GDC 4, this project is entitled Extremely Low Probability of Rupture (xLPR).

1.2 Program structure

From the NRC staff perspective, a long term goal is to develop a modular based, probabilistic fracture mechanics tool capable of determining the probability of leakage and rupture for RCS components. These systems include not only primary piping, but also pressure vessels and steam generators. This computer code will eventually be capable of considering all known degradation mechanisms that may contribute to failure events, while properly handling the uncertainty in the failure process. It will be comprehensive with respect to all known and significant 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 in order to accommodate evolving and improving knowledge. The code will be structured in a modular fashion so that as additional sources of change arise, additions or modifications can be easily incorporated without code restructuring.

The need for this modular-based code is strongly driven by the need to quantitatively assess the LBB-approved piping systems previously shown to be compliant with GDC-4 on regular intervals (time). Therefore, the initial focus is on piping systems susceptible to PWSCC. The development of a sophisticated probabilistic software tool that meets quality assurance (QA) and technical requirements is a technically and programmatically challenging task. In particular, the management structure, probabilistic framework, and data handling are just a few of the issues that need to be addressed early in the software development effort. In order to meet these needs, a pilot study was conducted. This study demonstrates the initial feasibility of the proposed process for developing xLPR to address PWSCC degradation in piping systems.

Following the pilot study, a more detailed, long term study will be completed to generalize the analysis procedures to all primary system piping. The long-term study will employ the same basic organizational, management, and NRC-industry cooperative structure as the pilot study. Technical and programmatic lessons learned in the pilot study will be incorporated into the long-term study. Technical issues from the pilot study left unresolved due to their complexity will also be addressed in the long-term study.

1.3 Cooperative program

The xLPR project is a cooperative effort between industry (represented by the Electric Power Research Institute (EPRI)) and the NRC Office of Nuclear Regulatory Research (RES). Because of this cooperation, a degree of organizational structure and operational rigor was needed. In response to this need, an xLPR Addendum to the ongoing Memorandum of Understanding (MOU) between EPRI and the NRC was implemented to define and guide the project specific interactions between the two parties [1].

Under the MOU addendum, a project management organizational structure with balanced NRC/industry representation was developed consisting of four topical technical task groups coordinated by an overarching Project Integration Board (PIB). A diverse team of experts in the various technical specialties reflected in this complex analytical project was assembled under funding from both the NRC and EPRI to develop the detailed analytical methodologies, identify the necessary input data and mathematical models, and assemble them into a functioning set of computational tools. The technical task groups and highlights of their areas of responsibility are:

- *Computational*: integration of the computational elements (models) into a robust, fully developed, tested, and verified computational tool.
- *Models*: selection, documentation, and coding of the mathematical model building blocks.
- *Inputs*: identification, collection, and presentation of the data required for the models and the sample problem as input tables and distributions.
- *Acceptance Criteria*: formulation of probabilistic acceptance criteria for assessing the code results.

The interaction between these groups is shown in Figure 1. In addition, an Independent Review Board will periodically review various aspects of the xLPR Project, provide new perspectives, and ensure that concerns of all constituents are addressed in a robust, scientifically-balanced, and technically sound manner.

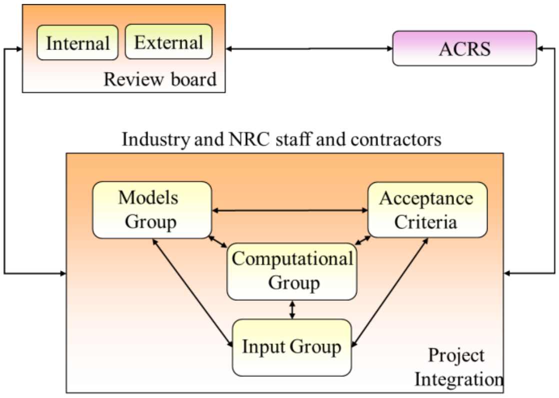


Figure 1 xLPR Group Interaction

1.4 Computational framework

In developing the computational implementation for the xLPR Version 1.0 code, the choice of the appropriate computational framework is essential to assure that the code can be developed in a modular fashion and still be able to provide the structure for uncertainty handling. Currently, there are many commercial probabilistic computer codes that use Monte Carlo (MC) simulation for dynamically modeling complex systems. These codes have been developed to support decision and risk analysis by simulating future performance, while quantitatively representing the uncertainty and risks inherent in all complex systems. Commercial software also has desirable advantages including:

- Verified and Validated, Off-the-Shelf Availability
- Tested and debugged, i.e., vetted
- Technical Support and Troubleshooting
- Life Cycle Maintenance (updates, bug fixes, service packs)
- Version Control and Software Configuration Management
- Documentation and User's Guides

However, as with all commercial software, licensing fees[1] may be required, and the development of the code may be limited by the capabilities available in the software. A commercial software evaluation must consider the flexibility of the software using the native capabilities and the ability of the vendor to respond to requests, if necessary, for software enhancements.

On the other hand, many of the routines needed to develop the xLPR structure exist in open-source literature. Therefore, it seems reasonable to investigate the ability to custom-develop the framework using available source code. The main advantage of this method is that all of the code is available for review and editing without commercial licensing fees. In addition, the ability to develop a flexible code is not limited to the native capabilities of the commercial software package, or the ability of the commercial software vendor to respond to modification requests, but can be written specifically tailored for the problem at hand. However, the disadvantage is that the development of such a complex code from scratch may be both time and cost prohibitive, as many of the advantages of commercial software (e.g., software

[1] In some cases, once the development of the probabilistic framework is complete, the code can be run through a player file, thus eliminating the need for licensing fees for the user of the software.

debugging, verification, SCM, life cycle maintenance, documentation, etc.) need to also be developed.

Therefore, within this initial xLPR pilot study, two unique framework codes were implemented to investigate the advantages and disadvantages of two approaches: 1) use of available commercial software, and 2) use of open source code. The commercial software, GoldSim, was used to investigate the commercial software approach of the xLPR Model, while the open source code, SIAM-PFM, using available open source code was investigated to demonstrate the open source approach. The details of each framework code are provided in this report.

1.5 xLPR Version 1.0 report structure

The purpose of this report is to document the details of the first version (Version 1.0) of the xLPR code and discuss the pilot study results. This report is intended to be a summary report, with many of the details being compiled from the following reports:

- GSxLPR v1.0 Framework Report [2] – This report, written by Sandia National Laboratory, describes the configuration management structure developed and implemented for the pilot study, the development and details of the GoldSim xLPR Version 1.0 framework, a discussion of the probabilistic methodology, development of the post processing tools, results of the pilot study analyses, and a parameter uncertainty and sensitivity analysis using the GSxLPR v1.0 framework.
- SIAMxLPR v1.0 Framework Report [3] – This report, written by Oak Ridge National Laboratory, describes the development and details of the SIAM xLPR Version 1.0 framework and results of the pilot study analyses using the SIAMxLPRv1.0 framework.
- xLPR v.1.0 Models/Inputs Report [4] – The report, written by the Models and Inputs Groups, details the analytical models used in Version 1.0 and the rationale for choosing those models.

Section 2 of this report provides a description of the pilot study, including the scope, the problem statement, and the assumptions in this effort. Section 3 discusses the configuration management structure developed and implemented in the pilot study effort. Section 4 describes the xLPR computational process, including the uncertainty handling, the time-loop structure, the sampling schemes, and the outputs. Section 5 provides a brief description of the models and inputs used to solve the pilot study problem. Section 6 gives an overview of the framework architecture for each of the two evaluated platforms. Section 7 describes the pilot study problem statement, while Section 8 presents the results from these analyses. Finally, Section 9 describes some of the lessons learned from the pilot study, and Section 10 provides an overall project summary.

In addition to the main body of the report, four appendices provide additional information on the xLPR Version code. Appendix A contains detailed information on the random number sampling techniques used in the pilot study. Appendix B contains the pilot study problem statement. Appendix C contains the pilot study, base case inputs. Appendix D contains an impact assessment that addresses errors found in the code after the pilot study problems were completed and fully documented.

2 Pilot Study Description

2.1 Program scope

The development of a sophisticated probabilistic software tool that meets quality assurance (QA) and technical requirements is a technically and programmatically challenging task. The management structure, the probabilistic framework, and data handling are just a few of the issues that need to be addressed early in the software development effort. In order to meet this need, a pilot study was conducted. This study was used to demonstrate the feasibility of the proposed NRC-industry cooperative process for developing xLPR to address degradation in piping systems. It was also used to help in the selection of an appropriate probabilistic framework for calculating relative, order-of-magnitude estimates of the probability of rupture for a pressurizer surge nozzle dissimilar metal (DM) weld. These analyses and results are not meant to be absolute, and can be used to identify areas requiring more focused attention in future studies.

A key goal involved in the pilot study was to use currently available information, wherever possible, in the construction of a probabilistic assessment tool for DM pressurizer surge nozzle welds. This included modifying existing code, using degradation, leak rate, and flaw stability analytical models that already exist, and plant input information that has already been assembled and published in order to address a very specific PWSCC piping problem. In addition, several of the modules were written specifically for this study. However, as described later in this report, existing and new code developed for the pilot study was incorporated into the framework consistent with the configuration management program.

2.2 Assumptions

The DM weld at the pressurizer surge nozzle was chosen for this study due to the abundance of readily available published information. As part of a prior industry program [5], data was collected for fifty one pressurizer nozzles from nine different operating nuclear power plants. This group consists of thirty five safety and relief nozzles, eight surge nozzles, and eight spray nozzles, each with an Alloy 82/182 DM weld. The information available includes geometries, weld fabrication methodologies, weld repair histories, piping loads (pressure, deadweight, normal thermal, and safe shutdown earthquake (SSE)), and surge line thermal stratification effects. Much of the information used in the development of the pilot study problem statement (see Section 7.0) was taken directly from this the industry program.

In addition to the physical information, several other assumptions were made in the development of the xLPR Version 1.0 code.

- Only one weld location is analyzed per xLPR run. Multiple runs are required to understand the probability of rupture for a piping system.
- Only PWSCC cracking was considered in the pilot study. Since the pressurizer surge nozzle is at a location where the effects of fatigue are small as compared to PWSCC, this assumption is reasonable. With this assumption, the input of transient loads is not necessary.
- Only circumferential cracking was considered in the pilot study. Even though the hoop weld residual stresses are high and would tend to lead to axial crack leakage prior to circumferential crack leakage, the effects of axial cracks will have a secondary effect on increasing rupture probabilities because of their significant rupture tolerance. The presence of axial cracks may reduce the rupture probabilities due to the removal of

circumferential cracks found when mitigating a leaking axial crack Ignoring axial cracks produces non-conservative leakage probabilities.

- Manufacturing defect distributions were not considered in the pilot study. Only PWSCC initiated cracks were considered. This assumption likely produces non-conservative leakage and/or rupture probabilities.
- Weld residual stresses were assumed to be axi-symmetric in the pilot study. This assumption may produce non-conservative leakage probabilities and conservative rupture probabilities since local weld repairs, which are not axi-symmetric, may promote leakage over rupture due to their altered residual stress distributions.
- Cracking was assumed to be planar in the pilot study. Actual PWSCC will rarely be planar. It is unclear how this assumption affects leakage and rupture probabilities.

In addition, there are a variety of assumptions made within the xLPR models that are described in Section 5.0 and Reference [4].

2.3 Outcome

The programmatic outcome of the pilot study is twofold. First, it is necessary to determine if the organizational structure discussed in Sections 1.2 and 1.3 is sufficient to guide the development of a more complex probabilistic computer code. Within this report, as well as in the other task group reports, gap assessments were conducted that identified lessons learned and gaps in data, research, and management structure. Each group used this gap assessment and lessons learned over the course of the pilot study to identify and prioritize recommendations. Second, it is necessary to determine the computational framework structure that is the most appropriate for not only the pilot study problem, but for the full xLPR code as well. For this task, a contractor not affiliated with the xLPR program was used to review and evaluate each computational framework to a set of common metrics, and recommendations were made based on their findings [6]. The complete results of the pilot study are discussed in detail in the pilot study final report [7].

3 Configuration Management

The development of a sophisticated probabilistic software tool that meets robust QA and technical requirements is a challenging task. The management structure, the probabilistic framework, and data handling are just a few of the issues that need to be addressed early in the software development effort. As a result, it is necessary to have a formal process for establishing and maintaining consistency of the xLPR model and its functional attributes with its requirements, design, and operational information throughout the software life cycle. A traditional software configuration management (SCM) process identifies the functional and physical attributes of software at various points in time, and performs systematic control of changes to the identified attributes for the purpose of maintaining software integrity and traceability throughout the software development life cycle. The SCM process further defines the need to trace changes, and the ability to verify that the final delivered software has all of the planned enhancements, and that those enhancements are functioning as intended. The SCM process is also the foundation necessary to demonstrate compliance with QA requirements. Additionally, the xLPR models themselves (e.g., leak, crack growth, residual stress, etc.) need a process similar to the SCM process that deals with the verification and validation of each model for the engineering or scientific question under investigation. For the pilot study, a configuration management (CM) plan has been established to ensure the integrity of the Version 1.0 code, which will be used to define the requirements for the longer term xLPR project. The xLPR CM plan consists of a systematic approach applied to both the developed software and the individual models to ensure that the fundamentals of SCM and a QA program are both met, including: 1) Access Control; 2) Version Control; 3) Verification (e.g., Checking); and 4) Traceability (e.g., Documentation). The xLPR CM plan ensures that a systematic approach is used to meet the requirements and includes documentation of each step in the process. The CM process is implemented as detailed in a series of Guidance Documents which outline the specific steps for each of four key components of the xLPR pilot program: 1) Module Development; 2) Framework Development; 3) Model Parameters and Inputs for the pilot study test case; and 4) xLPR Model Production Runs and Uncertainty/Sensitivity Analyses for the pilot study test case.

Each CM item (e.g., module, framework model, input set, etc.) is developed and controlled using a systematic process, and includes documentation that each item meets the design requirements and can be verified independently (e.g., without consultation with the originator). The CM process used for the xLPR program is based upon the concept of agile software development, which refers to a group of software development methodologies based on an iterative procedure, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams [8]. The CM process incorporates the necessity for thorough documentation and issue tracking through the development process. This includes 'snap-shots' in time of the iterative model development. For example, the xLPR pilot study included both an alpha and beta versions of the code. The alpha version was used as the basis for developing the beta version, as well as to help define the beta version requirements. Once the beta version was checked and locked, which established Version 1.0 of the code, this version of the code was used to run the pilot problems, and the results from those analyses were used to define the longer term xLPR project requirements.

The CM process for xLPR Version 1.0 is a flexible paradigm that is adaptable to multiple CM software systems, which is a necessity when collaborating with teams distributed geographically and utilizing different computational platforms. The xLPR model framework and its associated modules, including source code, documentation, and inputs, are controlled by storing them in a set of access controlled subdirectories on the xLPR file server. The electronic file server for controlled storage of xLPR model files uses the web accessible Microsoft SharePoint process and document management software. Modifications to the CM items (e.g., module source code and xLPR model inputs) were tracked and documented on the SharePoint server. Controlled versions were then available for download when the xLPR model was ready for execution. This central repository enabled the development of the modules and framework models independently, across organizational and geographic boundaries. The developer would check out a CM item from the SharePoint server (e.g., module source code) and make any necessary software modifications, and then upload the file version to subsequently be independently checked and verified. The documentation could also be checked out, modified and checked back in for subsequent independent verification. The central CM repository concept even works well with a standard SCM system software to meet the software and QA requirements for the open-source development of the SIAM framework. The CM items were posted to the SharePoint repository and updated at each iteration or control point defined in the xLPR program. Specifically, the development history, documentation, and issue tracking for the SIAM framework and modules were contained within a separate SCM system (see [3] for details), but the controlled CM items for the alpha and beta model versions were posted to the xLPR SharePoint repository.

4 xLPR Version 1.0 Process

This section provides an overview of the flow of the xLPR frameworks. The xLPR process is embedded within a looping structure to track and propagate uncertainties in the analyses. The discussion will first focus on how the xLPR frameworks handle uncertainties, followed by the process flow for xLPR.

4.1 Uncertainty characterization

The framework for calculating the probability of primary system pipe rupture evokes a systematic approach to uncertainty characterization and the propagation of probability distributions. This section of the report discusses methods for treating uncertainties with a unified approach that allows consistent treatments to be developed regardless of the computer model being used.

Daneshkhah [9] provides the following definitions for aleatory and epistemic uncertainty:

Definition 4.1, Aleatory Uncertainty: This uncertainty arises because of natural, unpredictable variation in the performance of the system under study. The knowledge of experts cannot be expected to reduce aleatory uncertainty although their knowledge may be useful in quantifying the uncertainty. Thus, this type of uncertainty is sometimes referred to as *irreducible* uncertainty.

Definition 4.2, Epistemic Uncertainty: This type of uncertainty is due to a lack of knowledge about the behavior of the system that is conceptually resolvable.

Epistemic uncertainty can, in principle, be eliminated with sufficient study and data development; expert judgments may be useful in its reduction. Epistemic, or internal, uncertainty reflects the possibility of errors in general knowledge. As a simple example, one may believe that the population of City A is less than the population of City B, but there is uncertainty in this belief. Further study, in this case a census, would reduce, and perhaps eliminate, this uncertainty.

Apostolakis [10] claims: “probability is fundamentally the same concept regardless of whether it appears in the model of the world or in the subjective distributions for the parameters. There is only one kind of uncertainty stemming from our lack of knowledge concerning the truth of a proposition, regardless of whether this proposition involves the possible values of the hydraulic conductivity or the number of earthquakes in a period of time. Distinctions between probabilities are merely for our convenience in investigating complex phenomena. Probability is always a measure of degree of belief."

Fundamentally, the simultaneous treatment of multiple uncertainties can be performed in any order[2], and performing inner and outer loops for simulation methods that result in enormous computational times may be inefficient for performing probabilistic analyses. Much depends upon the question asked and what results are desired.

The appropriate separation of aleatory and epistemic uncertainty can be an important component of the design and computational implementation of an analysis of a complex system as well as the decisions that are made on the basis of that analysis. This point can be illustrated

[2] Assuming the stochastic variables are all independent. Order is, of course, important for dependent variables but is left for future development.

with a simple example. Suppose an analysis concludes that the probability of a particular component failing to operate correctly is 0.01. Without the specification of additional information, there are two possible interpretations to the indicated probability. The first interpretation, which is inherently aleatoric, is that 1 in every 100 components of this type will fail to operate properly; or, put another way, there is a probability of 0.99 that a randomly selected component will operate properly and a probability of 0.01 that a randomly selected component will not operate properly. The second interpretation, which is inherently epistemic, is that once a representative component is defined, there is a probability of 0.99 that it will operate properly and a probability of 0.01 that it will not. Clearly, the implications of the two interpretations of the indicated probability are very different and, as a consequence, any resultant decisions about the system under study can also be expected to be very different.

The analysis of a complex system typically involves answering the following three questions[3] about the system:

- What can happen? (Q1)
- How likely is it to happen? (Q2)
- What are the consequences if it happens? (Q3)

and one additional question about the analysis itself:

- How much confidence exists in the answers to the first three questions? (Q4)

The answers to Questions (Q1) and (Q2) involve the characterization of aleatory uncertainty, and the answer to Question (Q4) involves the characterization of epistemic uncertainty. The answer to Question (Q3) typically involves numerical modeling of the system conditional on specific realizations of aleatory and epistemic uncertainty.

Depending on the problem considered and the people studying it, the uncertainty of a parameter could be considered as solely epistemic, solely aleatory, or having both components. The categorization of uncertainty is therefore not totally objective and may change depending on the analysis under consideration and the expert responsible for the characterization. This does not mean that the process is arbitrary and random, but that a careful effort should be placed on the description of each uncertain parameter, including a rationale for its characterization. The interpretation of results will be dependent on this characterization. The classical interpretation of aleatory uncertainty is normally used to identify the variability over which there is no control, e.g. earthquake loading. When one identifies an aleatory uncertainty, it is separated from that uncertainty over which further research, model development, or testing could be beneficial in reducing the risk. However, if the calculated probabilities are still at an unacceptable level, one may want to treat epistemic uncertainty as aleatory within a sensitivity study. This allows the user to rank the epistemic uncertainty (per uncertain parameter) according to its contribution to the response (total) uncertainty, which may give the user basis for directing future research efforts. A final use of the epistemic/aleatory construct is to identify the importance of different physical models. Rather than having a standard Probability Density Function (PDF) represent a random variable, different models can be used.[4]

[3] The three questions provide the basis for the Kaplan-Garrick ordered triple representation of risk.

[4] A probability of various models being correct (our “degree of belief”) could be assigned, but this is a detail outside of the scope of the pilot study, and will be considered in the full xLPR development.

Currently, several additional concepts exist in representing the uncertainty, including fuzzy logic, p-boxes, and evidence theory. Although each of these methods may be more appropriate, depending on the kind of analysis considered and the information available with respect to uncertainty, the pilot study only considers a probabilistic approach. Therefore, a probability distribution is associated with each input parameter that is considered uncertain[5].

In ensuring that the set or combination of input parameters generated from these distributions is physically possible, constraints or correlations among some of the variables are introduced. Finally, within the xLPR pilot study framework, it is possible to replace continuous probability distributions with discrete distributions, which may improve efficiency and allows any variable to be treated as either epistemic or aleatory. The creation of discrete probability distributions is straightforward for standard (normal, lognormal, etc.) distributions and is no different from standard Monte Carlo sampling. Non-standard distributions have not been included in the pilot study, but have been developed elsewhere. Since only standard distributions were provided by the Inputs Group, additional programming was not needed.

As a summary, for the xLPR pilot study code:

- Uncertainty in parameters are handled using probability distributions, defined by the Inputs Group.
- Parameter properties are classified as epistemic, aleatory, or constant by being stored in a specific location.
- Parameters can be easily moved from one category to the other.
- It is possible to correlate some inputs.
- Distributions can be easily redefined.
- The probabilistic framework is decoupled from the numerical (deterministic model) so that any change in the uncertainty treatment will not affect the physical model.

4.2 Uncertainty propagation

The uncertainty in the input is propagated through the model using sampling-based methods. The appropriate way to propagate uncertainty is ultimately dependent on the computational constraints as well as the nature of the inputs and outputs under consideration. As stated earlier, the uncertainty is classified as either irreducible (aleatory) or lack of knowledge (epistemic).

The model development for the analysis of xLPR incorporates the capability to perform sampling-based uncertainty and sensitivity analyses. An inner and outer loop approach is used in the model framework to separate the epistemic and aleatory uncertainties. 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 (N_A). This is repeated for the total number of epistemic realizations (N_E). Thus each epistemic outer loop has N_A number of possible outcomes and (N_E x N_A) represents the total number of possible outcomes generated in the model simulation. Each epistemic realization is an average over all of the aleatory samples. This structure allows the probability of a desired output parameter (per the aleatory uncertainty) to be a distribution (per the epistemic uncertainty) instead of a point value.

[5] With the possible exception of differing models for the same physical process.

The output of interest is the probability of rupture, which is expected to be extremely low for primary piping systems. Therefore, random sampling will generate many runs without any rupture. In this case, the probability of rupture determined by a few runs will, in consequence, be poorly estimated. Therefore, it is appropriate and necessary to use a variety of sampling techniques, which include importance sampling, in order to cover with greater accuracy the regions where the pipe may rupture. However, these regions are not known a priori and have to be determined in an initial step. The xLPR Version 1.0 code is capable of allowing a variety of sampling techniques.

4.3 Sampling techniques

Within Version 1.0 of the xLPR code, several random variable sampling techniques are employed. Random sampling, Latin hypercube sampling (LHS), and discrete probability distribution (DPD) sampling schemes are employed in this version. In addition, importance sampling can be used through the DPD methodology without any changes to the input distribution. However, it must be known what variables to importance sample. Therefore, sensitivity studies must be run first to determine the variables that are driving the rupture events. An adaptive sampling technique is being developed that will allow the code to determine the variables causing rupture without detailed initial sensitivity studies. This technique will be included in Version 2.0 of the xLPR code.

In order to simply illustrate the differences in the sampling techniques, a brief description of each technique is given below. Details for each technique can be found in Appendix A.

4.3.1 Random sampling

The basic principles of Monte Carlo analysis are straightforward. Given a set of inputs {x_1, x_2, …, x_N} that are uncertain and described by a PDF denoted as $f(x_i)$, the cumulative distribution function (CDF), $F(x)$, is constructed by integrating $f(x_i)$:

(1)

A way to sample using the Monte Carlo technique is to randomly and uniformly select a number, denoted R, between 0 and 1 and use the inverse of the CDF to obtain a value for each x_i:

$$x_i = F_i^{-1}(R) \tag{2}$$

The values for x_i are input to the physics model or equation and a response is calculated, denoted Y_1. The entire process is repeated to generate a set of responses, Y_K. These responses are then representative of the distribution of the responses that would be generated if an infinite number of samples were taken. The accuracy of the estimation depends on the output considered and the sample size K. The larger the number of samples taken the more accurate the representation. An illustration is shown in Figure 2, which shows ten random samples (symbols) from a given CDF (solid line).

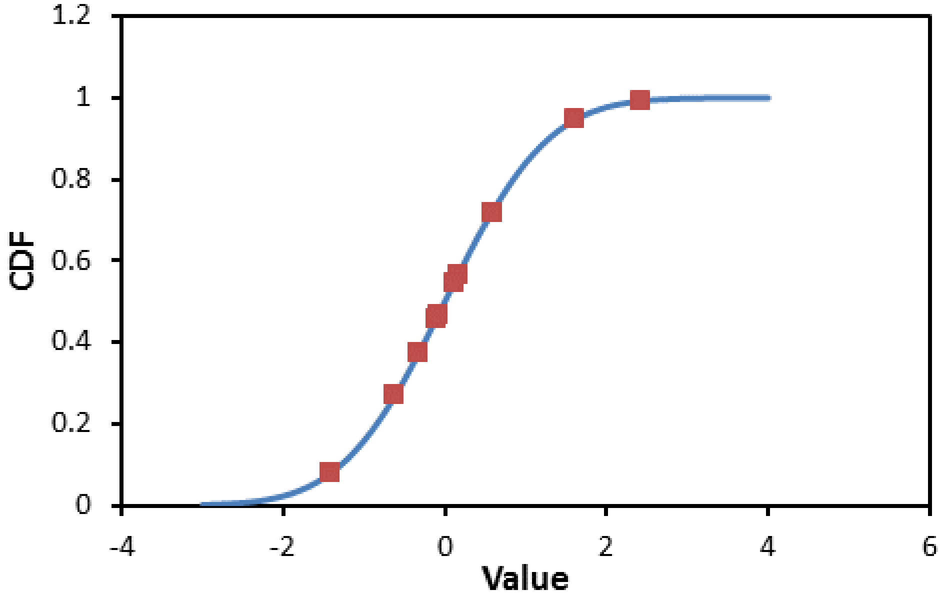


Figure 2 Random sample example

4.3.2 Latin hypercube sampling

The purpose of LHS is to provide a “dense” stratification of each random input or process to a physical model. The LHS is constructed by dividing the input response distribution into N equal probability intervals. This is done for each of the inputs. The first interval for the first variable is then randomly paired with an interval from the second variable, leading to a couplet of (x_1, x_I) where I is the selected random interval for variable 2. If there is a third interval, then this couplet is randomly paired with an interval from the third variable leading to a triplet, (x_1, x_I, x_J), where J is the random interval selected for the third variable. If there are M random variables, then this process is repeated M-1 times leading to an M-tuplet (x_1, x_I, x_J, …, x_M). To obtain the actual value of x_L, a random value would be generated according to the PDF of the variable selected from interval L. This M-tuplet is the input that generates a single response. To obtain the next set of inputs, the same process is repeated except that if a value has been previously selected it cannot be selected again. Thus a *sampling without replacement* scheme is used. This implies that there will be exactly N responses generated. Thus, for M variables, there are N^M possible combinations of the inputs. The LHS design will sample an N^{1-M} fraction of the response space. An example of sampling from a single variable is shown in Figure 3. In this figure, two samples are taken from each of the five bins. Comparing this to Figure 2 illustrates how LHS covers the distribution more efficiently than random sampling.

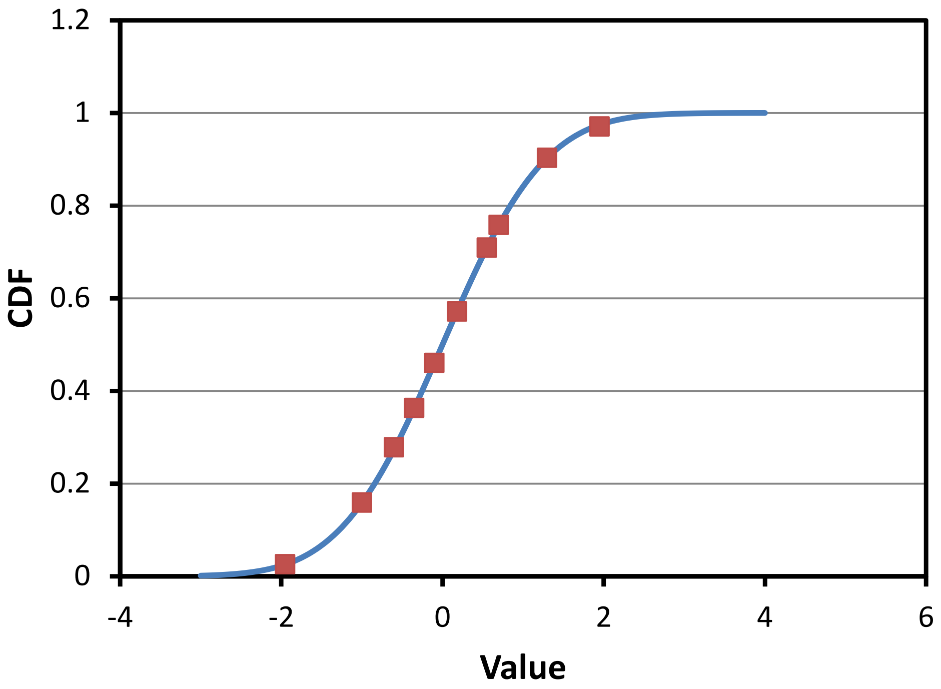


Figure 3 Latin hypercube example for one variable

4.3.3 Discrete probability distribution (DPD) space sampling

In the LHS sampling, when an interval is selected, a value is sampled from within the interval. A modified version of this sampling would employ the same strategy as the LHS method, but simply use the conditional mean of the interval, instead of a random sample within the interval. This modified version for LHS comparisons is used since duplicate design runs are not required to compare similar strategies.

In addition, the way in which the individual DPD's are generated can be changed for the input PDF. Rather than using *equal* probability intervals, *unequal* probability intervals are used to perform importance sampling. An example of this type of sample is illustrated in Figure 4. Using 10 bins biased toward the tails with one sample per bin, the distribution is well represented.

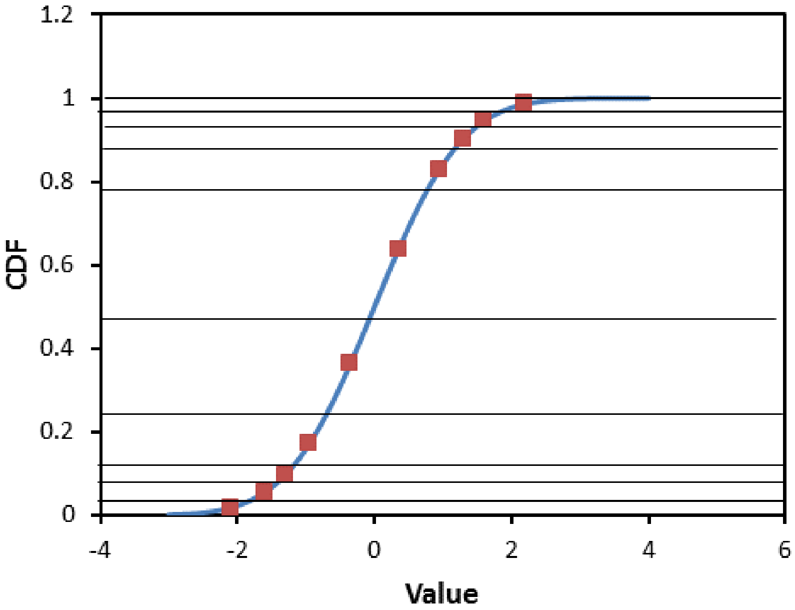


Figure 4 DPD with importance sample example

4.3.4 Adaptive sampling

Multiple random variables and fractured response surfaces cause many issues when trying to calculate low probabilities of rupture, even with the sampling techniques listed above. In addition, it is not always (and, in fact, may rarely be) the case that the area of most interest is in the tails of a distribution. For example, low values of the initial crack length are not the most likely situation to lead to pipe rupture. On the other hand, low values of yield and ultimate strength can lead to pipe rupture. Therefore, in an analysis where it is unknown which combination of inputs may lead to a failure of consequence, the sampling technique becomes vital to finding these combinations within a reasonable amount of simulations.

If the variable or variables that are controlling the results are known, e.g., the weld residual stress, then importance sampling can be used on those variables. If controlling variables are unknown, sensitivity analyses can be run in order to determine the critical variables to importance sample. This may be time consuming, since a probabilistic analysis would need to be conducted with enough refinement to capture the important variables. Another method is to specify a value of the response of interest.[6] In this case, the input values are saved and the PDF is focused, using an adaptive technique, on the saved value for each input. Through an automated process, the DPD bins are contracted around this value and expanded farther away from the current value. With this method, the controlling variables are sampled such that they affect the response of interest. The user would not need to indicate which variables are to be importance sampled. This technique, which is described in Appendix A, shows great promise and will be further investigated for Version 2.0. In addition, other optimization methods, such as FORM/SORM, will also be investigated.

[6] For example, the crack depth is greater than 50% of the wall thickness.

4.4 Time loop

As described previously, the uncertainty propagation structure for the Version 1.0 xLPR code consists of an inner aleatory loop and an outer epistemic loop in which each random variable is sampled. Afterward, the initial conditions for that particular realization are constructed to form a time line for the input parameters.

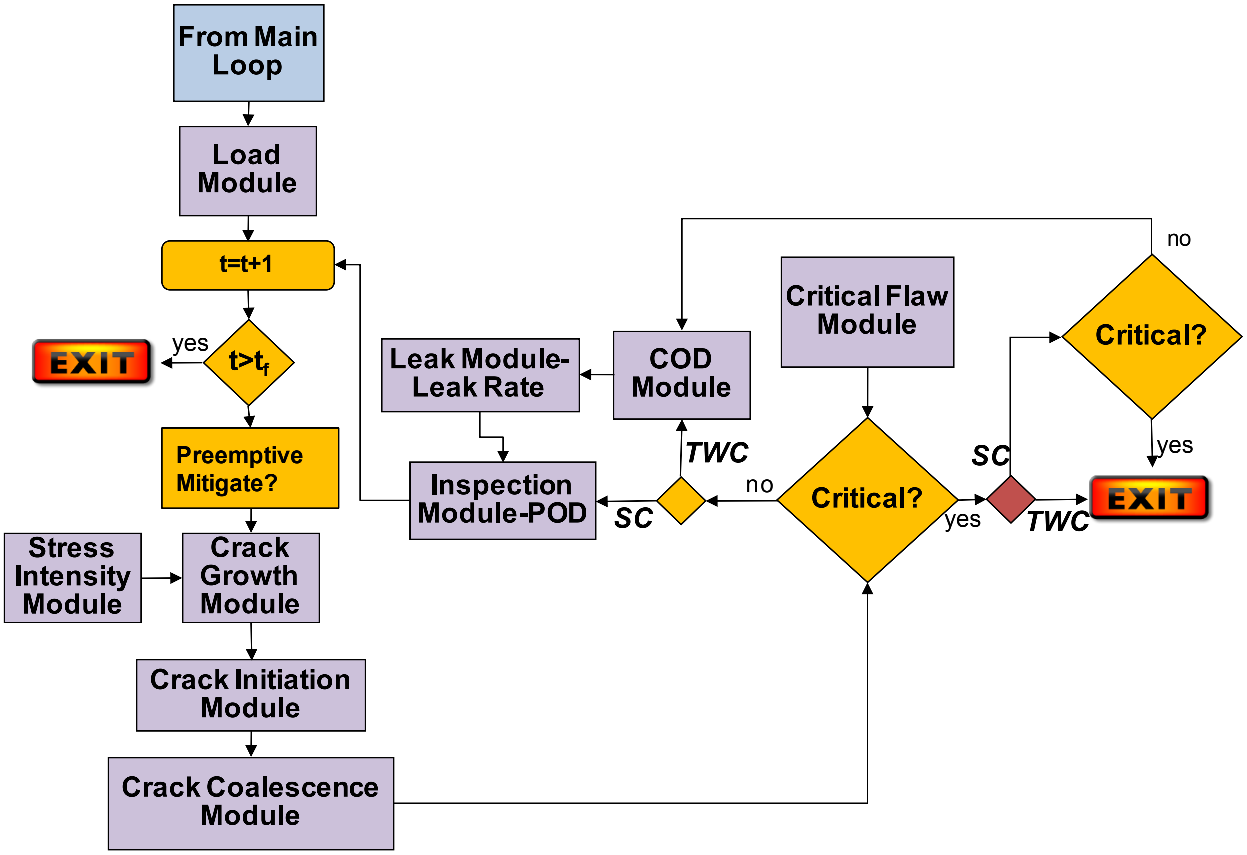


Figure 5 Time loop flow chart for Version 1.0 xLPR code

Figure 5 illustrates the time loop for the Version 1.0 code. The purple boxes in this figure are individual modules that are summarized in Section 5 and detailed in Reference [4]. Before entering the time loop, the load module is called. The load module (described in Sections 5.1 and 5.2) calculates the stress values, including the weld residual stress, for each of the load inputs. This module calculates all of the appropriate stress values for the crack initiation, growth and stability modules.

For each time increment (set by user input), the code checks whether the analysis is beyond the predefined time period for the analysis. If it is, the time loop is exited. If not, it continues.

If a pre-emptive mitigation (described in Section 5.3) is to be performed at this time increment, the code will apply mitigation and continue.

If the crack initiation model (described in Section 5.4) dictates that a crack initiates in this time step, a single, surface-breaking crack is placed in the model with the sampled size and location.

The coalescence model (as described in Section 5.7) will not permit locating a crack at the same location as an existing crack.

The crack growth module (described in Section 5.6) is then used to calculate the crack growth increment for any existing cracks. As a first step to this calculation, the stress intensity factor module (described in Section 5.5) is called and appropriate stress intensity factors are calculated from the instantaneous loads, including the residual stress. These stress intensity factors are then used with the crack growth model to calculate the crack growth increment. Each of the existing crack sizes is updated, and if any surface crack has reached 100% through-wall, it is transitioned to a through-wall crack. In addition, a check is made to determine if any surface cracks or through-wall cracks coalesce (as described in Section 5.7). If they do, they are combined.

Next, the crack stability module (described in Section 5.12) is used to determine if any existing cracks have reached a critical size. At any time increment, through-wall cracks (TWC), or surface cracks (SC) may exist in the analyses. For existing through-wall cracks, if the instantaneous crack size is larger than the critical crack size, a double-ended break (severance of the pipe) is assumed. For existing surface cracks, if net-section collapse failure is predicted at the operating loads, the crack transitions to a through-wall crack. In this case, if the resultant through-wall crack length is greater than the critical through-wall crack length, a double ended break is assumed. The size of this opening at failure is recorded and the time loop is exited.

If a through-wall crack is not critical, the leakage module (described in Section 5.11) is used to determine the level of the leakage. A leakage calculation is performed for the through-wall crack using the calculated crack-opening displacement (described in Section 5.10). The calculated leak rate for this time increment is stored and the time loop continues.

If a surface crack and/or a through-wall crack is found to be stable, the inspection module (described in Section 5.8) is used to calculate a probability of detection for each crack in the analysis. The time loop then continues.

4.5 Outputs and post-processing

The xLPR Version 1.0 code was structured so that all data (all variables, every realization, and every time step) generated can be saved via a database[7]. Even though the file structure is different between the SIAM and GoldSim versions of the code, the same output data is stored. Early in the developmental process, it was decided that certain outputs would be generated for both debugging purposes and data analysis for the pilot study problem. In both framework cases, the codes output text files that are used to calculate:

- Time-dependent crack depth for any relevant crack.
- Time-dependent half crack length for any relevant crack.
- Time-dependent cracked fractional surface area.
- Time-dependent stress intensity factor for any relevant crack.
- Time-dependent probability of non-detection for any relevant crack.
- Time-dependent leak rate for any relevant crack.
- Total time dependent leak rate.
- First leakage probability as a function of time.

[7] In both cases, some file size optimization is needed before all of the data are saved.

- Crack opening area (COA) > 1-inch equivalent break diameter ($507mm^2$) probability as a function of time.
- COA > 3-inch equivalent break diameter (4,560 mm^2) probability as a function of time.
- Rupture probability as a function of time.

The raw data generated by both framework codes is somewhat cumbersome and difficult to use, i.e., for each variable, data is output for every realization and every time step, which can be an enormous amount of data. To aid in the data reduction effort, a set of post-processing tools have been developed as separate, stand-alone software to estimate some output variables (such as probability of rupture) under a variety of conditions without having to rerun the framework model, while still accounting for the separation of aleatory and epistemic uncertainties or importance sampling. There were two post-processing software applications developed as part of the xLPR pilot study, Transformers and Expectation.

The post-processing code Transformers calculates the credit for leak detection and in-service inspection from the framework output. As described above, the leak rates for each through-wall crack at each time increment are stored in the output files. The user is required to indicate the leak rate threshold. Based on this value, the Transformers code generates an indicator function (a set of 1 and 0) using leak rate history (from the code database output files) for each realization. This indicator function is used as a multiplier for any variable of interest, e.g., probability of leak, probability of failure, fractional surface area cracked, depth, and length of cracks.

For in-service inspections, the user inputs an inspection schedule. Several options are available to combine the effect of detection of multiple cracks and the efficiency of subsequent inspections, leading to different options in the post-processing techniques used (see Reference [2] for details). Once the options have been selected, Transformers will construct a correction matrix (as is done for leak detection). The difference is that this matrix will not be made with a set of 1 and 0, but will instead also include values between 0 and 1 representing probabilities of finding a crack during the inspection. This correction is then used as a multiplier on the variable of interest.

The post-processing code Expectation is used to calculate the expected value for certain variables (refer to the list of outputs above) over time, as well as complementary cumulative distributions functions (CCDF) and associated statistics (see Reference [2] for details) at a selected time-step. When the Monte Carlo method is used for sampling on aleatory uncertainty, the calculation of expected value is straightforward, as it is a simple average over the aleatory samples for a particular epistemic realization. If importance sampling is used, then the weight associated with each simulation has to be used in order to calculate the expected value correctly. Once the expected values are estimated, the mean for the expected value is calculated over the epistemic uncertainty (with or without importance sampling). Quantiles, representing the epistemic uncertainty, are estimated using sorting techniques and “counting” (with equal or unequal weight) up to the desired value.

5 Module Descriptions

In this section of the report, a brief description of the deterministic modules used in the xLPR pilot study is presented. This discussion is only meant to give an overview of the models implemented in this version of the xLPR code. A detailed description of the models and inputs can be found in the group report [4].

5.1 Operating load and stresses

For the xLPR pilot study, the piping loads were taken directly from MRP-216 [5], and are shown in Table 1. Except for pressure and temperature, the piping loads inputs are assumed to be constant. For the xLPR pilot study, the mean operating pressure was assumed to be 15.51 MPa (2.25 ksi) and the mean temperature was assumed to be 344.9 °C (653 °F). The distributions for both pressure and temperature are discussed in Reference [4].

5.1.1 Normal operation

The static normal operating load contributions are the result of pipe pressure, temperature, dead weight, and through-thickness weld residual stress (see Section 5.3 for weld residual stress). The axial stress components due to internal pressure (P), dead weight (DW), and normal thermal loading (NTE) are calculated as:

$$\sigma_{oP} = \frac{F_P}{A}, \sigma_{oDW} = \frac{F_{DW}}{A}, \sigma_{oNTE} = \frac{F_{NTE}}{A} \tag{3}$$

where:

A is the cross-sectional area of the pipe, in^2 (m^2)
F_{DW} is the dead weight axial load, lb (kN)
F_{NTE} is the normal thermal axial load, including stratification, lb (kN)
F_P is the load due to P, and is defined as

$$F_P = P\left(\frac{\pi D_i^2}{4} + f_{cracked}\left(\frac{\pi\left(D_o^2 - D_i^2\right)}{4}\right)\right) \tag{4}$$

D_o is the outer diameter, in (m)
D_i is the inner diameter, in (m)
$f_{cracked}$ is the crack area divided by pipe cross-sectional area. This term is time-dependent, and is updated as the percentage of the cracked area increases.

Table 1 Normal operating loads from MRP-216 [5]

	Fx		Mx		My		Mz	
	kips	kN	in-kips	kN-m	in-kips	kN-m	in-kips	kN-m
Normal Thermal	0.87	3.87	577.96	65.30	-509.32	-57.54	468.98	52.99
Deadweight	0.07	0.31	11.63	1.31	1.90	0.21	8.99	1.02
Safe Shutdown Earthquake (SSE)	6.30	28.02	286.67	32.39	524.43	59.25	839.86	94.89
Normal Thermal Stratification	3.91	17.39	22.26	2.51	-715.11	-80.79	778.04	87.90

The global bending stresses resulting from dead weight or thermal expansion are calculated from their bending moment and torque components. An effective moment is calculated using Equation 5:

$$M_{eff} = \sqrt{M_y^2 + M_z^2 + \left[\frac{\sqrt{3}}{2}M_x\right]^2} \quad (5)$$

where,

M_y = sum of in-plane moment about the y-direction, in-kips (kN-m)
M_z = sum of in-plane moment about the z-direction, in-kips (kN-m)
M_x = sum of torque about the x-direction, in-kips (kN-m)

The global bending stress is then calculated using the elastic relationship given by Equation (6):

$$\sigma_B = \frac{M_{eff} R_o}{I} \quad (6)$$

where

R_o = Outside pipe radius, in (m)
I = Moment of inertia, in^4 (m^4) = $\pi(R_o^4 - R_i^4)/4$
R_i = Inner pipe radius, in (m)

For the analysis location of interest, the spatial distribution of stress is calculated. The axial membrane stress is constant around the circumference of the pipe, but the bending stress is calculated at the maximum location (Equation 6), and then scaled according to the azimuthal location. The scaled bending stress is given by Equation (7):

$$\sigma_{B-L} = \sigma_B \cos(\phi) \quad (7)$$

where ϕ is the azimuthal location (radians) of the crack center relative to pipe's top dead center.

In the xLPR Version 1.0 code, both primary and secondary stress components are both conservatively treated as primary. This assumption will be revisited in future versions of the xLPR code.

5.1.2 Transient loads

For the Version 1.0 xLPR code, only transient loads associated with the pressurizer surge nozzle were considered. Since the pilot study only focused on PWSCC, fatigue is not considered. Therefore, the transient loads listed in this section only affect the critical crack size predictions and not the crack growth predictions. Linear superposition of loads is assumed in all cases.

5.1.2.1 Thermal stratification

A load condition that applies to pressurizer surge nozzles is thermal stratification [5]. Thermal stratification occurs in the surge line of pressurized water reactors as a result of the temperature difference between the (hotter) pressurizer and the (colder) RCS hot leg, which are connected by the surge line. The stratification produces a pipe bending load resulting from the colder fluid flowing along the bottom of the horizontal surge line and the hotter fluid flowing along the top of the line. This bending load is restrained by the relatively stiff pressurizer surge nozzle in the safe end region. The thermal stratification loads during normal operation are approximately equal to the piping thermal expansion loads without the presence of thermal stratification. (i.e., the load resulting from thermal expansion of the line due to heating of the pipe). The stratification loads for some plants become significant during plant heat-up and cool-down, when the pressurizer and hot leg temperature differential may be large and flow in and/or out of the pressurizer occurs. The limiting thermal loads for all surge nozzles are reported as the entire maximum thermal load including all other effects, such as piping expansion.

5.1.2.2 Earthquake

Earthquake loads can be significant, and were therefore considered in the Version 1.0 xLPR code. Piping failure from earthquake loads can be a combination of overload and low cycle fatigue. The typical earthquake is a low frequency, short term event, with maximum loads (variable amplitude) significantly higher than loads that occur during normal operating conditions. Since the event time for an earthquake is significantly shorter than the time increment used in the Version 1.0 xLPR code, it is assumed that the earthquake occurs within one time step. Also, since version of the code is focused only on PWSCC and neglecting contributions from fatigue, the low cycle fatigue aspect of earthquakes are ignored.

The earthquake input loads are the maximum membrane and bending loads. The number of cycles that occur per earthquake is another input, but this is neglected in the Version 1.0 xLPR code. This omission is slightly nonconservative, and will need to be considered for the final xLPR code. Since fatigue is not considered, earthquake loading only affects crack stability.

In order to correctly account for the probability of an earthquake in the overall rupture probabilities, the earthquake probability of occurrence would normally also be needed. However, the earthquake probability is neglected in Version 1.0 of the xLPR code. The limiting thermal stratification loads from MRP-216 [5] were assumed to be normal operating loads making the stresses from the SSE event small in comparison to both the normal operating loads and the weld residual stress. Therefore, for Version 1.0 of the code, the rupture probabilities are calculated with and without SSE loading, assuming a probability of occurrence for an earthquake event equal to 1. In future versions of the code, earthquake loading will be handled in a more robust manner.

5.2 Weld residual stress

Weld residual stress (WRS) is one of the major drivers to stress corrosion cracking and must be included for accurate predictions of subcritical crack growth. Factors such as weld repairs, grinding, etc., all impact the probability of leakage and the potential for rupture. In the xLPR Version 1.0 code, the effects of local weld repair are neglected and the weld residual stress is assumed to be axis-symmetric. To account for uncertainty in the weld residual stress, the inside surface (ID) stress and through-thickness location where the weld residual stress is zero, X_c, are both considered to be random (see Figure 6). The ID stress (mean, standard deviation, and distribution type, i.e., uniform, normal, lognormal, etc.) and values for X_c (mean, standard deviation, and distribution type), where X_c is defined in terms of a fraction of the pipe wall thickness (x/t), are input and used to generate the coefficients for a third-order polynomial with the following form:

$$\sigma_{WRS} = \sigma_{oWRS} + \sigma_{1WRS}\left[\frac{x}{t}\right] + \sigma_{2WRS}\left[\frac{x}{t}\right]^2 + \sigma_{3WRS}\left[\frac{x}{t}\right]^3 \tag{8}$$

where

σ_{nWRS} = Curve fit coefficients for n = 1, 2, and 3
x = distance from ID, in (m)
t = wall thickness, in (m)

In addition to the ID stress and X_c, the following constraints are used:

1.) The area under the WRS curve must equal zero (i.e., the stress equilibrates through the thickness).
2.) The stress on the outside surface (OD) is simulated using a uniform random number that ranges between 0 and 0.5*σ_{0WRS}.
3.) If X_c >0.4, the OD stress has the opposite sign as σ_{0WRS}. If X_c<0.4, the OD stress has the same sign as σ_{0WRS}. If X_c = 0.5, the stress is linear through the wall with the OD stress equal but opposite in sign to σ_{0WRS}.

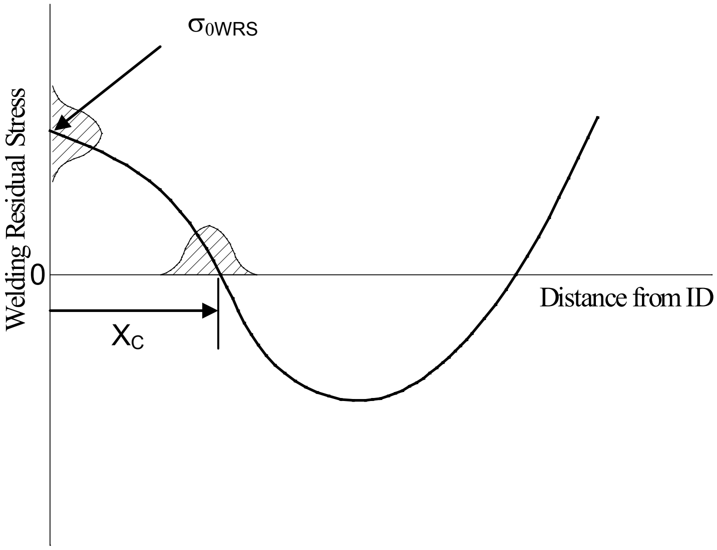


Figure 6 Weld residual stress distribution schematic

5.3 PWSCC mitigation

As a demonstration of the capability of evaluating PWSCC mitigations, a stress-based mitigation model has been implemented in the xLPR Version 1.0 code. For this version of the code, only a change in the weld residual stress behavior caused by the mitigation is assumed to occur at a user-defined time. Using the mitigation weld residual stress definition, the load module (see Section 5.2) is called, and the stress values in the mitigated condition are calculated. Each of the modules, including the crack initiation model, was defined to handle this instantaneous change in stress.

The effects of hydrogen on the PWSCC growth rate are considered and discussed in Section 5.6. As described in Reference [4], the effects of zinc and hydrogen on PWSCC initiation were considered, but not implemented, in the xLPR Version 1.0 code.

Finally, as described in Section 5.8, inspection-based mitigation is not considered, and any crack found by inspection is assumed to be fully repaired.

5.4 Crack initiation

Crack initiation models are an important driver to predicting probability of rupture in piping systems. Flaws that can lead to loss of coolant accidents (LOCAs) can either initiate as a result of residual stresses (e.g., stress corrosion cracks), service loadings (e.g., mechanical or thermal fatigue cracks) or they can grow from pre-existing flaws that are introduced during the welding process and associated imperfections (e.g., lack of fusion, porosity, slag, etc.). For the pilot study problem, since PWSCC initiation and growth mechanisms control the behavior for the surge nozzle selected for evaluation, only PWSCC was considered, i.e., pre-existing defects and fatigue initiation were not considered.

The initiation of PWSCC is a complicated process where phenomenological models are not matured to the point of being able to include their effects in probabilistic fracture mechanics software. Therefore, for Version 1.0 of the xLPR code, only models that are empirically driven were considered. Three separate models are currently incorporated into the code, as shown in Figure 7. Of the three models, two are time-based models that are corrected for temperature and stress. The third model is a Weibull model that is also corrected for temperature and stress. Any of these models can be calibrated to either laboratory or service-based crack initiation data. For the Version 1.0 code, the models were calibrated to the service data found in Reference [5], which leads to an arrival rate of approximately 0.01 cracks/year. The details of the crack initiation models and their calibrations can be found in Reference [4].

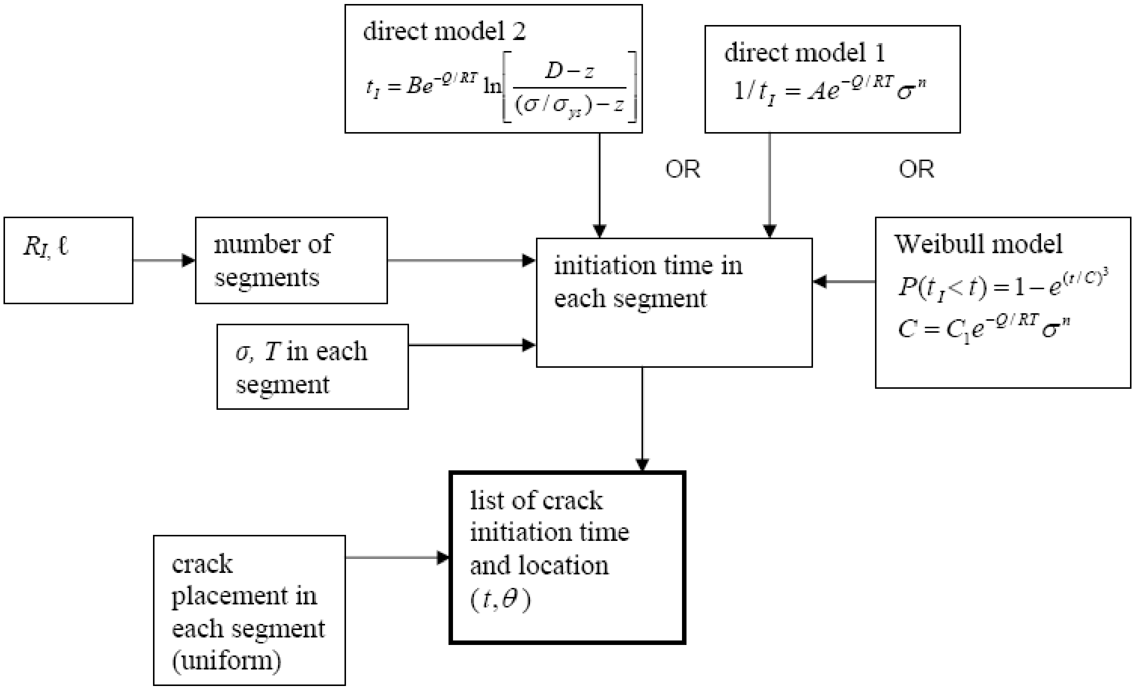

direct model 2
$t_I = Be^{-Q/RT} \ln\left[\frac{D-z}{(\sigma/\sigma_{ys})-z}\right]$
direct model 1
$1/t_I = Ae^{-Q/RT}\sigma^n$
OR
OR
R_I, ℓ
number of segments
initiation time in each segment
Weibull model
$P(t_I < t) = 1 - e^{(t/C)^3}$
$C = C_1 e^{-Q/RT}\sigma^n$
σ, T in each segment
list of crack initiation time and location (t, θ)
crack placement in each segment (uniform)

Figure 7 Flow Diagram of Version 1.0 xLPR Crack Initiation Module.

The three crack initiation models treat size effects by breaking the weldment up into segments of length, ℓ. Only one crack can initiate in each segment, as shown in Figure 8.

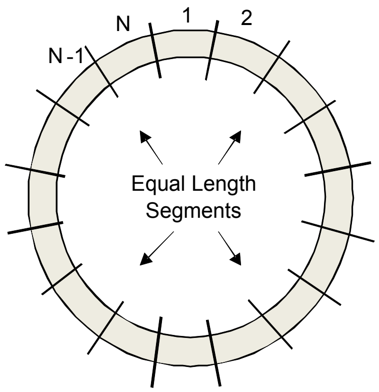


Figure 8 Segmentation of the Pipe Circumference

Each of the segments can have a different stress and temperature. The uncertainty in the "constants" in the relation defining the initiation time has within-heat and heat-to-heat contributions (or within-weld and weld-to-weld). The weld-to-weld uncertainty defines the scatter in the time until the first crack initiates in a particular weld. The within-weld uncertainty parameter defines the subsequent scatter in the time to cracking for segments within that particular weld. It is expected that the time to cracking is more tightly correlated for the wetted material of a particular weld than for wetted material for multiple welds, given that some key welding parameters do not vary for a particular weld.

The relative variances of heat-to-heat versus within-heat cracking may control the propensity to form long cracks as compared to cracks that are isolated around the circumference of the pipe. The choice ofℓ is therefore based on specimen size (such as one inch or two inches), with modifications based on calibrations to field data. For the pilot study, 19 segments (each about 2-inches in length for the surge nozzle safe end) were chosen. Sensitivity studies were conducted and demonstrated that the leak probabilities are not highly influenced by small changes in the number of segments. Cracks initiated within a segment are randomly placed within that segment.

5.5 Stress intensity factor solutions

For a cracked structure under remote or local loads, the stress intensity factor (K) is a measure of the stress field ahead of the crack. In elastic fracture mechanics, when the applied value of the stress intensity factor exceeds the material's critical value, crack advance occurs. For subcritical cracking, the process of crack advance is linked to the applied value of the stress intensity factor through curve fits that are based on extensive experimental data.

For the xLPR Version 1.0 code, the Anderson K-solutions for both surface and through-wall cracks in cylinders are used. Anderson's K-solutions for a circumferential surface crack on the inside pipe diameter are given in Reference [11]. The solutions in this report were generated for R_i/t values from 3 to 100, c/a values from 1 to 32 and a/t values from 0.2 to 0.8. Anderson generated influence functions[8] G_0, G_1, and G_5 (global in-plane bending) using finite element techniques. The influence functions G_2, G_3, and G_4 are inferred from the weight function formulas given in Reference [11]. For the case of a circumferential, semi-elliptical surface crack, the crack growth at both the deepest (90 degrees) and surface (0 degrees) locations are

[8] Influence functions are dimensionless constants that are a function of geometry and loading

calculated and applied to the initial crack sizes. The finite length surface crack is always assumed to remain semi-elliptical.

There are several shortcomings to these solutions. First, the influence functions were only generated for a/t values from 0.2 to 0.8. This leads to a shortcoming when trying to predict crack behavior from initiation to failure. Several assumptions were made to accommodate this shortcoming. First, it was assumed that the influence functions can be extrapolated from a/t=0.8 to a/t=1.0. Second, a solution by Chapuliot [12] was used for a/t values approaching zero. Linear interpolation was used between these values and Anderson's results at a/t = 0.2.

In addition to elliptical surface cracks, Anderson also generated K solutions for a/c = 0 (infinitely-long surface crack). Since long surface crack K-solutions fall between these two extremes and are currently not available, it was assumed that for surface cracks with c/a greater than 32, the K solution at the free surface is equal to the K-solution at c/a = 32. At the deepest point, the K-solution was assumed to equal the K-solution for a/c = 0. These assumptions are conservative in the length direction, because as the crack length gets longer, the influence functions (hence the K-solution) at the free surface tend toward zero. By using the K-solution at the free surface equal to c/a = 32, slightly larger crack growth will occur, producing conservative leak probabilities.

The Anderson K-solutions for a circumferential through-wall crack in a pipe are given in Reference [13]. These solutions were generated for R/t values from 1 to 100 and to crack lengths of about 66 percent of the circumference. The solutions were generated for both the inside and outside surface of a through-wall crack. However, only the G_0, G_1 and G_5 influence functions are available. In Reference [13], the through-wall crack K-solutions were curve fit and the coefficients are presented for R/t values of 1, 3, 5, 10, 20, 60, and 100. These coefficients were used in this initial version of xLPR, and linear interpolation was used to predict the coefficients for other R/t values. The influence function on both the inside and outside surface of the through-wall crack are calculated, and then averaged to get the K-solution for through-wall-crack growth.

5.6 Crack growth

For the pilot study, PWSCC of DM butt welds is the only subcritical cracking mechanism that is considered. For each time increment, PWSCC growth is calculated using the loads and stress intensity solutions presented previously. The PWSCC crack growth model incorporated into the xLPR Version 1.0 code was developed from Alloy 82/182 DM weld laboratory experimental data used to measure the growth rate of PWSCC. The collective experimental data has been gathered and analyzed by EPRI in MRP-115 [14]. The PWSCC growth model developed in that effort includes a temperature-corrected crack growth rate, with uncertainty in the rate both from a within-weld and weld-to-weld perspective. The effects of dissolved hydrogen on the MRP-115 crack growth rate are detailed in MRP-263 [15], and were also implemented into the xLPR Version 1.0 code. A modification to the MRP-115 equation was developed in the MRP-263 study adding the dependence on dissolved hydrogen concentration (for the case of Alloy 182/132):

(9)

where:

$\dot{a}$ = crack growth rate, (m/s)
Q_g = thermal activation energy for crack growth (mean value of 130 kJ/mole)

R = universal gas constant (8.314×10^{-3} kJ/mole-K)
T_{abs} = absolute temperature at location of crack (K) (=T+273.15)
T = temperature at location of crack (°C)
T_{ref} = absolute reference temperature used to normalize data (598.15 K)
α = power-law coefficient (2.01×10^{-12} (m/s)(MPa-m$^{0.5}$)$^{-1.6}$)
f_{weld} = common factor applied to all specimens fabricated from the same weld to account for weld wire/stick heat processing and for weld fabrication (mean value of 1.0 for probabilistic assessments; deterministic value of 1.39 based on 75th percentile of the f_{weld} distribution)
f_{ww} = "within weld" factor that accounts for the variability in crack growth rate for different specimens fabricated from the same weld (mean value of 1.0 for probabilistic assessments; deterministic value of 1.0)
K_I = crack-tip stress intensity factor (MPa-m$^{0.5}$)
K_{Ith} = crack-tip stress intensity factor threshold, below which $\dot{a}$ is zero (MPa-m$^{0.5}$)
β = stress intensity factor exponent (1.6)
$\Delta ECP_{Ni/NiO}$ = electrochemical potential difference at the current hydrogen concentration to the Ni/NiO transition at temperature (mV)
P_R = peak-to-valley ratio for effect of electrochemical potential on crack growth rate (mean value of 9.5)
c_w = characteristic width of crack growth rate peak versus electrochemical potential (mean value of 22.5 mV)

Note that a "within-weld" factor was not implemented for the xLPR Pilot Study, but is recommended for future versions to account for this additional source of variability in the crack growth rate.

The difference in electrochemical potential between the nickel/nickel oxide (Ni/NiO) transition and the electrochemical potential at the current concentration of hydrogen is calculated as follows:

$$\Delta ECP_{Ni/NiO} = 29.58\left(\frac{T+273.15}{298.15}\right)\log\left(\frac{[H_2]}{[H_2]_{Ni/NiO}}\right) \tag{10}$$

where:
$[H_2]$ = concentration of dissolved hydrogen in the primary water (cc/kg at standard temperature and pressure (STP))
$[H_2]_{Ni/NiO}$ = concentration of dissolved hydrogen corresponding to the nickel metal / nickel oxide (Ni/NiO) transition at temperature at location of crack (cc/kg at STP)

The concentration of dissolved hydrogen corresponding to the potential at the Ni/NiO transition is temperature dependent, and is calculated as follows:

$$[H_2]_{Ni/NiO} = 10^{(0.0111T-2.59)} \tag{11}$$

Again, the set of inputs developed for the MRP-263 crack growth module, which are incorporated into xLPR Version 1.0, with regard to the influence of dissolved hydrogen concentration are specific to Alloy 182. Future version of the xLPR code will include the influence on all materials of interest.

5.7 Coalescence

Crack coalescence is an important part of estimating that the cracks that are initiated in the code are representative of the long surface cracks found in service. For circumferential surface cracks, as shown in Figure 9, when the distance between the surface cracks becomes less than one half the deepest surface crack depth, the cracks coalesce and are merged together as one single flaw. The depth of the new crack is equal to the deeper of the two combined surface cracks, and the length is equal to the sum of the lengths of both cracks plus the distance between them.

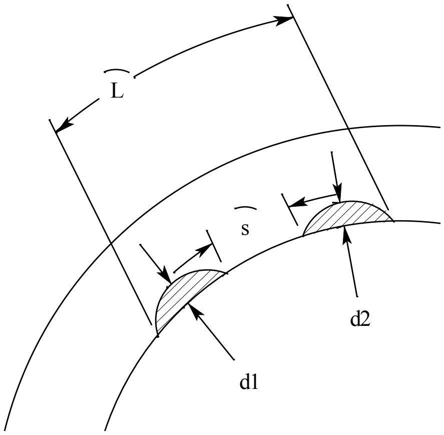


Figure 9 Surface crack coalescence

Another case of coalescence is when two through-wall cracks interact. Realistically, this case has a low probability of occurrence due to leak detection. If two through-wall cracks, with total leakage less than the technical specification limit, are present, they coalesce when the crack ends touch.

There is also a possibility that a through-wall crack may interact with a surface crack. In this case, if the crack ends touch, a complex crack, with a portion that is part through-wall and a portion that is fully through-wall, is formed. However, as with the penetrating crack, proper stress intensity solutions for complex cracks are unavailable. For the xLPR Version 1.0 code, it was assumed that if a through-wall crack and a surface crack interact, a through-wall crack is formed with a crack area equal to the sum of the two interacting crack areas. Under this assumption, the crack length of the new crack is shorter than the sum of the lengths of the two merged cracks

5.8 Inspection

One of the large drivers in predicting rupture probabilities is the credit given for in-service inspections. During the development of the Version 1.0 code, much discussion occurred on how to account for crack depth detection, crack length detection, flaw sizing, post-repair flaw distribution and flaw orientation. It was decided to focus on flaw depth detection as a demonstration of the influence of in-service inspection.

The flaw detection and sizing results from the ASME Code, Section XI, Appendix VIII, Supplement 10 (DM welds) qualifications for examination from the outside surface of the surge line location were used to develop the Probability of Detection (POD) and sizing uncertainty distributions. Based on recent efforts by EPRI [16], the POD for DM welds is represented in Version 1.0 by the following functional form:

$$POD\left(\frac{a}{t}\right) = \frac{e^{\beta_1+\beta_2\left(\frac{a}{t}\right)}}{1+e^{\beta_1+\beta_2\left(\frac{a}{t}\right)}} \tag{12}$$

where β$_1$ and β$_2$ are model coefficients from maximum likelihood estimate regression analysis. For surge nozzle size pipe welds, the coefficients are given in Table 2:

Table 2 POD Coefficients for surge nozzle

diameter range , in	12-14
thickness range, in	1.2-2.3
mean β_1	2.7076
mean β_2	0.31
stdev β_1	0.2085
stdev β_2	0.45
Correlation (β_1, β_2)	-0.85912

From these curves, the probability of non-detection (PND) is calculated as (1-POD). At each time increment where a crack is present, the PND is stored for each instantaneous flaw depth in the analysis. The Version 1.0 code does not take credit for inspection during program run-time, but continues assuming no inspection occurs.

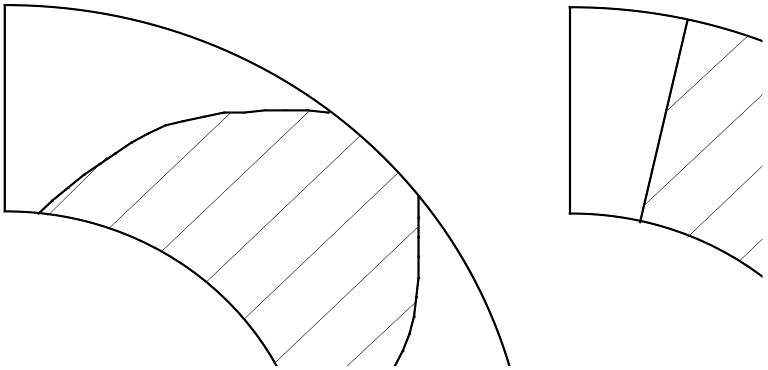
Through the post-processing routines (see Section 4.5), the credit for in-service inspection (ISI) is applied. Using an inspection schedule, the code adjusts the calculated output for the PND at the inspection intervals. In the presence of multiple cracks, the user can select whether the probability of detecting a crack is independent of the presence of other cracks. If each crack is considered independently, the probability of detecting no cracks at all is equal to the multiplication of probabilities of not detecting each crack. However, one can also consider that the method used to detect a surface crack is such that if the most obvious crack is not detected, the other cracks will also not be detected. In this case, it is more appropriate to take the minimum of the sampled probabilities of non-detection values (this minimum represents the probability associated with the easiest crack to detect) rather than the product of the values.

With respect to the efficiency of subsequent inspections, the user can choose to have independent inspections (so the probability of not detecting a crack at time T_1 and then at subsequent time T_2 is equal to the probability of not detecting a crack at T_1 times the probability of not detecting a crack at time T_2), or he may choose to want some dependency, such that if a surface crack was not detected the first time, it is unlikely to be detected again (unless it becomes a through-wall crack). This option checks the status of each potential crack at the time of each inspection and will not correct for any crack detection on subsequent inspection, unless the crack status changes, e.g., surface crack becomes a through-wall crack.

In addition to PND, a sizing model for surface cracks in surge nozzles was developed, but it was not completed in time to include in the Version 1.0 code.

5.9 Transition to through-wall crack

As an internal surface crack first penetrates the wall thickness, only a small breach of the pressure boundary is observed. For an internal surface crack that becomes a through-wall (leaking) crack, the crack length on the OD is much smaller than that on the ID due to the initial flaw shape and previous growth of that flaw. In idealized, through-wall behavior, the crack front is assumed to run radially, so the actual OD crack length is longer than that on the ID, as shown in Figure 10.



(a) Penetrating crack (b) Idealized through-wall crack

Figure 10 Illustration of penetrating cracks and idealized through-wall crack

For the crack shape shown in Figure 10a, general stress intensity factor solutions do not exist. Although there are flat plate solutions in the WinPraise manual [17], and cylindrical solutions for R/t=8 that were developed through the NURBIM program [18], the accuracy of those solutions relative to the low R/t values for primary piping is unknown. Therefore, for the xLPR Version 1.0 code, it was assumed that, as the surface crack penetrates through-wall, an idealized through-wall crack with the same crack area is formed. The assumption is non-conservative from a leakage perspective since the penetrating crack will leak at a much lower rate than an idealized through-wall crack with the same crack area due to the difference in the OD crack length. This difference will cause the leakage probability using the idealized through-wall crack assumption to be low as compared to that using the penetrating crack assumption.

5.10 Crack opening displacement

Calculation of Crack Opening Displacement (COD) has always been an essential aspect of ductile tearing fracture analysis for flawed, through-wall, cracked piping. It is also essential in the prediction of leakage through cracks, and is intimately related to the moment-tension carrying capacity of the flawed pipe. Every through-wall, cracked pipe stability analysis has a companion COD analysis, rooted in the same base assumptions as the stability analysis. In spite of these ties, COD calculations are independent of a stability analysis, affording the opportunity to select the “best” COD method with no regard to the related stability assessment approach.

There is a range of COD prediction performance as documented in Reference [19], where the GE/EPRI method is suggested to be the “best” method for calculating COD. The basis for the recommendation in Reference [19] was presented in Reference [20], where the details of a comparison of a variety of COD analysis methods with finite element calculations are presented. The conclusion was that the GE/EPRI method is the best choice for COD calculation. Based on these results and recommendation, the decision was made to use the GE/EPRI COD analysis in xLPR.

In their work originally performed for GE/EPRI on elastic-plastic fracture mechanics, Kumar and German [21] did not give any consideration to combined tension and bending loading cases. Subsequently, a combined tension and bending solution was added, but it has a very restricted range of applicability. In Reference [20], the authors describe a blending routine for combined tension and bending; however, errors were discovered in that solution. The work conducted within the xLPR program [4], which is currently coded into the Version 1.0 software, corrected the errors in the Reference [20] publication by independently deriving the equations for the blending routine.

5.11 Leakage

A review of existing thermal-hydraulic models [22] indicated that the Henry-Fauske model is the best currently available representation of fluid flow through tight cracks in a piping system. This model allows for non-equilibrium vapor generation rates as the fluid flows through the crack. The rate at which vapor is formed approaches the equilibrium value using an exponential relaxation correlation, with the correlation coefficients determined from the experimental data of Henry. As part of prior NRC-funded research, this methodology was encoded in a computer code called SQUIRT (Seepage Quantification of Upsets In Reactor Tubes), which predicts the leakage rate for cracked pipes in nuclear power plants. In all cases, the fluid in the piping system is assumed to be water at a given temperature and pressure. The Henry-Fauske model, used as a default in SQUIRT, is applicable to fluid that begins as sub-cooled liquid, and transitions to two phase flow as it passes through a tight crack.

There are two other models that can be employed depending on the size of the opening and the thermodynamic state of the fluid inside the pipe. The other two models are:

1. Single-phase liquid model. This model predicts the leakage rate through a pipe crack when the fluid inside the pipe is under pressure, but the fluid temperature is below the saturation temperature corresponding to the ambient pressure outside of the pipe. In this case, the fluid remains a liquid as it flows through the pipe crack and as it is discharged from the crack. This model solves the flow equations associated with non-compressible fluid flow.
2. Superheated, single-phase steam model. This model predicts the leakage rate through a pipe crack when the fluid inside the pipe is superheated steam. By definition, superheated steam has a steam quality of 100%. In this case, the fluid remains a gas as it flows through the pipe crack as it is discharged from the crack. This module solves the flow equations associated with compressible gas flow.

For the xLPR Version 1.0 code, the current version of SQUIRT, which contains the models listed above, plus a COD-crack morphology model, was used. For the pilot study, the crack morphology parameters (PWSCC) and the crack opening shape (elliptical) were held constant. Details of the SQUIRT module can be found in Reference [4]

5.12 Crack stability

The behavior of through-wall cracks and surface cracks in nuclear grade piping has been the subject of many experimental programs conducted by the NRC and industry. Many reports have been written, with the majority of the past research summarized in Reference [23]. This summary describes flaw stability in base metals and similar metal welds. Although limited research has been performed on flaw stability for DM welds, the NRC published a technical note [24] that recommends several methodologies for handling cracks in DM welds.

In cases of elastic-plastic fracture, which are prevalent for the pilot study problem, the operating stresses used to calculate the critical flaw size are independent of the local weld residual stress. The plasticity that forms during the deformation process eliminates the influence of the local weld residual stress. Therefore, for elastic-plastic crack stability, the total stress used in making critical crack determinations is given as:

$$\sigma_{CS} = \left(\sigma_{0DW} + \sigma_{0NTE} + \sigma_{0P}\right) + \sigma_{B-L} \tag{13}$$

5.12.1 Surface crack

Since Version 1.0 of the xLPR code is focused on high toughness Alloy 82/182 DM welds, it is appropriate to assume that the failure of a surface crack (low crack tip constraint) will be driven by net-section collapse. The methodology for net section collapse of circumferential surface cracks is described in detail in Reference [25], and the level of uncertainty compared to experiments is discussed in Reference [4]. For DM welds, the largest uncertainty in net-section collapse analyses is the use of appropriate material properties. Analyses have been conducted that suggest that a combination of the two adjoining base metal properties adjacent to a weld is appropriate for making critical surface crack predictions [26]. These analyses also suggest that if the crack is located near the stainless steel material, the stainless steel stress flow properties control the collapse.

Since the net-section collapse analysis is relatively simple to implement from a computational standpoint, the collapse bending load is predicted (assuming a semi-elliptical flaw) after each crack growth increment based on the material properties and pipe and crack geometry. If the operating bending loads are greater than the calculated net section collapse bending load, the surface crack is transitioned to an idealized, through-wall crack with the ID length equal to that of the critical surface crack[9].

5.12.2 Through-wall crack

As described in Reference [23], there are several estimation schemes that have been developed for analyzing the elastic-plastic fracture behavior of circumferential through-wall cracks in nuclear piping materials. For overall behavior, the LBB.ENG2 method has been shown to accurately predict the maximum moment for through-wall, cracked pipe experiments [4]. Therefore, this method was implemented into xLPR Version 1.0. The LBB.ENG2 estimation method proposed by Brust and Gilles [27] for evaluating the J-integral of cracked tubular members subjected to combined tensile and bending loads was used for assessing the stability of idealized through-wall cracks. The method of analysis is based on (1) classical deformation theory of plasticity, (2) a constitutive law characterized by a Ramberg-Osgood model, and (3) an equivalence criteria incorporating a reduced thickness analogy for simulating system compliance due to the presence of a crack in a pipe. The method is general in the sense that it may be applied in the complete range between elastic and fully plastic conditions. Since it is based on J-tearing theory, it is subject to the usual limitations imposed upon this theory, e.g., proportional loading, etc. This has the implication that the crack growth must be small although, in practice, J-tearing methodology is used far beyond the limits of its theoretical validity with acceptable results [23].

In some cases, due to the high toughness of the weld metal used in this study, net-section collapse may control the behavior. Therefore, in addition to the LBB.ENG2 method, the net

[9] This difference in the transition from surface crack to through-wall crack is due to the failure of the surface crack ligament.

section collapse solution from Reference [25] is also incorporated to Version 1.0 of the xLPR code.

For both the net section collapse and the LBB.ENG2 methods, the choice of the appropriate material properties to predict the failure in a DM weld is crucial. As described in Reference [26], a combination of strength properties (based on crack location) of both base metals, along with the weld-metal toughness, appears to predict the crack driving force accurately. However, in Version 1.0 of the code, the axial location of the crack was not considered, and the tensile properties of the stainless steel and the weld toughness properties are used in the analyses. This assumption is conservative because using the low strength stainless steel tensile properties will yield the smallest critical flaw size.

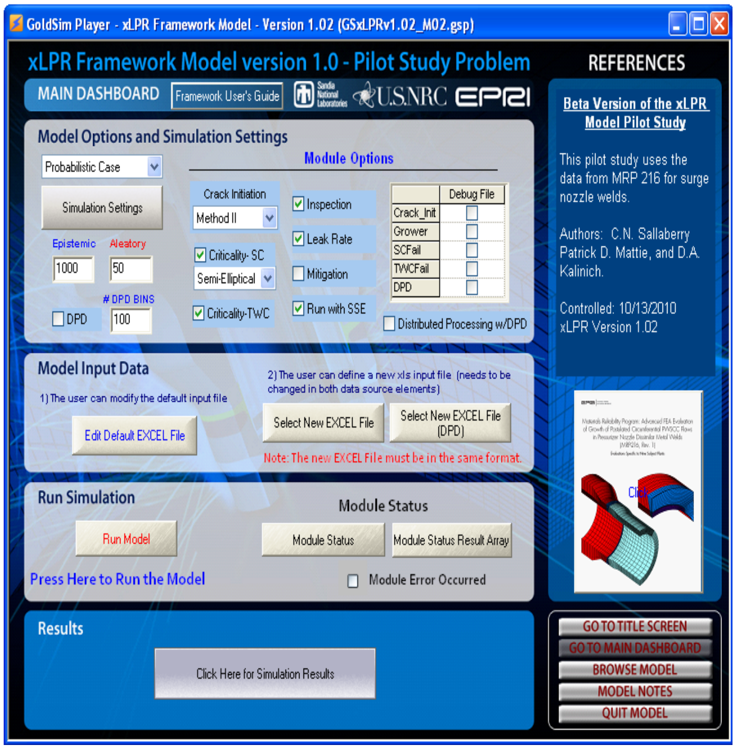
In determining stability, the code calculates the critical crack size margin using both the LBB.ENG2 (crack size) and the net-section collapse (moment) analyses for both normal and faulted conditions. For the normal operating loads, if the lowest margin is less than 1.0, the code assumes a double-ended break occurs. The break is recorded and the program exits the time loop. For emergency and faulted loads, the code records the failure, but does not exit the time loop.

6 Framework Descriptions

One of the objectives of the xLPR pilot study was to investigate the computational frameworks that are capable of meeting the objectives of this investigation. The choice of the appropriate computational framework is essential to assure that the code can be developed in a modular fashion and still be able to provide the structure for uncertainty handling. To meet this objective, both commercial and open source framework software were considered. In this section, a brief description of each of the two frameworks selected for use in developing xLPR Version 1.0 is presented. Details for each framework can be found in References [2,3].

6.1 GoldSim

The xLPR Version 1.0 model and pilot study case was constructed using existing fracture mechanics software coupled to a commercial software framework. After a comprehensive review of the commercial software available for this effort, the Version 1.0 xLPR model was constructed using the commercial software, GoldSim Pro, which is compatible with the free downloadable version of the GoldSim Player software. GoldSim Pro includes both a model developer’s version and the simulation software. GoldSim Player allows the user to view and navigate through the model logic, run an existing GoldSim model, and display the results, without having to purchase GoldSim Pro. The xLPR framework model player file was created such that key inputs to the model can be modified before running the code, as shown in Figure 11.



Model Options and Simulation Settings

Probabilistic Case

Epistemic 1000

Aleatory 50

DPD BINS 100

□ DPD

Module Options

Crack Initiation: Method II

☑ Criticality- SC: Semi-Elliptical

☑ Criticality-TWC

☑ Inspection

☑ Leak Rate

□ Mitigation

☑ Run with SSE

	Debug File
Crack_Init	□
Grower	□
SCFail	□
TWCFail	□
DPD	□

□ Distributed Processing w/DPD

Model Input Data

1) The user can modify the default input file

2) The user can define a new xls input file (needs to be changed in both data source elements)

Note: The new EXCEL File must be in the same format.

Run Simulation

Press Here to Run the Model

Module Status

□ Module Error Occurred

Beta Version of the xLPR Model Pilot Study

This pilot study uses the data from MRP 216 for surge nozzle welds.

Authors: C.N. Sallaberry Patrick D. Mattie, and D.A. Kalinich.

Controlled: 10/13/2010
xLPR Version 1.02

Figure 11 GoldSim xLPR Framework Model Dashboard

The GoldSim software is dynamic, probabilistic simulation software developed by GoldSim Technology Group, LLC. This general-purpose simulator is a hybrid of several simulation approaches, combining an extension of system dynamics with some aspects of discrete event simulation, and embedding the dynamic simulation engine within a Monte Carlo simulation framework [28].

The modular-based GoldSim framework model for the Version 1.0 xLPR model manages input variables (e.g., material properties) and model output (e.g., results), as well as the flow of information that includes the system level model logic. The GoldSim framework for xLPR was constructed with an option to use standard Microsoft Excel spreadsheets to define the inputs as well as dynamically pass simulation results to Microsoft Excel for advanced post-processing. The commercial framework simulation software serves as the integrating shell that links various modules used in the xLPR Model.

The GoldSim xLPR model framework controls the order in which the modules are called and the passing of variables into and out of modules. The xLPR approach is to create all of the modules independently, so that the modules can be created by collaborators in any programming language. Both simple and complex calculations are coded as modules and then are directly coupled to the xLPR GoldSim framework using dynamic link libraries (DLLs) by wrapping the original module source code in a simple standard DLL shell [28], as shown in Figure 12.

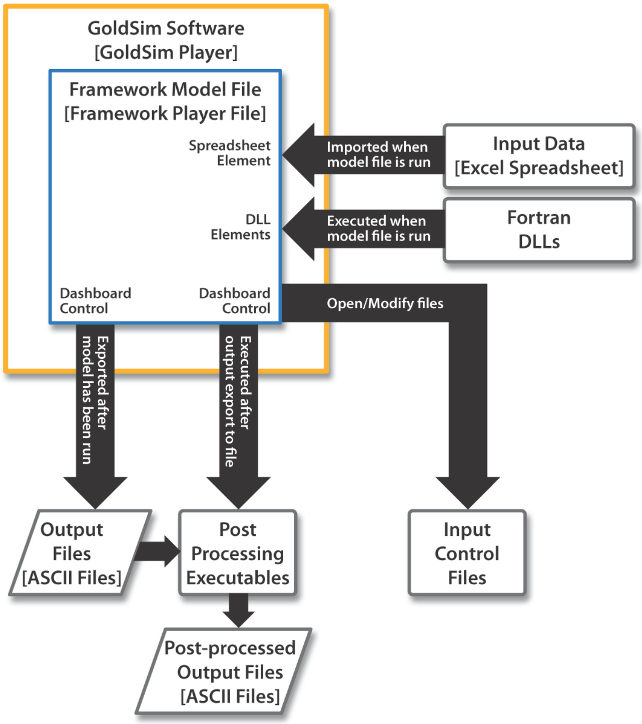


Figure 12 GoldSim xLPR Model Framework

The framework utilizes the GoldSim software libraries of probability distribution functions and the capability to correlate variables and perform multiple-realization stochastic analyses in a Monte Carlo approach. The framework benefits from the GoldSim software's ability to store simulation data from large numbers of realizations and generate statistics on global probability distributions. GoldSim permits each run to be saved in a single action, including all input data and results from Monte Carlo analysis. Finally, the GoldSim framework has built in graphical user interface (GUI) functions that allow the developer to quickly assemble specific model runs and to create interactive player files for end-users, which allow for viewing, navigating, and even modification of input values and model options to run the xLPR model without requiring a software license, using the free GoldSim Player software application [28].

The GoldSim software provides a visual and hierarchical modeling environment, in which the xLPR framework model was constructed by adding “elements” (model objects) native to the software that represent data, equations, module interface, processes or events, and linking them together into graphical representations that resemble influence diagrams. Influence arrows are automatically drawn as elements are referenced by other elements. The complex xLPR systems can be translated into hierarchical GoldSim models by creating a layer of “containers”. Visual representations and hierarchical structures help users to build very large, complex models that can still be explained to interested stakeholders (e.g., government regulators, elected officials, and the public).

In addition, the GoldSim framework for xLPR includes the software’s ability to track changes that have been made to a model file. This feature (referred to as versioning) allows the differences between the current version and a previous version of a model file to be quickly determined [28]. The version history is an integral part of the model file, providing an easy-to-access history of all of the changes that have occurred over the life of the model. Providing this configuration management capability is particularly useful for coordinating model changes when multiple people have the ability to access and modify the same model file, and as a Quality Assurance/Quality Control feature allowing for verification and documentation of where and when changes have been made to a model file. Details of the GoldSim framework can be found in Reference [2].

6.2 SIAM-PFM

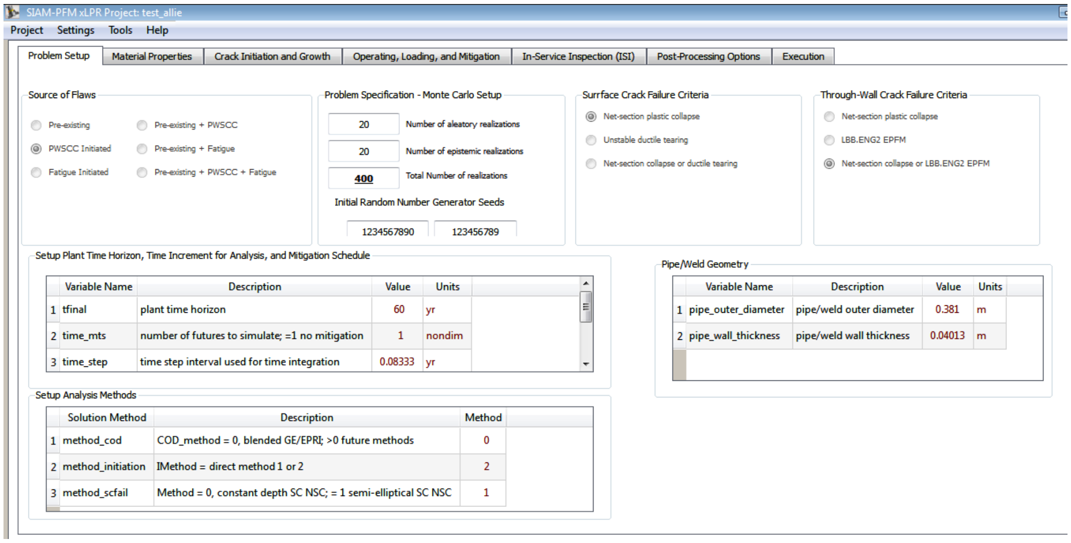
The SIAM-PFM Framework is a problem-solving environment. The acronym SIAM-PFM (SIAM for short) stands for Structural Integrity Assessment Modular – Probabilistic Fracture Mechanics. SIAM-PFM is an Object Oriented Open Source (OOOS) framework, within which a wide range of nuclear power plant safety issues can be addressed in a systematic and consistent way by using modern principles of probabilistic risk assessment. Probability techniques are applied to problems in fracture mechanics in order to predict fracture behavior, and thus to assess the structural integrity of a variety of nuclear power plant components that passively bear large loads over long periods of time. This platform is readily extensible to different problem classes. A common feature of the different applications is that they are all the subjects of probabilistic risk assessment and, therefore, represent “risk-informed” analyses.

Every SIAM-PFM component is written using the Python programming language and Python frameworks. They are easily installed and un-installed on Windows operating systems, and all components are potentially portable to other operating systems such as UNIX and OS X.

In the SIAM-PFM problem solving environment, all components use the same working principle: workspaces that contain projects (SIAM projects), or directories in which all inputs and outputs of a given test case, are saved in both binary and in text files. Users can navigate through the different projects in the project-explorer panel as on any IDE (Integrated Development Environment) or windows explorer and create projects that represent test cases. Convenient plots of the outputs are also provided to visualize data, and users can also extract the raw data in text files to create custom plots.

The implementation of xLPR within SIAM (SIAM-xLPR) presents a series of tabs to define the case conditions. In these tabs, pre-defined input values have been set by default to the probabilistic base case (see Figure 13). The SIAM-xLPR main GUI framework has seven tabs. Input data can be entered on the first six tabs, in any order. Default input values are provided, and can be modified as needed. The seventh tab displays the SIAM-xLPR “Execute Utility” window that presents a command line view of program executions, where realizations are created and executed according to the program flow, as discussed in Section 4. Details of the SIAM framework can be found in Reference [3].

The SIAM-xLPR framework has been developed primarily using the Python v.2.6 programming language and Python frameworks scipy, numpy and PyQt.


SIAM-PFM xLPR Project: test_allie
Project
Settings
Tools
Help
Problem Setup
Material Properties
Crack Initiation and Growth
Operating, Loading, and Mitigation
In-Service Inspection (ISI)
Post-Processing Options
Execution

Source of Flaws

○ Pre-existing
● PWSCC Initiated
○ Fatigue Initiated
○ Pre-existing + PWSCC
○ Pre-existing + Fatigue
○ Pre-existing + PWSCC + Fatigue

Problem Specification - Monte Carlo Setup

20 Number of aleatory realizations
20 Number of epistemic realizations
400 Total Number of realizations

Initial Random Number Generator Seeds

1234567890 123456789

Surrface Crack Failure Criteria

● Net-section plastic collapse
○ Unstable ductile tearing
○ Net-section collapse or ductile tearing

Through-Wall Crack Failure Criteria

○ Net-section plastic collapse
○ LBB.ENG2 EPFM
● Net-section collapse or LBB.ENG2 EPFM

Setup Plant Time Horizon, Time Increment for Analysis, and Mitigation Schedule

	Variable Name	Description	Value	Units
1	tfinal	plant time horizon	60	yr
2	time_mts	number of futures to simulate; =1 no mitigation	1	nondim
3	time_step	time step interval used for time integration	0.08333	yr

Pipe/Weld Geometry

	Variable Name	Description	Value	Units
1	pipe_outer_diameter	pipe/weld outer diameter	0.381	m
2	pipe_wall_thickness	pipe/weld wall thickness	0.04013	m

Setup Analysis Methods

	Solution Method	Description	Method
1	method_cod	COD_method = 0, blended GE/EPRI; >0 future methods	0
2	method_initiation	IMethod = direct method 1 or 2	2
3	method_scfail	Method = 0, constant depth SC NSC; = 1 semi-elliptical SC NSC	1

Figure 13 SIAM-xLPR presents a series of tabs to define the case conditions

7 Pilot Study Problem Statement

To demonstrate the feasibility of conducting analyses to assess the probability of rupture in pressurizer surge nozzles resulting from PWSCC, and to compare the results from the two frameworks developed in this effort, a pilot study problem statement was developed. This problem statement consisted of two deterministic analyses, a probabilistic base case, and a series of sensitivity analyses that are intended to demonstrate the features of the Version 1.0 code. The detailed problem statement for the pilot study is in Appendix B.

To verify the two framework codes are performing the deterministic calculation correctly, two separate deterministic analyses were defined:

> Deterministic Analysis #1: Single Crack at time = 0 years, with no mitigation. The location of the crack is at the top of the weld (ϕ = 0 rad). The input information for this case can be found in Appendix C.
>
> Deterministic Analysis #2: Three Cracks at time = 0 years, with no mitigation. This analysis is an extension to the first deterministic analysis that assumes three cracks. The three cracks are the same size as in Deterministic Analysis #1, with respective locations ϕ = 0 rad., ϕ = 0.6 rad and ϕ = -1 rad. The input information for this case is otherwise identical to Deterministic Analysis #1, and can be found in Appendix C.

In addition, one base case and five sensitivity cases were analyzed for the pilot study, and are summarized in Table 3. The analyses were conducted using the controlled versions of both the GoldSim and SIAM framework models developed for the xLPR pilot study.

Table 3 xLPR Version 1.0 Analyses.

Analysis	Description
Probabilistic Base Case	Probabilistic base case analysis using Monte Carlo sampling.
Sensitivity Study	
Stress Mitigation	Analyses evaluate different mitigation times, for the same stress-based mitigation.
Chemical Mitigation	Chemical effects of increasing the hydrogen concentration in the water on the crack growth module. Three hydrogen concentrations were evaluated.
Crack Initiation	Considers the crack initiation model uncertainty.
Safe End Evaluation	Considers stainless steel safe end weld, which causes a through-thickness bending stress that can reduce the tensile inner-diameter stress.
Importance Sampling	DPD analysis with importance sampling using the Safe End Evaluation analysis option.

For the sensitivity cases, the following inputs varied from those listed for the base case in Appendix C:

- Stress mitigation
  - Three cases were run with mitigation at time 10, 20, and 40 years.
  - Revised weld residual stress inputs:
    - ID weld residual stress: mean value of -344.75 MPa and a standard deviation of 34 MPa with a minimum value of -447 MPa and a maximum value of -242 MPa).
    - X_c: mean value of 0.38 and a standard deviation of 0.038, with a minimum value of 0.26 and maximum value of 0.5.
- Chemical Mitigation
  - Two cases, where the base case hydrogen value of 25 cc/kg-STP was increased to 50 and 80 cc/kg-STP, respectively.
- Crack Initiation
  - Direct Model I was used in place of Direct Model II
- Safe End Evaluation
  - Revised weld residual stress inputs:
    - ID weld residual stress: mean value of -16.2 MPa and a standard deviation of 117 MPa with maximum value of 300 MPa and a minimum value of =-300MPa.
    - X_c: mean value of 0.18 and a standard deviation of 0.036 with maximum value of 0.5 and a minimum value of 0.1.

In addition to the sensitivity analyses, the base case and the safe end evaluation case were both post-processed to take credit for leak detection and inspection. For each case, leak rate detection limits of 0.1, 1, 10 and 50 gpm were considered. The inspection intervals assumed were 5, 10, 20 and 30 years.

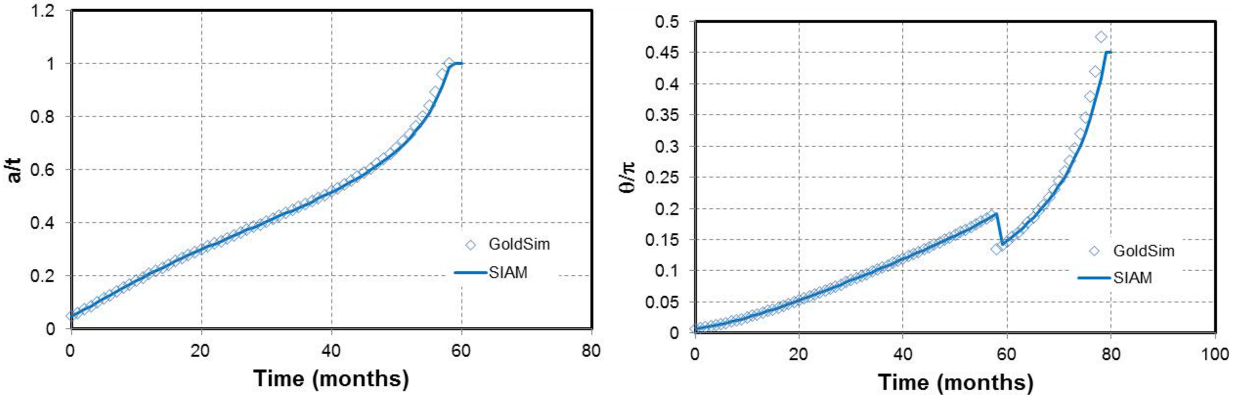
During the development of this report, several errors were discovered in framework, modules, and input decks for the pilot study problem. An impact assessment was completed to document and significant changes that may have occurred in the results due to these errors. In most cases, the errors lead to less than a factor of two change in the probability of rupture. The largest changes impacted the crack initiation model and the effects of hydrogen on the leakage and rupture probabilities. Details of the impact assessment can be found in Appendix D.

8 Pilot Study Results

In this section of the report, the results from the pilot study problem are presented and discussed. These comparisons include results from both the GoldSim and SIAM versions of xLPR. In addition to the comparisons in this section, initial benchmark testing of GoldSim and SIAM with the WinPraise code can be found in Reference [29].

8.1 Deterministic results

As mentioned above, two deterministic problems, developed to verify that the codes are performing correctly, were run on each code and no major differences were observed. The results from these two deterministic problems, each of which contained a single crack located at the top of the pipe, are shown in Figure 14.



(a) Normalized Crack Depth **(b) Normalized Crack Length**

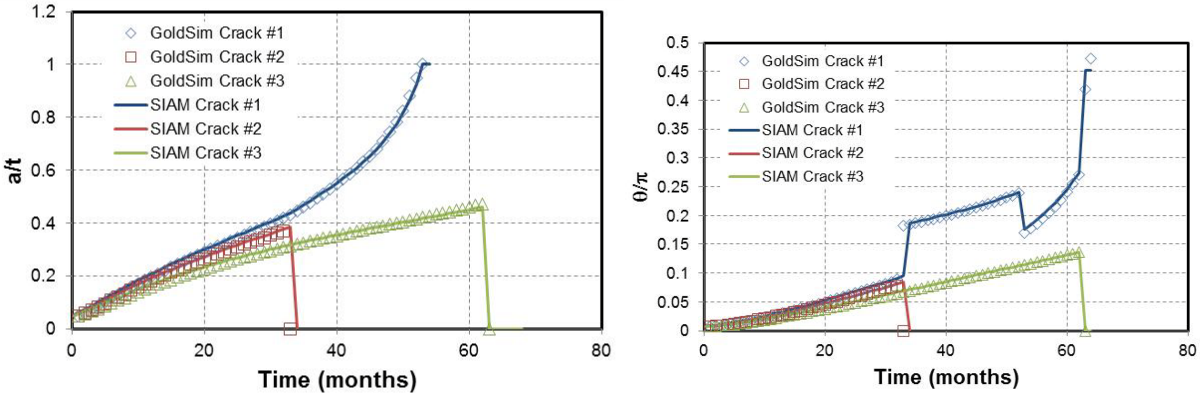
Figure 14 Crack size results as a function of time for the first deterministic problem

In Figure 14, the solid line represents the results from SIAM, while the symbols represent the results from the GoldSim framework. Note that the drop in crack length seen in Figure 14b is due to the crack transition assumption discussed in Section 5.9. The results illustrate that the two framework codes produce very comparable results. In fact, the GoldSim framework predicted that the crack will penetrate through-wall (a/t=1.0) in 58 months, while the SIAM framework predicted 59 months. A similar comparison resulted for rupture; the Goldsim framework predicted 79 months, while the SIAM framework predicted 80 months. Similarly, as shown in the normalized crack length plot (Figure 14b), the crack length at rupture is slightly different between the two codes. The GoldSim framework predicted 47.5% of the circumference, while the SIAM framework predicted 45.1% of the circumference. This difference is due to the time at which the crack size is updated in each code, i.e., at the beginning of the time step (Goldsim) or at the end of the time step (SIAM)[10].

The results from the second deterministic problem are shown in Figure 15. In this problem, three cracks were initiated at time zero and allowed to grow until rupture. The normalized crack depth results (Figure 15a) are very similar to those for the first problem, and also show the difference of one time step between the GoldSim and SIAM results that was illustrated in the

[10] Differences in the time loop development between the GoldSim and SIAM framework are discussed in References [2,3], while a comparison of the code differences is described in Reference [6].

first problem. The normalized crack length results illustrate the coalescence of cracks 2 and 3 into crack 1. The overall comparison between the two codes is excellent.



(a) Normalized Crack Depth **(b) Normalized Crack Length**

Figure 15 Crack size results as a function of time for the second deterministic problem

8.2 Base case

The results from the Version 1.0 base case analyses are summarized in this section of the report. The inputs for these analyses are included in Appendices B and C. The uncertainty categorization for each input variable is also shown in Appendix B, and is described in detail in Reference [4]. Results for the base case were generated by both the GoldSim and SIAM versions of the xLPR Version 1.0 code. Throughout this section, comparisons between the results from the two codes are presented. In certain cases, results from only one code are shown since the trends between the codes were similar. When this occurs, the results from the code that is used in any particular figure are identified by a model result designation, i.e., GSxLPRv1.0 (GoldSim) or SIAM_v1.0 (SIAM). For the base case results presented in this report, the total number of realizations was chosen to be 50,000 (1,000 epistemic and 50 aleatory).

8.2.1 Probability of first crack

The probability of first crack initiation is controlled by the crack initiation module. Per Appendix C and Reference [4], all of the input uncertainties for the crack initiation model, except for the initiated crack length, are considered epistemic. The results of the probability of crack initiation are shown in Figure 16. In this figure, the light vertical grey lines represent the probability of initiation for each epistemic realization, i.e., representing aleatory uncertainty. In addition, the mean and standard quantiles are also shown in Figure 16. For this case, the mean value suggests that there is a 47% chance of initiating a crack in 60 years (720 months). These results illustrate that there are many realizations where cracks do not initiate.

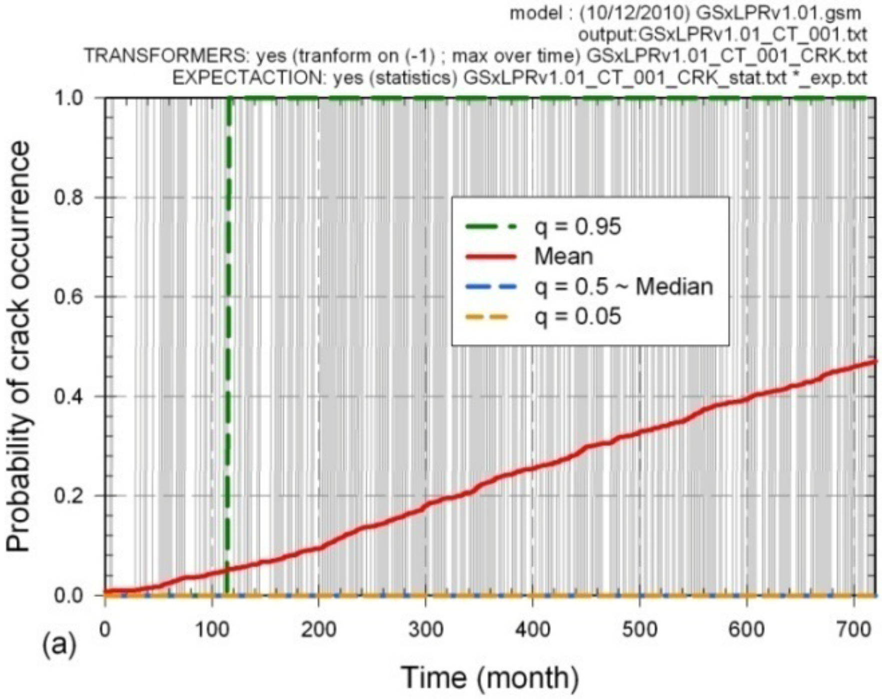


Figure 16 Probability of crack initiation as a function of time for the base case

This fact is further illustrated by the lack of data in the lower quantiles. For both the 50th and 5th percentile, the probabilities are zero, which indicates that more than 50% of the time, there are no initiation times less than 60 years. The 95th percentile increases from 0 to 1 around 120 months, which means that after about 10 years, there is at least 5% of the results with at least one crack.

As mentioned previously, the vertical grey lines in Figure 16 represent each individual epistemic realization. In this case, for each epistemic realization, there were 50 aleatory realizations. The fact that each grey line is vertical indicates that for each epistemic realization, all 50 aleatory realizations either initiated a crack or did not initiate a crack, i.e., there was no effect of the aleatory uncertainty. In addition, the fact that the vertical lines occur well distributed across the 60-year time period indicates that the epistemic uncertainty is controlling the behavior.

A comparison of the probability of crack initiation for both the GoldSim and SIAM framework models is given in Figure 17. Each model was run with the same number of realizations (50,000 corresponding to 1000 epistemic and 50 aleatory realizations). The comparison between the codes is very good, with the SIAM results being slightly lower than the GoldSim results for time periods less than 10 years.

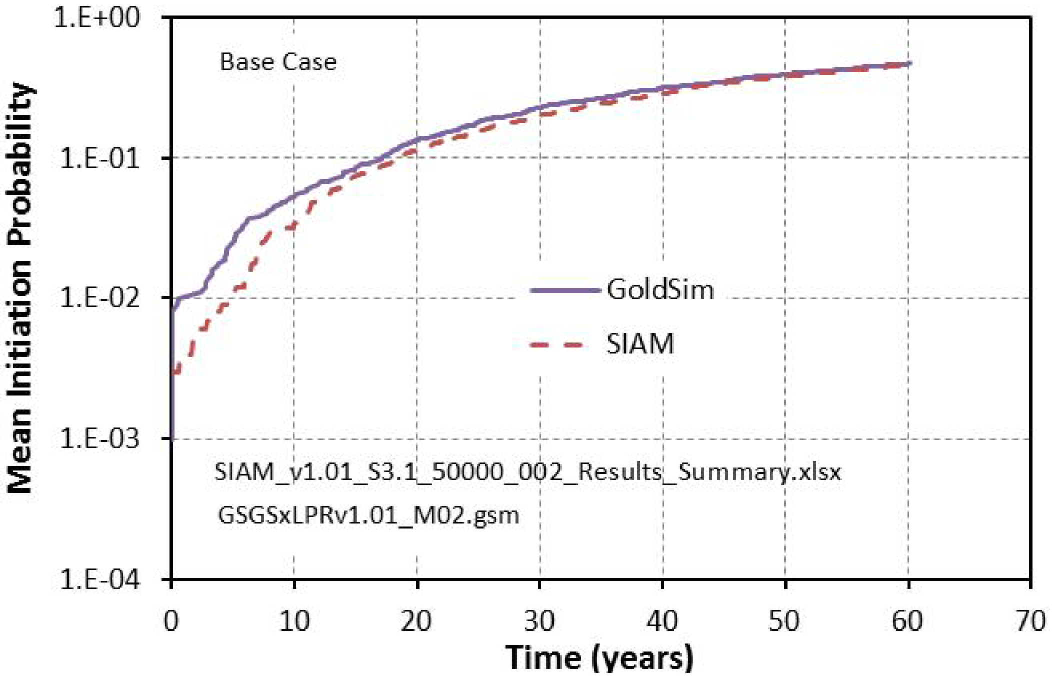


Figure 17 Comparison of GoldSim and SIAM Framework for mean probability of crack initiation

8.2.2 Fractional surface area cracked

The fractional surface area cracked is equal to the percentage of the total pipe cross sectional area that is cracked. This value includes all active cracks for each time period. This variable allows a quick view into how the cracks are progressing. The expected value of the fractional surface area cracked as a function of time for the base case is shown in Figure 18. This data are presented in the same fashion as the initiation data described in the previous section.

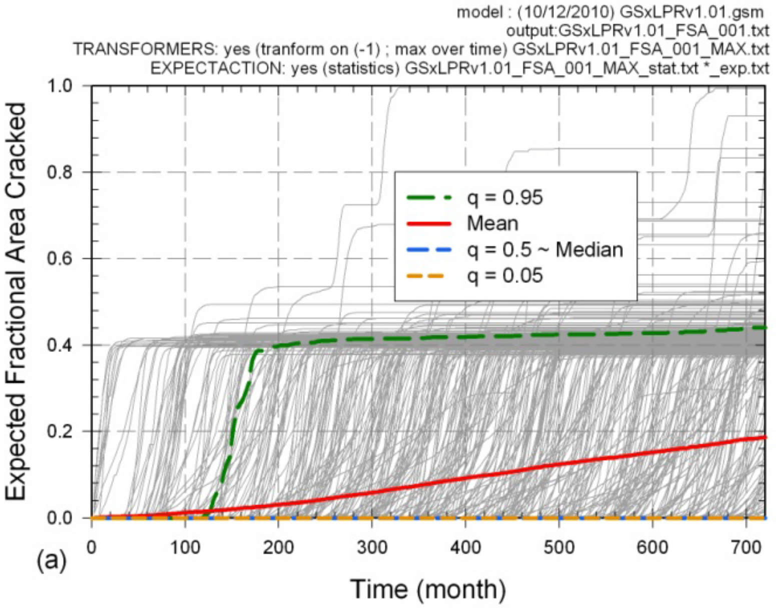


Figure 18 Fractional surface area cracked as a function of time

From Figure 18, the epistemic realizations show that for most cases, the pipe cracked to about 40% of the total cross sectional area of the pipe. This corresponds well with the expected critical though-wall crack size for the mean material property data. Approximately 2% of the cases lead to a higher cracked fractional surface area, and the cases correspond to long and thin surface cracks that will lead to rupture at the same time they become through-wall cracks. These cases are important when calculating low probability of rupture that is conditional on leak rate detection and/or inspection.

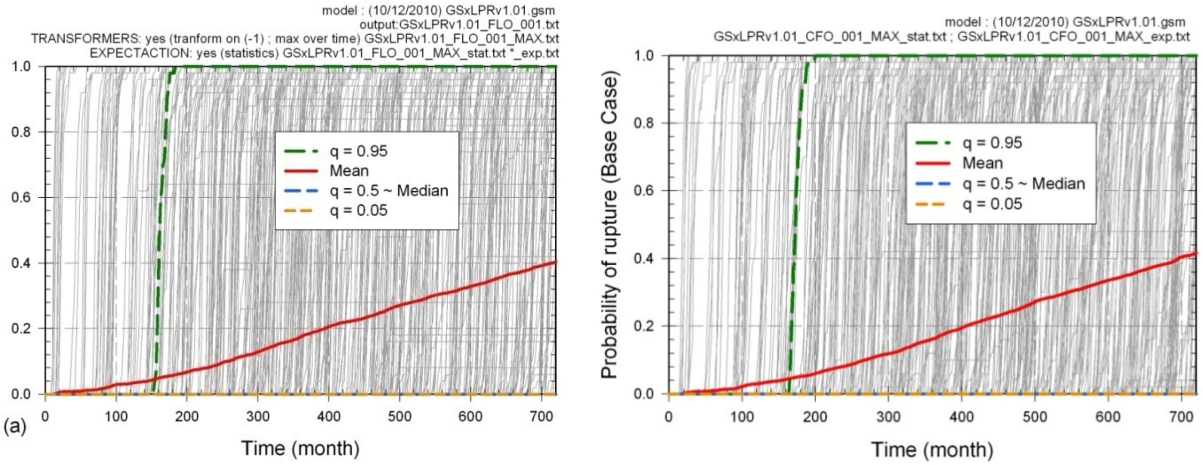
As with the initiation results, a data median value equal to 0 shows that no cracks occurred at least 50% of the time. Because of this fact, the mean value of the expected cracked fractional surface area is approximately 0.2 after 60 years.

8.2.3 Probability of leakage and rupture

In this section, the calculated probabilities of leakage and rupture for the base case are presented. The probability of leakage is separated into the following three categories:

- Through-wall penetration, i.e., first leakage
- COA > 1 – Crack opening area greater than the equivalent break diameter of 1-inch (506.7 mm^2 [0.78in^2])
- COA > 3 – Crack opening area greater than the equivalent break diameter of 3-inch (4,560.4mm^2 [7.1 in^2])

The base case results for probability of first leakage and rupture are shown in Figure 19. The graphs shown in this figure appear very similar to the probability of crack initiation results shown in Figure 17. This similarity suggests that the crack initiation is controlling the rupture and leakage behavior. Again, the straight grey lines, which represent the epistemic realizations, suggest very little contribution from the aleatory uncertainty. Also, for at least 50% of the realizations, there was no leakage or rupture.



(a) First Leakage **(b) Rupture**

Figure 19 Probability of leakage and rupture for the base case

A comparison of the mean probability values of leakage and rupture is shown in Figure 20. These results indicate that there is very little difference between the leakage and rupture behavior for this case. By investigating individual realizations, it is seen that, typically, for the inputs used for this problem, a crack will initiate and grow to rupture in less than 10 years, with only approximately 2 years between first leakage and rupture. The PWSCC growth is significant for the input operating conditions, which explains the relatively small time difference between first leakage and rupture.

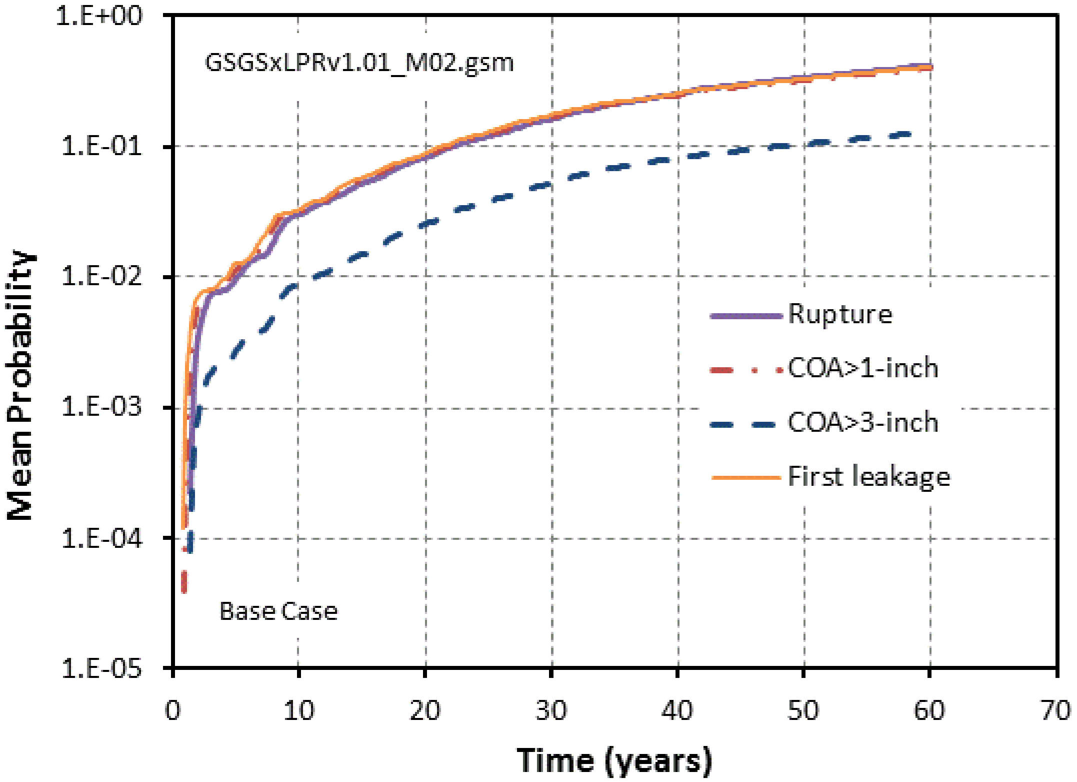


Figure 20 Mean probability of leakage and rupture for the base case

Of interest is the observation that the mean probability of having a COA greater than an equivalent pipe diameter of 3-inch is much lower than either the leakage or rupture probabilities. This difference is explained by investigation of the distribution of crack opening area at rupture, as shown in Figure 21. This figure suggests that the median value of COA at rupture is equivalent to a 2.4-inch diameter pipe. For a 3-inch diameter opening or greater, the probability of occurrence is less than 30%, i.e., 30% of the rupture that occurred had an opening greater than 3-inch equivalent diameter. For a 1-inch diameter opening or greater, the probability of occurrence is 93%. These values correspond well with the probabilities shown in Figure 20.

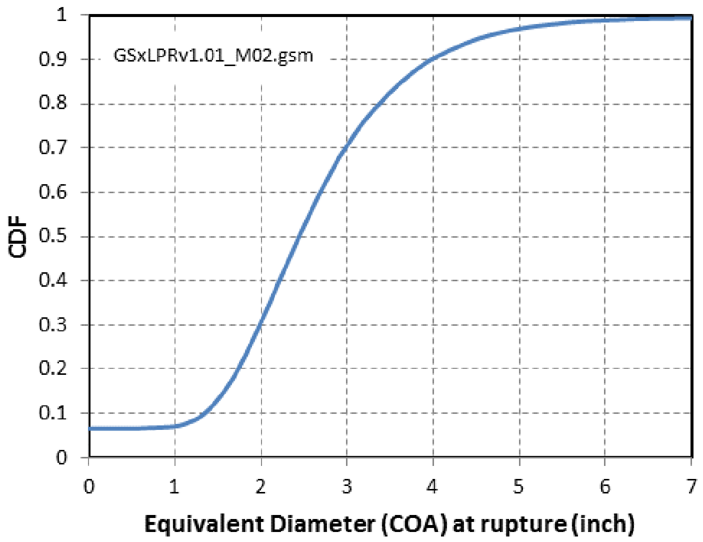


Figure 21 Distribution of equivalent diameter crack opening area at rupture

A comparison of the mean probability of rupture using the GoldSim and SIAM xLPR framework codes is shown in Figure 22. The results show a good comparison between the codes, with the main difference resulting from the difference in the crack initiation probabilities described above.

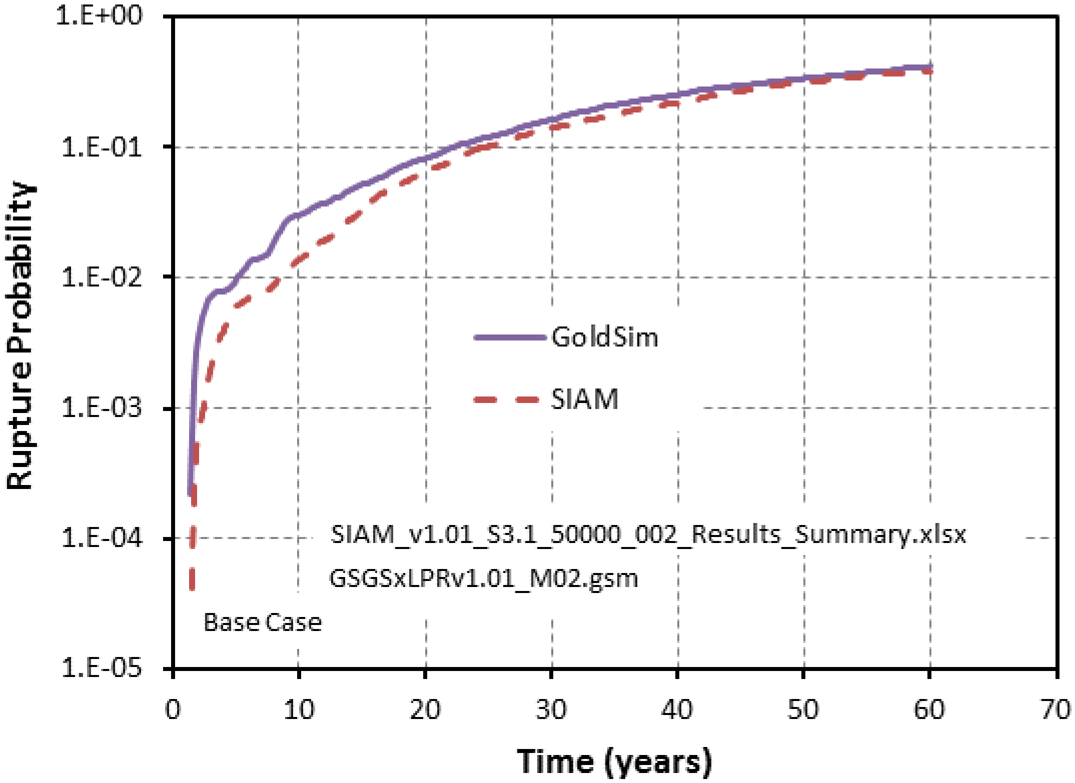


Figure 22 Comparison of mean probability of rupture for the base case between GoldSim and SIAM

The effect of both the epistemic and aleatory uncertainties was investigated by considering the Complementary Cumulative Distribution Functions (CCDF) at certain times. CCDFs illustrate how much variation is due to aleatory uncertainty (spread within a CCDF) and epistemic uncertainty (spread between CCDFs). The CCDF for the probability of rupture for the base case is shown in Figure 23. Within the code, for any single realization, either rupture occurs (P = 1) or it does not occur (P = 0). In this figure, the CCDF is relatively flat, i.e., the probability of rupture quickly changes from 0 to 1. This illustrates that the aleatory uncertainty has little effect on the probability of rupture. At 60 years, there is a 55% chance that no future will lead to rupture (P = 0), and there is a 39% chance that all futures will lead to rupture (P = 1). This observation demonstrates that the aleatory uncertainty only affects 6% of the results.

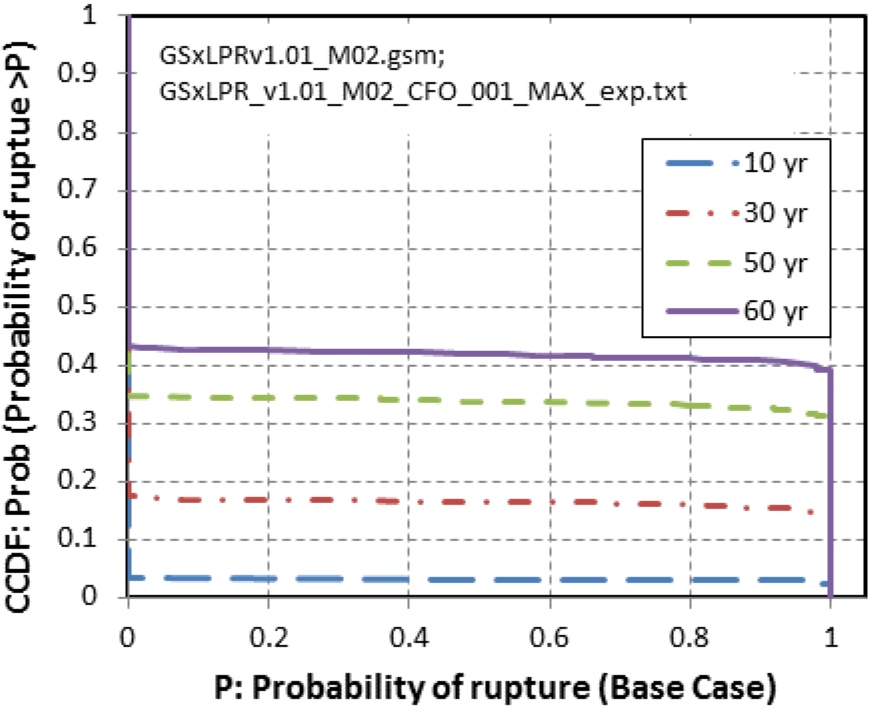


Figure 23 CCDF for probability of rupture at a variety of times

8.2.4 Credit for inspection

To demonstrate the benefit that may be obtained from inspection for this problem, inspection intervals of 10, 5, 3, and 2 years were considered[11]. The resulting probabilities of rupture are shown in Figure 24. The results from this figure illustrate that, for this problem, the effects of in-service inspection on the rupture probability are small. In fact, a 2-year inspection interval decreases the probability of rupture by only one order of magnitude. In addition, these results are slightly non-conservative since the inspection module for the pilot study assumes that, if a flaw is detected, it is fully repaired, i.e., it does not account for a poor repair, and ineffective repair, or the addition of fabrication flaw that may be introduced by the repair process. The reason for the small change in rupture probability due to inspection is illustrated in Figure 25. In this figure, the distribution of time from initiation to failure for all cracks is shown. The results

[11] These inspection intervals are slightly different than presented in the problem statement since shorter intervals were needed to demonstrate a benefit of inspection due to the high PWSCC growth rates

from this figure show that about 85% of the time, a crack will fail within 10 years of its initiation time. There is also a 96% chance that a flaw would rupture within 20 years. Therefore, for a 10 year inspection period, there is a very low probability that a flaw would be active for two 10-year inspection periods, as most flaws would rupture within this time due to the high crack growth rate.

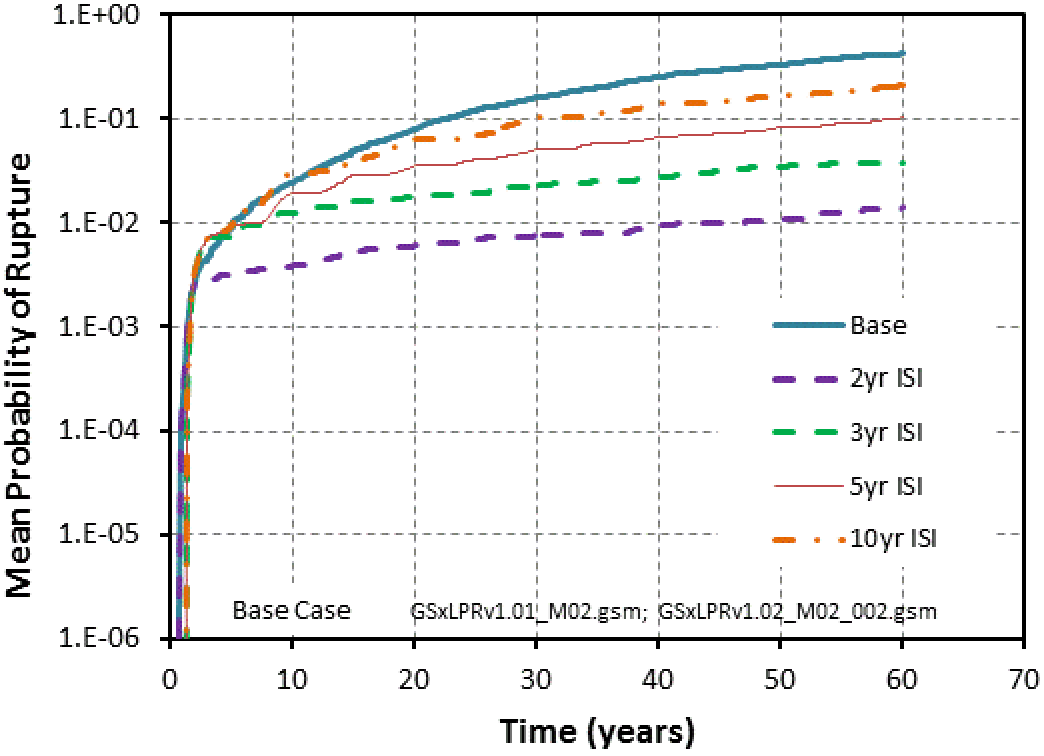


Figure 24 Mean probability of rupture with inspection for base case

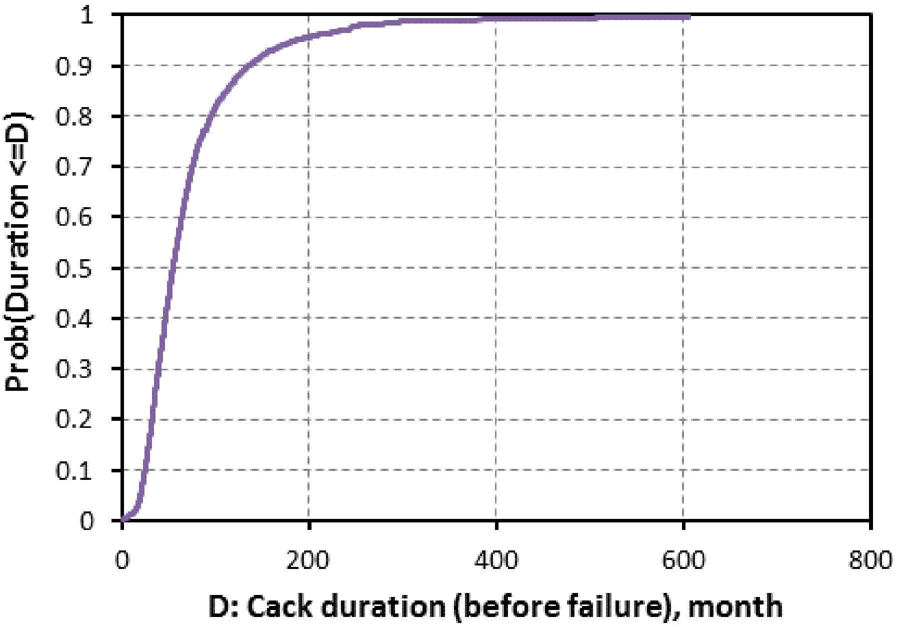


Figure 25 Probability of crack duration before failure

8.2.5 Credit for leak detection

As with inspection, the credit for leak detection can be demonstrated for a variety of leak detection limits, i.e., 50, 10, 1 gallons per minute (gpm). The mean probability of rupture for the base case with leakage detection is shown in Figure 26. This figure illustrates that the benefit for leak detection is about one order of magnitude, and is insensitive to the leak detection limit. The probability of rupture results for the 1 and 10 gpm leak detection limits are seen to be identical, and the probability of rupture for the 50 gpm leak detection limit is only slightly higher.

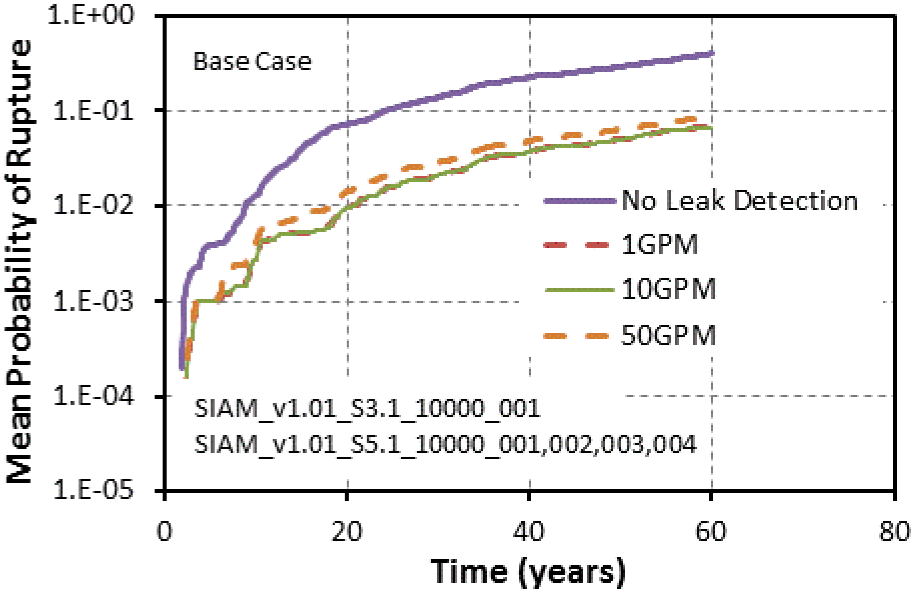


Figure 26 Mean probability of rupture with leak detection for the base case

Investigating the leakage probabilities aids in understanding the results shown in Figure 26. The distribution of leakage at the time of through-wall crack penetration, i.e., first leak, is shown in Figure 27. This figure illustrates that there are many cases where the initial leakage is very high. In fact only 7% of the cases that leak have a rate less than 10 gpm at first leakage. For the base case, the lowest recorded first leakage was 2.2 gpm. Therefore, there is no difference in the probability of rupture with leak detection limits of 1 and 10 gpm. The leakage rate distribution immediately before rupture is also shown in Figure 27. In this case, about 6% of the realizations ruptured as soon as they became through-wall cracks, i.e., there was no leakage before rupture was simulated. Also, the smallest leakage recorded before rupture was 14 gpm, with only 6.7% of the cases having a leak rate at rupture of below 50 gpm.

The large first leak rates can be attributed to the model for surface crack to through-wall crack transition. As described in Section 5.9, when a surface crack penetrates the wall thickness, the resulting through-wall crack is created with radial crack ends such that the area of the through-wall crack matches that of the surface crack at the time of first leakage. This assumption causes a small drop in the ID surface length of the through-wall crack, and the through-wall OD crack length becomes longer than the ID crack length. This assumption results in a relatively long through-wall crack. Realistically, the OD through-wall crack length will be relatively short compared to the ID crack length at first leakage, thereby giving a non-idealized through-wall crack shape. Since there are no published stress intensity solutions for this type of crack, it was ignored in the pilot study.

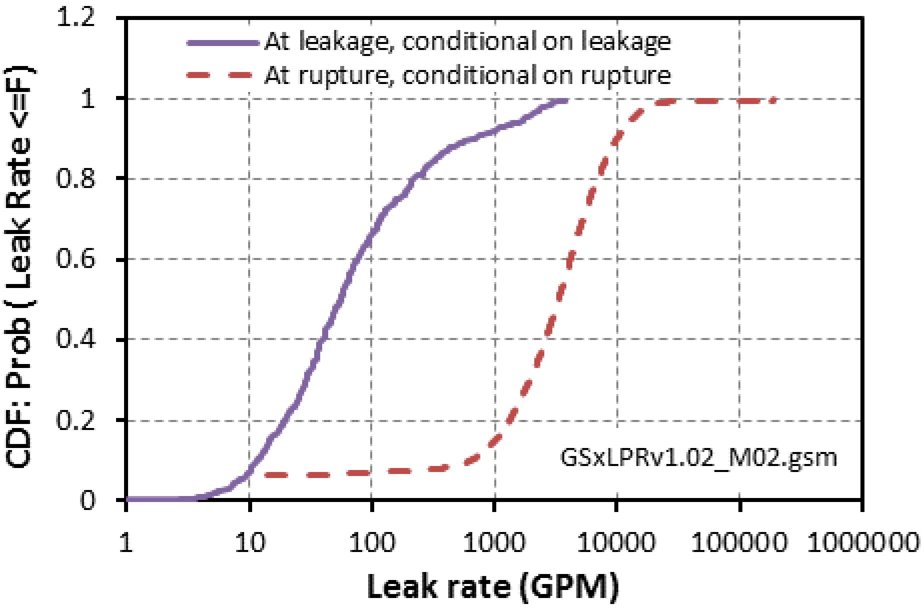


Figure 27 Distribution of leak rates for the base case.

The effect of the combination of leakage detection and inspection on the base case mean probability of rupture is shown in Figure 28. These results illustrate an almost four-order of magnitude decrease in the probability of rupture when credit is taken for in-service inspection and leak detection.

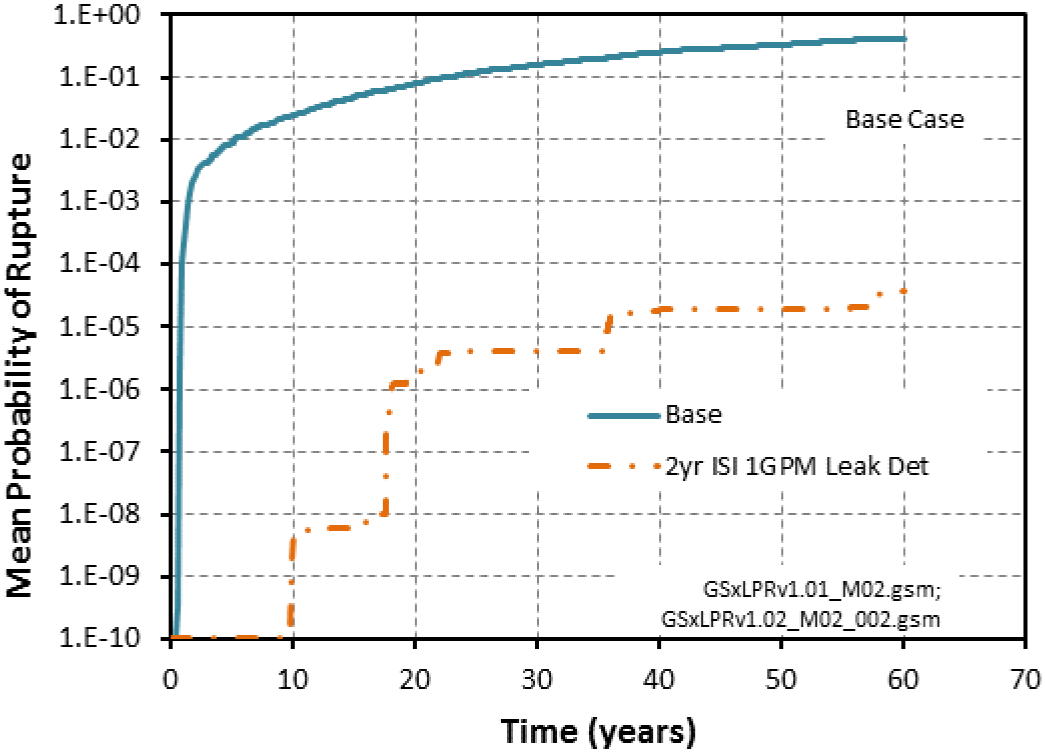


Figure 28 Effect of inspection and leak detection on base case mean probability of rupture

8.2.6 Parameter sensitivity analyses

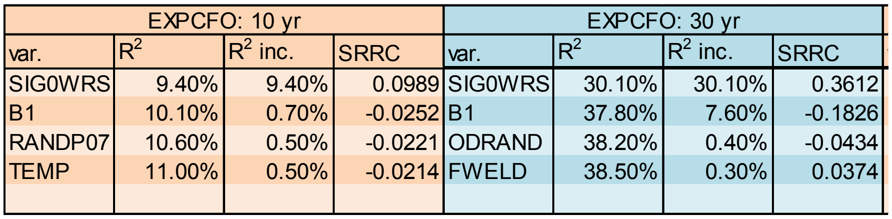
Parameter sensitivity analysis refers to the determination of the contributions of individual, uncertain analysis inputs (parameters) to the uncertainty in analysis results. To quantify and rank the importance of the variance of each uncertain input on the variance of the output of interest, linear and rank regressions were judged to be the most appropriate. Because rank regression often gives better results than linear regression for essentially the same computing cost, Standardized Rank Regression Coefficients (SRRCs) were used in place of their parametric linear equivalent. The coefficient of determination of the regression model (R^2) was used to provide information about the quality of the regression and, consequently, of the quality of the sensitivity analysis. The approach that was the most appealing graphically was the case where the importance of some parameter was estimated in a stepwise fashion (i.e., stepwise regression) for non-time-dependent parameters, and at a specific time-step for time-dependent parameters.

A complete description and the details of the xLPR Version 1.0 parameter sensitivity study conducted are included in Reference [2]. For the purposes of this report, only the uncertainty influencing the probability of rupture is presented. The analysis was conducted at 10 years, 30 years, 50 years and 60 years for the base case analysis. The results are shown in Table 4 and Table 5.

At each time step analyzed, the variable name (in order of importance), the cumulative R^2 (how much of the output variance is explained with the current input and all previous inputs), the incremental R^2 (how much variance is explained by the addition of this input), and SRRC are listed. A positive SRRC denotes a positive relation (in the sense that high values of input are associated with high values of output and low values of input are associated with low values of input), while a negative SRRC denotes a negative relation (for which high values of input are associated with low values of output and low values of input are associated with high values of output).

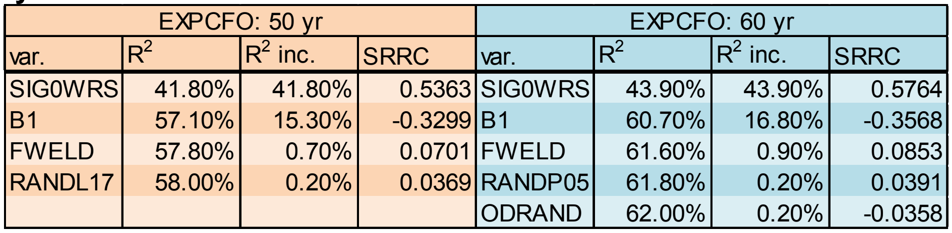
Most of the results at the early time (10 years) give low R^2 values since not many cracks have initiated at this time.

Table 4 Parameter sensitivity analysis results for probability of rupture at 10 years and 30 years



EXPCFO: 10 yr				EXPCFO: 30 yr			
var.	R^2	R^2 inc.	SRRC	var.	R^2	R^2 inc.	SRRC
SIG0WRS	9.40%	9.40%	0.0989	SIG0WRS	30.10%	30.10%	0.3612
B1	10.10%	0.70%	-0.0252	B1	37.80%	7.60%	-0.1826
RANDP07	10.60%	0.50%	-0.0221	ODRAND	38.20%	0.40%	-0.0434
TEMP	11.00%	0.50%	-0.0214	FWELD	38.50%	0.30%	0.0374

Table 5 Parameter sensitivity analysis results for probability of rupture at 50 years and 60 years



EXPCFO: 50 yr				EXPCFO: 60 yr			
var.	R^2	R^2 inc.	SRRC	var.	R^2	R^2 inc.	SRRC
SIG0WRS	41.80%	41.80%	0.5363	SIG0WRS	43.90%	43.90%	0.5764
B1	57.10%	15.30%	-0.3299	B1	60.70%	16.80%	-0.3568
FWELD	57.80%	0.70%	0.0701	FWELD	61.60%	0.90%	0.0853
RANDL17	58.00%	0.20%	0.0369	RANDP05	61.80%	0.20%	0.0391
				ODRAND	62.00%	0.20%	-0.0358

The major contributors to the variance of crack occurrence are weld residual stress at the pipe ID (Sig0wrs) and heat-to-heat material variability in the crack initiation model (B1), which controls the crack occurrence. Both of these variables have epistemic uncertainties. The SRRC values indicate that high values of Sig0wrs and low values of B1 will lead to higher probability of rupture. The other parameters explain only a fraction of the variance (less than 1% each) and are probably spurious. The results at 10 years give a low R^2, which is not surprising considering that only 5% of the realizations have a crack at this time.

Stepwise regression is designed to capture linear (when raw data are used) and monotonic (when rank data are used) influences between selected inputs and outputs. Any other types of influence, notably quadratic, will fail to be captured in the stepwise regression. As an example, the base case sensitivity analysis does not show any influence of the X_c parameter (through thickness location where weld residual stress crosses through zero), which was expected to have an influence. Bubble plots were used in Reference [2] to illustrate the impact of X_c on the uncertainty in the probability of rupture results. Additional statistical tools will be needed to capture the non-monotonic influences between selected inputs and outputs.

8.2.7 Mean value confidence

The effects of the aleatory and epistemic uncertainties are illustrated in Figure 23 for the probability of rupture. The results suggest that the aleatory uncertainty has little impact on the rupture probabilities, but the epistemic uncertainty (especially those associated with crack initiation) are controlling the uncertainty in the probability of rupture. However, these results do not indicate the confidence in the mean values calculated, i.e., is the stability of the results. The confidence in the mean value of rupture probability was estimated using replicate analyses with different seeding. An example of the confidence in the mean probability of rupture is given in Figure 29. In this figure, the red, dashed line represents the upper confidence bound (~97.5th percentile), and the blue, dashed line represents the lower confidence bound (25th percentile) of a 0.95 confidence interval. Clearly, the confidence in the mean value for the base case is good, i.e., the number of realizations (10,000 epistemic and 50 aleatory) provides a well-converged solution.

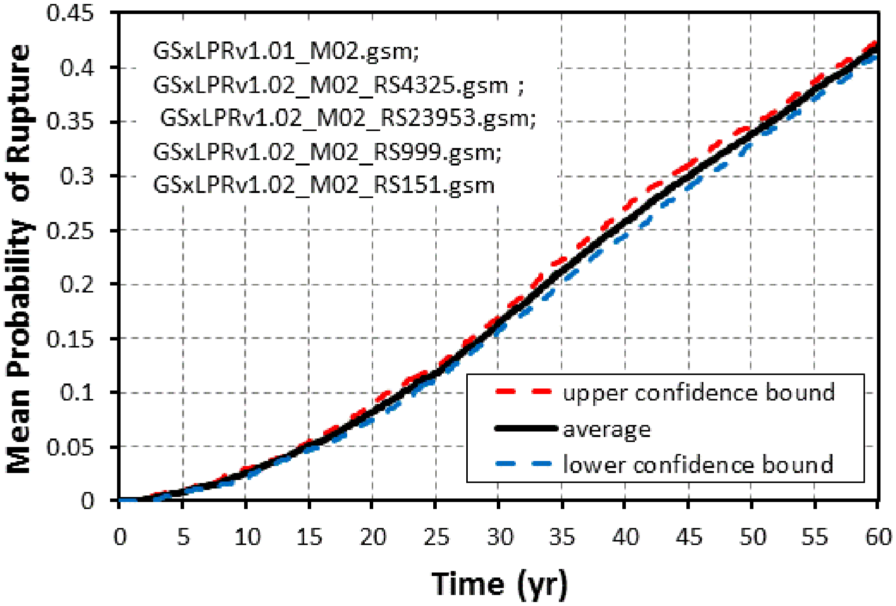


Figure 29 Confidence levels in the mean probability of rupture for the base case

The same type of analyses was performed for the base case with leak detection, as shown in Figure 30. The spread in the confidence is much greater for this analysis when compared to the base case analysis. This difference is due to the number of realizations that caused rupture for the base case and the 10 gpm leakage detection case. For the case with leakage detection, the number of realizations with rupture was much lower, thus creating a larger spread in the confidence. Additional realizations, or the use of a more appropriate sampling technique, would be required to increase the confidence in this mean value.

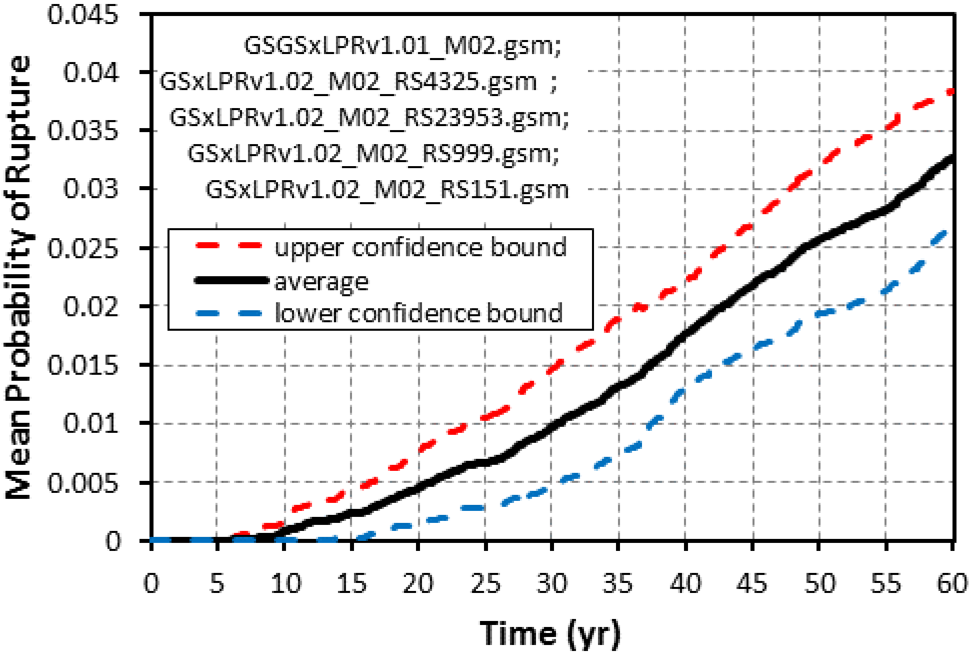


Figure 30 Confidence levels in the mean probability of rupture for the base case with 10 gpm leak detection

8.3 Stress mitigation

The effect of pre-emptive mitigation on the mean rupture probabilities is shown in Figure 31. Note that, as described in Section 5.3, the only mechanical mitigation method that is incorporated into xLPR Version 1.0 is a pre-emptive stress-based mitigation. For this option, the user inputs a mitigated weld residual stress distribution (see Section 7) and a time at which that mitigation is to occur. This modification of the stress profile affects both the crack initiation and growth models. For the example shown in Figure 31, the mitigation was applied at 10, 20 and 40 years. The results from these analyses demonstrate that for the inputs considered, the application of the stress-based mitigation causes the probability of rupture to no longer increase with time. Since the data shown in Figure 31 represents the cumulative probability of rupture, a horizontal line represents no additional ruptures during that time period. However, upon a close investigation of the results from Figure 31, it is clear that before the probabilities cease to increase, they rise slightly above the non-mitigated rupture probabilities. This is due to the tensile zone in the mitigated weld residual stress distribution. For a/t values between 0.5 and 0.9, the weld residual stress becomes tensile. In those realizations where a crack is present that is at least 50% deep at the time the mitigation occurs, the crack growth rate is increased. This effect causes the slight increase in the rupture probabilities before the mitigation effects become apparent.

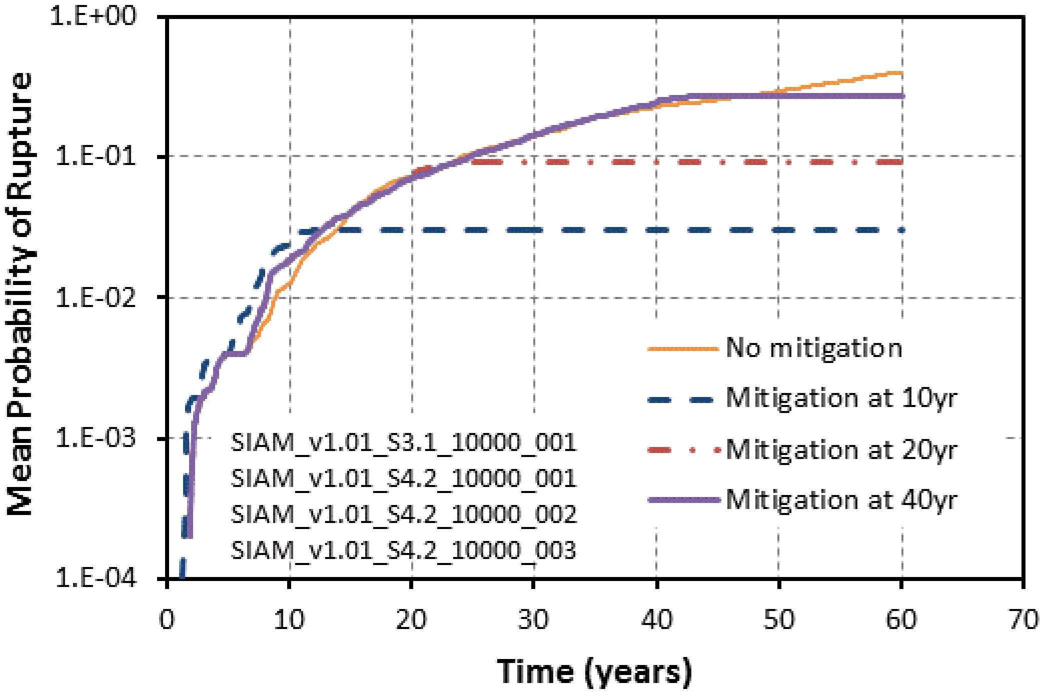


Figure 31 Effect of pre-emptive mitigation on the mean rupture probability

8.4 Chemical mitigation

As described in Section 5.3, the effects of hydrogen on the PWSCC growth rate was implemented in the xLPR Version 1.0 code. The effects of hydrogen and zinc on the initiation of PWSCC was considered, but not implemented. The effects of increasing the hydrogen concentration are shown in Figure 32.

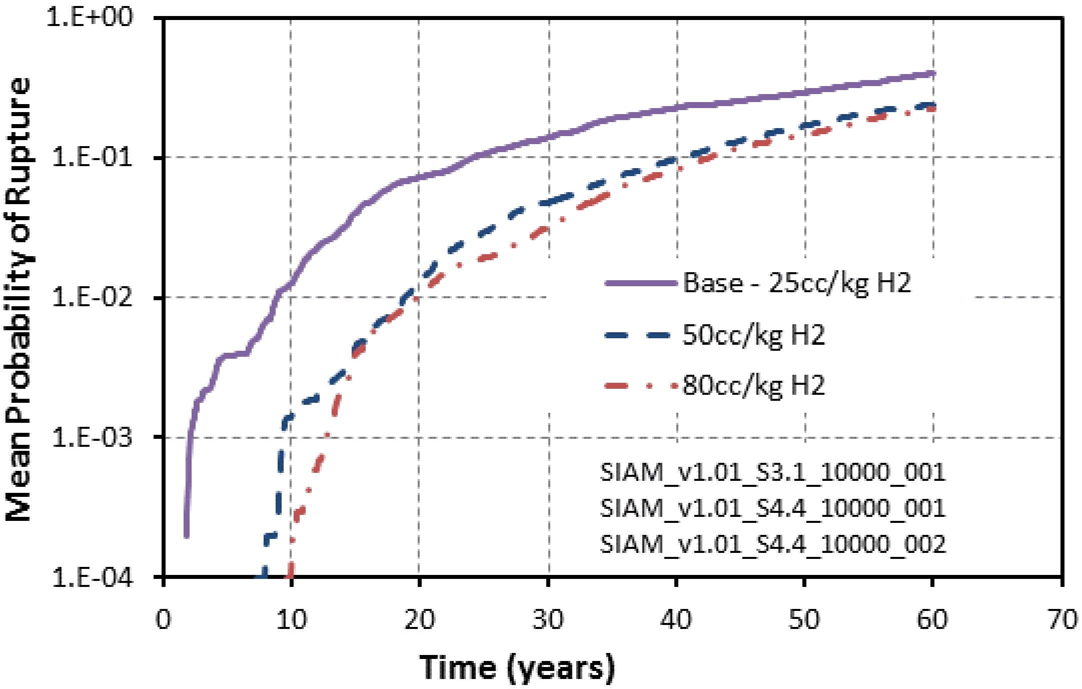


Figure 32 Effect of hydrogen on the mean probability of rupture

The increase in hydrogen caused a decrease in the mean probability of rupture. This change is attributed solely to the change in the crack growth rate due to the increased hydrogen concentration. A large change in rupture probability occurred when the hydrogen content was increased from 25 cc/kg to 50 cc/kg. However, only a marginal increase in rupture probability occurred when the hydrogen concentration was increased from 50 cc/kg to 80 cc/kg. Overall, the decrease in rupture probability is only about a factor of two when the hydrogen concentration increases from 25 cc/kg to 80 cc/kg at 60 years.

8.5 Crack initiation

8.5.1 Direct method I

A sensitivity case was run that considered the crack initiation model uncertainty. The crack initiation module includes three alternative models for crack initiation, Direct Method I, Direct Method II, and a Weibull solution. In both direct methods, the initiation time is an explicit function of stress and temperature, with randomness in the parameters used in the function. Method II was used in the base case analysis. Method I was run for comparison to the base case to evaluate the effects of the initiation model uncertainty on the results. The details for these initiation models can be are documented in Reference [4]. The resulting comparison of the number of cracks initiated is shown in Figure 33.

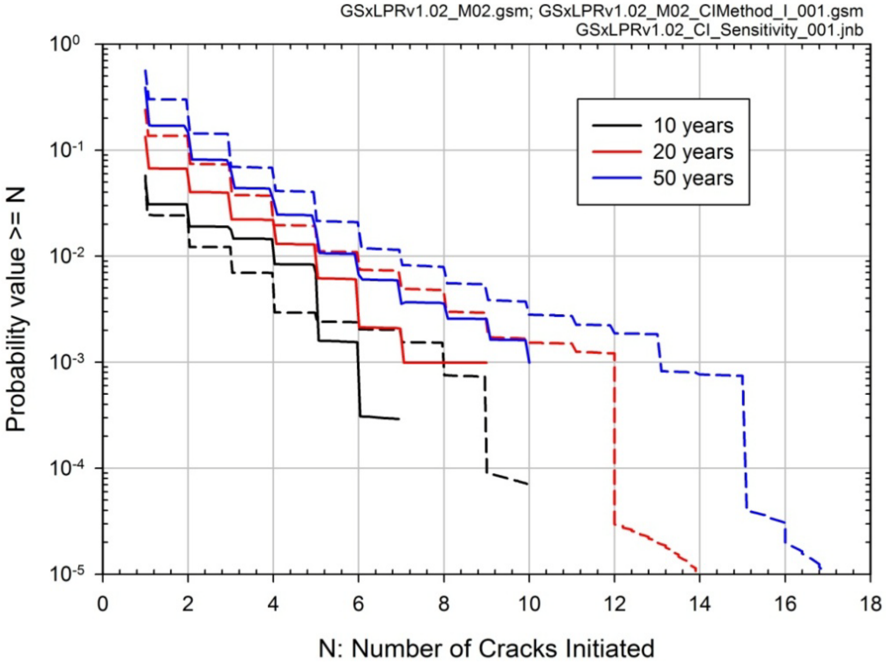


Figure 33 Comparison on number of cracks initiated between Direct Method I (dashed lines) and Direct Method II (solid lines)

The results from Figure 33 indicate that there is not a very large difference in the number of cracks initiated throughout the analyses. In fact, Direct Method I shows a slightly higher number of cracks early in life, while Direct Method II shows a higher number of cracks later in life. This difference in initiated cracks promulgates through to the probability of rupture, as illustrated in Figure 34. The results from this figure show that the mean probability of rupture using Direct Method I and Direct Method II is about 30 percent different at 60 years. The largest difference is seen earlier in life and can be attributed to the difference in the number of cracks initiated.

These similarities in results are expected, since each model was calibrated to the same operating experience.

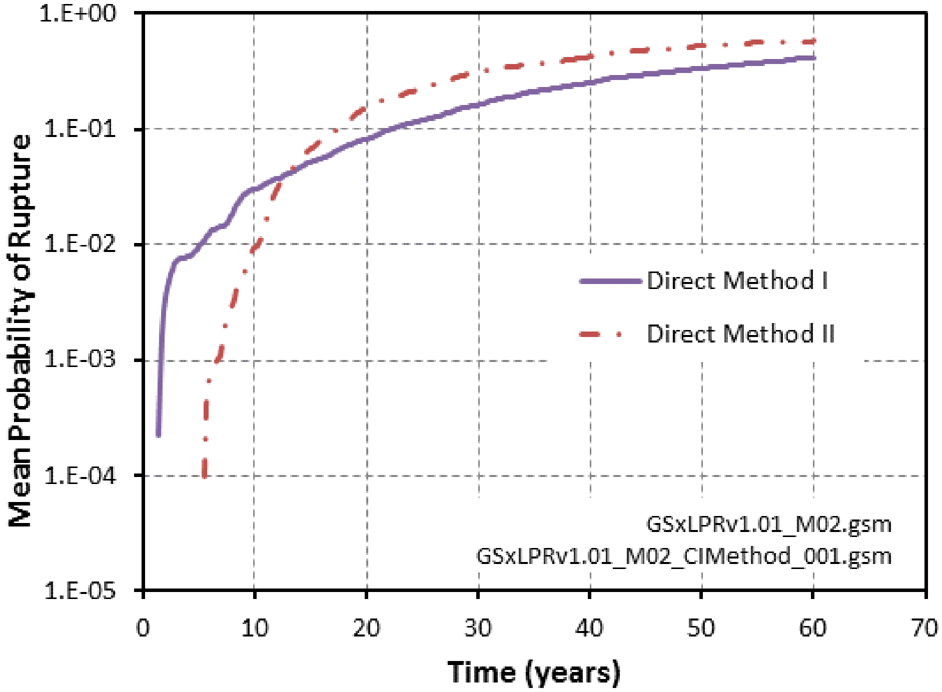


Figure 34 Comparison of mean probability of rupture with Direct Method I and direct Method II crack initiation models

8.5.2 Change of uncertainty classification

The uncertainty analysis that was conducted, as shown in Figure 19 through Figure 23, indicates very little influence of the aleatory uncertainty on the probability of leakage and rupture. Notably, the time to crack initiation is entirely controlled by epistemic uncertainty and not randomness (e.g., aleatory). As a result, the probability of first crack, first leak and even rupture are, for most expected values (e.g., a single epistemic sample comprised of an average over the aleatory samples), either equal to 0 or 1.

For the pilot study problem, the uncertainty classification was performed by subject matter experts in the xLPR Models and Inputs Task Groups [4]. However, the time occurrence of events, such as the initiation of a crack, is often considered as random in similar analyses. With this in mind, a sensitivity case was conducted by changing the classification of the parameters controlling the time when cracks initiate and the location where cracks initiate from epistemic to aleatory. The sensitivity analysis was conducted to give insights on the effects of crack initiation and the separation of aleatory and epistemic uncertainties in the response.

In this sensitivity case, the crack initiation uncertain input parameters B1, BWH_stdev, RandU3, and RandULoc (see Reference [4] and Appendix C for a detailed definition of these parameters) were reclassified as aleatory uncertainty, without changing their distribution type or value. All other parameters in the analyses were kept identical to those used in the base case.

A comparison of the probability of crack initiation with the initiation parameters characterized as epistemic and aleatory is given in Figure 35. For the base case, with crack initiation classified as an epistemic uncertainty, there is no uncertainty in the time to crack initiation for a given epistemic set (no randomness). Therefore, for each epistemic set, the probability of occurrence is either 0 or 1 for a selected time-step. As a result, each probability is displayed as a step function on the left plot in Figure 35. With reclassification of the crack initiation parameter to aleatory uncertainty, the probability of crack initiation varied with time, leading to a smoother probability estimate over the 60-year timeframe. The probability of rupture (Figure 36) shows similar changes with the application of the new uncertainty classification. An interesting consequence from these assessments is that the quantile curves are now completely different. Because the time of crack occurrence was not fixed for each epistemic set, it is more likely to have at least one crack for each epistemic realization (but a smaller chance that ALL realizations within an epistemic set leads to rupture). While the estimate of the mean probability of rupture gives similar results, it is not the case for the quantile values. Their interpretation changes considerably from one assumption to the other. In the second case, for instance, a median of 0.4 at 60 years means that half of the epistemic realizations have a 40% chance of initiating a crack in the future, while in the base case, there was absolutely no chance of rupture for half of the epistemic realizations.

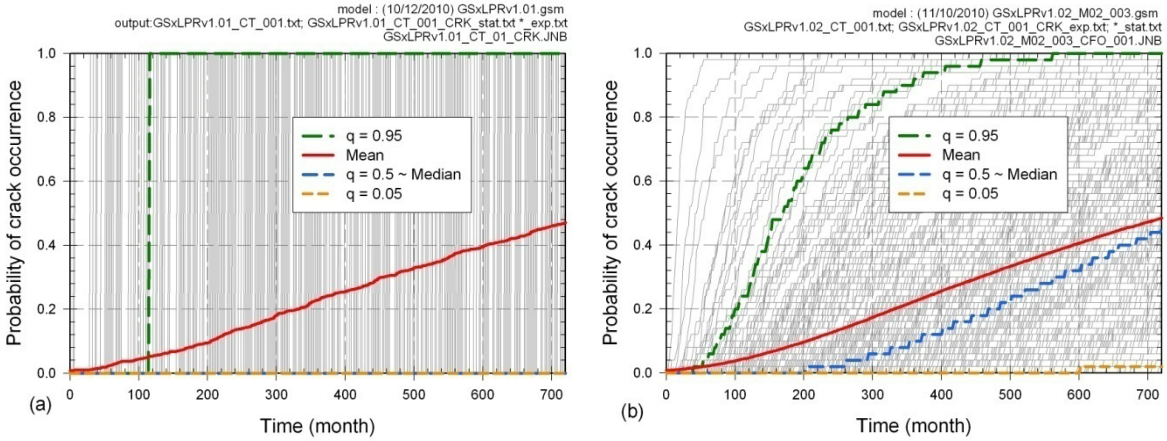


Figure 35 Probability of first crack for base case (a) and with crack initiation uncertainty changed from epistemic to aleatory (b)

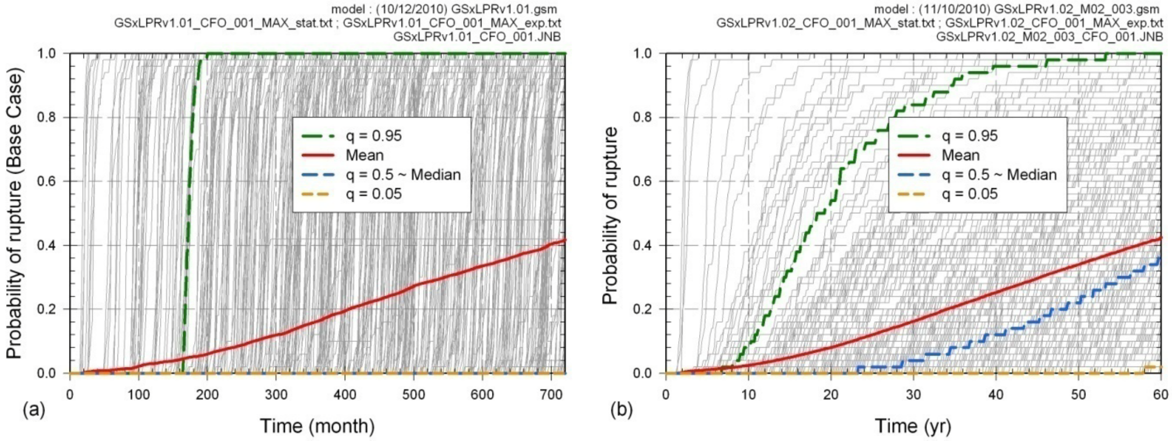


Figure 36 Probability of rupture for base case (a) and with crack initiation uncertainty changed from epistemic to aleatory (b)

A comparison of the CCDF for the probability of rupture between the Base Case and this sensitivity case (Figure 37) clearly shows the significant difference resulting from the re-classification of the crack initiation uncertainty parameters. At 60 years, when initiation is treated as an epistemic uncertainty, there is a 55% chance that no future crack will lead to rupture (P=0) and a 39% chance that all futures will lead to rupture (P=1), which demonstrates that the aleatory uncertainty only affects 6% of the results. When the initiation is treated as an aleatory uncertainty, there is a 3% chance that no future will leak to rupture, and a 6% chance that all futures will leak to rupture. In this case, the aleatory uncertainty affects 91% of the results.

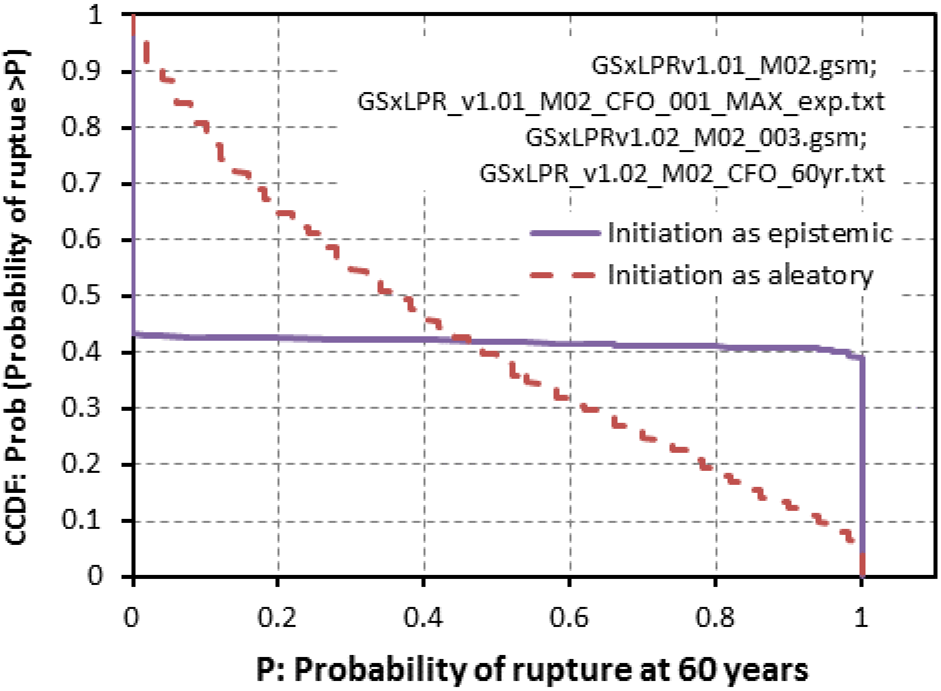


Figure 37 CCDF of probability of rupture at 60 yr for Base Case (initiation as epistemic) and with crack initiation uncertainty changed (initiation as aleatory)

8.6 Safe end weld

The stainless steel safe end weld that attaches the safe end to the surge nozzle piping causes a through-thickness bending stress that can reduce the tensile ID stresses that are present at the DM weld. The extent of the effect on the DM weld is a direct function of the length of the safe end. In the base case for the pilot study, it was assumed that the safe end was long enough such that the safe end weld did not affect the stresses in the DM weld. This case includes the consideration of a short safe end length. For the safe end length considered, it was assumed that the distribution of the ID weld residual stress and X_c was normal. For the ID weld residual stress (Sigma0_WRS), the following values were assumed: mean = -16.2 MPa, standard deviation = 117 MPa, maximum = 300 MPa, and minimum = -300 MPa. For the normalized distance through-wall stress first passes through zero (X_c), the following values were assumed: mean = 0.18, standard deviation = 0.036, maximum = 0.5, and minimum = 0.1. Refer to Section 5.2 for a definition of the weld residual stress distribution inputs. Except for the weld residual stress inputs, all other inputs were identical to the base case.

The mean probability of leakage and rupture for the safe end weld sensitivity case are shown in Figure 38. Note that this case used the same number of total realizations (50,000) as the base case. The results suggest that the rupture probabilities are considerably lower than those in the base case, although the trends between the different levels of leakage and rupture are similar to those in the base case. Also, the comparison between the results from the SIAM and GoldSim framework are reasonable. The differences in rupture probabilities is most likely due to the difference in sample sizes for this problem, i.e., 1,000 epistemic and 50 aleatory for GoldSim and 10,000 epistemic and 1 aleatory for SIAM. These results demonstrate that the weld residual stress has a large impact on the rupture probabilities.

However, the confidence in the solution using these methods is very poor. For the 1,000 epistemic realizations (GoldSim), only 6 cases lead to a crack. Therefore, the estimation of the mean is poor. In addition, when inspection or leak detection is considered, a zero probability of rupture is calculated. This indicates that either a larger (epistemic) sample size, or the use of importance sampling (presented in the next section), is necessary.

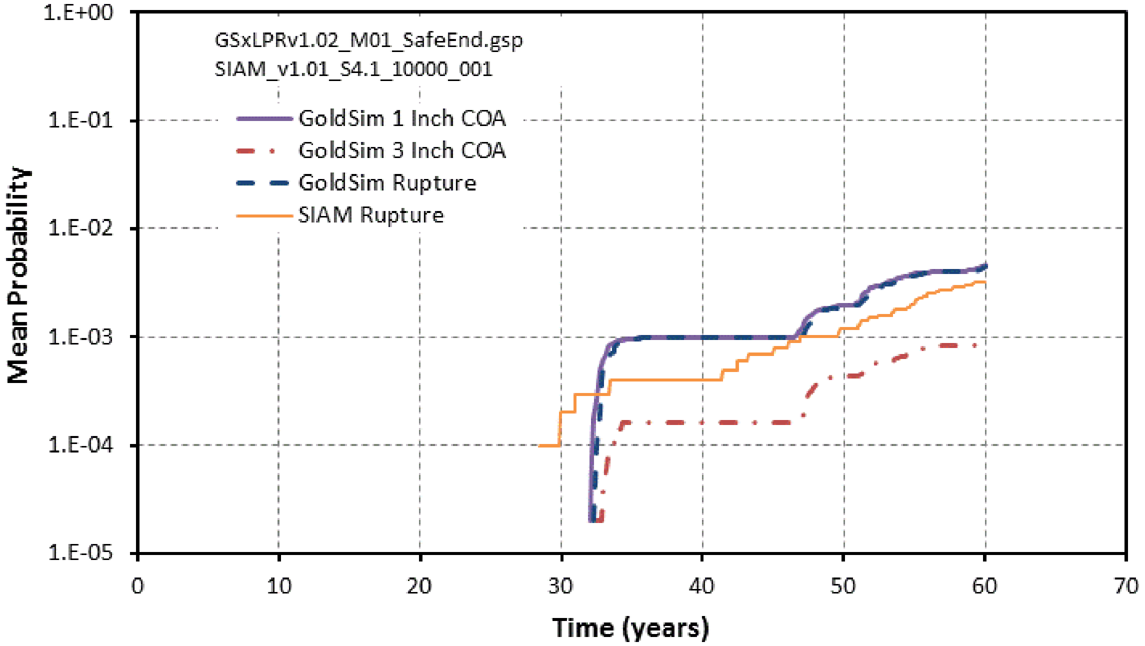


Figure 38 Mean probability of leakage and rupture for safe end weld sensitivity case

8.7 Importance sampling

As discussed earlier in the report, importance sampling consists of intelligently selecting portions of the distribution that lead to a particular response, i.e., rupture. For demonstration purposes, the base case and safe end weld sensitivity case were used to demonstrate the feasibility of importance sampling with both DPD and LHS distributions. As described in Section 4.3.3, the DPD can be defined with unequal probability increments that can be biased toward the tails of the distribution. This feature allows more sampling to be conducted in the tails of the distribution, without modifying the distribution associated with the variables of interest. In addition, importance sampling with LHS can be used by replacing the initial distribution of selected parameters/variables considered as important with a distribution that is designed to cover the region of interest more precisely. In both cases, a corrective term must be applied to represent the importance of the region (or “weight”) covered by each realization.

As an example of modifying the distribution to support importance sampling with LHS, the distribution used in the safe end analyses conducted in Section 8.6 is shown in red in Figure 39. The results from the previous analyses suggest that it is unlikely to have failures below a weld residual stress level of 150 MPa. Therefore, the distribution was sampled for weld residual stress greater than 150 MPa. It was decided to use a log-triangular distribution with minimum = 100, maximum = 300 and mode = 299, as shown in Figure 39 in red. The region covered by the importance sampling distribution is smaller than the one for the original distribution, and this discrepancy has to be taken into account when estimating the final probabilities. In a normal distribution using a mean value of -16.2 MPa and a standard deviation value of 117MPa, the value of 100MPa represents approximately the 84th percentile. Therefore, only 16% of the distribution is sampled with importance sampling, and the results must be corrected for this weight.

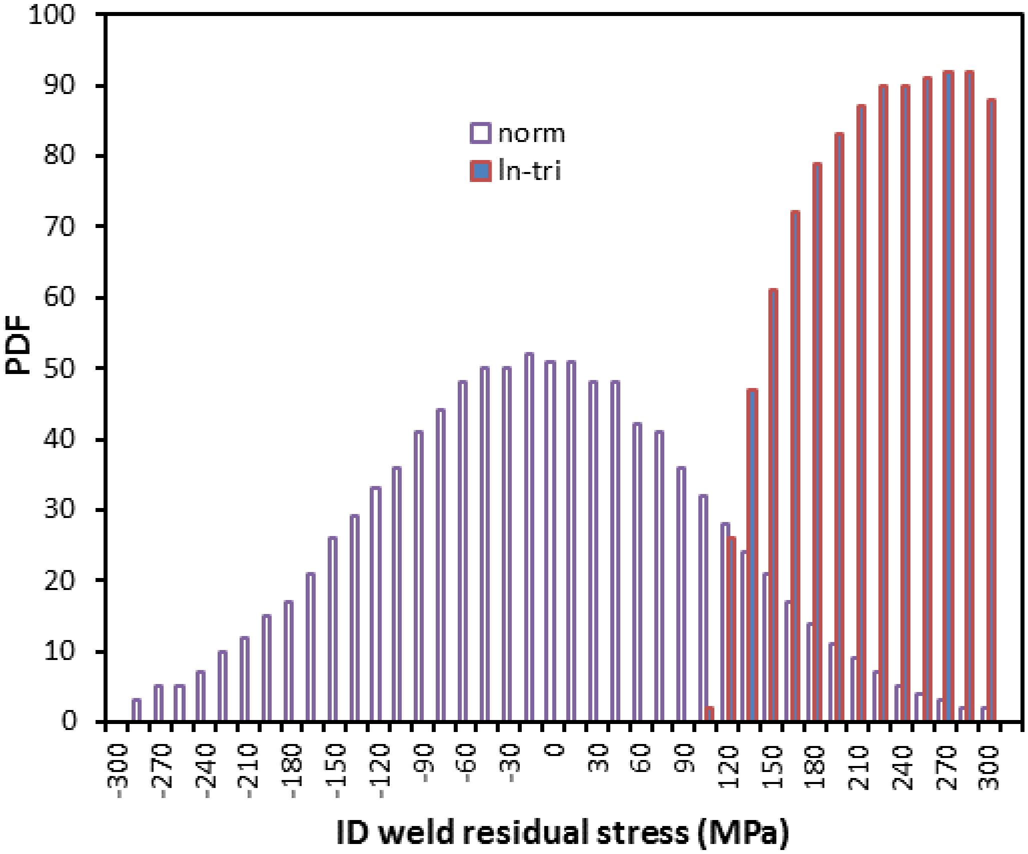


Figure 39 Comparison of PDF for ID weld residual stress using classical (blue) and importance (red) distribution in the safe end case.

As discussed in Section 8.2.6, a parameter sensitivity study was performed on the base case, which indicated that the weld residual stress and the crack initiation parameters were controlling the epistemic uncertainty in the problem. Further investigations discussed in Reference [2] suggest that high values of the weld residual stress and low values of the crack initiation parameters control the ruptures. Therefore, it was decided to use these variables in the importance sampling study. The variables that were importance sampled included, the ID weld residual stress, X_c (distance where WRS crosses zero), and B1 (heat-to-heat material variability in the crack initiation model).

8.7.1 Base case

Figure 40 shows a comparison of the original sampling technique and importance sampling for the base case probability of rupture with and without inspection and/or leak rate detection. In this figure, the solid lines represent the results with the original sampling (1,000 epistemic and 50 aleatory) using the original distributions. The dashed lines represent the importance-sampled cases using the weighted distributions with LHS and 10,000 realizations. For the case with no ISI or leak detection, and the case with a 2 year ISI schedule, the difference between the importance samples and original sampling is minimal. This suggests that the original sampling was sufficient for this output. However, a larger difference is illustrated for the case with both a 2-year ISI schedule and a 1 gpm leak detection limit, especially for times less than 20 years. This difference illustrates the low confidence in the mean values when low probabilities are calculated, and suggests the need for importance sampling or an increased number of realizations.

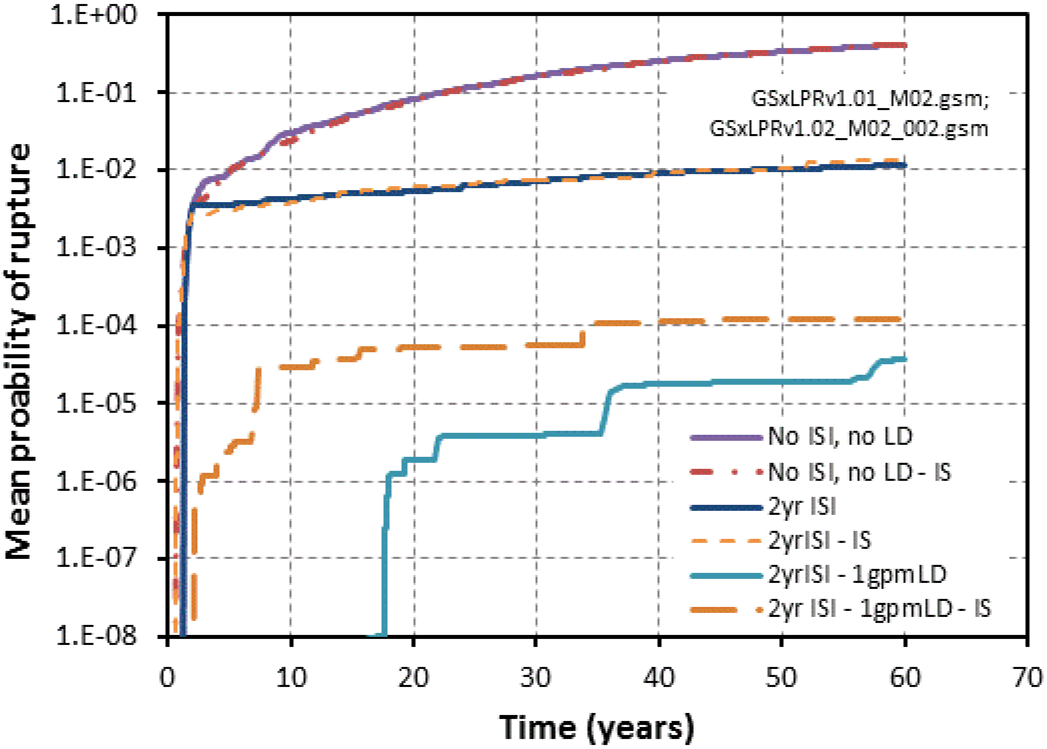


Figure 40 Mean probability of rupture for the base case with and without importance sampling

8.7.2 Safe end weld

The sensitivity study case using a weld residual stress that contains the effects of the safe end weld was also used to demonstrate the impact of importance sampling. As with the previous example, the uncertain variables associated with weld residual stress (Sigma0_WRS and X_c) and crack initiation (B1) were importance sampled.

Figure 41 shows the mean probability of rupture with and without importance sampling. In this figure, the solid purple line represents the initial analysis without importance sampling, inspection and leak detection. In this case (1,000 epistemic and 50 aleatory realizations), there were no ruptures calculated for times less than 30 years. The dashed lines in this figure represent the case without inspection or leak detection, but with importance sampling. The advantage of importance sampling (whether using DPD or LHS approach) is seen to produce a higher confidence in the probability of rupture, especially at times less than 30 years. In fact, the differences between importance sampling with DPD or LHS are minimal for the case without inspection or leak detection. In addition, the probability of rupture at 60 years appears to be well-converged for the case without inspection and leak detection.

Importance sampling was mandatory for the safe end weld case when leak detection and inspection were taken into account, since the original sampling produced no ruptures over 60-years. However, the difference between the solid blue and solid yellow lines indicates that there are some differences in the importance sampling routines. The confidence in each of the results for the case with inspection and leak detection is shown in Figure 42. The confidence for each of these means is calculated using the bootstrap method [2], which consists of sampling with replacement over the response generated by the original analyses. The central limit theorem states that when the mean and variance of the initial distribution are finite, the mean distribution should be asymptotically normal. Therefore, if the distribution of the mean values is normal, the sample size should be large enough to represent a stable solution. For the cases shown in Figure 42, both results represent normal distributions relatively well. One important note is that, despite the variation, the mean probability results are within one order of magnitude from each other at 60 years, which is reasonable considering the magnitude of the probability estimated (close to 10^{-6}) and the original sample size used (10^{4}). Better confidence can be obtained by increasing the number of realizations that are importance sampled, or by selecting the region of importance more appropriately.

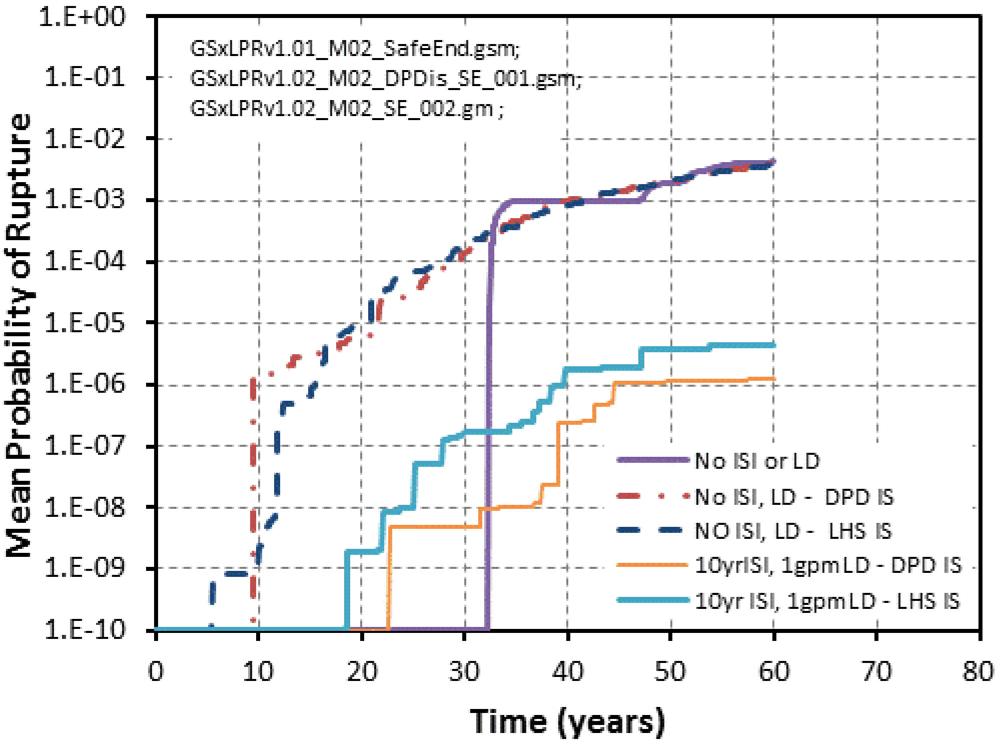


Figure 41 Mean probability of rupture for the safe end weld sensitivity case with and without importance sampling

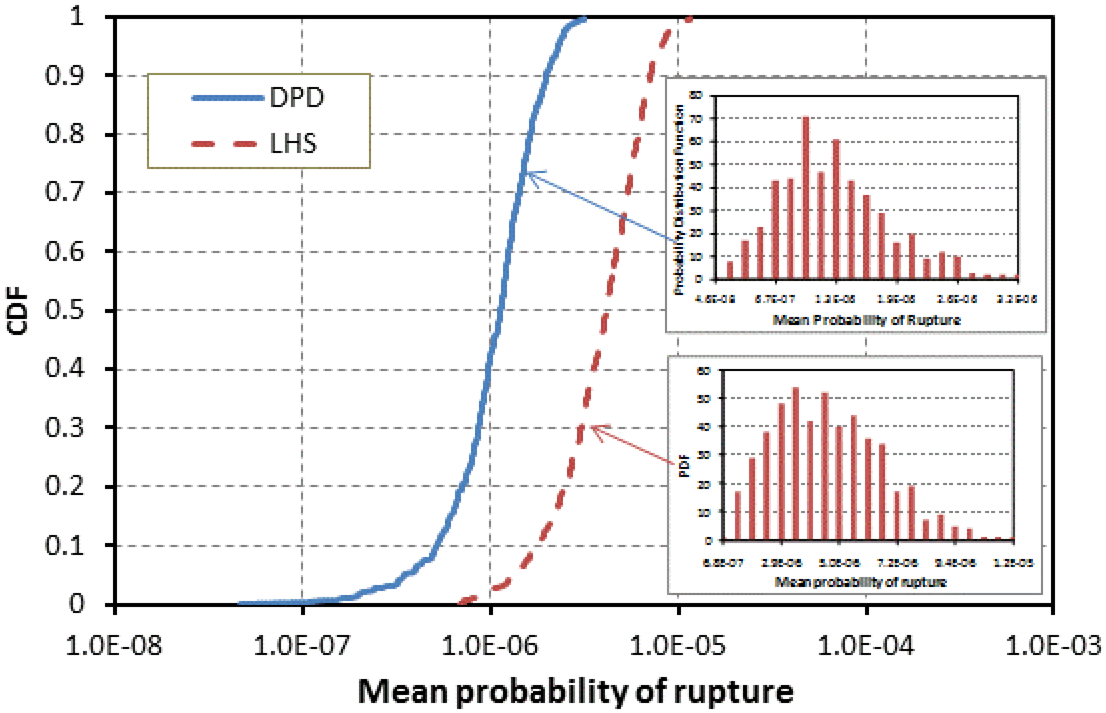


Figure 42 Distribution of mean probability of rupture (at 60 years) for safe end weld sensitivity case using bootstrapping technique

As a final demonstration, mitigation was added to the safe end sensitivity case previously discussed. For this example, a mitigation time of 20 years was chosen. As before, 10,000 realizations were used with importance sampling on both the weld residual stress and the crack initiation parameters. The results of the analyses are shown in Figure 43. As shown in this figure, the mitigation at 20 years reduced the rupture probability at 60 years by two orders of magnitude, while the leak detection and ISI reduced the rupture probabilities by about three orders of magnitude. The combined effect caused a reduction of almost six orders of magnitude on the rupture probability at 60 years. Since only a few of the 10,000 realizations produced rupture with mitigation, inspection, and leak detection, the confidence in the mean value is low, as shown in Figure 44. As with the previous example, the confidence is estimated using the bootstrap method, which would produce a normal distribution of the mean if the analysis was stable. However, the results in Figure 44 suggest an exponential distribution and indicate a lack of stability in the analyses. Additional realizations would be needed to obtain a better estimate of the mean value for the probability of rupture. However, the distribution reported in Figure 44 spans approximately one order of magnitude, which is considered good based on the sample size (10^5) compared to the calculated probability of rupture ([10^{-9}-10^{-8}]). Based on the previous results, increasing the number of realizations leading to rupture by a factor of 2 or 3 may be enough to significantly increase the accuracy. Therefore, it may not be necessary to increase the sample size by an order of magnitude.

Moreover, while it requires more manipulation from the user, a careful selection of the input parameters may allow increasing accuracy without increasing sample size. As Figure 43 suggests, the cracks that may occur after mitigation will not lead to rupture and will not change the probability of rupture significantly. Therefore, forcing the sampling to generate cracks at an early time (and correcting the results accordingly) may be a way to increase the accuracy without changing the sample size.

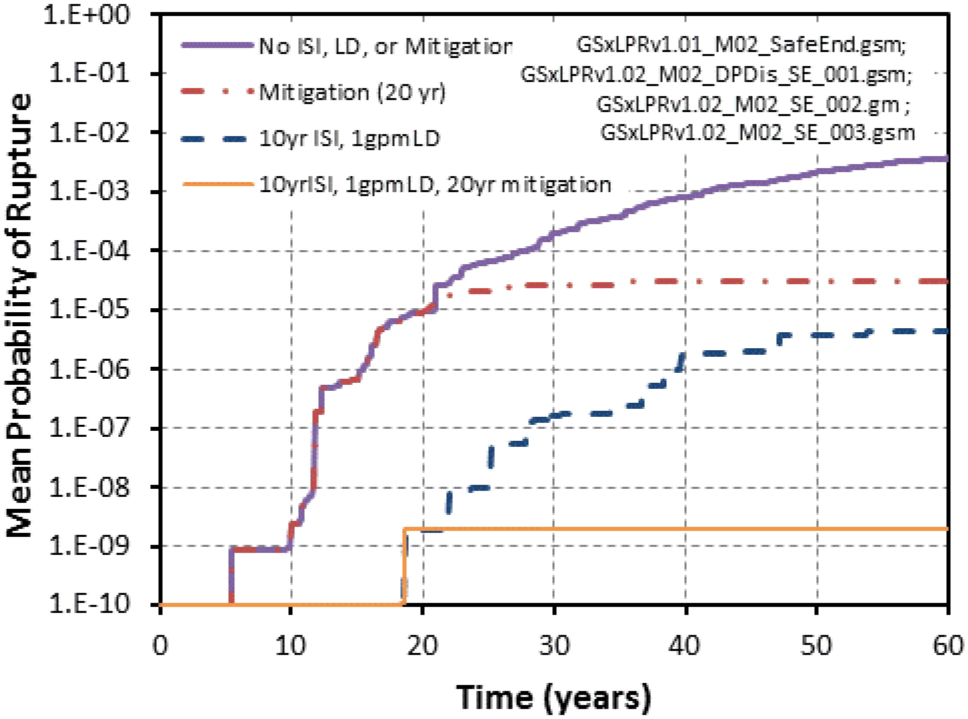


Figure 43 Mean probability of rupture for safe end sensitivity case with mitigation, leak detection and inspection

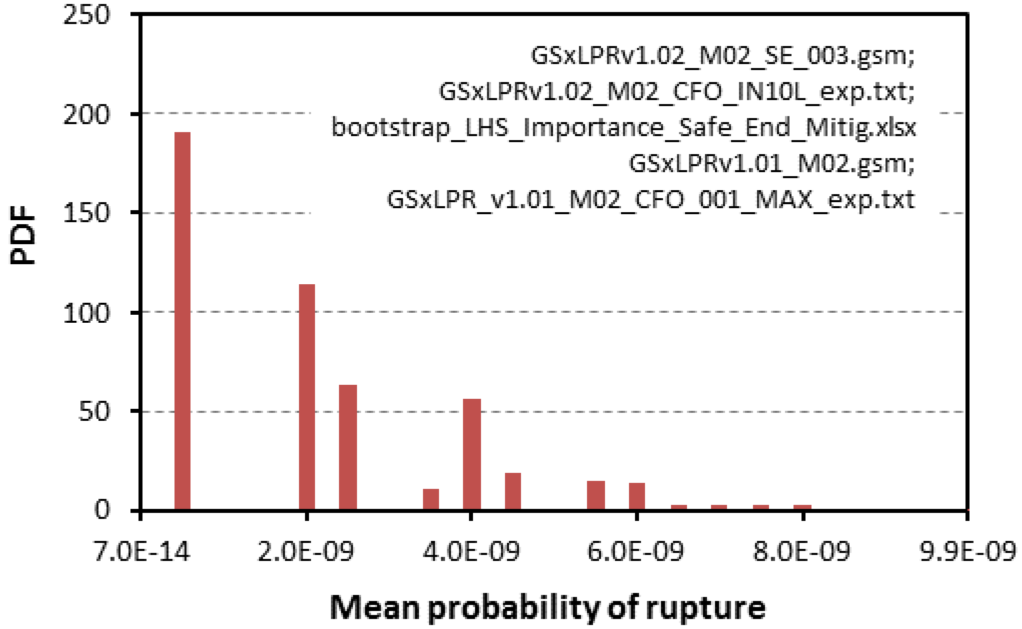


Figure 44 Confidence in the mean probability of rupture at 60 years for safe end sensitivity case with mitigation, inspection and leak detection

9 Lessons Learned and Version 2.0 Recommendations

The xLPR task groups worked very closely over an approximately 1.5-year period to develop two separate framework codes. This effort included not only developing the framework structure and flow, but also development of the configuration management program, documentation and verification of the framework, and the modules development, as well as understanding and implementing the physics-based modules necessary for the calculation of probability of rupture. Over that time period, the team developed an appreciation for the complexity of this problem, and the project structure needed for successful completion of the pilot study. Through the process, many important lessons were learned. This section of the report documents the lessons learned throughout the pilot study program. These lessons include not only technical lessons from the module and framework development and implementation, but also from the organization and program management viewpoint. Using these lessons learned, suggestions for xLPR Version 2.0 are presented.

9.1 xLPR organizational structure

As stated in Section 1 of this report, the xLPR program was organized using a team structure that consisted of both NRC and EPRI staff and contractors. The task groups (computational, inputs, models, and acceptance) were coordinated by a Project Integration Board (PIB), which consisted of two members from each task group, one management member from both the NRC and EPRI, an NRC-NRR representative, and an industry representative. The purpose of the PIB was to coordinate the effort and make programmatic decisions on an as-required basis.

9.1.1 Interactions with task groups and PIB

A key to organizational success is communication. Within the xLPR program, the PIB's responsibility was to guide and direct the overall program, and make key decisions on the developmental process. In order for this to occur, regular communication between the task groups and the PIB was necessary. In the initial stages of the pilot study, the task groups made a strong effort to communicate with the PIB and obtain advice and approval on paths forward. For instance, near the start of the program, there were strong opinions that the xLPR framework code needed to be completely open source to allow free distribution of the code. Others, through experience in past NRC regulatory efforts, saw the value in using commercial software for the framework development. The computational group proposed to the PIB a parallel path where two frameworks were developed and compared at the conclusion of the pilot study. In another case, it became clear early in the program that the models development and the framework development needed to occur in parallel if the pilot study was to be completed in a reasonable timeframe. Therefore, the computational group suggested to the PIB, through the development of an alpha program plan, to create the initial framework using "temporary" modules developed by the computational group. This process allowed the framework to be developed with a structure such that the final pilot study modules easily replaced the temporary alpha modules.

As the project progressed, communication between the task groups and the PIB diminished. More and more decisions were made between task group leads without PIB input. Much of this lack of communication stemmed from schedule pressures to complete the Version 1.0 code. Early PIB discussions led to the conclusion that it was necessary for the pilot study to be completed in approximately 2 years. As that time period expired, quick decisions were made at the expense of contacting the PIB. As further work progresses past the pilot study, it is imperative that the PIB and the decision-making process be revisited. From the task group viewpoint, and considering schedule deadlines, the process for obtaining PIB review and approval is vague and cumbersome. Each of the discussions presented above occurred during face-to-face PIB meetings. It is impractical to assume that meetings are required whenever a decision is needed. Although e-mail is not necessarily the best forum for discussing decision-based issues, video or teleconferences may be appropriate for most cases. A formal process or procedure for PIB review and approval that gets incorporated into the change control process is required to ensure that timely discussions occur to allow decisions to be made efficiently.

9.1.2 Interactions within and between task groups

Interactions and communication within and between the task groups is also of vital importance. The organizational structure of this program is such that the task groups need to rely on each other for pertinent information. For instance, in order for the framework to be modular in nature, it is imperative that the inputs and outputs for each module are clear and well understood by the Computational Group. Without communication between the Models and Computational Groups, the modular-based coding would be burdensome. In fact, early in the process, the communication between the Models and Computational Groups was inefficient. Oftentimes, the Computational Group would meet and assign actions items to the Models Group without their input or knowledge. Similarly, the Models Group would have discussions of uncertainties and uncertainty propagation without Computational Group input. This issue led to frustration within the different task groups. As a solution, it was decided that a representative from each task group would be present at all other task group meetings. This allowed each of the task group’s concerns to be heard at each of the individual task group meetings. Once this began occurring on a regular basis, the flow of information and completion of action items occurred much more efficiently.

The same level of communication and coordination is required within each task group. For the Computational Group, regular face-to-face meetings with the entire group membership were necessary to ensure progress was continually being made on the software coding. Without those meetings, the framework development would have floundered. This process worked well for the Computational Group, since each member was focused on the same final goal, which was completing the framework. The communication effort was more complicated for the Models Group. Since each model represented a somewhat unique field of expertise, the members split into modeling subgroups and thus the entire group was rarely focused on a single common goal. Each module subgroup met on an as-needed basis as the individual models were being developed, but the individual team member did not have any direct interaction with how their model was incorporated into the framework. In some cases, this led to confusion on how the model should be implemented within the framework. Having a dedicated team member joining each of the individual task group meetings and reporting back to the Models Group members became the best way to foster efficient communication between and within the task groups.

9.1.3 Task group staff

Staffing the task groups is highly dependent on available resources and funds. Even with the utilization of both NRC and EPRI staff and contractors, having the appropriate people involved within the task groups was difficult. Within the Computational Group, the project was fortunate to have experts in solid mechanics, probabilistic fracture mechanics, weld residual stress, computer programming, configuration management and uncertainty treatment and propagation on the team. In fact, three members had direct experience in coding other pipe probabilistic fracture mechanics software codes. In this case, the staffing choices were appropriate and well-suited for the development of the framework.

However, for the Models Group, there were some module subgroups that lacked the appropriate staff. For instance, the Mitigation Subgroup was not staffed at all, and the Computational Group developed the mitigation model for the Version 1.0 code. The Crack Initiation Subgroup, while staffed with highly capable people, did not have a materials expert (for initiation mechanisms for PWSCC) on the team. In some of the models subgroups, the staffing was minimal, i.e.., one or two people deciding and implementing the models. The limited staff raises the question of whether the model chosen was appropriately vetted for use in this code. In other subgroups, the staffing was appropriate and the development of the models in those subgroups proceeded smoothly. The issues of staffing in the Models Group stemmed from the availability of funds and resources, as well as the impact of other work priorities on group members' time and not the ability to find the appropriate staff. Priorities and other commitments limit the time available for many model experts.

As learned through the "growing pains" of the pilot study project, three very important aspects of the organizational structure are required for program success:

- Dedicated team members, whose qualifications cover the important aspects of the group responsibility. It is advantageous if team members understand the basic computational process to aid in the incorporation of elements into the overall model.
- Enthusiastic team leadership.
- Efficient communication process within and among the teams.

9.2 Framework development

9.2.1 Quality assurance and configuration management

At the start of the pilot study, it was decided that the Version 1.0 code needed to be developed in a controlled manner. It was envisioned that the final xLPR code would be developed to meet quality standards, e.g., ASME-NQA-1-2008. However, it was unreasonable to tackle this task for the pilot study. Therefore, the project team focused on the development of a configuration management process that would satisfy quality assurance (QA) requirements (see Section 3). QA is much more than configuration management, and consists of a systematic and documented practice of monitoring the software and model development processes and methods used to ensure quality. Software QA (SQA) encompasses the entire software development process, which includes processes such as requirements definition, software design, coding, source code control, code reviews, change management, configuration management (CM), testing, release management, and product integration. SQA is organized into goals, commitments, abilities, activities, measurements, and verification. SQA typically follows an industry consistent process (e.g., ISO-9000 or ASME-NQA-1-2008), regardless of the application. Model development follows a similar process that incorporates the fundamental aspects of QA, including version control, reviews, change management, testing, CM, and release management. CM is the process that focuses on demonstration, documentation, and control of the steps taken, and the products developed, under a QA program. A robust CM system includes both electronic and programmatic controls that are linked to a QA program that is well-defined. The link between the CM and QA Program usually takes the form of guidelines or a CM plan, which provides the roadmap between the required QA steps and methods and the CM system that maintains the configuration control. The CM process described in Section 3 was not linked to a QA process. A finalized plan for the xLPR QA Program for software development, model development, and input development is needed (e.g., NUREG/BR-0167).

Fundamentally, all xLPR participants must participate in the QA process for Version 2.0. Establishing the QA plan and controls for xLPR is the first step in the development process. Very costly re-work and schedule delays will result if the development process does not begin under defined QA processes. The xLPR program goals for QA, software, inputs, models, etc. must be clearly articulated in a QA Plan for xLPR. Once this has been established, a CM process can be defined. Each organization needs to generate a work plan and QA/CM plan to map the xLPR project goals to the products. As part of Version 2.0 planning, the xLPR program needs to consider the benefits and costs associated with supporting a single, project-wide program, or integrating several systems located between the various xLPR organizations. Regardless, each organization needs to develop a plan that identifies the process used to comply with the program requirements. The xLPR program needs to have a transparent and traceable CM system that will cover the xLPR code lifecycle.

9.2.2 Code modularity

The concept behind the modular xLPR code was to allow the ease of modification and addition of models without major modifications to the overall framework. However, as the pilot study project progressed, it became clear that the framework development could not be done independently from the rest of the analysis, but instead must be considered as an integrated part of the whole project. In many occasions during the pilot study project, the framework logic needed revision, or a module needed to be modified, in order to correctly capture the phenomenon considered or to appropriately represent the response in a downstream model. For a complex problem where many factors influence the desired results, the framework development cannot be considered as a simple "plugging" of modules within a probabilistic loop. The framework has to be developed and validated so that the flow between modules is appropriate. Adding new modules/models where the inputs and outputs vary greatly from the implemented modules will always require some modification to the framework, and an understanding by the framework developers on how this new module affects the downstream modules and the overall probabilistic flow is necessary. As this project progresses beyond the pilot study, it will be necessary for the development (computational) team to be involved at a high level so that all members can understand the overall purpose of the project. It is also recommended that the development team have sufficient time, and use an iterative approach, to facilitate their understanding of each part of the model, as well as the global model, at the physical, mathematical and computational levels. This approach will assure that the problem physics is captured correctly, and the process is performing as expected.

9.2.3 Module and framework coding

The idea of code modularity allows the independent development of sections of code that can be easily compiled into the overall framework. In order to make this transition smooth, the code must be written in an orderly and well-defined manner that removes coding structure that may hinder the performance or cause compilation errors. In xLPR Version 1.0, the modules were coded without guidance or support. In some cases, legacy code was used which was not well documented or written. These ad-hoc coding behaviors lead to developmental delays due to difficulties in code verification, compilation issues, and code run-time issues, even though each module followed the pre-defined CM process. Therefore, it is recommended that well-written unambiguous software requirements be developed and followed for future xLPR versions.

Each of the modules within xLPR Version 1.0 was written in FORTRAN. However, this was not necessary, in that the frameworks were developed such that any engineering-based computer language could be used. Wrappers for both SIAM and GoldSim were developed so that the modules would communicate correctly with the framework structures. This allowed for greater flexibility in module coding.

However, the framework coding was not as flexible. The commercial software GoldSim requires the developer to purchase a GoldSim Pro license in order to develop or write code within that framework. The development software is object-based, and requires some programmer training before the user is capable of developing software. However, GoldSim is well-documented with an extensive User Guide, an online knowledge base, and technical support.

The open source software, SIAM, was written by ORNL in Python. Python is an open-source, general-purpose, object-oriented, scripting language that has found extensive application worldwide. Its cross-platform capabilities allow its use on such diverse operating systems as Microsoft Windows, Macintosh OS-X, Unix, and Linux. Python is used extensively in the computer programming arena, but many engineers, developing scientific-based software applications such as xLPR, do not have Python experience for code development.

In addition, commercial software has built-in functions and optimizations that allow the developers to save time and money from a design, development, and QA standpoint. Open source software can be programmed to have these capabilities, and, in fact, can better match the program needs than the commercial software. However, the time and cost needed to bring the open source code to the level of functionality in the commercial codes needs to be taken into consideration.

Whichever framework structure is chosen for future versions of xLPR, staff with flexible coding abilities will be required. This is true not only for original code development, but also for the verification process, i.e., those verifying the software must be able to understand the coding structure. Since one of the goals of this project is to develop a code that does not rely on a single person, or single talent, for further development, care must be taken to choose the correct staff for the framework structure chosen. Investigating the framework development learning curve may aid in understanding the impact of the choice of framework software.

9.3 Code technical limitations

Throughout the development of the xLPR Version 1.0 code, certain assumptions and limitations had to be made due to time constraints and model availability. In some cases, it was unknown the effect of these assumptions or limitations would have on the rupture probability. In this section of the report, these limitations are discussed. When available, the effects of the limitations are discussed and recommendations for Version 2.0 are made.

9.3.1 Models/Inputs

Several model assumptions were made early in the developmental process for the xLPR Version 1.0 code. These assumptions and limitations were made because the main purpose of the pilot study was to demonstrate the feasibility of a modular-based probabilistic fracture mechanics computer code developed under a strict configuration management process. It was not the intent of this code to be able to accurately capture the probability results, but rather to make estimates to demonstrate feasibility. However, as progress is made toward Version 2.0, these limitations should be considered and eliminated, if possible.

1. No manufacturing defect distributions

In the pilot study code, only PWSCC-initiated defects were considered. This assumption could lead to non-conservative predictions of the rupture probability. If flaws are present at the start of the analyses, the rupture may occur earlier. However, the added number of flaws may make the credit for inspection much larger. The difficultly comes in properly documenting the flaw distribution, as well as the flaw size distribution, for DM welds. Both embedded and surface flaws should be considered since fatigue crack growth may cause embedded defects to become surface breaking defects, thereby exposing the crack faces to the water environment, which enhances the crack growth rate.

2. No fatigue initiation or growth

For simplicity, it was assumed in xLPR Version 1.0 that the effect of fatigue initiation or growth was not significant with regard to the rupture probabilities. For many situations, this assumption has been demonstrated to be valid, especially for the case of a pressurizer surge nozzle where the operating temperatures are high, thus yielding very fast PWSCC growth rates. However, as the Version 2.0 development gets underway, and other piping systems and fabrication flaws are considered, fatigue crack growth due to normal heat up, cool down, and other normal/upset transients may become significant. Therefore, appropriate fatigue initiation and crack growth models, including reactor water environmental effects, should be considered for Version 2.0

3. Simplistic load model

Since the pilot study was focused on a single pressurizer surge nozzle location, and it was agreed upon a priori that data from Reference [5] would be used, the load model generated for the pilot study was simplistic. It was assumed that all operating loads, except pressure and temperature, were constant. It was also assumed that all loads were static, i.e., no transients were defined. For Version 2.0, it may be necessary to expand the load module to include transient stress calculations, through-wall stress distribution calculations, and allow for uncertainty in the loads. In addition, the proper handling of earthquake loading must be considered for both multiple occurrences of small (operating basis) events, and a single occurrence of a life limiting (safe shutdown) event, each with multiple loading cycles. Other loadings, such as thermal stratification, should also be considered.

4. Only circumferential surface-breaking cracks considered

In the pilot study, only circumferential surface breaking cracks were considered. This decision was made because circumferential cracks will control the rupture behavior (axial cracks are limited in length to the susceptible material, i.e., Alloy82/182 weld), and the stress intensity solutions were readily available. However, the presence of axial cracks could have a large impact on the rupture probabilities. Since the total hoop stress for axial cracks is higher than the axial stress applicable to circumferential cracks, it is expected that axial cracks will leak before circumferential cracks. This will cause the probability of small leaks to be higher, but will significantly reduce the rupture probability in cases where leak detection is considered, because removal of leaking axial cracks will also remove subcritical circumferential cracks. Therefore, for Version 2.0, axial cracks with limited crack growth that are located only in the weld should be considered.

5. Only idealized crack behavior

In Version 1.0, surface cracks are assumed to be semi-elliptical in shape, while through-wall cracks are assumed to have radial crack fronts. These idealized assumptions were made due to the availability of the stress intensity solutions and stability estimation schemes. An additional assumption was that, when a surface crack transitions to a through-wall crack, the resulting crack is idealized in shape and has a crack area equal to that of the surface crack at the time of leakage or failure. Recent studies [5] have illustrated that PWSCC may not grow in such an idealized manner, and the assumed transition may not be appropriate as modeled in the Version 1.0 code. Studies [30] have suggested that idealized surface crack solutions can be used to predict the time to leakage and the flaw size at leakage for PWSCC growth, but the leakage will be overestimated using these solutions, as shown in Figure 27. This figure suggests that the first leakage is extremely large, and low leakage detections would be ineffective. For Version 2.0, non-idealized crack behavior and realistic crack transition behavior should be considered for inclusion. Also, the Version 1.0 stress intensity factor solutions (Section 5.5) that are used for calculating crack growth assume that all the stresses are primary and the loading will not be reduced by the compliance change and stress relaxation due to crack growth, which is conservative. If possible, the Version 2.0 stress intensity factor solutions should consider the effects of relaxation of secondary stresses, i.e. displacement-controlled thermal stresses, in calculating the crack growth.

6. Simplistic WRS model - 3rd order approximation and constant around circumference

For determining PWSCC behavior, the weld residual stress is of vital importance. Sensitivity studies from this effort demonstrated that the ID weld residual stress value is a significant contributor to the uncertainty in the rupture probability results. The model chosen for Version 1.0 represents the weld residual stress as a third order polynomial. For many cases, this model may be sufficient. However, as shown in Figure 45, the model can give poor predictions for some cases of weld residual stress. In this figure, which is for a pressurizer surge nozzle DM weld with the local stress modification effects of the stainless steel safe end weld included, the solid lines represent 100 simulations using the Version 1.0 model. The symbols represent the finite element results from different analyses that were used in generating the distributions for the model. Clearly, the model does not predict the stresses mid-wall or at the outer diameter reliably. These differences can have a significant effect on the leakage and rupture probabilities.

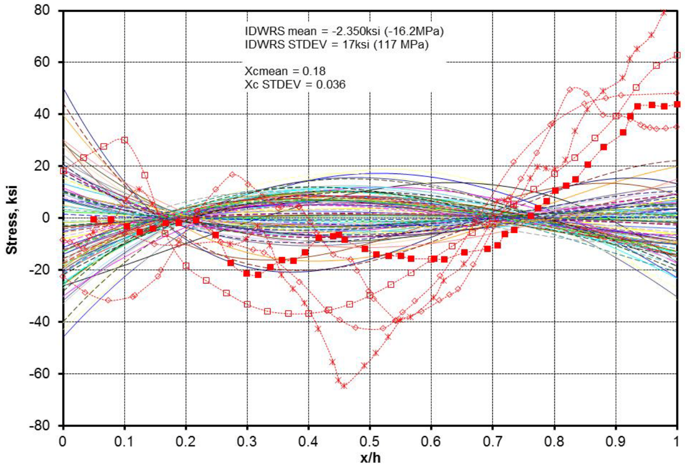


Figure 45 Weld residual stress model for safe end weld case

In addition, in Version 1.0, it is assumed that the weld residual stress is constant around the circumference of the pipe. Local weld repairs, which produce non-uniform stresses around the circumference, may lead to earlier leakage with shorter flaws at leakage. This effect may decrease the failure probabilities. Therefore, for Version 2.0, a more sophisticated weld residual stress model should be considered. In Version 1.0, it was conservatively assumed that each flaw contributed equally to the probability of both leakage and rupture instead of being controlled solely by the largest flaw. Better modeling of the weld residual stress distribution through the wall and around the circumference in Version 2.0 will reduce the effects of this conservatism.

7. Simplistic surface crack stability model

In Version 1.0, a net-section collapse solution is used to assess surface crack stability. This solution takes the whole uncracked cross section into consideration with respect to net section stress. However, this assumption may be non-conservative. A remaining ligament failure criterion, or an elastic plastic-fracture mechanics criterion, may be more appropriate for predicting surface crack failure. Therefore, for Version 2.0, a more sophisticated surface crack stability criterion that still addresses the differences in primary and secondary stresses should be considered.

8. Simplistic inspection model

Project schedule constraints led to only the inclusion of the POD model for inspections in the Version 1.0 code. As described in Reference [4], much more work had been completed on the development of a flaw sizing model, repair/remediation techniques, and post-repair crack distributions. However, these were typically developed for circumferential cracks in surge nozzles. As the development of Version 2.0 begins, axial cracks will have to be considered, as well as other geometries, and possibly other weld types. Therefore, for Version 2.0, the inspection module development should continue.

9. Simplistic mitigation model

As with the inspection model, the mitigation model in the Version 1.0 code was very simple to demonstrate the credit that may be obtained from the use of mitigation. However, there are currently many different PWSCC mitigation strategies that can be incorporated into the code. These include:

- Weld overlay, (both full structural and optimized)
- Mechanical stress improvement
- Weld inlay/onlay
- ID surface treatment
- Inspection-based/material replacement mitigation (e.g. Alloy 690 pipe with Alloy 52/152 welds)
- Detailed incorporation of chemical mitigation.

The stress-based mitigation scheme in the Version 1.0 code was incorporated to demonstrate the feasibility of including mitigation into the code. It was also selected due to the ease of implementing the technique. Some of the other techniques will require significant additional effort to incorporate, e.g., weld overlay affects not only stress, but also a change in the wall thickness, material strength, toughness, and crack growth. These details were not incorporated into Version 1.0 due to the project schedule constraints. Those mitigation techniques currently being implemented by industry should be considered for the Version 2.0 code.

9.3.2 Framework

As with the models selected for implementation into Version 1.0 of the code, certain assumptions and limitations were imposed early in the developmental process. Some of these were due to pilot study deadlines, while others were byproducts of the decisions made in the developmental process. As with the model selections, the development of the framework was meant to demonstrate the feasibility of a modular-based probabilistic framework. Even with this programmatic limitation, certain framework limitations were obvious and should be avoided as the developmental process for Version 2.0 begins. Some of these limitations are discussed below.

1. Inputs and outputs

For the pilot study, a majority of the work was spent in the development of the modules and framework coding. For both SIAM and GoldSim, the input and output structures were unrefined due to the pilot study schedule and the need to only demonstrate feasibility of the process. In both cases, the definition of input parameters, selection of output parameters, and location and meaning of output files were sometimes vague and difficult to understand and follow. Since the input of data and the presentation of results is key to usability for any software tool, it is recommended that more emphasis be placed on development of the input and output structures for future versions of xLPR.

2. Uncertainty classification and analysis

The classification of uncertainty is a difficult task. In many cases, the data available that describe the uncertainty may be sparse, and the choice of the distribution to fit to that data may be arbitrary. The choice of which parameters are uncertain or constant, the classification of this uncertainty (aleatory or epistemic) or the selection of distribution to represent uncertainty may greatly change the results of the analyses. For example, in Version 1.0, wall thickness was treated as a constant, yet it affects the calculation of stresses, stress intensity factors, crack growth and stability, and the leak rate. As illustrated in Section 8.5.2, changing the crack initiation parameters from epistemic to aleatory had a large impact on how the uncertainty in the rupture probability was calculated and interpreted by the analysts. For the input to Version 1.0, the uncertainties for some key input parameters were classified as either aleatory or epistemic. For the input to Version 2.0, partitioning the uncertainties for all key input parameters into both epistemic and aleatory components should be considered, since, in most cases, both are present, although the epistemic is typically the larger of the two uncertainties.

Uncertainty classification and quantification is not trivial and needs involvement at all levels of development of a complex system. The Inputs Group has a major role in describing the uncertainty of each input, but that group has to work conjointly with the Models Group and the Computational Group in order to understand exactly the context in which each input will be used. Ideally (although not always possible due to time constraints), uncertainty characterization should follow a cycle because, once sensitivity analysis is performed, the results should be communicated to the Models Group and Inputs Group in order to check that they are reasonable, and that each part of the code, as well as the code as a whole, perform as expected and give reasonable results. As the xLPR program progresses, it is recommended to follow such an approach, insuring that results are reasonable and explainable, and that no uncertain quantity has been incorrectly characterized. It is common to propose “conservative” values for parameters to insure that some extreme cases are not under-represented. However, conservatism can lead to over-estimates of the probability of leakage or rupture and invalidate the model. Moreover, conservatism at a subsystem (i.e., model) level can become non-conservatism when implemented at the system level. Most of the uncertainty characterizations should be revisited at least once as a part of future xLPR development.

3. Importance sampling

The calculation of low probability events for a complex system with a variety of random inputs can be extremely difficult using standard sampling techniques. Not only are an extreme number of standard realizations required, the data storage capacity for running such analyses is prohibitive. Since many of the low probability events occur when the tails of the input distributions are controlling the event, importance sampling is required in order to produce acceptable results in a reasonable time, while still maintaining reasonable storage capacity limits. As has been discussed in this report, low probability events ($<10^{-6}$) can be calculated with reasonable ease when the correct variables are importance sampled. However, multiple sensitivity analyses must be conducted before it is evident which variables are important to the output of interest. The use of importance sampling is conditional to a good understanding of the system and the effect of input parameters on the outputs of interest. The choices of the distribution used and, to a lesser extent, the parameters selected, are mainly based on user experience. A bad selection may focus the analysis on the wrong area (either an area without interest, or an area of such low probability of occurrence that it will not affect the final result)), which can make the importance sampling useless, or even worse in some unlucky conditions. Therefore, it is recommended that the adaptive sampling methods described in Section 4.4 and Appendix A be further developed. In addition, many other optimization and reliability methods have been developed to handle this issue, and it is recommended to study the possibility of using some of these techniques in future versions of the xLPR code.

4. Data storage and handling

It was decided early in the framework developmental process to store all of the data from each realization generated from the probabilistic runs. The thinking behind this decision was that if the data were available, any results desired by the user could be obtained by post processing the data instead of re-running the code. However, this decision leads to result files that are extremely large and hard to handle. In fact, in many cases, the codes could not complete runs due to hard drive storage capacity limits. This was the case even though both codes used compressed formats for storing results. In addition, outputting the results into useable text format was a time consuming process that, in some cases, took more time than the code run time itself. Therefore, for future versions of the xLPR code, it is recommended that the data storage and handling process be revisited. It is possible to streamline the amount of data saved, and output only the data necessary for the results required.

5. Post processing

In the development of the xLPR Version 1.0 code, a majority of the development time was spent on creating a framework that conducted the calculations required for determination of the probability of rupture. It was only near the end of the pilot study project that focus shifted to the development of post processing tools needed for the calculation of rupture probabilities, the effects of uncertainty characterizations, and the effects of leak detection and inspection on the output. While the software developed was robust and formally checked, it was found to be somewhat confusing to use and understand. This issue stems directly from the lack of developmental time allotted for this software. Also, additional post-processing software may be needed for parameter sensitivity analyses when non-monotonic influences between inputs and outputs are present. Therefore, for future versions of the xLPR code, it is recommended that sufficient time be allotted to the development of post-processing software. It is imperative that correct and easy-to-use software be available to post process the large dataset that is developed from this complex probabilistic code.

10 Summary

This report documents the technical basis and results from the pilot study for Version 1.0 of the xLPR code. The xLPR code is being written cooperatively through an addendum to the US NRC and EPRI Memorandum of Understanding, and is intended to aid in assessing whether LBB-approved systems experiencing PWSCC quantitatively comply with regulations. In addition, the code is used to calculate the probability of various failure modes for RCS components. This first version of the code is meant to demonstrate the feasibility of both the management and coding structure needed for the development of a more robust and comprehensive code. To meet this goal, a pilot study was conducted that focused on a single location with a single degradation mechanism, i.e., pressurizer surge nozzle dissimilar metal weld with PWSCC.

For the pilot study, experts were gathered to determine the appropriate models and inputs needed to determine the probability of rupture for the pressurizer surge nozzle DM weld. These experts complied, coded, and verified, using a pre-defined configuration management process, the modules needed for the stated purpose. These modules included loads (with weld residual stress,), crack initiation, crack growth, crack coalescence, crack stability, crack opening displacement, leakage, inspection, and mitigation. In addition, these self-contained modules were incorporated into two distinct probabilistic frameworks developed using open source and commercial software. The frameworks were developed to control the time flow of the analyses, while linking the modules and properly accounting for and propagating the problem uncertainty.

To demonstrate the capabilities of the codes developed, a problem statement was developed that consisted of a base case and sensitivity studies. These runs focused on demonstrating the calculation of the probability of rupture with and without mitigation, with and without inspection, and with and without leakage detection, while also accounting for irreducible and reducible uncertainties. To propagate these uncertainties, the code used nested loops to separate the uncertainty types. Importance sampling was used for cases of extremely low probability of rupture.

The complete xLPR pilot study effort, which includes not only the code development efforts, but the management structure, the pilot statement problem, and the detailed analysis of the results clearly demonstrate that **it is feasible** to develop a modular-based computer code for the determination of probability of rupture for LBB approved piping systems. Many lessons learned throughout the process both from a managerial and computational standpoint were identified:

- Three very important aspects of the organizational structure are required for program success:
  - Dedicated team members, whose qualifications cover the important aspects of the group responsibility. It is advantageous if team members understand the basic computational process to aid in the incorporation of elements into the overall model.
  - Enthusiastic team and group leadership.
  - Efficient communication process within and among the teams.
- Configuration management is only a small portion of quality assurance. Establishing a program QA plan and controls for xLPR is the essential first step in the continuing development process. Each organization involved in the development of this code needs to develop a plan that identifies the process used to comply with the program QA requirements. The xLPR program needs to have a transparent and traceable CM system that will cover the xLPR code lifecycle.
- A well-written unambiguous software requirements document needs to be developed and followed for future xLPR versions.
- From the models standpoint, certain assumptions were made due to the limited scope of the pilot study. Best-estimate models with properly characterized uncertainties that cover all aspects of degradation of LBB-approved lines are required before this code can be used to assess full compliance with regulations.

- Focus should be placed on data storage, data handling, and post-processing to ease the difficulty of running this code for the end user.
- Importance sampling is necessary for the calculation of the probability of rupture in piping systems. Emphasis needs to be placed on processes and procedures for identifying the variables that need to be importance sampled. Adaptive sampling, or other reliability methods, needs to be considered.
- The classification of uncertainty is very important to understanding the uncertainty in the probability of rupture. Uncertainty is not trivial and needs involvement at all levels of development of a complex system. Knowing which variables control the rupture and what part of the uncertainty in those variables is epistemic and can be reduced will not only inform the regulators, but will also help direct future research in this area.

11 References

1 Sheron, B. and Modeen, D., “Addendum To Memorandum of Understanding Between U.S. Nuclear Regulatory Commission and Electric Power Research Institute On Cooperative Nuclear Safety Research - Extremely Low Probability Of Rupture (xLPR)”, October 2009

2 P. D. Mattie, C. J. Sallaberry, J. C. Helton and D. A. Kalinich, “Development, Analysis, and Evaluation of a Commercial Software Framework for the Study of Extremely Low Probability of Rupture (xLPR) Events at Nuclear Power Plants,” SAND2010-8480, Letter report to the Office of Nuclear Regulatory Research, US NRC, December 2010

3 H. B. Klasky, P. T. Williams, S, Yin, and B. R. Bass, “SIAM-xLPR Version 1.0 Framework Report” , ORNL/NRC/LTR-248, Letter report to the Office of Nuclear Regulatory Research, US NRC, September 2010

4 *Materials Reliability Program: Models and Inputs Developed for Use in the xLPR Pilot Study (MRP-302)*. EPRI, Palo Alto, CA 2011. 1022528

5 Materials Reliability Program: Advanced FEA Evaluation of Growth of Postulated Circumferential PWSCC Flaws in Pressurizer Nozzle Dissimilar Metal Welds (MRP-216, Rev. 1) EPRI, Palo Alto, CA: 2007. 1015383. MRP-216, Rev. 1.

6 Pensado, O., et al., “Assessment of Capabilities of Extremely Low Probability of Rupture (xLPR) Software – GoldSim and SIAM Version 1.0,” Center for Nuclear Waste Regulatory Analyses, 2011, San Antonio, Texas.

7 Harrington, C., and Rudland D., “xLPR Pilot Study Final Report: Project Summary, Outcomes, and Recommendations,” NUREG-xxxx, EPRI xxxxxx, 2011.

8 Agile software development. (2010, February 23). In *Wikipedia, The Free Encyclopedia*. Retrieved 17:03, February 24, 2010, from http://en.wikipedia.org/w/index.php?title=Agile_software_development&oldid=345922721

9 A. R. Daneshkhah, *Uncertainty in Probabilistic Risk Assessment: A Review*, http://www.shef.ac.uk/content/1/c6/03/09/33/risk.pdf, August 2004

10 Mosleh, A., Bier, V. M., and Apostolakis, G. *A critique of current practice for the use of expert opinions in probabilistic risk assessment*. Reliability Engineering and system safety, 20, 63-85. (1988).

11 Anderson, T.L., Thorwald, G., Revelle, D.A., and Lanaud, C., “Stress Intensity Solutions for Surface Cracks and Buried Cracks in Cylinders, Spheres, and Flat Plates,” Structural Reliability Technology final report to The Materials Property Council, Inc., March 14, 2000.

12 Chapuliot, S., Lacire, M.H and Le Delliou P., “Stress Intensity Factors for Internal Circumferential Cracks in Tubes Over a Wide Range of Radius over Thickness Ratios,” PVP-Vol. 365, Fatigue, Fracture and High Temperature Design Methods in Pressure Vessels and Piping, American Society of Mechanical Engineering, 1998.

13 Anderson, T.L., “Stress Intensity and Crack Opening Area Solutions for Through-wall Cracks in Cylinders, and Spheres,” Structural Reliability Technology final report to The Materials Property Council, Inc., January 29, 2003.

14 G.A. White, N.S., Nordmann, J. Hinkling, and C.D. Harrington, “Development of Crack Growth Rate Disposition Curves for Primary Water Stress Corrosion Cracking (PWSCC) of Alloy 82, 182 and 132 weldments,” *TMS (The Minerals, Metals and Materials Society)*, Proceedings of the 12th International Conference on Environmental Degradation of Nuclear Power Systems – Water Reactors, (2005)

15 Materials Reliability Program: Technical Bases for the Chemical Mitigation of Primary Water Stress Corrosion Cracking in Pressurized Water Reactors (MRP-263), EPRI, Palo Alto, CA: 2009. 1019082. [Proprietary]

16 EPRI Presentation, “Results of POD Assessment for Dissimilar Metal Welds in LBB Scope,” September 29, 2009.

17 D.O. Harris and D. Dedhia, *WinPRAISE 07: Expanded PRAISE Code in Windows*, Structural Integrity Associates, San Jose, California, July 2009

18 Brickstad, B., Chapman, O.J., Schimpfke, T, Schulz, H., and Muhammed, A., “Review and Benchmarking of SRM and Associated Software,” NURBIM Final Report D4, Contract FIKS-CT-2001-00172, DNV, Stockholm, 2004.

19 Scott, P., Olson, R. and Wilkowski, G.; “Development of Technical basis for Leak-Before-Break Evaluation Procedures”; NUREG/CR-6765, May 2002.

20 Rudland, D., Wang, Y., and Wilkowski, G.; “Comparison of Crack-Opening Displacement Predictions for LBB Applications”; *International Journal of Pressure Vessels and Piping*; Vol 79; 2002: pp. 209-217.

21 Kumar, V. and German, M.; “Elastic-Plastic Fracture Analysis of Through-Wall and Surface Flaws in Cylinders”; NP-5596, Research Project 1237-5, Final Report; Electric Power Research Institute; January 1988.

22 Paul, D., Ahmad, J., Scott, P., Flanigan, L., and Wilkowski, G., “Evaluation and Refinement of Leak-Rate Estimation Models,” NUREG/CR-5128, Rev. 1, June 1994.

23 Wilkowski, G.M., Olson, R.J., and Scott, P.M., “State-of-the-Art Report on Piping Fracture Mechanics,” U.S. Nuclear Regulatory Commission report NUREG/CR-6540, BMI-2196, February 1998.

24 G. Wilkowski, D. Rudland, D.-J. Shim, and H. Xu , “Technical Note on Critical Crack Size Evaluations for Circumferential Cracks in Dissimilar Metal Welds,” May 2007, ADAMS Accession number ML071560385.

25 Rahman, S.; "Net-Section-Collapse Analysis of Circumferentially Cracked Cylinders - Part II: Idealized Cracks and Closed-Form Equations"; Engineering Fracture Mechanics; Vol. 61; 1998; pp. 213-230.

26 D.-J. Shim, D. Rudland, G. Wilkowski, B. Brust, and K. Ogawa, “Development Of Z-Factor For Circumferential Part-Through Surface Cracks In Dissimilar Metal Welds,” PVP2008-61110, Proceedings of ASME-PVP 2008, 2008 ASME Pressure Vessels and Piping Division Conference, July 27-31, 2008, Chicago, IL, USA.

27 Gilles, P., and Brust, F. W., “Approximate Methods for Fracture Analysis of Tubular Members Subjected to Combined Tension and Bending Loads,” Proceedings of the 8th OMAE Conference, Hague, The Netherlands, 1989.

28 Goldsim User’s Guide, Version 10.0; Goldsim Technology Group: Issaquah, WA, February 2009.

29 A. Gubbi, D.O. Harris, C.H. Lange and D. Dedhia, “xLPR Version 1.0 Goldsim and SIAM Benchmark Testing with Comparisons to WinPRAISE Calculations,” EPRI-xxxxx, 2011

30 D. Rudland, D.-J. Shim and A. Csontos, “Natural Flaw Shape Development Due To Stress Corrosion Cracking,” Journal of Pressure Vessel Technology, February 2010, Vol. 132

Appendix A[*]

Comparison of Non-Standard Simulation Methods for Performing Extremely Low Probability Assessments to Monte Carlo Analysis

1 Introduction

While there are a variety of simulation methods for performing risk analysis, the following three were utilized in xLPR:

(1) Monte Carlo
(2) Latin Hypercube Sampling (LHS)
(3) Discrete Probability Distributions (DPD)

Furthermore, use of the importance technique for the last two sampling strategies was investigated.
The next section describes the theory followed by a discussion of the applications.

2 Monte Carlo Analysis

The basic principles of Monte Carlo analysis are straightforward. Given a set of inputs {x_1, x_2, …, x_N} that are uncertain and described by a Probability Density Function (PDF) denoted as f(x_i), a cumulative distribution function (CDF), F(x), can be constructed by integrating f(x_i):

$$F(x) = \int_{-\infty}^{x} f(z)dz$$

Figure 1 shows a comparison of CDFs for a normal and lognormal distribution with a mean of 20 and a standard deviation of 8.

One way to perform a Monte Carlo analysis is to uniformly select a random number, denoted R, between 0 and 1, and to invert the CDF to obtain a value for each x_i. The procedure is repeated as many times as necessary to converge to the desired solution:

$$x_i = F_i^{-1}(R)$$

Figure 2 shows one such sampling for both the normal and lognormal distributions. In this example, R is 0.5. Note that the value which would be selected for the normal case is 19.649, while for the lognormal case, R is 21.052. These values of x_i are input to the equation and a response is calculated, denoted as Y_1. The entire process is repeated to generate a set of responses, Y_K. These responses are then representative of the distribution of the responses that would be generated if an infinite number of samples were taken. How representative they are depends upon the value of K. The larger the value of K, the more accurate the representation.

[*] This appendix was written by Robert E Kurth, Senior Research Leader, Probabilistic Mechanics, Engineering Mechanics Corporation of Columbus.

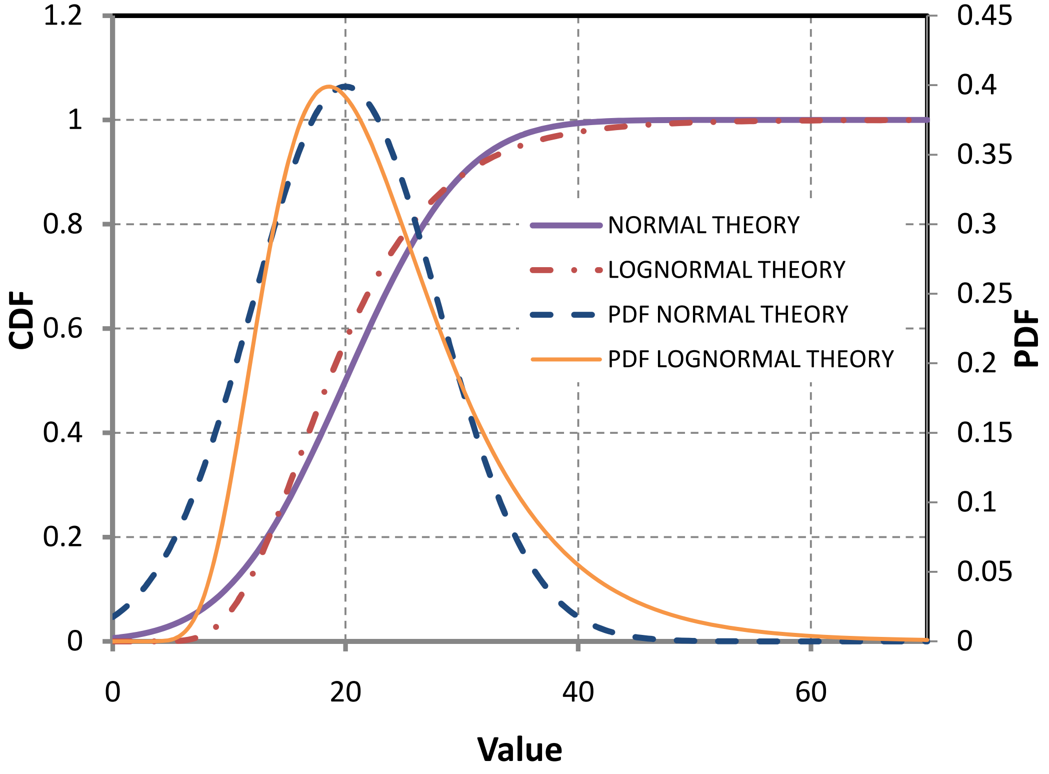


Figure 1 PDF and CDF for Normal and Lognormal Distributions

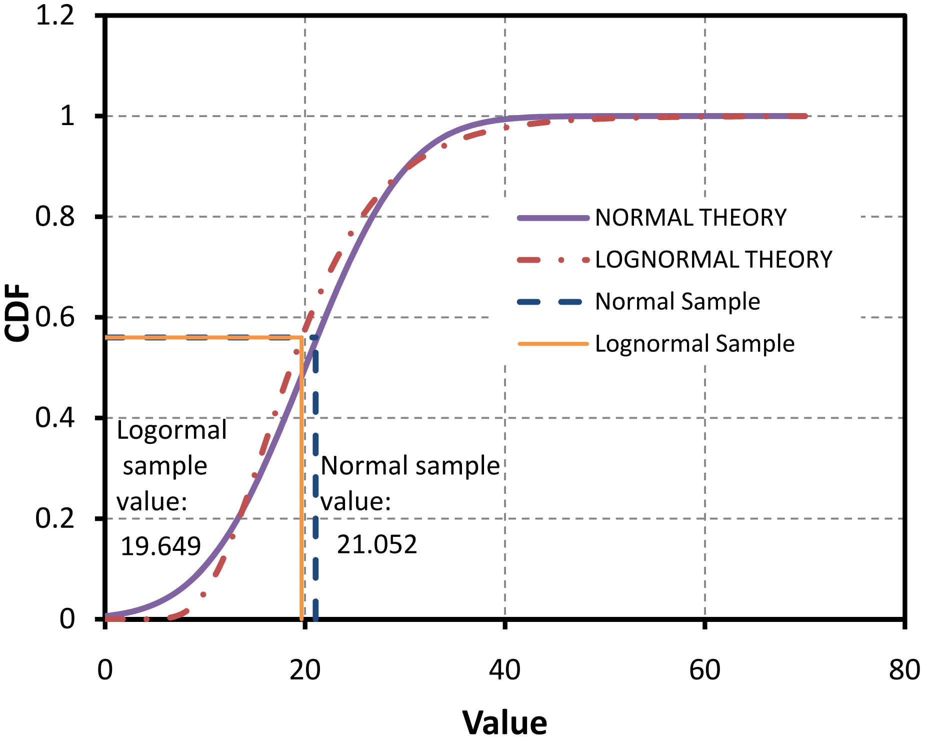


Figure 2 Monte Carlo Sampling Illustration

The Monte Carlo method is almost always the method of choice when the response calculation is fast. However, when there are limitations on the number of samples that can be generated, alternative methods need to be examined.

3 Latin Hypercube Sampling

The issue of sampling for low probability events has been a topic of intense scrutiny over the years. Many strategies have been employed, but it is not the purpose of this discussion to provide a survey of these methods. Rather, the focus of this section is on one of these methods that has been used extensively in Probabilistic Risk Assessment (PRA): Latin Hypercube Sampling (LHS).

The purpose of LHS is to provide a "dense" stratification over each individual input or process to a physical model. The concept of density over each input does not extend necessarily to the input hyperspace: one of the major assumptions of LHS is that most of the uncertainty will be driven by individual input parameters, and not from conjoint influence of these parameters. Moreover, while the main purpose of LHS is not to estimate extreme quantiles, but to capture where most of the uncertainty lies. Therefore, it does warrant a dense coverage of the response space.

The LHS is constructed by dividing each input distribution into N equal probability intervals. This is done for each of the inputs. The first interval for the first variable (i.e., $x_{1,1}$, where the first subscript index indicates the variable number, and the second subscript index indicates the interval selected) is then randomly paired with an interval from the second variable, leading to a couplet of ($x_{1,1}$, $x_{2,I}$), where I is the selected random interval for variable 2. If there is a third interval, this couplet is randomly paired with an interval from the third variable, leading to a triplet, ($x_{1,1}$, $x_{2,I}$, $x_{3,J}$), where J is the random interval selected for the third variable. If there are M random variables, this process is repeated M-1 times leading to an M-tuplet ($x_{1,1}$, $x_{2,I}$, $x_{3,J}$, …, $x_{M,K}$). To obtain the actual value of $x_{\alpha,L}$, a random value would be generated according to the PDF of the variable, α, from interval, L. This M-tuplet then becomes the input that generates a single response. To obtain the next set of inputs, the same process is repeated, except that each selected interval cannot be selected again. Thus, a *sampling without replacement* scheme is used. This implies that there will be exactly N responses generated. Thus, for M variables, there are N^M possible combinations of intervals. The LHS design will therefore sample N^{1-M} fraction of the response space.

Returning to the example in the Monte Carlo Analysis, a five point LHS design is examined.* Figure 3 shows a possible LHS sample.

* Five points are far too few to consider for a realistic assessment. This value is used solely to illustrate the method.

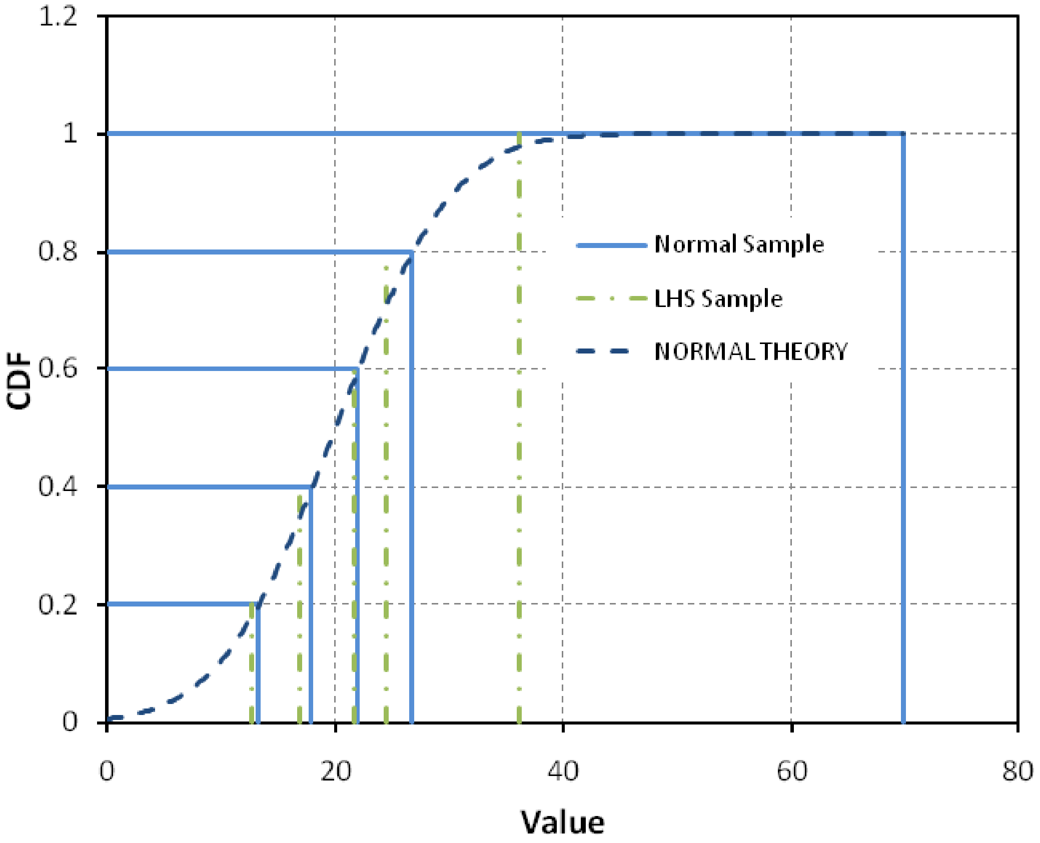


Figure 3 LHS Sample - Example 1

In Figure 3, the blue lines provide the limits for each interval, while the green lines indicate the values that could be used for each x_i in the response evaluation. Each of these five values would be paired, without replacement, with the other inputs, leading to a total of five response evaluations. Figure 4 shows, for the same LHS design, how the input values could change using another set of random numbers to select a value within each interval.

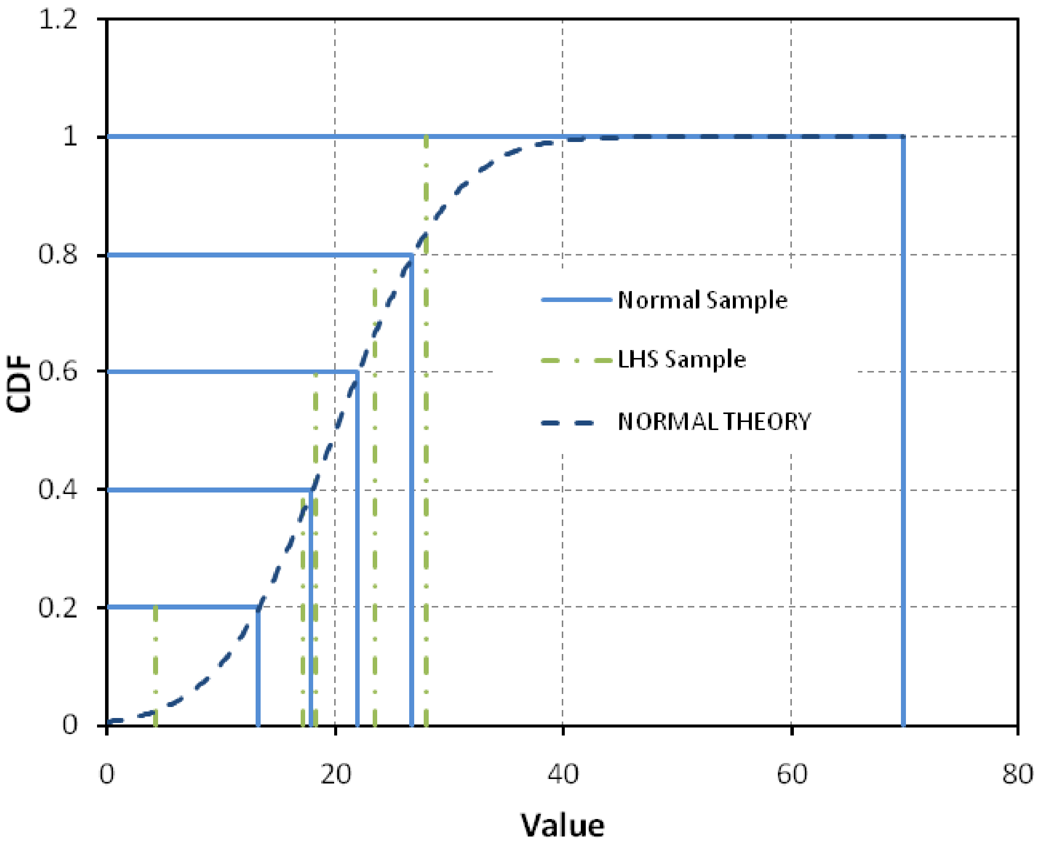


Figure 4 LHS Sample - Example 2

4 Discrete Space Sampling

In the LHS sampling, after an interval is selected, a value has to be sampled within that interval. A modified version of this technique would employ the same strategy as the LHS method but use the conditional mean of the interval. This modified version for LHS comparisons is employed since duplicate designs do not have to be run to compare similar strategies.

The goal is to calculate the response of a model to a variety of input values. The response of the system is denoted as $\Re$, the inputs to the analysis are denoted as the vector **x**, and the relationship between $\Re$ and **x** is denoted as f. Then:

$$\Re = f(\mathbf{x}),\ \mathbf{x} = (x_1, x_2, \ldots, x_N)$$

The function, f(x), may or may not be analytic. In the xLPR code, f(x) is not analytic. Rather, f(x) represents a complex computer analysis. The input vector, **x,** consists of several random variables. The following demonstrates a simplified example that allows the methodology to be discussed in a simple, easy-to-understand manner without the necessity of describing the details of the engineering analysis. The example selected is the addition of two random variables. Specifically:[*]

$$\Re = x_1 + x_2$$

[*] Different formulas would apply if the variables are not independent. For illustration, only those formulas for the independent case are provided.

If both of the inputs can be represented by a normal distribution with a mean value of μ_k and a standard deviation of σ_k for k equal to 1 and 2, the mean and variance for $\Re$ can be determined as:

$$\mu_{\Re} = \mu_1 + \mu_2$$
$$\sigma_{\Re}^2 = \sigma_1^2 + \sigma_2^2$$

In the classic Monte Carlo analysis, a random number is generated, and the value of x_1 is obtained by inverting the normal distribution CDF. A second random number is selected, the value of x_2 is obtained by inverting the normal distribution CDF, and the two numbers are added. This process generates a value for $\Re$ that is denoted as R_1. This process is repeated many times to generate a vector of responses denoted as $\mathbf{R} = (R_1, R_2, \ldots, R_M)$. As M approaches infinity, the CDF for the response, $\Re$, is approached. If the extreme tails of the response distribution are desired to be calculated, on average, many samples are needed. For the addition of two random variables, this is not a very severe limit with modern-day computational capability. This becomes critical when for realistic engineering analysis.
An alternative method for generating the response CDF is to limit the calculations to points in the discrete space. In this case, a Discrete Probability Density (DPD) function is defined for each of the inputs. Thus:

$$x_1 = \{(x_{1,1}, p_{1,1}), (x_{1,2}, p_{1,2}), \cdots, (x_{1,N_{BIN}}, p_{1,N_{BIN}})\}$$
$$x_2 = \{(x_{2,1}, p_{2,1}), (x_{2,2}, p_{2,2}), \cdots, (x_{2,N_{BIN}}, p_{2,N_{BIN}})\}$$

where each duplet, $(x_{i,j}, p_{i,j})$, represents the value of the j^{th} discretized value of variable x_i, as well as its "weight" or probability.

The response DPD is constructed by taking all possible combinations of the input DPDs. Thus:

$$R_1 = (x_{1,1} + x_{2,1}, p_{1,1}*p_{2,1})$$
$$R_2 = (x_{1,1} + x_{2,2}, p_{1,1}*p_{2,2})$$
$$\vdots$$
$$R_{NBin} = (x_{1,1} + x_{2,Nbin}, p_{1,1}*p_{2,Nbin})$$
$$R_{NBin+1} = (x_{1,2} + x_{2,1}, p_{1,2}*p_{2,1})$$
$$R_{NBin+2} = (x_{1,2} + x_{2,1}, p_{1,2}*p_{2,2})$$
$$\vdots$$
$$R_{NBin^2} = (x_{1,Nbin} + x_{2,Nbin}, p_{1,Nbin}*p_{2,Nbin})$$

The resulting DPD is a PDF since the following is true:

$$p_I p_J \leq 1 \forall I, J$$
$$\int_{-\infty}^{\infty} P dP = \sum_{I=1}^{N_{BIN}} \sum_{J=1}^{N_{BIN}} p_I p_J = \sum_{I=1}^{N_{BIN}} p_I \sum_{J=1}^{N_{BIN}} p_J = \sum_{I=1}^{N_{BIN}} p_I = 1$$

The above equations reflect the two conditions which must be satisfied in order for a function to be a PDF.

As an example, a three point discretization of the normal distribution Is shown in Table 1.

x_i	p_i
8.909	0.33334
10.000	0.33333
11.091	0.33334

Table 1. Three Point DPD for N(μ, σ) = N(10,1)

For a normal distribution with a mean value of 10.0 and a standard deviation of 1.0, one-third of the distribution lies between negative infinity and $\mu - 0.42\sigma$ (approximately). The next third of the distribution lies between $\mu - 0.42\sigma$ and $\mu + 0.42\sigma$ (approximately). The remaining third lies between $\mu + 0.42\sigma$ and plus infinity. However, it is questionable where to place this probability mass. For the mean and standard deviation assumed, the end points are (-∞, 9.58), (9.58, 10.42), and (10.42,∞).

If the probability mass is placed at these endpoints, an artificial bias will be introduced into the response PDF. If the probability mass is placed at the *conditional mean of the interval,* then when a value is selected randomly in this interval, the error between the sampled value and the DPD value will, on the average, be too large in 50% of the samples, and 50% of the time the value will be too small. Thus, the error cancels out *on the average*. This is reflected in Figure 5.

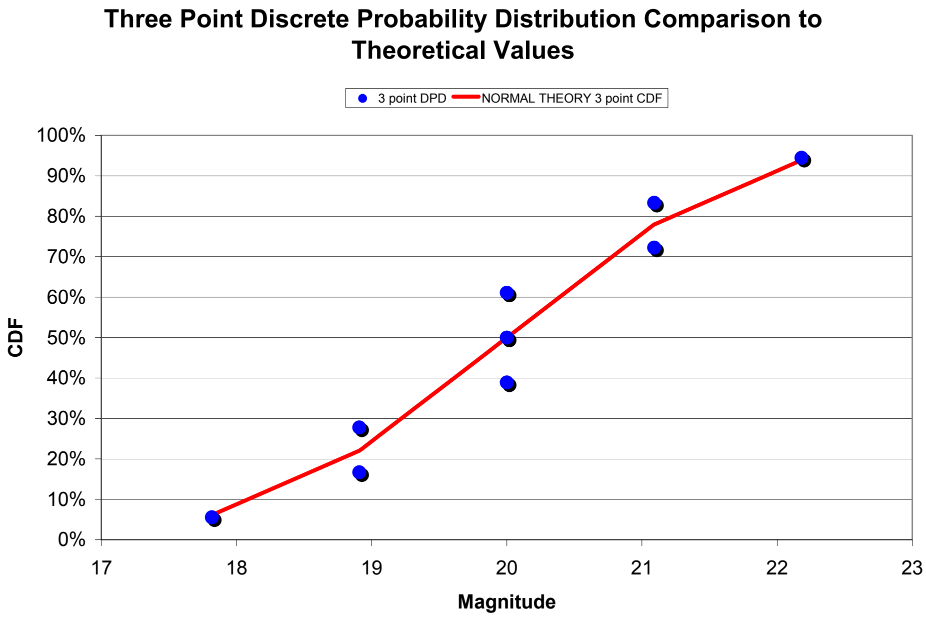


Figure 5. Three Point DPD Calculations

Again returning to the Monte Carlo example, a comparison is shown in Figure 6

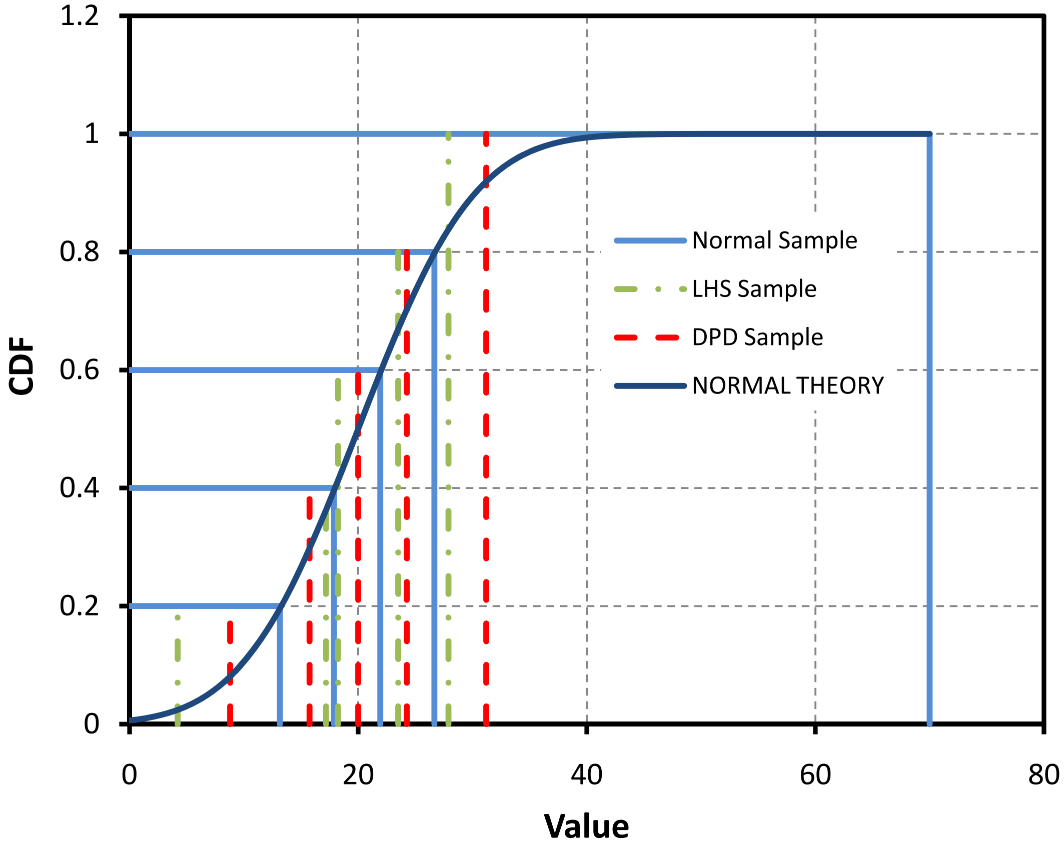


Figure 6 DPD Sampling Compared to LHS

In Figure 6, the DPD sample points are compared to the LHS points from Figure 4. It is important to note that the DPD will never change, while the LHS points will change each time a new design is generated.

5 Some Comparisons and Observations

5.1 Single Input Variable

The use of a 5-interval design for either LHS or DPD will not provide many points in the tails of a distribution. However, using replicate LHS designs can produce results in the tails, while the DPD design will never do so.

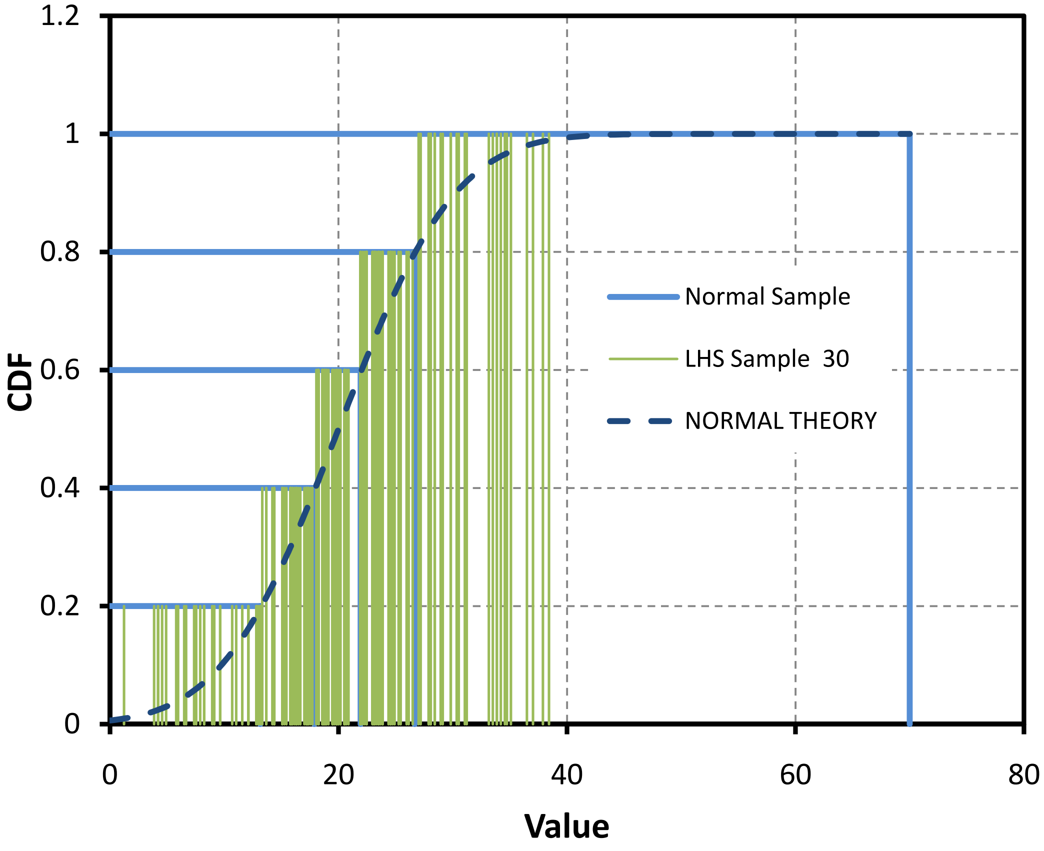


Figure 7 Replicate LHS-5 Designs versus DPD-5 Design

Figure 7 shows what occurs with a replicate 30 LHS design. As more and more replicates are performed, the LHS design will approach the Monte Carlo design, independent of the number of intervals selected. If there is only one input value to the analysis, the maximum number of distinct results from the DPD method is equal to the number of intervals selected.

On the other hand, increasing the number of intervals increases the DPD efficiency. If the number of intervals is set to 100 for the DPD method, we can estimate the lower tail probabilities more accurately. In Figure 8, a comparison of the DPD with 100 bins is compared to the LHS with 5 bins replicated 30 times. In this case, the two designs show similar sampling patterns.

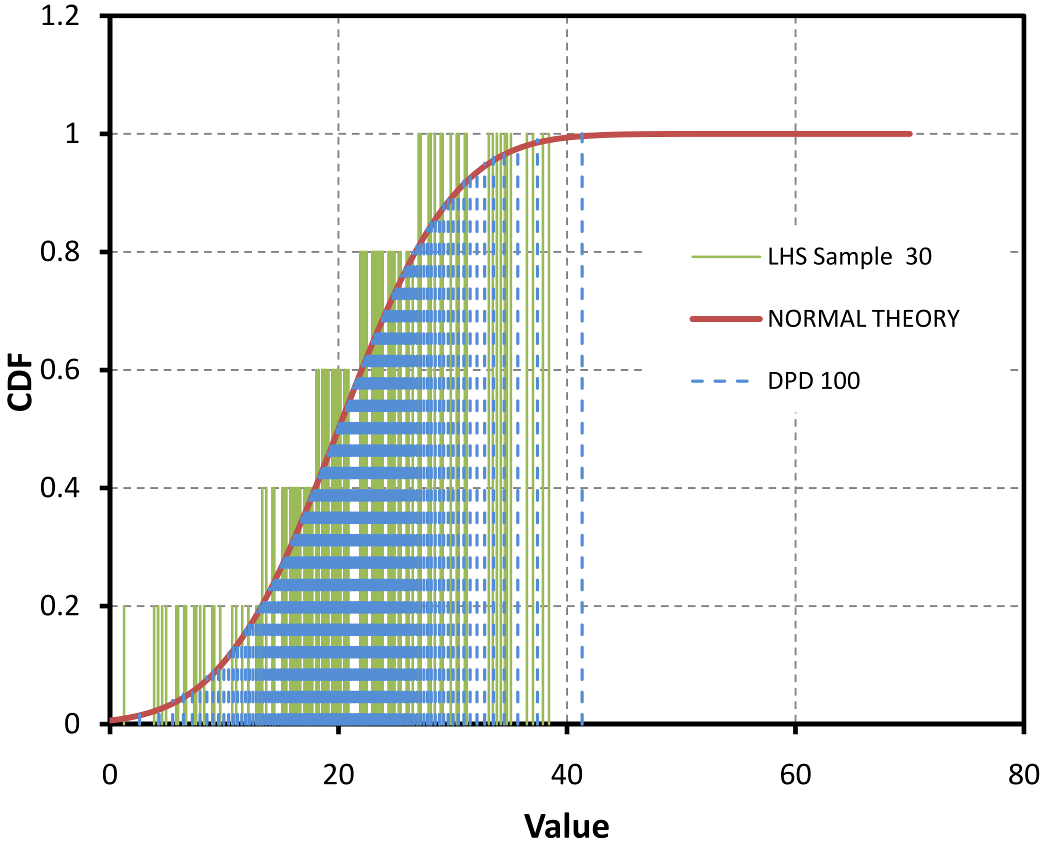


Figure 8 Comparison of DPD 100 with LHS 5 Replicate 30

There are very few analyses that have single inputs that are of critical importance. The differences with multiple inputs are discussed next.

5.2 Multiple Inputs for Complex Models

The current xLPR model has dozens of random inputs and complex physical models. Rather than describing this model in detail, a simple example is used to illustrate the concepts. For this example, it is assumed that there are three inputs to the model:

$$R = e^{Ax_1} + Bx_2^3 + \frac{C}{x_3}$$

with A = -1, B = -4, and C = 10. If 100 intervals are used for both LHS and DPD methods, the results shown in Figure 9 are obtained.

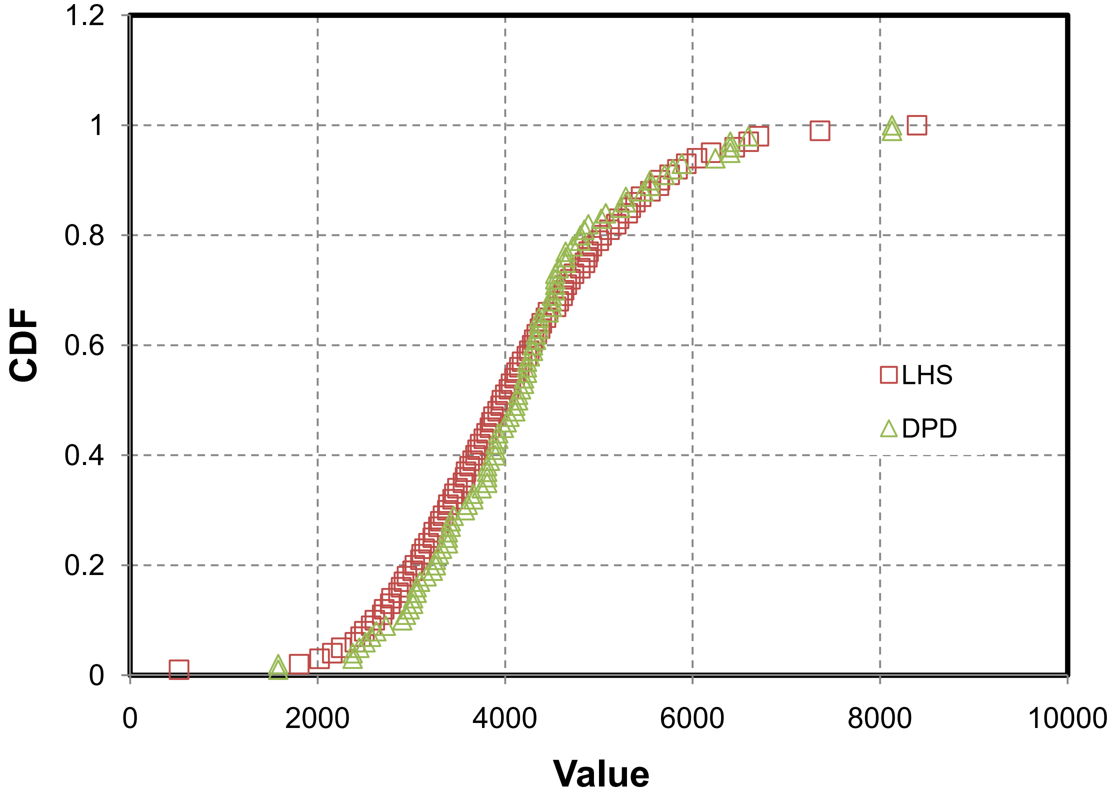


Figure 9 LHS and DPD Results for 100 Bins

Figure 9 shows that the estimated response distributions are not dramatically different. Because the DPD values are fixed at the conditional mean points for a small number of samples relative to the number of bins (on the order of less than 5 x N_{BIN}), the predicted CDF is not smooth because some intervals may not be sampled, while others can be sampled multiple times.

The real benefit of DPD is when portions of the distribution are sampled more frequently. Because LHS uses the same sampling strategy as Monte Carlo, there are no known a priori frequencies, or weights, with the LHS scheme. When the DPD method is used, it is known exactly how frequently these points occur, so responses can be generated that are not of equal probability. Such a scheme is illustrated in Figure 10.

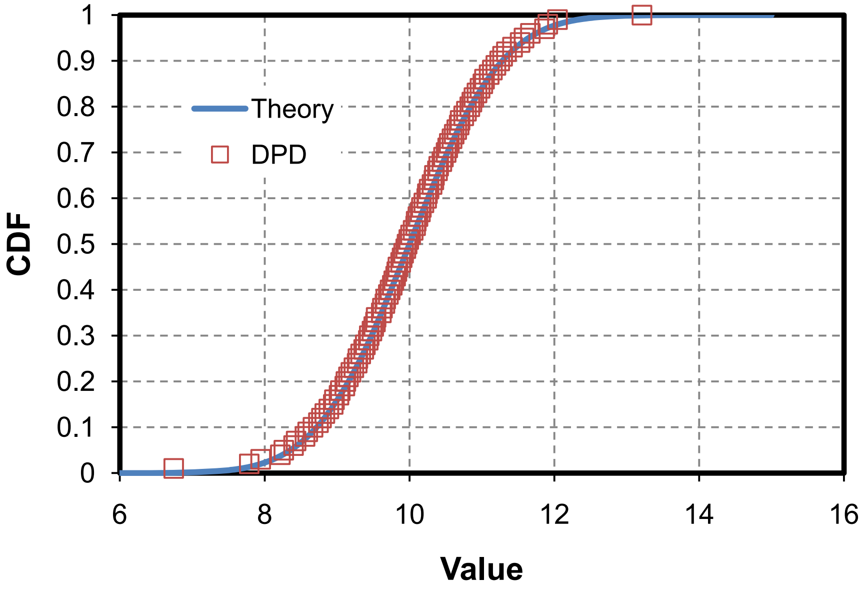


Figure 10 Comparison of DPD PDF and CDF to Theory

Figure 10 shows that there is no discernible difference in the DPD and theory. If this distribution is input for the x_i values, the response shown in Figure 11 is obtained.

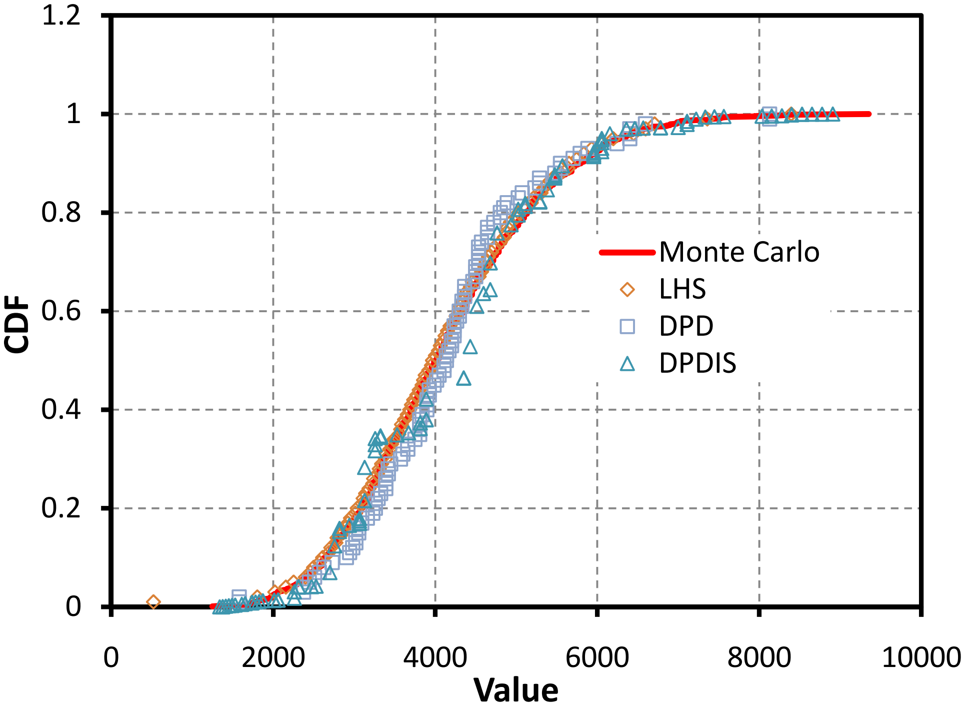


Figure 11 Response Distribution Using DPD Importance Sampling

Figure 11 shows that the lack of smoothness in the results is more pronounced in the 50 percentile region, but smoother near the tails. These results are further examined on a log scale in Figure 12.

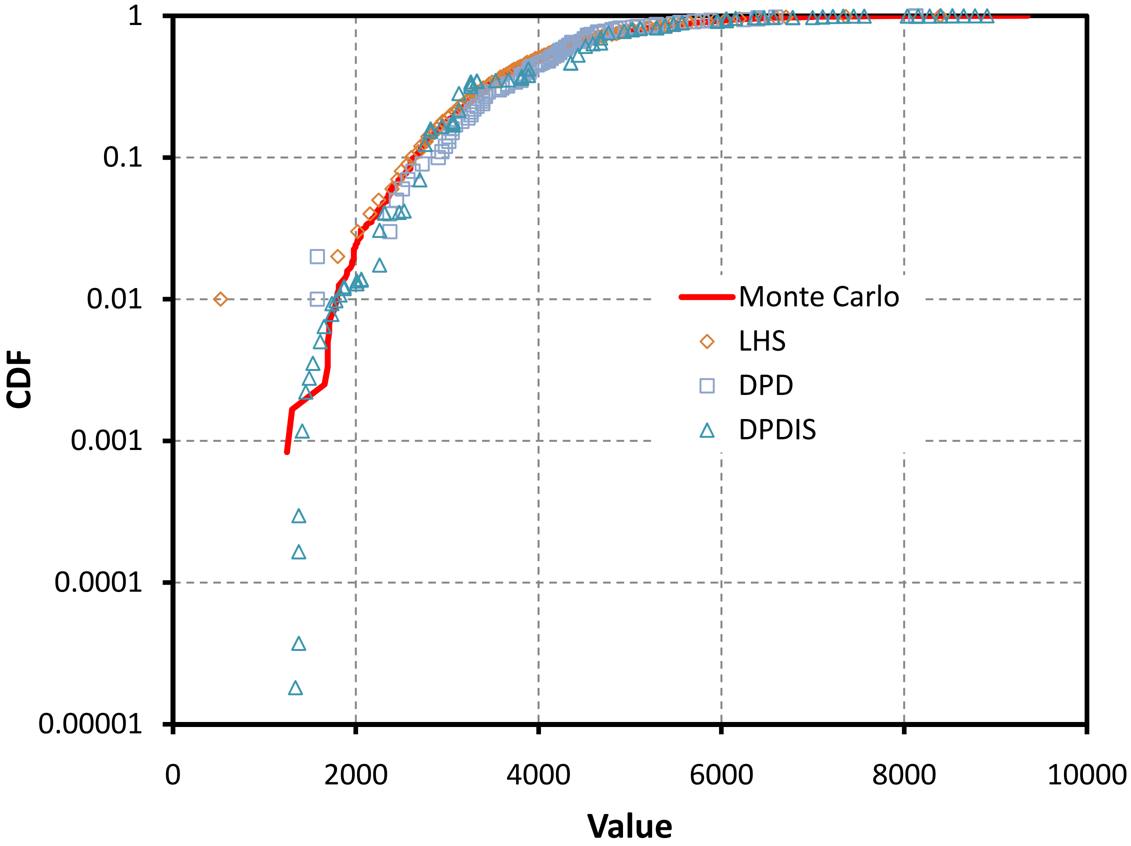


Figure 12 Response Distribution Using DPD Importance Sampling Log Scale

A classical LHS design will only be able to estimate a response on the order of 1/100. Using importance sampling, DPD provides an estimate for a response on the order of 1/10,000 using same sample size. While in both cases the estimate accuracy will improve by increasing sample size, this comparison shows the benefit of using importance sampling when extremely low probability events need to be captured.

6 Summary

The DPD and LHS methods both have advantages and disadvantages to solving probabilistic problems. When the same number of intervals is used, as long as they are large enough, each method gives similar estimates of the response distributions for a large range of the CDF. The response ranges from about $3/N_{BIN}$ to $1 – 3/N_{BIN}$, where N_{BIN} is the number of intervals. Increasing the sample size will increase this range. When extremely low probabilities are the interest of the study, efforts should be focused on the tail of the distribution for the output of interest. This requires the use of importance sampling over the “important” input parameters in order to densely cover the region of interest (in xLPR, this represents is the region of highest probability that leads to rupture for the considered scenario). Currently, DPD software allows automatic coverage of the tails of the distributions for selected inputs, and estimates the associated weighting factor for each resulting realization. While the LHS software does not allow for such a technique to be used, it has been demonstrated that a similar approach can be applied by modifying the initial distribution and calculating the resulting weight for each realization.

A useful strategy is as follows:

1. If the response of interest can be evaluated very efficiently, i.e., on the order of 1,000,000 per 4 hours, Monte Carlo may be used for any type of analysis that does not require estimates of the response of frequency less than 1 in 1,000,000.

2. If the response frequency desired is low, but the evaluation takes a long time (i.e., greater than several days) then:
    a. If mean behavior and the uncertainty in the mean is of most interest, then LHS will perform better because it provides a dense covering of the input space. For example, if the mean time to rupture and the uncertainty in this estimate is desired, LHS will provide this in a more efficient manner.
    b. If rare events are of the most interest, the DPD method with importance sampling should be used to provide the most efficient method for obtaining low probability estimates.

The DPD and LHS methods offer significant improvements over the standard Monte Carlo simulation method. Each has been used to advantage in previous studies. The selection of one method over another depends on many factors. For example, it is desirable to characterize the source of uncertainty, LHS may be more useful. If the extreme tails of the hyperspace are desired to be examined, the DPD method with importance sampling may be more useful. A combination of both methods may also be most efficient. However, the foregoing examples have demonstrated that both methods can be implemented, both methods have their respective merits, and each method can be used to demonstrate the uncertainty partition and/or importance sampling. In Version 2.0 of the xLPR code, a more detailed implementation of these approaches will be developed and implemented.

Appendix B
xLPR Pilot Study Problem Statement

1 Purpose

The intended purpose of the xLPR Pilot Study is to develop a prototype xLPR model and pilot study case leveraging existing fracture mechanics models and software coupled to both a commercial and open source code framework to determine the framework and architecture requirements appropriate for building a modular-based code with this complexity. The xLPR pilot study is being conducted to demonstrate the feasibility of the proposed developmental process and framework for a probabilistic code to address degradation mechanisms in piping system safety assessments. The pilot study will address the specific issue of assessing the probability of rupture of dissimilar metal (DM), pressurizer surge nozzle welds degraded by primary water stress corrosion cracking (PWSCC), particularly those previously assessed for which a considerable amount of publicly available information already exists. The pilot study will provide a short term, learning experience that should benefit the longer term program and code development by identifying areas requiring more focused effort.

The analytical output of the pilot study will be a probabilistic assessment of surge nozzle DM welds to include:

- Probability of leakage at various crack opening sizes
- Probability of rupture

These results will include a comparison of results with and without the effects of inspection and pre-emptive PWSCC mitigation. The pilot study will provide relative, order-of-magnitude estimates of piping rupture probabilities; such analysis will identify areas requiring more focused attention in the long-term study. Sensitivity studies will also be carried out to exercise, verify and debug the code.

2 Configuration Management/Quality Assurance

The xLPR pilot program model and results of the analyses described in this document, as well as any additional analyses, will be used to evaluate or determine the longer term program and code development requirements. In making recommendations for the best computational framework, models and input distributions for use in the pilot study, a gap assessment will be conducted, identifying gaps in both data and research. This gap assessment and lessons learned over the course of the pilot study will be used to identify and prioritize research recommendations. The final outcome of the pilot study will be a research plan for moving forward to attain the long term goal of a fully modularized, probabilistic assessment tool for primary piping systems. Therefore, following appropriate, controlled processes and procedures is paramount to developing a traceable and reliable xLPR model and analysis. This process will form the foundation necessary to demonstrate compliance with QA requirements during Phase II of the xLPR program.

The CM process utilized for xLPR Module Development and Framework Development will be utilized for the Model Parameters and Inputs and the xLPR Model Production Runs and Uncertainty/Sensitivity Analyses for the pilot study test case. The xLPR pilot study program utilizes a systematic approach to ensure the basic fundamentals of a QA/CM program are met, including: 1) Access Control; 2) Version Control; 3) Verification/Validation (e.g., Checking); and 4) Traceability (e.g., Documentation). The CM program ensures that a systematic approach is used to meet the requirements and includes documentation of each step in the process. Completed xLPR model and sensitivity analyses are stored in a controlled subdirectory on the xLPR file server. The pilot study problem analyses will be archived for traceability in the production runs directory of the xLPR web site hosted by Battelle using the Microsoft SharePoint process and document management software:

https://websps1.bettelle.org/nrcnureg/home/xLPR_CM

2.1 Production run CM process/guidelines

Step 1: LOG THE ANALYIS - For each model run, a unique alphanumeric designator for the analysis shall be selected using a consistent naming convention established for the pilot study analyses. The unique alphanumeric designator for analysis will be used to identify the input and output files as well as any plots or data tables created from the analysis.

For example: GS_BETA_v2.01_M02_00400_000.gsm is the name of the xLPR model run using the GoldSim framework version 2.01, Module set M02, 400 realizations, case #000. GS_Beta_V2.01_M01_00400_001.gsm would be used for the next 400 realization analysis. GS_Beta_v2.01_M01_10000_000.gsm would be used for a 10,000 realization run.

A folder with this unique identifier should be created in the production runs directory on the electronic CM system.

CM > Production Runs > Beta Model Runs
(https://websps1.battelle.org/nrcnureg/home/xLPR_CM/Production%20Runs/Forms/AllItems.aspx?RootFolder=%2fnrcnureg%2fhome%2fxLPR%5fCM%2fProduction%20Runs%2fBeta%20Model%20Runs&FolderCTID=&View=%7b7E47AD2F%2dE402%2d4C5A%2dB405%2dA8B6BBBD4F48%7d).

Step 2: DOWNLOAD FROM A CONTROLLED SOURCE - For each unique analysis described in this document, the controlled xLPR model files (modules, framework, and inputs) should be downloaded from the controlled subdirectories on the xLPR file server. The electronic file server for controlled storage of xLPR model files is hosted by Battelle using the Microsoft SharePoint process and document management software and is Web accessible (https://websps1w.battelle.org/nrcnureg/xLPR_CM). Modules for the xLPR model are stored on the xLPR electronic file server along with their documentation and verification checklists in controlled subdirectories. The Input parameter spreadsheet with the base case values for the xLPR pilot study analyses are controlled and stored in a controlled subdirectory on the xLPR file server. The input values and parameters have been checked and verified for the base case. For each analysis, the controlled files (Modules, Input Spreadsheet and Framework) should be downloaded from the xLPR Share Point Site (CM > Controlled Files) to the local machine on which the model run will be executed. This step should occur each time, to ensure that the controlled files are used in the analysis.

Step 3: MODIFY THE INPUT AND/OR MODEL FILES- The input file spreadsheet, output files, and model framework file should be re-named using a unique alphanumeric designator as outlined in Step 1. Changes to the input data, parameters and/or model structure including use of alternative modules or logic (e.g., sensitivity case using direct method I for crack initiation module) need to be made to the renamed files and the changes should be logged using the xLPR Change Checklist. The intent of the analysis should be described (including the technical basis) in a Conceptual Description document.

Step 4: RUN ANALYIS/UPLOAD FILES- After the changes have been made and documented in the accompanying Change Checklist and Conceptual description, and the files have been saved, the analysis should be run. After the analyses have been completed and executed to the satisfaction of the analyst running the case, all changed input files, including changes to, or use of, additional modules not in the controlled file set, need to be uploaded to the folder created for the analysis in the production run directory on the xLPR CM share point server. The change checklist and conceptual description files need to be uploaded with the model files, as well as any files needed for a complete independent check/review of the analysis. All analysis results need to be stored on the CM site, including any plots and/or Excel files used to post process or evaluate the results, which may be used in the pilot program reports. Due to file size limitations in Share Point, not all of the model results can be saved. However, the model file used to run the analysis should be saved so that an independent check and review can be completed.

Step 5: INDEPENTENT CHECK/REVIEW – An independent check and review will be conducted and documented to verify the analysis was executed correctly. This includes verification that the changes made to the model were implemented correctly including traceability. This check needs to be completed by someone other than the originator of the analysis. The checker will review the input deck, model file, and compare with the changes listed on the change checklist and the objectives outlined in the conceptual description. The checker will initial the checklist when satisfied that the changes to the model file are correct.

Step 6: RESULTS CHECK/REVIEW – Verify any post-processed results, plots, additional calculations or documentation used to support a given case or set of cases. The additional files will be stored in the controlled subdirectory on the xLPR file server along with the model files. These should be independently checked and reviewed, with documentation that the check/review was completed.

3 Model Verification

Other conditions specific to the xLPR model, such as spatial, temporal, and stochastic discretization, convergence, and stability will be checked as part of both development and post-development activities. These and other xLPR model calibration activities will be documented in the xLPR Model Report. The following is a list and description of the analyses that will be used for model verification.

3.1 Deterministic run(s)

A deterministic run using constant input is used to verify against a hand calculation (using EXCEL) that the xLPR model framework is operating as expected. Two deterministic analyses will be run as outlined below.

1. Deterministic Analysis #1: Single Crack at 0 years, with no mitigation. The location of the crack is at the top of the weld (theta = 0 rad). The input deck for this case can is included with the controlled version of the inputs spreadsheet for xLPR.
2. Deterministic Analysis #2: Three Cracks at 0 years, with no mitigation. The inputs remain the same as for the first deterministic analysis, except that there are three cracks. The three cracks are the same size as the crack size used in Deterministic Analysis #1. Their respective locations are theta = 0 rad., 0.6 rad., and -1 rad. The input deck for this case is included with the controlled version of the inputs spreadsheet for xLPR.

3.2 Stability testing

Model stability testing activities include three types of stability tests: statistical stability, temporal stability, and spatial stability or discretization. Collectively, these three tests are referred to as model stability testing.

1. Statistical stability testing involves a number of activities related to demonstrating that a sufficient number of stochastic realizations have been run to achieve a numerically stable mean, including: (1) determining confidence intervals (generating several replicates with different random seeds and using t-test) around selected output; (2) demonstrating numerical accuracy of the mean results by comparing the results of the base case with analyses using more realizations and different random seeds. The stability of mean and other quantiles will be considered for both aleatory and epistemic uncertainties.
2. Temporal stability refers to the use of an appropriate time step size necessary to achieve a stable solution. The time steps must collectively encompass the range of events and processes. The degree of stability will be shown in graphical comparisons of the results of the stability analysis, using time steps as short as one month, two months, six months, and one year.

4 xLPR Base Case Analysis

4.1 Probabilistic base case description

A probabilistic base case analysis will be run with an appropriate sample size using the Monte Carlo method. The probabilistic analysis is divided into two loops. The outer loop, capturing the epistemic uncertainty, would correspond to a sample of size nE. The inner loop, capturing the aleatory uncertainty, would correspond to a sample size of nA. A total sample size of nS = (nE * nA) will be used. The total number of samples and number of epistemic and aleatory samples will be determined.

The base case consists of the surge nozzle geometry, with the appropriate loads and inputs taken from published data. The main driver for PWSCC is the weld residual stress, therefore, for the base case, the weld residual stress distribution assumed is shown in Figure 1. In this figure, the surge nozzle is assumed to have an ID repair and an Alloy 182 fill-in weld for seating the thermal sleeve. It is assumed that the safe end weld is far away from the dissimilar metal weld.

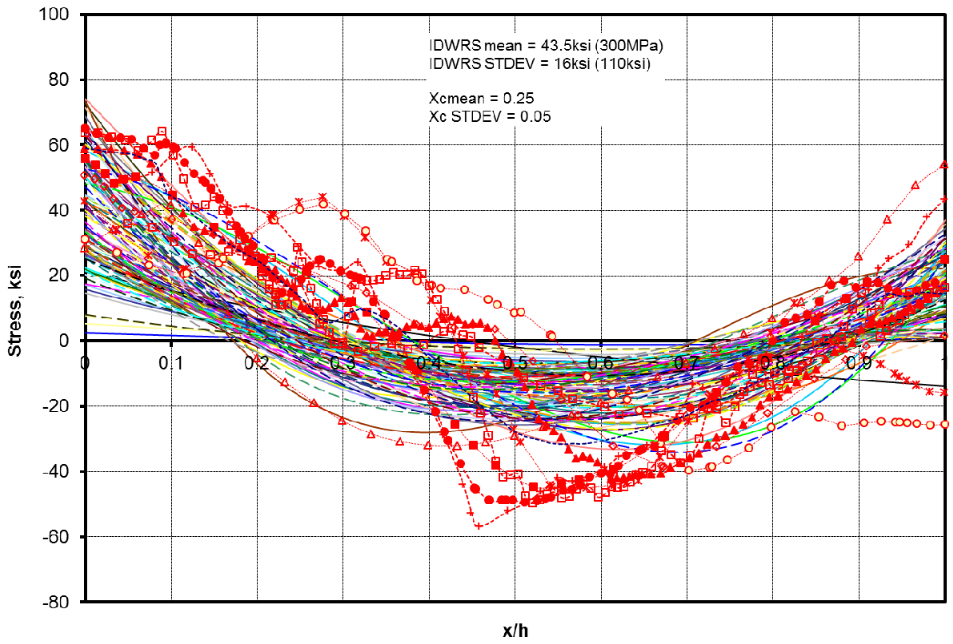


Figure 1 Base case weld residual stress

4.2 xLPR pilot study outputs

4.2.1 Type of analysis for sampling based methods

The purpose of this pilot study is to estimate the extremely low probability of rupture of a single weld. However, it is important to analyze several intermediate outputs rather than a single output. Since each output depends on its aleatory and epistemic set, it can be analyzed in several ways, depending on whether one integrates over aleatory uncertainty, epistemic uncertainty, both, or neither :

1. Expected value over aleatory uncertainty: these values are obtained by averaging over aleatory uncertainty. Because simple Monte Carlo sampling techniques are used for aleatory uncertainty, the averaging is a classical sum divided by the number of parameters, and then including correction due to conditionality. Critical failure will not occur if there is no crack. Therefore, only realization with at least one crack is considered. The probability of having at least one crack must be estimated for each realization and used as a corrective term for each parameter in the following way: Expected value = P(no crack)*Value_if_no_crack + P(at least one crack) * Value_if_at_least_one_crack
2. Expected value over epistemic uncertainty: this approach is the symmetric of the previous approach. It may be harder to implement as some of the aleatory uncertainty may depend on some epistemic values. As LHS is used to generate epistemic uncertainty, the simple arithmetic mean can be used to estimate expected value.

3. Aleatory variation for a fixed epistemic set: Once one epistemic set is selected, aleatory uncertainty is represented. For aleatory uncertainty representing the risk, a classical representation would be a CCDF. The display of one CCDF for each epistemic set will lead to a horsetail plot of CCDFs. As for the expected value over aleatory uncertainty, it is important to take into account the set of futures involving no cracks at all.
4. Epistemic variation for a fixed aleatory set: A symmetrical analysis of the previous approach,whicht is less used because results are harder to interpret (except on the basis of a selected future), and because the future (aleatory set) depends on some epistemic value.

Methods 1) and 3) will be considered here. Method 1) will be complemented with an estimate of the mean and quantiles of the expected values.

4.2.2 Outputs to be generated

For an xLPR run in the pilot study, the results file contains all output for each realization and each time step. This bulk data is to be processed to determine the following output list.

1. Time-dependent crack depth (expected over aleatory uncertainty) for any relevant crack.
2. Time-dependent half crack length (expected over aleatory uncertainty) for any relevant crack.
3. Time-dependent fractional surface area cracked (expected over aleatory uncertainty).
4. Time-dependent stress intensity (expected over aleatory uncertainty) for any relevant crack.
5. Scatterplot stress-intensity vs. crack area for specific times (10 years, 30 years, 60 years) (expected over aleatory uncertainty).
6. Time-dependent probability of non-detection (expected over aleatory uncertainty) for any relevant crack.
7. Average duration of surface crack (over aleatory uncertainty and over all cracks).
8. Time-dependent leak rate (expected over aleatory uncertainty) for any relevant crack.
9. Total time dependent leak rate (expected over aleatory uncertainty).
10. First leakage probability as a function of time.
11. COA>1-inch equivalent break diameter (506.71 mm^2) probability as a function of time.
12. COA>3-inch equivalent break diameter (4,560.37 mm^2) probability as a function of time.
13. Rupture probability as a function of time.

Each of these outputs will be generated for the base case, but only the final four outputs will be generated for the sensitivity analysis cases.

5 Sensitivity Analyses

A set of sensitivity analyses will be conducted to demonstrate xLPR model functionality. Sensitivity analyses are used to evaluate or quantify the impacts of some of the modeling assumptions and various alternative model processes not selected for the base case analysis.

5.1 Effect of safe end length

The stainless steel safe end weld that attaches the safe end to the surge nozzle piping causes a through thickness bending stress that can reduce the tensile inner diameter stresses at the dissimilar metal weld. The extent of the effect on the dissimilar metal weld is a direct function of the length of the safe end. In the base case for the pilot study, it was assumed that the safe end was long enough such that the safe end weld did not affect the stresses in the dissimilar metal weld. This case will consider a short safe end length. For the safe end length considered, the distribution of weld residual stress is shown in Figure 2. The symbols in the figure represent the detailed finite element analysis predictions of weld residual stress, while the lines represent the fit to that data using the weld residual stress model in xLPR. A unique distribution for the axial stress component of the epistemic parameter S0_WRS and Xc are shown in Figure 2. It is assumed that the distribution is normal and:

- S0_WRS = -16.2 MPa mean and 117 MPa Stdev (maximum=300 MPa, minimum =-300 MPa)
- Xc = 0.18 mean and 0.036 Stdev (maximum = 0.5, minimum = 0.1)

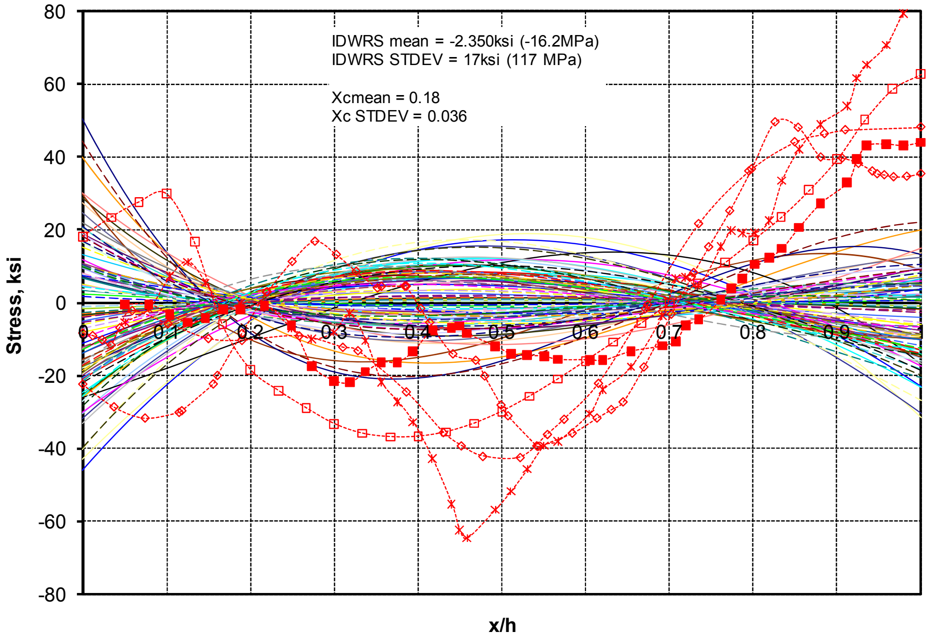


Figure 2 Weld residual stress distribution for surge nozzle with safe end weld

5.2 Effect of stress mitigation

Mitigation analyses will be run as part of the beta model evaluation. These runs will evaluate different mitigation times, as well as the mitigation effectiveness over the representative distributions for Sigma0_wrs_mitigated and Xc_mitigated. Three sensitivity cases (n = 10,000) will be run for mitigation:

1. Mitigation time of 10 years.
2. Mitigation time of 20 years.
3. Mitigation time of 40 years.

The distribution of weld residual stress to be used for the mitigation is shown in Figure 3. For these cases, a normal distribution should be assumed with:

- Sigma0_wrs_mitigated = (-344.75 MPa mean and 34 Stdev, with a minimum = -447 and a maximum = -242)
- Xc_mitigated = (0.38 mean and 0.038 Stdev, with a minimum = 0.26 and maximum = 0.5)

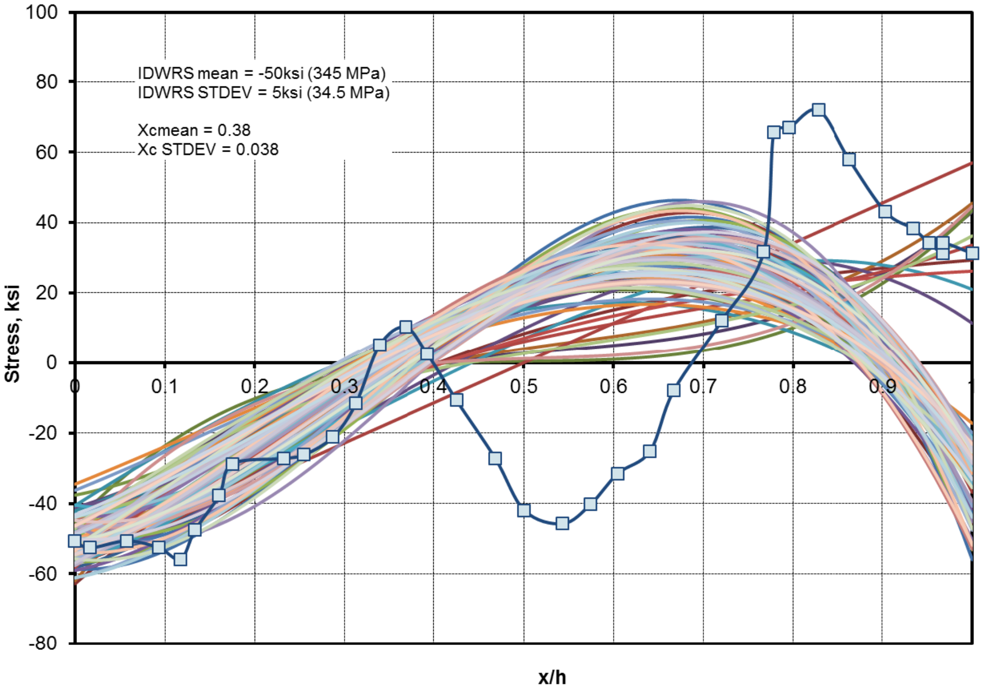


Figure 3 Mitigated weld residual stress for beta sensitivity analyses

5.3 Crack initiation model uncertainty

A sensitivity case will be run that considers the crack initiation model uncertainty. The crack initiation module includes three alternative models for crack initiation. Method 2 was used in the base case analysis. Method 1 will be run for comparison to the base case to evaluate the effect of the initiation model uncertainty on the results.

5.4 Chemical mitigation

A sensitivity case will be run that considers the effects of increasing the hydrogen concentration in the water on crack growth. Even though there is some documented evidence of the effect of hydrogen and zinc on crack initiation, the models are not mature and are not currently included in the beta code. A comparison of mean results will be conducted.

For the base case, the hydrogen concentration was set at 25 cc/kg-STP. For these analyses, the hydrogen concentration will be increased to 50 and 80 cc/kg-STP to demonstrate the impact.

5.5 DPD analysis

A sensitivity case will be run that substitutes the DPD method for sampling the uncertain parameters. A DPD analysis with importance sampling of Sig0_WRS, Xc, and other parameters identified as important, will be run using the safe end sensitivity case.

6 Post-Processing Analyses

The base case and sensitivity analyses will need to be post-processed using a set of tools developed to evaluate the extremely low probability of failures. The desired output, defined in Section 3.2 of this problem statement and including inspection and leak detection will be evaluated using post-processing analyses and the post processing code developed by Sandia National Laboratories for the xLPR Pilot Study.

6.1 Leak detection capability

In order to demonstrate leak detection capability, the output of select cases will be analyzed to demonstrate the effect on the output probabilities (last four bullets of Section 3.2.2) of leak detection limits of 0.1, 1, 10, and 50 gpm. The cases to be analyzed will include:

- Base case
- Short safe end case

It is not necessary to redo any specific analysis when leak detection is changed. It is assumed that once a leak is detected, the weld is replaced and won't fail again. Therefore, all calculations are done assuming that the leaks are not detected, leading sometimes to pipe rupture. The user can select a detection threshold that will lead to a correction of output data of interest if a leak is detected.

It is possible also to suppose that the weld is replaced by a weld of similar strength and then to recreate a potential history based on the previous runs after the leak is detected.

The credit leak detection software to be used for this effort can be found on the share point site at CM > Beta Model Dev > Modules > TRANSFORMERS v1.0. See the associated documentation for details on the features of this module.

6.2 Inspection schedule

In order to demonstrate the effect of in-service inspections on the output probabilities, the output of select cases (last four bullets of Section 3.2.2) will be analyzed using inspection intervals of 30, 20, 10 and 5 years. The cases to be analyzed will include:

- Base case
- Short safe end case

While inspection is a little more complex than leak detection, it is handled in a similar way. Once again, all calculations are done assuming that nothing is detected during inspection while, at each time step, a probability of non detection is estimated. As a post-processing task, results will then be corrected in consequence, as follows:

Once an inspection is scheduled at a given timestep, the probability of non-detection is reported, which identifies the probability that nothing will change in the analysis. The probability of detection (1-pnd) will be associated with a change in event that can be :

- A perfect fix leading to no more cracks
- A weld replacement with the same quality (randomly select another future, including future with no cracks at all),

The probability of failure, as well as other output of interest (last four bullets of Section 3.2.2), will be corrected to take the inspection into account.
The credit inspection software to be used for this effort can be found on the sharepoint site at CM > Beta Model Dev > Modules > TRANSFORMERS v1.0. See the associated documentation for details on the features of this module.

6.3 Uncertainty and Sensitivity Analysis

Uncertainty and sensitivity analysis (statistical sensitivity analysis, which is different than what is presented in Section 4) are traditional techniques used when a probabilistic analysis is performed. They allow the user to analyze and summarize the uncertainty in the outputs of interest, and the influence from the uncertain input variables to these outputs. They are also a powerful verification and validation (V&V) tool, highlighting any strange behavior.

Uncertainty analysis will consist of classical statistical techniques such as CDF and CCDF representation and calculation of mean and quantiles, and is described in more detail in Section 3.

The sensitivity analysis proposed for the base case (e.g., beta model results) pilot study will focus on well-known and easy-to-understand methods, focused on detecting monotonic relationships between the inputs and outputs. It will include:

- Estimate of Partial Rank Correlation Coefficients (PRCCs) over time and display as a graph
- Estimate of Standardized Rank Regression Coefficients (SRRCs) and Coefficients of Determination (R^2) of a stepwise regression at selected times
- Scatter plots of the outputs of interest vs. the most important input parameters in terms of uncertainty
-

The parameter sensitivity analyses will be completed using the methodology and codes developed by Sandia National Laboratories for the xLPR Pilot Study. The sensitivity analysis software to be used for this effort can be found on the sharepoint site at CM > Beta Model Dev > Modules > Stepwise Regression Code. See the associated documentation for details on the features of this module.

Appendix C
xLPR Version 1.0 Base Case Input Parameters

Appendix C lists the input values for the base case in the analyses documented in this report.

List of Input Parameters and Distributions

Table C.1-1: Constant Parameters and Base Case Values

Variable name	Description	Unit	Input for models	Value for base case
SC analysis method	Flag to establish SCC analysis method	N/A	SC-FAIL	1
Mitigation_Time	User selected time at which mitigation occurs	yr	Crack initiation	60
Pipe Outer Diameter	Pipe outside diameter	m	All	0.381
Pipe Thickness	Pipe wall thickness	m	All	0.040132
Alpha	PWSCC power law constant	$(m/s)/(MPa\text{-}m^{0.5})^{1.6}$	Grower	2.01E-12
Beta	PWSCC exponent in equation (13) of program plan	N/A	Grower	1.6
Kth	Reference crack stress intensity factor	$MPa\text{-}m^{0.5}$	Grower	0
Tref	Absolute reference temperature to normalize data	K	Grower	598.15
Fx_NT, Fx_NT_c	Axial force for thermal expansion	kN	Load	3.87
Mx_NT, Mx_NT_c	Moment in x direction for thermal expansion	kN-m	load	65.3
My_NT, My_NT_c	Moment in y direction for thermal expansion	kN-m	Load	-57.54
Mz_NT, Mz_NT_c	Moment in z direction for thermal expansion	kN-m	Load	52.99
Fx_NTS, Fx_NTS_c	Axial force for thermal stratification	kN	Load	17.39
Mx_NTS, Mx_NTS_c	Moment in x direction for thermal stratification	kN-m	Load	2.51
My_NTS, My_NTS_c	Moment in y direction for thermal stratification	kN-m	Load	-80.79
Mz_NTS, Mz_NTS_c	Moment in z direction for thermal stratification	kN-m	Load	87.9
Fx_SSE, Fx_SSE_c	Axial force for SSE	kN	Load	28.02

Variable name	Description	Unit	Input for models	Value for base case
Mx_SSE, Mx_SSE_c	Moment in x direction for SSE	kN-m	Load	32.39
My_SSE, My_SSE_c	Moment in y direction for SSE	kN-m	Load	59.25
Mz_SSE, Mz_SSE,c	Moment in z direction for SSE	kN-m	Load	94.89
Fx_DW, Fx_DW_c	Axial force for deadweight	kN	Load	0.31
Mx_DW, Mx_DW_c	Moment in x direction for deadweight	kN-m	Load	1.31
My_DW, My_DW_c	Moment in y direction for deadweight	kN-m	Load	0.21
Mz_DW, Mz_DW_c	Moment in z direction for deadweight	kN-m	Load	1.02
sig4	Weld residual stress fitting parameter (4th)	MPa	Ksurf	0
Youngs Mod. For Alloy 600	Young's modulus for pipe material	MPa	Crack initiation	2.07E+05
SigYS Alloy 600	Yield Strength (MPa) for alloy 600 (NOT 82/182) (used for Imethod = 2 only)	MPa	Crack initiation	344.7
SigUTS Alloy 600	Ultimate tensile strength (MPa) for alloy 600 (used for Imethod = 2 only)	MPa	Crack initiation	689.5
H2	Concentration of hydrogen in primary water	cc/kg-STP	Grower	25
Zinc	Concentration of Zinc in primary water	cc/kg	Grower	0
QoverR	Q/R constant for crack initiation	MPa	Crack initiation	22000
SigTH	Threshold stress for initiation	MPa	Crack initiation	137.9
XN1	Exponent of Method 1 equation (used for Imethod=1 only)	N/A	Crack initiation	4
XN3	Exponent of Method 3 equation (used for Imethod=3 only) (dimensionless)	N/A	Crack initiation	4
Nunits	Number of sub-units	N/A	Crack initiation	19
MTS	Number of intervals in the operating history	N/A	Crack initiation	1
MTS_1	Duration of interval in the operating history	yr	Crack initiation	60
MTS_2	Duration of interval in the operating history	yr	Crack initiation	0
MTS_3	Duration of interval in the operating history	yr	Crack initiation	0
MTS_4	Duration of interval in the operating history	yr	Crack initiation	0

Variable name	Description	Unit	Input for models	Value for base case
MTS_5	Duration of interval in the operating history	yr	Crack initiation	0
MTS_6	Duration of interval in the operating history	yr	Crack initiation	0
MTS_7	Duration of interval in the operating history	yr	Crack initiation	0
MTS_8	Duration of interval in the operating history	yr	Crack initiation	0
MTS_9	Duration of interval in the operating history	yr	Crack initiation	0
MTS_10	Duration of interval in the operating history	yr	Crack initiation	0
Initiation Method Flag	Initiation method to use (1, 2, or 3)	N/A	Crack initiation	2
COD analysis method	0 = blended, > 0 reserved for future methods	N/A	COD	0
Elasticity_TP304	Young's modulus for TP304 material	MPa	COD	177100
elasticity_A516_Gr_70	Young's modulus for A516Gr70 material	MPa	N/A	186300
elasticity_Alloy182	Young's modulus for Alloy 182 material	MPa	N/A	203100
C1	Sample value for distribution of C1 (used for Imethod=3 only). The value of C1 (deterministic) is derived from data.	N/A	Crack initiation	0.04

Table C.1-2: Uncertain Parameters and Distributions

variable name	description	unit	input for models	*origin for base case*	value for base case
half_crack_length_init	half_crack_length_init / half_crack_length_init_random	m	Crack initiation	*Distribution Type*	*Normal*
				Mean	*3.00E-03*
				Stdev	*1.50E-04*
				Deterministic	*3.00E-03*
				Type	*Aleatory*
crack_depth_init	crack_depth_init / crack_depth_init_random	m	Crack initiation	*Distribution Type*	*Normal*
				Mean	*1.50E-03*
				Stdev	*7.50E-05*
				Deterministic	*1.50E-03*
				Type	*Aleatory*
B1	Heat-to-heat variability. Sample value for distribution of B1 (used for Imethod=2 only). For each segment, sample from the Within-Heat	Nunits+1	Crack initiation	*Distribution Type*	log$\mathcal{N}$
				Geometric mean	1.20E-09
				Second parameter (H-H) (std dev. of log N B1)	1.607

variable name	description	unit	input for models	*origin for base case*	value for base case
	distribution. The Heat-to-Heat sampled value is the median for within-Heat distribution.			*Deterministic*	1.20E-09
				Type	*Epistemic*
BWH_Stdev	Standard deviation for the normal distribution of within heat distribution BmuWH [Nunits_Max], used with imethod=2 only.	N/A	Crack initiation	*Distribution Type*	$\log\mathcal{N}$
				Geometric mean	0
				Geometric Stdev	1.7419
				Deterministic	0
				Type	*Epistemic*
A	Heat to Heat sampled value for distribution of A (used for Imethod=1 only), used with to generate distribution of AmuWH[Nunits_max]	N/A	Crack initiation	*Distribution Type*	$\log\mathcal{N}$
				Geometric mean	3.1629
				Second parameter (H-H) (std dev. of log N A)	1.1595
				Deterministic	3.1629
				Type	*Epistemic*
AWH_Stdev	Standard deviation for the normal distribution of within heat distribution AmuWH [Nunits_Max], used with imethod=1 only.	N/A	Crack initiation	*Distribution Type*	$\log\mathcal{N}$
				Geometric mean	0
				Geometric Stdev	2.915
				Deterministic	0
				Type	*Epistemic*
random_placement (RandULoc)	Placement of a new crack (cannot be changed by the user)	N/A	Crack initiation	*Distribution Type*	$\mathcal{U}[0,1]$
				Min	*0*
				Max	*1*
				Type	*Epistemic*
random_number for time (RandU3)	Time of crack initiation (Cannot be changed by the user)	N/A	Crack initiation	*Distribution Type*	$\mathcal{U}[0,1]$
				Min	*0*
				Max	*1*
				Type	*Epistemic*
sig0_wrs	Axial stress component for wrs	MPa	Load	*Distribution Type*	$\mathcal{N}ormal$
				mean	300.3
				Stdev	110
				Min	150
				Max	551
				Deterministic	150
				Type	*Epistemic*
sig0_wrs_mitigated	Axial stress component for wrs - mitigated	MPa	Load	*Distribution Type*	$\mathcal{N}ormal$
				mean	-344.75
				Stdev	34
				Min	-447

variable name	description	unit	input for models	*origin for base case*	value for base case
				Max	-242
				Deterministic	-150
				Type	*Epistemic*
Xc	Location in the pipe where weld residual stress equal 0	N/A	Load	*Distribution Type*	Normal
				mean	0.25
				Stdev	0.05
				Min	0.125
				Max	0.5
				Deterministic	0.143
				Type	*Epistemic*
Xc_Mitigated	Location in the pipe where weld residual stress equal 0 - mitigated	N/A	Load	*Distribution Type*	Normal
				mean	0.38
				Stdev	0.038
				Min	0.26
				Max	0.5
				Deterministic	0.38
				Type	*Epistemic*
OD_stress_random	WRS in the outer diameter	N/A	Load	*Distribution Type*	Uniform
				min	0.5
				max	1
				Deterministic	0.5
				Type	*Epistemic*
f_Weld	Weld factor : common factor applied to all specimens fabricated from the same weld to account for weld wire/stick heat processing and for weld fabrication	N/A	Grower	*Distribution Type*	log$\mathcal{N}$
				Geometric mean	0.99894
				Geometric Stdev	1.83475
				Min	0
				Max	2.71
				Deterministic	1.074897
				Type	*Epistemic*
QoverR	ratio of thermal activation energy for PWSCC crack growth over universal gas constant	K	Grower	*Distribution Type*	Normal
				Mean	*15636*
				Stdev	*601*
				Deterministic	*15636*
				Type	*Aleatory*
P	peak-to-valley ratio	N/A	Grower	*Distribution Type*	Normal
				Mean	*9.5*
				Stdev	*1.36*
				Deterministic	*9.5*

variable name	description	unit	input for models	*origin for base case*	value for base case
				Type	*Aleatory*
c	characteristic width of crack growth rate curve	mV	Grower	*Distribution Type*	$\mathcal{Normal}$
				Mean	*22.5*
				Stdev	*3.21*
				Deterministic	*22.5*
				Type	*Aleatory*
POD_detection	random number determining whether a crack is detected or not (not used in alpha version but needed for ISI module)	N/A	ISI	*Distribution Type*	$\mathcal{Uniform}$
				Min	*0*
				Max	*1*
				Deterministic	*0.5*
				Type	*Aleatory*
POD_beta1	parameter □1 for probability of detection of a Surface Crack	N/A	ISI	*Distribution Type*	$\mathcal{Normal}$
				mean	2.7076
				Stdev	0.2085
				Deterministic	2.7076
				Type	*Epistemic*
POD_beta2	parameter □2 for probability of detection of a Surface Crack	N/A	ISI	*Distribution Type*	$\mathcal{Normal}$
				mean	0.0031
				Stdev	0.0045
				Correlated to POD_beta1	0.86
				Deterministic	0.0031
				Type	*Epistemic*
Pressure	pressure in the pipe	MPa	SQUIRT, TWCFail, SCFail	*Distribution Type*	$\mathcal{Normal}$
				mean	15.5132
				Stdev	0.1551
				Deterministic	15.5132
				Type	*Epistemic*
Temperature	temperature in the pipe	C	SQUIRT, Grower	*Distribution Type*	$\mathcal{Normal}$
				mean	344.9
				Stdev	0.0882
				Deterministic	345
				Type	*Epistemic*
sigy_TP304	yield stress of TP 304 pipe material	MPa	TWC-FAIL	*Distribution Type*	$\log\mathcal{N}$
				Geometric mean	*168.763*
				Geometric Stdev	1.232792
				Deterministic	*168.763*
				Type	*Aleatory*
sigu_TP304	ultimate stress of TP304 pipe material	MPa	TWC-FAIL	*Distribution Type*	$\log\mathcal{N}$
				Geometric mean	*450.6127*

variable name	description	unit	input for models	*origin for base case*	value for base case
				Geometric Stdev	*1.123960*
				Correlate sigy_ TP304	0.6066
				Deterministic	*450.6127*
				Type	*Aleatory*
F_TP304, RO_F	TP304 Ramberg-Osgood fit parameter, F	MPa	TWC-FAIL	*Distribution Type*	$\log\mathcal{N}$
				Geometric mean	*562.1217*
				Geometric Stdev	*1.080277*
				Deterministic	*562.1217*
				Type	*Aleatory*
n_TP304, RO_n	TP304 Ramberg-Osgood fit parameter, n	N/A	TWC-FAIL	*Distribution Type*	$\log\mathcal{N}$
				Geometric mean	*4.260565*
				Geometric Stdev	*1.141419*
				Correlate F_TP304	-0.6047
				Deterministic	*4.260565*
				Type	*Aleatory*
Jic_Alloy_182, Resist_Jic	Alloy 182 material initiation J-resistance	N/mm	TWC-FAIL	*Distribution Type*	$\log\mathcal{N}$
				Mean	*482.7*
				Stdev	*1.783853*
				Deterministic	*482.7*
				Type	*Aleatory*
C_Alloy_182, Resist_C	Alloy 182 material initiation J-resistance coefficient	N/mm$^{(exponent+1)}$	TWC-FAIL	*Distribution Type*	$\log\mathcal{N}$
				Mean	*260.1*
				Stdev	*1.621629*
				Correlate Jic_Alloy_182	0.9
				Deterministic	*260.1*
				Type	*Aleatory*
m_Alloy_182, Resist_m	Alloy 182 material initiation J-resistance exponent	N/A	TWC-FAIL	*Distribution Type*	$\log\mathcal{N}$
				Mean	*0.612089*
				Stdev	*1.173811*
				Deterministic	*0.612089*
				Type	*Aleatory*
sigy_Alloy_182	Alloy 182 yield strength	MPa	SQUIRT, TWCFail, SCFail	*Distribution Type*	$\log\mathcal{N}$
				Geometric mean	361.5464
				Geometric Stdev	1.269675
				Deterministic	361.5464
				Type	*Aleatory*
sigu_Alloy_182	Alloy 182 ultimate strength	MPa	SQUIRT, TWCFail,	*Distribution Type*	$\log\mathcal{N}$
				Geometric mean	580.1362

variable name	description	unit	input for models	*origin for base case*	value for base case
			SCFail	*Geometric Stdev*	1.104332
				Correlated to sigy_Alloy_182	0.5
				Deterministic	580.1362
				Type	*Aleatory*
sigy_A516_Gr_70	A516Gr70 yield strength	MPa	SQUIRT, TWCFail, SCFail	*Distribution Type*	$\log\mathcal{N}$
				Geometric mean	227.4765
				Geometric Stdev	1.099388
				Deterministic	227.4765
				Type	*Aleatory*
sigu_A516_Gr_70	A516Gr70 ultimate strength	MPa	SQUIRT, TWCFail, SCFail	*Distribution Type*	$\log\mathcal{N}$
				Geometric mean	519.1096
				Geometric Stdev	1.056711
				Correlated to sigy_A516_Gr_70	0.4866
				Deterministic	519.1096
				Type	*Aleatory*
F_A516_Gr_70	A516Gr70 Ramberg-Osgood Fit parameter	MPa	SQUIRT, TWCFail, SCFail	*Distribution Type*	$\log\mathcal{N}$
				Geometric mean	911.5219
				Geometric Stdev	1.093895
				Deterministic	911.5219
				Type	*Aleatory*
n_A516_Gr_70	A516Gr70 Ramberg-Osgood Fit parameter	N/A	SQUIRT, TWCFail, SCFail	*Distribution Type*	$\log\mathcal{N}$
				Geometric mean	4.288899
				Geometric Stdev	1.132017
				Correlated to F_A516_Gr_70	-0.8565
				Deterministic	4.288899
				Type	*Aleatory*

Appendix D

Impact Assessment for xLPR Model Version 1.0

During post-model development activities following completion of the xLPR Version 1.0 report, several issues were identified by the program team related to errors in the framework implementation, module source codes, and input parameter values. This appendix includes an evaluation of the sensitivity of the Version 1.0 model results with regard to the outstanding issues. Section 1.0 includes a summary table (Table 1) which lists the issues that have been identified and logged into the Version 1.0 issue tracking log on the xLPR SharePoint Site (https://websps1.battelle.org/nrcnureg/home/xLPR_CM). Section 2.0 documents the results of several impact analyses that were conducted using the GoldSim framework model to assess the impact of the issues on the results presented in this Version 1.0 report.

1 Summary of Issues in the Issue Tracking Log for Version 1.0 of the xLPR Model

Table 1.0 includes a summary of each issue, the model component affected, and the impact on the xLPR Version 1.0 model results. The GoldSim model framework was used to assess the effects with both qualitative assessments and quantative impact assessments (documented in Section 2.0).

Table 1: Summary of Issues in Version 1.0 of the xLPR Model

Issue No.	Model Component	Issue Description	Impact on Version 1.0 Results
I-1	Coalesce Module	**Number of cracks per time step:** If the coalesce module is sent the current number of cracks, it assumes to start with crack 1, crack 2, etc. If there are two non-sequential cracks (i.e., crack 1 and 3), the module uses crack 1 and 2 data to check for coalescence. Passing the total number of cracks that have initiated every time step ensures the module goes through the entire array in the correct order. Logic was added to the framework to set the number of cracks to 1 when the current number of cracks is 1, otherwise use the total number of cracks initiated as before.	**None.** For multiple cracks the framework passes the total number of cracks that have initiated to ensure the module goes through the entire array in the correct order. [number of cracks =1 if the current number of cracks is 1, else, total number of cracks initiated. Both frameworks have this logic implemented]]

Issue No.	Model Component	Issue Description	Impact on Version 1.0 Results
I-2	TWCfail SCFail Modules	**Use of flow stress in stability:** For surface and through-wall cracks in dissimilar metal welds, the location of the crack in the weld dictates the material properties to use to make accurate stability predictions. The stainless steel flow stress is used which may be conservative. The models group may want to consider other options such as in PVP2008-61110.	**Unknown**. Assumed to be conservative.
I-3	SCFail Module	**EPFM for surface cracks:** [SCFail] Net-section collapse is assumed for surface cracks. However, for some welds, elastic plastic fracture may control surface crack failure. An EPFM scheme for surface cracks is needed.	**Unknown.** Assumed to be conservative.
I-4	Grower Module	**Crack can grow longer than circumference**: There is no check in Grower module which limits the half crack length to be <= to the 1/2 circumference. Although there is a logical check in coalesce to make sure this doesn't happen, coalesce is only used when you have two or more cracks. For one crack, the half-length is not constrained.	**None.** A logical check was added to both frameworks which limits the half crack length to 1/2 the circumference.
I-5	Grower Module	**Error in PWSCC Rate:** [v2.0/2.1] In regard to the following line of code in the PWSCCRate subroutine: DECP = 29.58*((Temp + TZ)/298.15) * Log(H2/H2NiNiO) the log term should be the "Log10" Fortran function (not the natural log function "Log"), per Equation 2-4 of MRP-213 (same as Equation 7-2 of MRP-263): DECP = 29.58*((Temp + TZ)/298.15) * Log10(H2/H2NiNiO) This affects the Grower module for the case that the MRP-263 CGR model including the effect of hydrogen is used. It does not affect the growth time calculation when the hydrogen concentration is set to zero, and the MRP-115 CGR model is used.	**Small.** The impact is most pronounced in the first 30 years. Additionally the effects of increasing hydrogen concentrations are not as large. GSxLPRv1.02_M02_008 GSxLPRv1.02_M02_009 GSxLPRv1.02_M02_010

Issue No.	Model Component	Issue Description	Impact on Version 1.0 Results
I-6	Initiation Module	**A and B1 Input Errors:** Input errors were found for the parameters B, A, and Awh_Stdev, affecting both Direct Method I and Direct Method II. The base case uses Direct Method II. For Direct Method I, the mean value of A was incorrect. When this was discovered, it was also noticed the original input values of the mean (0.141) and standard deviation (1.16) for A and Awh_Stdev (1.07) were not calibrated correctly. Those values were updated on 12/3/2010 by SIA, so the spreadsheet input for Direct Method I was corrected for the mean (0.2), and standard deviation (in the spreadsheet exp(1.16) or 3.189933) of A, and Awh_stdev(in the spreadsheet exp(0.4) or 1.491824). For Direct Method II, the standard deviation value of B (1.607) was incorrect, but was also corrected (exp(1.607) or 4.9878).	**Small.** Impact analysis has been run and results are documented in Section 2.0. This error will be corrected in the next version of the code. GSxLPRv1.02_M02_005 GSxLPRv1.02_M02_CIMethod_I_002
I-7	Inspection Module	**Incorrect mean and standard deviation for Beta2:** A mean of 0.0031 and standard deviation of 0.0045 were used in the Version 1.0 model base case input deck. The correct values should have been: 0.31 and 0.45 for the mean and standard deviation respectively.	**Negligible.** An Impact Analysis has been completed and results are documented in Section 2.0. GSxLPRv1.02_M02_007
I-8	ISI Module	**Incorrect relation between beta1 and beta2:** In the conditional distribution, the standard deviation is incorrectly formulated, using sigma1 instead of sigma2.	**Negligible.** An Impact Analysis has been completed and results are documented in Section 2.0. GSxLPRv1.02_M02_006
I-9	kSurf Module	**Inaccuracies in surface crack curve fits:** It has been demonstrated (PVP2008-61205) that the curve fit solutions used are slightly inaccurate compared to the lookup table version. The curve fits may need to be revisited to increase accuracy.	**Negligible.** The solution accuracy is believed to be sufficient for the pilot study problem.
I-10	kTWC Module	**Linear stress distribution for through wall cracks (TWC):** In the K-solutions for TWC, only a linear stress distribution through wall is allowed. Others have developed higher order stress distribution K-solutions for TWC that may be more accurate.	**Negligible.** The solution accuracy is believed to be sufficient for the pilot study problem.

2 Impact Analyses

2.1 Issue I-5—Error in PWSCC rate function in the Grower Module v2.0 and v2.1

In Versions 2.0 and 2.1 of the Grower module source code, specifically with regard to the following line of code in the PWSCCRate subroutine,

```
DECP = 29.58*((Temp + TZ)/298.15) * Log(H2/H2NiNiO)
```

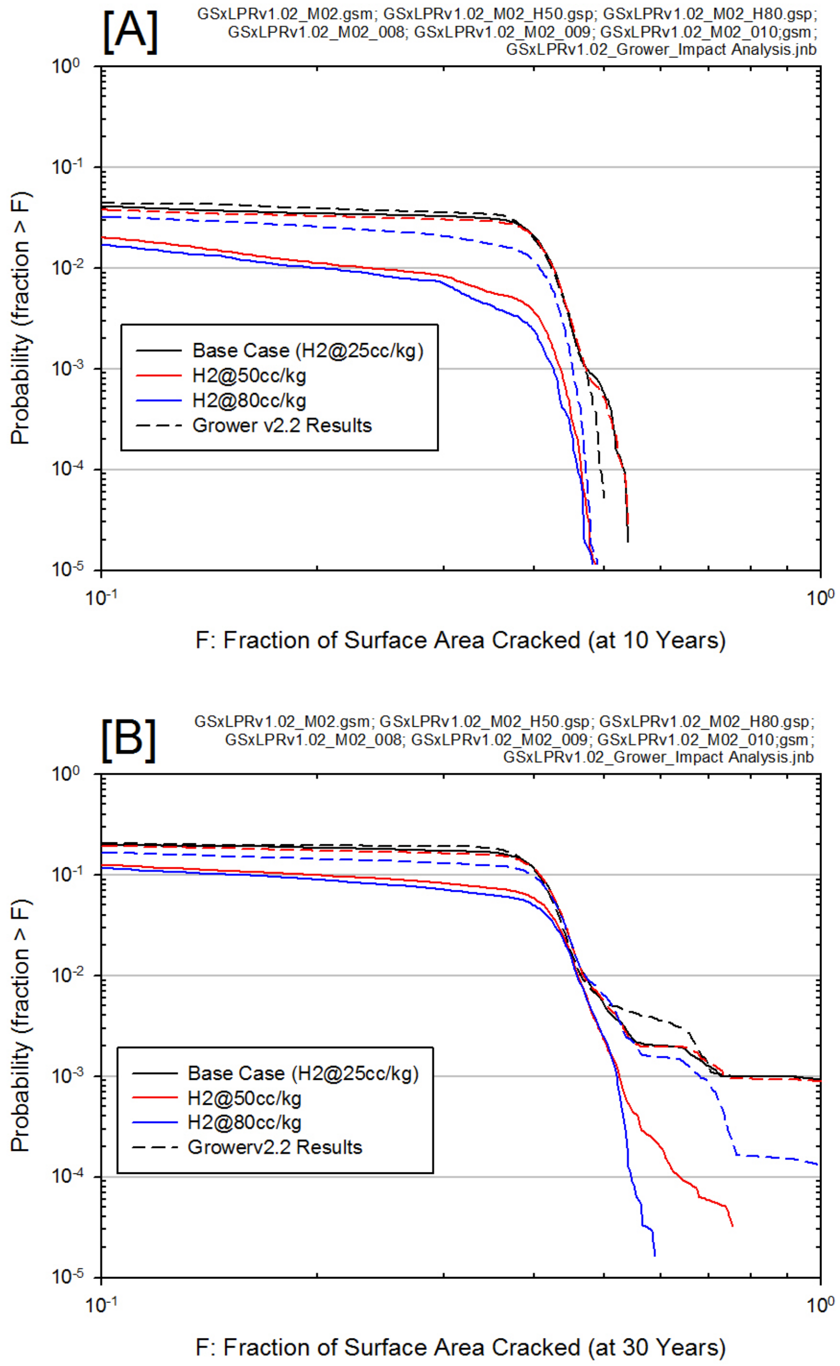
the log term should be the "Log10" Fortran function (not the natural log function "Log"), per Equation 2-4 of MRP-213 (same as Equation 7-2 of MRP-263):

```
DECP = 29.58*((Temp + TZ)/298.15) * Log10(H2/H2NiNiO)
```

This affects the Grower module for the case that the MRP-263 CGR model including the effect of hydrogen is used. It does not affect the growth time calculation when the hydrogen concentration is set to zero, and the MRP-115 CGR model is used.

2.1.1 Issue I-5 Impact Assessment

In order to estimate the impact on xLPR results, sensitivity runs have been performed using the GoldSim xLPR framework. The Grower module was recompiled as Version 2.2 with the source code corrected. The base case was re-run, as well as the two chemical sensitivity analyses with hydrogen at 50 and 80 cc/kg. The results of the impact analysis are displayed in Figures 1 through 3. The corrected Grower module (v2.2) yields a higher probability of larger crack sizes for early times. This is most pronounced before 30 years. This effect is seen in all of the fractional surface area plots, as well as, in the probability of rupture, probability of first leak, and maximum total leakage rates. This change illustrates that the addition of hydrogen has a much lower impact on the probabilities of leakage and rupture than the case with the error. These results show that the Grower model error resulted in crack growth rates that were under predicted. This change impacts Figure 32 in the main body of this report.


[A]
GSxLPRv1.02_M02.gsm; GSxLPRv1.02_M02_H50.gsp; GSxLPRv1.02_M02_H80.gsp;
GSxLPRv1.02_M02_008; GSxLPRv1.02_M02_009; GSxLPRv1.02_M02_010;gsm;
GSxLPRv1.02_Grower_Impact Analysis.jnb
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
Probability (fraction > F)
Base Case (H2@25cc/kg)
H2@50cc/kg
H2@80cc/kg
Grower v2.2 Results
10^{-1}
10^0
F: Fraction of Surface Area Cracked (at 10 Years)
[B]
GSxLPRv1.02_M02.gsm; GSxLPRv1.02_M02_H50.gsp; GSxLPRv1.02_M02_H80.gsp;
GSxLPRv1.02_M02_008; GSxLPRv1.02_M02_009; GSxLPRv1.02_M02_010;gsm;
GSxLPRv1.02_Grower_Impact Analysis.jnb
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
Probability (fraction > F)
Base Case (H2@25cc/kg)
H2@50cc/kg
H2@80cc/kg
Growerv2.2 Results
10^{-1}
10^0
F: Fraction of Surface Area Cracked (at 30 Years)

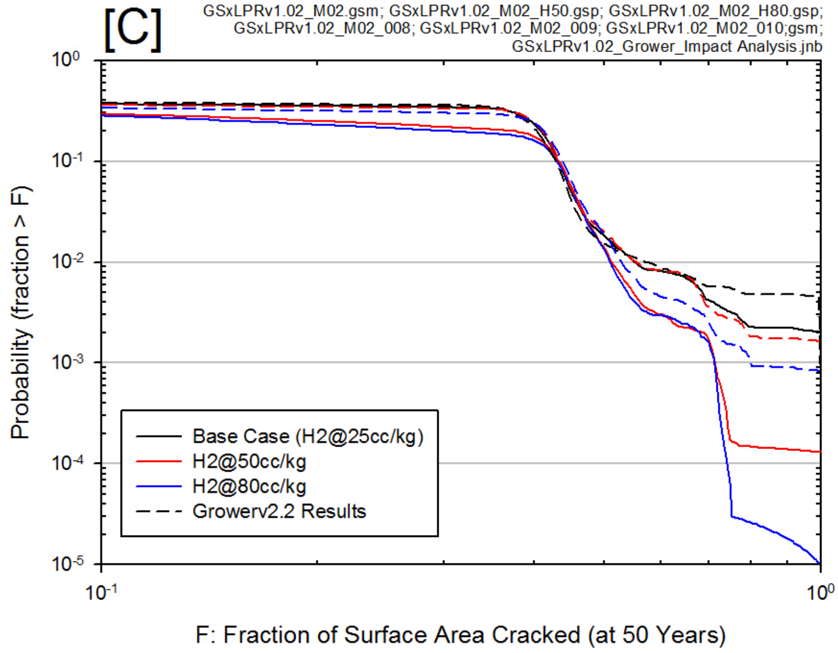


Figure 1 **CCDF's of fraction of surface area cracked at 10, 30, and 50 years for impact analyses (dashed lines) versus the base case results (solid lines).**

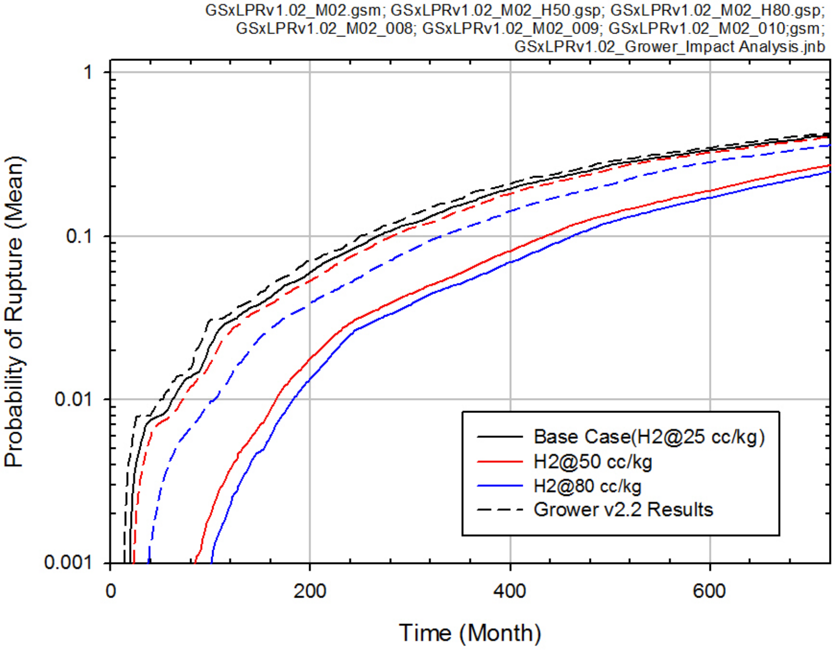


Figure 2 **Probability of rupture for impact analyses (dashed lines) versus the base case results (solid lines).**

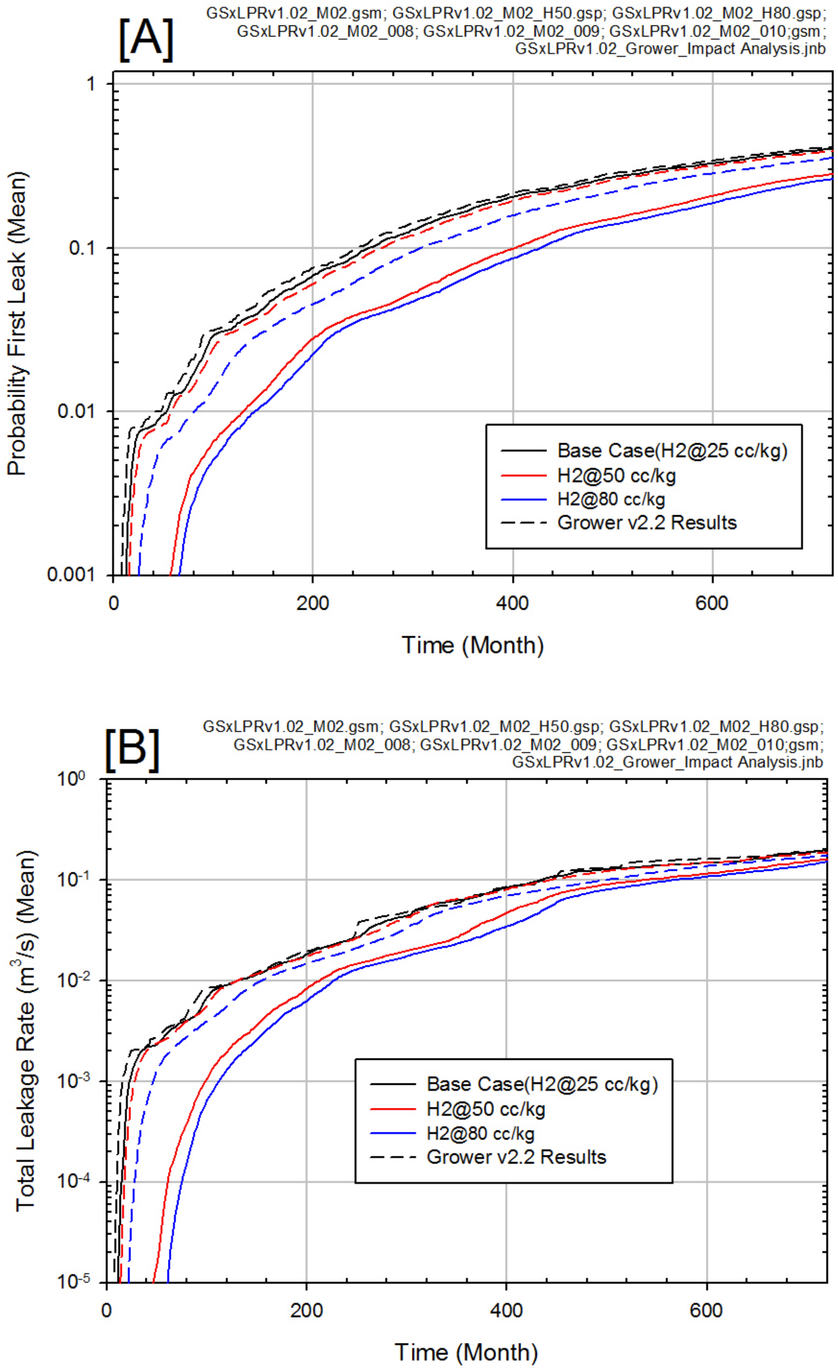


Figure 3 **[A] Probability of first leak and [B] total leakage rate for impact analyses (dashed lines) versus the base case results (solid lines).**

2.2 Issue I-6—Crack Initiation Module Parameter Input Errors

Input errors were found for the parameters B, A, and Awh_Stdev, affecting both the Direct Method I and Direct Method II. The base case uses Direct Method II. For Direct Method I, the mean value of A was incorrect. When this was discovered it was also noticed that the original input of the mean (0.141) and standard deviation (1.16) for A and Awh_Stdev (1.07) were not correct. Those values were updated on 12/3/2010 by SIA, so the input for Direct Method I was corrected for the mean (0.2) and standard deviation (exp (1.16) or 3.189933) of A and the value of Awh_stdev (exp (0.4) or 1.491824). For Direct Method II, the standard deviation (1.607) of B was incorrect, but was corrected (4.9878 or exp (1.607)).

2.2.1 Issue I-6 Impact Assessment

In order to estimate the impact on xLPR results, sensitivity runs have been performed using the GoldSim xLPR framework. Results are displayed and commented below.

Direct Method I. The comparison between the probability of first crack using the original Crack Initiation (CI) distribution and the corrected values using Direct Method I is presented in Figure 4. With the new distribution, the probability that at least one crack has occurred at 20 years (i.e. month = 240) is 0.3110 compared to 0.2430 previously. The resulting annual frequency is then 0.019 cracks per year, compared to 0.014 initially. Both methods give a higher annual frequency than expected (0.01/yr). At 60 years, the probability reaches 0.587 (compared to 0.621), which results in an annual frequency of 0.015 cracks per year, compared to 0.016 initially. Figure 5 displays the comparison between original and corrected crack initiation distribution using Direct Method I for the mean probability of rupture.

As for Direct Method II, the evolution through time has a concave behavior with the corrected distribution, while the original distribution was closer to a linear behavior. In Figure 6, probabilities of rupture are compared with leak rate detection (set to 10 gpm). As the corrected distribution leads to cracks earlier in time, the probability of rupture is higher earlier in life as compared to the original analyses. Since the cracks occur early, it is more likely to have a rupture earlier, despite leak rate detection. The original distribution leads to a smaller probability that will persist until the end of the simulation as late cracks have less time to evolve and cause rupture. The conclusion is, therefore, similar to the one reached when comparing results for Direct Method I.

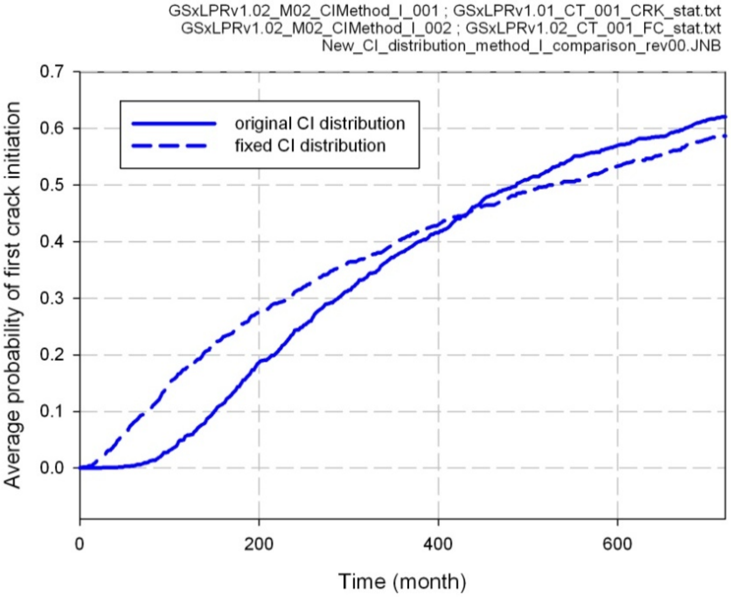


Figure 4 Average probability of first crack initiation for original crack initiation distribution (solid line) and fixed crack initiation distribution (dashed line)

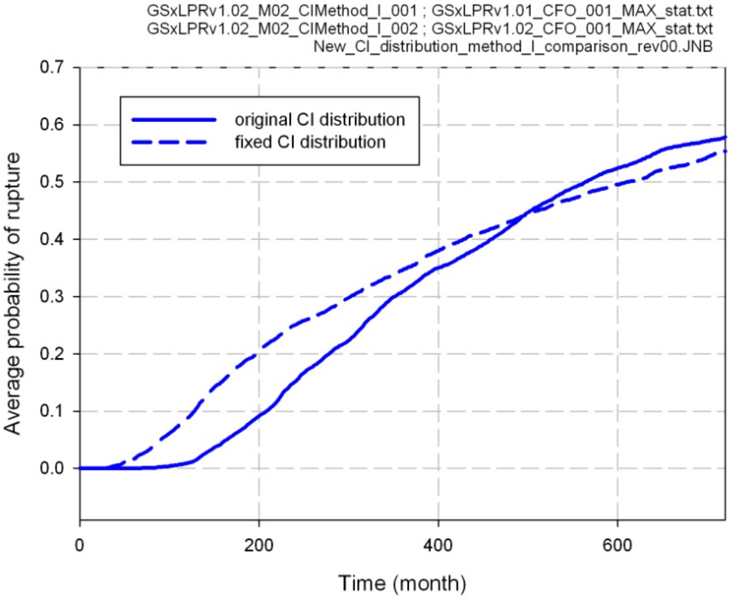


Figure 5 Average probability of rupture for original crack initiation distribution (solid line) and fixed crack initiation distribution (dashed line)

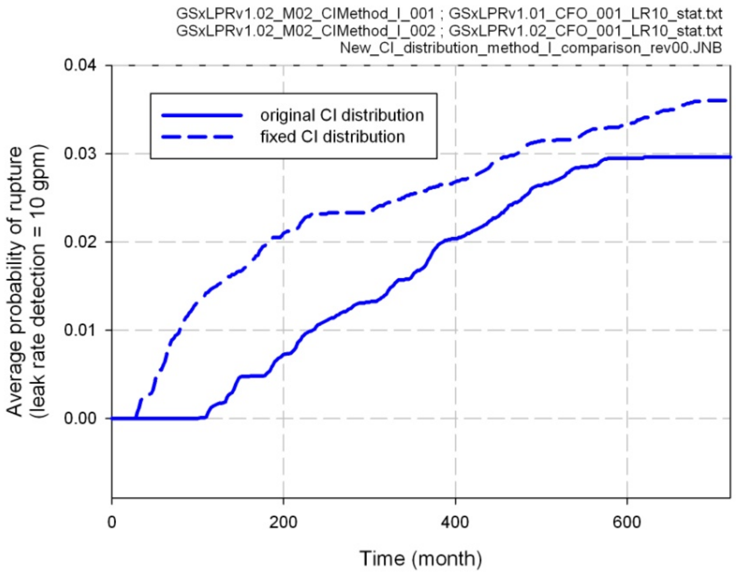


Figure 6 Average probability of rupture considering leak rate detection of 10 gpm for original crack initiation distribution (solid line) and fixed crack initiation distribution (dashed line)

Direct Method II (Version 1.0 base case results). The comparison between the probability of first crack using the original Crack Initiation (CI) distribution and corrected values using Direct Method II is presented in Figure 7. With the new distribution, the probability that at least one crack has occurred at 20 years (i.e. month = 240) is twice as large with the new distribution parameters (0.264 compared to 0.135 initially). The resulting annual frequency is then 0.015 cracks per year, compared to 0.007 initially. At 60 years, the probability reaches 0.5150 (compared to 0.4710), which results in an annual frequency of 0.012 cracks per year, compared to 0.011 initially. The evolution through time is concave, while the original distribution was slightly convex, but closer to a linear behavior. Figure 8 presents a comparison similar to Figure 1 for the mean probability of rupture. As expected, the same differences can be noticed in this figure, leading to the same conclusions.

In Figure 9, probabilities of rupture are compared once again, after leak rate detection (set to 10 gpm) has been taken into account. In the original distribution, only 2% of the epistemic realizations were leading to rupture when including leak rate detection. It represents about 20 epistemic realizations for an epistemic sample of size 1,000. As the cracks tend to occur early with the corrected crack initiation distribution parameters, it is more likely to generate rupture without leak rate detection at an early time. As a result, the probability presented in Figure 9 changes more drastically when the new distribution parameters are used.

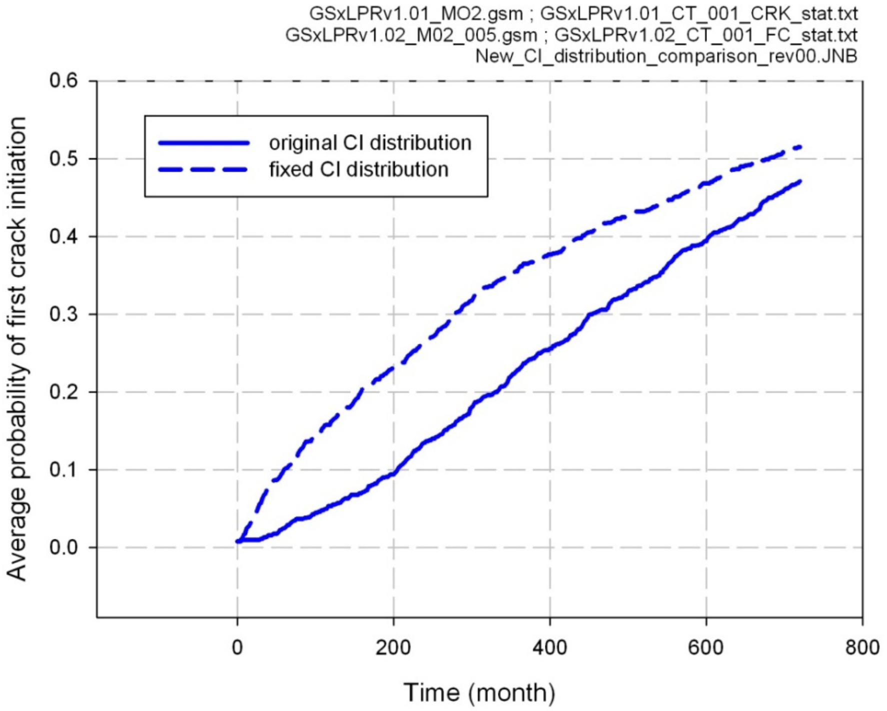


Figure 7 Average probability of first crack initiation for original crack initiation distribution (solid line) and fixed crack initation distribution (dashed line)

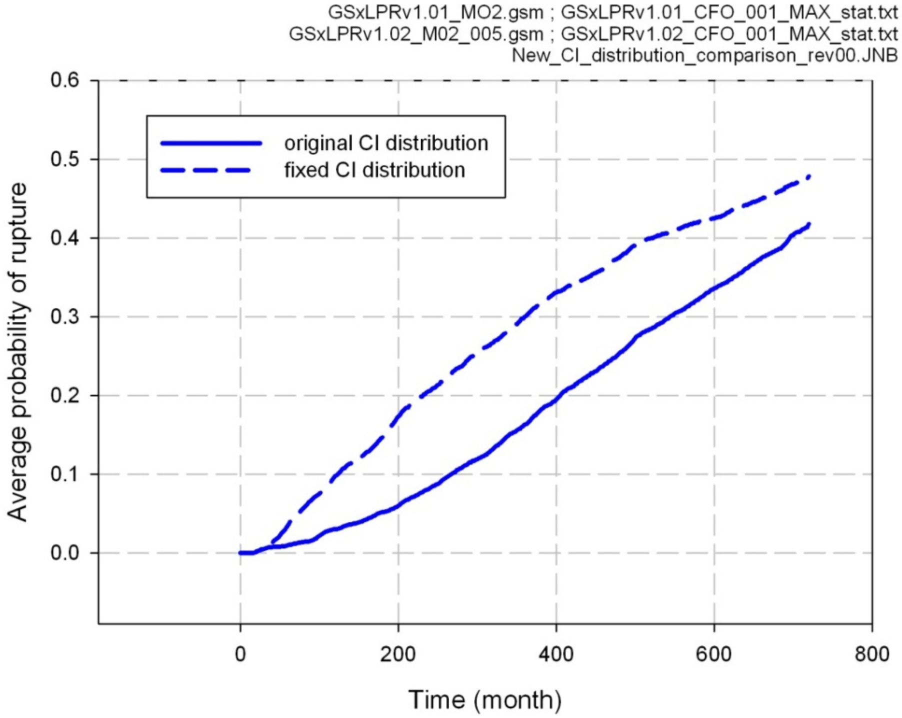


Figure 8 Average probability of rupture for original crack initiation distribution (solid line) and fixed crack initiation distribution (dashed line)

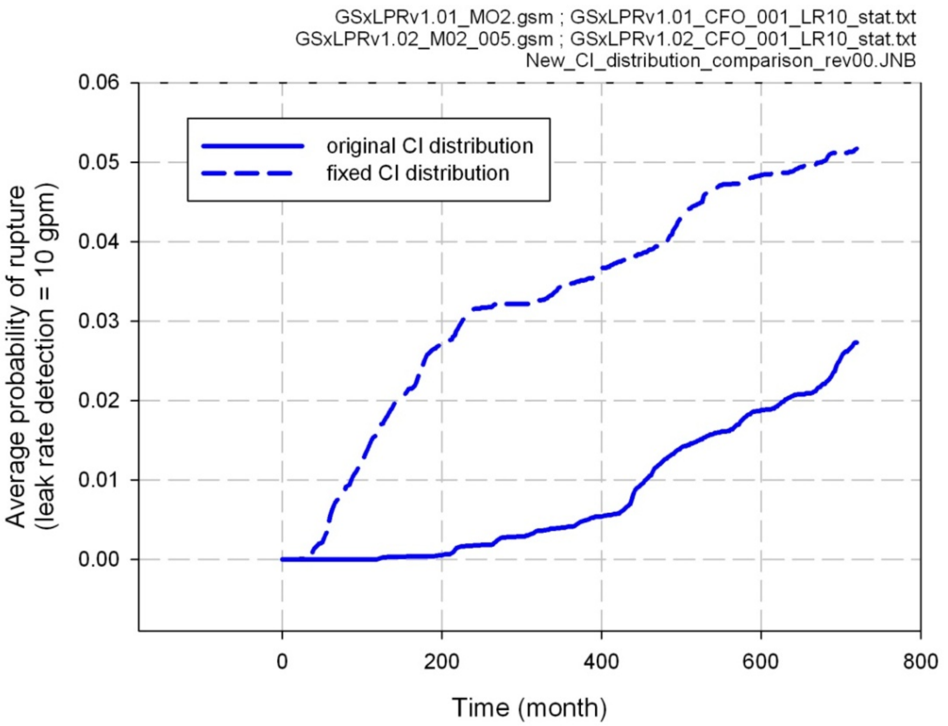


Figure 9 **Average probability of rupture considering leak rate detection of 10 gpm for original crack initiation distribution (solid line) and fixed crack initiation distribution (dashed line)**

2.3 Issue I-7—Inspection Model Parameter Input Error for β_2

For the parameter $\beta_{2,}$ a mean of 0.0031 and standard deviation of 0.0045 were used in Version 1.0 of the base case model input deck for the Inspection (ISI) Module. The correct values should have been: 0.31 and 0.45 for the mean and standard deviation respectively.

2.3.1 Issue I-7 Impact Assessment

In order to estimate the impact on xLPR results, sensitivity runs have been performed using the GoldSim xLPR framework. Figure 10 presents a scatterplot comparison on each of the 1,000 values estimated at 60 years for the probability of non-detection. A perfect match between the original relation (x-axis) and the fixed relation (y-axis) results will lead to a plot perfectly on the isoline (dark line). While one can see some deviation from the isoline, the most important deviations are associated with a low probability of non-detection, which will account for only a small portion of the average estimate.

This conclusion is supported by Figure 11. This figure compares mean results for the probability of non-detection of a crack for the first occurring crack (beta1 and beta2 are used to estimate this probability). The curves are identical (results match up to the second digit when the mean is estimated), indicating that the change does not affect the result.

Figure 12 extends the comparison to probability of rupture conditional on inspection every 10 years (in order to include the effect of probability of non-detection for all existing cracks). Once again, identical results indicate the change has no effect on the results or conclusions.

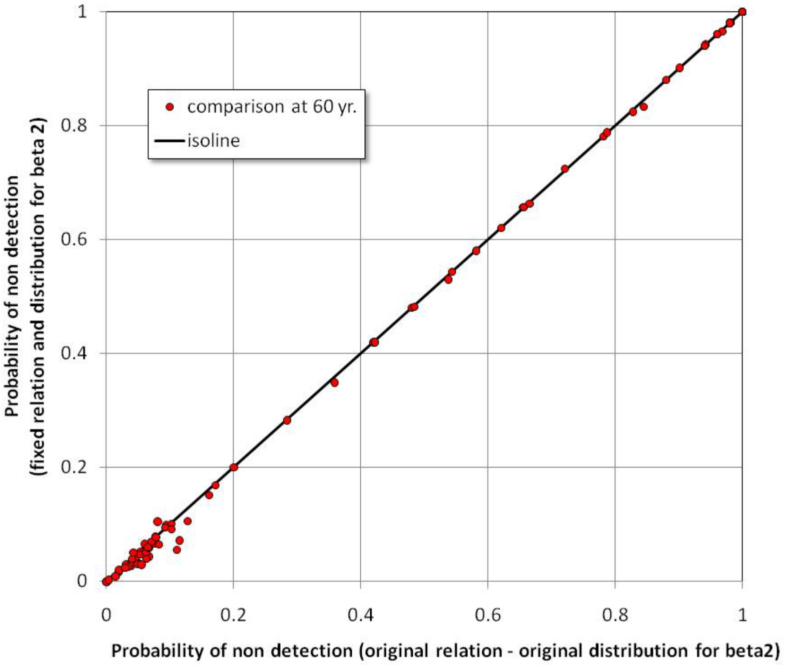


Figure 10 Scatterplot comparisons in probability of non-detection for the first crack at 60 years between original relation (x-axis) and fixed relation (y-axis)

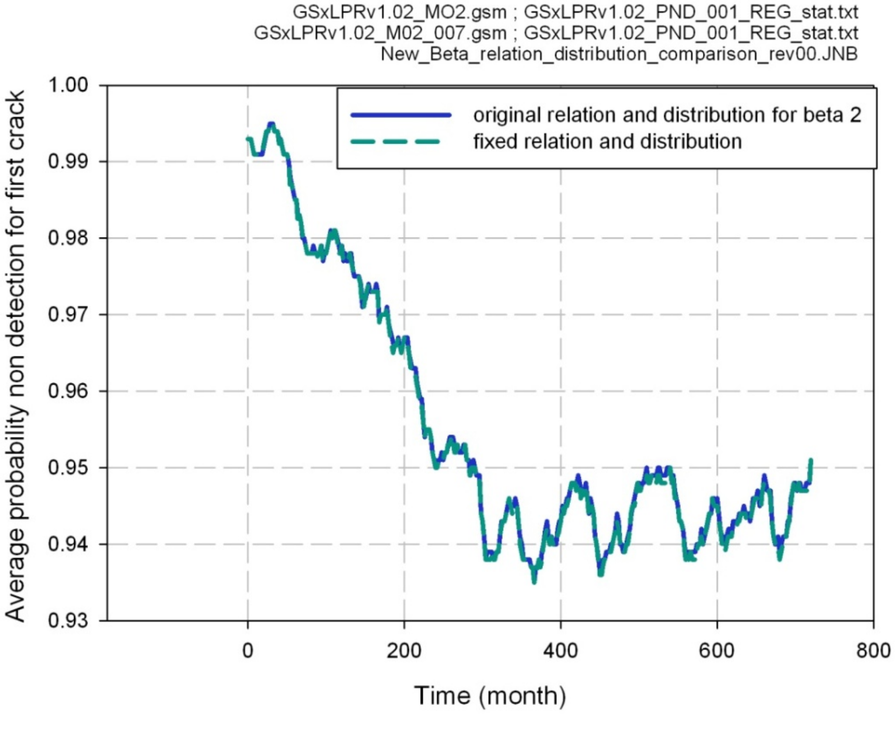


Figure 11 Comparison of average probability of non-detection for first crack between original relation and fixed relation

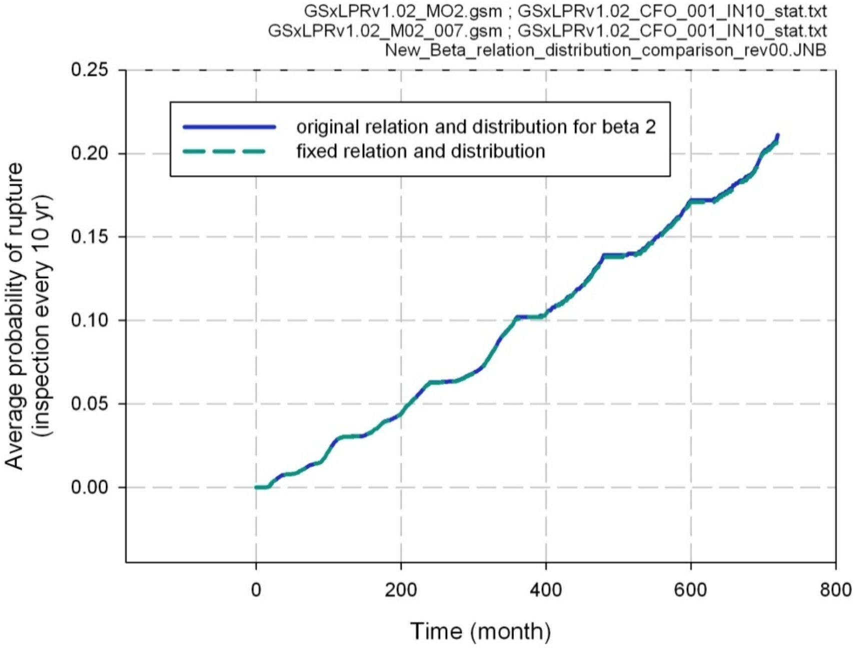


Figure 12 Comparison of average probability of rupture

2.4 Issue I-8—Incorrect relation between β1 and β2

An error was found in the formula used for the conditional normal distribution used to represent the bivariate normal distribution between parameters β_1 and β_2.

The formula used was:

$$X_2|X_1 \sim N\left(\mu_2 + \frac{\sigma_2}{\sigma_1}\rho(x_1 - \mu_1), \sqrt{1-\rho^2}\sigma_1\right)$$

While the correct formula is:

$$X_2|X_1 \sim N\left(\mu_2 + \frac{\sigma_2}{\sigma_1}\rho(x_1 - \mu_1), \sqrt{1-\rho^2}\sigma_2\right)$$

Where μ_1=2.7076 and σ_1=0.2085 represent the parameters (resp. mean and standard deviation) for β_1 while μ_2=0.0031 and σ_2=0.0045 represent the parameters for β_2.

2.4.1 Issue I-8 Impact Assessment

In order to estimate the impact on xLPR results, a sensitivity run has been performed using the GoldSim xLPR framework. Results are displayed and commented below.

Figure 13 presents a scatterplot comparison on each of the 1,000 PND values estimated at 60 years. A perfect match between the original relation (x-axis) and the corrected relation (y-axis) results will lead to a plot perfectly on the isoline (dark line). As it can be seen, while some of the results deviate from the isoline, the variation is very small and can be considered negligible.
This conclusion is supported by Figure 14. This figure compares the mean results for probability of non-detection of a crack for the first occurring crack (beta1 and beta2 are used to estimate

this probability). The curves are identical (results match up to the third digit when the average is estimated), indicating that the change does not affect the result.

Figure 15 extends the comparison to probability of rupture conditional on inspection every 10 years (in order to include the effect of probability of non-detection for all existing cracks). Once again, identical results indicate the change has no effect on the results or conclusions.

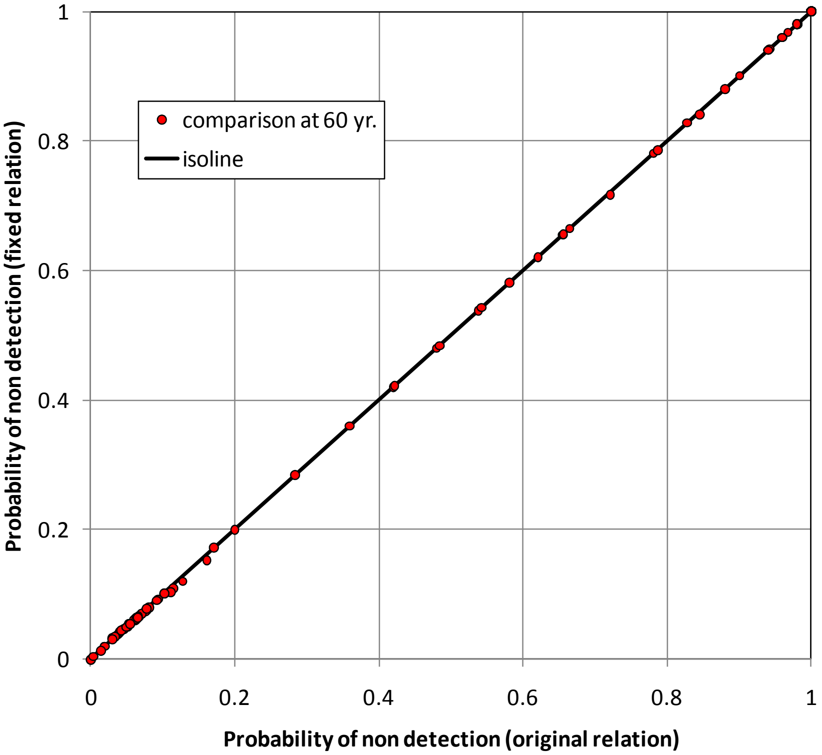


Figure 13 Scatterplot comparisons in probability of non-detection for the first crack at 60 years between original relation (x-axis) and fixed relation (y-axis)

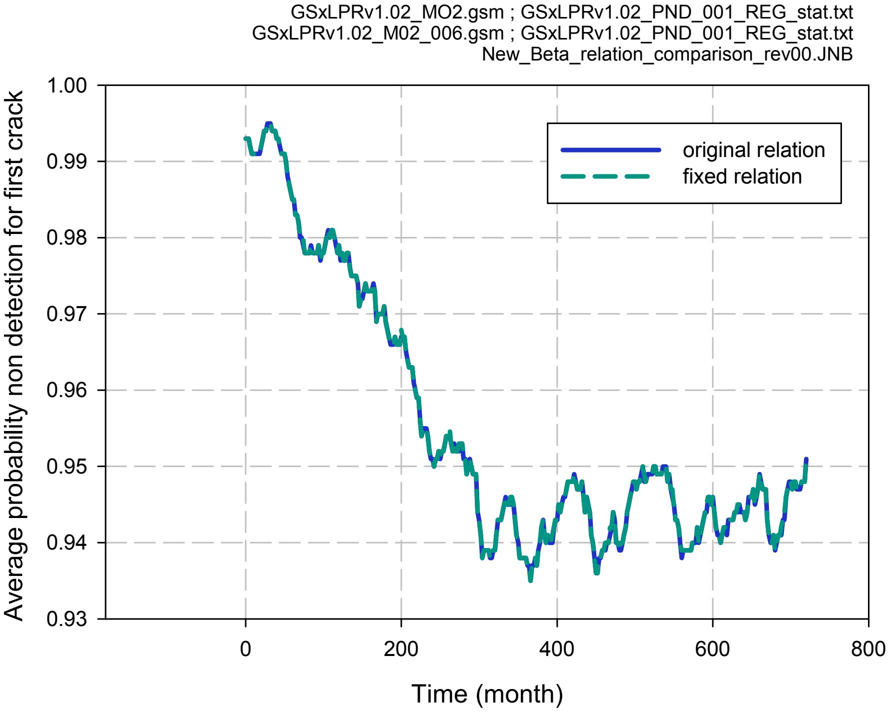


Figure 14 Comparison of average probability of non-detection for first crack

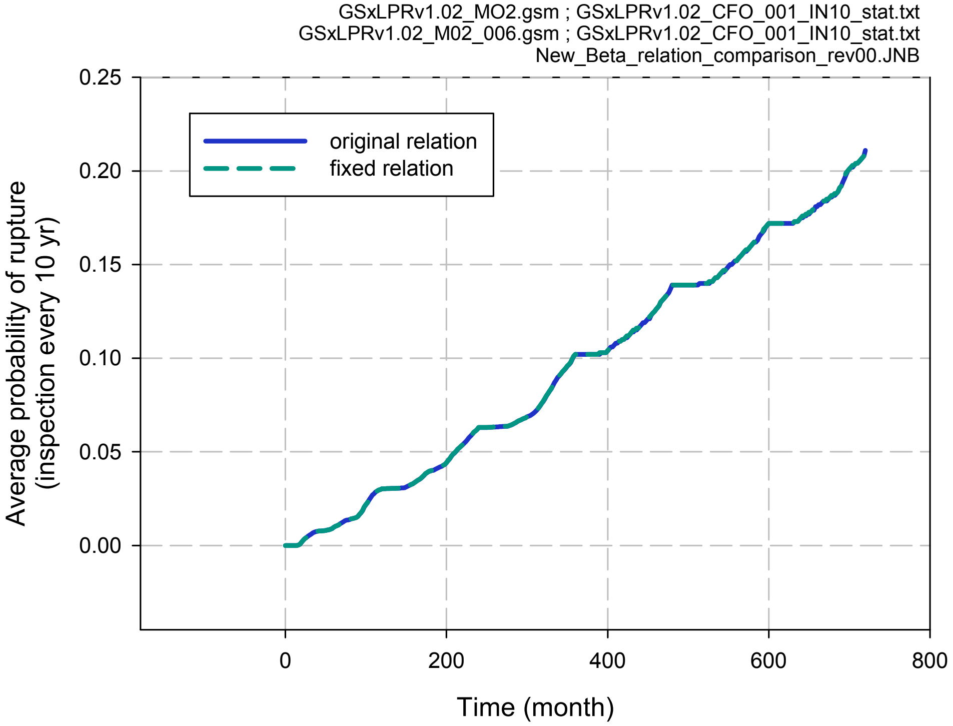


Figure 15 Comparison of average probability of rupture

3 Summary

The issues presented in this appendix are summarized in Table 1 with their anticipated impact on the results discussed in the preceding sections. Since all of the issues evaluated are estimated to have negligible to small impact, the combined effect of correcting the errors is expected to be small. In none of the issues evaluated does the probability of rupture increase appreciably (by a factor of two or more) above the results presented with Version 1.0, except in the case for the two elevated H2 sensitivity analyses as detailed in I-5, Section 2.0.

The xLPR program includes a formal configuration management process for the lifecycle of the xLPR modeling system (models and codes). Issue tracking and resolution is part of the xLPR problem reporting and corrective actions process already in place for Version 1.0. Thus, the confidence in the base-case results can be maintained, and the validation activities performed on the base-case results remain applicable. The issues documented herein and any subsequent issues will be addressed in Version 2.0 of the xLPR model.