SANDIA REPORT
SAND2010-8480
Unlimited Release
Printed December 2010

Development, Analysis, and Evaluation of a Commercial Software Framework for the Study of Extremely Low Probability of Rupture (xLPR) Events at Nuclear Power Plants

Patrick D. Mattie, Cedric J. Sallaberry, Jon C. Helton and Donald A. Kalinich

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories. Operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacture, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online order:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2010-8480
Unlimited Release
Printed December 2010

Development, Analysis, and Evaluation of a Commercial Software Framework for the Study of Extremely Low Probability of Rupture (xLPR) Events at Nuclear Power Plants

Patrick D. Mattie, Cedric J. Sallaberry,
Jon C. Helton and Donald A. Kalinich

Reactor Modeling & Analysis
Nuclear Energy & Global Security Technologies
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS0748

Abstract

Sandia National Laboratories (SNL) participated in a Pilot Study to examine the process and requirements to create a software system to assess the extremely low probability of pipe rupture (xLPR) in nuclear power plants. This project was tasked to develop a prototype xLPR model leveraging existing fracture mechanics models and codes coupled with a commercial software framework to determine the framework, model, and architecture requirements appropriate for building a modular-based code. The xLPR pilot study was 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 includes a demonstration problem to assess the probability of rupture of DM pressurizer surge nozzle welds degraded by primary water stress-corrosion cracking (PWSCC). The pilot study was designed to define and develop the framework and model; then construct a prototype software system based on the proposed model. The second phase of the project will be a longer term program and code development effort focusing on the generic, primary piping integrity issues (xLPR code). The results and recommendations presented in this report will be used to help the U.S. Nuclear Regulatory Commission (NRC) define the requirements for the longer term program.

ACKNOWLEDGMENTS

This report documents a major effort as part of a program funded by the United States Nuclear Regulatory Commission (USNRC), Office of Nuclear Regulatory Research. The Sandia team that worked on the xLPR Project would like to thank the xLPR Computational Task Group Lead, David Rudland of USNRC Office of Nuclear Regulatory Research, Division of Engineering. In addition, Aladar Csontos of the USNRC Office of Nuclear Regulatory Research and Craig Harrington of Electric Power Research Institute (EPRI) for their support of this Project. The views herein are those of the authors and do not represent an official position of the USNRC or EPRI.

A number of people at Sandia National Laboratories including, Randall Gauntt, manager Severe Accident Analysis (Org. 6232), Alicia Aragon (Org. 6234), Yvonne McClellan (Org. 6232), Mona Aragon (Org. 6810) and Margaret Caledon (Org. 6223) have made significant contributions to production of this report including an internal technical review of this report.

The Sandia team sincerely appreciates our associates in the xLPR Computational Task Group and their generous support during the xLPR pilot study project. Specifically, Robert Kurth of Engineering Mechanics Corporation of Columbus and Battelle, Dilip Dedhia, David Harris, Anita Gubbi, and Clifford Lange of Structural Integrity Associates, Bruce Bishop and Nathan Palm of Westinghouse, and Paul Williams and Hilda Klasky of Oak Ridge National Laboratories.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

3MRA	Multimedia, multi-pathway, multi-receptor exposure and risk assessment
ARAMS	Adaptive Risk Assessment Modeling System
BRISC	Burner Reactor Integrated Safety Code - prototype
BWR	Boiling water reactor
CC	Complex crack
CCDF	Complementary cumulative distribution function
CD	Crack Depth
CDF	Cumulative distribution function
CFO	Critical Failure Occurrence
CM	Configuration management
CNWRA	Nuclear Regulatory Center for Nuclear Waste Regulatory Analysis
COA	Crack Opening Area
COD	Crack Opening Displacement
COTS	Commercial off the shelf
COV	Coefficient of variance
CPU	Central processing unit
DAKOTA	Design Analysis Kit for Optimization and Terascale Applications
dB	database
DEGB	Double-ended guillotine break
DLL	Dynamically linked library
DM	Dissimilar metal
DOD	U.S. Department of Defense
DOE	U.S. Department of Energy
DPD	Discrete Probability Distributions
EMC2	Engineering Mechanics Corporation of Columbus
EPA	U.S. Environmental Protection Agency
EPRI	Electric Power Research Institute
FLO	First Leakage Occurrence
FORM	First Order Reliability Method
FRAMES	Framework for Risk Analysis in Multimedia Environmental Systems
FSA	Fractional Surface Area
GDC	General Design Criteria
GE	General Electric
GTG	GoldSim Technology Group
GUI	Graphical User Interface

ID	Inner diameter
IGSCC	Intergranular stress corrosion cracking
I/O	Input/Output
ISI	Inspection Sizing
KSurf	Stress Intensity for a surface crack
KTWC	Stress Intensity for a through wall crack
LBB	Leak-Before-Break
LHS	Latin Hypercube Sampling
LOCA	Loss of coolant accident
MEPAS	Multimedia Environmental Pollutant Assessment System
MRP	Materials Reliability Program
MTS	Number of Intervals in the Operating History
NPS	Nominal Pipe Size
NRC	U. S. Nuclear Regulatory Commission
NURBIM	Nuclear Risk Based Inspection Methodology
OD	Outer diameter
ODBC	Open database connectivity
OOC	Object Oriented Commercial
OOOS	Object Oriented Open Source
ORNL	Oakridge National Laboratory
OS	Open source
PA	Performance assessment
PCC	Partial correlation coefficient
PDF	Probability density function
PIB	Project integration board
PND	Probability of non-detection
PNNL	Pacific Northwest National Laboratory
POD	Probability of detection
POR	Probability of repair
PRCC	Partial Rank Correlation Coefficient
PWR	Pressurized water reactor
PWSCC	Primary water stress-corrosion cracking
QA	Quality assurance
RCS	Reactor coolant system
SC	Surface crack
SCC	Stress corrosion crack

SI	International system of units
SIA	Structural Integrity Associates
SME	Subject matter expert
SNL	Sandia National Laboratories
SORM	Second Order Reliability Method
SQA	Software quality assurance
SQUIRT	Seepage Quantification of Upsets in Reactor Tubes
SRC	Standardized regression coefficient
SRP	Standard Review Plan
SRS	Simple Random Sampling
SRRC	Standardized Rank Regression Coefficient
SSE	Safe Shutdown Earthquake
STP	Standard temperature and pressure
TLR	Total Leakage Rate
TWC	Through-wall crack
UQ	Uncertainty Quantification
V&V	Verification and validation
WRS	Welding residual stress
xLPR	Extremely low probability of rupture
XML	Extensible Markup Language

1. Introduction

The U.S. Nuclear Regulatory Commission (NRC) has a requirement (10CFR50, Appendix A, General Design Criteria (GDC 4)) that primary piping system must exhibit extremely low probability of rupture (xLPR) in nuclear power plants in order to remove hardware meant to restrain the dynamic effects associated with such ruptures. NRC accepted the premise that a pressurized water reactor (PWR) primary loop piping rupture was unlikely to occur, provided nuclear power plants (i) conducted conservative deterministic fracture mechanics analyses and (ii) implemented and maintained leakage monitoring systems for detecting potential leaks. It is now permissible to eliminate the dynamic effects of postulated high energy pipe ruptures from the design basis for nuclear power plants using Standard Review Plan (SRP) 3.6.3 Leak-Before-Break (LBB) procedures.

However, there are current situations where plants are exhibiting active degradation mechanisms that violate the screening criteria in the SRP LBB procedures. The NRC and its licensees have recently discovered primary water stress-corrosion cracking (PWSCC) in various welds in the reactor coolant systems (RCS) of PWRs. An immediate investigation and inspection was conducted that revealed short through-wall cracks (TWCs) in welds, DM weld cracks, and pressurizer surge nozzle-to-safe end weld cracks (NRC Bulletin 88-11). These findings raised safety concerns based on the size and location of the occurrences. In addition, NRC staff has observed that PWSCC is occurring in systems that have been granted LBB exemptions to remove pipe-whip restraints and jet impingement shields (e.g., V.C. Summer Nuclear Power Station & Wolf Creek Generating Station).

After these potential safety concerns were identified, analyses were conducted to demonstrate that public safety is maintained despite a deviation from the SRP 3.6.3 prohibition against active degradation mechanisms. These activities included the following:

- Qualitative arguments have been made that the great majority of observed cracking is of limited extent and shallow depth. These factors tend to mitigate the risk of piping rupture,
- PWSCC mitigation activities have been implemented (e.g. reduction of mechanical stresses via the application of weld overlays or inlays over the PWSCC-susceptible welds), and
- An aggressive in-service inspection schedule to locate and mitigate any likelihood of this cracking.

While such actions are prudent, timely, and warranted, they indicate a need to quantitatively demonstrate the continued compliance with the regulation using a comprehensive piping system assessment methodology. To address this need, an assessment tool is required for the NRC and its licensees that can be used to directly assess compliance with the probabilistic acceptance criteria (10CFR50 Appendix A, GDC 4). Given the recent advances in probabilistic methodologies, performing a probabilistic analysis of primary system piping that fully addresses and quantifies uncertainties and directly demonstrates compliance with GDC 4 may be an appropriate alternative. This can be achieved by properly modeling the effects of both active degradation mechanisms and the associated mitigation activities. The proposed tool must be (i) comprehensive with respect to known challenges, (ii) vetted with respect to scientific adequacy of models and inputs, (iii) flexible enough to permit analysis of a variety of in-service situations and (iv) adaptable to accommodate evolving and improving knowledge.

To address these issues, the NRC and the Electric Power Research Institute (EPRI) cooperated through a memorandum of understanding on cooperative nuclear safety research for a pilot study for the development of a methodology and corresponding software tool to address the extremely low probability of pipe rupture (xLPR) in nuclear power plants primary piping systems. This pilot study was tasked to demonstrate the feasibility of developing a probabilistic software tool to address degradation mechanisms in piping system safety assessments. This pilot study is a proof-of-concept effort to develop a simplified assessment tool for DM pressurizer surge nozzle welds, for which a considerable amount of publicly available information exists. To complete this study, it was necessary to develop a prototype xLPR model leveraging existing fracture mechanics models. A deterministic and probabilistic analysis was conducted to test the framework and architecture requirements for building a modular-based code. Model verification, sensitivity and uncertainty analysis were conducted to demonstrate the xLPR model functionality. A stringent configuration management (CM) and quality assurance (QA) program was initiated to ensure access control, version control, verification, and traceability. The framework, model, and analysis were used to evaluate and determine the longer term program and code development requirements.

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. The pilot study addresses only the specific issue of assessing the probability of rupture of DM pressurizer surge nozzle welds degraded by PWSCC. The pilot study provided a short term learning experience that should benefit the longer term program and code development by identifying areas requiring more focused effort.

This document describes in detail the work SNL conducted in support of the NRC's goals established for the pilot study project.

1.1 Purpose

The purpose of this report is to provide a description of the multi-laboratory multi-disciplinary pilot study program focusing on SNL development, analysis, and evaluation of a commercial software framework for an extremely low probability of DM pressurizer surge nozzle weld leak and/or rupture (xLPR) events at nuclear power plants. This report provides a summary of SNL's contributions to the pilot study outlined in Section 1.2 Scope, including (i) the probabilistic methodology, (ii) a description of a robust analysis methodology for appropriately treating epistemic and aleatory uncertainties, (iii) framework model architecture design and development methodology and (iv) the results from the prototype xLPR model. The work presented herein supports the xLPR pilot study objective to evaluate the feasibility of verifying, validating, benchmarking, and documenting a software tool for use in support of licensing, rulemaking, design, and regulatory decisions by both industry and the NRC.

1.2 Scope

The xLPR project objective is to implement a cooperative research program that focuses on (i) probabilistic evaluation methods and (ii) development of a software tool for the evaluation of nuclear plant primary piping systems. The successful execution of the xLPR project involves a complex and diverse array of technical specialties and includes a team of experts representing various disciplines. The overall project organizational responsibilities and specific xLPR task groups are outlined in Section 1.3 Project Roles and Responsibilities. The xLPR project has been divided into two parts, (i) a short term pilot study and (ii) a long term program. As discussed previously the purpose of this report is to provide a description of the xLPR pilot study program. Much of the work performed on the pilot study project was distributed to various organizations through separate contract actions by NRC and EPRI. SNL's role in the computational task group, including the development, analysis, and evaluation of commercial software framework, is outlined in Section 1.3.1 Sandia National Laboratories Roles and Responsibilities.

1.3 Project Roles and Responsibilities

This section summarizes the complex coordination and cooperation of multiple-organizations working together to achieve xLPR project success. The NRC encourages cooperation in nuclear safety research and contracted SNL to participate in cooperation with EPRI, Oakridge National Laboratory (ORNL), Battelle, Westinghouse, and Structural Integrity Associates (SIA) to develop the xLPR framework and model. This cooperative effort includes the evaluation of probabilistic methods and development of software to evaluate the potential failure of nuclear power plants primary piping systems due to PWSCC. The objectives of this effort include (i) the development of a robust analysis methodology for evaluating RCS piping rupture probabilities; (ii) the selection of appropriate, technically sound input data and models to produce best-estimates output results with quantified uncertainties; (iii) the development of a computational software tool that applies the input data and models and appropriately treats epistemic and aleatory uncertainties; (iv) and the verification, validation, benchmarking, and documentation of the software tool to enable its use in support of licensing, rulemaking, design, and regulatory decisions by both industry and the NRC.

Successful completion of the long term xLPR project will be accomplished with a well-organized, disciplined, structured team of technical experts in the following areas: plant design; component geometry; environmental loading; flaw inspection and assessment; degradation modeling, critical flaw stability and detectible leak rate modeling; software development and programming; probabilistic acceptance criteria; and program management.

The xLPR project required a tiered approach for managing the complex structure, roles, responsibilities, requirements, and task groups. Specific task groups were developed and include the following: Models, Inputs, Computational, and Acceptance. An xLPR project integration board (PIB) was established that included representatives from the NRC, the nuclear power plant industry, and the four task groups to oversee the project. The PIB provided overall direction and ensured effective communication with and between task groups. In addition, the PIB made project decisions with consideration of each task group's input and recommendations.

The Table 1-1 summarizes the actions and task group responsibilities performed for the xLPR project. The actions listed below were assigned to the task groups to accomplish.

Table 1-1. Description of task group responsibilities and actions.

Task Group	Responsible Party	Actions
Project Integration Board	EPRI NRC	Provide overall project direction & resolve conflicts. Develop a detailed project scope of work. Set and review project milestones. Define goals & expectations for pilot study. Identify and provide resources to task groups. Establish project review schedule. Facilitate communication between task groups and constituencies (NRC, industry, and contractors). Ensure progress towards short- and long-term goals. Develop recommendations for long-term study. Generate plan for user training, user manual development, and program maintenance.
Inputs	EPRI - Primary NRC – Secondary	Determine input types for pilot study. Finalize recommended inputs. Develop input distributions where possible. Provide support to models and computational groups.
Models	EPRI NRC	Develop model assessment protocol. Assess available models relative to protocol. Develop & populate model database. Recommend consensus selection of models for pilot and final studies.
Computation	EPRI NRC	Conduct uncertainty workshop. Review and assess legacy codes. Develop modular computational framework using open source and commercial software. Develop, debug, and exercise codes. Provide for rigorous tracking of uncertainties. Run pilot problem and provide recommendations for long-term study. Produce final code with thorough documentation and training materials.
Acceptance Criteria	EPRI NRC	Develop software QA plan to support release and use of final product. Recommend definition of “failure” (e.g., flaw

Task Group	Responsible Party	Actions
		detection, flaw growth to some size, leakage, rupture). Recommend maximum tolerable "failure" frequency.

1.3.1 Sandia National Laboratories Roles and Responsibilities

This section details the Sandia xLPR program tasks including both programmatic and technical accomplishments supporting the NRC's xLPR pilot program objectives for the computational task group. The specific Sandia program requirements were integral to a complex coordination and cooperation of multi-organizations comprising the computational task group. The computational task group work scope was contracted to SNL, Oakridge National Laboratory (ORNL), Battelle, Engineering Mechanics Corporation of Columbus (EMC2), and two EPRI supported contractors (Westinghouse and SIA).

The Table 1-2 outlines the Sandia tasks supporting the primary computational group roles and responsibilities. The xLPR project start date was April 09, 2009.

Table 1-2. Task Group Actions and SNL Roles & Responsibilities.

Task Group Actions	Sandia National Laboratories Roles and Responsibilities
Conduct uncertainty workshop.	Develop and conduct an uncertainty workshop for the models, input and computation task groups. – *Completed task June 2009*
Review and assess legacy codes.	Review of legacy codes and assessment for the pilot study problem and beyond. [SNL, Battelle, NRC, SIA, EMC2]
Develop modular computational framework using open source* and commercial software.	Determination and comparison of available commercial software alternatives. [Sections 1.4, 6.2, & Appendix A] Develop computational framework architecture. [Sections 1.6 & 3.0] Provide a CM process for software development. [Section 1.5] Develop computational framework using commercial software. [Sections 1.4 & 3.0]
Develop, debug, and exercise codes. [Modules for the xLPR Framework]	Develop and conduct a workshop to ensure successful development of modules from models group. June 2009, September 2009, October 2009, December 2009.
Provide for rigorous tracking of uncertainties.	Investigate uncertainty handling methodologies and recommend procedures for xLPR code. [Sections 1.6, 6.0 & Appendix D]
Run pilot problem and provide recommendations for long-term study.	Review and revise NRC xLPR Program Plan. [Section 2.0 & Appendix B] Define inputs for pilot study. [Appendix C] Coordination and Pilot Study Problem Statement Development. [Appendix E] Conduct parameter sensitivity and uncertainty analyses. [Section 5.0] Sensitivity studies between Monte Carlo with LHS and DPDs with importance sampling. [Section 5.3] Assess results and make recommendations for further xLPR and full

Task Group Actions	Sandia National Laboratories Roles and Responsibilities
	modular code development. [Section 6.0]
Produce final code with thorough documentation and training materials.	Develop and document User's Guide for commercial software xLPR framework. [SAND2010-7131] Develop and document pilot problem, analyses, and recommendations. [this report]
Project Integration and Scheduling	
CM for software development and pilot study analyses	Develop CM plan and procedures for pilot study problem. [Section 1.5] Design CM system within contemporary project resources. [Section 1.5.3] Conduct CM training workshops. *December 2009, October 2009, June 2009.*
Reference: xLPR Initial Framework Development, NRC Form 189, Job codes N6820-00 and N6820-01. **Development of computational framework using open source software was tasked to ORNL.*	

This report documents many of the specific SNL tasks completed for the xLPR program. An entire series of reports, as outlined in Figure 1-1, encompasses the complete documentation of the xLPR pilot study. The report flow chart follows the integral xLPR program structure described in Section 1.3 and reflects the coordination and cooperation of multiple-organizations comprising the xLPR program.

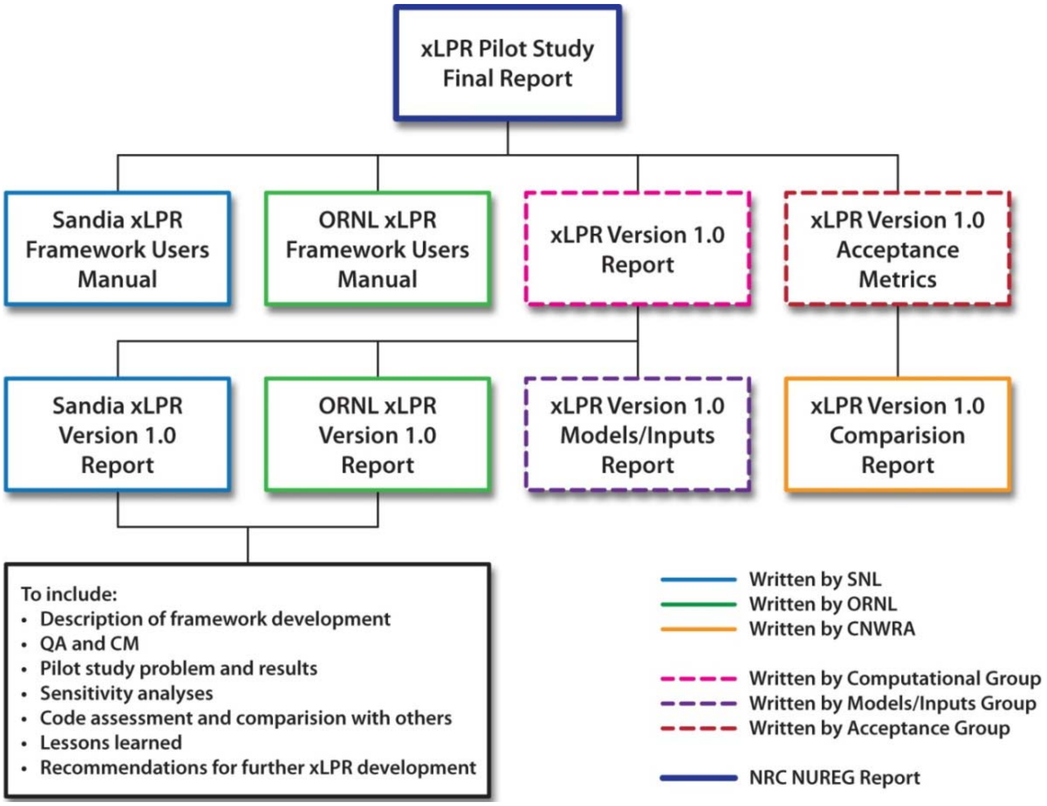

xLPR Pilot Study Final Report
Sandia xLPR Framework Users Manual
ORNL xLPR Framework Users Manual
xLPR Version 1.0 Report
xLPR Version 1.0 Acceptance Metrics
Sandia xLPR Version 1.0 Report
ORNL xLPR Version 1.0 Report
xLPR Version 1.0 Models/Inputs Report
xLPR Version 1.0 Comparision Report
To include:
• Description of framework development
• QA and CM
• Pilot study problem and results
• Sensitivity analyses
• Code assessment and comparision with others
• Lessons learned
• Recommendations for further xLPR development
Written by SNL
Written by ORNL
Written by CNWRA
Written by Computational Group
Written by Models/Inputs Group
Written by Acceptance Group
NRC NUREG Report

Figure 1-1. xLPR Pilot Study Reports.

1.4 Evaluation of Commercial Software for the xLPR Framework

SNL was tasked to survey and evaluate commercially available software for use as a probabilistic framework for the xLPR Modular code. After an extensive web based review, four potential commercial software packages were selected for further evaluation as part of this effort (listed in Section 1.4.1). Three commercial software packages were developed mainly to support system dynamic based problems of varying complexity. Most of the commercial software packages had common features including: graphical icon based user interface for model building, multi-level hierarchical model structure, dashboard interface, the ability to add text and graphic to the model interface, and the ability to link to Microsoft® Excel for dynamic data importing and exporting. To test the feasibility of alternative commercial software a simplified xLPR demonstration problem was defined. The commercial software was screened based upon initial review of the software's available features and after consultation with SNL staff with the appropriate experience with the software. Two of the commercial software, Vensim® and Powersim Studio®, listed sufficient capabilities (e.g., link to external modules), to be potentially used as a framework for xLPR. However, it was determined that both commercial software were limited in terms of their ability to execute probabilistic simulations of complex model systems. Based on resident expertise, Powersim Studio was determined to be most likely to meet the xLPR framework model criteria and was used to build the demonstration test case.

The results of the test problem revealed that while Powersim Studio can match the time-dependent nature of the xLPR crack propagation problem, it could not make calls to the xLPR Fortran dynamically linked libraries (DLLs). This is primarily due to an inability in VBScript to make direct DLL calls. SNL found no alternative commercial software acceptable for the xLPR framework based upon this evaluation and previous participation in a commercial software evaluation effort with a similar scope. The results of that study corroborate the SNL evaluation for xLPR [1].

Due to the limited number of suitable commercial software for the xLPR model framework model criteria, several U.S. Department of Energy (DOE) sponsored software specifically developed as software frameworks were considered, and are listed in Section 1.4.2. Two software considered, FRAMES and DAKOTA, are sufficiently supported and have been extensively developed to be considered equal to commercially developed products. As each software was specifically developed as a framework model platform, each has the basic features necessary to meet the xLPR framework model requirements. The FRAMES software was developed specifically as an environmental framework. Whereas, the DAKOTA framework is primarily for large scale engineering optimization and uncertainty analysis. In addition, two of the software, the BRISC and DAKOTA, were developed primarily for LINUX based parallel computational platforms. BRISC was developed as a framework for modeling complex multi-physics systems. The laboratory developed software was screened based upon initial review of the software's available features and after consultation with SNL staff with the appropriate technical experience with the software. However, as each is desirable for their specific design capabilities; only FRAMES and BRISC were selected to be evaluated using the demonstration problem. The DAKOTA software was determined to be more useful as a potential addition to xLPR for uncertainty analysis.

The simplified test problem was not successfully implemented in the FRAMES code due to the specialized module requirement and lack of resident experience at SNL creating these modules. However, a thorough evaluation of the software is contained in Appendix A.3. The FRAMES software is primarily used and tailored to the environmental science domain. This software has been used as a framework platform for a modular model system as envisioned for the xLPR application. However, a major drawback against potential application to the xLPR is that the software in its current version lacks the infrastructure for probabilistic analysis.

BRISC did successfully implement and run the xLPR test case problem. However, there are some limitations. BRISC was developed as a research project to demonstrate the efficient solution of coupled multi-physics problems in parallel. As such, it was primarily a research code designed for knowledgeable developers without much support or focus to aid the end user. There is no graphical user interface (GUI), the input file syntax is extremely minimal and the code is sparsely documented. Even with these usability issues, BRISC is a very capable multi-physics coupling framework that, combined with uncertainty quantification (UQ) capabilities, represents a very compelling computational engineering tool for xLPR.

A detailed evaluation of each software evaluation based upon the xLPR demonstration problem is provided in Appendix A.

1.4.1 Available Commercial Software Considered

1. Powersim Studio® – Graphical system dynamics simulation environment. http://www.powersim.com/ ;

2. Vensim® - Vensim is used for developing, analyzing, and packaging dynamic feedback models. http://www.vensim.com/software.html ;

3. STELLA®/iThink® – STELLA is a system dynamic's based software package with an icon based GUI and influence tracing capabilities. www.isesystems.com; and,

4. Simcad® - Simcad is a discrete event simulation package. It is designed for tracking individual discrete items throughout the lifecycle process http://createasoft.com/.

1.4.2 Other Software Considered

1. FRAMES – FRAMES (Framework for Risk Analysis in Multimedia Environmental Systems) is a software platform developed by Pacific Northwest National Laboratory (PNNL) for selecting and implementing environmental software models for risk assessment and management problems. http://mepas.pnl.gov/FramesV1/index.stm

2. BRISC - SNL developed as a laboratory funded research project to demonstrate the efficient solution of coupled multi-physics problems in parallel [2].

3. DAKOTA – The SNL developed DAKOTA (Design Analysis Kit for Optimization and Terascale Applications) toolkit provides a flexible, extensible interface between analysis codes and iterative systems analysis methods. http://dakota.sandia.gov/index.html.

1.4.3 Selection of GoldSim

GoldSim is dynamic, probabilistic simulation software providing several simulation approaches (e.g., system dynamics, discrete event simulation) embedded in a Monte Carlo simulation framework. The GoldSim software is the platform used to visualize, simulate, and manipulate the xLPR framework and model. This software allows the user to readily evaluate how a system will evolve over time in order to predict future behaviors and events.

GoldSim is a general-purpose simulation software program designed to simulate complex systems. Some of GoldSim's key features that make this software appropriate for xLPR include the ability to:

- Quantitatively address the inherent variability and uncertainty that is present in real-world systems (using Monte Carlo simulation).
- Superimpose the occurrence and consequences of discrete events onto continuously varying systems.
- Build top-down models using hierarchical containers that facilitate the simulation of large, complex systems while keeping them easy to understand and navigate.
- Dynamically link external programs or spreadsheets directly into your GoldSim model.
- Directly exchange data between any Open Database Connectivity (ODBC)-compliant database and your GoldSim model.
- Insert graphics, notes and hyperlinks (to documents, websites, and presentations) in order to clearly explain your model to diverse audiences.
- Create custom designed graphical interfaces (called dashboards) to explain and demonstrate your models using the GoldSim Dashboard Authoring Module.
- Save your models as player files that anyone can view and run using GoldSim Player (a free download).
- Utilize powerful extension modules to address problems that can't be adequately represented using simple simulation approaches.

GoldSim is a powerful and flexible platform for visualizing and dynamically simulating nearly any kind of physical, financial or organizational system. You build a model in an intuitive manner by literally drawing a picture (an influence diagram) of your system. GoldSim Pro is the commercial version of GoldSim and provides all the basic functionality needed by commercial users to build powerful, full-featured GoldSim models for business, scientific, and engineering applications. GoldSim Player is a ***free download*** that allows anyone to view, navigate, and run GoldSim models. As a result, GoldSim Pro users are allowed to share their models with people who don't have GoldSim Pro or GoldSim Academic. GoldSim Player does not allow users to create new models or edit existing models. However, Player versions can be developed in a way to let the user change inputs values as well as specificities of the Monte Carlo simulation (sample size, random seed used).

A detailed summary of the general ways that GoldSim differs from other simulation software is provided on their web page: http://www.goldsim.com/Web/Products/GoldSimPro/SimulationApproaches .

1.5 xLPR Model Development Process

The xLPR pilot study program was divided into three phases. The first two phases cover the model freeze points, Alpha and Beta, for the xLPR model development effort. The final phase of the pilot study is Version 1.0 of the xLPR model. Version 1.0 of the xLPR model is used to conduct the final analyses used to evaluate the framework and module software. This section discusses processes used to control the xLPR model during model development and analysis of xLPR Model Version 1.0. The controls are applicable for model development, model testing, correction of model errors, and the production of xLPR simulations. The controls cover the entire life-cycle of the model, from management direction of what changes are to be made to the model, to the control of completed xLPR simulations and their results. These controls provide guidance for implementing the program requirements.

The controls and guidelines for this work were documented in a series of draft documents for use by all xLPR scientists. The work scope covered included xLPR Pilot Study Model Development (Alpha and Beta Versions), the completed/final xLPR Pilot Study Model, and the Supporting Production Runs and Sensitivity and Uncertainty Analyses. The xLPR program has made a commitment to implement the principles of CM in the subject work scope.

Each process is defined within the draft guidance documents (Table 1-3) to address access controls, version control/change control, verification (inputs, parameters, conceptual models, model coding or implementation, model calculation, and results), and documentation that the process has been completed and can be verified independently (e.g., without consultation with the originator). 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.

Table 1-3. Draft xLPR CM Process Guidance Documents.

CM Desktop Guidance Document	Description of Process
xLPR-DSK-001	Module and Model CM
xLPR-DSK-002	Control of Inputs and Model Parameters
xLPR-DSK-003	Module and Model Change Control Process
xLPR-DSK-004	Module and Model Documentation and Checking
xLPR-DSK-005	Plot Generation
https://websps1.battelle.org/nrcnureg/home/xLPR_CM/CM%20Documents/Forms/AllItems.aspx	

1.5.1 Versioning and Version Control

Model Development is the updating of the current xLPR model in incremental changes that are sized to facilitate checking of the model with the final change resulting in the completion of the xLPR Alpha or Beta Model Version. In order to create and understand the xLPR Framework Model there are two types of cases that were produced.

1. **Model Development File(s) Unique ID.** The naming convention that will be used for the development of the xLPR Model Files. (Alpha and Beta Versions)
2. **Model Production File(s) Unique ID.** The naming convention that will be used for the xLPR Model Files once the xLPR Model is frozen. (Version 1.0)

Model Development cases (i.e., Alpha or Beta) and all supporting documentation are stored in a separate Development case folder on the Battelle SharePoint Site. The naming convention (see example below) for the Model Development framework file and directories is:

AAAA_vB.CC_GSxx.yy.zzz_Mqq
AAAA = Alpha or Beta Version
vB.CC = Global case version number *(format #.##)*
GSxx.yy.zzz = GoldSim Software version *(format example GS10.11)*
NOTE: zzz is not used if the version does not include this level of detail.
Mqq = Module (DLL) set for the model version *(format example "M01")*

Example: Naming Convention for Model Development
Alpha_v1.00_GS10.02_M01.gsm (or .gsp)
Beta_v2.02_GS10.11_M02.gsm (or .gsp)

In addition to the naming convention, the GoldSim Pro software provides the ability to internally track changes that you have made to your model file. This feature (referred to as versioning) allows you to quickly determine the differences between the current version of your model file and previous versions of the file. Providing this CM capability is particularly useful for:

1. Coordinating model changes when multiple people can access and modify the model file;
2. As a QA/Quality Control feature enabling you to demonstrate and document when and what changes have been made to a model file.

Changes to a model file are tracked by creating model file versions. A version is an internal "snapshot" of your model file at a particular point in time. During the development process versions were created and assigned a title (e.g., in accordance with the naming convention). Each version can be compared to the current model (the model as it exists right now) or to any previous model version. The GoldSim Pro software can report the differences between the current model and any previous version. Note this feature is not available in the free GoldSim Player software.

Upon completion of the Model Development process, xLPR program switches from a Model Development Phase into a Production Phase. It is during the Production Phase that sensitivity case model runs are performed. These sensitivity case model runs are referred to as the Production Runs. The Production cases are run from the controlled file set.

The naming convention for the Production case is:

GSxLPRv1.0C_Mzz_xxxxx

- GSxLPR_ = xLPR Pilot Study
- v1.0C = Version 1.0x of the controlled version.
- Mzz = Module Set
- xxxxx = unique name (e.g., E1000A50 for a 1,000 epistemic and 50 aleatory run, or Mit30 for a mitigation sensitivity at 30 years).

Example: GSxLPRv1.02_M02_Deterministic.gsp

In addition to the framework model files, each module developed and used in the xLPR pilot study was given a unique identifier. Changes to a module are tracked by creating module file versions. The naming convention for the modules is as follows:

name_va.aa

- name_ = Module Name (e.g., COD, DPD, Grower, ISI, etc)
- va.aa = Module version. 1.0 = Alpha and 2.0 = Beta. For revisions after the module was frozen, the convention 1.01 or 1.02 is used.
- An optional _DLL or _DLLx was added after the module name for the DLL compiled code.
- va.aa = For Module DLL. The conversion from modules developed as executables (.exe) to GoldSim compatible DLL's were given the additional identifiers v1.1 or v2.1. For revised module executables (e.g., v1.01) the DLL version of the revised module would be indexed to v1.2.

Examples: Coalesce_v2.0 (Beta Version of exe for coalesce module)
Coalesce_v2.01 (revision to Beta Version of exe for coalesce module)
Coalesce_DLL_v2.1 (DLL module version for coalesce module v2.0)
Coalesce_DLL_v2.2 (DLL module version of revised module v2.01)

In addition to the naming convention, the GoldSim software provides the ability to internally track module versions in the module source code. The GoldSim software DLL interface requirements include that a module version number is passed to the GoldSim software during runtime and this version number is automatically recorded in the model run log. This feature (referred to as module versioning) allows for quickly tracking the versions of each module used in any model run.

Many modules are used in the xLPR model and are collectively referred to as a "Module Set". Each module in a set appears in the controlled framework model name and is uniquely identified by name and version number. Each Module Set used in this work scope was assigned a unique ID number. The supporting modules that comprise that Module Set are identified in a Module Set Readme file. When the Module Set needs to be updated (e.g., as new modules are added or the existing modules are revised), the new set number will increase by an increment of one. During the module and model development phase, establishment of a Module Set was determined by the xLPR CM Lead. Three module sets were created: M00, M01, and M02. The first two sets were used during the Alpha model and early Beta development phases and the last model set (M02) was used for the final Beta model development and xLPR Version 1.0 model.

1.5.2 Configuration Management

CM is the process of identifying and defining the configuration items in a system, controlling the release and change of the items in the system, reporting the status of the items in the system, and verifying the completeness and correctness of the items in the system. The first process in the CM system, called configuration identification, is the unique identification of all the items to be managed in the system. Configuration identification consists of selecting the items to be managed and recording their functional and physical characteristics. The second process is configuration change control. Configuration change control is the mechanism for approving or disapproving all proposed changes to the system that is being managed. Configuration change control ensures that changes to any configuration items are approved and controlled so that consistency among components is maintained. The third process is called configuration status accounting. Information contained in the status accounting system will document the evolution of the xLPR model in a transparent and traceable manner. The last process is review. The review consists of checking the configuration items to verify that they match the requirements. The pilot study has implemented a CM process including technical review of the xLPR model CM items.

The xLPR CM process consists of 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 process ensures that a systematic approach is used to meet the requirements and includes 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 study: 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 process is defined within the guidance documents (listed in Table 1-3) to address access controls, version control/change control, verification (inputs, parameters, conceptual models, model coding or implementation, model calculation, and results), and documentation that the process has been completed and can be verified independently (e.g., without consultation with the originator). CM for software, model, and model inputs utilized in xLPR model are discussed in the following subsections.

1.5.2.1 Software and Module Configuration Management

All software source code and compiled software (e.g., modules) used for the xLPR model have been verified (checked) and placed under the controls of the CM program. Baseline software is software that has been formally reviewed, can only be changed through a formal change process, and is ready for project use. Each module in CM is uniquely identified with a version number. The electronic storage system used for CM controlled software includes information on the software name, version, and operating system.

To support the xLPR model, a number of software codes were used. The codes may be used for providing supporting information and directly implementing the xLPR model. In addition, the pilot study involved the assessment of the feasibility of creating a modular-based computer code using an Object Oriented Commercial (OOC) Software Framework. In this approach the software was used to in conjunction with the inputs and models (modules) to demonstrate the feasibility and agility of the software framework to address the requirements for the xLPR code. For the purposes of CM identification, two software types were used:

GoldSim - GoldSim is a Windows-based OOC software code that is the model framework simulating the xLPR model. Probabilistic simulations are represented graphically in GoldSim. Models are created in GoldSim by manipulating graphical objects, where these objects represent the data, functions, and logic defining the system being simulated. GoldSim is flexible in its ability to incorporate a variety of data tables, other software modules, and information in defining the overall system model. The GoldSim software is a verified and validated commercial software product. As such the CM process was used to manage only the xLPR model framework file development and following the process described in Section 1.5.2, Control of the xLPR Model. Only the requirement for the commercial software was that the version of the commercial software used during model development and production runs was tracked and documented.

xLPR Modules – A set of software codes (modules) are used to simulate the processes that are important to the evaluation of degradation mechanisms in piping system safety assessments. The software modules were developed primarily in Fortran or C++ and compiled as a DLL compatible with the GoldSim software. The source codes were developed or extracted from existing software codes and are used to model the initiation processes, degradation mechanisms, and inspection and mitigation components of the pilot model system. The modules are standalone applications that were verified independently from other modules and the framework software. The CM process for the xLPR Modules follows the process described in Section 1.5.2.

1.5.2.2 Model Configuration Management

Software codes (modules) are used to simulate the processes that are important to the evaluation of degradation mechanisms in piping system safety assessments which are coupled to a framework, which together with the data input file comprise an integrated probabilistic xLPR model. A detailed description of each component of the xLPR Model is contained on the xLPR CM electronic storage system. All of the functionality implemented in GoldSim as well as the external software modules was developed and verified (e.g., checked) in accordance with the xLPR CM guidelines. During development, the GoldSim model file (e.g. xLPR framework), external modules and inputs were stored in a controlled directory (e.g., electronic storage device). All proposed changes or modifications to the controlled files were reviewed and approved by the designated CM administrator. All changes to the model were checked. For specific details on the model development, model checking and model change control, see ***Control of the xLPR Model***.

Configuration Management of xLPR Model Inputs

All parameters implemented in the model were controlled as part of the xLPR model file. The xLPR model input parameters were developed and verified by the xLPR Models and Inputs task group. The input file was stored in a controlled directory (e.g., electronic storage device).

Control of the xLPR Model

Controls were applied for framework and module development, model testing, correction of model errors, and the xLPR production simulations. The controls cover the entire life-cycle of the model, from management direction of what changes are to be made to the model, to the control of completed xLPR simulations and their results. The controls and guidelines for this work are documented in a series of guidance documents and CM templates for use by all xLPR scientists (Table 1-3).

Description of the xLPR Model

The pilot xLPR model consists of three major parts: the framework model file (e.g., GoldSim model file), the modules (e.g., DLLs called by the GoldSim framework model file), and the set of inputs and parameters used in the pilot study (e.g., uncertain and constant input data to the GoldSim model file which is passed to the modules during a simulation).

Management Control

Once the baseline model was established (e.g., the Alpha or Beta model was frozen) the xLPR model required modification for a number of reasons (e.g., module and/or framework error correction, sensitivity studies, uncertainty characterization, model enhancement, etc.). The management control of these changes is presented in this section.

The xLPR model, including framework, modules and inputs developed for the pilot xLPR model was periodically reviewed by the appropriate subject matter experts (SMEs) to ensure that the implementation was consistent with the SMEs intent for the model. Any discrepancies were addressed. Potential changes to the xLPR model due to identified errors or to address review comments were implemented and reviewed by xLPR project staff. Additional xLPR model changes (e.g., change in model logic) were identified as part of xLPR model development. Changes within the xLPR model development scope and schedule were approved by the NRC xLPR computation group lead. Management approval of changes to the xLPR model were predicated on whether the change is necessary to comply with program requirements, if the final input feed date had passed for the requested change, or if the xLPR model itself has been finalized or “frozen”.

Changes to the xLPR model were tracked by the xLPR model lead. For the production stage, written approval from the xLPR Project Manager, xLPR Model Calculations Lead, and xLPR CM Lead was required to change/introduce new functionality, modules, inputs or parameters into the controlled xLPR model. The written authorization specifies the source(s) of these changes using a change approval form.

During the model development phase and after version 1.0 freeze date (for controlled files), any errors in the xLPR Model (framework, modules, associated files) were tracked with an Issue Tracking Log on the SharePoint web site (https://websps1.battelle.org/nrcnureg/home/xLPR_CM/Lists/Issue%20Tracking%20Log/AllItems.aspx).

Physical Control of Files

The xLPR model file and its associated modules, and inputs are controlled by storing them in a set of controlled subdirectories on the xLPR file server. The electronic file server for controlled storage of xLPR model files was created for this pilot study by SNL and hosted by Battelle using the Microsoft® SharePoint process and document management software: (https://websps1w.battelle.org/nrcnureg/home). Read access to these subdirectories is limited to xLPR scientific staff and granted by permission of the NRC xLPR Computational Task Group lead. Write access was controlled and was limited to the module or model developer, xLPR Model Calculations Lead, the xLPR CM Lead, and the System Administrator.

Model and module development subdirectories were established for control and documentation of the development process. Once development is completed and the product verified, the subdirectory was locked to read only by the xLPR Model Calculations Lead and the xLPR CM Lead.

A baseline file set was established by the xLPR CM Lead. Any subsequent changes to the input files were documented as changes to the baseline and were documented by the xLPR Model Calculations Lead and the xLPR CM Lead using a change control form. Modules for the xLPR model are stored in a controlled subdirectory on the xLPR electronic file server. The modules along with their documentation and verification checklists are stored in controlled subdirectories used for model development.

Input parameters (both certain and uncertain) for the xLPR model are controlled. The inputs are stored in a controlled subdirectory on the xLPR file server. The input database for the pilot study is an Excel spreadsheet which is maintained under configuration control.

Completed xLPR model runs and sensitivity cases are stored in a controlled subdirectory on the xLPR file server, Production Runs. Also, any post-processed results, plots, additional calculations or documentation to support a given case or set of cases are stored in this controlled subdirectory on the xLPR file server. Due to the large file sizes with results and file size limitations of SharePoint, only the input spreadsheet, model file, plots and post processed files will stored for traceability and verification purposes.

xLPR Model and Module Change Control and Checking

Approved changes to the xLPR model and modules are documented in a conceptual description of the changes, a checklist of the changes to the model, and a change approval form for changes to the controlled set. The conceptual description provides an overview of the changes that were incorporated into the model or module version. It also contains documentation of any development and testing work that was performed to support the change to the xLPR model.

The checklist documents the specific changes made to the model files or modules. The change log provides a record of what changes were actually made (e.g., changes to or addition of new parameters in the GoldSim framework model file or changes to existing module source code). The conceptual description, checklist, and xLPR model file or module source code were checked to verify that the changes are correctly implemented into the xLPR model.

Checking was performed by a qualified individual (assigned by the xLPR Model Calculations Lead), who was not involved in modifying the controlled model file or module. Checking was done incrementally during model development.

Two types of checks were done; parameter-level checking and conceptual model checking. Parameter-level checking verifies that all of the changes to the model file, source code and/or input files were done correctly. Conceptual model checking considers whether the implementation in the model correctly reproduces the conceptual model in the associated scientific analysis.

Parameter-level checking was documented in a change checklist. The steps involved in this check include:

1) Check changed/added parameters or source code (e.g. GoldSim elements, Fortran source code) against their source information to verify that they were changed correctly.
2) Verify that the input links of added elements or modules are correct.
3) Verify that the output links of added parameters or functions are correct.
4) Check that the links to and from any deleted elements, functions, arrays, etc. have been appropriately reconnected.
5) Verify (by inspecting source references for changes) that each change is correct.

For controlled models a full multiple-realization run of the model was performed. The results of this run were compared with the previous controlled version to verify that the changes had the expected affect. During model framework development there was not baseline controlled model to compare, so a hand check using a deterministic test case was used. The modules were verified using a “standalone” framework, rather than executing the entire xLPR model. A series of test cases were provided by the module developer that was run independently by the checker to confirm the module was working as intended.

The conceptual check evaluated whether the changes to the module or model system (e.g., framework, modules, data input, etc.) correctly reflects the conceptual model changes. The conceptual description includes a description of the changes and or functionality incorporated in the updated version, including the technical basis supporting the change (if needed). Any development and testing work to support the changes was also documented in the conceptual description. General questions that the conceptual checklist addressed (if applicable, if not applicable then so noted) include:

Does the modified portion of the model respond appropriately to its inputs?

Do the model components downstream from the modifications respond appropriately?

Are model inputs and outputs within their specified ranges?

Can each entry in the GoldSim run log or notification of compiler errors be shown to have no or a negligible impact on the run?

Any differences between the results of the initial and modified case were resolved and properly documented by the checker and originator.

Managing xLPR Model Inputs

xLPR model input parameters (excluding simulation settings and xLPR system parameters) were managed through the Excel xLPR input spreadsheet. The spreadsheet does not perform any calculations or logical evaluations; rather it strictly acts as a central storage location for input parameters used in xLPR model file.

Since the xLPR inputs are part of the overall xLPR model, they were developed, controlled, and documented in the same manner as the other parts of the xLPR model. This was the responsibility of the xLPR Inputs Task Group. However, the SNL computational task group team and CM Lead established the configuration control of this CM item. Changes to the controlled version of the Excel input file required a change approval form and checking.

Control of xLPR Model Results

xLPR model results consist of the model simulation files (i.e., .gsm file, modules not in the baseline, modified input spreadsheet, etc.), information extracted from the model (e.g., plots, tables), and post-processed information (e.g., expected value, inspection, leak detection, uncertainty and sensitivity analyses).

The xLPR model simulations were documented using change checklist and conceptual description. The files contain descriptions of the simulation, the supporting documentation, any changes to input files and input parameters, and the software used. The conceptual description and changes to the xLPR model files were checked to ensure the simulation and results were conducted as expected.

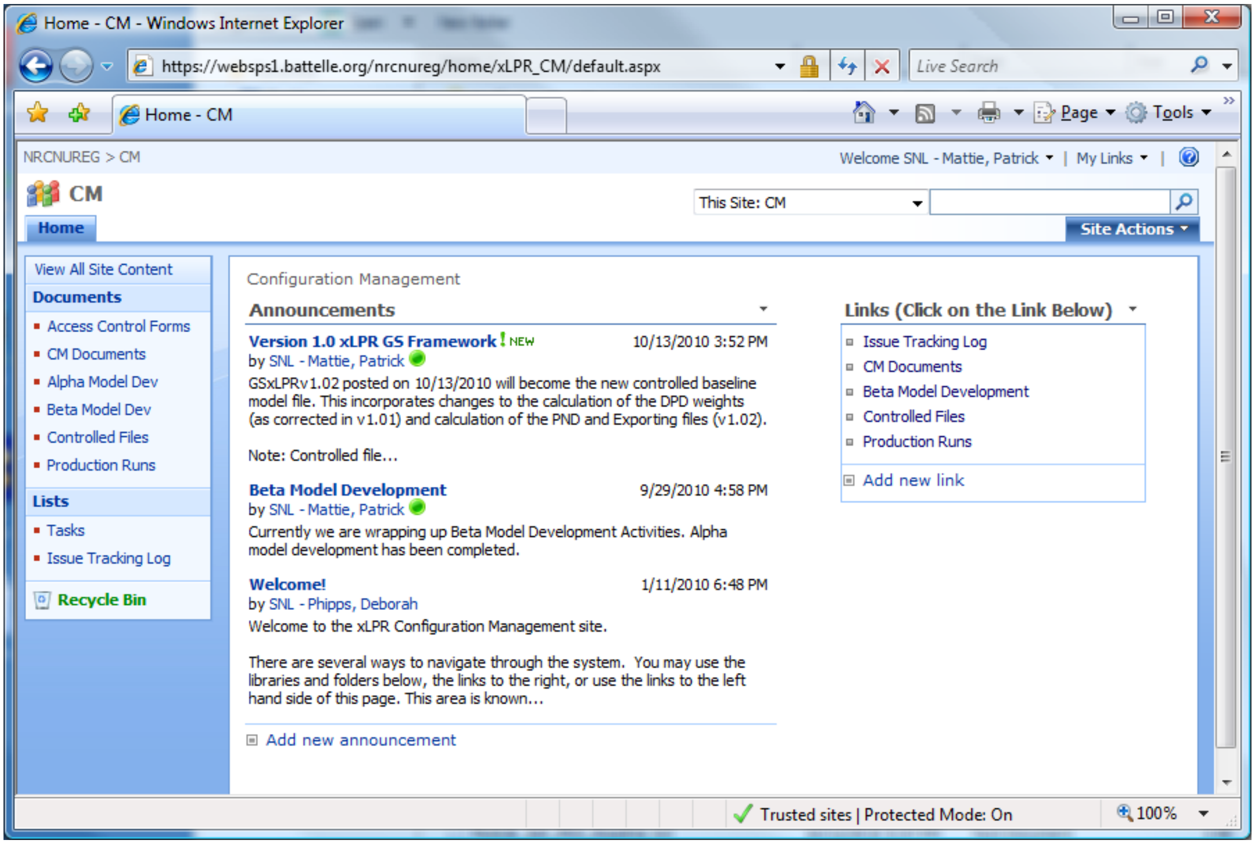
1.5.3 SharePoint Site

A specific internet-based file share site was established and provided change control of Alpha and Beta models. This site has access controls to prevent loss, damage, or theft of the work products. The xLPR electronic storage system (Battelle hosted Microsoft® SharePoint Site: https://websps1w.battelle.org/nrcnureg/home) was established as the repository for the CM items for xLPR. This SharePoint site serves two primary functions (1) a controlled source for data and files (approved for use), and (2) provided sufficient tracking and traceability of the CM items used in the xLPR model.

Microsoft® SharePoint is a family of software products developed by Microsoft® for collaboration, file sharing and web publishing. SharePoint seeks to provide a wide range of messaging, collaboration, publishing and document management features in one server. For xLPR CM, the NRC's SharePoint site hosted by Battelle was selected to host the electronic CM file storage system. The NRC's Battelle SharePoint site was selected for the xLPR electronic file storage system mainly because it was an already established and funded system that could be exploited to meet the CM needs of the xLPR pilot study. SharePoint is primarily for document collaboration on a central server or server farm and is accessed by users either through a compatible web browser or directly via Microsoft® Office. However, while not ideal due to limits on file size and lack of desirable file level controls, the SharePoint site was sufficient to meet the needs of the SNL developed CM process for the xLPR pilot study.

The features of SharePoint include the ability for site, directory, subdirectory, and file access level controls. The web interface allowed for collaboration amongst the multi-organizational team members across institutional and geographic boundaries. The site offers a convenient method of archiving the program guidance. SharePoint keeps a version record (change control list) for all uploaded files, ideal for traceability and rollback options.

An xLPR CM site was created on the NRC's SharePoint Site hosted by Battelle Labs (https://websps1.battelle.org/nrcnureg/home/xLPR_CM). The structure of the site is outlined as in Figure 1-2.



Configuration Management

Announcements

Version 1.0 xLPR GS Framework ! NEW 10/13/2010 3:52 PM
by SNL - Mattie, Patrick

GSxLPRv1.02 posted on 10/13/2010 will become the new controlled baseline model file. This incorporates changes to the calculation of the DPD weights (as corrected in v1.01) and calculation of the PND and Exporting files (v1.02).

Note: Controlled file...

Beta Model Development 9/29/2010 4:58 PM
by SNL - Mattie, Patrick

Currently we are wrapping up Beta Model Development Activities. Alpha model development has been completed.

Welcome! 1/11/2010 6:48 PM
by SNL - Phipps, Deborah

Welcome to the xLPR Configuration Management site.

There are several ways to navigate through the system. You may use the libraries and folders below, the links to the right, or use the links to the left hand side of this page. This area is known...

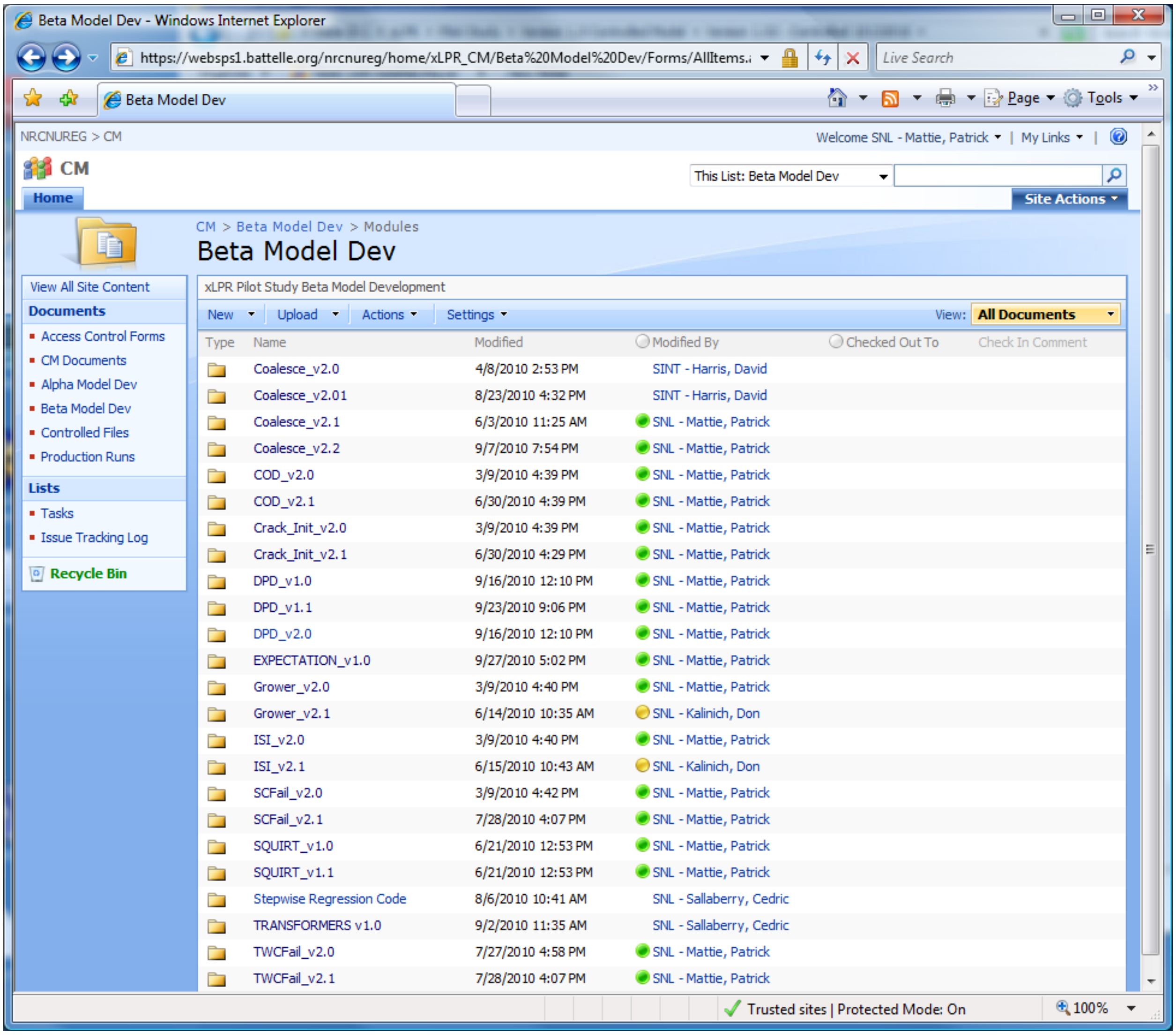
Figure 1-2. xLPR CM SharePoint Site.

The content of the site can be browsed by navigation through the web interface as displayed in Figure 1-2. The content of the subdirectories is displayed when a link is selected. The user can drill down to the area of interest. Table 1-4 lists the subdirectories with a brief description of their content.

Table 1-4. xLPR CM System SharePoint Subdirectories

Subdirectory Name		**Description**
Access Control Forms	-----	Stores the Access Control List. (Access control lists were not used for the pilot study, access to the xLPR site was controlled by the NRC project manager, subdirectory access was controlled by the SNL CM Lead)
CM Documents	CM Form Templates Desktop Guides Program Plan	Contains the CM desktop guidelines, form templates for checking and CM documentation, and xLPR Program Plan.
Alpha Model Dev	Framework Inputs Modules	Contains the model development files (source codes, framework models, inputs) and documentation for the Alpha Development version.
Beta Model Dev	Framework Inputs Modules	Contains the model development files (source codes, framework models, inputs) and documentation for the Beta Development version.
Controlled Files	-----	Contains the controlled file set (modules, input, and framework) to be used for running the xLPR model.
Production Runs	Alpha Model Runs Beta Model Runs Version 1.0	Storage area for the model results. Contains the documentation of the model changes and output used in the xLPR documentation.
Lists		
Issue Tracking Log		Issue tracking log used to record and disposition model errors, deficiencies, or needed improvements.
Tasks		Default list not used for the xLPR CM program.

In each development subdirectory, each module has a unique identifier and subdirectory name as show in Figure 1-3. During the development phases the module subdirectories were open for read/write access to the model developer and checker. After the checking was completed the module was frozen and access to the directory was limited to read/download only. All of the necessary documents are stored in the subdirectories as well as a record of the changes to the source code as listed on the change checklist. The uploaded files are automatically tracked in SharePoint so that each file has a unique version history which can be viewed as demonstrated in Figure 1-4. This traceability feature allows for roll back to an earlier version during checking and review if necessary and provides an objective record of the development and checking process. An identical process was used for the framework development.



https://websps1.battelle.org/nrcnureg/home/xLPR_CM/Beta%20Model%20Dev/Forms/AllItems.a

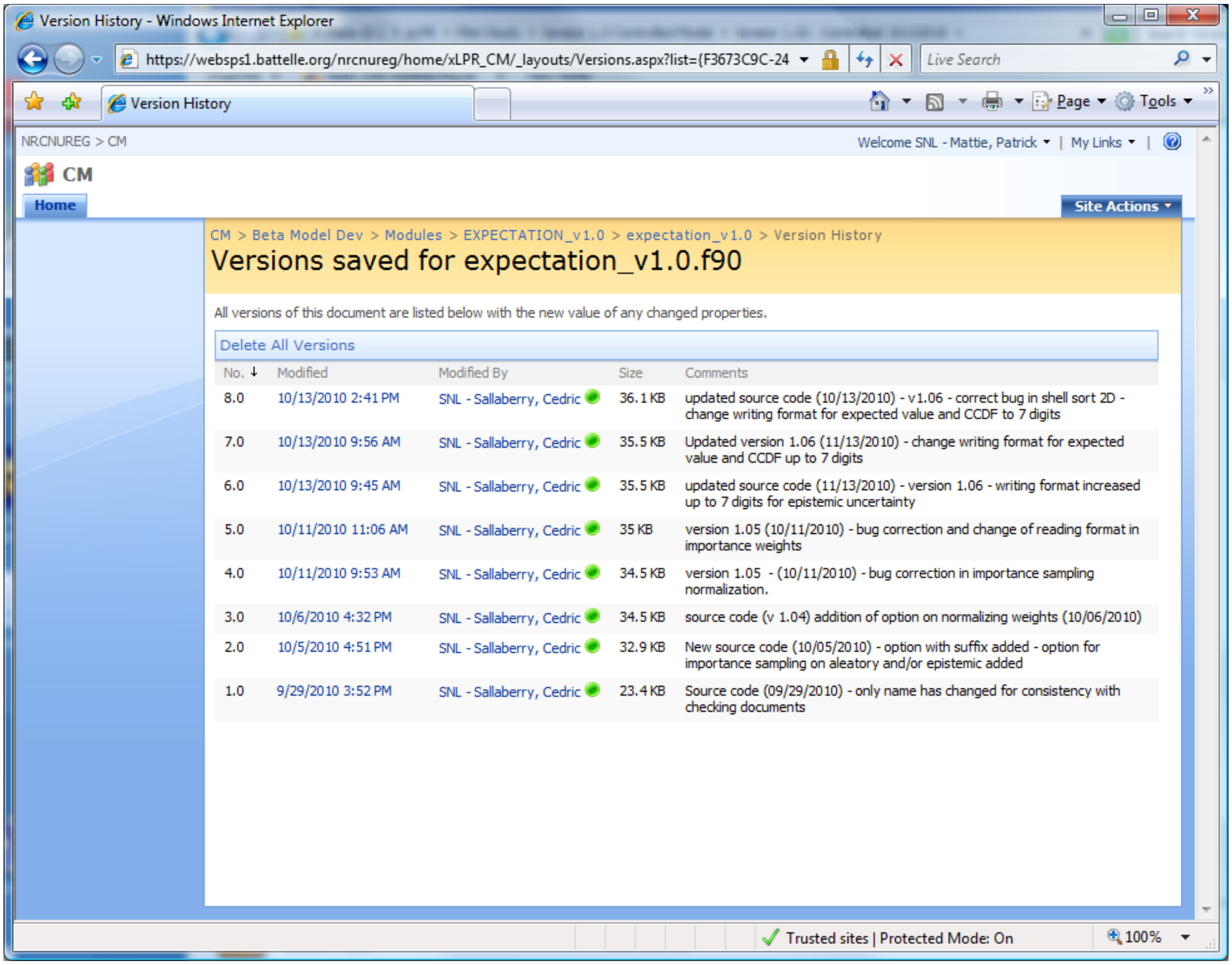
CM > Beta Model Dev > Modules

Beta Model Dev

xLPR Pilot Study Beta Model Development

Type	Name	Modified	Modified By	Checked Out To	Check In Comment
	Coalesce_v2.0	4/8/2010 2:53 PM	SINT - Harris, David		
	Coalesce_v2.01	8/23/2010 4:32 PM	SINT - Harris, David		
	Coalesce_v2.1	6/3/2010 11:25 AM	SNL - Mattie, Patrick		
	Coalesce_v2.2	9/7/2010 7:54 PM	SNL - Mattie, Patrick		
	COD_v2.0	3/9/2010 4:39 PM	SNL - Mattie, Patrick		
	COD_v2.1	6/30/2010 4:39 PM	SNL - Mattie, Patrick		
	Crack_Init_v2.0	3/9/2010 4:39 PM	SNL - Mattie, Patrick		
	Crack_Init_v2.1	6/30/2010 4:29 PM	SNL - Mattie, Patrick		
	DPD_v1.0	9/16/2010 12:10 PM	SNL - Mattie, Patrick		
	DPD_v1.1	9/23/2010 9:06 PM	SNL - Mattie, Patrick		
	DPD_v2.0	9/16/2010 12:10 PM	SNL - Mattie, Patrick		
	EXPECTATION_v1.0	9/27/2010 5:02 PM	SNL - Mattie, Patrick		
	Grower_v2.0	3/9/2010 4:40 PM	SNL - Mattie, Patrick		
	Grower_v2.1	6/14/2010 10:35 AM	SNL - Kalinich, Don		
	ISI_v2.0	3/9/2010 4:40 PM	SNL - Mattie, Patrick		
	ISI_v2.1	6/15/2010 10:43 AM	SNL - Kalinich, Don		
	SCFail_v2.0	3/9/2010 4:42 PM	SNL - Mattie, Patrick		
	SCFail_v2.1	7/28/2010 4:07 PM	SNL - Mattie, Patrick		
	SQUIRT_v1.0	6/21/2010 12:53 PM	SNL - Mattie, Patrick		
	SQUIRT_v1.1	6/21/2010 12:53 PM	SNL - Mattie, Patrick		
	Stepwise Regression Code	8/6/2010 10:41 AM	SNL - Sallaberry, Cedric		
	TRANSFORMERS v1.0	9/2/2010 11:35 AM	SNL - Sallaberry, Cedric		
	TWCFail_v2.0	7/27/2010 4:58 PM	SNL - Mattie, Patrick		
	TWCFail_v2.1	7/28/2010 4:07 PM	SNL - Mattie, Patrick		

Figure 1-3. Beta Model Development Subdirectory with Module Subdirectories listed by module identifier.



Version History - Windows Internet Explorer
https://websps1.battelle.org/nrcnureg/home/xLPR_CM/_layouts/Versions.aspx?list={F3673C9C-24
Live Search
Version History
Page
Tools
NRCNUREG > CM
Welcome SNL - Mattie, Patrick
My Links
CM
Home
Site Actions
CM > Beta Model Dev > Modules > EXPECTATION_v1.0 > expectation_v1.0 > Version History
Delete All Versions
Trusted sites | Protected Mode: On
100%

Versions saved for expectation_v1.0.f90

All versions of this document are listed below with the new value of any changed properties.

No. ↓	Modified	Modified By	Size	Comments
8.0	10/13/2010 2:41 PM	SNL - Sallaberry, Cedric	36.1 KB	updated source code (10/13/2010) - v1.06 - correct bug in shell sort 2D - change writing format for expected value and CCDF to 7 digits
7.0	10/13/2010 9:56 AM	SNL - Sallaberry, Cedric	35.5 KB	Updated version 1.06 (11/13/2010) - change writing format for expected value and CCDF up to 7 digits
6.0	10/13/2010 9:45 AM	SNL - Sallaberry, Cedric	35.5 KB	updated source code (11/13/2010) - version 1.06 - writing format increased up to 7 digits for epistemic uncertainty
5.0	10/11/2010 11:06 AM	SNL - Sallaberry, Cedric	35 KB	version 1.05 (10/11/2010) - bug correction and change of reading format in importance weights
4.0	10/11/2010 9:53 AM	SNL - Sallaberry, Cedric	34.5 KB	version 1.05 - (10/11/2010) - bug correction in importance sampling normalization.
3.0	10/6/2010 4:32 PM	SNL - Sallaberry, Cedric	34.5 KB	source code (v 1.04) addition of option on normalizing weights (10/06/2010)
2.0	10/5/2010 4:51 PM	SNL - Sallaberry, Cedric	32.9 KB	New source code (10/05/2010) - option with suffix added - option for importance sampling on aleatory and/or epistemic added
1.0	9/29/2010 3:52 PM	SNL - Sallaberry, Cedric	23.4 KB	Source code (09/29/2010) - only name has changed for consistency with checking documents

Figure 1-4. Display of version history for Expectation_v.1.0 source code.

For production runs (i.e., analyses conducted for evaluation and incorporation in xLPR project documentation), a set of controlled files is stored on the SharePoint site, each with a unique identifier. For each model run, the model files were downloaded from the controlled files subdirectory to ensure that the appropriate files were used in the analysis. The results of the xLPR pilot study are uploaded and archived on the SharePoint site with documentation and unique identifiers, in order to provide a traceable and transparent record to link the results to the xLPR model version. The xLPR CM SharePoint site meets the requirements of the CM process as outlined in Section 1.5.2.

1.6 Treatment of Uncertainty

In the design and implementation of analyses for complex systems, it is useful to distinguish between two types of uncertainty: aleatory uncertainty and epistemic uncertainty [4-16].

Aleatory uncertainty arises from an inherent randomness in the properties or behavior of the system under study. For example, the weather conditions at the time of a reactor accident are inherently random with respect to our ability to predict the future. Other potential examples include the variability in the properties of a population of system components and the variability in the possible future environmental conditions that a system component could possibly be exposed to. Alternative designations for aleatory uncertainty include variability, stochastic, irreducible and type A.

Epistemic uncertainty derives from a lack of knowledge about the appropriate value to use for a quantity that is assumed to have a fixed value in the context of a particular analysis. For example, the pressure at which a given reactor containment would fail for a specified set of pressurization conditions is fixed but not amenable to being unambiguously defined. Other possible examples include minimum voltage required for the operation of a system and the maximum temperature that a system can withstand before failing. Alternative designations for epistemic uncertainty include state of knowledge, subjective, reducible and type B.

The appropriate separation of aleatory and epistemic uncertainty is an important component of the design and computational implementation of an analysis of a complex system and also of the decisions that are made on the basis of this analysis. This point can be made with a simple notional 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 there is a probability of 0.99 that all components of this type will operate properly and a probability of 0.01 that no components of this type will operate properly. 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 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. The posing and answering of Questions (Q1)-(Q3) gives rise to what is often referred to as the Kaplan/Garrick ordered triple representation for risk [15].

The formal role of aleatory and epistemic uncertainty in the analysis of nuclear power plants and, more specifically, in pipe failures is described in Appendix D.1. The need to separate and analyze the effects of both aleatory uncertainty and epistemic uncertainty is an important determinant of the overall structure software for the analysis of xLPR events.

The analysis of epistemic uncertainty has two subcomponents: uncertainty analysis and sensitivity analysis, where uncertainty analysis involves the determination of the uncertainty in analysis outcomes that results from epistemic uncertainty in analysis inputs and sensitivity analysis involves the determination of the contributions of the epistemic uncertainty in individual analysis inputs to the uncertainty in analysis results [17-21]. Procedures for uncertainty and sensitivity analysis are discussed in Appendix D.2. The performance of uncertainty and sensitivity analyses is an essential part of the analysis of any complex system. Consistent with this importance, the software under development for the analysis of xLPR events incorporates the capability to perform sampling-based uncertainty and sensitivity analyses.

2. Pilot Study Problem Description

This section describes the pilot study problem for which the xLPR Version 1.0 code was constructed to solve. The pilot study is a proof-of-concept effort to develop an assessment tool for DM pressurizer surge nozzle welds, for which a considerable amount of publicly available information exists. The analytical output of the pilot study is a probabilistic assessment of pressurizer surge nozzle DM weld leakage and rupture. The pilot study is intended to provide relative, order-of-magnitude estimates of piping rupture probabilities; the pilot study analyses were used to identify areas requiring more focused attention in a longer-term study. Following the pilot study, a more detailed long-term study will be completed to generalize the analysis procedures to develop a software system which can be used to evaluate all primary system piping.

The surge nozzle problem is summarized in Section 2.1. The modular architecture, i.e., the division of the problem into logical parts, is described in Section 2.2. The input data and selected uncertain parameters are presented in Section 2.3. The problem description provided in this section is the basis of the framework model described in Section 3.

2.1 Surge Nozzle Problem

The pilot study addresses the specific issue of accessing the probability of rupture of DM welds for a pressurizer surge nozzle degraded by PWSCC. Inspection and leak detection were evaluated and the crack was assumed to be fully mitigated if detected. The resulting reduction in the rupture probabilities was calculated.

The location is a single weld, or a section of pipe, but only includes one circumferential plane. The geometry and material properties are limited to those of a typical surge nozzle. The xLPR framework is specifically focused on a pressurizer surge nozzle DM weld that is susceptible to PWSCC. It is assumed that the major pipe geometric features, i.e., diameter and wall thickness, are fixed. A cross-sectional-view schematic of the surge nozzle geometry is given in Figure 2-1. From MRP-216[40], a survey of nine operating power plants suggests that surge nozzle pipe is NPS 14 with a 15-inch outer diameter at the DM weld. The wall thickness at the weld is typically 1.58 inches. These values were used in the xLPR pilot study.

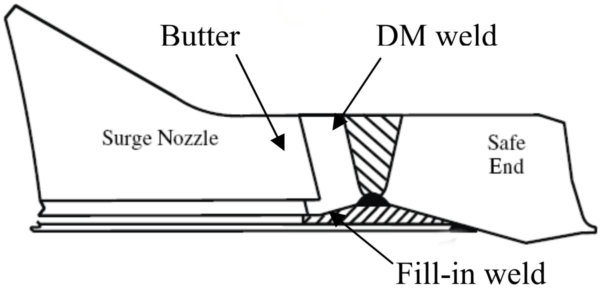


Figure 2-1. Surge nozzle geometry schematic.

The problem description including the physical processes included in the pilot study model was defined by the NRC program lead and a detailed description used for the Alpha Version is included in Appendix B. The final Version 1.0 has been modified from the initial Alpha Version and is described in Section 2.2.

The analytical output of the pilot study code includes a probabilistic assessment of surge nozzle DM welds including: (i) probability of leakage, (ii) probability of rupture, (iii) evaluation of the effects of inspection, (iv) effects of leak detection, (v) leak rates, (vi) probability of 1 inch and 3 inch crack opening area, and (vii) other relevant metrics.

2.2 Problem Architecture

A prototype xLPR model was constructed leveraging existing fracture mechanics models and a commercial software framework. The pilot study problem includes a number of modules that were selected for crack initiation and growth and to simulate the physical and mechanical processes to evaluate the probability of rupture. The program architecture requirements for the modular-based code are shown in Figure 2-2. A discussion of how this proposed process was included in Version 1.0 is outlined in this section. A summary description of each module shown in Figure 2-2 is included below. The modules were developed by the xLPR Models Task Group and detailed documentation of the technical basis and conceptual model descriptions are included in the xLPR Models/Input Task Groups report.

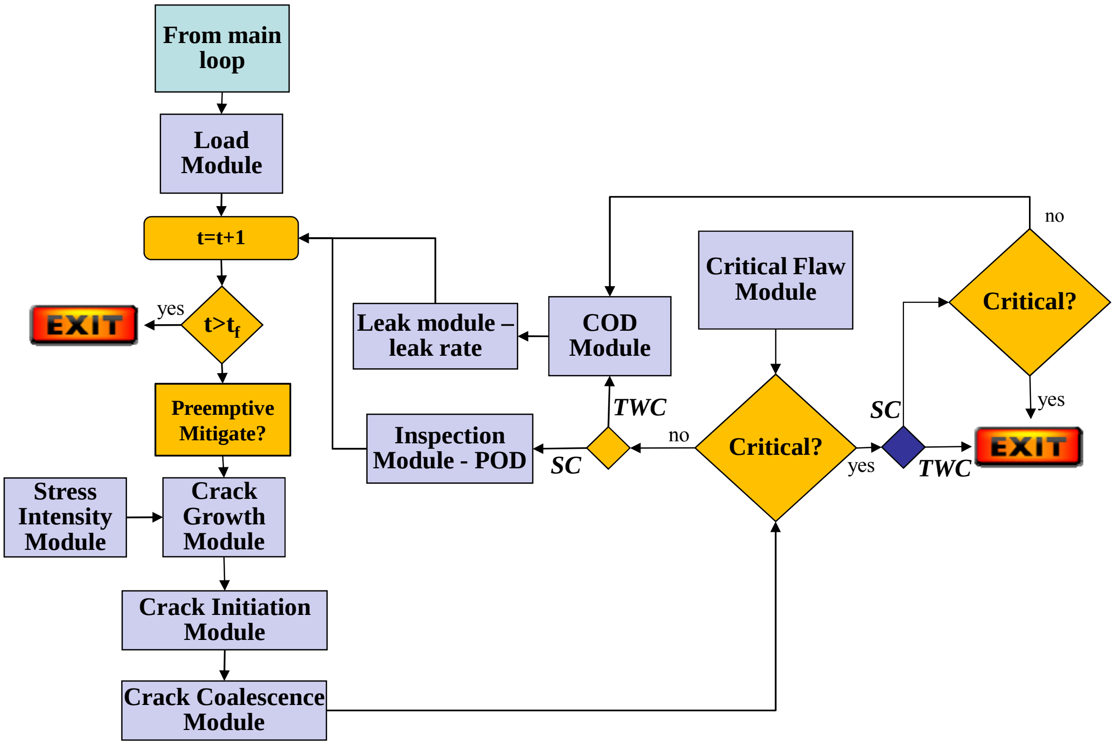


Figure 2-2. xLPR Process (time loop) details.

Loads: The loads on the pipe weldment are defined before beginning the analysis. Loads contributing to stresses normal to circumferential cracks are considered. Since the analysis concentrates on PWSCC, only sustained loads are needed (i.e. fatigue cycling is not considered). A set of tensile and bending loads are obtained from an average of values for selected operating plants [33]. The sustained loads from normal thermal, deadweight and thermal stratification are considered. The safe shutdown earthquake loads are also defined. The above loads are considered to be deterministic and constant through-wall. The as-welded residual stresses are also considered, and are taken to be axisymmetric. A self-equilibrating through-wall distribution is defined in terms of a third-order polynomial, with random parameters to describe the scatter of residual stresses.

Initiation Module: The initiation module defines the number of cracks that initiate, their respective times of initiation, and locations for subsequent crack growth analysis. All crack initiations are defined up front, that is they are all scheduled prior to beginning any crack growth analysis. The initiation times are determined from a probabilistic model that includes the effects of temperature and stress. Additionally, the module places the initiated cracks at specific locations on the weld perimeter. An expression is available that provides the initiation time within a weld segment as a function of stress and temperature. This expression has some randomness to describe scatter observed in the data used as the basis for the model. The weldment is composed of 19 segments of equal length. At a given time, the stress and temperature are constant within a segment. Only one crack can initiate within each segment. The depth and length of initiated cracks are random variables. For initiation probability two approaches were implemented:

> Direct: Initiation time given as explicit function of stress and temperature, with randomness in parameters in the function.
>
> Weibull: The initiation time is taken to be Weibull distributed with a slope of 3 and a scale parameter that depends on stress and temperature.

The sizes (depth and length) of the initiated cracks are sampled from user-defined statistical distributions. The crack is assumed to be surface-connected and semi-elliptical. The sampling size of initiated cracks is performed along with other sampling within the xLPR Model Framework, and is therefore not performed within the initiation module.

Crack Coalescence Module: Initiated cracks can coalesce, even before crack growth occurs. Although coalescence is more likely once the cracks are growing. The possibility of coalescence is checked once after a crack is initiated and once during each time step after crack growth has occurred. For semi-elliptical surface cracks, when the distance between the surface cracks becomes less than half the deepest surface crack depth, the cracks coalesce. The depth of the new crack is equal to the deepest surface crack and the length is equal to the sum of the lengths of each crack plus the distance between them. Two through-wall cracks coalesce when the crack tips touch. The length of the new crack is equal to the sum of the lengths of each crack. There is also a possibility that a through-wall crack may interact with a surface crack, although a complex crack may be formed, it is assumed that when a through-wall crack and a surface crack interact, a through-wall crack is formed with the length of the through-wall crack equal to the sum of the lengths of each crack. When two cracks coalesce, the crack with the higher crack ID number is considered as coalesced (absorbed crack) and the information (new crack size, new crack location and the new crack type) is passed to the crack with the lower crack ID number (parent crack). The absorbed crack has a crack type definition equal to the parent crack ID number. The crack size of the absorbed crack will be set to zero.

Stress Intensity Factors: Stress intensity factor solutions for part-through circumferential semi-elliptical cracks [30] and straight-fronted through-wall circumferential cracks [31] are included. The part-through solutions solve the through-wall stress distributions using a fourth order polynomial. The stresses vary with depth. Local stress intensity factors for the deepest point and surface point are provided. For TWCs, tension and through-thickness and global bending stresses are included.

Crack Growth: The (coalesced) initiated cracks are grown time step by time step. The surface- and depth-direction growth is controlled by the corresponding stress intensity factors. The Crack Growth Module uses a PWSCC mechanism. The crack growth module uses a model presented in MRP-115 [32] is a function of temperature and stress intensity. In addition to the model from MRP-115 [32], a model to account for the effect of hydrogen concentration was included from MRP-263 [43]. It is generally accepted that increasing the hydrogen concentration beyond the Ni/NiO equilibrium reduces the crack growth rate. The extent of the reduction depends on the initial and final hydrogen concentrations, the alloy (Alloy 182 or 82), and the temperature

Inspection: The influence of inspection is treated through the probability of non-detection (PND). The PND is a function of crack size that is estimated from test data; therefore, there is uncertainty in the PND. At each inspection, the PND for each of the cracks is recorded, and the influence of inspection on leak probabilities (leaks of various sizes) is evaluated during post-processing.

Crack Stability: The stability of part-through cracks is based on net-section collapse for combined tension and bending loading [34] and includes both a constant depth and a semi-elliptical SC net-section collapse solution. The stability of TWCs is based on tearing instability that uses both an ideal crack net-section collapse and elastic-plastic formulation for evaluation of the applied J-integral that is based on a reduced thickness analogy to estimate the compliance of cracked elastic-plastic tubes subject to tension and bending [35].

Transition from part-through to TWCs is handled by determining the TWC length where the cracked area is equal to the part-TWC area at through-wall penetration. Once a TWC becomes unstable, a double-ended break (rupture) is assumed to occur. Such an event is recorded, and the program exits the time loop.

Crack Opening Displacement: The crack opening displacement (COD) for TWC is estimated using a modification to the General Electric (GE)/EPRI solutions wherein the individual original GE/EPRI bending and tension solutions are blended to yield a combined tension and bending solution [36]. Load relaxation due to the presence of the crack is not considered. The COD and crack length define the crack opening area (for evaluation of leak rates) assuming the crack opening to be rectangular.

Leak Rates: Leak rates for straight-fronted TWC are evaluated based on the Seepage Quantification of Upsets in Reactor Tubes (SQUIRT) software [37], which, in turn, is based on the Henry-Fauske model. Pressure drops due to entrance effects, friction, phase change (liquid to gas), and bends and protrusions are considered. If the leak rate for a TWC exceeds some specified limit, then it is recorded, but the time loop continues until the pipe ruptures. The effects of leak detection are analyzed during post-processing.

After the initial conditions are defined, the time loop, shown in Figure 2-2, begins. For each time increment, the code needs to check whether the analysis is beyond the predefined time period (60 years) for the analysis. If it is, the time loop is exited, if not, it will continue.

Pre-emptive mitigation (described in Program Description, Appendix B) was not included for Version 1.0. Instead mitigation is assumed to occur only if a crack is discovered with inspection of leak detection.

Next, if the crack initiation module dictates that a crack initiates in a time step, a single, surface breaking crack will be placed in the model with the appropriate size and location. Note that, in the xLPR process time loop, the coalescence module precedes the growth module to avoid placement of new cracks in an existing crack location.

The crack growth module is then used to calculate the crack growth increment for any existing cracks using the module described above. Within the crack growth module, the instantaneous loads, including the welding residual stresses (WRS), are used along with the crack and pipe geometry to calculate stress intensity factors. 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 SC has reached 100% through wall, it transitions to a TWC. In addition, a second check is made to determine if any SCs or TWCs coalesce. If they do, they will be combined.

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

If a TWC is not critical, the leakage module is used to determine leakage rates. The calculated leak rates are evaluated during post processing to determine if they are greater than pre-defined loss of coolant accident (LOCA) rates, the leak rate and crack opening area are recorded. Likewise, during post processing a series of user defined inspections are evaluated against the calculated PND for SCs. If the crack is detected it is assumed that all of the cracks in the weld are mitigated and the remainder of the simulation is terminated.

After the leak rate and PND have been calculated the time is then incremented and the analysis continues until the end of the simulation time or a rupture occurs.

2.3 Parameters and Input

It is assumed that the major pipe geometry features, i.e., diameter and wall thickness, are fixed. Data from MRP-216 [40], which includes a survey of nine operating power plants, defines the properties for surge nozzle pipe as NPS 14 with a 15-inch outer diameter at the DM weld. The material properties and load data is from MRP-216 and provided by the xLPR Inputs Task Group. The geometry, loads and material properties are listed in Appendix C.

Uncertainty in the input includes physical, material and environmental properties (e.g., temperature, pressure, toughness, weld factor, etc.) as well as module parameters. The xLPR code parameters derive in part from the models used in the modules. Each parameter has a distribution and has been identified as either aleatory or epistemic (the treatment of uncertainty is presented in Section 1.6). The probability of rupture comes from randomness in the behavior and/or properties of the system under study (a piping system in the context of xLPR). This uncertainty is aleatory and is what leads to the probability of rupture. The uncertainty in the probability of rupture is epistemic and derives from a lack of knowledge with respect to quantities used in the calculation of probability of rupture that are assumed to have fixed but imprecisely known values. Such quantities could be parameters that are used in distributions that characterize aleatory uncertainties or physical properties that that are used in modeling the behavior of the system under study. Most analyses of complex systems involve epistemically uncertain quantities of both types. The evaluation of the uncertain parameters and characterization of the parameters as either aleatory or epistemic was conducted by the xLPR Models task group and their recommendations are listed in Table 2-1. The parameters, distributions, and uncertainty types used in xLPR code version 1.0 are listed in Appendix C.

Table 2-1. Characterization of Uncertain Parameters for Version 1.0

Epistemic	Aleatory
Heat-to-Heat Variability (A,B1)	Initial Half Crack Length
Within Heat Variability (AmuWH, BumWH)	Initial Crack Depth
Random Crack Placement (RandULoc)	Q over R
Random Crack Initiation Time (RandU3)	Peak to Valley ratio (P)
Axial Stress for WRS (sig0)	Characteristic width of crack growth rate curve (C)
Location where WRS=0 (Xc)	Random number for probability of detection (POD)
WRS in the outer diameter	Pipe Yield Stress
Weld Factor (F_weld)	Pipe Ultimate Stress
Random Error for POD (Beta1, Beta 2)	Ramberg-Osgood Fit parameters
Pressure	J-Resistance Coefficient and Exponent
Temperature	Yield Strength Parameters
	Yield Stress Parameters

3. Description of Commercial Software Implementation of the xLPR Model Framework

The xLPR framework model was constructed using the commercial software GoldSim Pro and a player file, compatible with the free GoldSim Player software was created, Figure 3-1. GoldSim Pro is essentially both a model developer's version and the simulation software. GoldSim Player allows you 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 it. GoldSim is a flexible platform for visualizing and dynamically simulating complex systems using a top-down hierarchical approach. The GUI facilitates an intuitive organized structure that automatically constructs an influence diagram of the model system. The generic framework software is highly graphical and extensible, able to quantitatively represent the uncertainty inherent in diverse complex systems, and allows users to create compelling models that are easy to communicate and explain to diverse audiences.

Since this is a commercial software package, the description of the xLPR model framework model implementation will often refer to the detailed GoldSim Pro Software User's Guide [38 & 39]. One asset of the software is that models are self-documenting by default, since the code automatically generates graphical influence links between the subsystems and graphic elements. The software also has the ability to include author generated hypertext references, graphics and embedded descriptive text that have been used in this application to describe each subsystem. The figures used to illustrate the framework model description in Section 3.1 are for the most part screen captures of the model framework (Figure 3-1). It is recommended that the reader have a copy of the GoldSim Pro User's Guide and a copy of the xLPR framework model open while reading through the remainder of this description. Navigating through the model is fairly simple with a little practice. For a more detailed description of setting up and running the code, the reader is directed to the xLPR Framework Model User's Guide [41].

Section 3.1 describes the treatment of uncertainty, xLPR commercial software framework structure, and model scenarios. Section 3.2 discusses the post processing methodology for inspection and leak detection. Section 3.3 presents the format of for the presentation of the results of the analyses conducted in this study that are discussed in Section 5. And Section 3.4 describes the presentation of uncertainty and sensitivity analysis results that are included in Section 5.2.

GoldSim Player - xLPR Framework Model - Version 1.02 (GSxLPRv1.02_M02.gsp)

Last Modified: 10/13/2010
xLPR Version 1.02 (Pilot Study Controlled Version)

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Figure 3-1. Title Screen of the Framework Model (GSxLPRv1.02).

3.1 Description of the Commercial Framework Architecture

The construct of the xLPR framework model, displayed in Figure 3-2, follows the inherent capabilities of the commercial software in which models are developed in a hierarchical and modular manner by creating individual subsystems which are linked together as described in Section 3.1.2. The xLPR pilot study framework model as outlined in the xLPR program plan (Section 2.0 & Appendix B) was constructed using the built-in features in GoldSim Pro, Version 10.11. Each xLPR module was converted to a DLL that is called directly by the GoldSim software through built-in elements that use a standard interface as described in Section 3.1.2. New modules are added or old modules are replaced by simply wrapping the original module executable source code in the DLL wrapper code, and compiling as a DLL (see Appendix C in *ref*. [38]). The modules input and output array interface is defined in the framework model file; so as long as the new module respects the original array definitions, a new or updated version can be used without modification to the model framework. If the input or output arrays change, the model file needs to be modified to accommodate this by increasing or decreasing the expected array size (a feature not available in the GoldSim Player software).

Each subsystem (module) was constructed independently in a separate GoldSim model file and was imported into the main framework model by copy and pasting the graphical elements from the stand alone model file. This allowed each module subsystem to be independently tested and implemented as a standalone model without sacrificing model development time. The incremental development and verification processes are described in Sections 1.3 and 4 and is documented on the SharePoint xLPR CM site.

Uncertainty in processes and parameters are represented by specifying probability distributions for model inputs. GoldSim uses Monte Carlo simulation with LHS to propagate uncertainty through the model. The xLPR framework module uses a Microsoft® Excel spreadsheet file as an input deck, read through a built-in software interface as described in Section 3.1.3. The GoldSim software has a dashboard interface option [38, 39] that facilitates ease of defining the inputs (Section 3.1.3), model options and execution (Section 3.1.5), run status (Section 3.1.6), result display and navigation of the model (Section 3.1.4) using the GoldSim Player model. In addition the dashboard interface enables a two step process to be used for model simulation. In the first step, the framework is used to launch a modeling scenario (Sections 3.1.5 & 3.1.6). In the second step, the framework file is used to export the results and run the post-processing tools (Section 3.2). The two-step process provides for maximum adaptability and flexibility, while increasing efficiency. Once a modeling case has been run and the distribution of responses have been generated, the user can evaluate the effects of an unlimited number of combinations of inspection times and leak detection limits, using the post-processing tools (Section 3.2) without having to re-run the model case each time.

The framework model file is a GoldSim file. It contains GoldSim elements. Each element represents a building block of the model. The elements and their interconnections make-up the framework model file portion of the xLPR framework. As depicted in Figure 3-2 there are specific elements in the framework model file that link directly to the Fortran modules and input data. There are also elements that allow results from the model to be exported to American Standard Code for Information Interchange (ASCII) text files. Chapter 2 of the GoldSim User's Guide [38] provides a general explanation of building GoldSim models, using elements, and element interconnections. The following sections describe the various subsystems within the framework model in context to the GoldSim software features (elements, SubModels, containers, links, etc).

The framework was developed for adaptability and flexibility by using a common interface between the framework and modules. The framework was developed using the standard commercial software capabilities included with the GoldSim Pro software, which have been developed and optimized over many years of industry usage. The xLPR commercial framework model is coupled to a set of modules and an input deck. The complete file set embodies Version 1.0 of the xLPR model.

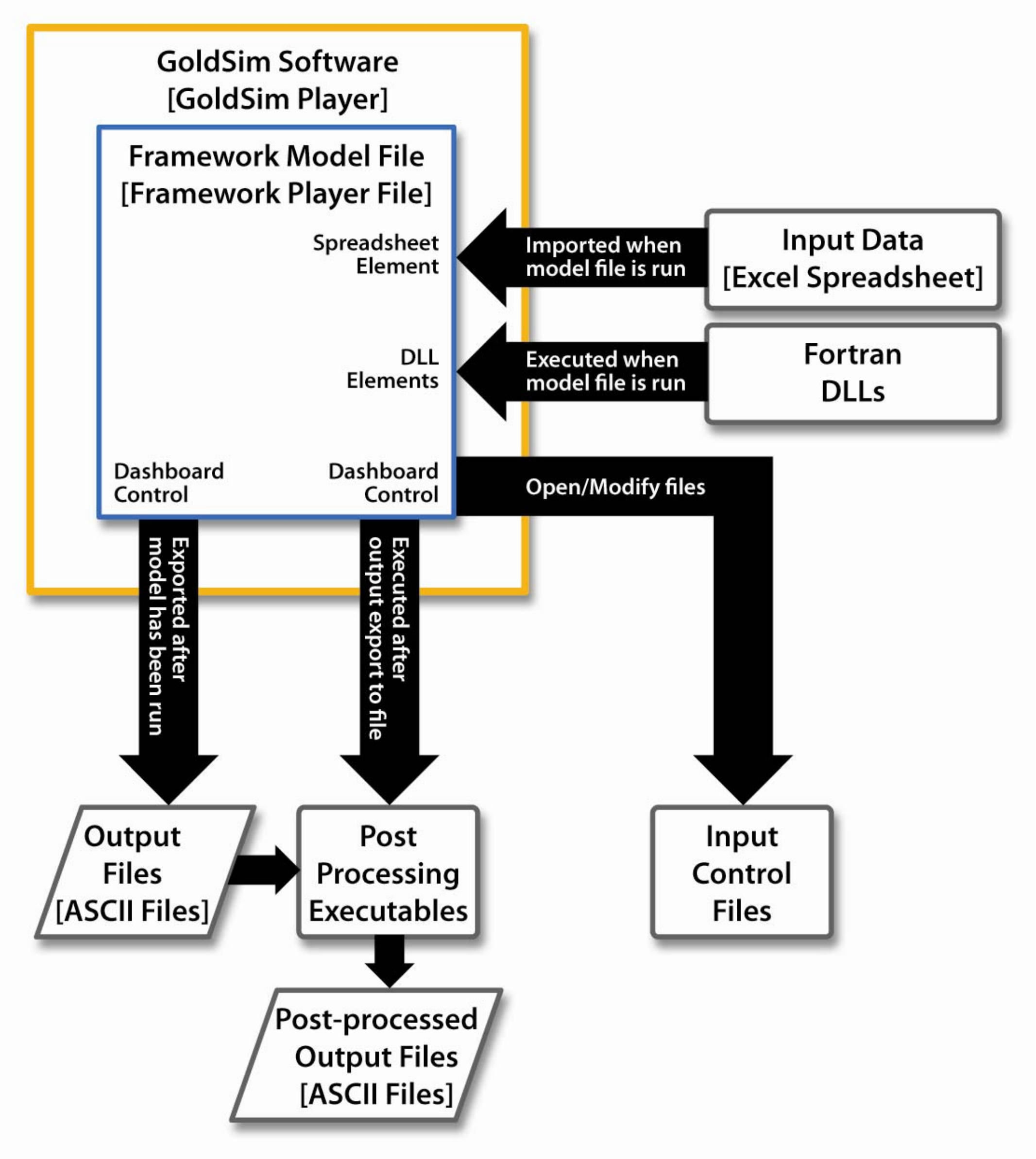


Figure 3-2. xLPR Model Framework.

3.1.1 Looping (Separation of Aleatory and Epistemic Uncertainties)

The xLPR pilot study is focused on the development of a software tool to predict the probability of rupture for a pressurizer surge nozzle DM weld. The xLPR framework model uses a loop approach consistent with the treatment of uncertainty presented in Section 1.6. The performance of uncertainty and sensitivity analyses is an essential part of the xLPR analysis. Consistent with this importance, the model development for the analysis of xLPR events incorporated 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 iteration, 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 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 (Section 3.4).

The GoldSim software was specifically designed to run Monte Carlo simulations. As such the software features include predefined stochastic distributions and Monte Carlo sampling capability which includes a Latin Hypercube Sampling (LHS) option. Effectively the nested loop approach requires embedding an entire GoldSim model within another GoldSim model. GoldSim provides a special element called a SubModel to facilitate this. A SubModel superficially looks like a Container, and conceptually shares some aspects with Containers. However, the functionality of a SubModel is quite different from a Container. Whereas a Container is simply a grouping of elements within a model, a SubModel is a completely separate "inner" model within an "outer" model. That is, it has its own simulation settings (time and Monte Carlo options) that are independent of the simulation settings of the outer GoldSim model within which the SubModel element is placed. Hence, when a SubModel element (i.e., the inner model) is triggered to do a calculation by the outer model, it runs a complete simulation. The SubModel has a special interface which allows the results from the embedded model to be transferred back to the outer model.

Two approaches could have been taken to implement the nested loop approach for the xLPR framework model. The first approach would be to embed the aleatory parameters and time-loop within an epistemic outer loop. Therefore for each outer loop realization the embedded SubModel would be run N_a times and pass back the distribution of the results to the outer model. The outer model would have N_E realizations, yet a total of (N_e x N_a) realizations would have been run. In the second approach the time loop would be contained in the outer model with the two separate epistemic and aleatory SubModels run at time=0 in order to generate the input sample set for each outer model realization. The outer model therefore needs to have a total number of realizations equal to (N_e x N_a). The first approach has advantages in terms of an intuitive design, however the functional outcome of embedding a complete GoldSim model of the complexity of the time-loop, greatly reduces the transparency of the calculations within the model calculations when the Player file is used. Therefore the second approach was selected and is described below. Future versions of the xLPR framework may want to use the first approach. To test the feasibility a prototype version of the first approach was created and is available for a future study.

To calculate the sampling sequence needed to correlate the aleatory and epistemic SubModels with the outer model (time loop) the following logic is incorporated into the model framework.

$$N_G = N_e \times N_{a1} \times N_{a2}$$

$$X_e = max\left(1, trunc\left(\frac{N_G - 1}{N_{a1} \times N_{a2}}\right) + 1\right)$$

$$X_{a1} = max\left(1, mod\left(trunc\left(\frac{N_G - 1}{N_{a2}}\right), N_{a1}\right) + 1\right)$$

$$X_{a2} = mod(N_G - 1, N_{a2}) + 1$$

where:

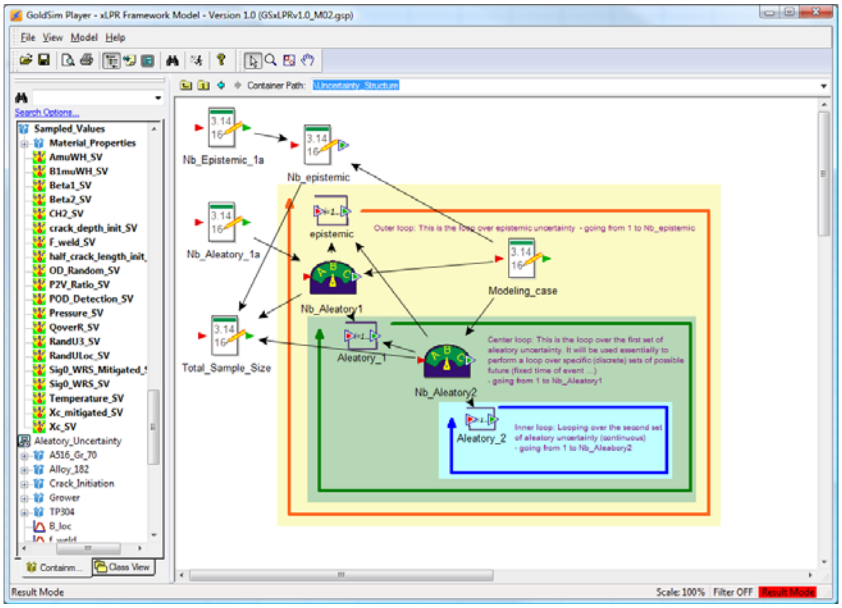
N_G	=	total number of GoldSim realizations
N_e	=	total number of epistemic loop realizations
N_{a1}	=	total number of aleatory loop 1 realizations
N_{a2}	=	total number of aleatory loop 2 realizations
X_e	=	the ith epistemic loop realization
X_{a1}	=	the ith aleatory loop 1 realization
X_{a2}	=	the ith aleatory loop 2 realization
X_G	=	the ith GoldSim realization

This looping structure is defined for one epistemic loop (outer loop) and two aleatory loops (center loop and inner loop). For the xLPR pilot study only the epistemic loop (outer loop) and 1st aleatory loop (inner loop) are used; the 2nd aleatory loop is not used[1] and was included as additional functionality for future version of the xLPR framework. Its purpose is essentially algorithmic as it allows using technique than sampling (e.g., quadrature) in order to integrate results over aleatory uncertainty. The default xLPR framework realization settings for the probabilistic case are:

$$N_e = 1000$$
$$N_{a1} = 50$$
$$N_{a2} = 1$$
$$N_G = 1000 \times 50 \times 1 = 10{,}000$$

The Uncertainty_Structure subsystem container contains the calculations used in the framework for the separation of epistemic and aleatory uncertainties (Figure 3-3). The elements and influence lines are graphically depicted in the framework model. The Main Dashboard is used to define the number of epistemic and aleatory realizations for each SubModel and the total number of realizations for the outer time loop model (Figure 3-4).

[1] The 2nd loop is “deactivated” by setting its total number of realizations equal to 1.



GoldSim Player - xLPR Framework Model - Version 1.0 (GSxLPRv1.0_M02.gsp)

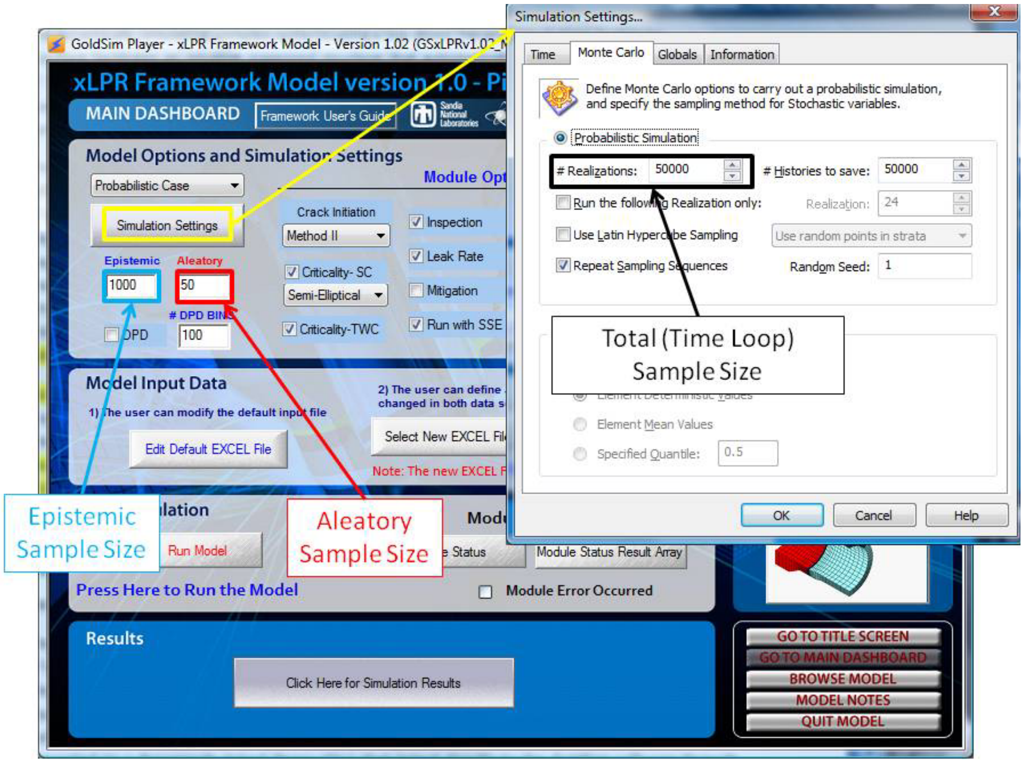
Container Path: \Uncertainty_Structure

Outer loop: This is the loop over epistemic uncertainty - going from 1 to Nb_epistemic

Center loop: This is the loop over the first set of aleatory uncertainty. It will be used essentially to perform a loop over specific (discrete) sets of possible future (fixed time of event ...) - going from 1 to Nb_Aleatory1

Inner loop: Looping over the second set of aleatory uncertainty (continuous) - going from 1 to Nb_Aleabory2

Figure 3-3. Contents in the Uncertainty_Structure Container.



GoldSim Player - xLPR Framework Model - Version 1.02 (GSxLPRv1.02_M

xLPR Framework Model version 1.0 - Pi

MAIN DASHBOARD

Model Options and Simulation Settings

Probabilistic Case

Epistemic: 1000

Aleatory: 50

DPD BINS: 100

☐ DPD

Crack Initiation: Method II

☑ Criticality- SC: Semi-Elliptical

☑ Criticality-TWC

☑ Inspection

☑ Leak Rate

☐ Mitigation

☑ Run with SSE

Model Input Data

1) The user can modify the default input file

2) The user can define ... changed in both data s...

Note: The new EXCEL F...

Press Here to Run the Model

☐ Module Error Occurred

Results

Simulation Settings...

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

● Probabilistic Simulation

Realizations: 50000

Histories to save: 50000

☐ Run the following Realization only: Realization: 24

☐ Use Latin Hypercube Sampling — Use random points in strata

☑ Repeat Sampling Sequences

Random Seed: 1

○ Element Mean Values

○ Specified Quantile: 0.5

Total (Time Loop) Sample Size

Epistemic Sample Size

Aleatory Sample Size

Figure 3-4. Defining the Epistemic, Aleatory, and Time Loop realizations.

The framework model also includes an optional sampling module, Discrete Probability Distribution (DPD) which has the capability of importance sampling. When the DPD module is activated, the uncertain parameters are combined and are no longer separated into aleatory and epistemic categories as done with the Monte Carlo approach. It is possible to run two instances of the DPD module, once for Aleatory and once for Epistemic, however, that capability was not deemed necessary for demonstrating DPD in the pilot study and therefore not included in the Version 1.0 model framework. The uncertainty sampling when the DPD module is activated is controlled by two parameters. The number of DPD bins determines how many samples are generated by the DPD module. The number of realizations in outer model time loop determines how many samples are pulled from the DPD generated sample matrix, with each realization using a randomly-sampled DPD sample set. The DPD module is used for importance sampling and to demonstrate the commercial software framework's ability to adapt to different probabilistic methods.

The choice of the sampling technique (e.g., Monte Carlo, LHS, and importance sampling) is not trivial and warrants some discussion. In complex systems analysis, the effect of individual input parameters as well as their conjoined influence is poorly known until the system is fully analyzed. Therefore the use of importance sampling upfront is dangerous, as the region of importance may be wrongly selected. It is better to first test with a more evenly distributed sampling method like Monte Carlo with LHS.

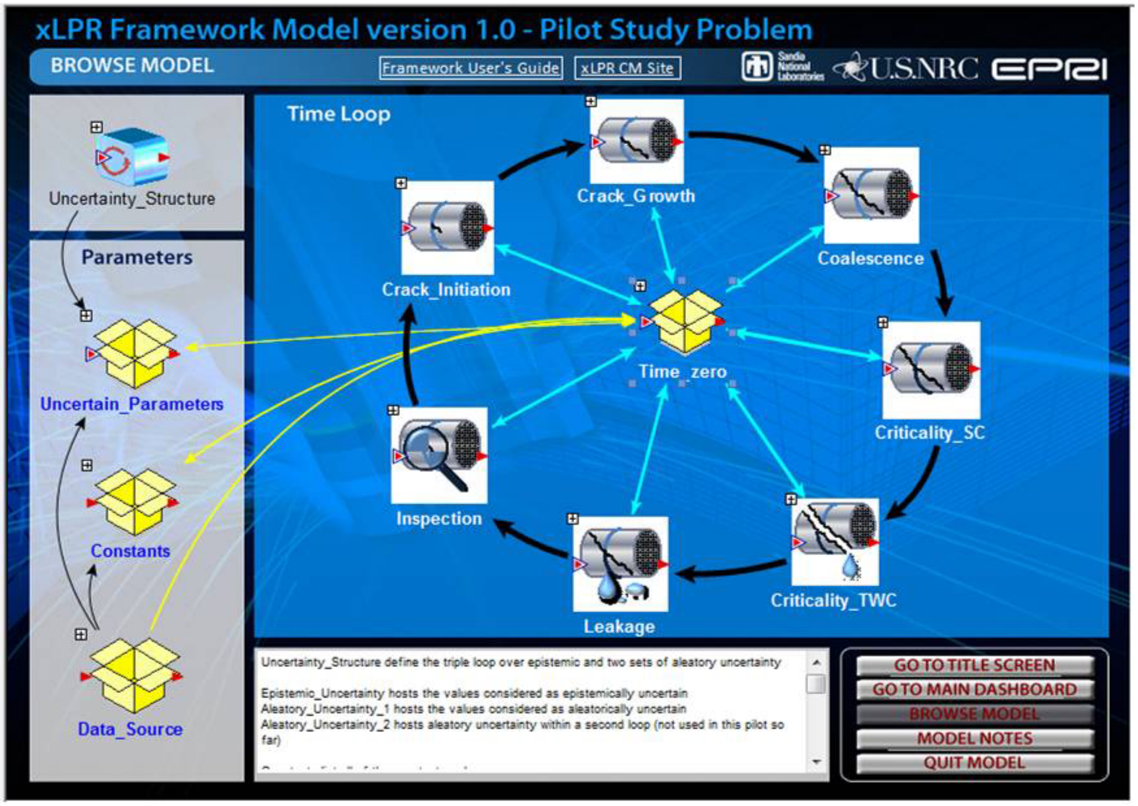
Several sampling techniques exist that cover the hypercube defined by the uncertain parameters. Some sampling techniques focus on densely sampling each parameter rather than ensuring the entire hyperspace has been sufficiently sampled (e.g., LHS). Other techniques focus on a better coverage of the hyperspace. The former techniques are more appropriate when only a few parameters have a strong influence while the latter gives better results when conjoined influence are involved.

Based on experience, dense stratification is generally more appropriate for epistemic uncertainty; therefore LHS is the method of choice. Each variable will be densely covered and any important variable would appear in the sensitivity analysis. At the same time the use of LHS would not be adequate for aleatory uncertainty. While we want to cover every possible value for epistemic parameters (which are supposed to have a true, but poorly known value), aleatory uncertainty represents randomness in a system. Since randomness is not supposed to follow any definite pattern, forcing stratification is inappropriate the aleatory parameters. For instance, if a crack occurs for a simulation between 10 and 12 years, it is possible that for another simulation, it will happen again (while a stratification will force a crack to occurs in every strata defined). Therefore, simple random sampling (SRS) is used to sample over aleatory uncertainty.

The analyses presented in Section 5, give insights on the behavior of the epistemic and aleatory parameters and a refined analysis with importance sampling in the region of interest is presented in Sections 5.3.5 and 5.3.6.

3.1.2 xLPR Time Loop Module Implementation

The time loop is implemented in the outer model as defined in Section 3.1.1. The time loop in the xLPR framework model has the same fundamental structure as defined in Section 2.0 and the xLPR Program Plan (Appendix B); depicted in Figure 3-5 as it appears in the model framework software. The time loop contains the 11 modules used in the xLPR analysis to model the physical degradation mechanisms used to assess the probability of rupture due to PWSCC for DM pressurizer surge nozzle welds. Table 3-1 lists the framework subsystem descriptions and the modules contained within each subsystem, as depicted by the software influence links displayed in Figure 3-5, in the sequence in which the subsystems are called within the time loop.



Uncertainty_Structure define the triple loop over epistemic and two sets of aleatory uncertainty

Epistemic_Uncertainty hosts the values considered as epistemically uncertain
Aleatory_Uncertainty_1 hosts the values considered as aleatorically uncertain
Aleatory_Uncertainty_2 hosts aleatory uncertainty within a second loop (not used in this pilot so far)

Figure 3-5. Time Loop Subsystem containers and influence diagram.

Table 3-1. Framework model Time Loop subsystem descriptions and module list.

Framework Subsystem	Module	Description
Time Zero	Load	Subsystem is run at time t = 0, after the Epistemic and Aleatory SubModels are run. Includes the calculations of the initial conditions for each realization of the time loop and the Load module.
Crack Initiation	Crack_init Coalescence	Subsystem contains a module that determines number of SC, initiation times, and locations. The crack_init module is run once at t = 0, after the Epistemic and Aleatory SubModels and after the Time Zero subsystem initial calculations. The coalescence module is run at t=0 (after the crack init module) and the beginning of each time step. New SC that initiate within an existing crack are immediately coalesced.
Crack Growth	Grower KSurf KTWC	Subsystem used to calculate the crack growth increment for any existing cracks. Within the crack growth subsystem, the instantaneous loads, including the WRSs, are used along with the crack and pipe geometry (from the time zero subsystem) to calculate stress intensity factors. These stress intensity factors are then used with the crack growth module to calculate the crack growth increment.
Coalescence	Coalescence	Subsystem contains the coalescence module used to check if any SC or TWC coalesce.
Criticality SC	SCFail	Subsystem contains the SC stability module. Once per time step the stability of existing SCs are tested twice, once for nominal and once for safe shutdown earthquake (SSE) loads.
Criticality TWC	TWCFail	Subsystem contains the TWC stability module. Once per time step the stability of existing TWCs are tested twice, once for nominal and once for SSE loads.
Leakage	COD Leakage	Subsystem contains the calculation of the COD and leakage rates when a TWC exists.
Inspection	ISI	Subsystem contains the inspection module which calculates the PND when SCs exist.

The time loop is run for a number of time steps (N_{TS}) per realization of the outer model. At time = 0, the aleatory and epistemic SubModels are run and a sample set is generated from the uncertain distributions; next the Time Zero subsystem is run to calculate the initial conditions; and then the time loop begins. The aleatory and epistemic SubModels and Time Zero subsystem are run only once per realization of the outer model. In the time loop, at time = 0, the crack initiation module is run once. The model has been created so that the time loop begins only when the first crack initiates. So the time step is automatically advanced to the time step in which the first crack initiates before the next module is called. The crack initiation, crack growth, and coalesce subsystems and embedded modules are called each time step when a SC or TWC exists. The remaining subsystems are conditional upon the existence of a SC or TWC. The GoldSim software allows a condition to be added to any subsystem. For instance, the Criticality TWC and Leakage subsystems are not called unless a TWC exists and Inspection and Criticality SC is only called when a SC exists. Likewise, using the interrupter element available in the GoldSim software, if a rupture occurs (critical failure of a TWC) the time loop is exited skipping the remaining time steps and the next realization is started.

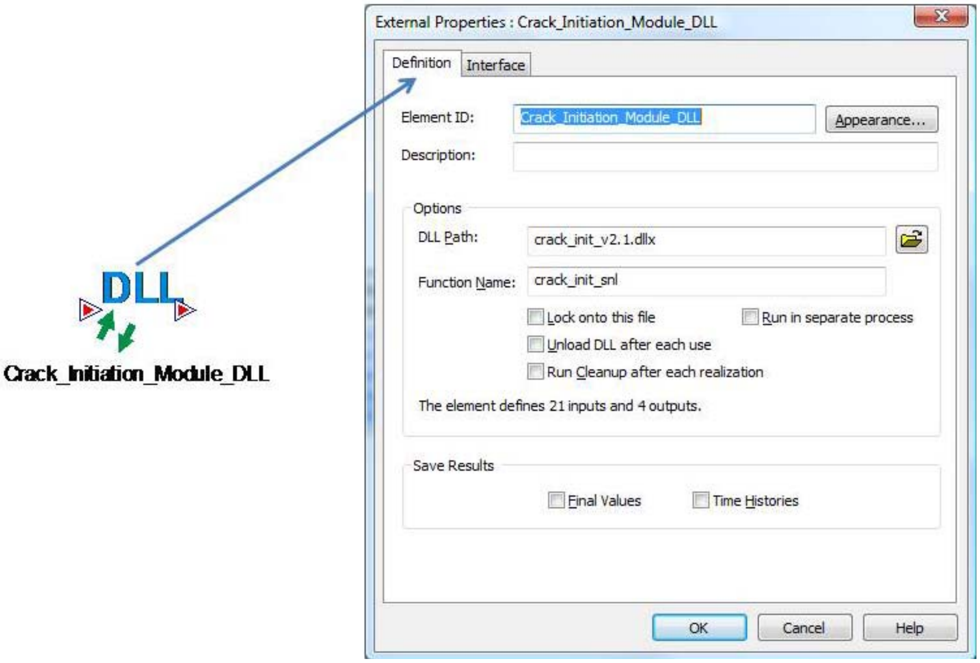
Each module is linked to the framework model using a standard DLL interface available with the GoldSim software. GoldSim was designed such that you can develop separate program modules (written in C, C++, Pascal, Fortran or other compatible programming languages) which can then be directly coupled with the main GoldSim algorithms. These user-defined modules are referred to here as external functions and the elements through which they are coupled to GoldSim are called external (DLL) elements. The external DLL element is used to pass information between the model framework and the external module. An external element, as displayed in Figure 3-6, will utilize:

- **DLL Path**: This is the name of the DLL containing the external function. The xLPR framework specifies just the file name; therefore GoldSim will look for the DLL only in the working directory containing the GoldSim model file. Modules can be easily swapped by adding a new module with the same name in the directory containing the GoldSim model file at run time. Caution has to be used if this is done. The internal version number of the module should be changed in the DLL source code for CM and tracking purposes. The internal version number is passed to GoldSim during runtime and automatically recorded in the run log. The run log is accessible by pressing Alt+M and selecting view run log.
- **Function Name**: This is the specific name of the external function in the DLL. This name is case-sensitive and must exactly match the name of the function in the source code for the external function.

There are also several options that the xLPR framework developer can set to control how the DLL is called by GoldSim (these features are not available in the player version of the model file):

- **Unload DLL after each use**: If this option is selected, GoldSim will unload the DLL (and continue the simulation). This is useful when running very large model files (in which the DLL only needs to be called infrequently). If the DLL is subsequently called again, GoldSim will automatically reload it.
- **Run Cleanup after each realization**: If this option is selected, GoldSim will call the DLL with a cleanup instruction at the end of each realization. See Appendix C of the GoldSim User's Guide [38].
- **Lock onto this file**: If this option is selected, GoldSim regards various properties of the file (including an alphanumeric code that can be used to determine whether the file contents have changed).
- **Run in separate process**: If this box is checked, GoldSim executes the DLL outside of the GoldSim process space. This can be useful for DLLs that need a lot of memory

or may be unstable, but slows down the overall running time when a DLL is called often.



External Properties : Crack_Initiation_Module_DLL

Element ID: Crack_Initiation_Module_DLL

Description:

Options

DLL Path: crack_init_v2.1.dllx

Function Name: crack_init_snl

☐ Lock onto this file ☐ Run in separate process

☐ Unload DLL after each use

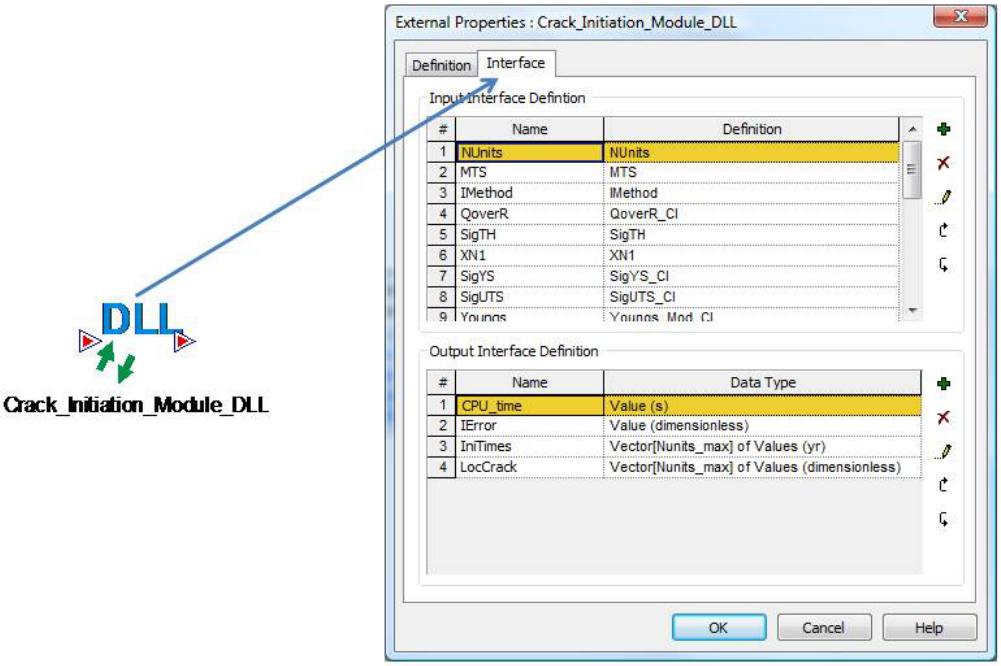
☐ Run Cleanup after each realization

The element defines 21 inputs and 4 outputs.

Save Results

☐ Final Values ☐ Time Histories

Figure 3-6. Defining the module interface through an external DLL element.



External Properties : Crack_Initiation_Module_DLL

Input Interface Defintion

#	Name	Definition
1	NUnits	NUnits
2	MTS	MTS
3	IMethod	IMethod
4	QoverR	QoverR_CI
5	SigTH	SigTH
6	XN1	XN1
7	SigYS	SigYS_CI
8	SigUTS	SigUTS_CI
9	Youngs	Youngs_Mod_CI

Output Interface Definition

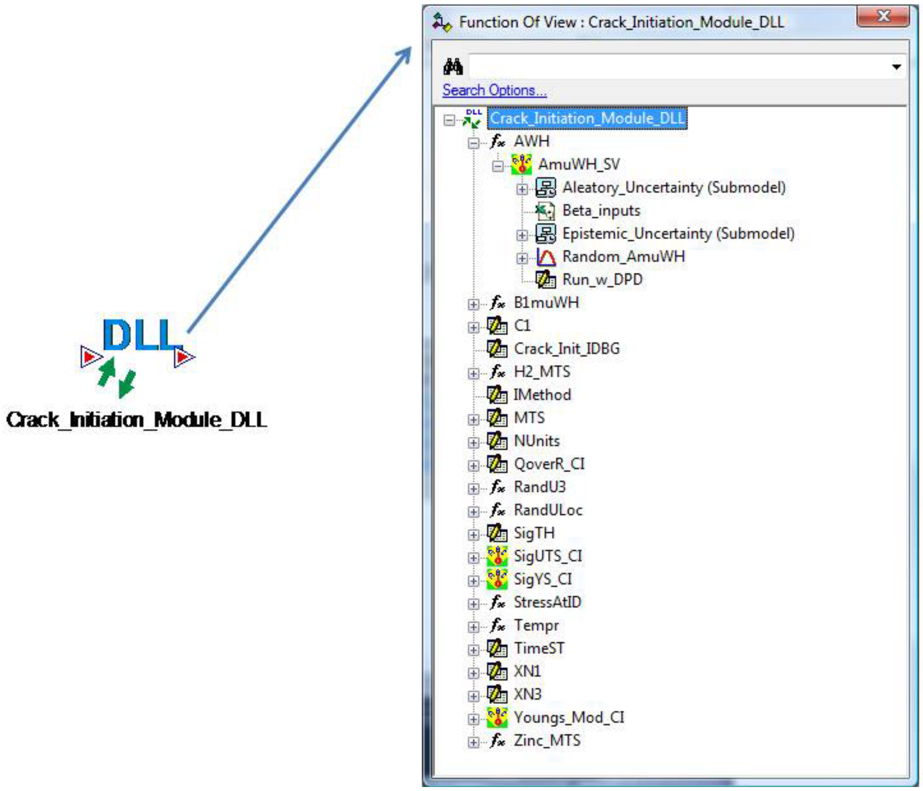
#	Name	Data Type
1	CPU_time	Value (s)
2	IError	Value (dimensionless)
3	IniTimes	Vector[Nunits_max] of Values (yr)
4	LocCrack	Vector[Nunits_max] of Values (dimensionless)

Figure 3-7. Defining the module interface arrays using an external DLL element.

The input arguments and output arguments are transferred between GoldSim and the external function in exactly the order in which they are listed on the **Interface** tab; illustrated in Figure 3-7 for the crack initiation module. Each of the modules connected to the xLPR Framework model use this standard interface.

A user can explore the interdependencies of the various elements (i.e., who affects who) in the GSxLPR framework model. The GoldSim software provides two very powerful utilities for doing this: the Function Of View, and the Affects View. If you right-click on an element and select **Function Of**, a floating window is displayed as demonstrated for the crack initiation DLL element in Figure 3-8. This is a specialized browser view of the model. It starts with the selected element and only shows those elements which affect that element (i.e., those elements which the selected element is a *function of*). In the example in Figure 3-8 it can be seen that the crack initiation DLL is a function of AWH, B1muWH, C1, IMethod and others. The user can use the function of view to drill down to trace the element's inputs to their source. As demonstrated in Figure 3-8 for AWH, which is a function of AmuWH_SV, AmuWH_SV itself is a function of five different parameters (Aleatory SubModel, Epistemic SubModel, Beta_inputs spreadsheet element, Random_AmuWH, and Run_w_DPD). By clicking any of these names the user can open the element and expose its content or if the model has results the user can display the quantity's value.

Similarly, if you select **Affects** from the context menu, a window like the function of view will be displayed. The affects view of the model starts with the selected element, and only shows those elements which are a function of that element (i.e., those elements which the selected element *affects*). Each of these can also be drilled down to trace the outputs through the model architecture.



DLL
Crack_Initiation_Module_DLL
Function Of View : Crack_Initiation_Module_DLL
Search Options...

```
Crack_Initiation_Module_DLL
  AWH
    AmuWH_SV
      Aleatory_Uncertainty (Submodel)
      Beta_inputs
      Epistemic_Uncertainty (Submodel)
      Random_AmuWH
      Run_w_DPD
  B1muWH
  C1
  Crack_Init_IDBG
  H2_MTS
  IMethod
  MTS
  NUnits
  QoverR_CI
  RandU3
  RandULoc
  SigTH
  SigUTS_CI
  SigYS_CI
  StressAtID
  Tempr
  TimeST
  XN1
  XN3
  Youngs_Mod_CI
  Zinc_MTS
```

Figure 3-8. Example of a Function Of View (Crack Initiation Model external DLL element).

The framework logic is rather simple in that each subsystem defined above for the time loop contains the xLPR modules which are linked to input and output elements. The input and output elements are a combination of scalars, vectors and matrices typically a function of number of cracks, but tailored for each module. The Time Loop subsystems are called by the GoldSim software in a sequence automatically determined by the software by the order that they must be computed. Static calculations are by default only calculated once per time loop. Since the majority of the modules run once per crack, for these subsystems (Grower, Criticality SC, Criticality TWC, Leakage, and Inspection) the xLPR framework model has been constructed to loop over the number of cracks initiated.

Additionally it should be noted that the framework model internal arrays used in the time loop have been optimized for a pilot study problem. There are a maximum of 30 subunits and 10 intervals in the operating history for a weld in the crack initiation model. Each of the 10 intervals can have a unique temperature, ID stress and water chemistry. The crack initiation module could yield a maximum of 30 cracks (one per subunit), however, the framework model and input parameters used in the module have been calibrated for 19 subunits. The time loop each subsystem utilizes the nb_crack array which has an array size of 19. The crack arrays track the crack properties (type, depth, lengths, etc.). These properties are passed to the modules and recorded from other modules and must be consistently defined in the model framework and in the module source codes. The xLPR framework model has been developed with the functionality to easily increase the array sizes, by using the GoldSim Pro software to edit the array definitions. The array view option in the GoldSim software allows the developer the opportunity to view and edit all array types used in the model. If new modules are added that require larger sized arrays, it is a simple change that can easily be verified and checked. This feature is not available in the Player software version.

3.1.2.1 Information Flow

This section provides an overview of how the subsystems and modules listed in Table 3-1 are connected within the xLPR Framework model. The subsystems and modules listed in Table 3-1 are listed in the order in which information flows within the framework model. Data is passed between modules using internal data elements (arrays and scalars). Browsing the framework file through the GUI a user can use the ***function of*** or ***affects*** views or by clicking on the influence links to actively explore the framework model information flow listed in the tables and described below. Figure 3-9 is a flow chart of the model framework.

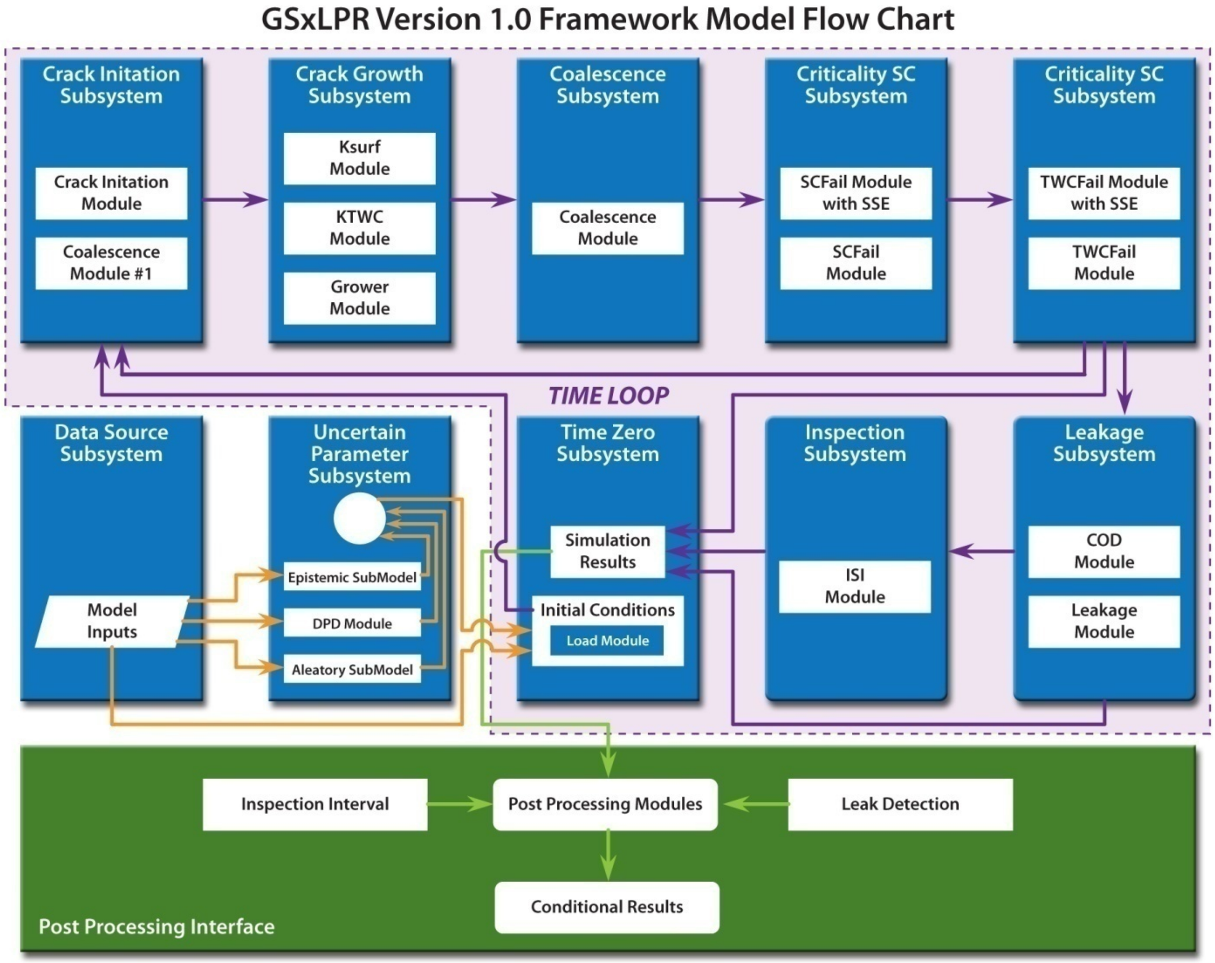


Figure 3-9. xLPR Version 1.0 Model Flow Chart with Subsystem containers.

Uncertainty Structure**:** The uncertainty structure subsystem is the first subsystem to be called by the framework model. As described in Section 3.1.1 and displayed in Figure 3-3, this subsystem calculates the sampling sequence used for the model simulation and is dependent upon the following inputs: number of aleatory realizations, number of epistemic realizations, total number of realizations (outer model), if DPD is used, and finally if a deterministic model option has been selected. The outputs from this subsystem control the sampling sequence for the Epistemic and Aleatory SubModels.

Table 3-2. Uncertainty Structure.

From Output Parameter	Input to Parameter or Container	Description
Nb_epistemic	Epistemic_Uncertainty.Number of Realizations	Number of epistemic samples
epistemic	Epistemic_Uncertainty.Realization to Run	Epistemic realization to run
Nb_Aleatory1	Aleatory_Uncertainty.Number of Realizations	Number of Aleatory samples
Aleatory_1	Aleatory_Uncertainty.Realization to Run	Aleatory realization to run
Nb_Aleatory2*	Aleatory_Uncertainty_2.Number of Realizations	Number of Aleatory samples
Aleatory_2*	Aleatory_Uncertainty_2.Realization to Run	Aleatory realization to run
*the second inner aleatory loop is not used.		

Data Source: The Data Source subsystem container contains the input interface between the model framework and the input deck (Section 3.1.3). It is logically the next subsystem called by the framework model. It receives input only from the input spreadsheet. Output from the Data Source subsystem includes all of the input values contained within the input spreadsheet for constants, uncertain parameters, conditions, and DPD specific inputs. The Data Source subsystem provides input to the Epistemic SubModel, Aleatory SubModel, Constants subsystem, and Time Zero subsystem.

Table 3-3. Data Source

From Output Parameter	Input to Parameter or Container	Description
Beta_inputs	Epistemic_Uncertainty [SubModel]	33 Uncertain parameter distributions are input to the SubModel
Beta_inputs	Aleatory_Uncertainty [SubModel]	33 Uncertain parameter distributions are input to the SubModel
Beta_DPD_inputs	DPD_Module_DLL	33 Uncertain parameter distributions are input to the DLL
Beta_inputs	Constants [Container]	26 Constants are passed to this Container
Beta_inputs	Time_Zero\Controls	1 input (mitigation time)
Beta_inputs	Time_Zero\Crack_Initiation_Inputs	22 inputs for the crack initiation model

Constants*:* The constants subsystem contains a set of data elements that parameterize the pipe geometry, material properties and loads. Outputs from this subsystem are listed in Table 3-4.

Table 3-4 Constants.

From Output Parameter		Input to Parameter or Container	Description
Thickness_C	Diameter_C	\Time_Zero\Pipe_geometry_inputs	Pipe geometry constants
Elasticity_TP304		\Time_Zero\Material_Properties_inputs	Material Property constants
Sig4_c	Mz_SSE_c	\Time_Zero\Loads_inputs	Inputs to the Load Module.
Mx_NT_c	Mx_DW_c		
My_NT_c	My_DW_c		
Mz_NT_c	Mz_DW_c		
Mx_NTS_c	Fx_NT_c		

My_NTS_c	Fx_NTS_c		
Mz_NTS_c	Fx_SSE_c		
Mx_SSE_c	Fx_DW_c		
My_SSE_c			
PWSCC_Tref_c	PWSCC_Beta_c	\Time_Zero\PWSCC_Inputs	Reference data for Crack Growth.
PWSCC_Kth_c	PWSCC_alpha_c		

Uncertain Parameters: The uncertain parameters subsystem contains the Epistemic and Aleatory SubModels and the DPD module. This subsystem receives input primarily from two subsystems, Data Source and Uncertainty Structure. However, it also receives input from the Main Dashboard, which is located in the Time Zero subsystem. The input from the Time Zero subsystem is simply a switch that runs the DPD module only if the model uses the DPD sampling. The output from this subsystem includes all of the uncertain values used in the model framework. This output is passed to the Time Zero subsystem where the data is stored and accessed by the modules run during the time loop.

Table 3-5. Uncertain Parameters

From Output Parameter	Input to Parameter or Container	Description
AmuWH_SV	\Time Zero\Crack Initiation Inputs	Within heat distribution of A (vector[Nunits_max])
B1muWH_SV	\Time Zero\Crack Initiation Inputs	Within heat distribution of B1 (vector[Nunits_max])
OD_Random_SV	\Time Zero\Loads_inputs	Random number used to calculation outer diameter stress
Beta1_SV	\Time Zero\ISI_inputs	Random number used to calculate PND
Beta2_SV	\Time Zero\ISI_inputs	Random number used to calculate PND
Pressure_SV	\Time Zero\Pressure	Sampled value from distribution of pressure.
F_weld_SV	\Time Zero\PWSCC_inputs	Sampled value fweld use in growth module
RandU3_SV	\Time Zero\Crack Initiation Inputs	Random number (vector[Nunits_max])
RandULoc_SV	\Time Zero\Crack Initiation Inputs	Random number (vector[Nunits_max])
Sig0_WRS_SV	\Time Zero\Loads_inputs	Sampled value for Sig0, used on Load Module
Sig0_WRS_Mitigated_SV	\Time Zero\Loads_inputs	Sampled value for Sig0, after mitigation.
Temperature_SV	\Time Zero\Temperature	Sampled value from distribution of temperature.
Xc_SV	\Time Zero\Loads_inputs	Sampled value for Xc, used on Load Module
Xc_mitigated_SV	\Time Zero\Loads_inputs	Sampled value for Xc, after mitigation.
half_crack_length_init_SV	\Time Zero\Crack Initiation Inputs	Sampled values for initial crack lengths vector[Nb_cracks]
crack_depth_init_SV	\Time Zero\Crack Initiation Inputs	Sampled values for initial crack depths vector[Nb_cracks]
sigy_TP304_SV	\Time Zero\Material Properties Inputs	Sampled value for yield strength (TP304)
sigu_TP304_SV	\Time Zero\Material Properties Inputs	Sampled value for ultimate strength (TP304)
n_TP304_SV	\Time Zero\Material Properties Inputs	Sampled value for n (TP304)
F_TP304_SV	\Time Zero\Material Properties Inputs	Sampled value for F (TP304)
Jic_Alloy_182_SV	\Time Zero\Material Properties Inputs	Sampled value for Jic (alloy 182)
C_Alloy_182_SV	\Time Zero\Material Properties Inputs	Sampled value for c (alloy 182)
m_Alloy_182_SV	\Time Zero\Material Properties Inputs	Sampled value for m (alloy 182)
POD_Detection_SV	\Time Zero\ISI_inputs	Random number used to calculate PND
P2V_Ratio_SV	\Time Zero\Grower_inputs	Sampled value of the peak to valley ratio
CH2_SV	\Time Zero\Grower_inputs	Sampled value of the characteristic crack growth rate curve.
QoverR_SV	\Time Zero\Grower_inputs	Sampled value of Q/R

Time Zero: This subsystem model is called at the beginning of each realization for the calculations used to define the initial conditions. In addition, because this subsystem stores the result elements, it is called once after each of the seven remaining subsystems are called. This subsystem therefore receives input from all of the other subsystems with the exception of the Uncertainty Structure subsystem. The Time Zero subsystem contains the model dashboards, run control options, calculations of the initial conditions, data arrays containing the sampled values used by the modules during the time loop calculation and stores the elements that record the simulation results displayed on the results dashboard, predefined plots, and automatically exported at text files.

Table 3-6. Time Zero Subsystem Output Parameters.

From Output Parameter		Input Subsystem Container	Description
NUnits	MTS	\Crack_Initiation [subsystem]	Input values for Crack_init module and initial conditions for the deterministic test case (crack initiation module is not run for a deterministic case).
number_cracks	SigTH		
Crack_depth_init	XN1		
half_crack_length_init	B1muwh		
inside_radius	C1		
thickness	XN3		
ring_area	Tmpr		
IMethod	TimeST		
QoverR_CI	H2_MTS		
SigYS_CI	Zinc_MTS		
SigUTS_Ci	AWH		
Youngs_mod_CI	RandULoc		
Crack_init_IDBG	RandU3		
StressAtID			
Deterministic_Test_Case			
Deterministic_initiation_times			
Deterministic_orientation			
inside_radius	Sig0_input	\Crack_Growth [subsystem]	Input values for ksurf, ktwc, and grower modules.
PWSCC_Beta	sig1		
PWSCC_Kth	sig2		
PWSCC_Tref	sig3		
temperature	sig4		
PWSCC_fweld	sig5		
PWSCC_alpha	H2		
inside_circumference	Zinc		
P2V_Ratio	CH2		
QoverR_Grower	sig0_WRS		
Start_run	pressure		
Grower_IDBG	thickness		
nb_cracks_per_timestep	Thickness	\Coalescence [subsystem]	Input values for the coalescence module.
Inside_radius	Start_run		
Outside_radius	Thickness	\Criticality_SC [subsystem]	Input values used for the SCFail module.
Axial_Load	pressure		
Bending_Moment	RO_alpha		
Inside_radius	RO_Sig0		
Bending_Moment_SSE	RO_n		
yield_Strength	Resist_Jic		

From Output Parameter		Input Subsystem Container	Description
ultimate_Strength	Resist_C		
Run_with_SC_Criticality	Resist_m		
SCFail_Method	Start_run		
SCFail_i_write	Run_w_SSE		
Outside_radius	Thickness	\Criticality_TWC [subsystem]	Input values used for the TWCFail module.
Axial_Load	pressure		
Bending_Moment	RO_alpha		
Inside_radius	RO_Sig0		
Bending_Moment_SSE	RO_n		
yield_Strength	Resist_Jic		
ultimate_Strength	Resist_C		
Run_with_TWC_Criticality	Resist_m		
TWCFail_Method	Start_run		
TWCFail_i_write	Run_w_SSE		
Axial_Load	RO_alpha	\Leakage [subsystem]	Input values used for the COD and leakage modules.
Bending_Moment	RO_Sig0		
yield_Strength	RO_n		
ultimate_Strength	Resist_Jic		
Inside_radius	Resist_C		
Outside_radius	Resist_m		
pressure	Start_run		
ambient_pressure	RO_epso		
temperature	sig0_WRS		
Thickness	sig1		
Run_with_leak_rate	sig2		
COD_Method	sig3		
	sig4		
Thickness	POD_Beta1	\Inspection [subsystem]	Input values used for the inspection module.
Start_run	POD_Beta2		
Run_w_inspection			
Run_w_SSE		\Constants [subsystem]	Input condition
Run_w_DPD	DPD_NBIN	\Uncertain_Parameters [subsystem]	Model input conditions.
DP_Run			

Crack Initiation: The crack initiation subsystem contains the crack initiation module and populates the arrays for the initial conditions for crack depths, lengths, orientation, and type used in the Time Loop. The crack initiation module is run once per outer model realization and provides the crack initiation times and locations. A call to the coalescence module is contained within this subsystem. In this subsystem, the coalesce module is run only when a new crack has initialized to ensure that the new crack is not placed within an existing crack. The crack arrays are updated at the beginning of each time step. Input from to the crack initiation subsystem comes primarily from the Time Zero subsystem, but the final conditions for the crack type, length, depth, and orientation are passed to the crack initiation subsystem from the Inspection subsystem at the end of the previous time step. This becomes the initial conditions for the beginning of the next time step. Output from the crack initiation subsystem is passed to only two other subsystems. Updated arrays of crack type, length, depth, and orientation are passed to the Crack Growth subsystem. The number of cracks and any module errors are passed to the Time Zero subsystem.

Table 3-7. Crack Initiation

From Output Parameter		Input to Subsystem Container	Description
Init_Crack_Type_i	Init_half_Crack_length_i	\Crack_Growth [subsystem]	Crack conditions at the start of the time step.
Init_crack_depth_i	Init_crack_orientation_i		
Nb_cracks_initiated	Fraction_area_cracked		
Current_nb_cracks			
Nb_cracks_CI	Initiation_Time_CI	\ Time_Zero [subsystem]	Results passed to pre-defined result elements.
Current_nb_cracks			
Crack_Initiation_Module_DLL.IErr			
Init_Coalescence_Module_DLL.IErr			

Crack Growth: The crack growth subsystem calculates the PWSCC induced crack growth. This subsystem model includes the stress intensity modules, Ksurf and Ktwc, which calculate the stress once each time step. This subsystem is called after the crack initiation subsystem and receives the crack information from the crack initiation subsystem. Output from this subsystem is passed to the Coalescence and Time Zero subsystems.

Table 3-8. Crack Growth.

From Output Parameter	Input to Subsystem Container	Description
Growth_Crack_Type	\Coalescence [subsystem]	Crack conditions updated by the growth subsystem
Growth _half_Crack_length		
Growth _crack_depth		
Growth _crack_orientation		
Current_nb_cracks_g		
Nb_cracks_initiated_g		
Growth_K	\ Time_Zero [subsystem]	Stress intensity and module error codes.
Crack_Growth_Submodels.Grower_Sc_Ierr		
Crack_Growth_Submodels.Grower_twc_Ierr		

Coalescence: The coalescence subsystem checks for coalescence when two or more cracks exist. This subsystem contains the coalescence module and is run once per time step after the Crack Growth subsystem has updated the crack conditions (when two or more cracks exist). Output from this subsystem is passed to the Criticality_SC subsystem. In addition, the coalescence half crack length is passed to the Time Zero subsystem to record when the maximum half length is equal to Pi (the Coalesce module has a maximum limit for a half crack length equal to Pi). This is an internal check to ensure that ensures that the downstream modules are operating within their ranges of validity.

Table 3-9. Coalescence.

From Output Parameter	Input to Subsystem Container	Description
Coalescence_Crack_Type	\Criticality_SC [subsystem]	Crack conditions updated by the coalescence subsystem
Coalescence _half_Crack_length		
Coalescence _crack_depth		
Coalescence _crack_orientation		
Nb_cracks_initiated_c		
Current_nb_cracks_c		
Coalescence_half_crack_length	\ Time_Zero [subsystem]	Module error codes.
Coalescence_DLL.Ierr		

Criticality SC: The Criticality SC subsystem checks the stability of SCs and is run only when one or more SCs exist. This subsystem contains two instances of the SCFail module, both are run when the simulation is run with the (default) option selected to evaluate SSE loads in addition to normal loads (see Section 3.1.5). One is run to check for SC failure under SSE conditions, the second checks the SC stability for normal loads, but the second instance is only run if the crack has already failed under the higher SSE loads. The Criticality SC subsystem is a subsystem that loops over the number of cracks that have been initiated. Additionally, the SCFail modules are conditional and called only on loops when the crack is a SC. It is not called for TWC or cracks that have been coalesced with other cracks. Output from this subsystem is passed to the Criticality TWC subsystem. Results are passed to the Time Zero subsystem.

Table 3-10. Criticality SC.

From Output Parameter	Input to Subsystem Container	Description
Criticality_Crack_Type	\Criticality_TWC [subsystem]	Crack conditions updated by the Criticality SC subsystem
Criticality _half_Crack_length		
Criticality _crack_depth		
Criticality _crack_orientation		
Nb_cracks_initiated_sc		
SC_Fail_Occurred	\ Time_Zero [subsystem]	SC Failure results and module error code.
SC_Time_of_Failure		
SC_Fail_Occurred_SSE		
SC_Time_of_Failure_SSE		
SCFail_IF_Flag		

Criticality TWC: The Criticality TWC subsystem checks the stability of TWCs and is run only when one or more TWCs exist. This subsystem contains two instances of the TWCFail module; both are run when the simulation is run with the (default) option to evaluate SSE loads in addition to normal loads (see Section 3.1.5). One is run to check for TWC failure under SSE conditions, the second checks the TWC stability for normal loads; but the second instance is only run if the crack has already failed under the higher SSE loads. The Criticality TWC subsystem is a subsystem that loops over the number of cracks that have been initiated. Additionally, the TWCFail modules are conditional and called only on loops when the crack is a TWC. It is not called for SC or cracks that have been coalesced with other cracks. Output from this subsystem is passed to the Leakage subsystem. Results are passed to the Time Zero subsystem.

Table 3-11. Criticality TWC.

From Output Parameter	Input to Subsystem Container	Description
	\Leakage [subsystem]	Crack conditions updated by the Criticality TWC subsystem
Critical_twc_Crack_Type		
Critical_twc _half_Crack_length		
Critical_twc _crack_depth		
Critical_twc _crack_orientation		
Time_Of_Failure	\ Time_Zero [subsystem]	TWC Failure results and module error code.
Critical_Failure_Occurred		
Time_Of_Failure_SSE		
Critical_Failure_Occurred_SSE		
TWCFail_IF_Flag		

Leakage: The Leakage subsystem calculates the COD and leakage rates for TWCs. The subsystem is run only when one or more TWCs exist. This subsystem contains the COD and SQUIRT modules. The leakage subsystem is a subsystem that loops over the number of cracks that have been initiated. Additionally, the modules are conditional and called only on loops when the crack is a TWC. It is not called for SCs or cracks that have been coalesced with other cracks. As mentioned previously, the maximum half crack length in the Coalescence Module is set to Pi. However the SQUIRT module will fail if half crack length is equal to Pi, therefore a maximum half crack length to 0.99*Pi is used. Output from Criticality TWC subsystem is passed through this subsystem to the Inspection subsystem. Since the leakage rate is not used in the Time Loop, but rather during the post processing step, subsystem results are passed only to the Time Zero subsystem.

Table 3-12. Leakage.

From Output Parameter	Input to Subsystem Container	Description
Leakage_Crack_Type	\Inspection [subsystem]	Crack conditions passed through from the Criticality TWC subsystem
Leakage _half_Crack_length		
Leakage _crack_depth		
Leakage _crack_orientation		
Nb_Cracks_initiated_leakage		
Leak_Rate_Sum	\ Time_Zero [subsystem]	COD and Leakage results and COD module error code.
COD_OD		
Leak_Occurred		
Time_of_Leak		
COD_IERR		

Inspection: The Inspection subsystem calculates the PND of a SC. For TWC the PND is set to zero. For rupture the PND is set to 1. The subsystem is run only when one or more SCs exist. This subsystem contains the ISI module. The subsystem loops over the number of cracks that have been initiated. The subsystem is conditional and called only when the crack is a SC. It is not called for TWCs or cracks that have been coalesced with other cracks. Output from Criticality TWC subsystem again passed through this subsystem since the inspection results are not used in the Time Loop, but rather during the post processing step. The subsystem results are passed only to the Time Zero subsystem.

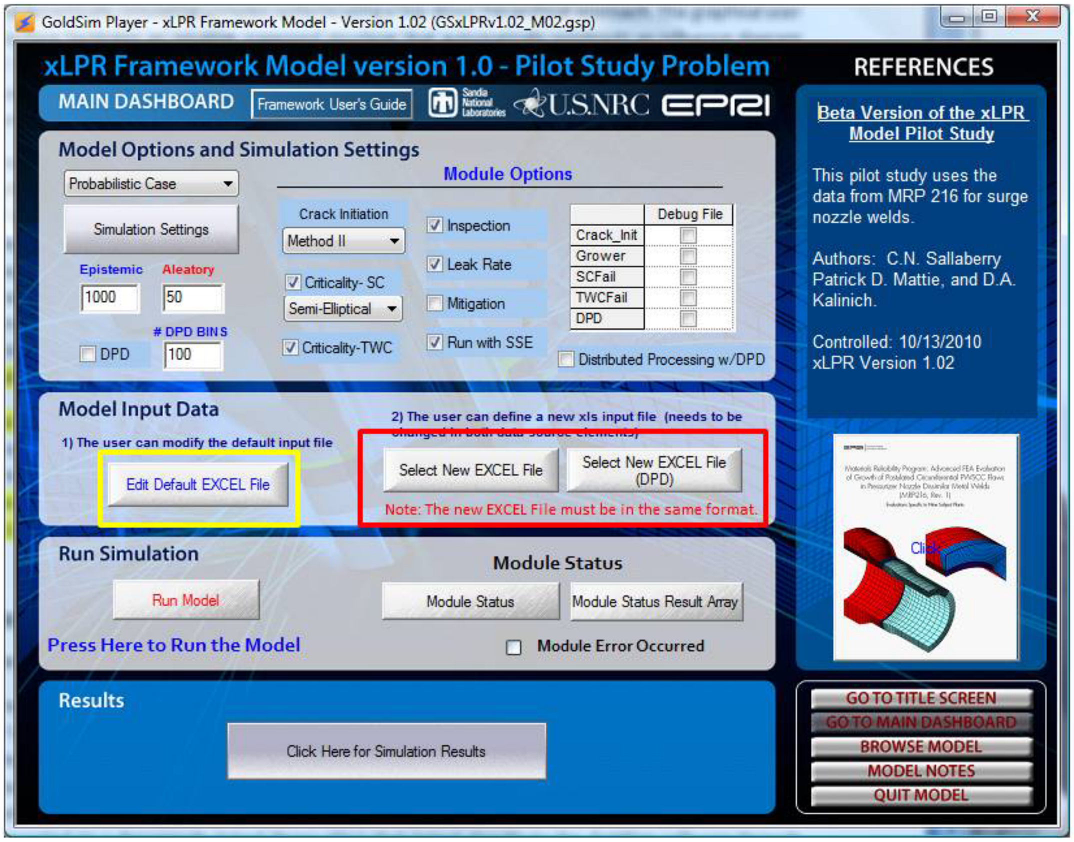
Table 3-13. Inspection.

From Output Parameter	Input to Subsystem Container	Description
Inspect_Crack_Type	\Crack Initiation [subsystem]	Crack conditions passed through from the Criticality TWC subsystem
Inspect _half_Crack_length		
Inspect _crack_depth		
Inspect _crack_orientation		
Inspect _half_Crack_length	\ Time_Zero [subsystem]	ISI PND results and crack conditions at the end of the time step.
Inspect _crack_depth		
Inspect _crack_orientation		
Nb_cracks_initiated_inspection		
pnd		
non_detection		

3.1.3 Input Interface and Model Input Decks

The xLPR framework model file utilizes a spreadsheet element which allows the exchange of data with a Microsoft® Excel spreadsheet file. Data is read from specified cells in the spreadsheet by the spreadsheet element and passed to the parameters in the framework model or in reverse; data from the framework can be saved in Excel. Using this interface enables the user to generate input decks using Excel without modification of the framework. In this way the simulation model can access the various data sources directly from verified input decks to facilitate and ensure the quality of the data transfer. GoldSim also provides a more powerful method. In particular, GoldSim elements can be linked directly to an ODBC-compliant database. After defining the linkage, you can then instruct GoldSim to download the data from a controlled database before the model is run. When it does this, GoldSim internally records the time and date at which the download occurred, along with other reference information retrieved from the database (e.g., document references), and this is automatically stored in the individual model parameter descriptions and in the model Run Log. This information is also actively displayed in the tool-tip for the linked element when the user browses the model. This allows you to confirm that the correct data were loaded into your model, and provides very strong and defensible quality control over your model input data. Once the dB link has been created using the GoldSim Pro software the data can be modified using the dB GUI and downloaded into a Player file by clicking a button on the dashboard interface. This feature is already available with the GoldSim Pro software, however due to the limited scope and schedule of the pilot study, a controlled database was not constructed; future versions of the xLPR framework model would likely necessitate utilizing this available software capability (Appendix F [38]).

The listing of the input parameters, values and their distributions for uncertain parameters are listed in Appendix C. Using the Excel spreadsheet provided with the xLPR Version 1.0 model, the user can easily change the input values used in the framework model by selecting the “Edit Default EXCEL File” button on the Main Dashboard screen, Figure 3-10. The new values will be automatically read in at run time and distributed to the appropriate model parameters. The modified spreadsheet must be in the local directory with the xLPR Framework model file. The user can easily switch between different spreadsheets using the “Select New EXCEL File” and “Select New EXCEL File (DPD)” buttons also on the Main Dashboard screen, Figure 3-10. Since the xLPR framework model was designed to be utilized as a GoldSim Player file, the Excel spreadsheet format cannot be changed. A user can replace any value within the range of parameter validity and its uncertainty type (i.e., whether it is considered as aleatory or epistemic), but the distribution type and input requirements must be maintained. In addition, neither the distribution type (e.g., normal, uniform, truncated normal, log normal, constant) nor the parameter units can be modified in the spreadsheet. While it is a trivial thing to change the distribution type using the GoldSim Pro software, it is not a feature currently available using the Player Software, and can only be done if a Player file is linked to an ODBC database. The framework model has pre-assigned units for each input parameter value (and hence dimensions) to the elements (and hence to the inputs and outputs) of the model. One of the more powerful features of GoldSim is that it is dimensionally-aware. The GoldSim ensures dimensional consistency and carries out all unit conversions internally. For example, you could add feet and meters in an expression, and GoldSim would internally carry out the conversion. (If you tried to add feet and seconds, however, GoldSim would issue a warning message and prevent you from doing so.) GoldSim has an extensive internal database of units and conversion factors. Data can be entered or displayed in any units and including user defined units. The pre-assigned parameter units in the xLPR model however, cannot be changed using the input spreadsheet. While changing the units is trivial using the GoldSim Pro software or with a file linked to an input database, changing the units in the Excel spreadsheet and running the player software will produce erroneous results.


GoldSim Player - xLPR Framework Model - Version 1.02 (GSxLPRv1.02_M02.gsp)
xLPR Framework Model version 1.0 - Pilot Study Problem
MAIN DASHBOARD
Framework User's Guide
Sandia National Laboratories
U.S.NRC
EPRI
REFERENCES
Beta Version of the xLPR Model Pilot Study
Model Options and Simulation Settings
Module Options
Simulation Settings
Model Input Data
Edit Default EXCEL File
Select New EXCEL File
Select New EXCEL File (DPD)
Run Simulation
Run Model
Press Here to Run the Model
Module Status
Module Status
Module Status Result Array
Results
Click Here for Simulation Results
Click
GO TO TITLE SCREEN
GO TO MAIN DASHBOARD
BROWSE MODEL
MODEL NOTES
QUIT MODEL

Probabilistic Case

Epistemic: 1000

Aleatory: 50

☐ DPD

DPD BINS: 100

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

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.

☐ Module Error Occurred

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

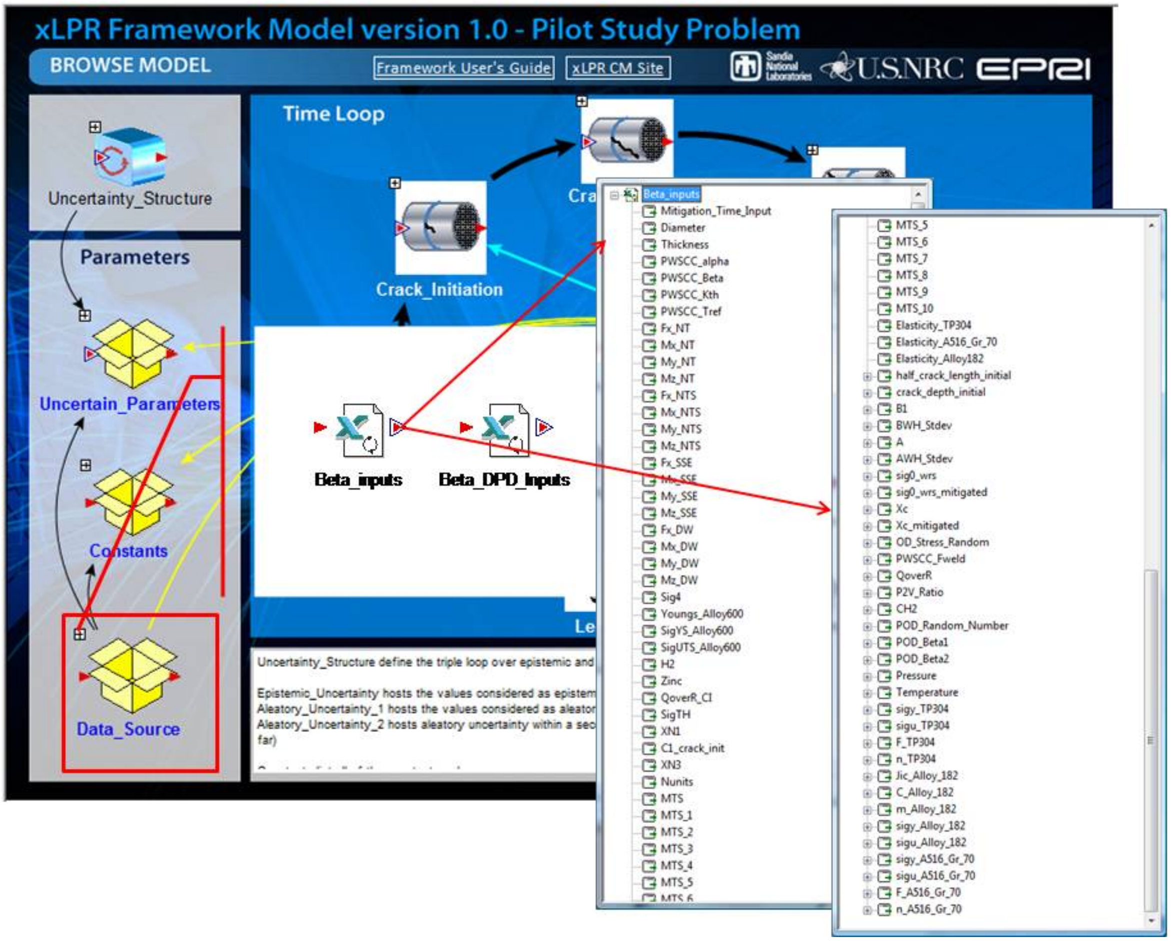
Authors: C.N. Sallaberry Patrick D. Mattie, and D.A. Kalinich.

Controlled: 10/13/2010
xLPR Version 1.02

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

Figure 3-10. Main Dashboard with highlighting for options for editing or selecting new spreadsheets used in the xLPR Framework Model.

The spreadsheet element allows for the automatic distribution of the model parameter inputs from a centralized location. Figure 3-11, shows the location in the model framework of the spreadsheet elements. Using the Affects View described above, the user can trace the input through the model file. The uncertain inputs are fed directly to both the epistemic and aleatory SubModels. The input spreadsheet contains a cell for each uncertain distribution in which the designation of epistemic or aleatory is defined. The code selects the sampled value from either the epistemic or aleatory SubModels based upon this input. This feature enables the user to conduct parameter sensitivity studies on the designation epistemic or aleatory.



Uncertainty_Structure define the triple loop over epistemic and

Epistemic_Uncertainty hosts the values considered as epistem
Aleatory_Uncertainty_1 hosts the values considered as aleator
Aleatory_Uncertainty_2 hosts aleatory uncertainty within a sec
far)

Figure 3-11. Affects View using the Beta_inputs spreadsheet element within the Data Source subsystem container.

DPD has a different format for the uncertain inputs and therefore a separate spreadsheet element was defined, as shown in Figures 3-10 & 3-11. This spreadsheet element reads the input from the DPD tab of the xLPR framework model input spreadsheet selected for the analysis. This tab is automatically updated from the uncertain parameters tab, so the user does not need to define the distributions twice. However, to perform the DPD calculations for importance sampling the endpoints corresponding to the Pth percentile for importance sampling (P3 values) need to be added directly to the designated cells on the DPD tab of the input spreadsheet. It is also important to note that the DPD tab of the input spreadsheet has an option to turn DPD off and on for individual parameters. This option has not implemented in this version of the framework. It is reserved for possible future use.

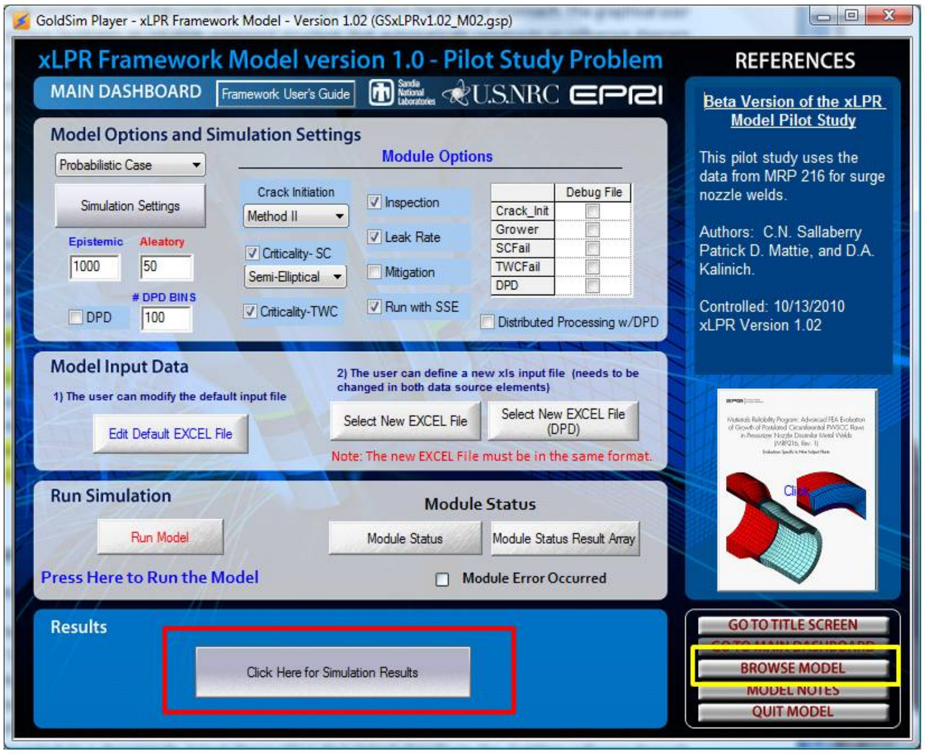
3.1.3.1 Additional Considerations

The xLPR Framework inputs spreadsheet was created in Excel 2007 and thus the default links are to the .xlsx file. The user therefore must have Excel 2007 installed on their computer to run the model. To accommodate an older version of Excel (e.g., Excel 97, Excel 2000, Excel 2002, Excel 2003) the user will have to convert the Excel 2007 file to their installed Excel version before attempting to run the file. The converted file needs to be selected as the a new Excel spreadsheet as explained above using the pre-defined buttons on the Main Dashboard depicted in Figure 3-10. Additionally, it should be noted that in controlled versions 1.0, 1.01, and 1.02 of the framework model, an option was inadvertently selected on the spreadsheet element that keeps Excel open during the entire simulation. Excel cannot be used for other applications until the simulation has completed. Later versions of the framework model that use the input spreadsheet will not have this limitation.

3.1.4 Saving and Navigating Results

The lower left-hand button on the right side of the Run Controller (the **Options** button) provides access to a menu to allow you to save the player file. The GoldSim Run Controller will automatically be present when you open the player file. The Run Controller is the “control panel” which allows you to run, monitor, pause and step through the simulation in a controlled manner [38]. It is highly recommended that a user save the model run before browsing the results or conducting any post processing analyses. The current version of the GoldSim Player software does not have the option available in GoldSim Pro to automatically save the file after the model run has completed.

The GoldSim GUI software was developed specifically for creating models that are transparent, easy to understand and navigate. The Windows based GUI interface and graphical element used by the GoldSim software allow the user to navigating the model by following the information flow defined by influence links. In addition, the dashboard interface developed for the xLPR framework model facilitates the display and interrogation of the model results. A series of customized dashboard interfaces have been built using the GoldSim Pro software and are the primary interface when using the framework model. When a simulation has completed the results can be accessed through the Results Dashboard or by browsing the model. Figure 3-12, shows the links to these options from the Main Dashboard.


GoldSim Player - xLPR Framework Model - Version 1.02 (GSxLPRv1.02_M02.gsp)

xLPR Framework Model version 1.0 - Pilot Study Problem

MAIN DASHBOARD

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

□ Distributed Processing w/DPD

	Debug File
Crack_Init	□
Grower	□
SCFail	□
TWCFail	□
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

Results

REFERENCES

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 3-12. Main Dashboard view showing the links to the Results Dashboard and Browse Model Option.

The **Click Here for Simulation Results** button in the Main Dashboard (Figure 3-12) screen will take the user to the Results screen. Model results can be displayed after the run has completed or monitored by the user during the run from the Results screen (e.g., the run is started by the user clicking on the GoldSim Run Controller Run button while the framework is on the Results screen) (see Figure 3-17). Note however, that the overhead associated with displaying the Simulation Status results during runtime causes the framework to run slower. During runtime monitoring indicators are provided for:

- number of cracks formed
- the number of cracks coalesced
- the number of SC
- the number of TWC
- the leakage rate
- if pipe rupture has occurred and at what time
- the maximum number of cracks that could occur

Default plots and tables displaying selected results for output variables listed in Table 3-14 can be displayed by clicking on the GoldSim Plots buttons provided on the Results screen (see Figure 3-17). A link is also provided that will take the user to the results container within the framework model where additional result are saved and can be plotted.

It is important to note that the results displayed in the GoldSim file are raw results that have not been post processed. The results are not averaged over the aleatory samples and therefore the distribution of the individual results represents the range over all the uncertainty in the system and therefore will not match the distribution of the expected values which are an average over the aleatory samples (see Section 3.3). The mean results are the same as the mean results for the expected values (averaged over the aleatory) produced with the post processor without inspection or leak detection applied (see Section 3.2). When the DPD module is used without importance sampling the results displayed and contained within the GoldSim model file do not need to be post processed unless inspection and leak detection is evaluated. Since there is currently no separation of aleatory and epistemic uncertainties, without inspection or detection both the distribution of results and the mean and percentile values can be calculated by the framework without the post processor. However, if importance sampling is used with the DPD module, the post processor must be used to as the DPD response weights are not applied automatically in the framework. The mean and percentile results will be much higher than what they actually are once the response weights are applied. The results interface was designed to monitor the simulation, evaluation of the mean results, and to interrogate individual inner and outer model results for debugging and model verification. It is possible to calculate the average over aleatory and the appropriately weighted mean when using importance sampling; these options need to be evaluated for inclusion in a future version of the framework.

Table 3-14. xLPR Framework Results Plots/Tables.

Number of Cracks	Probability of First TWC vs. Time
Crack Type	Probability of Rupture vs. Time
Crack Location	Probability of Leakage vs. Time
Crack Depth	Total Leakage Rate vs. Time
Half Crack Length	First Leak Time Distribution
Area/thickness (a/t)	PND
Theta/Pi	Probability of TWC Distribution
Distribution of Number of Cracks Predicted	TWC Times Distribution
Stress Intensity Deepest Point	Distribution of Rupture Probability w/SSE
Stress Intensity Crack Surface	Rupture Times Distribution w/SSE
Fraction of Surface Area Cracked	Probability of SC Failure w/SSE
Crack Opening Area	SC Failure Times Distribution w/SSE
Crack Opening Displacement	

Since the GoldSim software contains internal plotting capabilities, the user can generate a plot of the results from any graphical element in the GoldSim model file which can be displayed by browsing the model, selecting an element, and clicking the right mouse button. Elements with results saved will have a green arrowhead (bold in the navigation plane). A red arrowhead indicates that the element's results have not been saved. The GoldSim software has the ability to save both final value and time history results for every element in the model framework. In Version 1.02 of the GSxLPR framework model there are 892 model elements (data, functions, external DLL, results). In addition, the GoldSim software has the ability to save and view the status of every subsystem (e.g., conditional tests, looping count, number of calls, etc.). Therefore the user has the ability to interrogate the results of any element in the xLPR framework model. During debugging and development activities all of the results were saved which facilitated the checking and review process but resulted in very large file sizes, approximately 1 GB for the default 50,000 realization simulation. However, this can be misleading as the GoldSim Software automatically compresses the files at a ratio of about 10 to 1, so when the model file is saved, opened, or results are exported the software must first unzip the results, into approximately ten 1 GB files. This is very slow and uses a lot of computer resources. Thus saving all of the results is used only for development, verification tests and debugging. It is a simple process to save the results which can be done at the main level for all elements, at the subsystem level for all elements within the subsystem, or for each element individually. However, this feature is only available in GoldSim Pro, the Player Software does not have the ability to turn on and off which elements will be saved. Therefore a more tractable number of results have been pre-selected and available for review and export using the GoldSim Player software. Table 3-15 lists the saved results by subsystem and container.

Table 3-15. GoldSim Results by Subsystem and Container.

Subsystem\Container Names	Description Default Results Saved
Uncertainty_Structure	All elements final values are saved.
\Data_Source	Final values are saved.
\Uncertain_Parameters\Sampled_Values	Final values are saved.
\Crack_Initiation\Crack_Initiation_Module	Final values for Initiation Times, Orientation, Nb_Cracks_Initiated
\Time_zero	Final values Temperature, Pressure, Ambient_Pressure
\Time_zero\Controls	Final values for all control elements
\Time_zero\Controls Module_Completion_Status	All time histories and final values are saved for the module error status results.
\Time_zero\Controls\Dashboards_and_Results\Sampled_Value_Results	Contains the result element for the sampled values (final values)
\Time_zero\Controls\Dashboards_and_Results\Vector_Export_Results	Contains the time history results for all of the crack properties all 19 cracks.
\Time_zero\Controls\Dashboards_and_Results\Results	Parameters used to save default final values and time history model results.
\Time_zero\Controls\Dashboards_and_Results\Result_Plots	Plots used for results displayed on the Results Dashboard.

All of the results displayed using either the pre-defined plots or user generated plots can be edited using the default plotting software in GoldSim. Individual realizations can be screened with user defined criteria. The **Screening Condition** can be any conditional expression. Only realizations that meet the Screening Condition are displayed; the remainders are screened out. You specify such screening conditions within GoldSim via a dialog box accessible from the main menu under **Run|Screen Realizations**. Consult the GoldSim User's Guide [38] for more information on the software features available for analyzing and interrogating the results.

It is also important to note here that although internal calculations are carried out using double precision numbers, results are only stored as single precision numbers (in order to reduce storage requirements). This means that when results are viewed in tables or charts, the range of values that can be displayed is between -1.2E-38 and 3.4E38. This is also true for exporting results. This is significant in that the individual DPD response weights are multiplied by the number of DPD bins before each realization's response weight is calculated. The response weights are renormalized in the post processor.

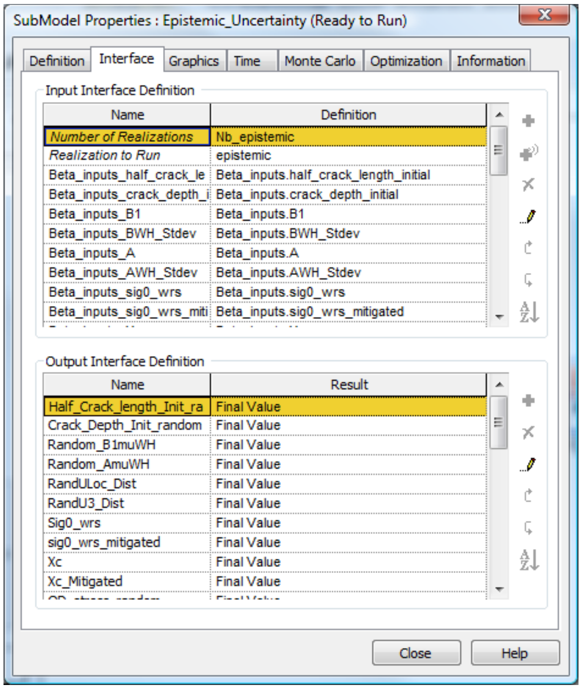
3.1.4.1 Implementation Logic used for Saving Results in the Framework

As discussed in Sections 3.1.1 and 3.1.2, the framework model uses both SubModel and Looping Containers and the various subsystems used to build the xLPR framework. Embedded SubModels are considered entirely independent of the outer time loop model and the information that passes between SubModel and the outer model time loop must be handled through a special interface. The looping containers, by default save only the results from the final loop, so again in order to capture all of the results, some additional logic needed to be added to the framework model.

SubModel Interface

A SubModel is a self-contained system (i.e., a separate model), elements inside a SubModel cannot by default link to anything on the outside (i.e., in the outer model). As a result, you cannot reference outputs that exist outside of the SubModel in the same way you would do so from inside a Container. Access to the outer model is done by creating an interface between the SubModel and the outer model. The interface is accessed via the Interface tab on the SubModel dialog box. The SubModel dialog box is accessed by viewing the SubModel properties (right click on the SubModel Icon). Figure 3-13 shows the SubModel interface for the Epistemic SubModel. The SubModel interface tab contains both the input and output interface required to pass the information between the embedded model and the outer time loop. All of the output listed in the output interface definition is available to be used in the outer time loop and can be saved.

It is important to note that by default the data is passed as a constant value or a time history. This is significant since a SubModel is a separate stand alone model, a complete simulation of a SubModel for multiple realizations can be run for each realization of the outer model. If the multiple realizations were run for each subsystem for each realization of the outer model, the results passed through the SubModel output interface array would be the mean value over the set of realizations. The software assumes that the submodel is either deterministic or probabilitisic and for probabilistic submodels has only the options available to pass percentiles, mean, median, or standard deviation. Since the construct of the xLPR framework model needs a different set of sampled values for each outer model realization, the option to run a single realization was utilized. For each realization of the outer model the Realization Number to Run is passed as a variable to the SubModel Monte Carlo tab. This enabled a single realization of each SubModel to be run for each realization of the outer time loop; e.g., one realization of epistemic and aleatory at a time as calculated using the process described in Section 3.1.1.



SubModel Properties : Epistemic_Uncertainty (Ready to Run)

Input Interface Definition

Name	Definition
Number of Realizations	Nb_epistemic
Realization to Run	epistemic
Beta_inputs_half_crack_le	Beta_inputs.half_crack_length_initial
Beta_inputs_crack_depth_i	Beta_inputs.crack_depth_initial
Beta_inputs_B1	Beta_inputs.B1
Beta_inputs_BWH_Stdev	Beta_inputs.BWH_Stdev
Beta_inputs_A	Beta_inputs.A
Beta_inputs_AWH_Stdev	Beta_inputs.AWH_Stdev
Beta_inputs_sig0_wrs	Beta_inputs.sig0_wrs
Beta_inputs_sig0_wrs_miti	Beta_inputs.sig0_wrs_mitigated

Output Interface Definition

Name	Result
Half_Crack_length_Init_ra	Final Value
Crack_Depth_Init_random	Final Value
Random_B1muWH	Final Value
Random_AmuWH	Final Value
RandULoc_Dist	Final Value
RandU3_Dist	Final Value
Sig0_wrs	Final Value
sig0_wrs_mitigated	Final Value
Xc	Final Value
Xc_Mitigated	Final Value

Figure 3-13. SubModel Interface Tab for the Epistemic Submodel.

Looping Container Interface

The GoldSim software includes the ability to create a looping container that carries out an iterative calculation at each time step with the number of iterations controlled by specific criteria. In the xLPR framework model, several of the modules are run once for each crack, and thus the framework needs to loop over the number of existing cracks. For five subsystems described in Section 3.1.2, which are looping subsystems, the number of loops is controlled by the number of cracks that have initiated in that realization. The looping subsystems are in the time loop, so therefore the number changes with time; progressively as more cracks initiate the number of loops increase. Each loop represents an evaluation of a single crack, starting with crack 1 and ending with the number of cracks that have been initiated. By default the modules are evaluating scalar values, rather than an array of 19 cracks. The values for the crack properties are extracted from the crack property vectors (crack type, crack depth, crack length, crack orientation, etc.), and passed to the module as scalar values. The module returns a scalar value, once each loop, once for each crack. Once each loop of the subsystem the crack array needs to be updated; once for each crack. This is accomplished in GoldSim using Discrete Change and Integrator software elements [38]. A discrete event is triggered once each loop. A discrete change is used to build a vector equal to the maximum number of cracks and inserts the scalar value passed back by the module into the vector element equal to the crack number (e.g., loop number). For the remaining cracks, the value inserted is zero (may be any value, it is not zero for the inspection subsystem). For each loop the integrator element is updated by the discrete event, and the updated vector is added to the results from the previous loop. At the end of the looping, one value has been added for each element in the vector, e.g., one for each crack that has been initiated. The resulting vector contained within the integrator element, has thus recorded a single value for each crack. This occurs at each time step, starting with the initialization of the results vector.

3.1.5 Model Options and Modeling Scenarios

The xLPR framework was constructed to be able to simulate a number of different modeling cases and model scenarios using the set of modules provided for the pilot study. In order to enhance the framework flexibility, an option has been provided to individually deactivate some of the modules. The default framework has been constructed as a probabilistic analysis. The simulation uses the nominal stresses with an option to evaluate the probability of occurrence of rupture with SSE stress. The SSE option does not account for the frequency of an initiating event. This would have to be evaluated during post processing. The xLPR Framework model uses the default GoldSim Monte Carlo with LHS option or an alternative module (DPD with importance sampling) to generate the sample set from distributions provided for the uncertain parameters (see Section 3.1.1).

The Main Dashboard is used as the GUI to allow the user to select between various model options as shown in Figure 3-14. The model options are divided into two main categories, probabilistic options and module options. The probabilistic options control the time step size, the number of samples generated from the uncertain distributions, the number of realizations and the probabilistic method. The module options control the capability to select which modules to run, selection between alternative conceptual models (where available), and the ability to print debug files from individual models. It should be noted that an optional deterministic modeling case has been developed. This case was used for verification testing and the results are presented in Section 5. The GoldSim Player software does not have the capability to switch between deterministic and probabilistic modeling options for the embedded SubModels. Therefore, two modeling cases were created for xLPR using the GoldSim Pro software, one for the probabilistic analyses and one for the deterministic verification cases.

3.1.5.1 Probabilistic Model Options

The xLPR framework probabilistic model options are changed using the input options as defined on the Main Dashboard and displayed in Figure 3-14 and are labeled as items #1-4.

1. **Probabilistic/Deterministic Pull-Down:** When using the GoldSim Player file version of the xLPR model, the pull-down menu is only for the deterministic framework model version and is used to switch between Deterministic Case #1 and Deterministic Case #2. Using the player file it cannot be used to change between the deterministic cases and the probabilistic case. Using GoldSim Pro you can use the switch to change the initial setting to deterministic. The number of aleatory and epistemic realizations will automatically be set equal to 1; the crack initiation module will be turned off, and replaced with the initial conditions for Deterministic Case #1 or Deterministic Case #2. Using GoldSim Pro the user must also delete the number of realizations and realization to run inputs from both of the SubModels (Epistemic and Aleatory) and select “Deterministic Run” in the SubModel simulation settings.

2. **Simulation Settings Button**: Clicking on the Simulation Settings button opens the Simulation Settings pop-up menu to one its tabs (see Figure 3-15). For the player file probabilistic framework the Monte Carlo tab, allows the user to change:

    a. the total number of realizations (this value must be consistent with the number of epistemic and aleatory realizations for a Monte Carlo run)
    b. the number of histories to save
    c. run only a single realization (specified by realization number)

    Clicking on the Time tab allows the user to access the framework's time settings. Here, the user can change:
    a. the Time Display Units
    b. the simulation duration
    c. the number of time steps and the plot (i.e., output) frequency

There are no changes to the framework need to be made in either the Globals or Information tabs.

3. **Epistemic and Aleatory:** The two input boxes allow the user to specify the number of epistemic and aleatory realizations. Note that the product of these two values must match the total number of realizations specified in the **Simulation Setting…/Monte Carlo** pop-up dialog box when running a Monte Carlo simulation. These values are not used when the DPD Module option is checked.

4. **DPD/# DPD Bins:** Checking the box causes the framework to use the DPD module to sample the uncertain parameters. The input box allows the user to specify the number of bins used by the DPD module. As noted in Section 3.1.1, when the DPD module is activated, all uncertain parameters are treated as epistemic. The uncertainty sampling when the DPD module is activated is controlled by two parameters. The number of DPD bins determines how many samples are generated by the DPD module. The number of realizations in the outer model time loop (set in Step #2, Simulation Settings) determines how many samples are pulled from the DPD generated sample matrix, with each realization using a randomly-sampled DPD sample set. The DPD module is used for importance sampling and to demonstrate the commercial software framework's ability to adapt to different probabilistic methods.

3.1.5.2 Module Options

The xLPR framework module options are changed using the input options as defined on the Main Dashboard and displayed in Figure 3-14 and are labeled as items #5-9.

5. **Crack Initiation:** This pull-down menu allows the user to select between the three (Method I, Method II, and Weibull) alternative conceptual models for crack initiation. There is not option to turn off the crack initiation model provided, however it is turned off automatically for a deterministic calculation.

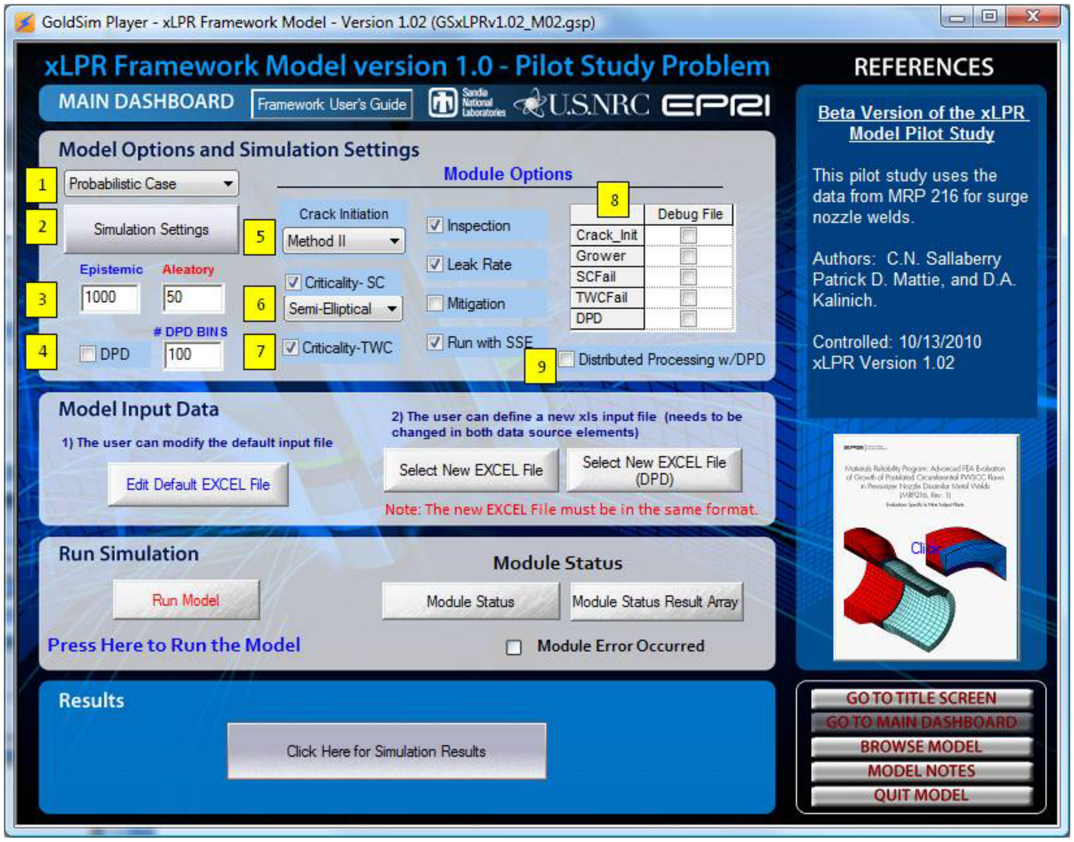
6. **Criticality –SC:** Checking the box activates the SC criticality module. The simulation can be run without the SCFail module by unchecking the box. In this case the simulation will never check the SC stability and a TWC will only occur when the crack depth is greater than or equal to the thickness. The pull-down menu allows the user to select whether SCs are modeled with semi-elliptical or constant depth geometry, two alternative conceptual models described in the documentation of the SCFail module.

7. **Module Activation Check Boxes:** The remaining modules used in the xLPR framework module have only a single module option. Or, in the case of TWCFail, the module automatically selects the most appropriate solution to use between alternative conceptual modules (see the description for TWC Fail in the xLPR Models/Input Task Group Report). Mitigation and Run with SSE are not modules, rather options included in the framework. Each option is described below:
    a. **Criticality – TWC:** Checking the box activates the TWC criticality module. If the TWCFail module is not run, the pipe will not rupture and the TWC will continue to grow until it reaches the maximum half length.
    b. **Inspection:** Checking the box activates the inspection module. If the ISI is not run, then the PND will always be equal to 1.
    c. **Leak Rate:** Checking the box activates the crack leak rate module. If the SQUIRT module is not run, then the leak rate will not be calculated.
    d. **Mitigation:** Checking the box activates the crack mitigation option. The user needs to ensure that when this option is checked that in addition to the default values the input spreadsheet has the following four constant inputs and two uncertain inputs defined:
        i. Constants:
            1. Mitigation time – time in years that the mitigation event occurs.
            2. MTS – should equal 2, the number of intervals in the operation history. One for pre-mitigation and one for post mitigation.
            3. MTS_1 – duration of the first interval in the operating history, pre-mitigation interval.
            4. MTS_2 – duration of the second interval in the operating history, total simulation time – mitigation time.
        ii. Uncertain Parameters
            1. Sig0_wrs_mitigated – Distribution of Sig0_wrs after mitigation.
            2. Xc_mitigated – Distribution of Xc after mitigation.

    e. **Run with SSE:** Checking the box causes the framework to run with the effects of the safe shutdown earthquake included in the calculation. Option to evaluate SC and TWC stability given SSE forces. As described in Section 3.1.2, when this check box is selected, the SCFail and TWCFail modules will be run twice, once using SSE loads and once for Nominal loads. Two separate probabilities are tracked, however, the SSE probability does not account for the probability of the initiating SSE event (this can be applied after post processing the results by weighting the probabilities by the event frequency). In addition, a SC that fails the stability criteria for SSE loads affects the SSE probability for SCFail, but does not impact the TWCFail probability under SSE. This occurs because the status of the crack is only changed due to normal loads. At each time step, the stability of a SC or TWC is checked first for failure under SSE loads, if that has occurred then stability is checked for normal loads. Only if the crack fails under normal loads is the crack status changed (e.g., from -1 to -2 for transition from a SC to TWC, or from -2 to 200 for transition from a TWC to pipe rupture). For SSE the model tracks when a SC would fail, but this does not affect the rupture probability. Therefore, evaluation of SSE in the TWC criticality system only changes the rupture probability for TWCs that have formed under normal loads. Both the SSE evaluations are tracked separately and independently.

8. **Debug Files:** An optional debug file can be written for the following modules by checking its box. See the module documentation for details on the contents of each debug file.
    a. Crack Initiation
    b. Grower
    c. SCFail
    d. TWCFail
    e. DPD

9. **Distributed Processing w/DPD:** Option added only for users who have the GoldSim Pro software license and the ability to run the simulation using multiple processors. The DPD module needs only be run once, on the first realization to create the DPD sample matrix from which the model pulls random sample sets. The default setting is run DPD once to fill this matrix. For distributed process runs the total number of simulations are divided into equal parts and distributed to the available processors. For each group of simulations, the DPD matrix must be filled, and thus the DPD module is run on each processor when this check box is selected.



GoldSim Player - xLPR Framework Model - Version 1.02 (GSxLPRv1.02_M02.gsp)

xLPR Framework Model version 1.0 - Pilot Study Problem

MAIN DASHBOARD

Model Options and Simulation Settings

1 Probabilistic Case

2 Simulation Settings

3 Epistemic 1000 Aleatory 50

4 □ DPD # DPD BINS 100

Module Options

5 Crack Initiation Method II

6 ☑ Criticality- SC Semi-Elliptical

7 ☑ Criticality-TWC

☑ Inspection

☑ Leak Rate

□ Mitigation

☑ Run with SSE

8

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

9 □ Distributed Processing w/DPD

Model Input Data

1) The user can modify the default input file

Edit Default EXCEL File

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

Select New EXCEL File

Select New EXCEL File (DPD)

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

Run Simulation

Run Model

Press Here to Run the Model

Module Status

Module Status

Module Status Result Array

□ Module Error Occurred

Results

Click Here for Simulation Results

REFERENCES

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

GO TO TITLE SCREEN

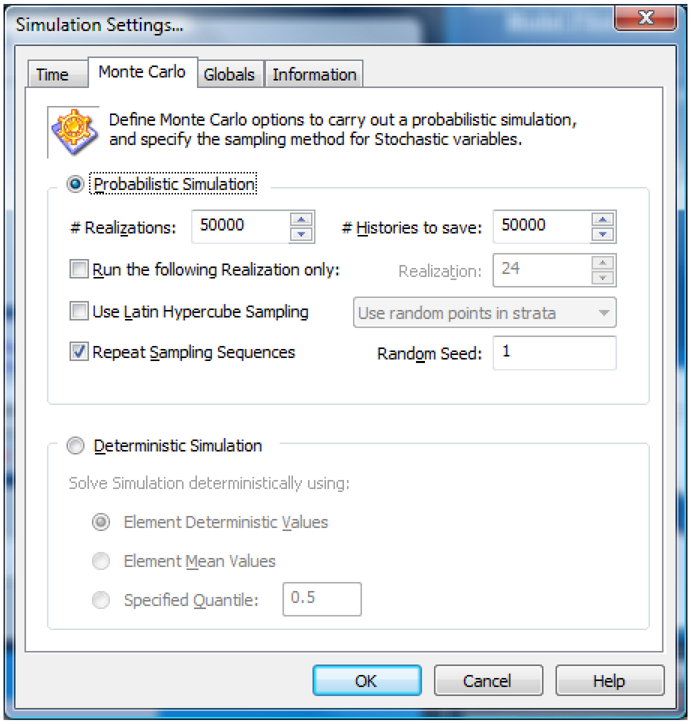
GO TO MAIN DASHBOARD

BROWSE MODEL

MODEL NOTES

QUIT MODEL

Figure 3-14. Model Options and Simulation Settings on the Main Dashboard.



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

● Probabilistic Simulation

Realizations: 50000

Histories to save: 50000

□ Run the following Realization only:

Realization: 24

□ Use Latin Hypercube Sampling

Use random points in strata

☑ Repeat Sampling Sequences

Random Seed: 1

○ Deterministic Simulation

Solve Simulation deterministically using:

● Element Deterministic Values

○ Element Mean Values

○ Specified Quantile: 0.5

Figure 3-15. GoldSim Player – Simulation Settings - Monte Carlo options.

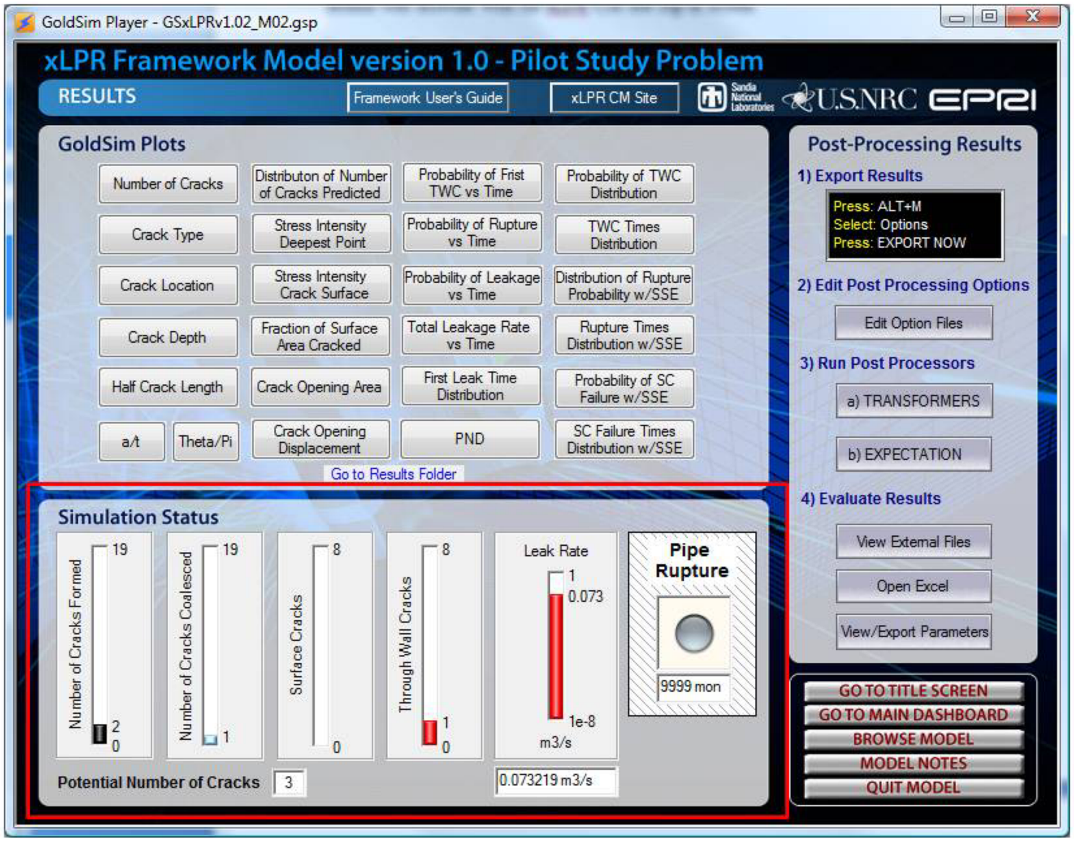
3.1.5.3 Additional Considerations

The GoldSim Pro software is required to change between repeated and non-repeated sampling sequences, activation status of LHS, and to change the random seed. This due to the fact that the GoldSim Player software does not have the capability to adjust the simulation settings for the embedded Epistemic and Aleatory SubModels, where these model options are controlled. In the framework, uncertain parameters defined as epistemic are sampled using LHS while those defined as aleatory are sampled using Monte Carlo sampling. The type can be changed by modifying the parameter in the input data spreadsheet.

3.1.6 Running the Model

To run the xLPR framework model all of the files should be downloaded from a controlled source and the GoldSim Player Software or GoldSim Pro software should be already be installed on the user’s computer. The model file can be opened by double clicking the .gsp or .gsm file or first opening the GoldSim Software and selecting the model file name. The user can then modify the model options. Once the file has been saved, the framework can be run by clicking on the Run button on the GoldSim Run Controller, clicking on the Run Model button on the Main Dashboard Screen, or pressing Alt+m and selecting Run Model from the pop-up menu. If the framework has been previously run, and hence contains results, a pop-up dialog box will appear to let the user know that the existing results will be destroyed if the framework is rerun and ask if the user wants to run again. The simulation will begin with the realization, time step and elapsed time updated continuously on the GoldSim Run Controller. However, if any of the module

DLLs or Excel input spreadsheets are missing, the framework will not run and an error message will pop-up telling the user the name of the missing file. The run controller lets the user run, monitor, pause, and step through a simulation in a controlled manner [38] and is the default interface for the GoldSim Player Software. As a demonstration of the software's capabilities an alternative customized dashboard interface was constructed to monitor xLPR specific conditions. The Results Dashboard, as depicted in Figure 3-16, has a display that is updated at each time step with several status indicators.


GoldSim Player - GSxLPRv1.02_M02.gsp
xLPR Framework Model version 1.0 - Pilot Study Problem
RESULTS
Framework User's Guide
xLPR CM Site
Sandia National Laboratories
U.S.NRC
EPRI
GoldSim Plots
Number of Cracks
Distributon of Number of Cracks Predicted
Probability of Frist TWC vs Time
Probability of TWC Distribution
Crack Type
Stress Intensity Deepest Point
Probability of Rupture vs Time
TWC Times Distribution
Crack Location
Stress Intensity Crack Surface
Probability of Leakage vs Time
Distribution of Rupture Probability w/SSE
Crack Depth
Fraction of Surface Area Cracked
Total Leakage Rate vs Time
Rupture Times Distribution w/SSE
Half Crack Length
Crack Opening Area
First Leak Time Distribution
Probability of SC Failure w/SSE
a/t
Theta/Pi
Crack Opening Displacement
PND
SC Failure Times Distribution w/SSE
Go to Results Folder
Simulation Status
Number of Cracks Formed
19
2
0
Number of Cracks Coalesced
19
1
Surface Cracks
8
0
Through Wall Cracks
8
1
0
Leak Rate
1
0.073
1e-8
m3/s
Pipe Rupture
Post-Processing Results
1) Export Results
2) Edit Post Processing Options
Edit Option Files
3) Run Post Processors
a) TRANSFORMERS
b) EXPECTATION
4) Evaluate Results
View External Files
Open Excel
View/Export Parameters
GO TO TITLE SCREEN
GO TO MAIN DASHBOARD
BROWSE MODEL
MODEL NOTES
QUIT MODEL

Press: ALT+M
Select: Options
Press: EXPORT NOW

9999 mon

Potential Number of Cracks 3

0.073219 m3/s

Figure 3-16. Results Dashboard with Simulation Status monitoring highlighted.

The Simulation Status on the Results Dashboard can be monitored by the user if the framework is run from the Results Dashboard (e.g., the run is started by the user clicking on the GoldSim Run Controller Run button while the framework is on the Results screen). Indicators are provided for: number of cracks formed, the number of cracks coalesced, the number of SCs at what time, and the maximum number of cracks that could occur as predicted by the crack initiation module (Note that the overhead associated with displaying the Simulation Status results causes the framework to run slower).

During the simulation the framework model has an indicator box on the framework's Main Dashboard screen that is updated during each realization and when checked indicates at least one module had at least one error during the course of the run. To monitor over all of the realizations, after the simulation has completed there are two buttons available (**Module Status** and **Module**

Status Result Array) that when clicked provide the user with plot/tabular results of error flag values for the following modules (see Figure 3-14). No error is indicated by an error flag value of 0. Errors are indicated by non-zero error flag values.

After the simulation has completed a pop-up dialog box will appear. At this time it is highly recommended that a user save the model run before browsing the results or conducting any post processing analyses. The current version of the GoldSim Player software does not have the option available in GoldSim Pro to automatically save the file after the model run has completed. Using the Player Software, the lower left-hand button on the right side of the Run Controller (the **Options** button) provides access to a menu to allow you to save the current file.

Whenever a model is run, a Run Log is produced. The Run Log contains basic statistics regarding the simulation (e.g., the version of the GoldSim, the date, the simulation length), and any warning or error messages that were generated by the software. The xLPR framework has been constructed to write a message to the software's default run-log when an xLPR module has an error. The user should inspect the GoldSim run-log for warning and/or error messages. The run-log can be accessed by pressing "Alt+m" and selecting **View Run Log** from the pop-up menu. The run-log will then be displayed the default test document viewer (it is saved as a text file: GoldSim Run Log.txt, in the directory containing the framework GoldSim file).

3.1.7 Framework Model Capabilities and Features

Table 3-16 is a tabular comparison of the availability of selected GoldSim software features and xLPR model capabilities between the GoldSim Pro Software and the free GoldSim Player Software. The list is not extensive and does not attempt to be, however, it does list some key capabilities and features that are discussed in the model description in Section 3.1. It is envisioned that the xLPR model end users will predominately use the Player version, and thus these limitations need to be considered.

Table 3-16. Comparison of GoldSim Software: xLPR Model Capabilities and Software Features.

Description of Capability or Feature	GoldSim	
	Pro	**Player**
Graphical User Interface	**x**	**x**
Modify or change time step sizes or number.	**x**	**x**
Change the number of realizations (aleatory, epistemic, total)	**x**	**x**
Run one realization, select specific realization to run	**x**	**x**
Switch Between Probabilistic and Deterministic Simulations	**x**	**NA**[1]
Change Random Seed and Activation Status of LHS	**x**	**NA**
Select new or modify data in the Excel input spreadsheet	**x**	**D**
Run, Pause or Abort Model	**x**	**x**
Run on multiple cores (Distributed Processing locally or over a LAN)	**x**	**NA**
Automatically Save Model File after run has completed	**x**	**NA**
Add or Modify a spreadsheet interface	**x**	**NA**
Swap DLL modules (e.g., replace existing modules)	**x**	**x**[2]
Change or Add New Modules, Modify Input/Output arrays and DLL options	**x**	**NA**
Change Parameters Uncertainty Classification (from Epistemic to Aleatory)	**x**	**x**[3]
Change Uncertain Parameter Distribution type (e.g., Uniform to Normal)	**x**	**NA**
Modify input data and uncertain distributions (excluding distribution type)	**x**	**x**[3]
Connect and download inputs and distributions from dB using ODBC	**x**	**x**
Change parameter types, distribution types, values, and units in a model file using a ODBC connected dB	**x**	**x**
Automatically or Manually Export Data as ACSII text files	**x**	**x**
Automatically export data to Excel	**x**	**x**[4]
Save and Plot any data from any element within the model file	**x**	**x**[5]
Modify and Save Plots	**x**	**x**
Modify, Add, or Replace model elements to change or enhance framework model logic and features	**x**	**NA**
Model versioning and version reporting	**x**	**NA**
Automatic Generation of a Model Run Log for Version and Error Tracking	**x**	**x**
Browse Model file (use Function of and Affects views)	**x**	**x**
Screen Realizations	**x**	**x**
Modify Internal Array Sizes or Add New Array definitions	**x**	**NA**
Create or Modify Dashboard Interface	**x**	**NA**
Add Graphics, Hypertext links, Text, to describe and reference model file	**x**	**NA**

Table footnotes:

NA – Indicates that the feature or capability is not available.

x - Indicates that the feature or capability is available in the software version

D - Indicates that the feature or capability is only available using a dashboard element. Dashboard elements have to be added by a developer using GoldSim Pro.

[1] – The use of SubModels for the Epistemic and Aleatory loops precludes this feature that would normally be available using the Player Software.

[2] – DLL modules can only be swapped if the new DLL has the same file name, function name and identical input/output arrays.

[3] – Using the predefined spreadsheet interface, values can be changed in the Excel spread sheet.

[4] – A spreadsheet element is used to export the data from GoldSim to Excel. Can only be added and defined using the GoldSim Pro software, but can be utilized by the Player Software. This feature has not been used in the xLPR Framework Version 1.0.

[5] – The player software cannot activate or deactivate elements to be saved.

3.2 Post-Processing

As discussed above, a set of post-processing tools have been developed as standalone software to estimate some output variables (such as probability of rupture) with inspection or leak detection, without having to rerun the framework model, while accounting for the separation of aleatory and epistemic uncertainty or importance sampling. The framework model was developed for maximum flexibility and to accommodate the post processing tools. As such, the post processing tools can be accessed either from the xLPR model framework through the Results Dashboard interface (Figure 3-16) or through a simple command prompt. The dashboard offers links to documentation tips for post processing which provide a convenient interface when analyzing the results. A more detailed description of this interface is included in the xLPR Framework User's Guide [41]. The following discussion focuses on the implementation of the post processing features that are included in the standalone software; detection (TRANSFORMERS) and expected value over the uncertainty (EXPECTATION).

Two types of detection are considered in this version. One is leak detection and the other is regular inspections of weld. One important assumption in this analysis is that once a crack is detected, the part is replaced in a way that it will not fail again for the remaining lifetime (i.e., the end of simulation).

3.2.1 Correction through Leak detection

It is assumed that if a TWC leads to a leak beyond a threshold value (specified by the user), the leak will be noticed and repaired.

As the leak rate is saved during the simulation and since we consider that a replaced part will never fail again, this correction can be done without rerunning the model. The analysis is extended until pipe rupture. Since at each time step the leak rate is saved, it is easy to perform a posterior check to determine if a crack will be detected at selected leak rates. In the post-processing tool TRANSFORMERS, the user is required to indicate the leak rate threshold. Based on this value, the code will generate an indicator function (a set of 1 and 0) using leak rate history for each realization. This function will be equal to 1 as long as the rate is below the threshold and set to 0 after.

This indicator function is then used as a multiplier to any variable of interest, at each time step and for each realization, setting the variable to 0 if a leak was detected. **It is important to use this method only if it makes sense to have a variable set to 0.** For most of the variables of interest (probability of leak, probability of failure, fractional surface area damaged, depth and length of cracks) this method will be appropriate. But if a reference value was different than zero in absence of cracks, then this method would not be appropriate. For example, if a user wanted to output the flow rate in the pipe (to study the variation of flow rates), it could not consider it to be zero when there is no crack. A reference flow rate should then be used.

3.2.2 Correction through inspections

Regular inspections will help find SC (or even TWC leading to leak rate below the threshold value) before they cause rupture. With some assumptions, it is possible to implement such correction as a post-processing analysis.

Some of the assumptions are identical to those made for leak rate detection. We suppose that once a crack is detected, the part will be replaced so that it will never fail for the duration of the simulation. We only analyze variables whose value would be equal to 0 if there was no crack.

A new assumption is made on the PND, regardless of the leak rate a TWC will always be detected at the time of an inspection.

For each occurring crack, the framework estimates a PND using the ISI module. Several options have been considered 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.

In the presence of multiple cracks, the user can select whether the probability of detecting a crack is independent of the presence of the other cracks. If each crack is considered independently, then 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 SC is such that if you miss the most obvious crack, then you will miss the others also. In this case, it is more appropriate to use the minimum over the probabilities of non detection (this minimum being associated with the easiest crack to find) than to use the product.

With respect to the efficiency of subsequent inspection, we considered that the user could either want to have them independent (so the probability of not detecting a crack at time T_1 and then at time T_2 is equal to the product of probability of not detecting them at T_1 times the probability of not detecting it at time T_2) or want some dependency, such that if a SC was not detected the first time, it is unlikely to be detecting again (unless it becomes a TWC). This latest 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.

Once the options have been selected, the post-processing code will construct a correction matrix (as it was done for leak detection). The difference being that this matrix will not be made with a set of 1 and 0 but will also include values between 0 and 1 representing probabilities of finding a crack during the inspection.

3.2.3 Calculations of uncertainty analysis and estimate of statistics

The post-processing code EXPECTATION is used to calculate expected value of the output over aleatory uncertainty for each time step, complementary cumulative distributions functions (CCDF) and associated statistics (see Section 3.3.1 for a description).

When a simple Monte Carlo method is used for sampling aleatory uncertainty the calculation of expected value of the output is a simple average.

$$E(output) = \sum_{i=0}^{n} \frac{output_i}{n}$$

If importance sampling is used, then the weight associated to each simulation must be used in order to calculate the expectation of the output correctly

$$E_{imp}(output) = \sum_{i=0}^{n} output_i.weight_i$$

Once expected value is estimated for each epistemic set, mean over epistemic uncertainty is calculated for the expected output over aleatory uncertainty (with or without importance sampling). Quantiles are estimated using sorting techniques and "counting" (with equal or unequal weight) up to the desired value.

3.3 Presentation of Model Results

3.3.1 Format of Results

The last step of a probabilistic approach consists in analyzing the results and drawing conclusions. This step is called uncertainty analysis and sensitivity analysis.

Several uncertainty and sensitivity analysis techniques are presented in Helton et. al. (SAND2006-2901:http://prod.sandia.gov/techlib/access-control.cgi/2006/062901.pdf).
Uncertainty analysis refers to the determination of the uncertainty in the analysis result that derives from the uncertainty in analysis inputs. This corresponds essentially to a statistical analysis of the set of output resulting from the sample created by the framework.

Most of the results presented in Section 5 will be based on uncertainty analysis, as mean values and statistics will give more insights than results from a single realization. Since separation of aleatory and epistemic uncertainty is considered, then results can be presented in different ways.

One way is to plot the output of interest as a set of CCDFs at a selected time step. Each CCDF (representing the effect of aleatory uncertainty) gives an answer to the Kaplan and Garrick risk triplet questions:

1) What can happen in the future?

2) How likely it is to happen?

3) What are the consequences if it happens?

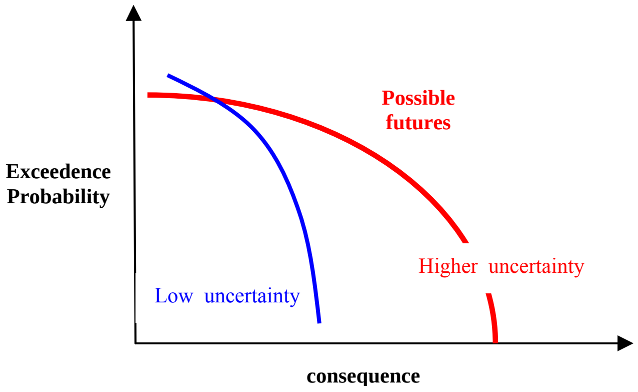


Figure 3-17. Consequence vs. Exceedence Probability

A steep curve will represent a low aleatory uncertainty (as most of the values are close to each other) while a broad distribution of consequences will represent large aleatory uncertainty (as the values vary significantly).

A set of these CCDFs (representing the effect of epistemic uncertainty) represents our state of knowledge on this "risk."

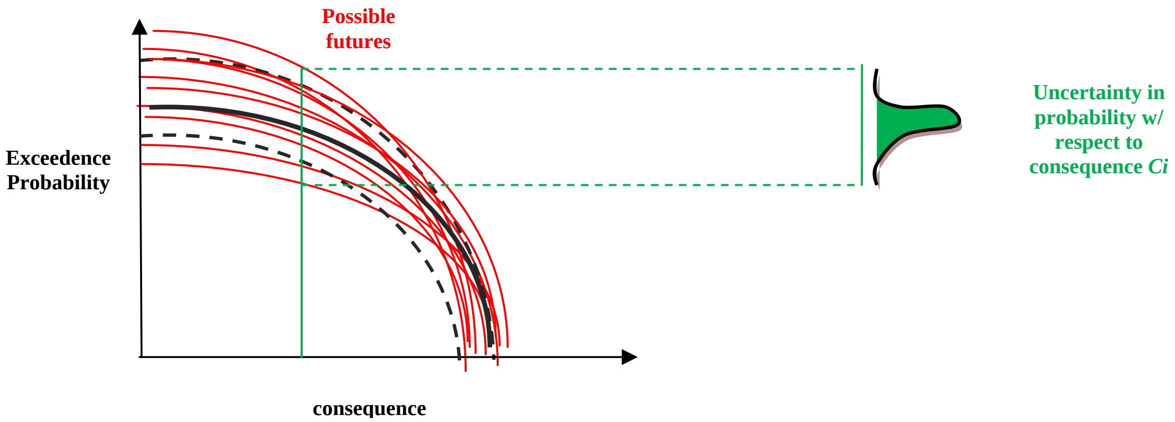


Figure 3-18 Uncertainty in Exceedence Probability with respect to consequence.

If the CCDFs are close to each other, then the epistemic uncertainty is small (and reducing it will not result in much improvement). If they are spread, then the epistemic uncertainty is large and one can gain accuracy if it is reduced.

While the display of these curves help understanding which part of the uncertainty is due to (non controlled) risk and which part is due to lack of knowledge, it is hard to visualize the variation over time, or even to compare one scenario to another. Moreover, this kind of representation gives a good qualitative representation of likelihood of having a rupture, but does not give directly the probability of rupture.

The risk is often represented as an expected value (or, in other words average) over aleatory (as aleatory uncertainty cannot be controlled). When considering rupture, this expected value will become equivalent to estimate a probability of rupture. In order to estimate this expected value, each CCDF is integrated individually. This operation leads to several estimates at each time step representing the (epistemic) uncertainty we have on the estimate of this expected value, due to lack of knowledge. This result can be represented as a function over time.

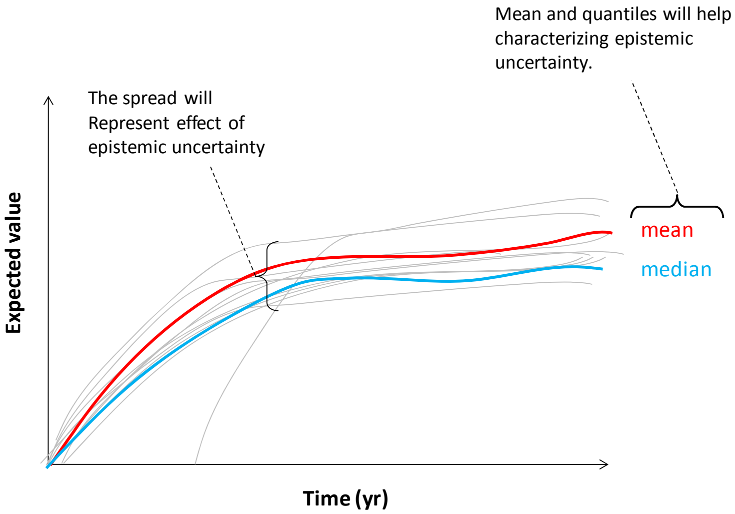


Figure 3-19. Epistemic uncertainty.

The spread of the multiple time-dependent curves represents the effect of epistemic uncertainty, the same way the spread of CCDF does. Often statistics such as mean and quantiles (median and more extreme quantiles) are included in order to give a better visualization of this uncertainty and simplifies comparison between two scenarios.

The quantile will summarize the effect of epistemic uncertainty in input on the output of interest. In many of the presented results, the 5th percentile and the median will be equal to zero. This is not an accuracy problem of inappropriate sample size, but a real and correct estimate of the distribution given the inputs. For half of the simulations performed (when importance sampling is not used) the input set is such that no crack will occur during the simulation duration. Within the range of the distributions selected and the models used, it is not possible to have a crack half of the time, therefore, the median is (correctly) estimated to be equal to 0. Increasing the sample size will not change this value, as one will still have half of the realization equal to 0.

When low probabilities are estimated, the accuracy of the estimate and its stability can be questioned. This problem mainly deals with the accuracy of the numerical method selected and the parameters of this method (i.e., simple Monte Carlo vs. LHS vs. importance sampling, sample size, choice of input parameters and of importance sampling applied to these parameters, etc.). For answering this question, two approaches are used and applied to probability of rupture. For LHS, replicated analyses are performed (using different random seeds). Mean and quantiles results are then compared. For importance sampling using DPD, bootstrap method has been used to generate a distribution of mean probability of rupture. Bootstrap method is not used on original LHS as it would not be appropriate to break the dense stratification.

3.3.2 Selected results

Among the many results saved by the framework, a few have been considered the most meaningful.

- First crack occurrence gives the probability of first crack occurrence over time.
- Fractional Surface Area (FSA) cracked shows the evolution of the fraction of surface area cracked. This value has the advantage over leak rate to incorporate SCs and the advantage over probability of first crack to look at the importance of each crack.
- First Leakage Occurrence (FLO) gives a summary of the probability of leakage over time.
- Total Leakage Rate (TLR) indicates the magnitude of the leak rate and its expected value over time.
- Critical Failure Occurrence (CFO) gives a summary of the probability of rupture over time.

Most of the scenarios will be compared to each other using statistics over the former output.

3.4 Presentation of Sensitivity Analysis Results

Sensitivity analysis refers to the determination of the contributions of individual uncertain analysis inputs to the uncertainty in analysis results.

To quantify and rank the importance of the uncertainty of each uncertain input on the variance of the output of interest, linear and rank regressions seem the most appropriate. As rank regression often gives better results than linear regression, for essentially the same computing cost, Standardized Rank Regression Coefficients (SRRCs) are used in place of their parametric linear equivalent. The coefficient of determination of the regression model (R^2) will inform of the quality of the regression and consequently of the quality of the sensitivity analysis. A previous approach that seems to be the most appealing graphically was estimating the importance of some parameter in a stepwise fashion (i.e. using stepwise regression) for non time-dependent parameters and at a specific time step for time-dependent parameters.

In order to see stability of the sensitivity analysis over time, stepwise regressions will be applied at four different time steps: 10 yr, 30 yr, 50yr and 60yr.

Listed for each time step are 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. 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 output) while a negative SRRC denotes a negative relation (for which high values of input are associated with low values of output and reciprocally).

Most of the results at early time (10 yr) will present low R^2 as most of the results are still equal to 0 (few existing cracks).

In the pilot study, it seems reasonable to use simple and reliable methods. Scatterplots of output vs. input remains one of the simplest and most useful techniques and complete the classical stepwise regression in a more qualitative but graphically appealing way. When two parameters are leading the sensitivity analysis, instead of a classical scatterplot, 3D contour plots can be used, showing the conjoint influence of both of these input parameters.

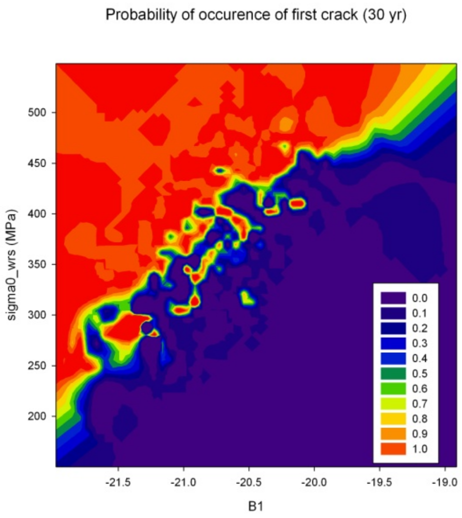


Figure 3-20. Probability of occurrence of first crack (30 yr).

The example in Figure 3-20 shows how the results are displayed. One axis is used for one parameter, and the other for the second parameter. The output of interest is displayed with color coding.

3.5 Summary of Presentation of Results

- CCDFs show how much variation is due to aleatory (spread within a CCDF) and epistemic (spread between CCDFs) – due to the separation of aleatory and epistemic.
- Sensitivity analysis shows you which epistemic parameters are the most important in terms of sensitivity of the results (all probability) – this is done via LHS (aleatory uncertainty averaged or sample size taken equal to 1).
- Importance sampling gives you a more accurate estimate of the mean of the output distribution (mean probability) (LHS helps determine importance sampling and gives an estimate in the same order of magnitude but more accurate results will require very large sample size)
- Using the Bootstrap method over importance sampling tells you how stable the mean estimate is.

4. xLPR Model Verification and Stability

Verification and validation (V&V) are two important components of any analysis and are intimately connected with the assessment and representation of uncertainty. Verification is the process of determining that a model implementation accurately represents the developers' conceptual description of the model and the solution to the model. Validation is the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model (p. 3, [22]; [23-28]). Specifically, validation involves determining if appropriate models are in use, and verification involves determining if an analysis has been implemented correctly.

Topics that must be considered in verification include (i) programming errors, (ii) data input errors, (iii) convergence of numerical procedures, (iv) appropriateness of sampling procedures and sizes in use, (iv) potential errors in the archival storage of analysis results, and (v) many other potential sources of error. Validation involves the comparison of model predictions with relevant real-world or experimental results. Appropriately performed V&V are necessary components of a credible analysis.

The sampling-based sensitivity analysis procedures described in Appendix D.2 are powerful tools for checking for analysis errors and thus is an important component of analysis verification. Further, model validation is an important contributor to the insights that ultimately lead to appropriate characterizations of epistemic uncertainty.

This section provides an overview of the key aspects of the xLPR framework model and its verification steps, including specific actions to be taken to enhance confidence and demonstrate stability and reliability of the statistical aspects of the numerical model. The documentation for the verification process is contained on the xLPR CM SharePoint Site.

The GoldSim software is a commercial product, and as such, has been verified by the software vender.

4.1 Module and Input Verification

Converting the numerical model to a set of computer code algorithms is a process that must be transparent and traceable. Links from the numerical model to the computer code were documented to permit easy checks on input construction. All inputs were checked, controlled, and documented. Computer code verification itself, to ensure that the code implements the numerical model correctly, is controlled in the CM process.

The modules are verified and eventually will be qualified (not for the pilot study) by their developers in the xLPR Models Task Group, both from the standpoint of being correct representations of their underlying conceptual and mathematical models, and in terms of their mechanical operation as a standalone module. It was required that the xLPR Models Task Group follow the CM process established for this effort as described in Section 1.5. The CM used during the development phase provides for documentation that the computer code is verified thoroughly and that the input construction is error-free. This process of verification included an independent check of the source code and verification that the test cases when run resulted in the expected response. In addition, a technical review was completed by a SME to ensure that the supporting scientific process models and/or abstraction models used in the xLPR modules were appropriate for their intended use. The documentation for the process of the selection of the modules will be included in a separate document written by the xLPR Models Task Group, and not in this xLPR model report.

Modules linked to the xLPR framework model are implemented as DLLs, which are separately compiled and linked modules or subroutines that are called by the xLPR Framework (e.g., GoldSim). As such the CM used during the initial module development phase was repeated to document that the modified source code and compiled DLL were verified thoroughly. This process of verification included an independent check of the source code and verification through standalone GoldSim test case(s). The verification test cases confirmed that when the DLL was run attached to GoldSim, the results matched the expected response and that the information transfer between the GoldSim software and DLL is verified. Since the ability to properly call DLLs is a feature for which the GoldSim is qualified, DLLs verified as previously described will, by default, be qualified to be called from within the xLPR GoldSim model file.

Each test case was independently checked by an xLPR computational team member not involved in the DLL development. The verification case was re-run and results were compared with the test cases provided by the original module developer. The documentation (checklists, test cases, and results) of the verification of each of the DLL modules used in Version 1.0 of the xLPR code is contained on the xLPR CM SharePoint Site.

Inputs to the xLPR model are controlled. Parameter values were entered in the xLPR framework model through a Microsoft® Excel Spreadsheet interface element. The parameters in the xLPR framework model were checked and verified during the model framework verification steps discussed below and the input data was developed and verified by the xLPR Inputs Task Group. The verified spreadsheet was added to the controlled files stored on the xLPR CM SharePoint Site. The documentation for the process of selection of the input parameters and data, including their uncertainties, will be included in a separate document written by the xLPR Inputs Task Group, and not in this xLPR model report.

4.2 Model Framework Verification

The xLPR model is comprised of a linkage of many model components. The supporting Models Task Group report documents the verification of the underlying process models and/or abstraction models. In principle, each of these models is already verified before being integrated with the xLPR framework model. In practice verification of a complex computer model of a physical system involves a series of steps taken during and after the development of the model, designed to generate and enhance confidence in the predictions of the model. During xLPR model development the xLPR model was modified in incremental versions that were sized to facilitate checking of the model with the judicious use of model snapshots or model freeze versions, resulting in the completion of the xLPR Alpha and Beta Model versions. The incremental checking and verification testing process used during both the Alpha and Beta model iterations is recorded in the CM process and provides the foundation for xLPR Framework model verification. CM draft desktop guide XLPR-DSK-00 (included in the controlled files directory of the xLPR CM SharePoint site), describes the methods for documenting and checking modeling activities (development and production runs) for the xLPR Pilot Study Program. Documentation includes preparing a conceptual description of a model change and recording specific model changes on a checklist. Checking involves reviewing both the conceptual approach to assess whether model results accomplish their stated purpose, and model implementation to assess whether the specific changes to the model were correctly performed. Each change documented on the change checklist was hand verified by a checker. In addition, the checker confirms that the changes didn't introduce unintended effects. When applicable, the model was run and the results were compared with the previous version or to standalone models used to verify the module implementation to determine if the changes had the desired effect (if any). In addition, the model Run Log was checked for error flags that signal that individual xLPR modules may be exposed to an invalid or out of range condition (module error status is recorded for all modules and available for review through the Main Dashboard interface). In addition, the run log was checked for error messages generated by the GoldSim software.

Prior stages of verification, the input construction, the coupling of the model components, and the internal data transfers are all demonstrated and documented to be correctly handled in the xLPR model. The final phase involves comparing model predictions with independently collected data and/or against results from independent models. For the xLPR pilot study this option was conducted by comparison of the results of two deterministic calculations (presented in Section 5.0) as a demonstration of the validity of the results. However, it is not intended that the pilot study produce a qualified model and therefore may not accurately predict the xLPR behavior. However, the verification steps ensure that the product is correct within the limitations, assumptions, and range of validity defined for the xLPR pilot study model.

4.2.1 Framework Verification Test Cases

Two deterministic test cases were run as defined in Appendix E to verify that the framework and coupled modules are working as expected. The deterministic test cases are simplified scenarios that use constant inputs for all variables. The results of these test cases can be verified against an independent hand calculation using an Excel spreadsheet and independently against the ORNL framework code. The results of these verification analyses will be used and discussed in the downstream documentation (Section 1.3) specifically, the xLPR Version 1.0 computational group report.

4.2.1.1 Deterministic Test Case #1.

In the first deterministic test case analysis, a single crack occurs at 0 years. The location of the crack is at the top of the weld (theta = 0 rad). The input deck for this case is included with the controlled version of the inputs spreadsheet for xLPR. There is no mitigation. The results are presented for the ratio of crack area over the weld thickness (a/t), the ratio of the crack location (theta) over Pi, and for the ratio of the stress intensity over a/t, for both the stress intensity at the crack edges (K(0)) and at the crack's deepest point (K(90)) (Figures 4-1 through 4-4). The crack begins at time zero as a SC and grows until 58 months when it becomes a TWC. The crack continues to grow until rupture at 78 months. The results of this analysis compare closely with the results from the ORNL analysis (SIAM). The differences will be addressed in the NRC's xLPR Version 1.0 computational group report, where a detailed comparison of the two codes is provided.

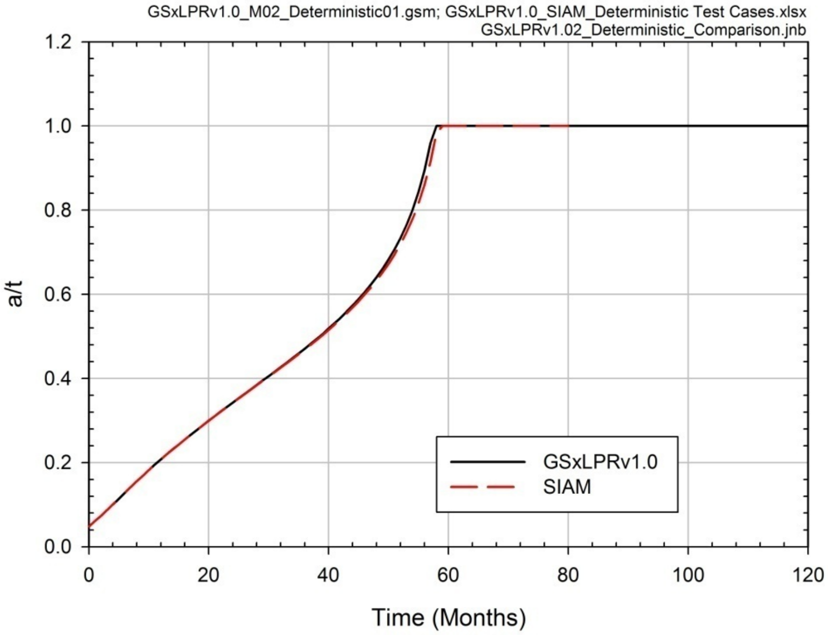


Figure 4-1. Ratio of the crack area over the weld thickness (a/t) versus time for Deterministic Test Case #1.

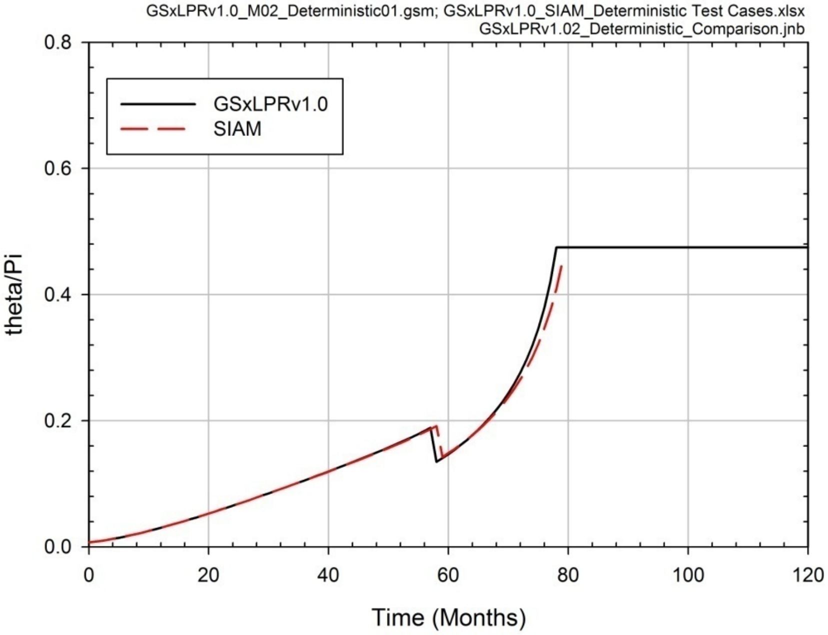


Figure 4-2. Ratio of the crack location over the Pi (theta/Pi) versus time for Deterministic Test Case #1.

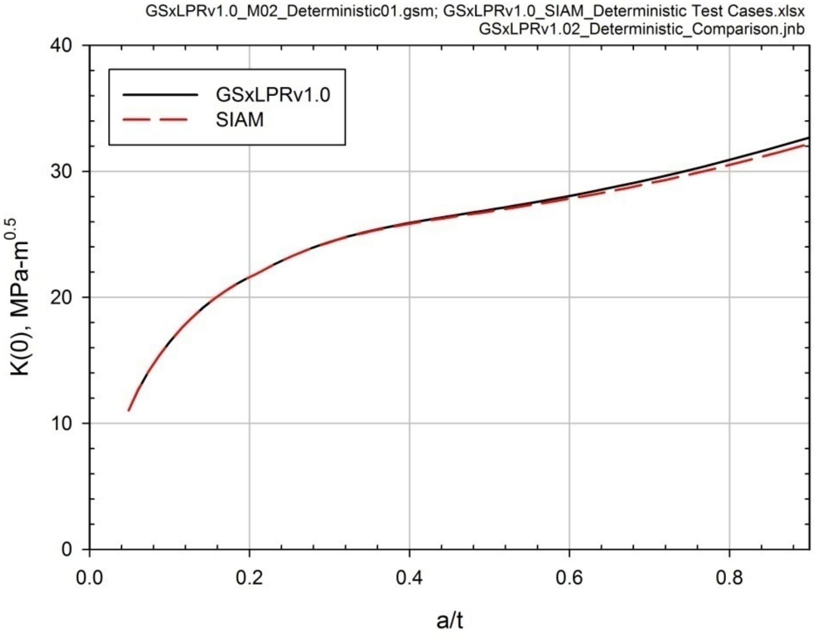


Figure 4-3. **Stress intensity at the crack edges – K (0) versus a/t for Deterministic Test Case #1.**

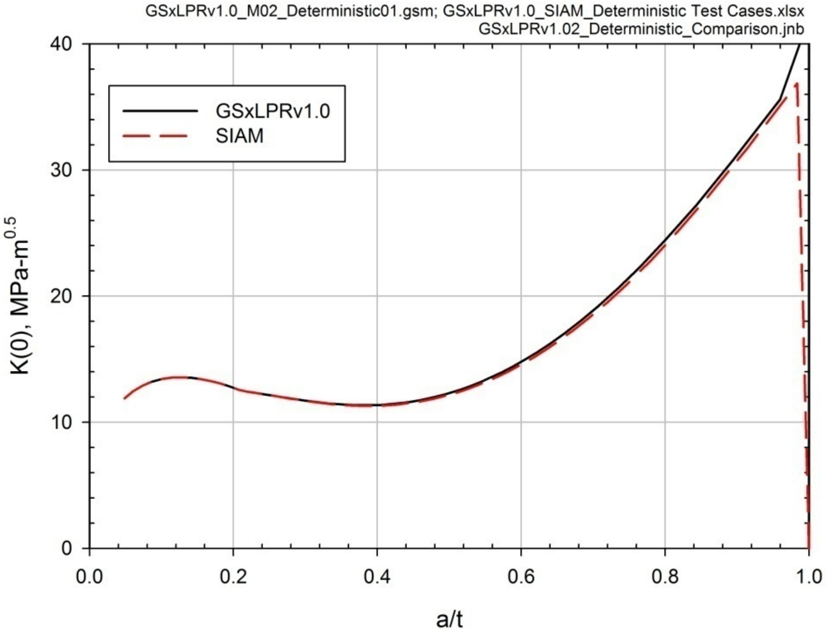


Figure 4-4. **Stress intensity at the crack's deepest point – K (90) versus a/t for Deterministic Test Case #1.**

4.2.1.2 Deterministic Test Case #2

In the second deterministic test case analysis three cracks occur at 0 years. The location of the cracks are theta [1] = 0 rad, theta [2] = 0.6 rad, and theta [3] = -1 rad. The input deck for this case is included with the controlled version of the inputs spreadsheet for xLPR. There is no mitigation. The results are presented for the ratio of crack area over the weld thickness (a/t), the ratio of the crack location (theta) over Pi, and for the ratio of the stress intensity over a/t, for both the stress intensity at the crack edges (K(0)) and at the cracks deepest point (K(90)) (Figures 4-5 through 4-8). The cracks begin at time zero as SCs and grow until 32 months when crack 1 and 2 coalesce. At 52 months crack 1 becomes a TWC. At 62 months, crack 1 and 3 coalesce. The rupture occurs in the next time step at 63 months. The results of this analysis compare closely with the results from the ORNL analysis (SIAM). The differences will be addressed in the NRC's xLPR Version 1.0 computational group report, where a detailed comparison of the two codes is provided.

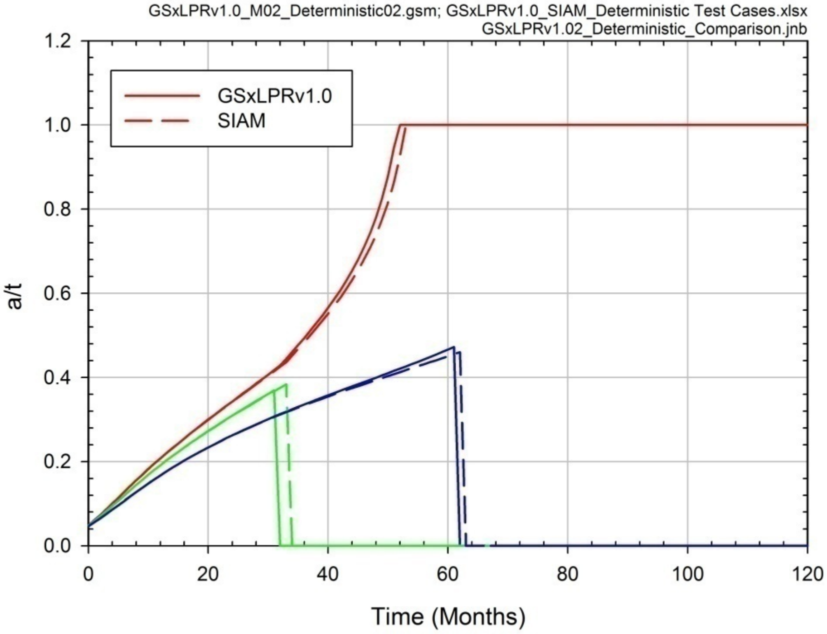


Figure 4-5. Ratio of the crack area over the weld thickness (a/t) versus time for the three cracks in Deterministic Test Case #2.

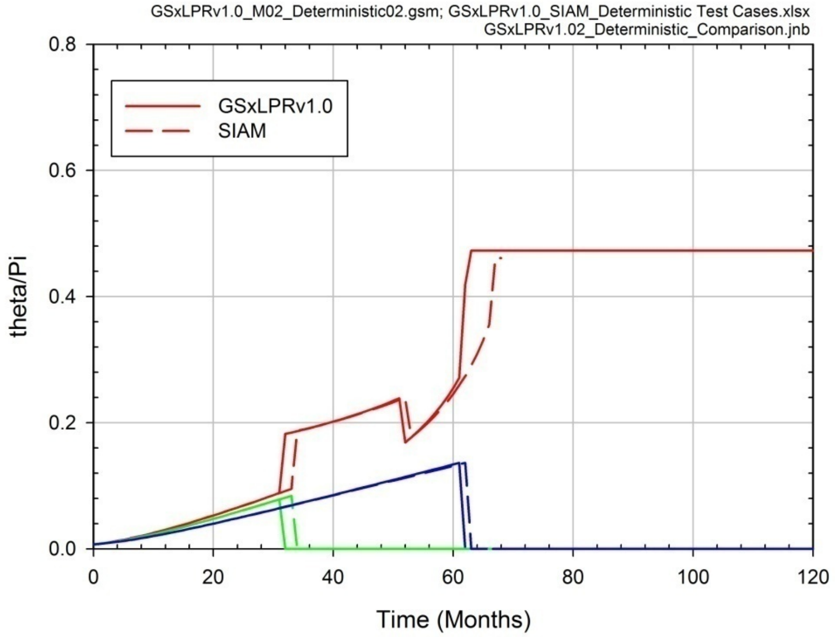


Figure 4-6. **Ratio of the crack location over the Pi (theta/Pi) versus time for the three cracks in Deterministic Test Case #2.**

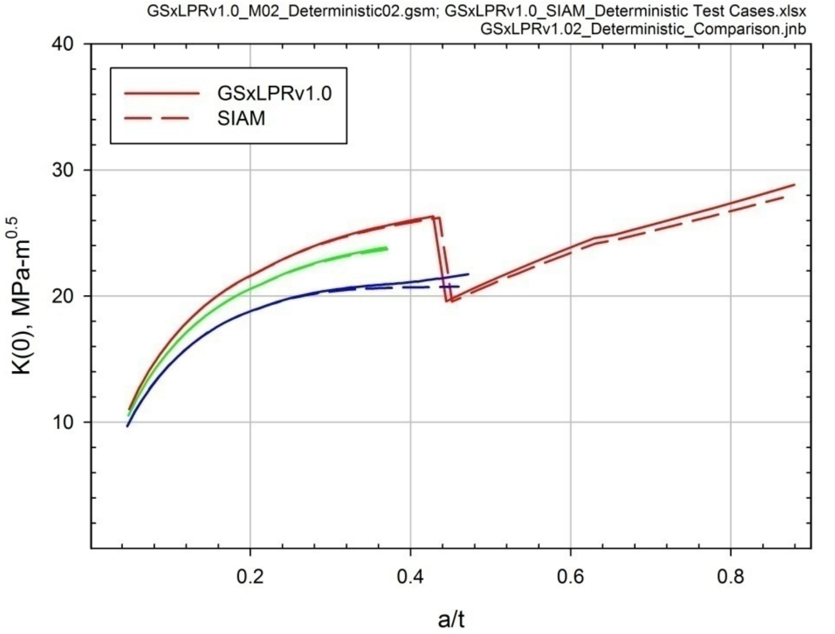


Figure 4-7. **Stress Intensity at the crack edges – K (0) versus a/t for the three cracks in Deterministic Test Case #2.**

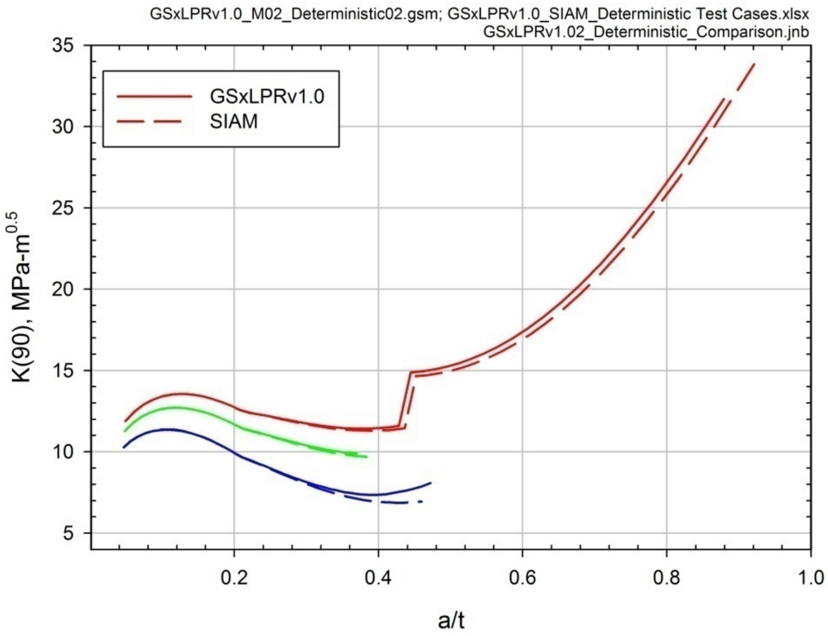


Figure 4-8. Stress Intensity at the cracks deepest point – K (90) versus a/t for the three cracks in Deterministic Test Case #2.

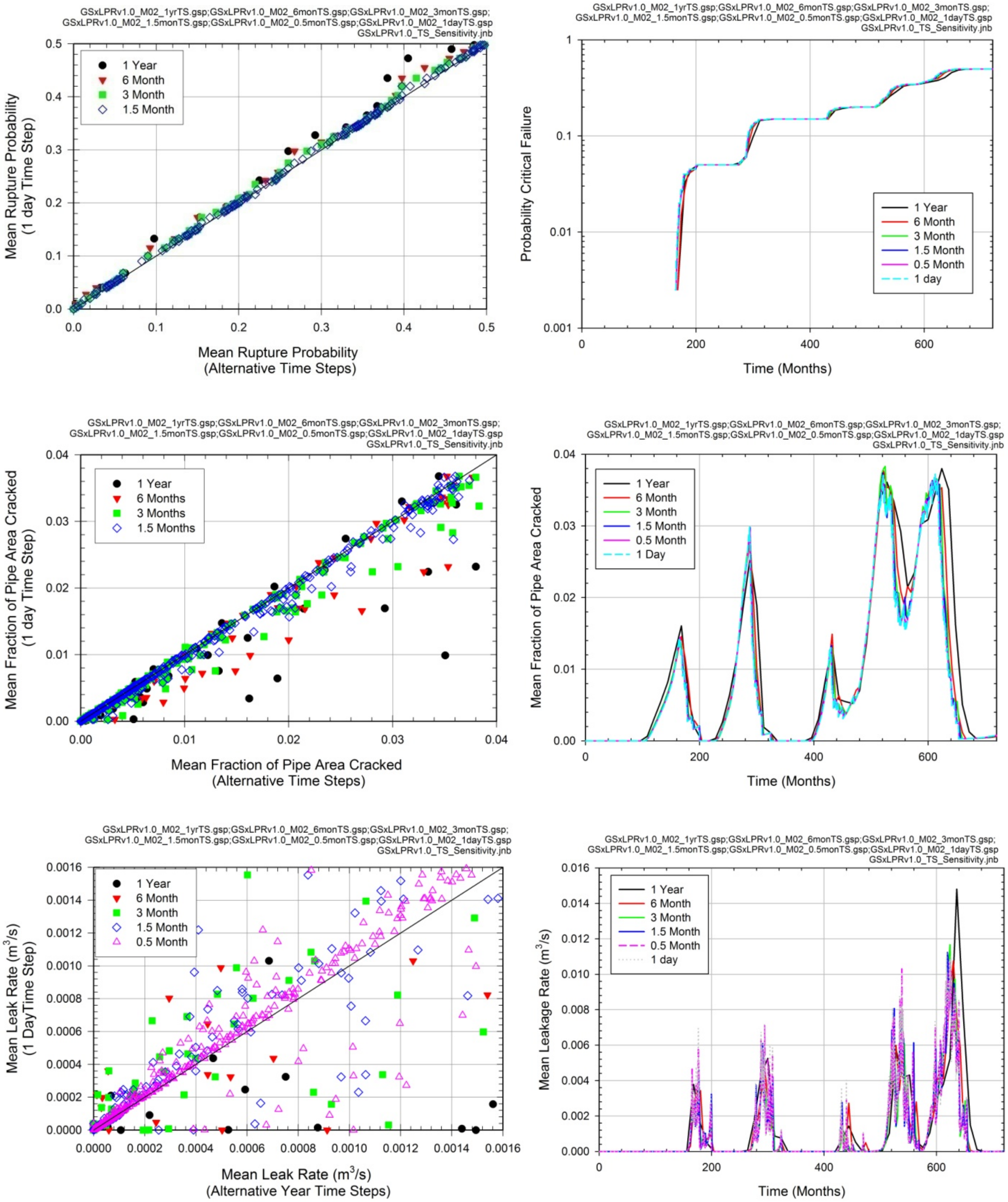
4.3 Model Stability

Other conditions specific to the xLPR model, such as temporal and stochastic discretization, convergence, and stability were checked as part of both development and post-development activities. These xLPR model calibration activities are documented in the sections below.

4.3.1 Temporal Stability

The numerical solution involves uses discrete time steps which are referred to as the temporal discretization. Temporal discretization may affect the accuracy of the solution of the module equations and, thereby, affect the results of the xLPR model. Additionally, large time steps may miss or average important transient peaks. The xLPR base case (Version 1.0) without mitigation with 20 epistemic realizations and 20 aleatory realizations using Monte Carlo and LHS was selected for this analysis. The small sample size was determined to be sufficient since the temporal physical processes, crack initiation, crack growth, leakage, etc., are all represented in the smaller sample size. Several different model runs were conducted to evaluate the potential for variability in model output due to time step size. Six cases were run with time step sizes of 1 year, 6 month, 3 month, 1.5 month, 0.5 month, and 1 day.

Three xLPR model results, probability of rupture, fraction of the surface area cracked and leakage rate (m^3/s) were evaluated and are presented in Figure 4-9. For the probability of rupture, the time history results over the six cases yield approximately the same mean behavior. When plotted on a scatter plot versus a 1 day time step, for each time step case the probability of rupture results show the majority of time points with a good correlation to the smallest time step case. Only a few of the mean values for 1 year and 6 month time step cases notably under predict the probability value calculated at the same time using the 1 day time step. For the mean fraction of the area cracked, it seems that most of the points on the scatter plot are below the reference line, indicating that for the longer time steps, the crack growth tends to be larger than that finer time step case. This makes sense since the growth rate is calculated once per time step and allowed to grow bigger before coalescence occurs since these two modules are run in series. The time history plot shows generally the same behavior over time. For the total leakage rates, there is quite a bit of divergence between the finest time step and rest of the cases. The trend on the scatter plot seems to be towards higher leak rates for the finest time step as the majority of points plot on or above the reference line, however, there are a notable number of points below the reference line. Even the 0.5 month time steps show the same behavior. The scatter in the data may reflect the interaction between rupture and leakage, when a pipe ruptures the leak rate is set to zero and an earlier rupture in the smaller time step cases may explain the higher predicted rates for the longer time steps. The scatter may also be the result of instability in the leakage module. However, there is a general trend towards higher values for the smallest time step size, for the lowest values the majority of points plot on or very close to the reference line. The time history plot of the mean leakage rate does show that over the duration of the simulation time, the highest peaks tend towards the smaller time steps and with mean higher leak rates for all cases at later times. The inconclusive scatter in the correlation needs to be explained, since the evaluation of leak detection is dependent on calculating an accurate leak rate. As it is not clear that the leak rates are more accurate at the finer time steps, these analyses suggest that a time step size of 2 months selected for the pilot study analyses will not significantly affect the base case model results. Refining the temporal discretization may be required if the variance in the leak rates cannot be explained or is determined to be problem specific.



Note: The 0.5 month case was plotted only for the Leak rate plots.

Figure 4-9. Time Step Sensitivity Analysis Results – Total Leak Rates.

4.3.2 Statistical Stability

This section describes the statistical stability testing which involves a number of activities related to demonstrating that a sufficient number of stochastic realizations have been run to achieve a numerically stable mean for the base case, including: demonstrating numerical accuracy of the mean results by comparing the results of analyses using more realizations and different random seeds. The stability of the mean and other quantiles will be considered for both aleatory and epistemic uncertainty.

Sampling techniques are numerical procedures and, as such, are subject to error. The error is dependent on the method and on the sample size used. Several types of sampling techniques have been considered in order to estimate mean probability of rupture. A Monte Carlo sampling method was used as the reference technique for the xLPR study; LHS is used for sampling over epistemic uncertainty while SRS is used for sampling over aleatory uncertainty, as explained in Section 3.1.1. While the reference method is an appropriate starting point to perform a sensitivity analysis and to understand key parameters driving the uncertainty, this method has limitations when trying to accurately estimate extremely low probabilities driven by extreme quantiles. Therefore, a DPD technique using importance sampling on selected parameters, as well as a modified LHS including importance sampling on the same selected parameters has been considered.

Results of interest can vary by orders of magnitude depending on uncertainty or variability in the input, scenario or sensitivity case considered. Therefore it is not the intention of this section to define a sample size that is appropriate for any of the potential calculations or inputs. Rather, this section presents a methodology and criteria that can be used to determine the stability of an analysis. The methodology has been applied to evaluate the stability of several selected pilot study scenarios. In Version 2.0 of the xLPR model, sufficient attention needs to be given to consider whether the results are stable enough to provide confidence in the conclusion.

As a note, it is important that the following tests be conducted with Version 1.0 of xLPR framework and values should not be compared to the ones in later versions.

4.3.2.1 Selection of Initial Sample Size

For the reference technique, a minimum LHS size of 100 over epistemic uncertainty was considered to be necessary, to present 5th and 95th percentiles as well as having dense coverage over the epistemically uncertain parameters. For the aleatory uncertainty, a sample size of 20 was selected, considering that once epistemic data was fixed, generating 20 futures was likely enough to stabilize the expected value over aleatory uncertainty as a first step. The following subsections will test the appropriateness of such a sample size using base case results.

Comparison of different aleatory sample size (epistemic sample set to 100) for reference sampling technique

The following figures display results with a fixed epistemic sample size of 100, while aleatory uncertainty sample size varies from 20 to 50 and finally 100. This first set of results compare the expected (over aleatory) total leak rate over time as shown in Figure 4-10. Expected results display similar behavior for individual curves and statistics. The main differences are seen on individual curves, as increasing the aleatory sample allows a better estimate of each expected value, as more samples are used to define the expected value curve. The effect on the overall mean is negligible, but some difference can be seen on the 95th percentile, most notably at early times.

The comparison of expected fractional area cracked shows a very good agreement in the results, whatever the aleatory sample size; as shown in Figure 4-11. One can notice, on this graph, the two outlier realizations leading to a fractional surface area cracked greater than 100%. The probability of rupture also seems stable whatever aleatory sample size is used, as shown in Figure 4-12.

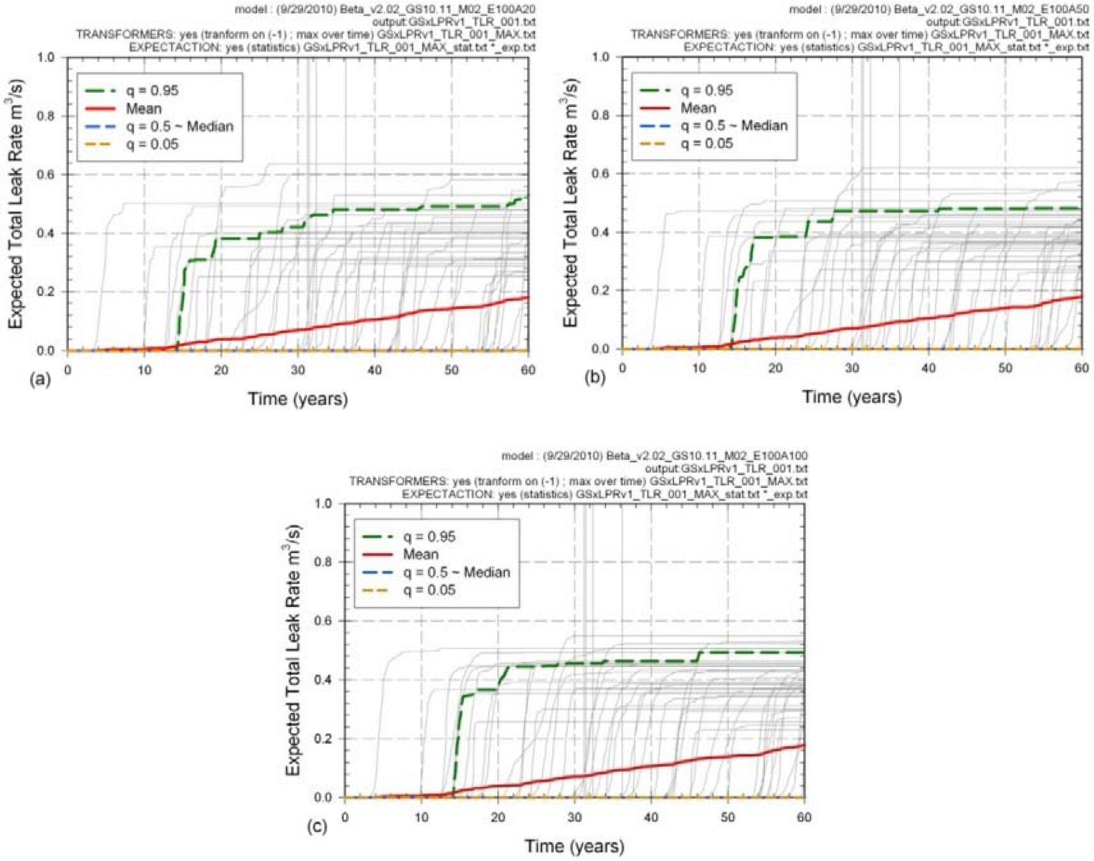


Figure 4-10. Comparison of expected results for total leak rate for aleatory sample size of 20 (a), 50 (b) and 100 (c).

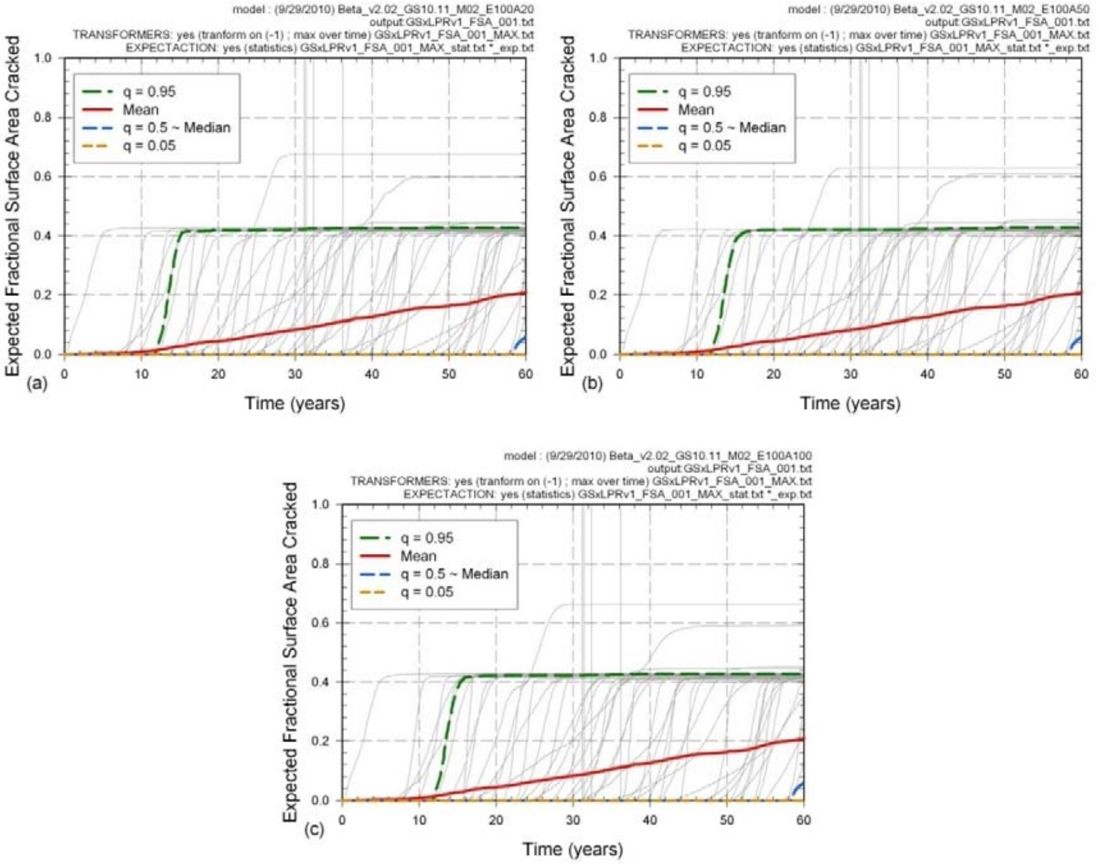


Figure 4-11. Comparison of expected results for fractional surface area cracked for aleatory sample size of 20 (a), 50 (b) and 100 (c).

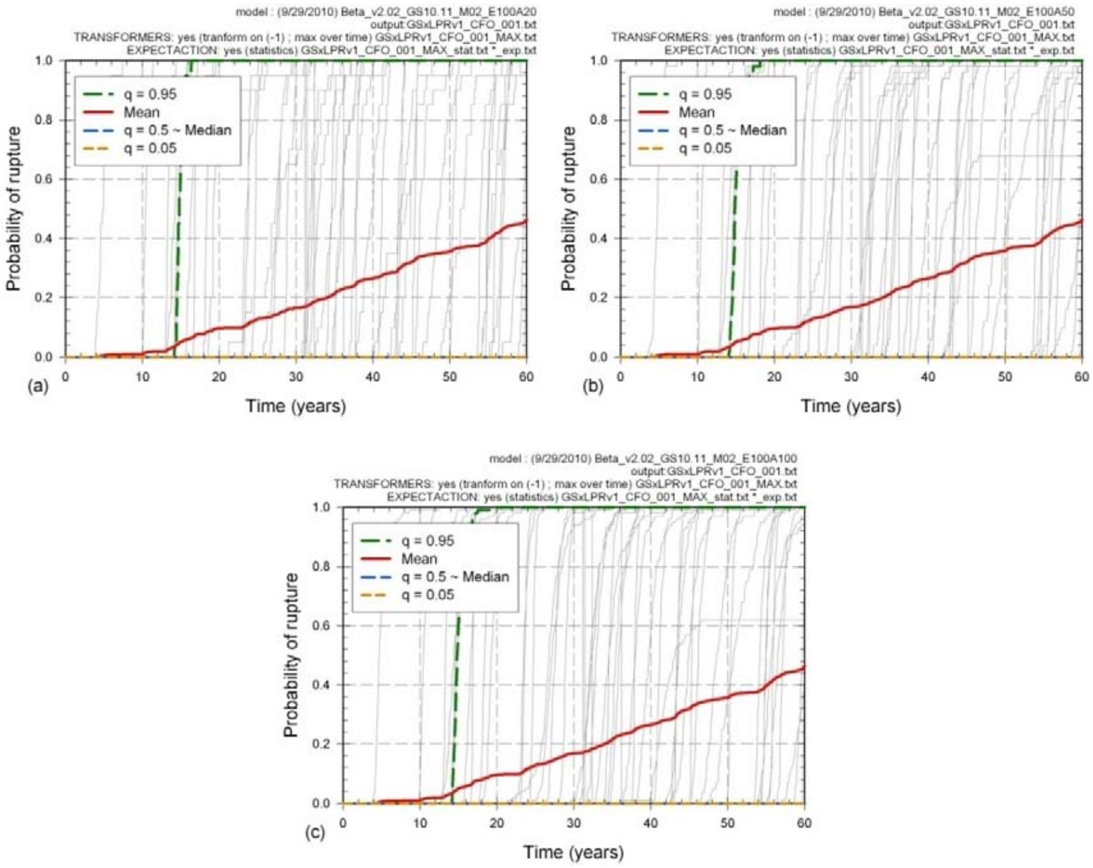


Figure 4-12. Comparison of expected results for the probability of rupture for aleatory sample size of 20 (a), 50 (b) and 100 (c).

A main objective of the xLPR pilot study model is to estimate extremely low probability of rupture, as shown in Figure 4-12. However, when the probabilities of rupture are estimated with a leak rate detection of 10 gpm (~ 6.30×10^{-4} m^3/s), only a handful of epistemic realizations (4 or less out of 100) lead to rupture when leak detection is implemented; underlining the differences due to sample size on the mean probability. Figure 4-13 shows the results when leak detection is applied to the three aleatory sample sizes. For all aleatory sample sizes, the results seem to be unstable and can be misleading. Although the results of sample size of 50 are close to sample size of 100, the results seem to show an over-estimation at early times decreasing with aleatory sample size, but a different random seed may have resulted in an underestimation. However, the instability in the mean result is mainly due to the small epistemic sample size (100). As shown in Figure 4-13, only to two to four epistemic realizations (grey lines) have at least one rupture (in the aleatory sample set) with leak detection at 10 gpm and are used in order to estimate the mean probability of rupture. Instability between each of the aleatory sample sizes is reflected in each of the epistemic realizations (grey lines), which represent an average over the aleatory (expected value). As a result, the estimate of mean probability is unstable at early times due to the aleatory sample size as noted previously, but is almost identical after 40 years.

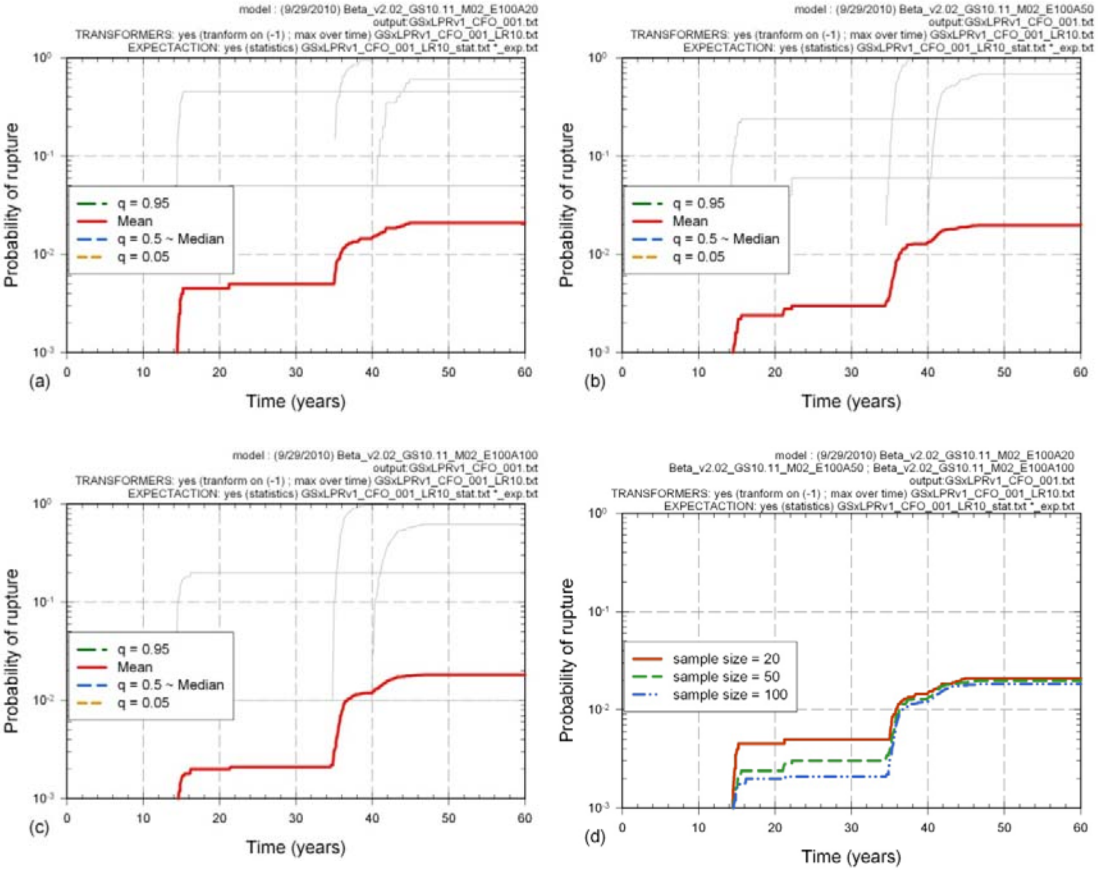


Figure 4-13. Comparison of expected results for the probability of rupture with a leak rate detection of 10 gpm for aleatory sample size of 20 (a), 50 (b) and 100 (c) and comparison of mean for all three cases (d).

A last comparison has been performed on the CCDF at 50 years for the fractional surface area cracked, which is displayed in Figure 4-14. Each epistemic realization is displayed in Figure 4-14 as a grey line. The spread in the distribution of grey lines reflects the uncertainty in the fractional surface area cracked driven by the uncertainty in the epistemic parameters. As noted previously, only 2-4% of the epistemic realizations result in a pipe rupture when leak detection is considered as shown in Figure 4-13. In addition, two realizations can be seen in all three cases with varying aleatory sample sizes that have a fractional surface area cracked greater than 50% at 50 years. Figure 4-14 also shows the accuracy of each CCDF over the aleatory sample sizes of 20, 50 or 100. The near vertical behavior of CCDFs (grey curves), with no outliers within each CCDF (i.e., all values are high or all low) underlines the fact that small aleatory sample sizes are enough to capture most of the aleatory uncertainty in the fractional surface area cracked.

When the mean CCDF is considered (red curve in Figure 4-14), only about 41% of the realizations have a crack occur as reflected by the maximum probability on the y-axis of 0.41. In addition, the majority of CCDFs are grouped together (grey lines) with only a few outliers, indicating that a larger sample size may be needed to correctly represent such outliers.

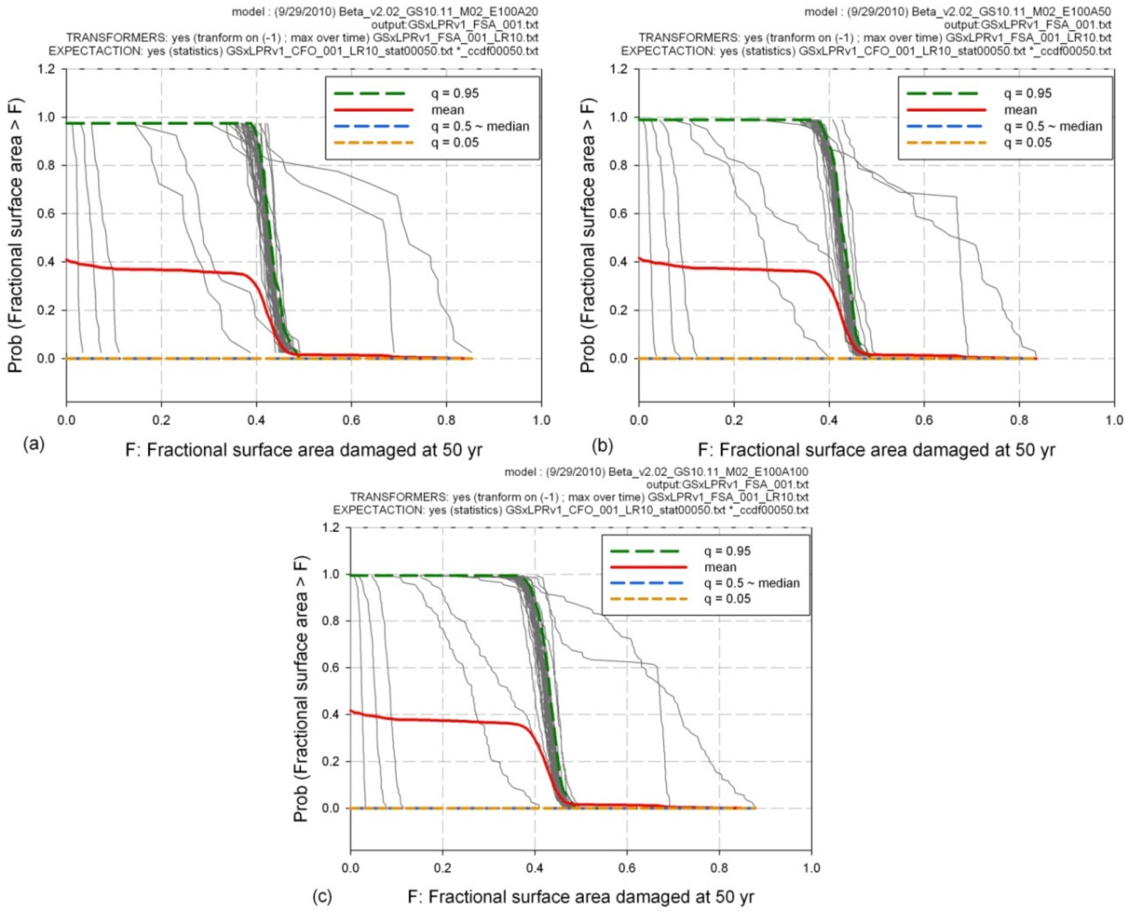


Figure 4-14. Comparison of expected results for the fractional surface area cracked with a leak rate detection of 10 gpm for aleatory sample size of 20 (a), 50 (b) and 100 (c).

In conclusion, a sample size of 50 for aleatory uncertainty seems reasonable and a good compromise between total sample size and accuracy of representation of the aleatory uncertainty in the results for the base case. It is likely that an aleatory sample size of 20 would have been sufficient for most of the output. Considering the results evaluated up to this point in the pilot study it is not conclusive that increasing the aleatory sample size above 50 would significantly increase the model accuracy given the computational expense required. As demonstrated in the following sections, when the reference methodology is used, extremely low probability occurrences are better estimated by enhancing the epistemic sampling. However, this conclusion is conditional on the current base case results. If the re-classification of some uncertainty from epistemic to aleatory occurs, if another problem was considered, or if a different output set was analyzed, this conclusion could change.

Comparison of different epistemic sample size (aleatory sample size set to 50)

Previous analyses have shown that the epistemic sample size of 100 is not adequate to estimate probability of rupture for the base case when leak rate detection is considered. It seems that about 2 out of 100 epistemic sets generated have a non-zero probability of rupture after leak rate detection is included. Using a smaller epistemic sample size would not be appropriate. Several higher sample sizes (200, 500, 1000 and 2000) have been tested over the same aleatory sample size in order to evaluate the stability of the model with respect to the epistemic sample size.

A comparison of the mean probability of rupture is shown in Figure 4-15. Figure 4-15(a) shows very few variations in the mean probability of rupture without inspection or leak detection. Results (except for early time) are almost identical. As about 40% of the realizations lead to failure if no inspection or detection is applied, this result is not surprising: a sample size of 100 is large enough to capture this kind of behavior, especially when LHS technique is used.

Once post-processing corrections are applied in order to account for inspection, the number of realization still leading to pipe failure dramatically decrease (Figure 4-15(b)). Results from smaller sample sizes (100 and 200) give results in the magnitude expected but relatively unstable over time, as seen by the large increases in the probability at discrete points in time. The sample size of 500 gives smoother results; it seems to underestimate the final probability when comparing to the largest LHS sample sizes. While convergence at both early and late times correlates with increasing the epistemic sample sizes, the 1000 LHS sample size yields results close to results of a sample size of 2000.

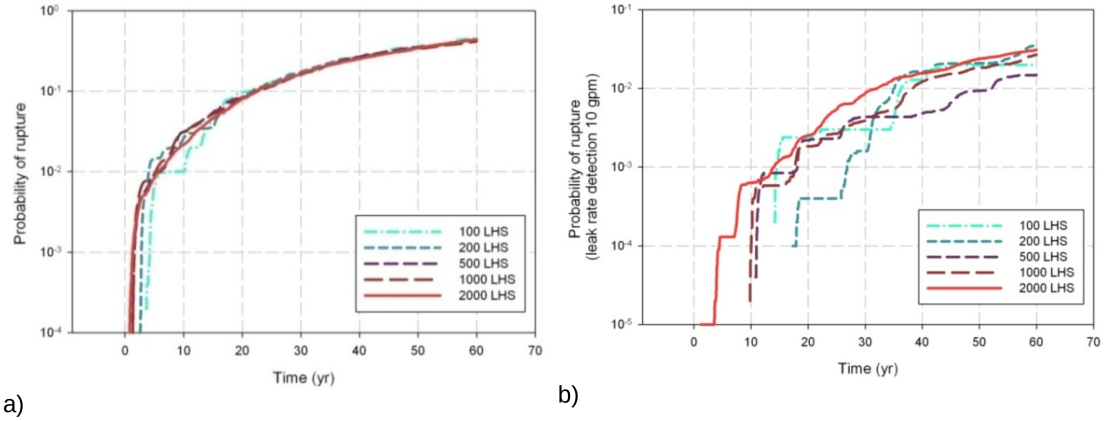


Figure 4-15. Comparison of expected results for (a) the probability of rupture for various epistemic sample sizes and (b) the probability of rupture for various epistemic sample sizes with a leak rate detection of 10 gpm.

In order to quantify the differences between different sample sizes, a hypothesis test has been performed on the mean probabilities at 60 yr. While we know that the inputs are sampled from the same distributions and therefore the distribution of results should be the same, it is still interesting to see whether such a test would lead to a rejection of the hypothesis of similarity of the mean result (indicating that the sample size is clearly not enough to obtain stable results).

The Welsh's t-test over a simple comparison of means was selected to test the similarity of the mean values. Since unequal epistemic sample sizes were considered the estimation of common standard deviation is subject to variation (especially for low sample size and for the leak rate detection case). For Welsh's t-test, the t-statistic is estimated using the formula: $t = \frac{\bar{X}_1 - \bar{X}_2}{s_{\bar{X}1-\bar{X}2}}$ with $s_{\bar{X}1-\bar{X}2} = \sqrt{\frac{s_1^2}{n1} + \frac{s_2^2}{n2}}$ is an unbiased estimator of the variance of the two samples, $\bar{X}_i$ the mean for sample i, s_i^2 its variance and ni its size. The p value is then estimated from the Student's t-distribution with the degrees of freedom estimated using:

$$ddf = \frac{\left(\frac{s_1^2}{n1} + \frac{s_2^2}{n2}\right)^2}{\left(\frac{s_1^2}{n1}\right)/(n1-1) + \left(\frac{s_2^2}{n2}\right)/(n2-1)}$$

In this test, a small p-value indicates that the similarity of means can be rejected, while a large value will indicate that the hypothesis cannot be rejected. Table 4-1 summarizes the p-values when the largest sample (2000) is compared with the smaller ones for the mean probability estimated at time T=60 years. Traditionally, a reference p-value of 0.05 is used to reject or accept a hypothesis (considering that having 5% of being wrong when rejecting a hypothesis is a reasonable risk). However, if the consequences of rejecting a hypothesis are serious, a smaller value of p (e.g., 0.01) can be considered. For the pilot study, the reference p-value of 0.05 seems reasonable.

In all cases but one, the p-value is large enough so that the hypothesis is not rejected. The only rejection occurs for the 500 LHS case when leak detection is applied, which had a mean value significantly lower than the other test cases (Figure 4-15(b)). One important feature of this hypothesis test is that the mean value has to be closer to the reference when the sample size increases, which explains why the comparison between the 1000 and 2000 sample sizes is not as good as the smaller sample sizes for the mean result without leak detection (but it is high enough). However, with leak rate detection of 10 gpm, at time T = 60 years the comparison between 1000 and 2000 LHS has a higher p value, p = 0.51.

Table 4-1. Estimated p-value over the hypothesis of similar mean values using Welsh t-test.

Modeling Case	100 vs. 2000	200 vs. 2000	500 vs. 2000	1000 vs. 2000
base case	0.66	0.88	0.95	0.19
base case (with leak rate detection 10 gpm)	0.40	0.69	0.008	0.51

Again the selection of the sample size must consider the increase in the model accuracy given the computational expense required. The use of 2000 LHS is a little prohibitive in terms of computational resources when coupled with the SRS size of 50 for the aleatory uncertainty; the total sample size would be 100,000. Since it is not conclusive that increasing the epistemic sample size above 1000 would significantly increase the model accuracy given the computational expense required, for the pilot study analyses a LHS size of 1000 for epistemic uncertainty was used, for a total sample size of 50,000. Analyses that require larger sample sizes indicate that if the reference methodology is used (Monte Carlo with LHS and SRS), extremely low probability occurrences would be better estimated by importance sampling. However this sample size should be enough to determine the effect of the uncertainty of each input parameter on the probability of rupture, and is integral in the development of an appropriate importance sampling analysis.

4.3.2.2 Stability of the Referenced Sampling Methodology using Replicates

One important feature of LHS is its dense stratification allowing a good coverage over the entire distribution of each uncertain input variable. One of the drawbacks of this stratification is the inter-dependence of the runs which invalidates any sub-sample taken from the total. In consequence, the use of bootstrap method in order to estimate the distribution of mean value (representing the error in estimating the mean value when using such technique) is not appropriate.

One way to estimate confidence bounds on the mean (or on any other statistics) is to perform replicates of the analysis using different random seeds. Each replicate gives an estimate of the statistics of interest, allowing construction of a distribution of this statistic. As the law of great numbers indicates, the distribution of the mean value should be normal. Therefore, one can use t-distribution in order to estimate its standard error and calculate quantiles over the mean value.

Figure 4-16 displays a comparison of mean value of probability of rupture with leak rate detection at 10 gpm for three replicates of 1000 LHS (and SRS size of 50 for aleatory sample size of 50) in linear (a) and log (b) scale (for xLPR Version 1.02).

Results have some differences but are still fairly close, within the same order of magnitude. As only about 3% of the runs leads to rupture despite leak detection, this represents 30 to 40 realizations out of 1000. Having some differences between the cases with 30 to 40 realizations with rupture is expected and not surprising.

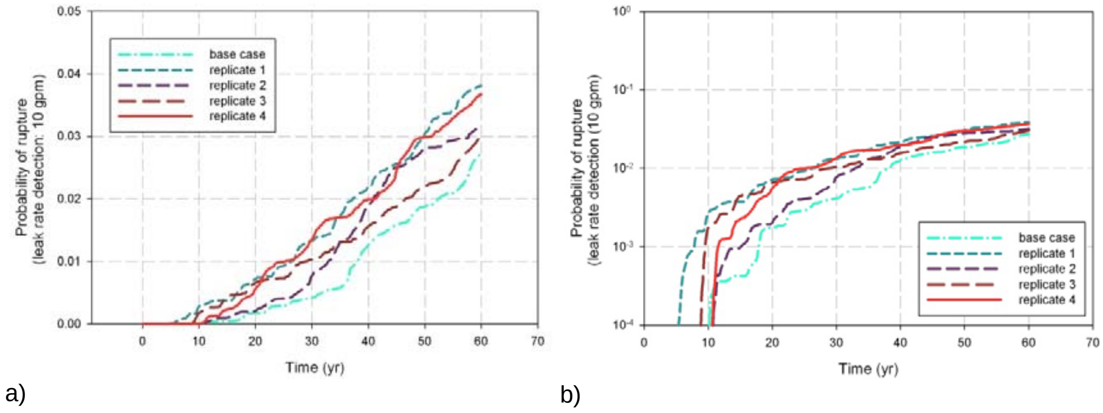


Figure 4-16. Comparison of expected results for the probability of rupture with a leak rate detection of 10 gpm for various epistemic sample sizes (a) linear y-axis and (b) log scale for y-axis.

The five replicates can be used to estimate an overall mean value (mean of five means) and confidence bounds around this mean, using the t-distribution. Figure 4-17 displays the results of the *q* = 0.95 confidence interval (estimated using q=0.025 and q=0.975).

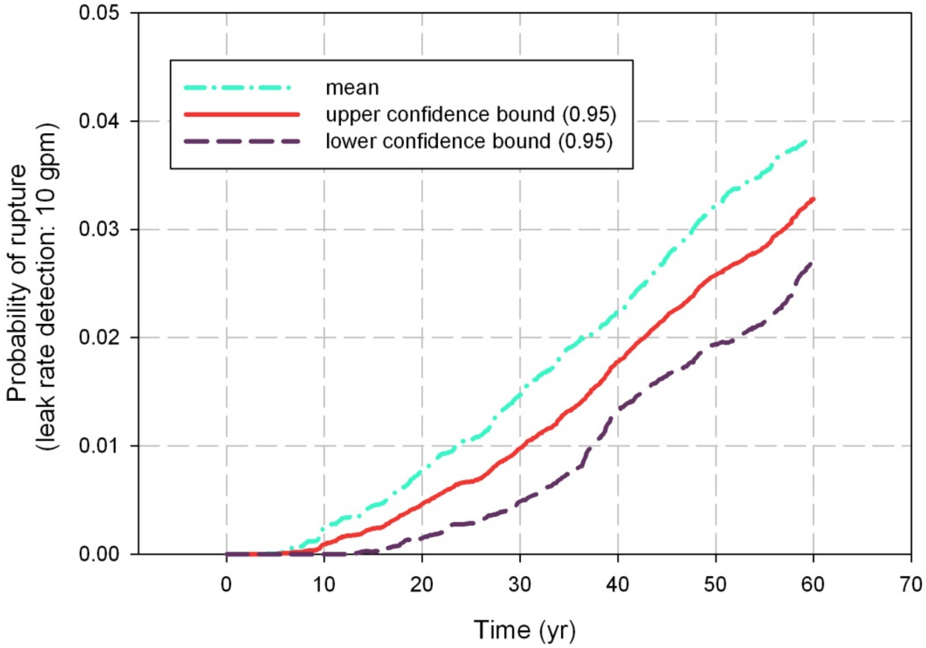


Figure 4-17. 0.95 confidence interval over mean probability of rupture with leak rate detection set to 10 gpm using 5 replicates.

Looking at such plots can help determine whether stability is achieved, but having some quantitative measure could be more desirable. Any quantitative measure will have to be problem specific as well as its interpretation. For instance, a factor of two increase in a value may have stronger consequences if the estimate is around 10^{-2} than if it is around 10^{-5}. However, there are some quantitative criteria that can be use in order to test for stability of replicated values.

The first consideration when using replicates is not to compare extreme values between them, as the more replicates you generate, the more likely you will have extreme values far from each other. It is thus necessary to average the differences between the replicates in order to evaluate the stability of the sampling in consideration.

Another method could be to look at the difference as ratio from a theoretical value, considering that if these ratios are small enough, then the results are stable. One way is to calculate the average percentage of deviation from the overall mean. First, the absolute difference between each replicated mean and the average of all means is estimated and normalized by the average of all means. It gives a ratio of deviation for each replicate. These ratios can be averaged to give a first estimate. This method is a variation of the calculation of the coefficient of variation (standard deviation divided by mean) using a different norm (L^1 instead of L^2). One limitation of this method is that the ratios can be pretty high at early years, when only a few replicates have a result. It is thus necessary to interpret them in light of the actual value in order to estimate whether the result is stable or not (see Figure 4-18).

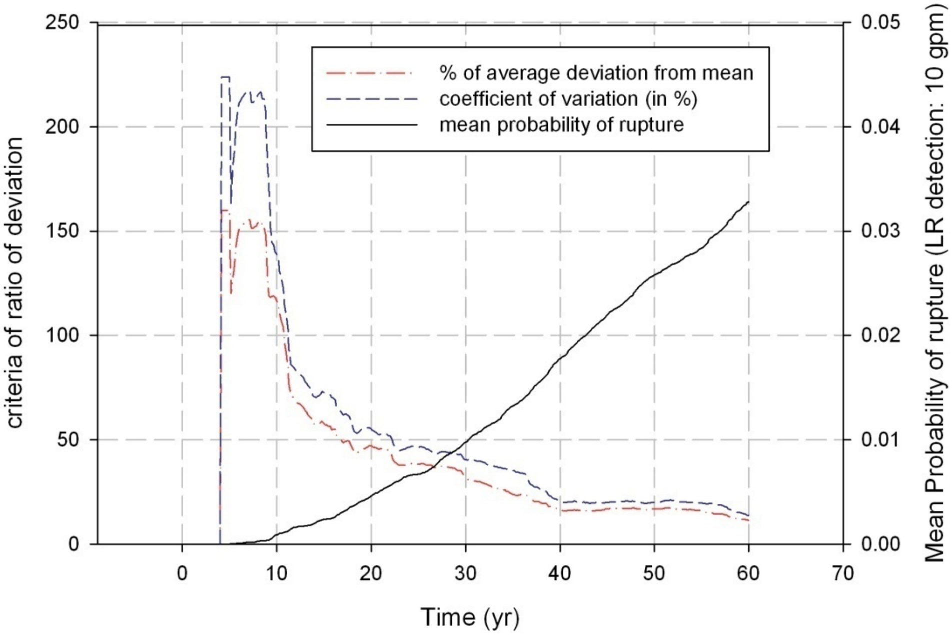


Figure 4-18. Estimate of stability using ratios, compared with mean probability of rupture for leak rate detection set to 10 gpm using 5 replicates.

Another way is to look directly at the standard deviation or the mean of absolute difference (once again variation of the same metric using different norm). Such a method, once again, should be considered in light of the original value as a variation of 10^{-4} will not have the same impact if the mean value is equal to 10^{-2} or 10^{-4}.

Finally, a user may not be interested in the variation in probability at a particular time (which corresponds to a vertical average) but more in the variation in a year for a particular probability estimate (which corresponds to a horizontal average). All the methods described above can be valid by inverting X-values (time) with Y-values (probabilities) in this case (see example in Figure 4-19).

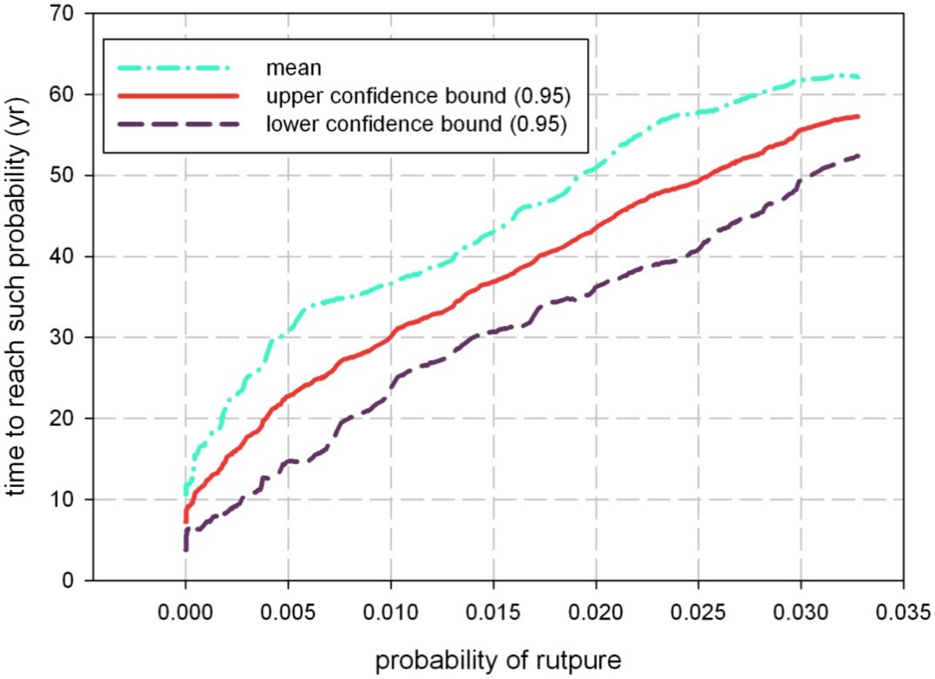


Figure 4-19. Confidence bound over average time to reach a defined probability of rupture (leak rate detection set to 10 gpm) using 5 replicates.

Such an analysis shows that the average standard deviation is about 5.5 years, with a maximum standard deviation of 7 years and a minimum of 2 years. For the purpose of demonstration we will consider that an average variation of 5 years is acceptable for estimating such probability.

4.3.2.3 Stability of DPD Importance Sampling via Bootstrap

When DPD is used, each realization is the result of a random combination of discrete values for each input parameter, following equal or unequal discretization (respectively for non importance and importance sampling). A global weight is estimated for each realization, representing the size of the region covered proportionally to the input hyperspace. As each realization is a random combination with replacement, the use of bootstrap is appropriate in order to estimate uncertainty due to sample size, since each subset of the original sample will be a valid sample by itself.

The bootstrap technique used here is slightly different than the classical one in order to take into account of the weight associated with each realization: 10,000 realizations are randomly selected over the pool of 10,000 values with replacement, keeping the weight associated to each value. The mean can then be estimated using weighted sum. The operation is repeated 500 times in order to create a distribution of mean values. Figure 4-20 presents results of bootstrapped sample distribution of the mean for probability of rupture in the safe end case. The central limit theorem states that, as the mean and variance of the initial distribution are finite, the mean distribution should be asymptotically normal. A PDF of the distribution shows a bell shaped curve that approximates a normal distribution. In this situation, the sample size is likely large enough to reach stability.

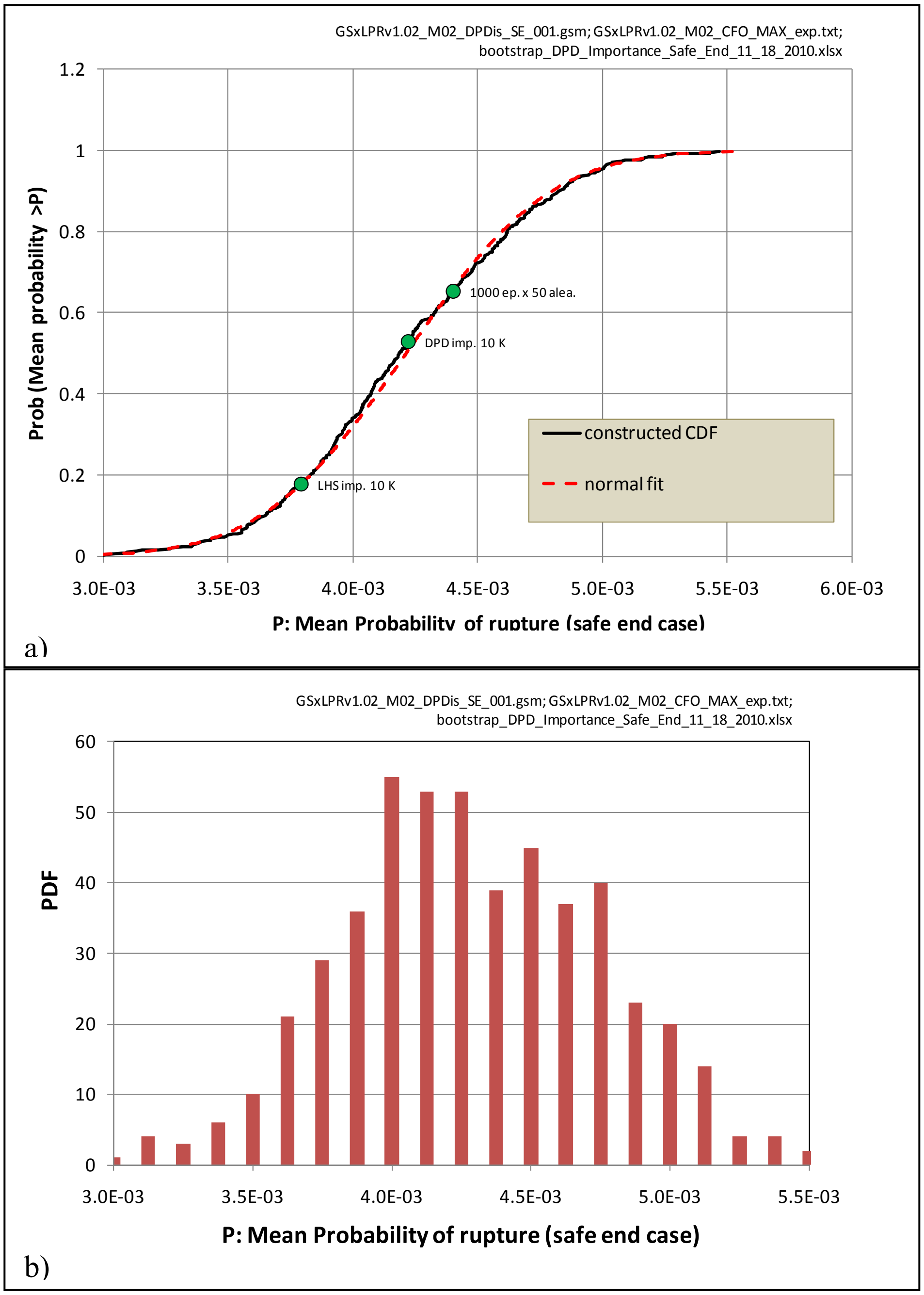


Figure 4-20. CDF (a) and PDF (b) of distribution of mean values using 500 bootstrap estimates based on 10,000 realizations, from an initial importance sample size of 10,000.

Figure 4.21 presents a CDF and PDF on the probability of rupture for the safe end case when leak rate detection (set to 1 gpm) and inspection (set to every 10 years) are considered. Now the distribution of the mean no longer follows a normal pattern: the original mean value is on quantile *q* ~ 0.6, while the PDF displays a noticeable skew to the right. In this case, we can assume that stability is not obtained with the selected sample size and importance sampling applied. Results indicate that the true mean probability may be underestimated based on the skewness. In this situation, it is recommended to either increase the sample size, or redefine the importance sampling applied in order to have a better coverage of the region of interest.

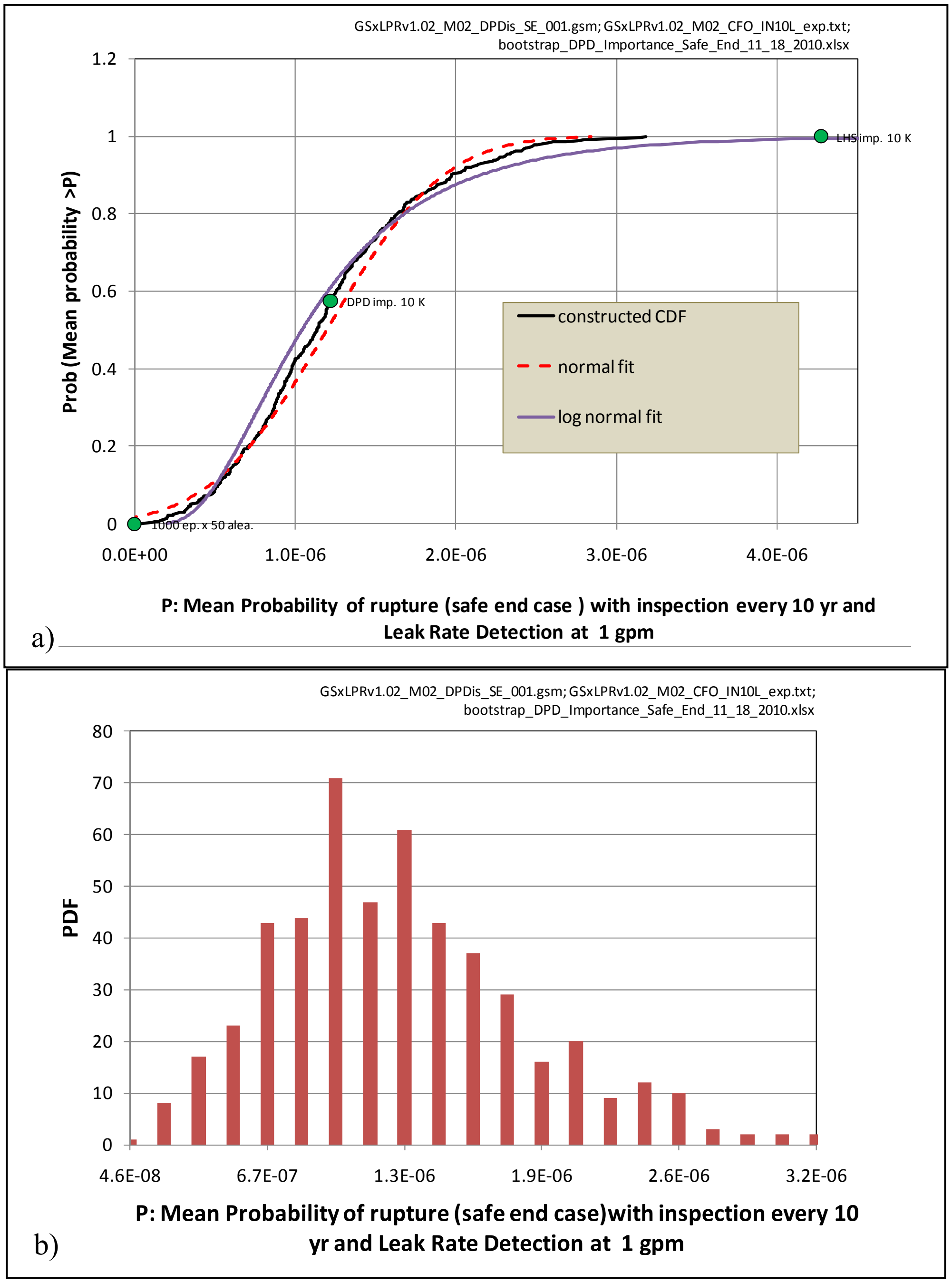


Figure 4-21. CDF (a) and PDF (b) of distribution of mean values using 500 bootstrap estimates based on 10,000 realizations, from an initial importance sample size of 10,000.

4.3.2.4 Stability of LHS importance sampling

The application of importance sampling over LHS values can be very powerful. However, as for the regular LHS, it is not appropriate to use bootstrap estimates in order to construct the distribution of the mean. The resampling with replacement will break the LHS structure that may lead to “holes” in the sampled input distribution. Therefore a more traditional approach would be to use replicates and estimate confidence bounds over the distribution of the mean using t-distribution. However, bootstrap can still be applied in a qualitative way in order to see how much the distribution deviates from normal. The deviation from a normal distribution in the CDF and PDF presented in Figure 4-22 is not as pronounced than for the DPD importance sampling case, but it does not mean that the probability estimate is better here.

Finally an indicator that can be used to determine whether the importance sampling is appropriate can be the count of realizations leading to rupture. An efficient importance sampling should have more realizations leading to rupture than a less efficient one. Of course this metric has to be considered cautiously. To correctly estimate low probability of rupture it is necessary to apply the importance sample over the output space leading to rupture with the greatest probability. If this area is not correctly covered, the probability of rupture will be underestimated. Table 4-2 lists the number of realizations leading to rupture in the case of safe end and safe end considering leak rate detection set to 1 gpm and inspection every 10 years for regular LHS, DPD importance sampling and LHS importance sampling.

Table 4-2. Number of realizations leading to rupture in safe end case with and without leak rate detection and inspection for three sampling techniques.

	1000 LHS epistemic x 50 SRS aleatory	DPD 10K importance sampling	LHS 10K importance sampling
Safe End	6 out of 1000 (0.6%)	1080 out of 10K (11%)	5145 out of 10K (~51%)
Safe End LR: 1 gpm INS: 10 yr	0 out of 1000 (0%)	152 out 10K (~1.5%)	227 out of 10K (~2.3%)

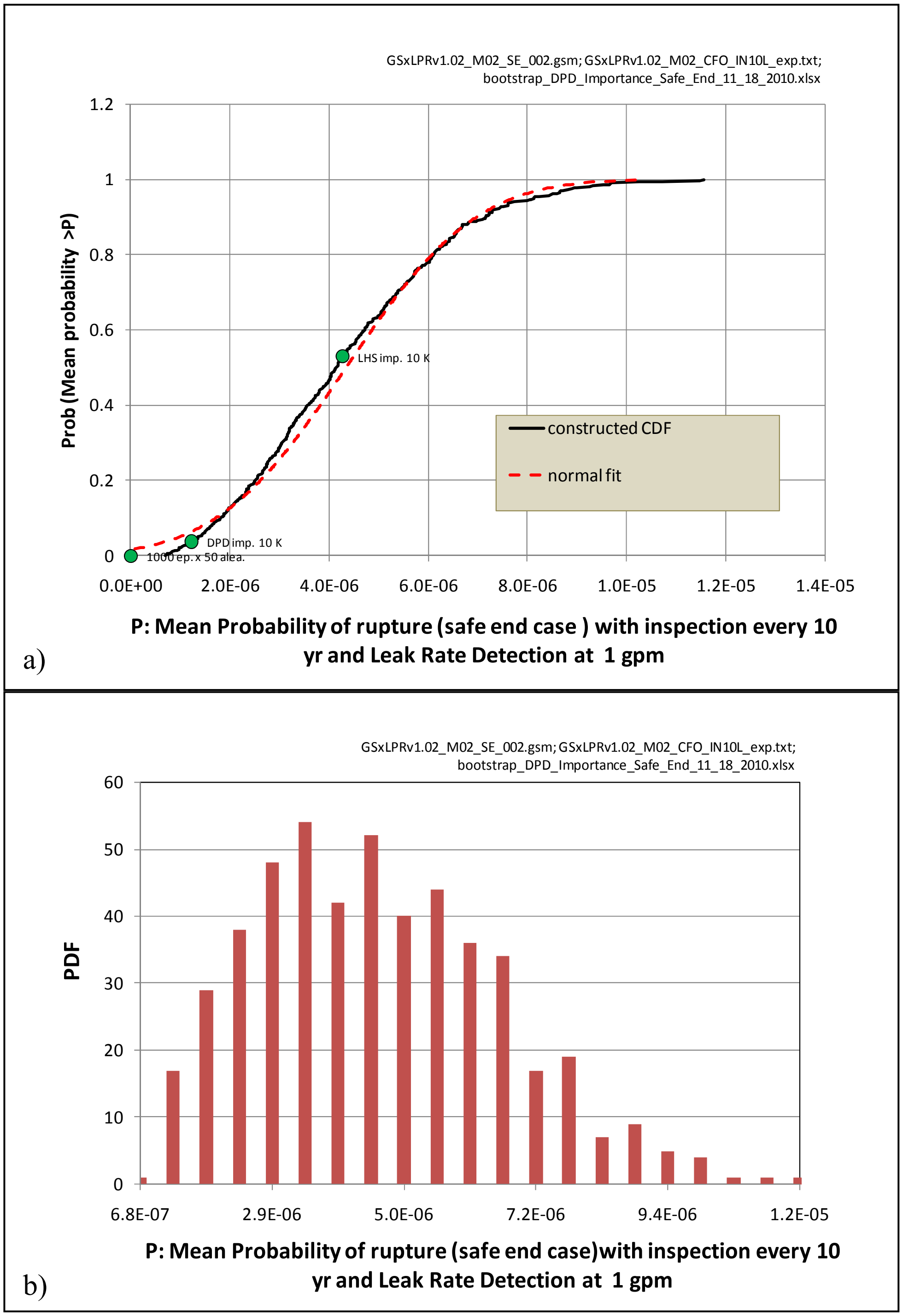


Figure 4-22. Qualitative estimate of distribution with bootstrap on probability of rupture for safe end case with leak rate detection of 1 gpm and inspection every 10 yr using LHS importance sample.

5. Pilot Study Analyses & Results

As stated in Section 1.0 and summarized in Section 2.0, the intended purpose of the xLPR Pilot Study is to develop a prototype xLPR model and conduct pilot study analyses. The xLPR pilot analyses were conducted to demonstrate the capabilities of the proposed developmental process and framework for a probabilistic code to represent degradation mechanisms in piping system safety assessments. The pilot study analyses illustrate the specific issue of assessing the probability of rupture of pressurizer surge nozzle DM welds degraded by PWSCC, for which a considerable amount of publicly available information already exists (see Section 2.0 and Appendix B).

The output of the pilot study is a probabilistic assessment of surge nozzle DM welds that include:

Probability of leakage at various crack opening sizes
Probability of rupture

These results include a comparison of results with and without the effects of inspection and pre-emptive PWSCC mitigation. The pilot study provides order-of-magnitude estimates of piping rupture probabilities; the analysis is used to identify areas requiring more focused attention in the long-term study.

In addition a set of sensitivity analyses were conducted to demonstrate pilot study xLPR model capabilities. Sensitivity analyses are used to evaluate or quantify the impacts of some of the modeling assumptions by evaluation of alternative model scenarios and various alternative model processes not included in the base case analysis. Section 5.1 summarizes the analysis presented in this section. Section 5.2 presents the results of the base case, including a discussion of the uncertainty in the results, as well as a study of the sensitivity of the model to the uncertain parameter inputs. Section 5.3 details the sensitivity analyses conducted for the pilot study and presents the results of these analyses. The analyses detailed in this section collectively present a demonstration of the probabilistic model capabilities that have been incorporated in the pilot study model for xLPR using the commercial software framework.

The CM process described in Section 1.3 was utilized for the xLPR Model runs and Sensitivity Analyses conducted for the pilot study test case as documented in this section.

5.1 Pilot Study Analyses

One base case and five sensitivity cases were analyzed for the pilot study report as described in detail in Appendix E. The results of these analyses are presented in Sections 5.2 and 5.3. The results discussed in this section were calculated using the controlled version of the GSxLPR framework model developed for the xLPR pilot study. The ten individual cases that comprise the analyses are listed in Table 5-1.

All of the cases were probabilistic analyses. For the probabilistic analysis using the Monte Carlo sampling method, the analysis consists of two loops (as described in Section 3.1.1). The outer loop, capturing the epistemic uncertainty, corresponds to a sample size of $nE = 1000$. The inner loop, capturing the aleatory uncertainty, correspond to a sample size of *nA= 50*. A total sample size of *nS* = (*nE* * *nA*) = *50,000* was used as determined in Section 4.3.2. The DPD analysis used a sample size of *80 BINs* and *10,000* outer loop samples.

The modifications to the controlled framework and input spreadsheet followed the CM process described in Section 1.3, are documented in change checklists, and are available on the xLPR CM SharePoint system.

Table 5-1. xLPR Version 1.0 Analyses.

Analysis	Description	GSxLPR Model Name
Base Case Analysis – Section 5.2		
Probabilistic Base Case	Probabilistic base case analysis using Monte Carlo sampling (with LHS).	GSxLPRv1.02_M02
Sensitivity Analyses – Section 5.3		
Stress Mitigation	Mitigation analyses will be run as part of the xLPR model evaluation. Analyses evaluate different mitigation times, as well as the mitigation effectiveness over the representative distributions.	GSxLPRv1.02_M02_Mit10
		GSxLPRv1.02_M02_Mit20
		GSxLPRv1.02_M02_Mit30
Chemical Mitigation	Chemical effects of increasing the hydrogen concentration in the water on the crack growth module. Three hydrogen concentrations were evaluated 25, 50, and 80 cc/kg.	GSxLPRv1.02_M02
		GSxLPRv1.02_M02_H50
		GSxLPRv1.02_M02_H80
Crack Initiation	Considers the crack initiation model uncertainty.	GSxLPRv1.02_M02_CIMethod_I_001
Safe End Evaluation	Considers stainless steel safe end weld which causes a through thickness bending stress that can reduce the tensile inner diameter (ID).	GSxLPRv1.02_M02_SafeEnd
Importance Sampling	DPD analysis with importance sampling using the Safe End Evaluation analysis	GSxLPRv1.02_M02_DPDis_SE_001
	Base Case analysis with alternative importance sampling methodology	GSxLPRv1.02_M02_002
	Safe End evaluation analysis with alternative importance sampling methodology	GSxLPRv1.02_M02_SE_002
Uncertainty Classification	Evaluation of effects of moving crack initiation uncertainty from Epistemic to Aleatory	GSxLPRv1.02_M02_003

5.2 Probabilistic Base Case Analysis

The base case analysis is described in Section 2.0. The input data and modules provided by the xLPR Models Task Group and commercial software framework, integrated together as described in Section 3.0, comprise the base case xLPR Model. The base case analysis appropriately treats the identified epistemic and aleatory uncertainties to produce expected value output results with quantified uncertainties. The probabilistic base case analysis was used as the benchmark to which all other sensitivity analyses were compared.

The base case model has been verified and documented for use in support of the xLPR program evaluation of a commercial software framework (Section 4.0), and is the result of a short term learning experience that should benefit the longer term program and code development by identifying areas requiring more focused effort.

Sections 5.2.1 and 5.2.2 present the uncertain results and analysis of the responses to the parameter uncertainty, respectively.

5.2.1 Uncertainty Results

5.2.1.1 Probability of Occurrence of Cracks

The mean probability of having at least one crack goes almost linearly from 0 to 0.48 over time (Figure 5-1**)**. In the base case, aleatory uncertainty has no effect in the time of crack initiation. Therefore the probability of first crack will be either 0 or 1 over time (which explains why the displayed probabilities in light gray look like vertical lines). Therefore, a mean result of 0.48 indicates that only 48% of the realization will see at least one crack over a 60 yr period. Median and 5th percentile are then equal to 0. For the same reason, all of the following results presented for the base case will not display the median and 5th percentile values. The 95th percentile goes from 0 to 1 around 120 months, which means that we have to wait about 10 years before having at least 5% of the results with at least one crack.

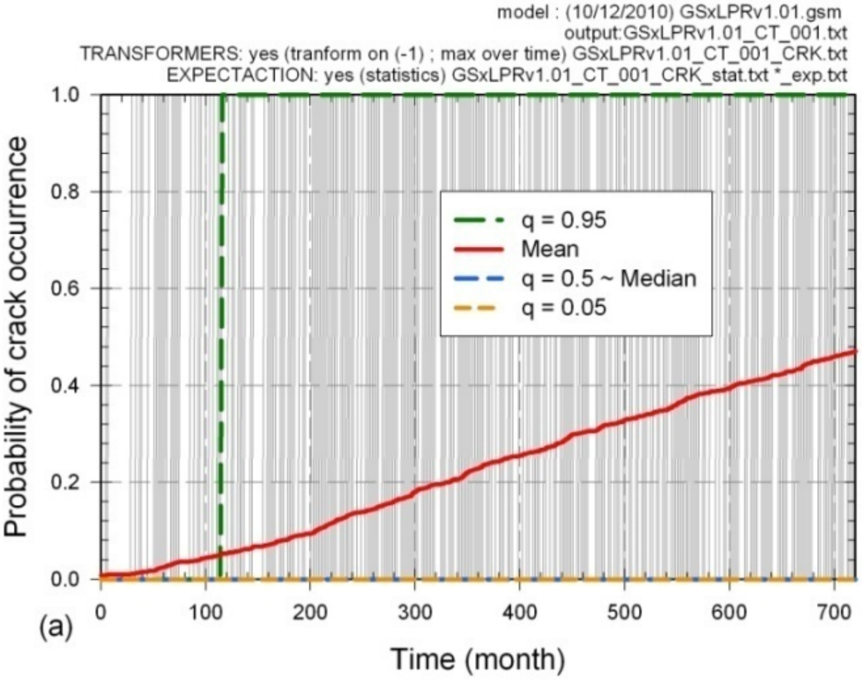


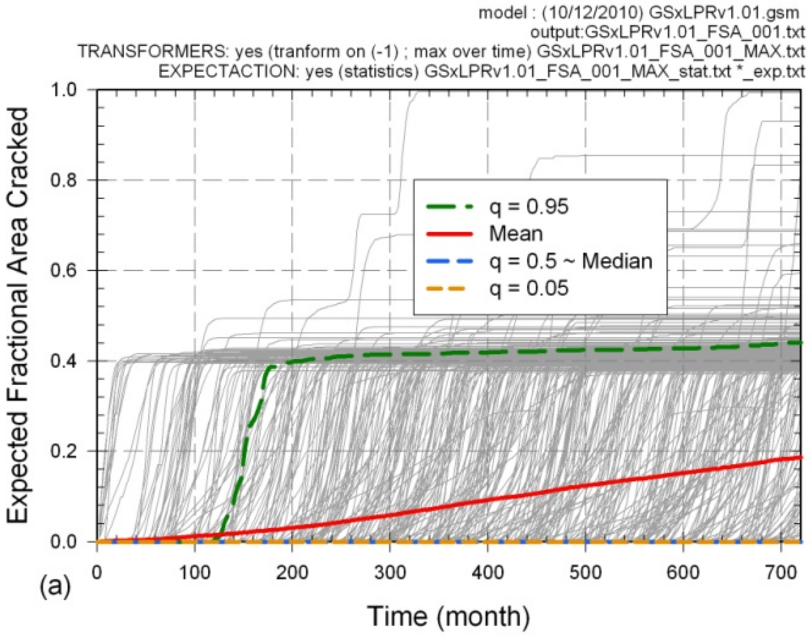
Figure 5-1. Base Case Mean and distributions for the probability of at least one crack.

5.2.1.2 Fractional Surface Area Cracked

Results for expected fractional area cracked show a definite pattern with the surface area increasing more or less quickly to a value of 0.4 which lead to crack failure (Figure 5-2). A value of 0.4 therefore seems to represent the limit beyond which a TWC will lead to pipe rupture. Only 2% of cases that lead to a higher surface area cracked will correspond to a long and thin SC that will rupture at the same time it becomes a TWC.

CCDFs (Figure 5-2 for 100 LHS epistemic) shows that this behavior (a long thin crack) is mainly controlled by the epistemic uncertainty. The individual CCDFs (grey curves) leading to high values of the expected fraction of surface area cracked (>50%) represent individual epistemic samples (each with 50 Aleatory samples) with the highest SC results. The majority of the CCDFs are between 0.40 and 0.50 values. These unlikely results (2% of the epistemic runs) above 0.5 may lead to pipe rupture even in the case of leak rate detection if the long thin SC leads to a pipe rupture immediately upon transition to a TWC. It is then the purpose of importance sampling to study more accurately the area of the input hyperspace corresponding to these results.

These results will not lead to pipe rupture once post-processing leak rate detection is used, as we will see later in this section.



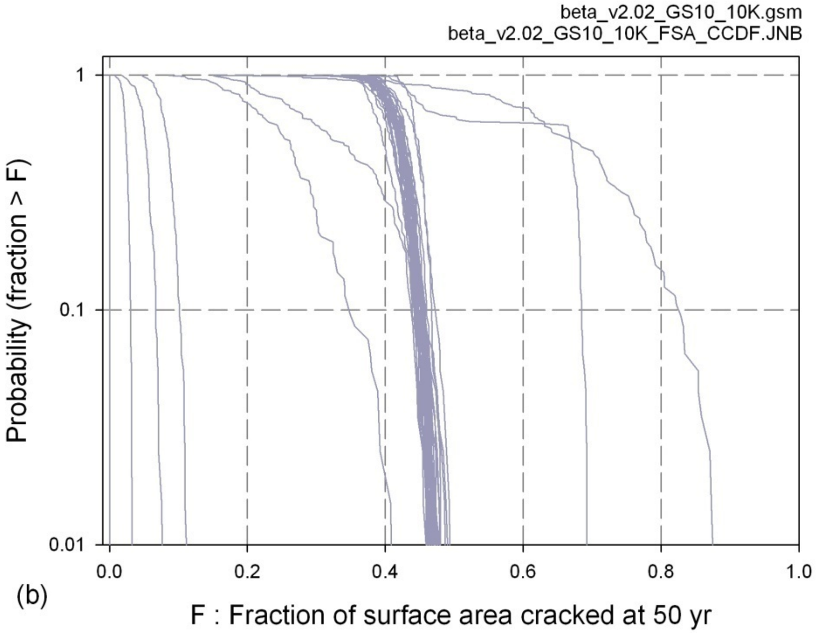


Figure 5-2. Base Case Expected Fractional Surface Area Cracked.

5.2.1.3 Probability of Through Wall Crack (TWC) Occurrence

As seen in the previous figures, crack evolution is identical for most of the realizations with a crack (representing about 40% of the runs). Therefore it is not surprising to see that the probability of first leak exhibit the same behavior as the probability of first crack, except for the fact that the time to reach it is longer (Figure 5-3).

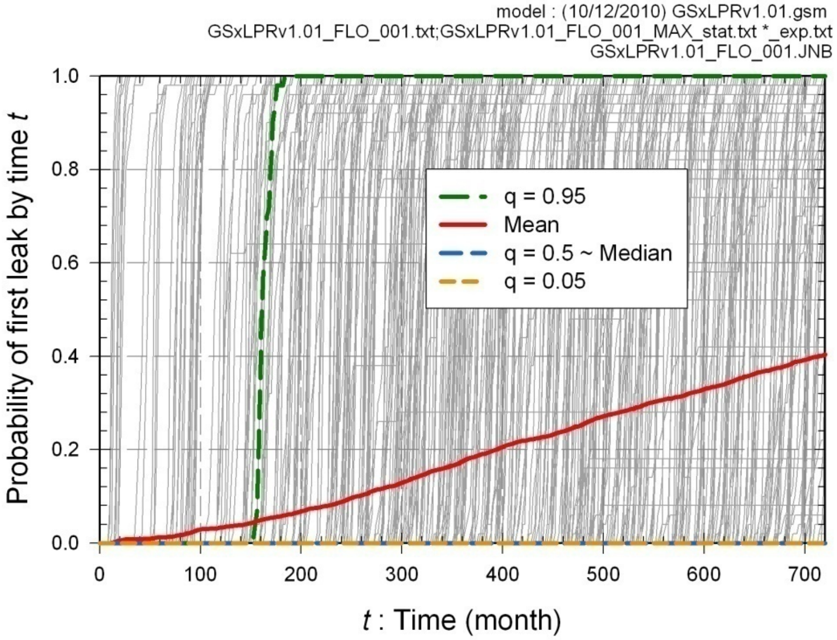


Figure 5-3. Base Case Probability of First Leak.

5.2.1.4 Expected Total Leak Rate

Similar behavior can be seen in Figure 5-4 for the expected total leak rate as seen in Figure 5-2 for expected fractional surface area crack. This makes sense as a larger crack area will lead to larger leak rate. A few runs show really large leak rate, which likely correspond to the epistemic realizations with the largest surface area cracked before rupture.

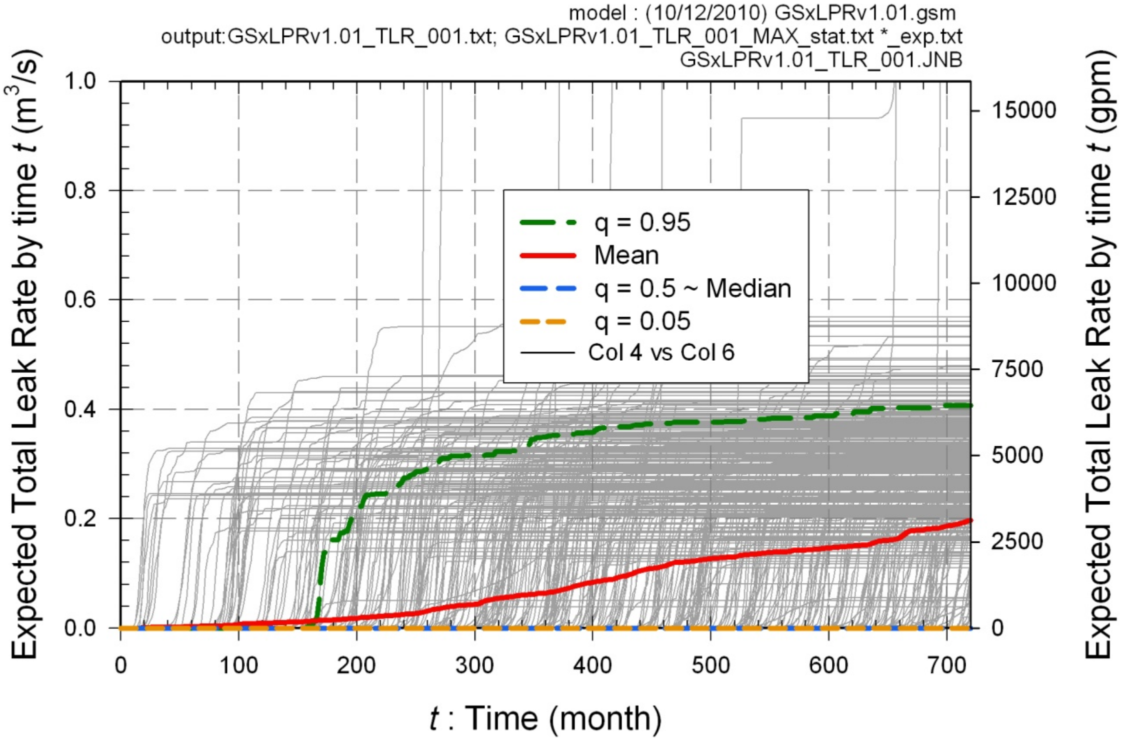


Figure 5-4. Base Case Expected Total Leak Rate.

5.2.1.5 Probability of Pipe Rupture

Figure 5-5 displays the probability of pipe rupture. The same behavior is observed amongst simulations that have a crack occur; most of them will lead to pipe rupture before the end of the simulation. Therefore the probability of rupture looks a lot like the probability of first occurrence and first leak, except for the fact that it is shifted more to the right, indicating a delay between first occurrence, first leak, and ultimately rupture.

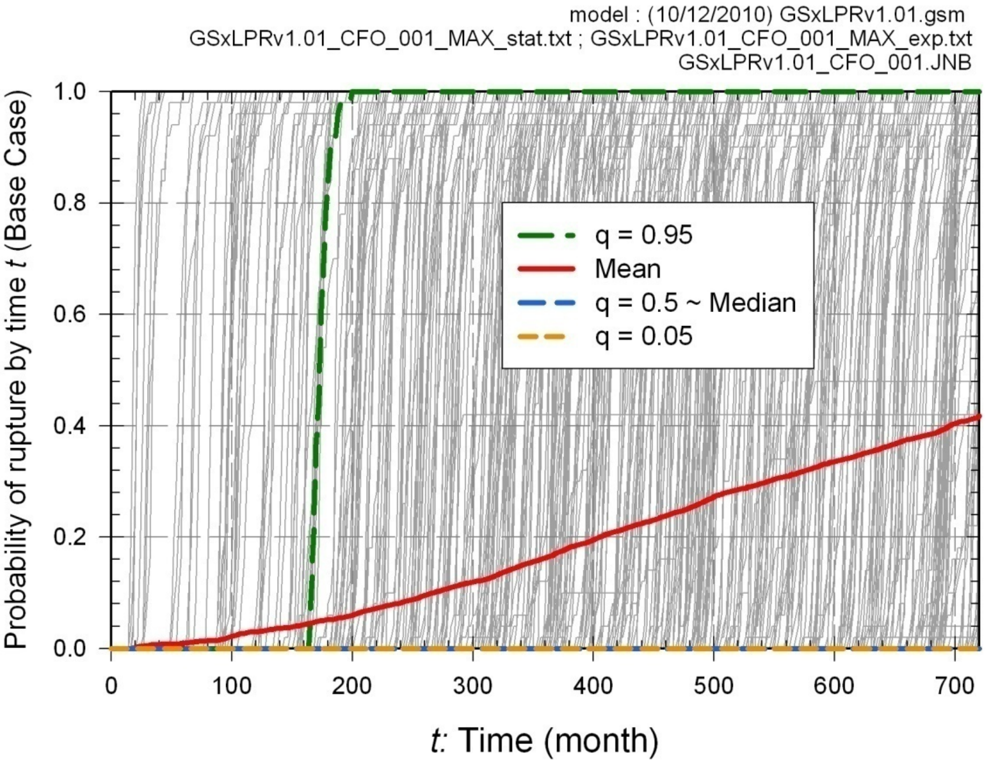
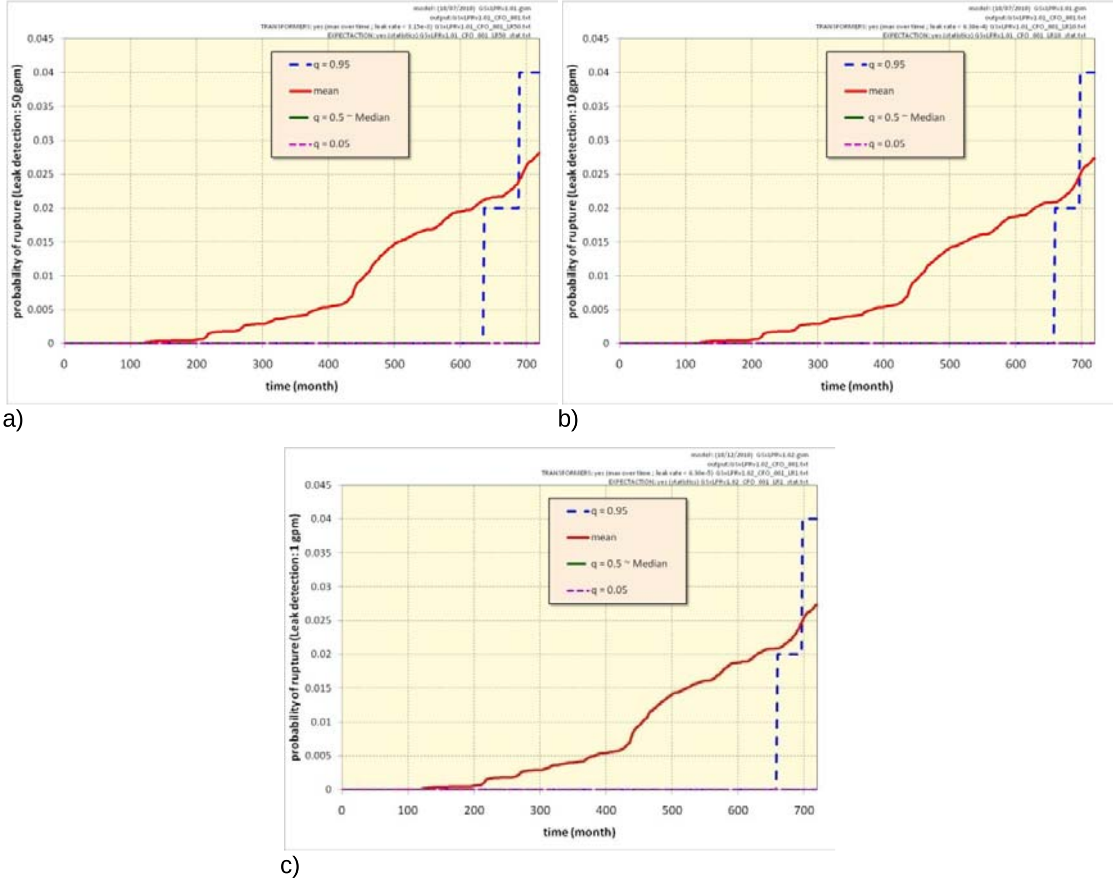


Figure 5-5. Base Case Probability of Rupture with statistics.

5.2.1.6 Probability of Pipe Rupture in Presence of Leak Rate Detection

When analyzing fraction of surface area cracked, one has observed that almost all of the fractions were bounded by a value around 0.40. This represents the history of a SC evolving into a TWC and failing once the TWC meets the critical failure criteria.

These kinds of cracks will generally be detected as they leak for several months. However there was a handful (about 28) of cracks that grew beyond this limit of 40%. These represent long and thin SCs that were able to damage most of the weld before becoming TWC and cause rupture at the same time step. Not surprisingly, these 28 cases out of 1000 epistemic samples (representing therefore 2.8%) are the only ones remaining when leak detection is applied to the rupture probability. The selection the Leak rate threshold does make much of a difference as most of the cracks will usually grow long enough to lead to a large leak rate, so the use of 1 gpm, 10 gpm or 50 gpm limits lead to identical results as shown in Figure 5-6.



a)

b)

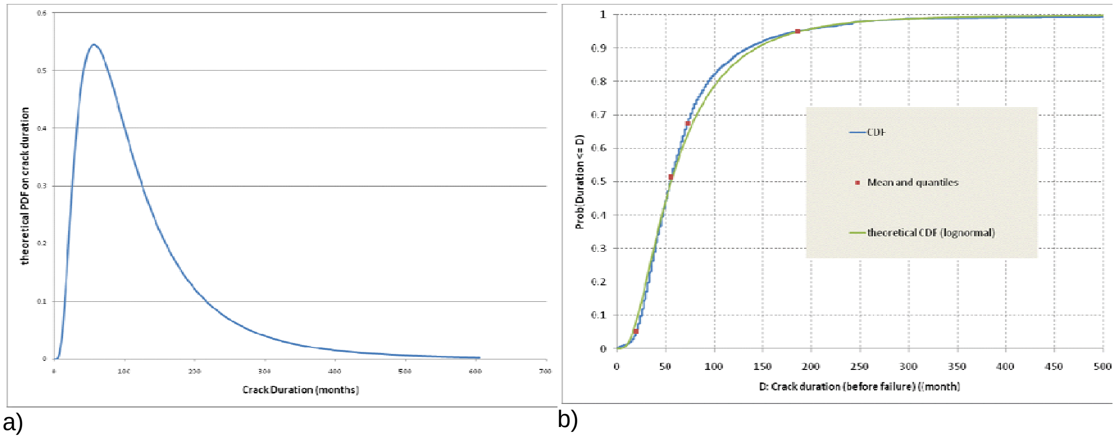
c)

Figure 5-6. Base Case Probability of rupture with leak detection.

5.2.1.7 Probability of Pipe Rupture in Presence of Inspection

When selecting inspection time, crack duration should be taken into account to assure that most of the cracks will be found before failure. Therefore, the distribution of crack duration needs to be analyzed. Figure 5-7 shows the crack duration distribution is close to a log-normal distribution. The red dots represent (from bottom to top along the curve) the 5th percentile, median, mean and 95th percentile, respectively.

Their values are respectively 20 months (about 1 year and a half), 56 months (about 4.5 years), 73 months (about 6 years) and 186 months (about 15.5years). It means that half of the time, a crack will appear and lead to a rupture in less than 5 years. On average, it will take 6 years but the average is close to the 67th percentile (which means that 2/3 of the crack will evolve faster). Moreover the 10 yr duration corresponds to the 87th percentile, meaning that almost 9 cracks out of 10 will already lead to rupture within 10 years after they have initiated.

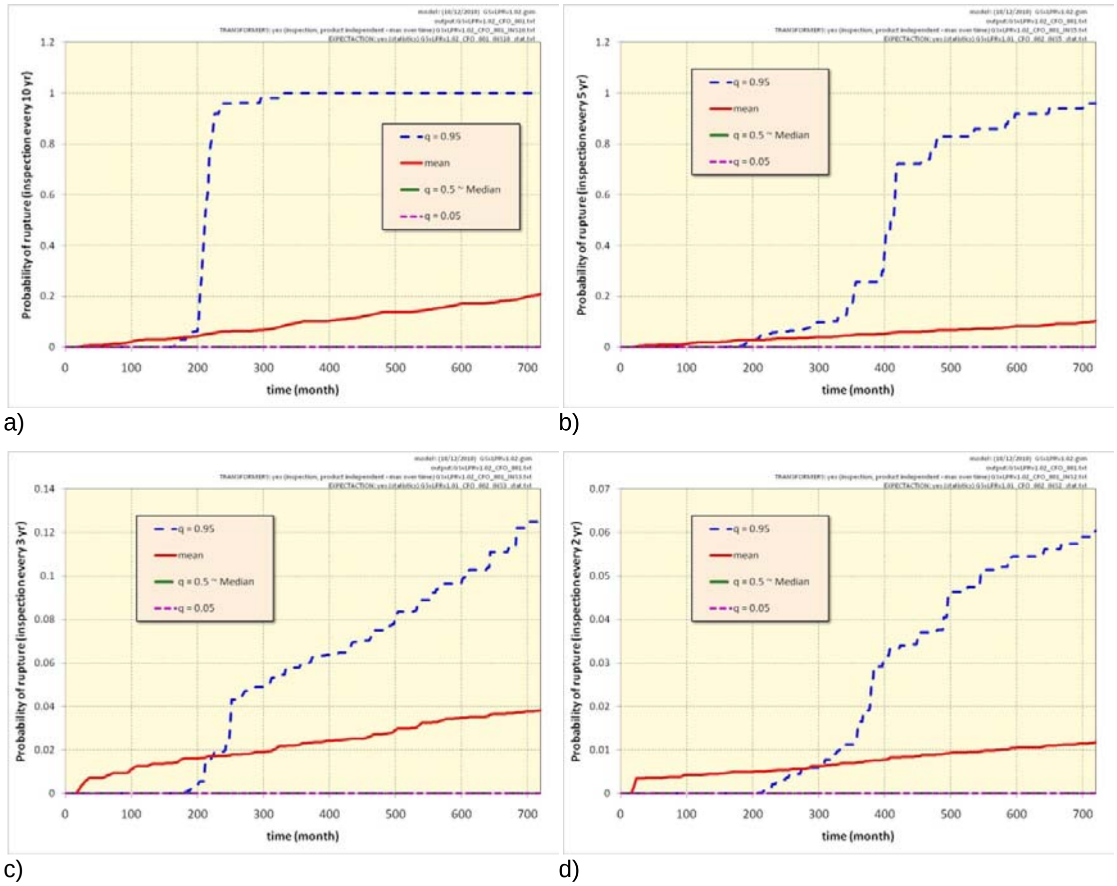


a) b)

Figure 5-7. Base Case Distribution of Crack Duration.

Applying inspection every 5 or 10 years, knowing that more than half to 90% of the crack will have time to initiate and lead to rupture in the timeframe is probably not appropriate. A more appealing quantile would be, for instance, the first quartile (meaning that only ¼ of the cracks will evolve faster than this value) or the first decile (meaning that only 10% of the cracks could initiate and lead to failure between two inspections). These convenient quantiles are close to inspections every 2 or 3 years.

The four plots in Figure 5-8 estimate the probability of rupture considering inspection every 10 yr, 5 yr, 3 yr and 2 yr. The y-axis (not the same scale for all frames in Figure 5-8) shows there is a slight reduction in the probability of rupture with an inspection every 10 or 5 yrs (not surprisingly as half of the crack may be missed or only simply inspected once). The 10 and 5 year inspection interval can be seen in the 95th parceling and only affects the mean by a factor of 2 to 4. Using a two or three year inspection interval, almost all the crack will have a good chance to be inspected at least one to two times during their duration, leading to a significant decrease in probability (at least one order of magnitude).



a)

b)

c)

d)

Figure 5-8. Base Case Probability of rupture with inspection every 10 yr (a), 5 yr (b), 3 yr (c) and 2 yr (d).

5.2.1.8 Comparison Probability of Pipe Rupture

The Figure 5-9 summarizes the effect of post-processing inspection and detection on the probability of rupture. In this analysis, due to the relatively short duration needed for a crack to lead to pipe rupture, inspection every 5 years or more is not efficient enough to reduce the probability of rupture significantly. Only inspection every 2 or 3 years will lead to a probability around 10^{-5}.

Leak rate detection plays an important role in reducing the probability of rupture; however, the threshold selection does not really influence the results. The reason is that the only possibility to have failure despite leak detection is to have long thin cracks that will cause failure in the same time step as they evolve into TWC. Since only a few of the simulations lead to such behavior (2.8% with a 1000 LHS sample size) the use of importance sampling is recommended to estimate more precisely these probabilities. However, the order of magnitude is consistent to what is expected.

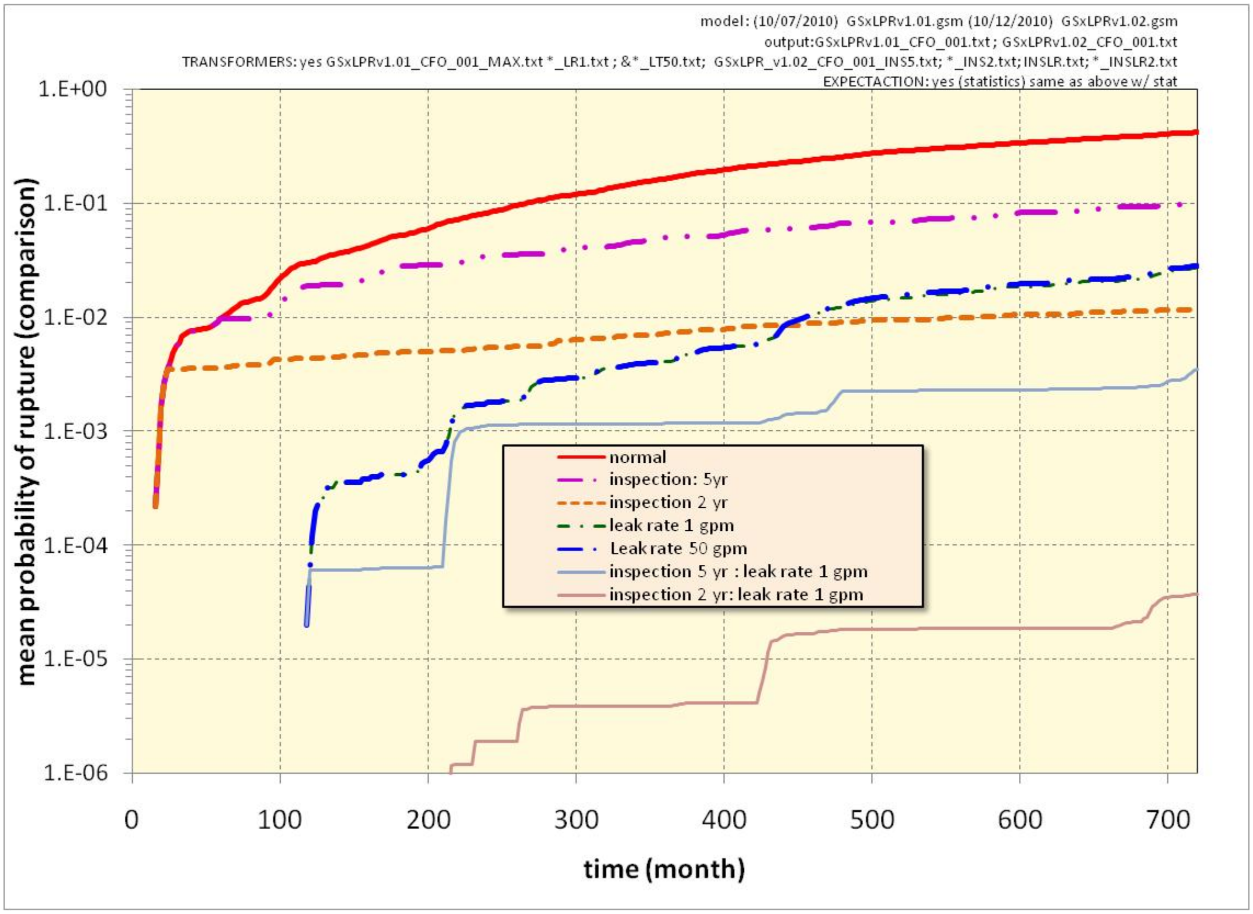


Figure 5-9. Comparison of the Base Case probability of rupture with inspection and leak detection.

5.2.2 Parameter Sensitivity Analysis

A sensitivity analysis has been performed on a total sample size of 50,000. The aleatory sample size is set to 50 and the epistemic sample size to 1000.

5.2.2.1 Probability of occurrence of crack

Table 5-2. Stepwise regression of Probability of first crack occurrence (PROBCRK) at 4 selected time-steps.

PROBCRK: 10 yr [a]				PROBCRK: 30 yr				PROBCRK: 50 yr				PROBCRK: 60 yr			
var. [b]	R^2 [c]	R^2 inc. [D]	SRRC [e]	var.	R^2	R^2 inc.	SRRC	var.	R^2	R^2 inc.	SRRC	var.	R^2	R^2 inc.	SRRC
SIG0WRS	0.14	0.14	0.14	SIG0WRS	0.35	0.35	0.43	SIG0WRS	0.42	0.42	0.55	SIG0WRS	0.44	0.44	0.57
B1	0.15	0.02	-0.05	B1	0.47	0.12	-0.26	B1	0.59	0.17	-0.35	B1	0.62	0.18	-0.37
				RANDP16	0.47	0.00	0.05	RANDP06	0.60	0.00	-0.05	RANDP18	0.62	0.00	0.04
				RANDL16	0.48	0.00	0.05	ODRAND	0.60	0.00	-0.04				
				RANDP08	0.48	0.00	-0.03	RANDP08	0.60	0.00	-0.04				
								RANDP05	0.60	0.00	0.03				

a: output variable and time-step in consideration
b: Variables listed in order of selection in stepwise regression
c: Cumulative R^2 value with entry of each variable into regression model
d: Incremental R^2 value with entry of each variable into regression model
c: Standard Rank Regression Coefficients (SRRCs) in final regression model

The major contributors to the variance of crack occurrence are Sigma0_wrs and B1. B1 (Heat-to-heat variability – see table D 1.2 for description) is controlling the likelihood of crack occurrence while sigma0_wrs represents the axial stress component for weld residual stress on the inner diameter of the pipe. The SRRC values indicate that high values of Sigma0_wrs and low values of B1 will lead to higher probability of first crack occurrence. The other parameters explain only a fraction of the variance (less than 1% each) and are probably spurious. The results at 10 yrs give a low R^2 which is expected considering that only 5% of realization have a crack at this time.

The 3D-contours in Figure 5-10 (which replace scatter plots) show a definite trend confirming the SRRC signs. Low values of B1 associated with high values of sigma0_wrs will lead to higher probability. One important feature is that only association of B1 and sigma0_wrs lead to higher probability of rupture, indicating that a conjoint sample could be appropriate here in term of optimization using importance sampling.

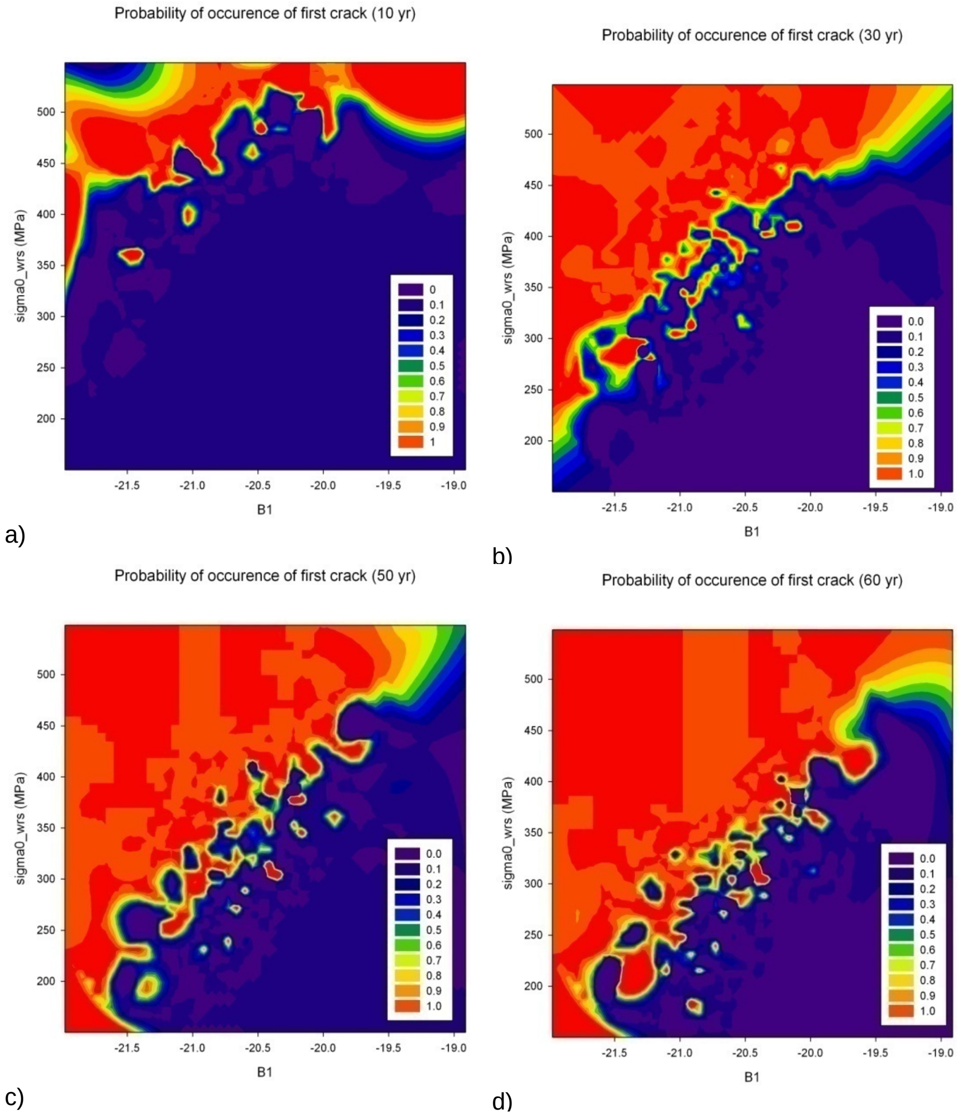


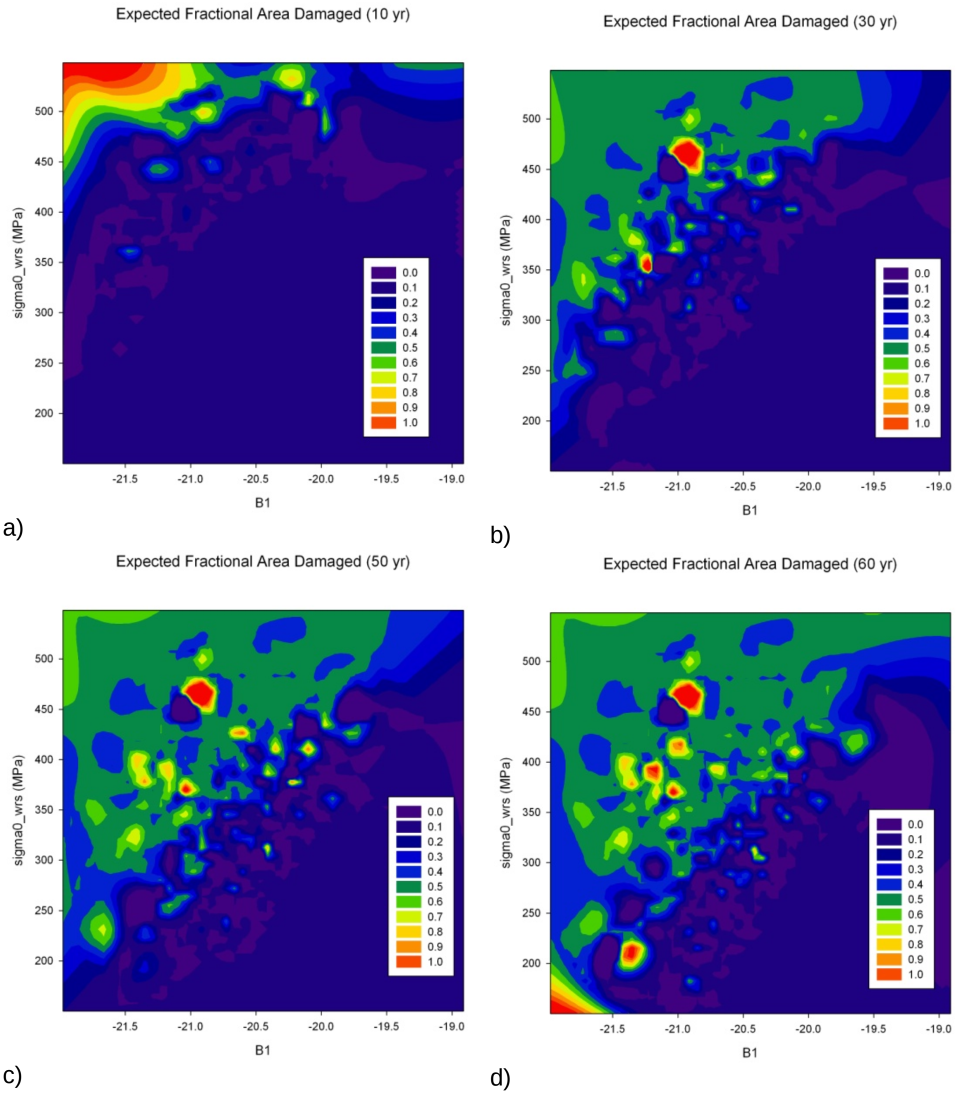
Figure 5-10. Contour plots for the Base Case probability of occurrence of the first crack at: a) 10 years, b) 30 years; c) 50 years; and d) 60 years.

5.2.2.2 Fractional Surface Area Cracked.

Table 5-3. Stepwise regression of Expected (over aleatory) fractional surface area damaged (EXPFSA) at 4 selected time-steps.

EXPFSA: 10 yr				EXPFSA: 30 yr				EXPFSA: 50 yr				EXPFSA: 60 yr			
var.	R^2	R^2 inc.	SRRC	var.	R^2	R^2 inc.	SRRC	var.	R^2	R^2 inc.	SRRC	var.	R^2	R^2 inc.	SRRC
SIG0WRS	0.14	0.14	0.14	SIG0WRS	0.36	0.36	0.44	SIG0WRS	0.44	0.44	0.58	SIG0WRS	0.45	0.45	0.62
B1	0.15	0.02	-0.05	B1	0.48	0.12	-0.26	B1	0.62	0.18	-0.37	B1	0.64	0.19	-0.40
				RANDP16	0.48	0.00	0.04	RANDP05	0.62	0.00	0.04	RANDP18	0.64	0.00	0.05
				RANDL16	0.49	0.00	0.04	RANDP06	0.62	0.00	-0.04	RANDP04	0.65	0.00	0.04
								ODRAND	0.62	0.00	-0.04	FWELD	0.65	0.00	-0.04
												RANDP05	0.65	0.00	0.04

As with the previous analysis, Sigma0_wrs and B1 are the most important parameters for fractional surface area cracked, the other parameters listed are likely due to random noise. Their influence is about the same as for the probability of first crack occurring.



a)

b)

c)

d)

Figure 5-11. Contour plots for the Base Case probability of fractional area cracked at: a) 10 years, b) 30 years; c) 50 years; and d) 60 years.

The 3D contours in Figure 5-11 present some new interesting features. If, at 10 yr, it is clear that the upper left corner is the most important area (very low values of B1, very high values of sigma0_wrs). Another small area seems to be of importance around B1 = -21 and Sigma0_wrs = 500 MPa. This feature that seems to be consistent over time, although slight in the 10 year plot, and could be part of the smoothing algorithm used by the graphing software or due to randomness in the sample generation; but this may also indicate an area of interest that needs more study.

5.2.2.3 Probability of Through Wall Crack (TWC) occurrence

Table 5-4. Stepwise regression of Probability of Through Wall Crack (PROBTWC) at 4 selected time-steps.

PROBTWC: 10 yr				PROBTWC: 30 yr				PROBTWC: 50 yr				PROBTWC: 60 yr			
var.	R^2	R^2 inc.	SRRC	var.	R^2	R^2 inc.	SRRC	var.	R^2	R^2 inc.	SRRC	var.	R^2	R^2 inc.	SRRC
SIG0WRS	0.09	0.09	0.10	SIG0WRS	0.31	0.31	0.37	SIG0WRS	0.39	0.39	0.52	SIG0WRS	0.40	0.40	0.55
B1	0.10	0.01	-0.03	B1	0.39	0.09	-0.20	B1	0.53	0.14	-0.31	B1	0.55	0.15	-0.34
RANDP07	0.11	0.01	-0.02	ODRAND	0.40	0.00	-0.04	FWELD	0.54	0.01	0.06	FWELD	0.55	0.01	0.07
TEMP	0.11	0.01	-0.02					RANDP06	0.54	0.00	-0.05	RANDP05	0.56	0.00	0.04
								RANDP01	0.54	0.00	0.04	RANDP03	0.56	0.00	-0.04
								RANDP05	0.55	0.00	0.04	RANDP06	0.56	0.00	-0.04
								ODRAND	0.55	0.00	-0.04	XC	0.56	0.00	0.04
								PRESS	0.55	0.00	0.04				
								XC	0.55	0.00	0.04				

The behavior here is identical to the previous analysis with about the same order of importance for both sigma0_wrs and B1.

Results displayed in Figure 5-12 are very close to the ones presented for fractional surface area cracked. One would usually expect the fractional surface area to be higher when the crack as evolved into TWC. The fact that the region observed previously is no longer present may indicate that it could be a region that needs to be studied using importance sampling.

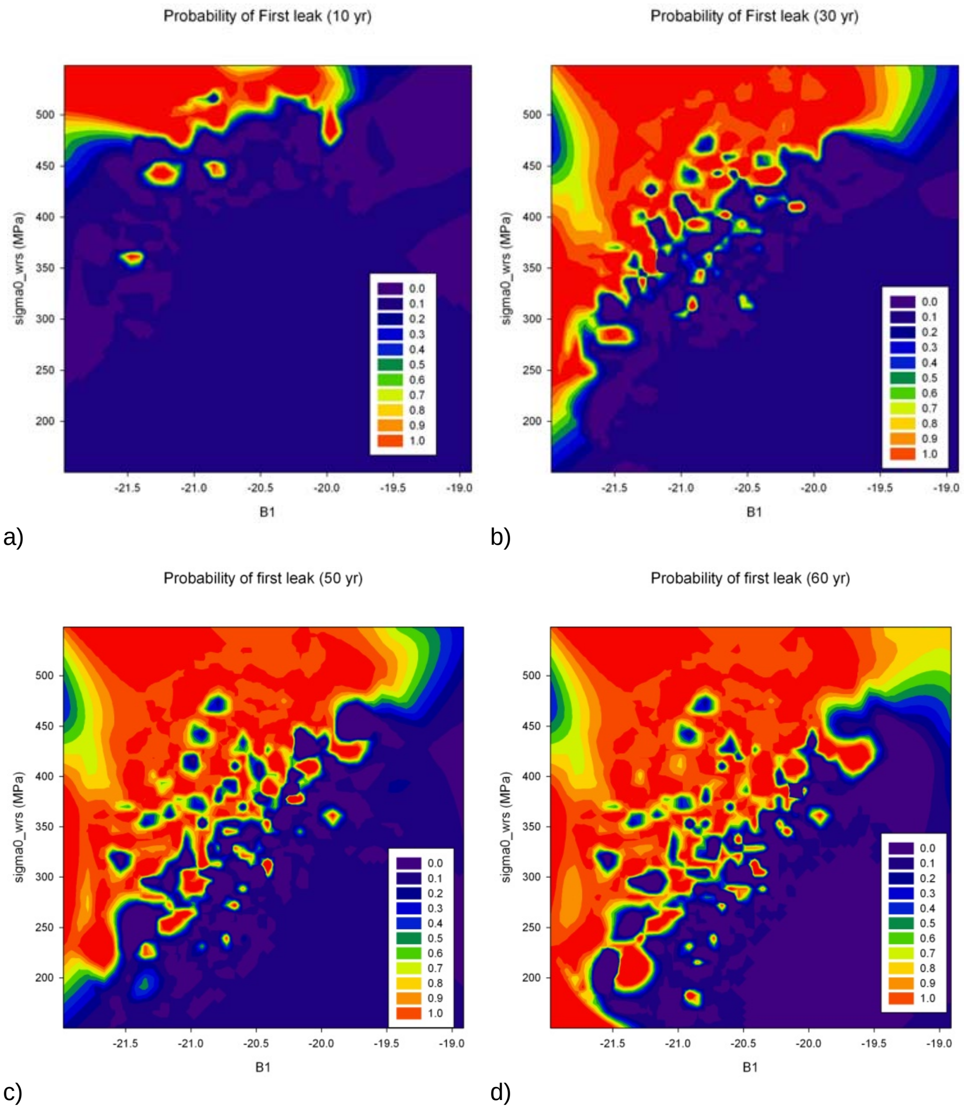


Figure 5-12. Contour Plots for the Base Case probability of First Leak at: a) 10 years, b) 30 years; c) 50 years; and d) 60 years.

5.2.2.4 Expected Total Leak Rate

Table 5-5. Stepwise regression of Expected (over aleatory) Total Leak Rate (EXPTLR) at 4 selected time-steps.

EXPTLR: 10 yr				EXPTLR: 30 yr				EXPTLR: 50 yr				EXPTLR: 60 yr			
var.	R^2	R^2 inc.	SRRC	var.	R^2	R^2 inc.	SRRC	var.	R^2	R^2 inc.	SRRC	var.	R^2	R^2 inc.	SRRC
SIG0WRS	0.09	0.09	0.10	SIG0WRS	0.31	0.31	0.38	SIG0WRS	0.40	0.40	0.54	SIG0WRS	0.42	0.42	0.58
B1	0.10	0.01	-0.03	B1	0.40	0.09	-0.20	B1	0.54	0.14	-0.32	B1	0.56	0.15	-0.34
RANDP07	0.11	0.01	-0.02	ODRAND	0.40	0.00	-0.04	RANDP05	0.54	0.00	0.04	RANDP05	0.56	0.00	0.04
TEMP	0.11	0.01	-0.02	RANDP16	0.40	0.00	0.04	RANDP06	0.55	0.00	-0.04	RANDP06	0.56	0.00	-0.04
								RANDL17	0.55	0.00	0.04	RANDP03	0.57	0.00	-0.04

Results from stepwise regression are consistent with the previous ones. The behavior displayed in the contour plots (in log scale) in Figure 5-13 is consistent with the one for fractional surface area cracked. The region of high leakage is a little larger indicating that the combination of both parameters leads to an area of greater cracks and greater leakage.

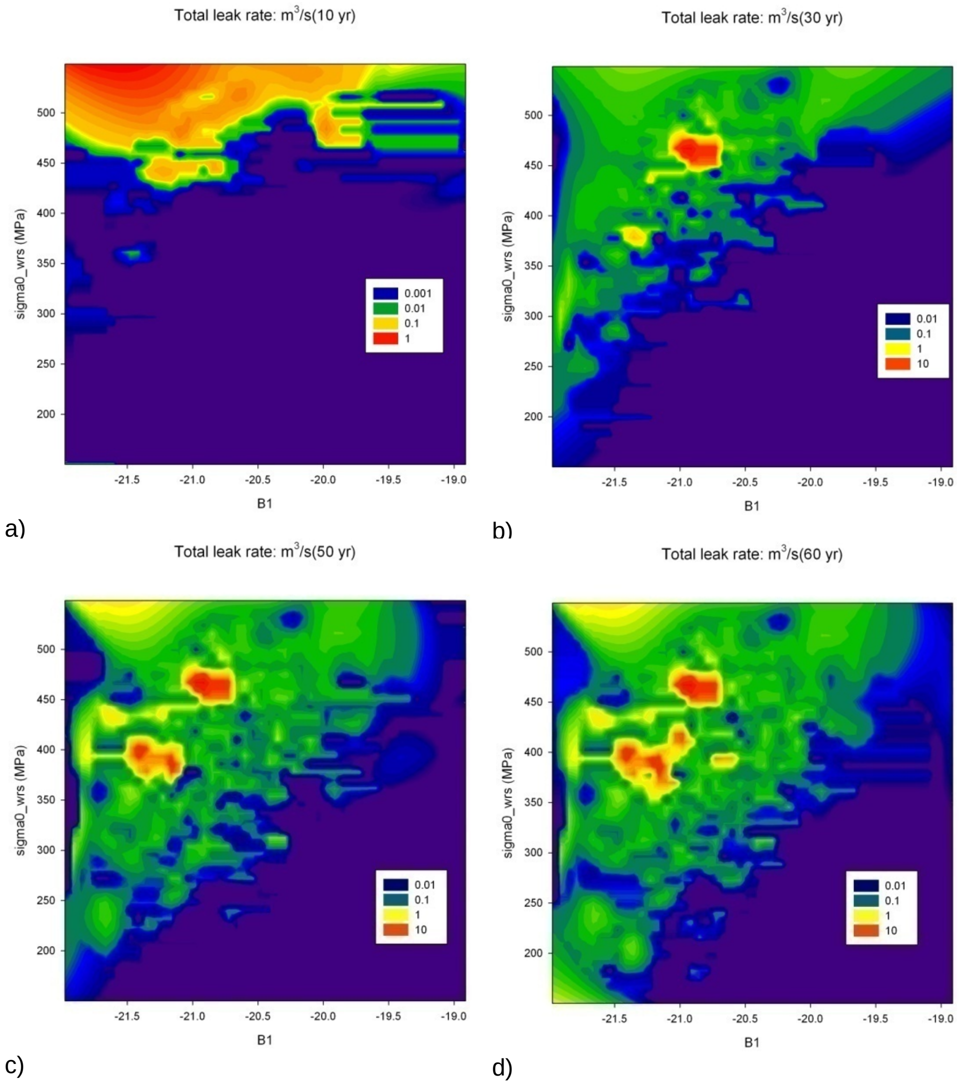


Figure 5-13. Contour plots for the Base Case total leak rate at: a) 10 years, b) 30 years; c) 50 years; and d) 60 years.

5.2.2.5 Probability of Pipe Rupture

Table 5-6. Stepwise regression of Probability of pipe rupture (PROBRUP) at 4 selected time-steps.

PROBRUP: 10 yr				PROBRUP: 30 yr				PROBRUP: 50 yr				PROBRUP: 60 yr			
var.	R^2	R^2 inc.	SRRC	var.	R^2	R^2 inc.	SRRC	var.	R^2	R^2 inc.	SRRC	var.	R^2	R^2 inc.	SRRC
SIG0WRS	0.09	0.09	0.10	SIG0WRS	0.30	0.30	0.36	SIG0WRS	0.42	0.42	0.54	SIG0WRS	0.44	0.44	0.58
B1	0.10	0.01	-0.03	B1	0.38	0.08	-0.18	B1	0.57	0.15	-0.33	B1	0.61	0.17	-0.36
RANDP07	0.11	0.01	-0.02	ODRAND	0.38	0.00	-0.04	FWELD	0.58	0.01	0.07	FWELD	0.62	0.01	0.09
TEMP	0.11	0.01	-0.02	FWELD	0.39	0.00	0.04	RANDL17	0.58	0.00	0.04	RANDP05	0.62	0.00	0.04
												ODRAND	0.62	0.00	-0.04

The sensitivity analysis for pipe rupture shows identical behavior than the other analysis performed above. The contour plots in Figure 5-14 display similar results to those seen for probability of first crack and first leak occurring.

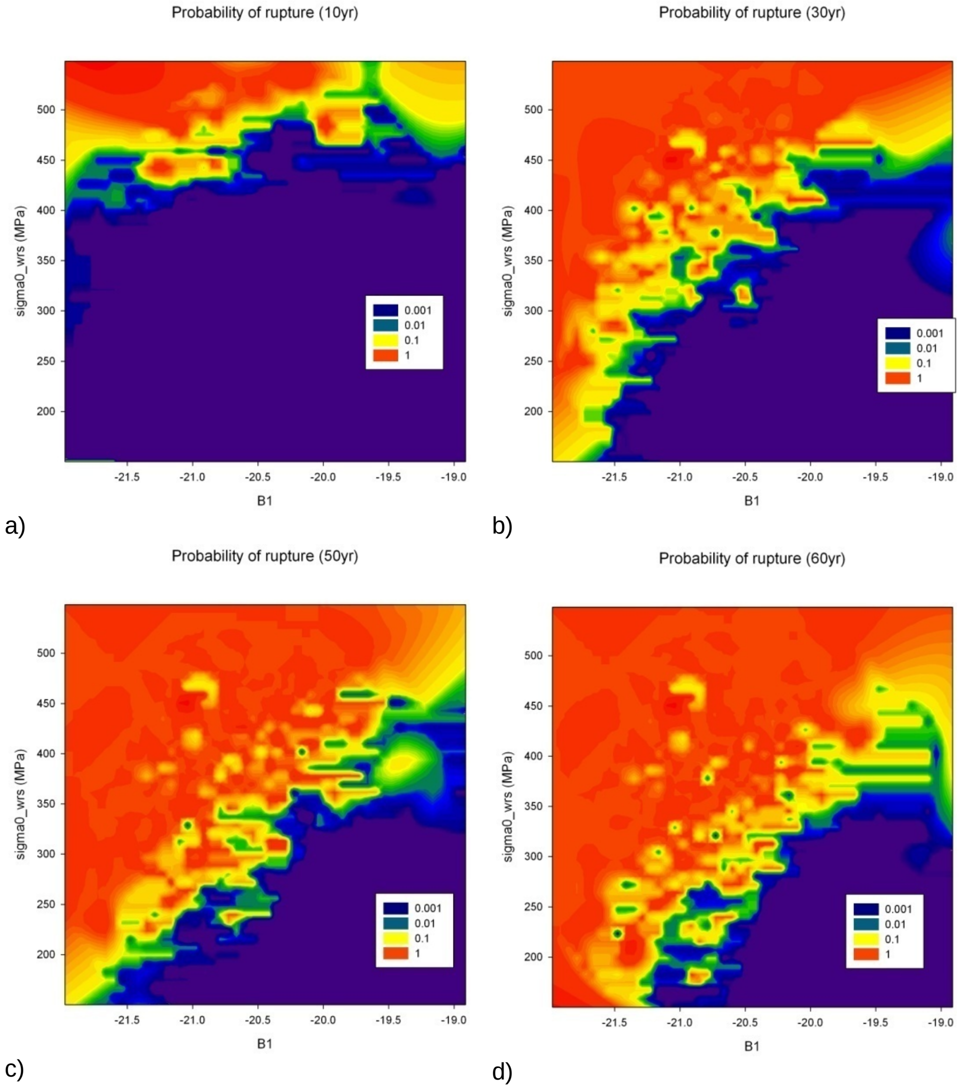


Figure 5-14. Contour plots for the Base Case probability of rupture at: a) 10 years, b) 30 years; c) 50 years; and d) 60 years.

5.2.2.6 Probability of Pipe Rupture When Leak Rate Detection (1 gpm) is Added.

Table 5-7. Stepwise regression of Probability of pipe rupture conditional on leak rate detection of 1 gpm (PROBRUP_LR1) at 4 selected time-steps.

PROBRUP_LR1: 10 yr				PROBRUP_LR1:30 yr				PROBRUP_LR1: 50 yr				PROBRUP_LR1: 60 yr			
var.	R^2	R^2 inc.	SRC	var.	R^2	R^2 inc.	SRC	var.	R^2	R^2 inc.	SRC	var.	R^2	R^2 inc.	SRC
				SIG0WRS	0.02	0.02	0.03	SIG0WRS	0.04	0.04	0.07	SIG0WRS	0.05	0.05	0.09
				B1	0.03	0.02	-0.02	B1	0.07	0.03	-0.06	B1	0.07	0.02	-0.06
				BETA1	0.04	0.00	0.01	XC	0.08	0.01	-0.04	XC	0.08	0.01	-0.04
				XC	0.04	0.00	-0.01	RANDP04	0.09	0.01	0.03	RANDL09	0.09	0.01	0.04
				RANDL15	0.04	0.00	0.01	RANDL09	0.10	0.01	0.03	RANDP04	0.09	0.01	0.03
								RANDP01	0.10	0.01	-0.03	RANDP01	0.10	0.01	-0.03
								RANDL01	0.11	0.00	-0.02	RANDL04	0.10	0.00	-0.02

In this analysis, correction for leak rate detection (at 1 gpm) is added. As a result, most of the 1000 epistemic sets lead to no rupture at all, and only between 20 and 30 runs are kept. For most of these runs, they tend to fail at the same time step they become TWC, representing long and thin SCs that fail as soon as they evolve into TWC.

The analysis is not as good as previous analyses, with very low R^2, which could be expected with only 2% of the data having meaningful results. It seems that Xc could be an important parameter with negative influence, but its effect is at the noise level and only importance sampling could confirm this behavior.

While with time the probability of failure as seen in Figure 5-15 seems to draw similar patterns as previously, it is hard to determine whether the area of low importance is linked to influence of other parameters or simply a sample size too small to generate enough cases. However, it seems clear that higher values of sigma0_wrs are important and that values below 250 are unlikely to lead to any rupture.

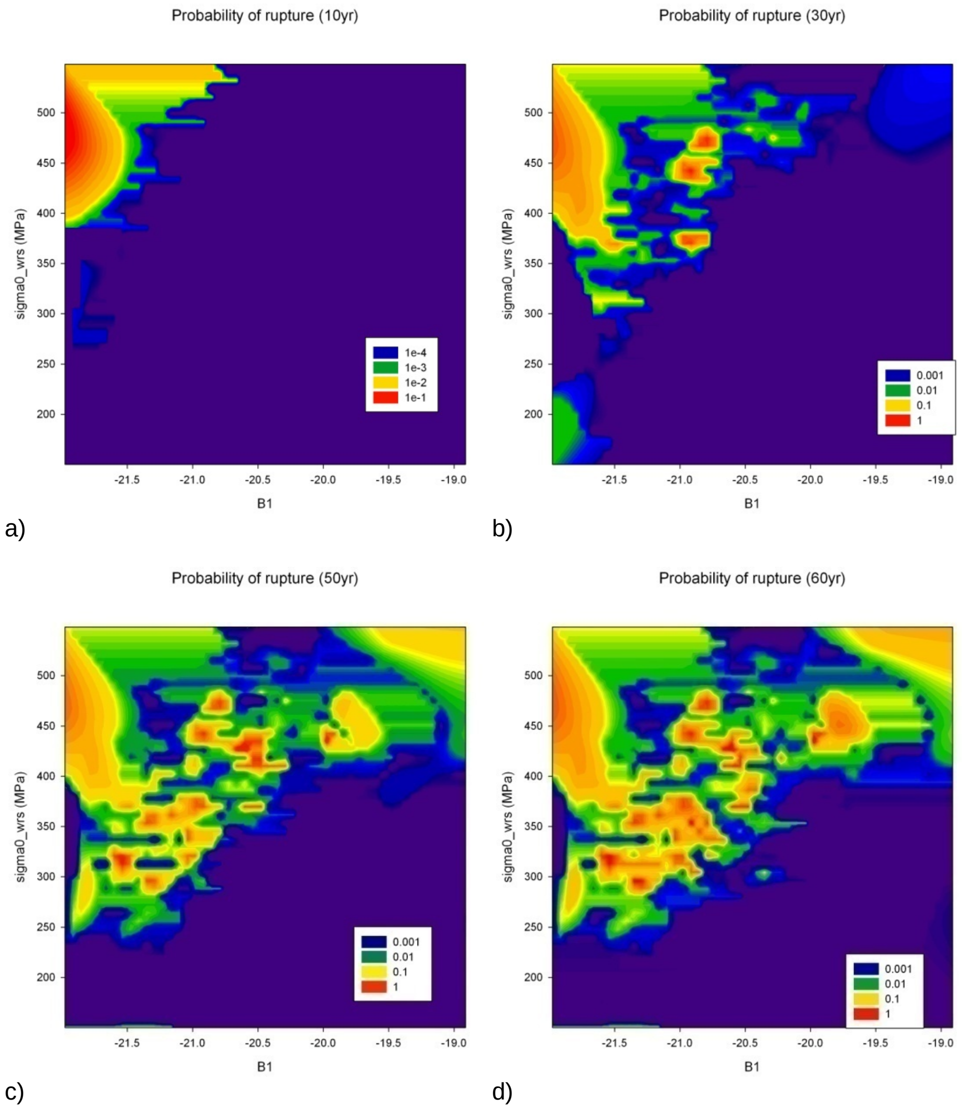


Figure 5-15. Contour plots for the Base Case probability of rupture with leak detection (1 gpm) at: a) 10 years, b) 30 years; c) 50 years; and d) 60 years.

5.2.2.7 Probability of Pipe Rupture When Inspection is Added Every 2 Years.

Table 5-8. Stepwise regression of Probability of pipe rupture conditional on inspection every 2 years (PROBRUP_INS2) at 4 selected time-steps.

PROBRUP_INS2: 10 yr				PROBRUP_INS2: 30 yr				PROBRUP_INS2: 50 yr				PROBRUP_INS2: 60 yr			
var.	R^2	R^2 inc.	SRRC	var.	R^2	R^2 inc.	SRRC	var.	R^2	R^2 inc.	SRRC	var.	R^2	R^2 inc.	SRRC
SIG0WRS	0.09	0.09	0.09	SIG0WRS	0.26	0.26	0.32	SIG0WRS	0.34	0.34	0.48	SIG0WRS	0.35	0.35	0.52
B1	0.09	0.01	-0.02	B1	0.32	0.06	-0.15	B1	0.44	0.09	-0.26	FWELD	0.44	0.09	0.28
TEMP	0.10	0.01	-0.02	FWELD	0.34	0.02	0.09	FWELD	0.50	0.06	0.20	B1	0.53	0.10	-0.27
FWELD	0.10	0.01	0.02	RANDP07	0.34	0.00	-0.04	RANDP06	0.50	0.00	-0.05	RANDL03	0.54	0.00	0.06
RANDP07	0.11	0.00	-0.02					RANDL17	0.50	0.00	0.04	BETA2	0.54	0.00	-0.04
RANDP05	0.11	0.00	0.02									RANDP03	0.54	0.00	-0.04
												RANDP06	0.54	0.00	-0.04

This analysis supposes that an inspection is performed every two years, increasing the chances to find a crack before it leads to rupture. Every TWC existing at the time of the inspection will be repaired. Results are then slightly different as Fweld (weld factor – see Table D1.2) now has some importance that increases over time. Its SRRC is positive, meaning that high values of Fweld will lead to higher values of probability of rupture.

This behavior is reflected on the contour plots in Figure 5-16. While the limit between having a crack or not is still visible, several areas in the domain where rupture occurred without inspection are now without rupture. Moreover, a region of importance is now strongly visible for the highest values of sigma0_wrs. A scatterplot of rupture vs. Fweld at 50 yrs (on a log scale) shows a noticeable impact of the input parameter on the uncertainty of the results (Figure 5-17).

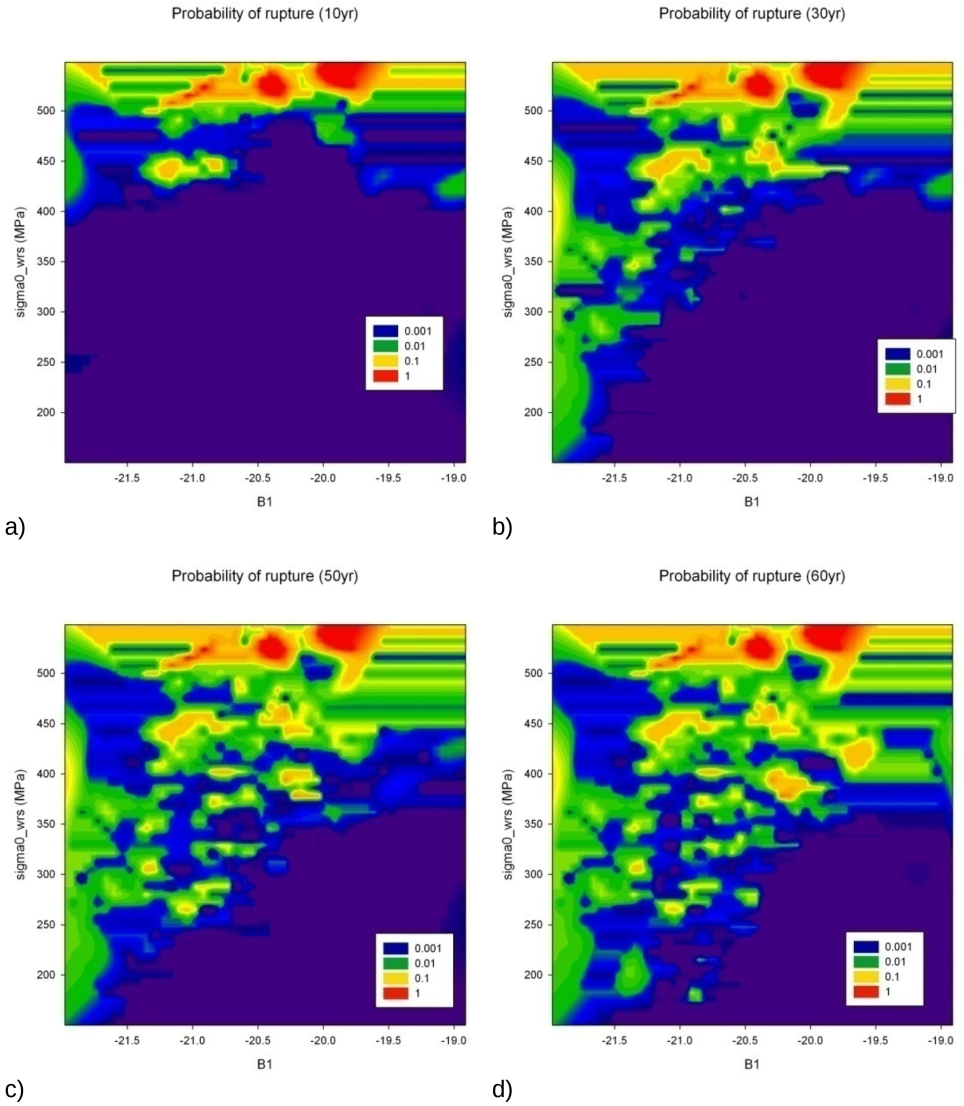


Figure 5-16. Contour plots for the base case probability of rupture with inspection every 2 years at: a) 10 years, b) 30 years; c) 50 years; and d) 60 years.

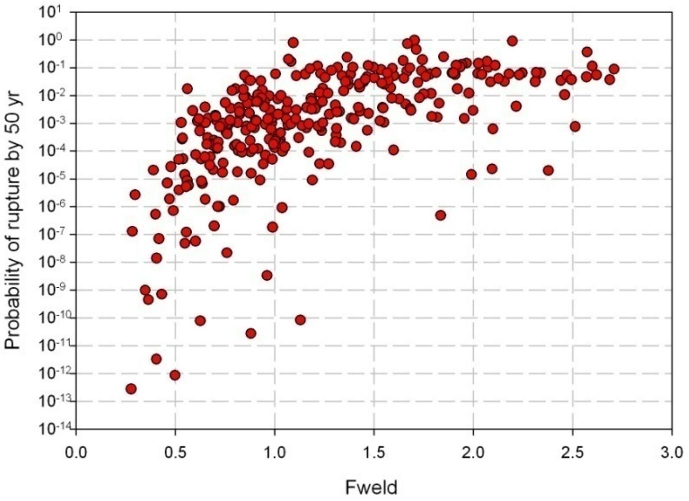


Figure 5-17. Base Case Fweld versus probability of rupture results with inspection every 2 years.

5.2.2.8 Non monotonic influence of *Xc*

The stepwise regression is designed to capture linear (when raw data is used) and monotonic (when rank data is 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 *Xc* parameter. However for many of the xLPR subject matter experts, *Xc* was expected to show up as an influential parameter. Usually, when one suspects that a parameter may have some kind of influence on a selected output, the relation between the input-output can be interrogated using scatterplots or box plots. However, the output of interest is the probability of rupture and since aleatory uncertainty has a very low influence, this probability is mostly equal to 0 or 1. Therefore, scatterplots and box plots will mainly display values of 0 and 1 making it difficult to see the influence of *Xc*.

Rather, a bubble plot has been used to determine what the influence of Xc was. In a bubble plot, the distribution of Xc has been divided into equally probable quantiles on the x-axis (meaning that the same number of realizations are used for each bubble). For each quantile, a bubble is used to represent how many realizations lead to a probability of rupture. In consequence, larger bubble sizes indicate more rupture. This analysis has been applied every five years, and is reported on the Y-axis. Figure 5-18 displays the resulting bubble plot. The non-monotonic influence of *Xc* can be seen distinctly: the lowest and highest quantiles lead to rupture more frequently than the center of the distribution. It confirms that Xc is an important parameter with respect to the uncertainty over probability of rupture.

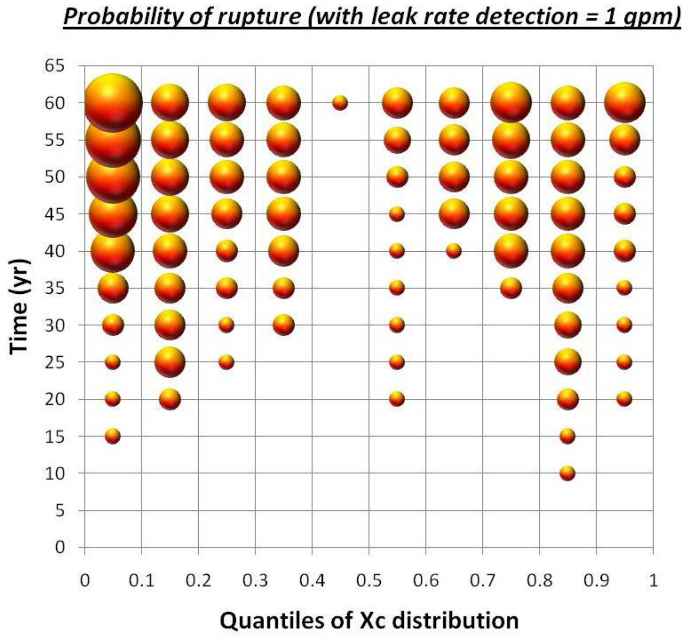


Figure 5-18. Bubble plot representing the non-monotonic influence of Xc over the probability of rupture (with leak rate detection set to 1 gpm) over time

5.3 Sensitivity Cases

The following sections present the results of sensitivity analyses conducted to demonstrate xLPR model functionality. These 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. The analyses presented do not represent an exhaustive investigation; rather they verify the framework capabilities and demonstrate the analyses that need to be refined in Version 2.0 of the xLPR code development.

5.3.1 Crack Initiation

A sensitivity case was run that considers the crack initiation model uncertainty. Two approaches were incorporated into the crack initiation module; a Direct Method and a Weibull solution. In the Direct Method the initiation time is an explicit function of stress and temperature, with randomness in the parameters used in the function. The crack initiation module includes three alternative models for crack initiation since the Direct Method approach also has two different methods available. Since Direct Method II was used in the base case analysis, Direct Method I was run for comparison to the base case to evaluate the effect of the initiation model uncertainty on the results. The Weibull approach was used in the Alpha model and was not evaluated in this sensitivity analysis.

The input parameters used in this sensitivity analysis are provided in a calibration document for the crack initiation module provided by the Models Task Group. The option to select Direct Method II for the crack initiation module is included with the default options provided in the xLPR framework model. No model errors were logged by the GoldSim software. However, in 2 of the 50,000 realizations SCFail reported an error code of -13, theta ≥ Pi. Likewise for those same 2 realizations an error code of -1 (theta ≥ Pi) was reported for the SQUIRT Module. SCFail will not provide a solution in this case. SQUIRT would fail, but the framework model instead sends SQUIRT a value of 0.99*theta in order to prevent this from happening as detailed in Section 3.0. Coalesce module limits the half crack length to be Pi and it takes one time step for TWC to fail, which occurs 2 realizations of the 50,000. This condition can be error trapped by sending SCFail and TWCFail a value of 0.99*Pi (slightly below the maximum value of Pi) or by modifying the coalesce module to send a maximum value of 0.99*Pi.

A discussion of the conceptual basis for the underlying models used in the crack initiation module is presented in the xLPR Models Task Group report.

5.3.1.1 Number of Cracks Predicted – Crack Initiation Sensitivity

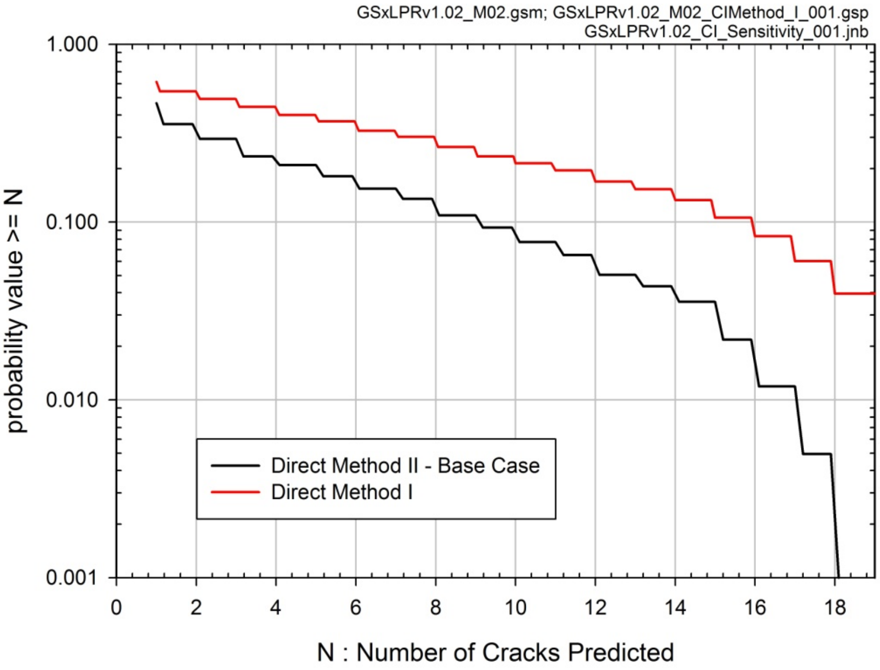


Figure 5-19. Comparison of Direct Method I and Direct Method II – Mean CCDF of the Number of Cracks Predicted.

5.3.1.2 Number of Cracks Initiated – Crack Initiation Sensitivity

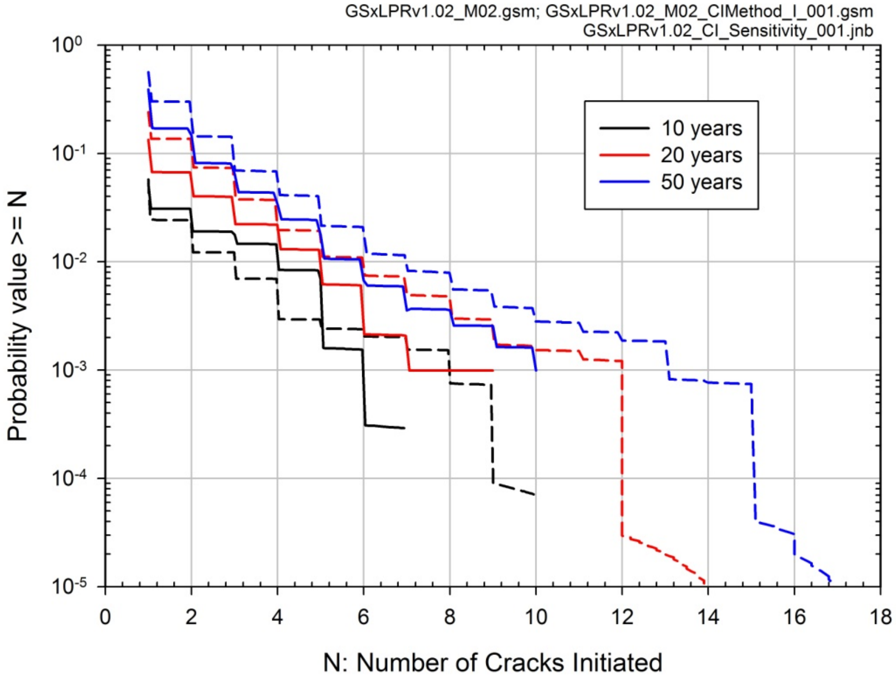


Figure 5-20. Comparison of Direct Method I and Direct Method II – Mean CCDFs of the Number of Cracks Initiated at various times. Solid lines are base case results (Direct Method II); dashed lines are for Direct Method I, without inspection or leak detection.

5.3.1.3 Probability of Rupture – Crack Initiation Sensitivity

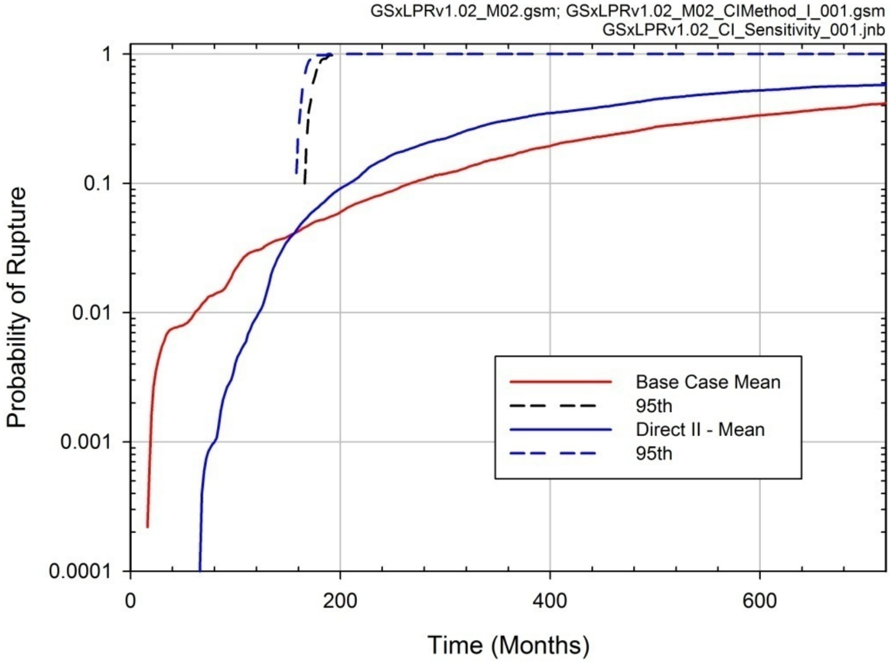


Figure 5-21. Comparison of Direct Method I and Direct Method II – Probability of Rupture versus time, without inspection or leak detection.

The mean CCDFs of the number of cracks predicted to occur in 60 years shown in Figure 5-19, indicates that Direct Method I of the crack initiation module predicts more cracks then with Direct Method II. It seems that the majority of the cracks predicted using Direct Method I occur after 10 years as shown in Figure 5-20. The CCDF at 20 years for Direct Method I is similar to the base case CCDF at 50 years, with a small likelihood of a larger number of cracks. The CCDF at 50 years for Direct Method I is higher than the base case and shows a much higher likelihood to initiate more cracks. The distribution of CCDFs for the base case, also shown in Figure 5-20, shows a spread in curves that indicate the cracks initiated more evenly distributed over time. Additionally, the CCDFs indicate that after 10 years there is a much higher likelihood of more cracks initiating using Direct Method I.

A fewer number of cracks are initiated before 10 years using Direct Method I, but after 10 years, as shown in Figure 5-20, more cracks are initiated. The probability can be expressed as a function of a rate (i.e., frequency) of crack initiation, λ:

$$p = e^{\lambda \cdot t_{Max}}$$

Where, t_{max} is the duration of the simulation, 60 years and p is the probability of at least one crack. The probability of having at least one crack is shown on Figure 5-19 and is 0.615 for Direct Method I and 0.466 for Direct Method I; the majority of the realizations do not have crack occurrence in the base case analysis. Rearranging the equation above to solve for lambda:

$$\lambda = -(\frac{1}{t_{Max}}) \times \ln(1 - p)$$

Overall the frequency of crack initiation for Direct Method II is around 0.016 yr-1, whereas using Direct Method II it is approximately 0.010 yr-1.

Figure 5-21 is a comparison of the mean, 5th and 95th percentile values for the probability of rupture. The figure shows that the mean probability of rupture using Direct Method I is about 0.58 at about 60 years whereas in using Direct Method II, the probability of rupture never exceeds around 0.42. Therefore, using Direct Method I given a probability on average of 0.615 that there is at least one crack and a rupture probability of 0.58; there is a 93% change that the occurrence of one crack or more leads to rupture if left unmitigated. Using Direct Method II, only 89% change that the occurrence of one crack or more leads to rupture if left unmitigated. This difference is likely due to the greater likelihood of having more cracks initiated using Direct Method I. Of note in Figure 5-21 is that although Direct Method I produces a higher probability of rupture at 60 years, the probability of rupture before 10 years is lower than using Direct Method II.

5.3.2 Stress Mitigation

The Stress Mitigation analysis was run to evaluate different mitigation times, as well as the mitigation effectiveness over the representative distributions for *Sigma0_wrs_mitigated* and *Xc_mitigated*. Three sensitivity cases were run for mitigation:

Mitigation time 10 years
Mitigation time 20 years
Mitigation time 30 years

The analyses used a normal distribution for WRSs (Sigma0_WRS) after mitigation defined by: mean = -344.75 MPa, stdev = 34 MPa, min = -447 MPa and max = -242 MPa. In addition after mitigation the Xc value was changed to a normal distribution with: mean = 0.38, stdev = 0.038, min = 0.26 and max = 0.5. It should be noted that the problem statement (Appendix E) lists the mitigation times at 10, 20, and 40 years, however a 30 year mitigation time was run in place of the 40 year case. The 30 year case was selected for equal mitigation time intervals and as it was expected to show a larger difference between the base case results than mitigation that occurred at 40 years.

The input parameters Sigma0_WRS_mitigated and Xc_mitigated used in this sensitivity analysis are provided in the controlled inputs file (Excel spreadsheet). However several constants on the input spreadsheet were modified to change the mitigation time, the number of intervals in the operating history (MTS interval) to 2, and provide the intervals for MTS1 equal to the mitigation time and MTS2 equal to the mitigation time – 720 months. The option to select mitigation is included with the default options provided in the xLPR framework model on the Main Dashboard.

5.3.2.1 Fraction of Surface Area Cracked – Mitigation Sensitivity

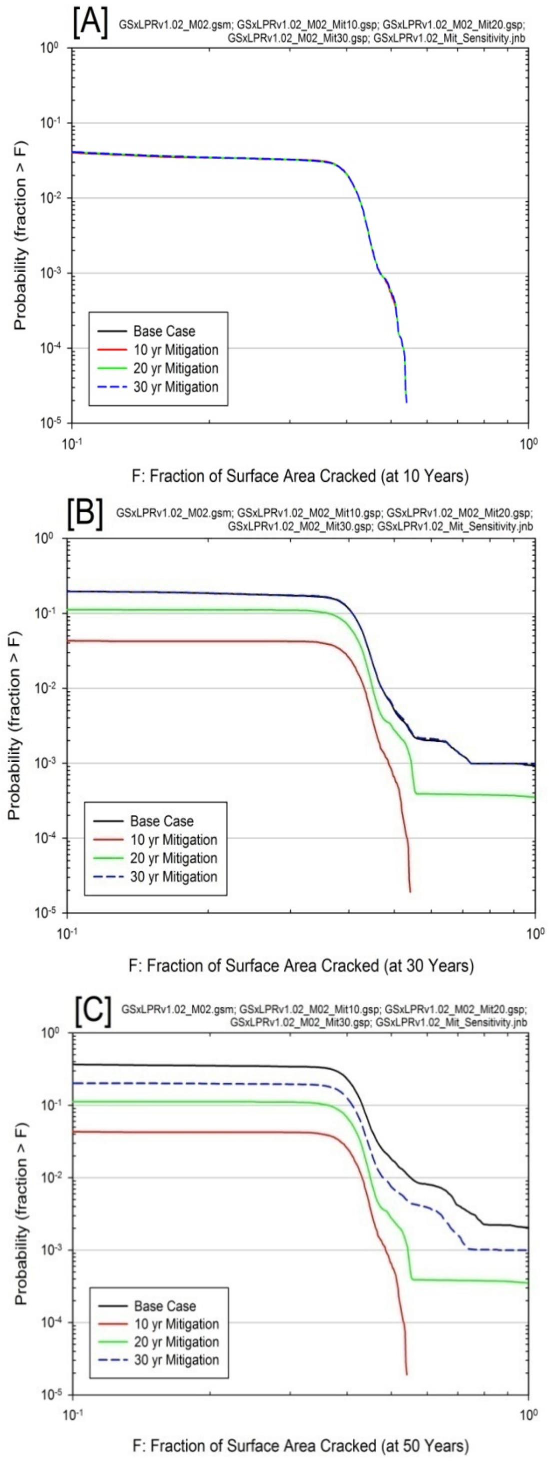


Figure 5-22. Comparison of Mitigation Cases with Base Case – Mean CCDFs of the Fraction of the Surface Area Cracked at a) 10, b) 30 and c) 50 years, without inspection or leak detection.

5.3.2.2 Probability of Rupture – Mitigation Sensitivity

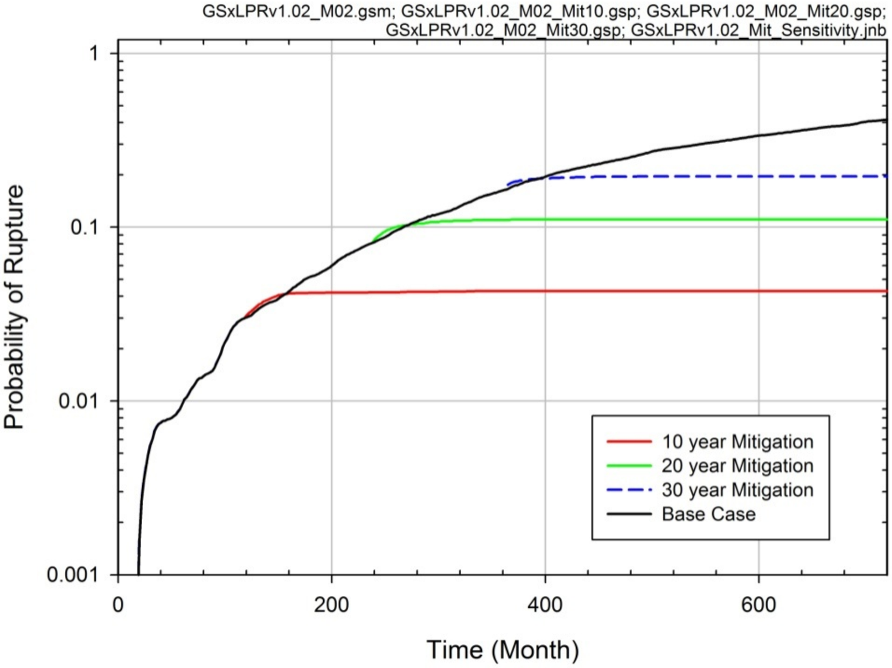


Figure 5-23. Comparison of Mitigation Case with the Base Case – Mean Probability of Rupture versus time, without inspection or leak detection.

The CCDFs of the fractional surface area cracked in Figure 5-22 show that mitigation for all three cases begins to show increasing effects with increasing time, with the 10 year mitigation having the greatest effects. Likewise, the probability of rupture shown in Figure 5-23 shows only rare early ruptures occur only before the mitigation events. In these simplified test cases, the cracks are effectively mitigated as the sigma0_wrs after mitigation has a maximum value of -242 MPa, which is not high enough stress for cracks to grow after the mitigation event occurs. This is shown in the constant value after the mitigation event for the probability of rupture.

5.3.3 Chemical Effects

A sensitivity case was run that considers the effects of increasing the hydrogen concentration in the water only as it affects the 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 Version 1.0. In the base case analysis presented in Section 5.2, the hydrogen concentration in the water used in the grower module was set at 25 cc/kg-STP. Two modeling cases were run with the hydrogen concentration in the water used for the grower module increased to 50 and 80 cc/kg-STP to demonstrate the effects of hydrogen in the water. No changes were made to the framework model options, and the base case module options were used in this analysis. The constant parameter, H2, listed in the controlled input file was changed to the higher values. No model errors were logged by the GoldSim software or reported by the xLPR modules.

5.3.3.1 Fraction of Surface Area Cracked – Chemical Effects Sensitivity

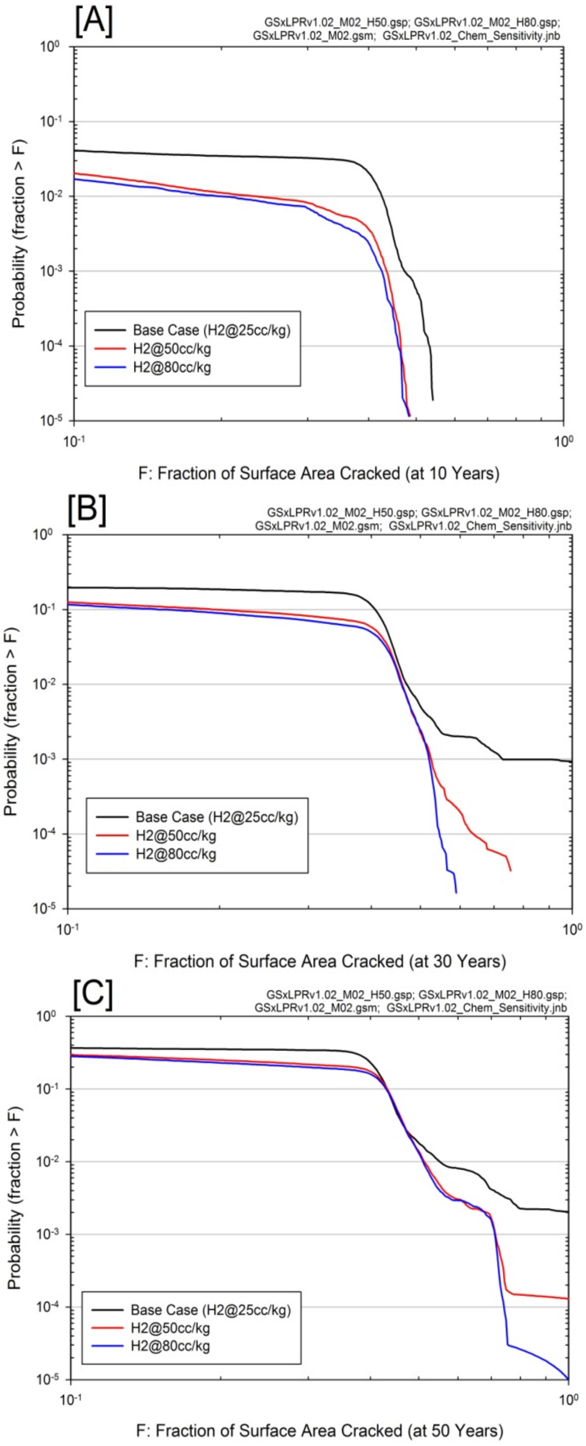


Figure 5-24. Comparison of Chemical Effects Cases with Base Case – Mean CCDFs of the Fraction of Surface Area Cracked at a) 10, b) 30 and c) 50 years, without inspection or leak detection.

5.3.3.2 Probability of Rupture– Chemical Effects Sensitivity

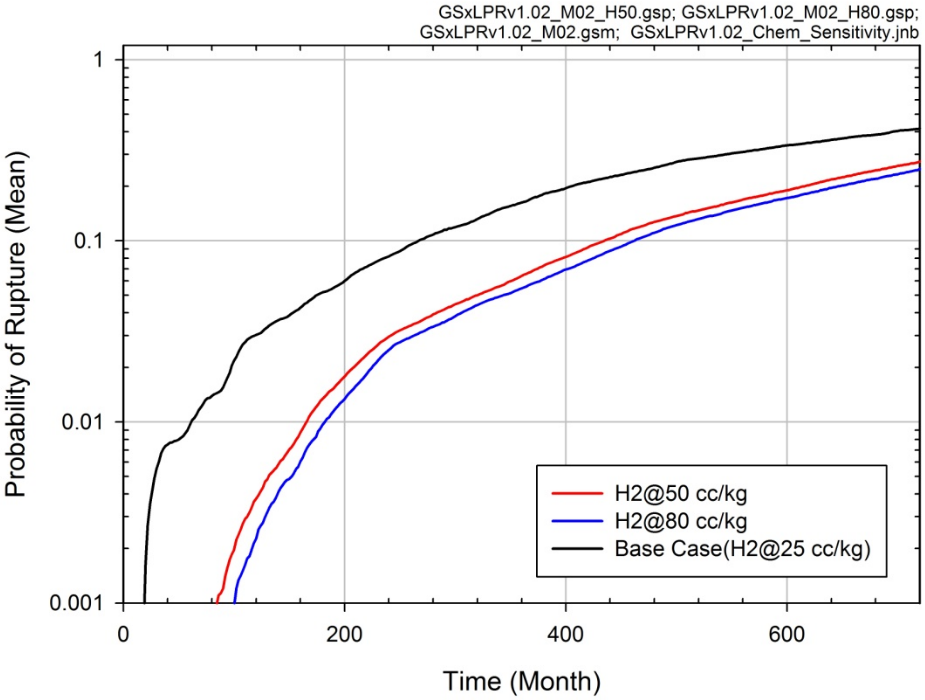


Figure 5-25. Comparison of Chemical Effects Cases with the Base Case – Mean Probability of Rupture versus time, without inspection or leak detection.

5.3.3.3 Leakage– Chemical Effects Sensitivity

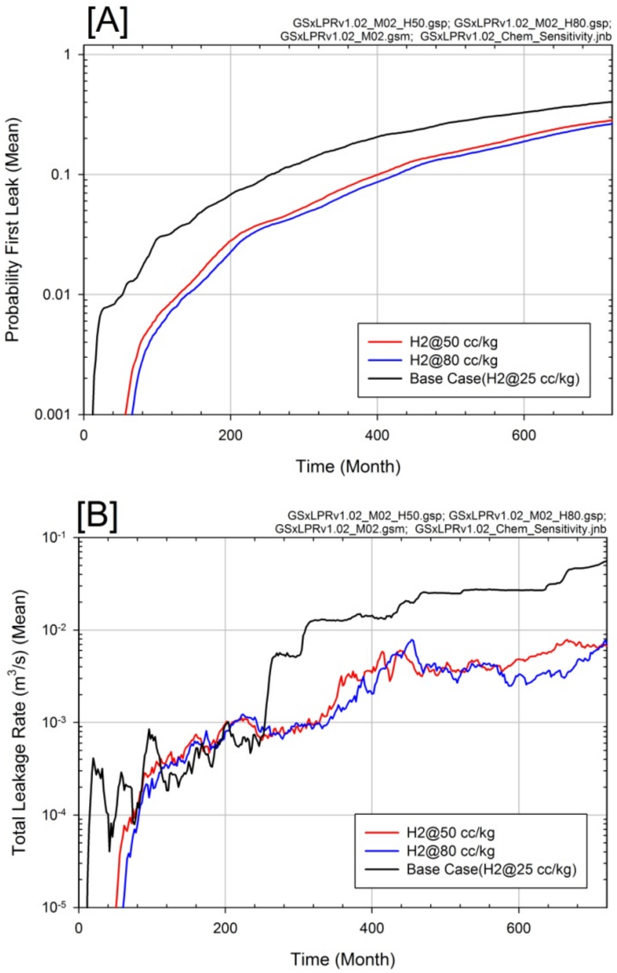


Figure 5-26. Comparison of Chemical Effects Cases with the Base Case: a) Mean Probability of First Leakage versus time and b) Mean Total Leakage Rate versus time, without inspection or leak detection.

It is clear from in the CCDFs of the fractional surface area cracked in Figure 5-24 that the presence of hydrogen slows down the crack growth, as evidenced by the increasing likelihood of lower fractions as the hydrogen concentration increases from 25 cc/kg to 80 cc/kg. This is true for at all three time points selected, 10, 30, and 50 years.

The probability of rupture is lower for the cases with higher concentrations of hydrogen, as shown in Figure 5-25. The difference between the probability of rupture for the base case and the cases with increased hydrogen decreases slightly with time. The decrease in the probability of rupture indicates that due the slower crack growth the cracks are less likely to become TWCs. Figure 5-26 shows the mean probability of first leakage and the mean total leakage rates. The lower probability of first leakage and the lower mean total leakage rates demonstrate the lower occurrence of TWC in cases with higher hydrogen concentrations. One final observation is that it appears that the grower module is more sensitive to the increase from 25 cc/kg (base case value) to 50 cc/kg, than from 50 cc/kg to 80 cc/kg, with noticeable differences between the CCDFs for fractional surface area occurring at the highest values for the 50 cc/kg and 80 cc/kg cases.

5.3.4 Safe End Evaluation

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 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 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 is assumed that the distribution is normal. For Sigma0_WRS: mean = -16.2 MPa, stdev = 117 MPa, max = 300 MPa, and min = -300 MPa). For Xc: mean = 0.18, stdev = 0.036, max = 0.5, and min = 0.1. No changes were made to the framework model options, the base case module options were used in this analysis. No model errors were logged by the GoldSim software or reported by the xLPR modules.

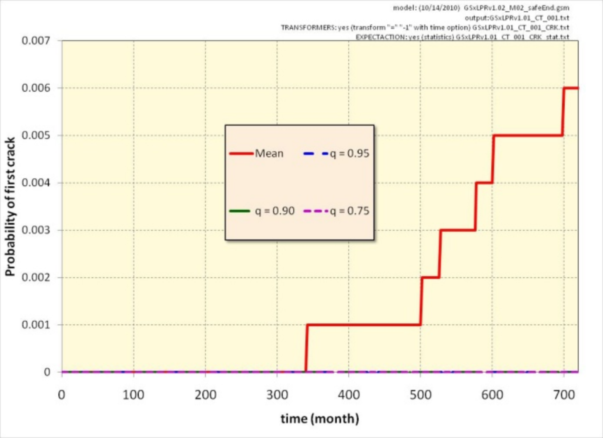
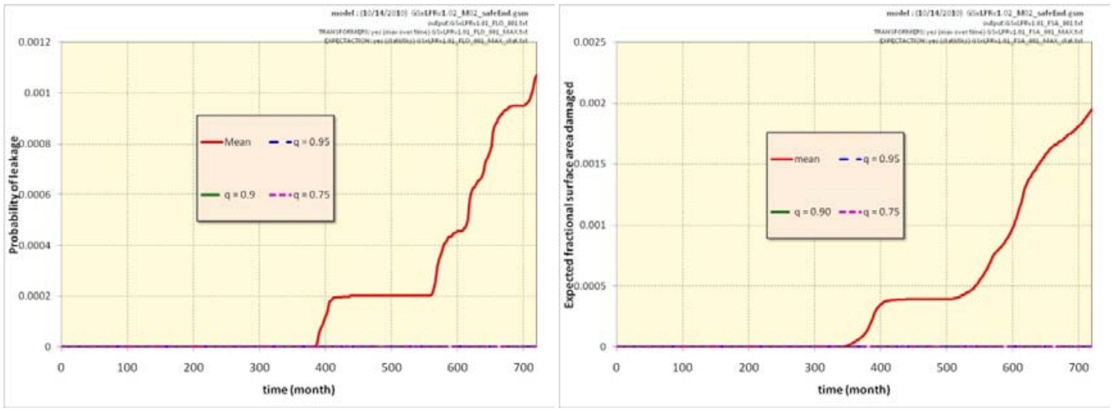


Figure 5-27. Mean and quantiles values over probability of having at least one crack for the Safe End case.

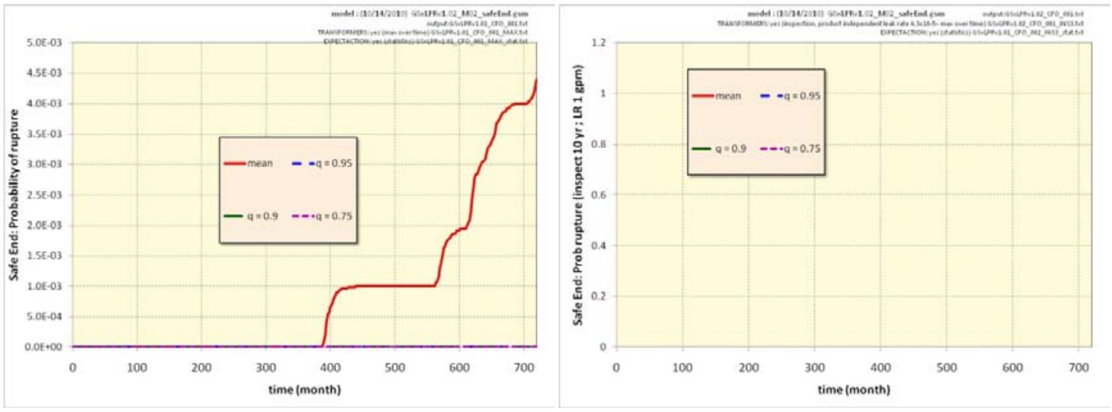
The probability of having at least one crack is reduced by a factor of 100 (two orders of magnitude) when compared to the base case. With an epistemic sample size of 1000, only 6 epistemic realizations lead to a crack occurring. The quantile selected (3rd quartile, 9th decile and 95th percentile) are all equal to zero (as the first positive value is only seen on the quantile, *q* = 0.994).



a) b)

Figure 5-28. Mean and quantile values over probability of having leakage and expected (over aleatory) fractional surface area damaged for the Safe End case.

As a result, the probability of leakage (Figure 5-28a) loses also two orders of magnitude (10^{-3}) after 60 years. The expected fractional surface area behavior is similar in shape to the probability of leakage.



a) b)

Figure 5-29. Mean and quantile values over probability of rupture for Safe End case (a) without any inspection or leak rate detection (b) with leak rate detection set at 1 gpm (~ 6.3x 10^{-5} m^3/s) and inspection every 10 yr.

The probability of rupture is around 4.5 x 10^{-3} after 60 yr, without any inspection. As so few realizations lead to at least one crack occurring the inclusion of leak rate detection and inspection goes beyond the estimate of the model and gives a 0 probability. Either a larger (epistemic) sample size or the use of importance sampling (presented in the following section) is necessary.

5.3.5 Importance Sampling with DPD

To evaluate the framework model's capability to utilize alternative sampling and to evaluate the need for importance sampling to get stable results for the low probability events, a sensitivity case was run that substitutes the DPD method for the default Monte Carlo sampling of the uncertain parameters. The stainless steel safe end weld sensitivity case described in Section 5.3.4 was selected for analysis using DPD with importance sampling due to the low probability of rupture for this case. The safe end values for Sig0_WRS and Xc were importance sampled, using P3 value of -0.00833333 and -0.0125 respectively in order to sample more frequently the tail of the distribution without going out of bound. P3 defines the upper and lower limit for importance sampling. When P3 is used then the P3 and (1-P3) percentile values are determined. These will not be equal probability intervals as opposed to the standard DPD method. A third variable (B1) was sampled using importance strategy with a P3 value of -0.0005, for a DPD bin size of 80. The remaining samples were sampled with DPD, but without importance sampling. The DPD sample set was sampled for 10,000 realizations of the outer loop. The results are presented below.

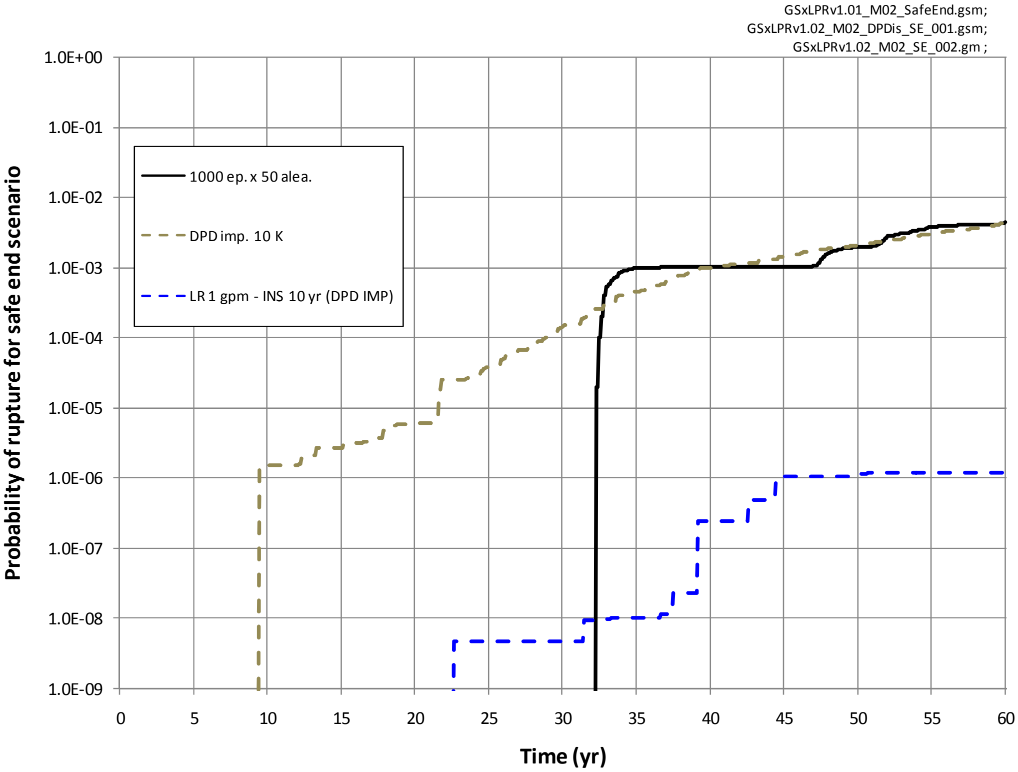


Figure 5-30. Mean probability of rupture for Safe End case using importance sampling with DPD without any inspection or leak rate detection (green dash curve) and with leak rate detection set at 1 gpm (~ 6.3x 10^{-5} m^3/s) and inspection every 10 yr (blue dashed curve) compared to probability of rupture for Safe End using reference sample (1000 epistemic x 50 aleatory).

The estimate for probability of rupture obtained using DPD size of 10,000 with importance sampling on the higher values of Sig0_wrs and Xc are a little higher than the values estimated without importance sampling (Figure 5-30). This is not surprising as so few runs lead to crack occurrence. Importance sampling generates more realizations with cracks, allowing a more accurate estimate of the probability of rupture in consequence. The smoothness of the curve over time (which is expected) in the case of importance sampling is also a good sign of convergence. As shown in Section 4.3.2 the bootstrapped distribution over the mean probability increases confidence in the stability of such result.

The real gain of using importance sampling is visible when inspection (every 10 yr in the result plotted) and leak rate detection (set at 1 gpm) are added. While simple Monte Carlo and LHS was not able to capture a single realization with a rupture, importance sampling leads to an estimate around 7 x 10^{-5} for probability of rupture. The curve is not as smooth as the previous one, indicating some instability, but the order of magnitude is likely to be representative.

5.3.6 Importance Sampling with LHS

Base Case and Safe end scenario were used to demonstrate the feasibility of importance sampling over LHS distribution. The approach consists in replacing the initial distribution of selected parameters considered as important such that the region of interest is covered more precisely, and then apply a corrective term to represent the importance of the region (or “weight”) covered by each realization. This method is not automatic and requires several steps in order to work.

5.3.6.1 Selection of input parameters

The first step in importance sampling is the determination of the important uncertain parameters. This selection is based on the sensitivity analysis performed on the probability of rupture. Sensitivity studies of the probability of rupture show that two parameters have a significant impact: sigma0_wrs (with highest value leading to more occurrence of rupture) and B1 (with lowest value leading to more occurrence of rupture). Moreover, Section 5.2.2.8 demonstrates the importance of Xc, with the occurrence of rupture increasing for high or low values of Xc (in opposition to centered values). Based on the sensitivity analysis, these three parameters were selected for importance sampling. All the other input parameters are sampled accordingly to their original distribution. The next step consists in selecting a new distribution that will cover the area of interest in greater detail than the original distribution.

5.3.6.2 Selection of appropriate distribution for each input parameter

In order to improve the accuracy of the estimate, it seems that covering the high range of the sigma0_wrs distribution and the low range of the B1 distribution when sampling these values. Of course, forcing more realizations in this region (e.g., more samples) will over cover the region better, but will lead to a disequilibrium over the entire sample space which needs to be corrected using weights. The second problem is to find exactly how many points need to be put in the region of interest and need to be left to cover the remainder of the sample space to have a correct estimate of the probabilities. As a test it was decided to use log-triangular distribution, as the formula is fairly simple and as the coverage was still broad over the whole domain.

The original distribution for sigma0_wrs was a truncated normal distribution with a mean of 300.3 MPa, a standard deviation of 110 MPa and minimum and maximum set to 150 and 551 MPa respectively. It is hard to determine upfront how strong the importance correction should be, as the purpose is to cover the region in the input space with greatest probability, leading to rupture. When several inputs are involved, the mathematical formulation can be a lot more complex: for instance, sensitivity analysis showed that it was possible to have rupture for lowest values of sigma0_wrs as long as B1 was small enough. Therefore, it was decided to select a distribution that was favoring the upper end of the sigma0_wrs distribution but with still keeping a good coverage of the rest of the distribution. In the base case for sigma0_wrs, the log-triangular distribution parameters were a minimum of 150 MPa, a maximum of 551 MPa and a mode of 550 MPa. The following figures (Figures 5-31 and 5-32) present a PDF and CDF comparison of the original distribution and importance distribution using a sample size of 1000 (for the purpose of illustration). As can be seen in Figure 5-31, the high end of the distribution has a better coverage with the log triangular distribution than with the original truncated normal distribution.

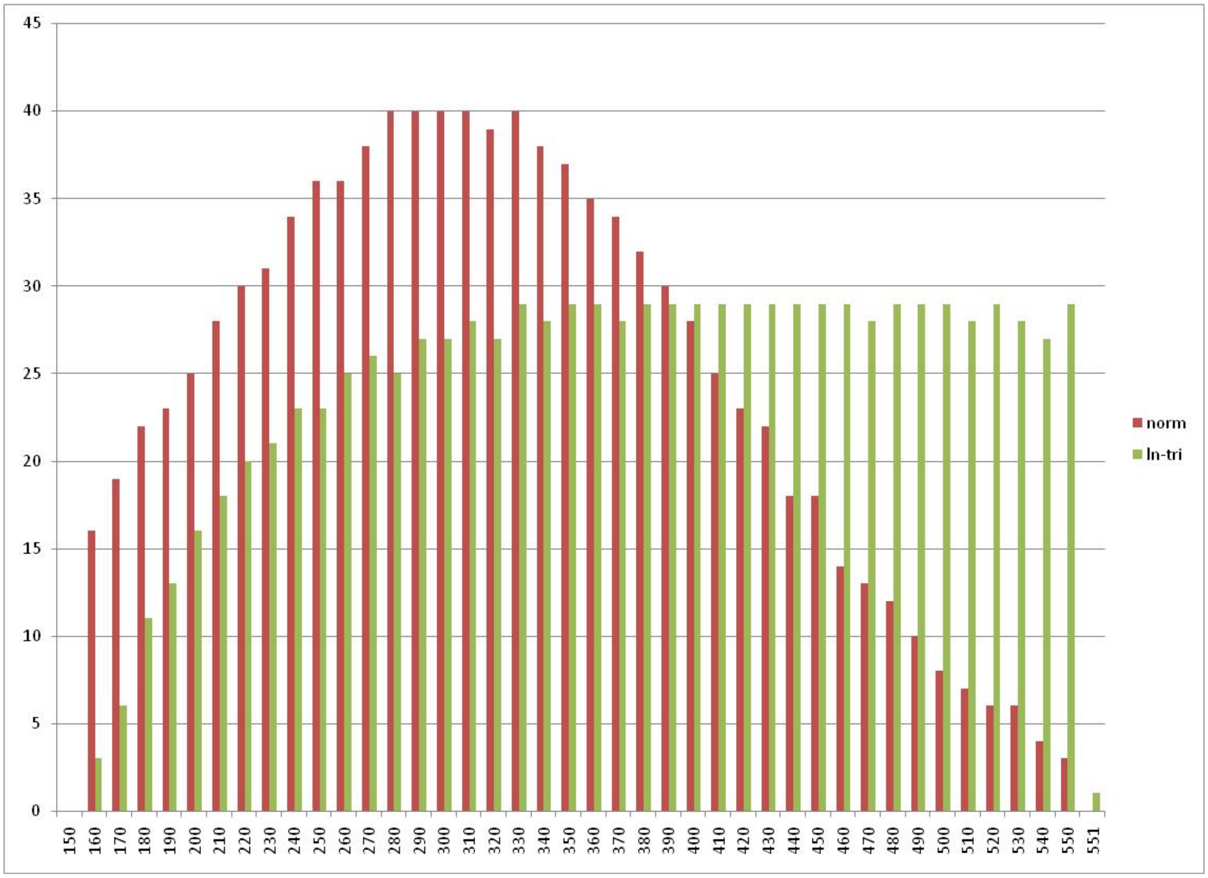


Figure 5-31. PDF comparison of original (truncated normal) distribution and importance (log-triangular) distribution for sigma0_wrs.

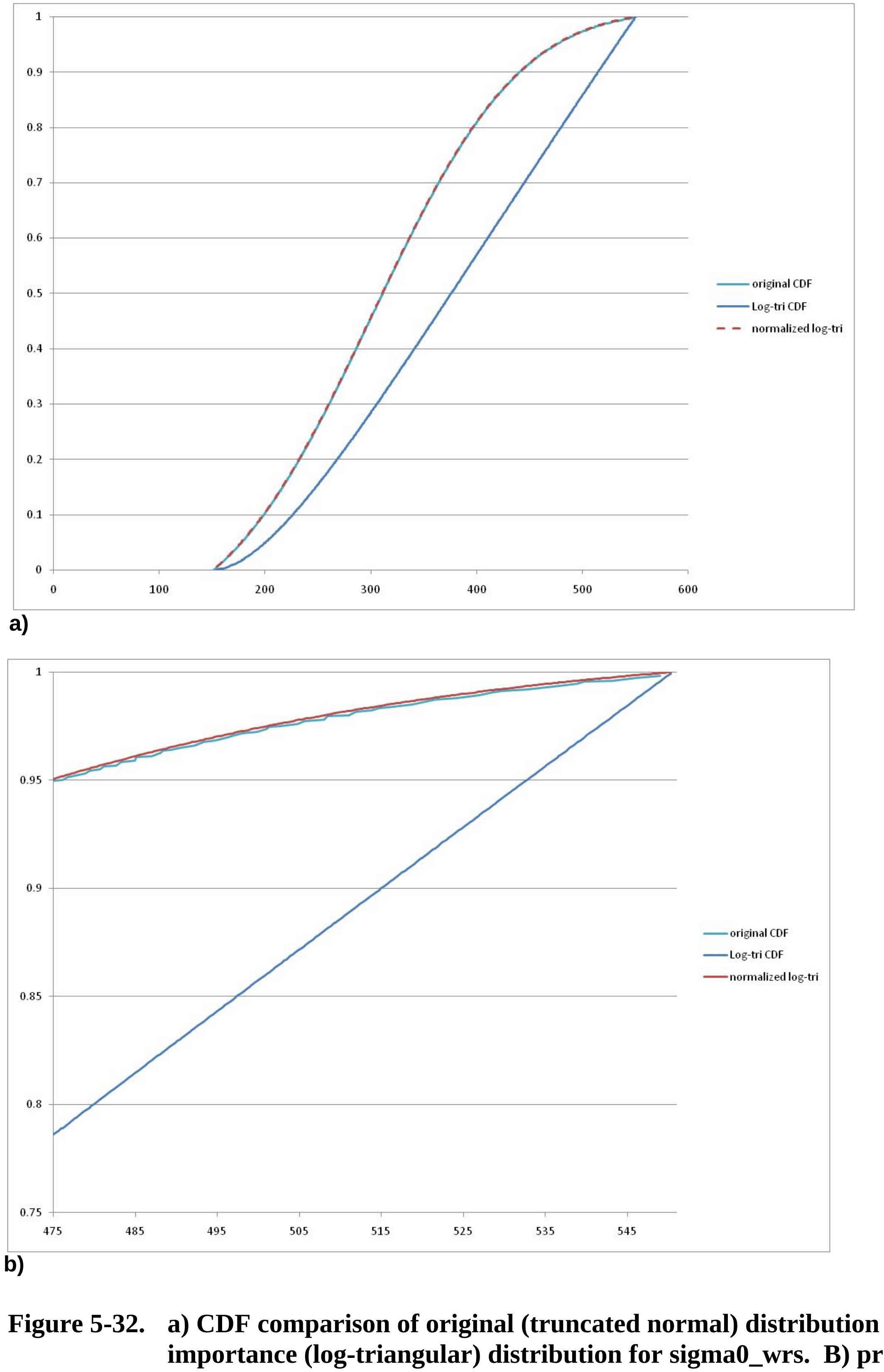


Figure 5-32. a) CDF comparison of original (truncated normal) distribution and importance (log-triangular) distribution for sigma0_wrs. B) presents zoom of upper tail.

A close look at the CDF (Figure 5-32) shows that the importance distribution favors the high end of the domain for sigma0_wrs. About 22% of the realizations using the log-triangular importance sampling will be used to estimate the last 5% of the original distribution. On the same graph, one can see that once the weight correction is applied, the original distribution is preserved. The result is that the high tail of the distribution is estimated more precisely using importance sampling (while the low tail is estimated not as precisely).

The same approach has been used on B1 with stronger emphasizes on low values. It was harder to define the log-triangular distribution as B1 is not truncated and can virtually go from negative infinity to positive infinity. As a first test, it was considered that the quantiles q = 5 x 10^{-4} and q = 0.999 were large enough to represent the domain of interest, as the region of interest was the low end of the distribution, the mode was taken close to the minimum at q = 5.1 x 10^{-4}. Figures 5-33 and 5-34 present the resulting PDF and CDF comparisons.

Figure 5-33 shows that the difference between the importance distribution and the original distribution.

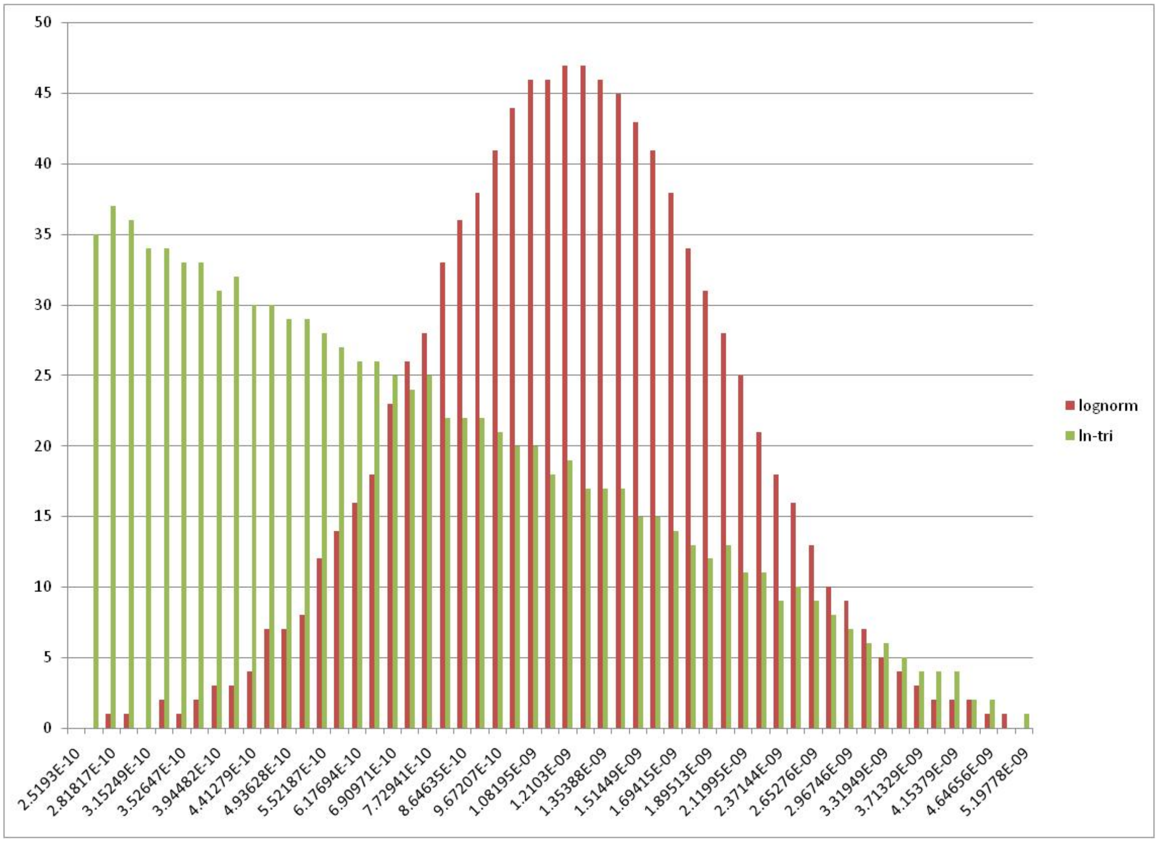


Figure 5-33. PDF comparison of original (normal) distribution and importance (log-triangular) distribution for B1.

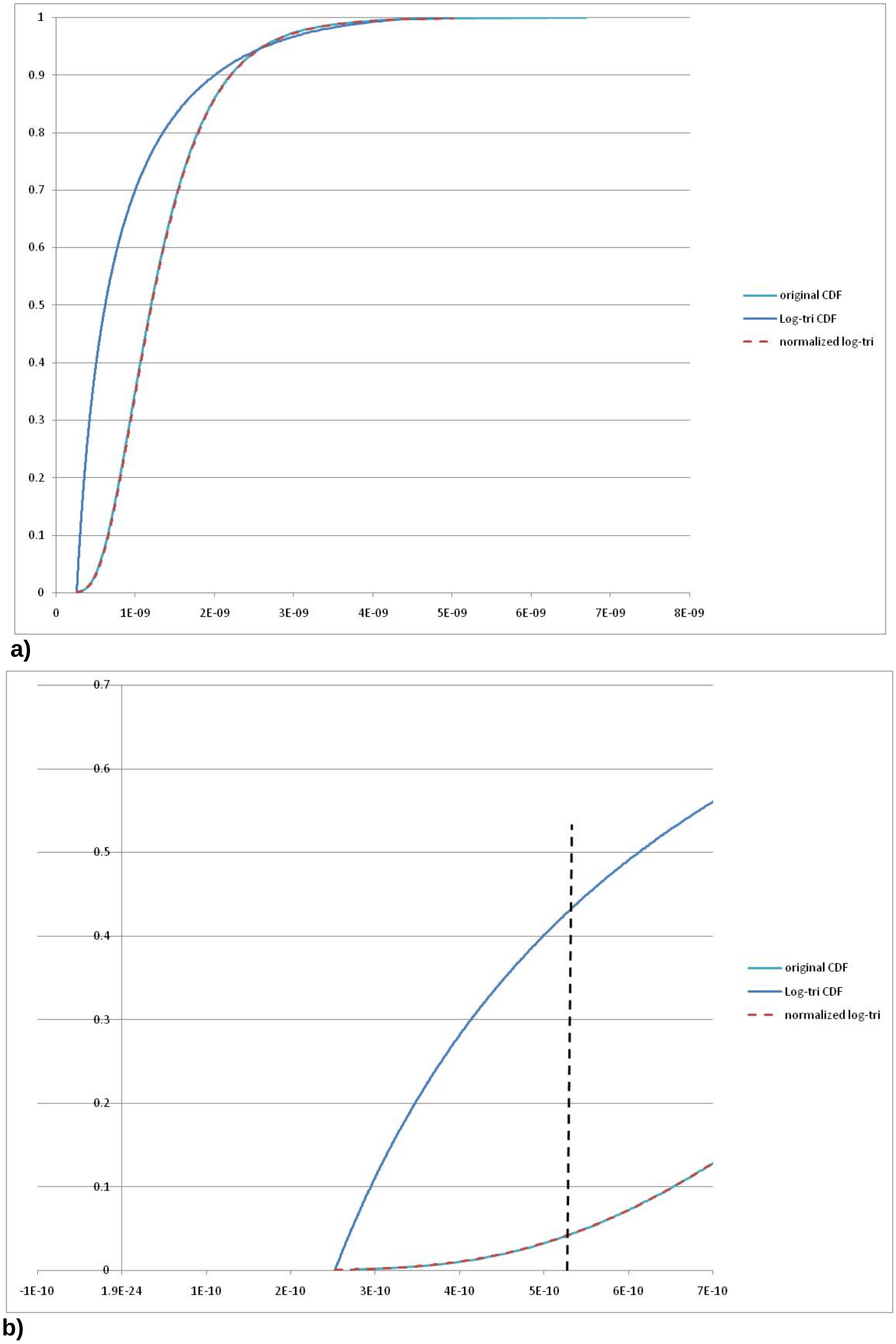


Figure 5-34. a) CDF comparison of original (normal) distribution and importance (log-triangular) distribution for B1, b) presents zoom of lower tail.

The CDF shows also a larger difference (Figure 5-34) as about 43% of the realizations are concentrated on the first 5% of the original distribution. Once again, a reconstruction of the original distribution from the samples from the importance sampling and using appropriate weights gives a good match to the original distribution.

In order to test the appropriateness of importance sampling coverage, scatterplots of realizations positions have been superimposed on one of the contour plots for probability of rupture after 50 yr (Figure 5-35). While more than 50% of the realization leads to no failure with the original sample, they are greatly reduced to about 20% of the realizations when importance sampling is used.

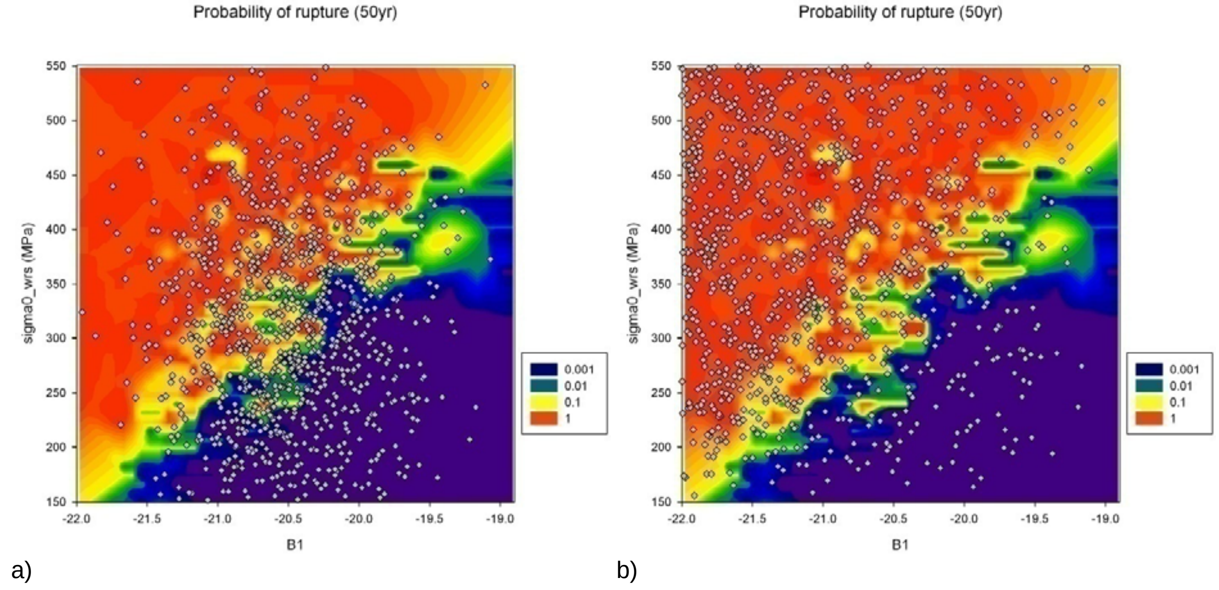


Figure 5-35. Comparison of original coverage (a) and importance sampling coverage (b) of the input domain superimposed on sensitivity analysis result over probability of rupture.

For the last parameter selected, *Xc,* the original distribution was truncated normal with a mean of 0.25, a standard deviation of 0.05 and a range between 0.1 and 0.5. A normal distribution covers more the area around the mean and less the area in the edges. A simple uniform distribution between 0.1 and 0.5 offers a better coverage over the edges without under sampling the center of the distribution.

The result is quite interesting (see Figure 5-36) as the high end of the distribution was completely uncovered in the original distribution. In such cases, it is possible that the importance sampling will focuses too much on regions of low probability and that it would have no affect the final estimate. However, since the uniformly distribution still insures a good coverage of the initial trace (about 50% of the realizations are within the range of 99% of the original distribution), a uniform distribution was still used.

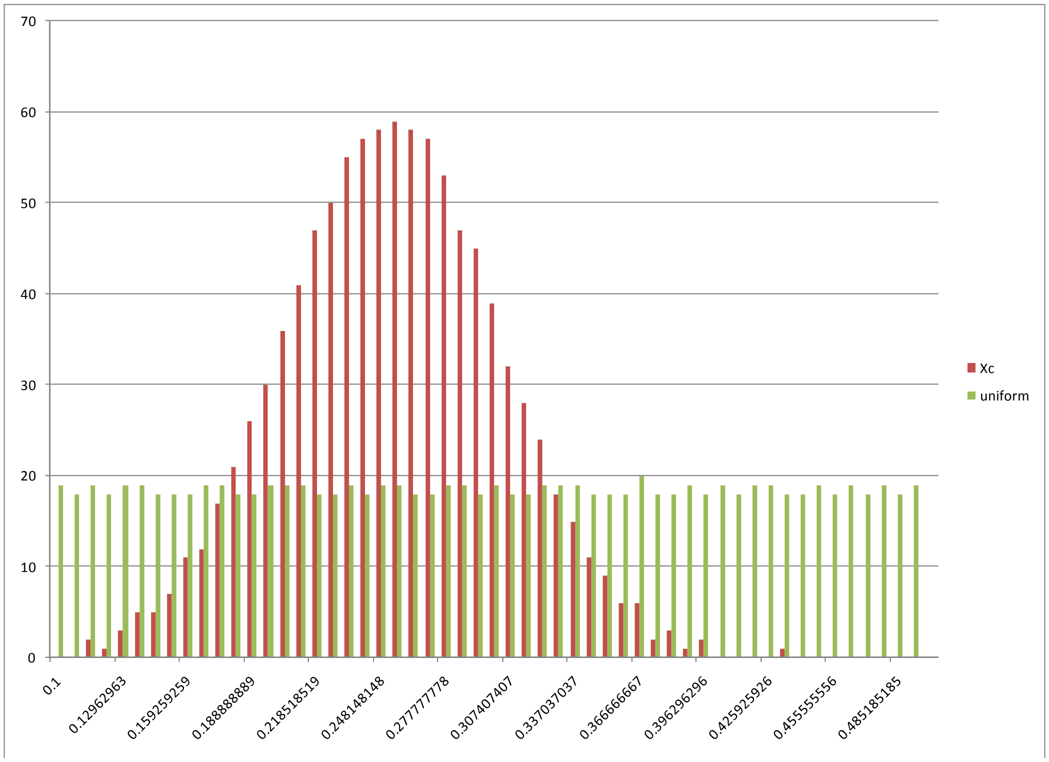


Figure 5-36. PDF comparison of original (normal) distribution and importance (uniform) distribution for Xc

The second set of tests was performed on the Safe End Case, for which both regular and importance sampling were performed. While the distribution for B1 was the same in the safe end case as the base case, both Sig0_WRS and Xc were changed for the safe end conditions. The distributions for Sig0_WRS, B1, and Xc were modified for the importance sampling case as described below.

The distribution for sigma0_wrs in the safe end case was truncated normal with a mean of -16.2 MPa a standard deviation of 117 MPa and a range of -300 MPa to 300 MPa. If rupture was possible for high values of sigma0_wrs, it was considered very unlikely to have any rupture below 150 MPa. So sampling on the whole range was considered as an inappropriate use of the importance sampling. In order to try to capture with more accuracy the effects of sigma0_wrs, it was decided to use a log-triangular distribution with minimum = 100, maximum = 300 and mode = 299. 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 of mean -16.2 and standard deviation 117, the value of 100 represents approximately the 84th percentile. Therefore only 16% of the distribution is sampled with the importance sampling (see Figure 5-37). In such situation, it is necessary to multiply any result by 0.16 in order to correct for this specific coverage and add to it the expected value that will be obtained for 84% of the non covered part of distribution, that is to say:

$$y_{final} = 0.16\ y_{calc} + 0.84\ y_{corr}$$

Where y_{final} represents the correct estimate, y_{calc} the calculation using importance sampling and y_{corr} the corrective term representing the value of y for the non-covered part of the distribution. In most of the output we are considering, n most of the output we are considering, y_{corr} will be equal to zero, simplifying the equation to a unique multiplication. Figures 5-37 shows a comparison of the PDF between original and importance distribution for sigma0_wrs.

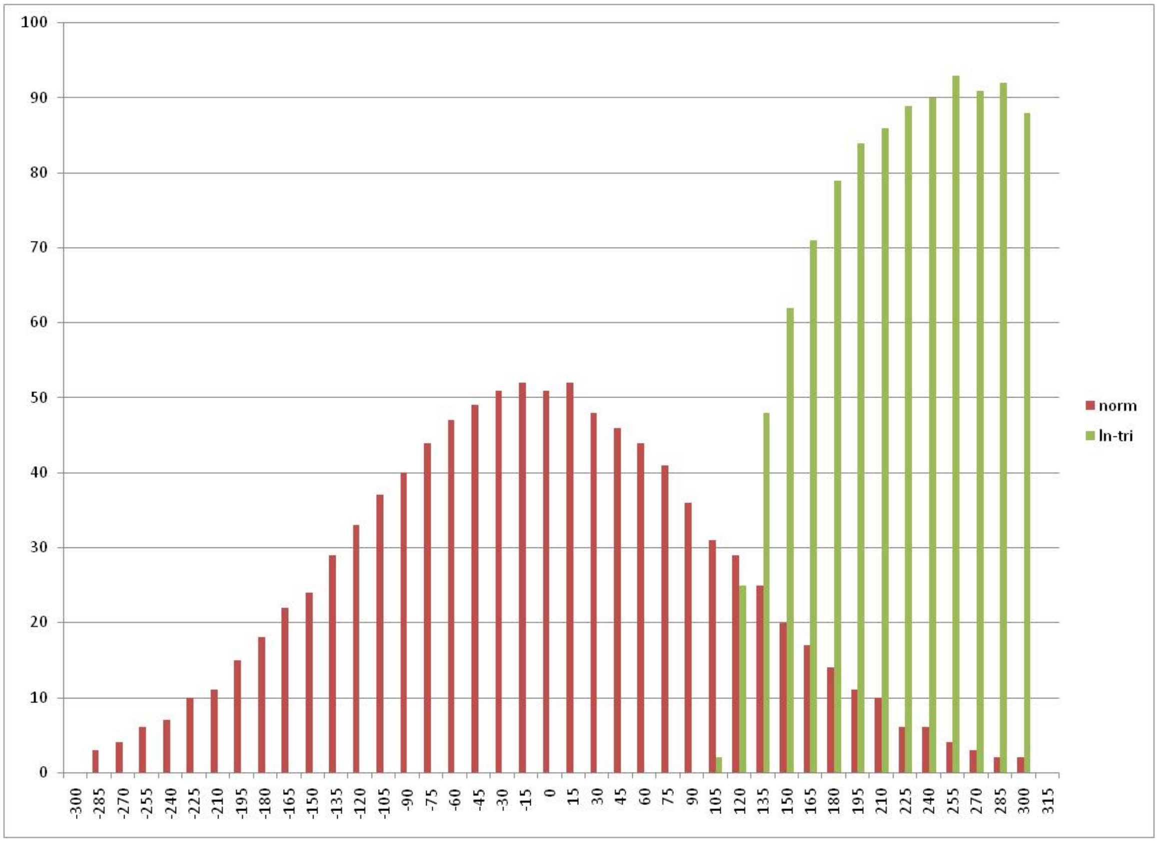


Figure 5-37. Comparison of PDF for sigma0_wrs using classical (red) and importance (green) distribution in the safe end case.

The distribution for B1 was centered even more to the left, using q = 5 x 10^{-6} and q = 5.1 x 10^{-6} respectively for minimum and mode. Figure 5-38 shows a comparison of the PDF between original and importance distribution for B1.

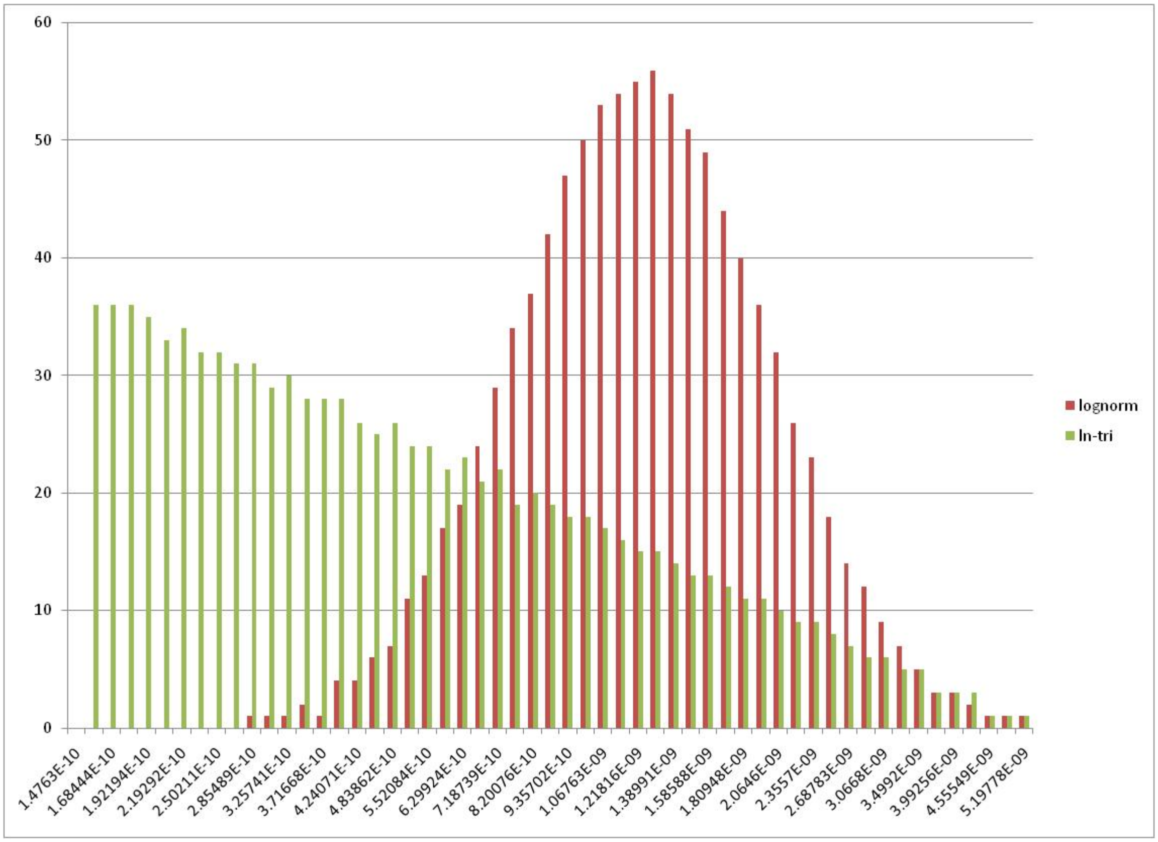


Figure 5-38. Comparison of PDF for B1 using classical (red) and importance (green) distribution in the safe end case.

For the last parameter selected, *Xc,* the original distribution was truncated normal with a mean of 0.18, a standard deviation of 0.036 and a range between 0.1 and 0.5. A normal distribution covers more the area around the mean and less the area in the edges. A simple uniform distribution between 0.1 and 0.5 offers a better coverage over the edges without under sampling the center of the distribution. Figure 5-39 shows a comparison of the PDF between original and importance distribution for *Xc*.

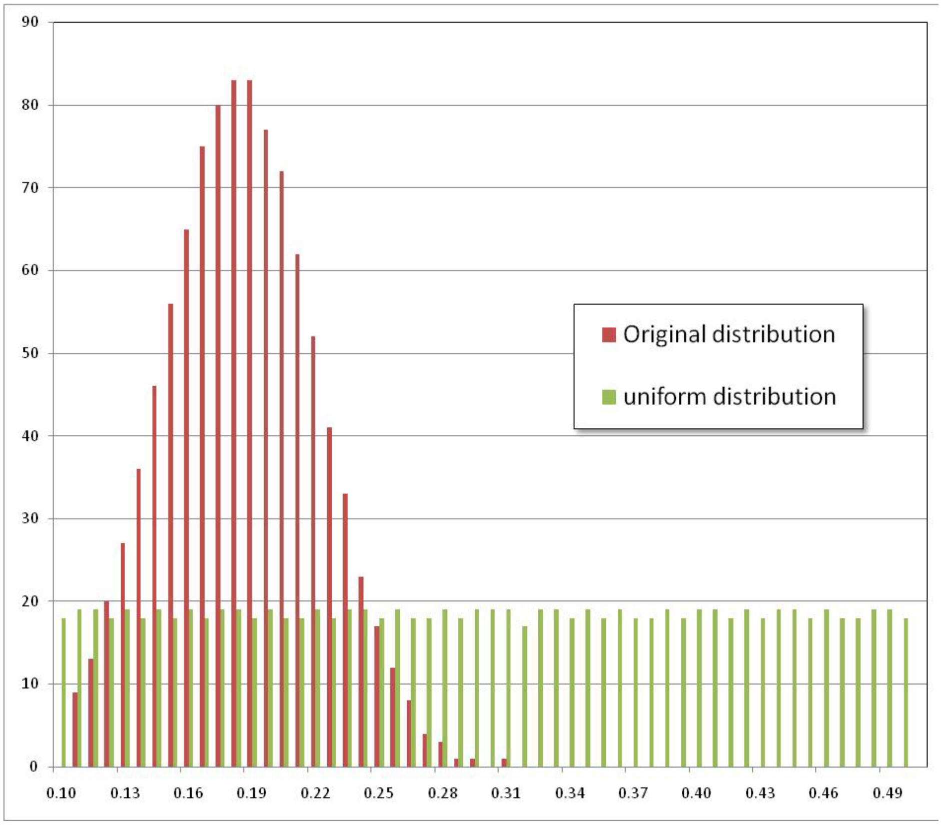


Figure 5-39. PDF comparison of original (normal) distribution and importance (uniform) distribution for Xc for the safe end case.

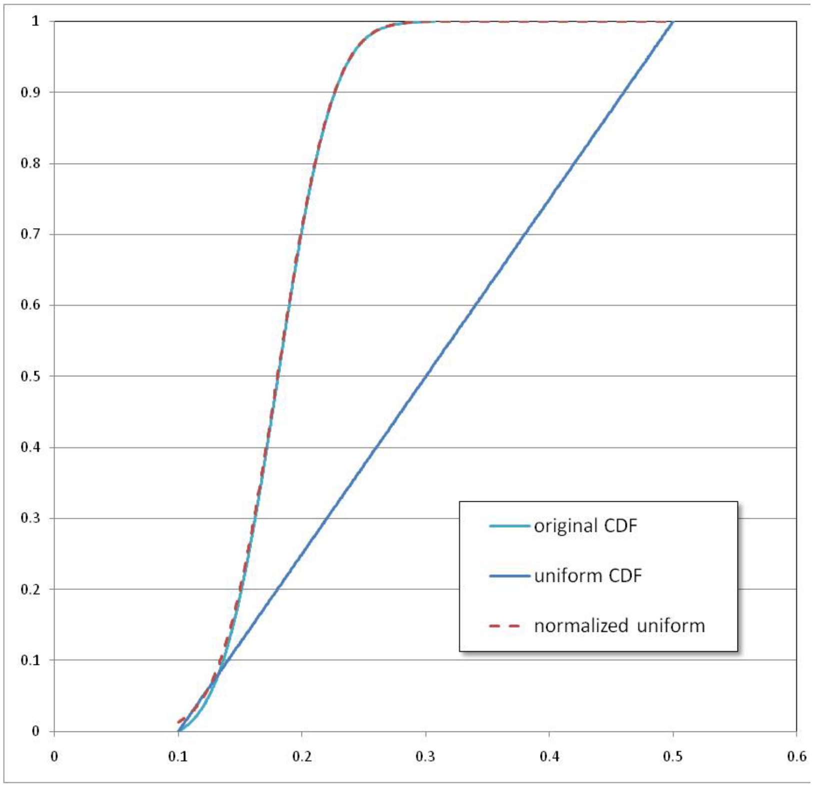


Figure 5-40. CDF comparison of original (normal) distribution for Xc with importance (uniform) distribution and renormalized distribution

For importance sampling analysis, the choice of the distribution used and to a lesser extent the parameters selected, are mainly based on the user experience. A bad selection may lead focus the analysis on the wrong area (either an area without interest, or an area of such low probability of occurrence that it won't affect the final result) which can make the importance sampling useless or worse in some unlucky conditions. An automatic selection of regions of interest is therefore an area of improvement that may be studied in the future.

5.3.6.3 Estimation of weight for each realization

The last part of the application of such type of importance sampling is to estimate the region of the input hyperspace covered by each realization (this region cannot be considered as equal for each realization). This correction is estimated by considering the area of coverage of each realization to each of the input parameter whose importance sampling is applied, multiplying the correction factor to consider all these parameters and normalizing the resulting weights.

Technically, for each input parameter having importance sampling, the corresponding quantile of the input value in the original space is computed. This quantile is compared with all other quantiles in order to estimate the weight for each realization. This operation is repeated for each importance sampled variables. All the weights associated to a single realization are multiplied together to know which area of the input hyperspace is covered by the realization. A normalization of all weights (i.e. summing all the weights and dividing each of them by this sum) leads to the final weight associated with each realization.

5.3.6.1 Importance Sampling Results

Figure 5-41 shows a comparison of original sampling technique and importance sampling of size 10,000 for the base case probability of rupture with and without inspection and/or leak rate detection. While the base case probability estimate is identical (which was expected as the original LHS was considered big enough), we can see some variation when inspection and leak rate detection are added. The most notable differences are seen when both inspection and leak rate detection are considered. The difference is important enough to lead to the conclusion that use of importance sampling (or larger sample size) is necessary in order to estimate accurately such probabilities.

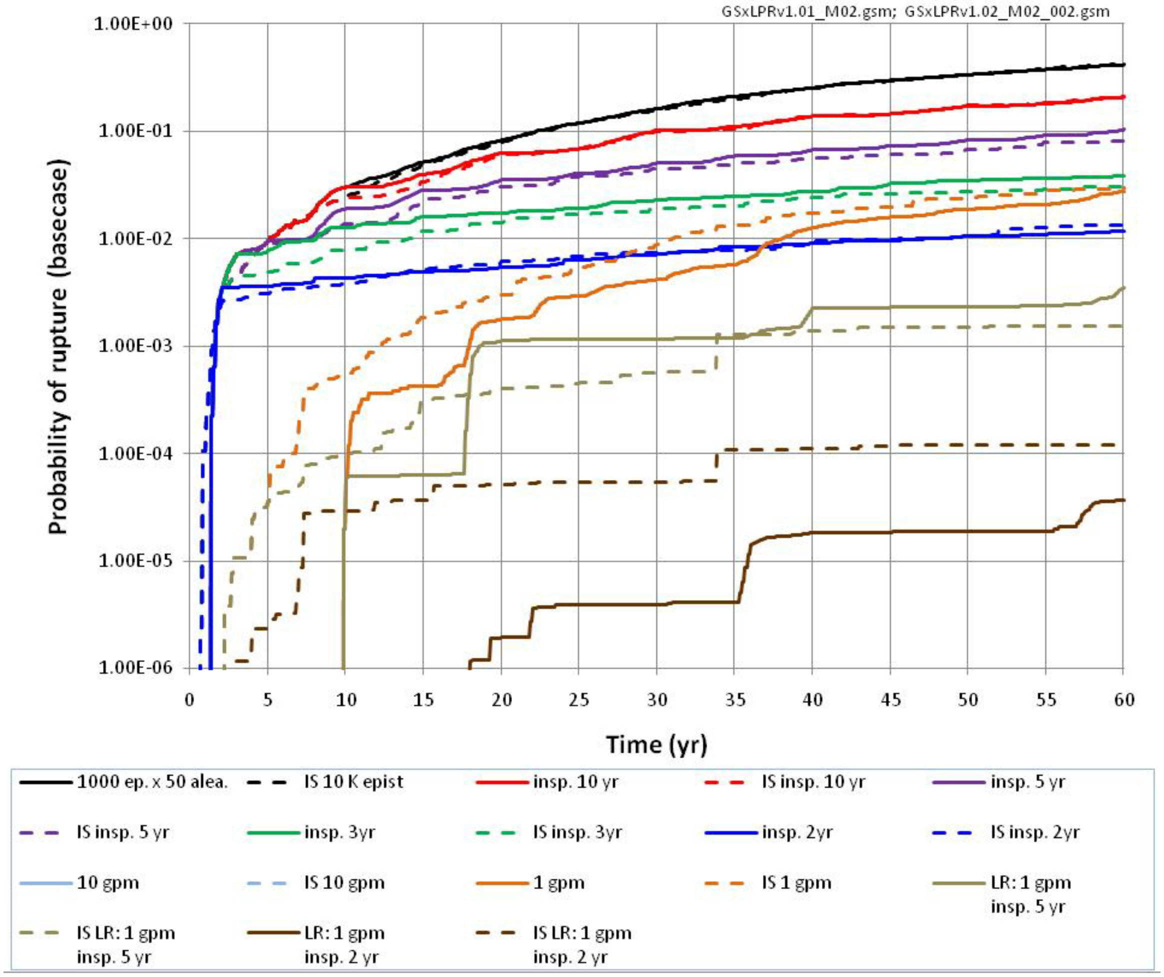


Figure 5-41. Comparison of the mean probability of rupture using traditional and importance sampling methods for the Base Case analysis.

The second importance sampling analysis on the Safe End case was also performed on the same three selected input variables using DPD. Figure 5-42 shows the comparison of the different methods used. One can clearly see the advantage of importance sampling (whether using DPD or LHS approach) to correctly estimate the probability of rupture; especially at early times (before 30 yr). While the importance sampling leads to a lot more ruptures (about 40% of the runs compared to 10% for the DPD importance sampling and 0.6% for the traditional case), the two importance sampling approaches give similar results.

Results when leak rate detection (of 1 gpm) and inspection (every 10 yrs) are a little better than with the original distribution (that failed to find a single rupture case), but the two estimates (in blue) are different and it is hard to determine which of the two results is the most accurate (it is also possible that both are equally inaccurate). Section 4.3.2 discusses about how to estimate stability of the two approaches using bigger samples size or replicates. In such situation, it would be tempting to use the highest estimate as a conservative solution, but such method could lead to inappropriate conclusions. One important note is that despite the variation, the mean results stay within less than one order of magnitude from each other at late times, which is not too bad considering the order of probability estimated (close to 10^{-6}) and the original sample size used (10^{4}) that would have likely led to no realization with failure without using importance sampling.

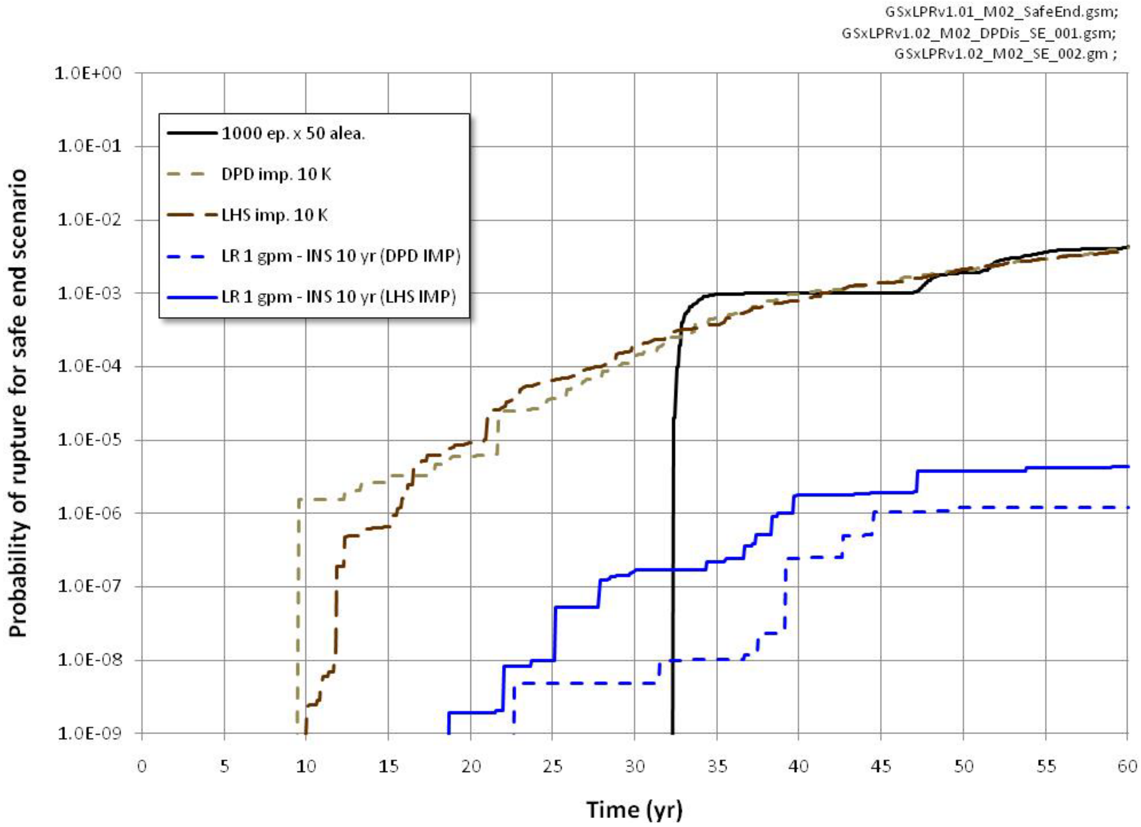


Figure 5-42. Comparison of classical method and two importance sampling types for safe end calculation.

5.3.7 Uncertainty Classification Sensitivity

The uncertainty analysis presented in Section 5.2.1 shows a very little influence of the aleatory uncertainty on the results. Notably, the time of occurrence of a crack in the future is 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.

The uncertainty classification has been performed by subject matter experts in the xLPR Models and Input Task Groups. However, the times of occurrence of events, like the initiation of a crack, are often considered as random in similar analyses. With this in mind, a sensitivity case was conducted after changing the classification of the parameters controlling time and location of when cracks initiate from epistemic to aleatory. The sensitivity analysis was conducted to give insights on the effects of uncertainty in crack initiation and the classification of the uncertainty (as aleatory or epistemic) on the uncertainty in the response.

In this sensitivity case, the crack initiation uncertain input parameters B1, BWH_stdev, RandU3 and RandULoc were reclassified as aleatory uncertainty, without changing their distributions. The run performed was otherwise the same as the base case with an epistemic sample size of 1000 using LHS and an aleatory sample size of 50 using simple random sampling and a time-step representing 2 months for a total of 361 time-steps.

The first output compared was the probability of first crack occurrence. In the base case, there is no uncertainty in the time of occurrence of first crack, for a given an 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 in Figure 5-43a. With reclassification of the crack initiation parameter to aleatory the time of first crack occurrence is spanned over time (Figure 5-43b), leading to a smoother probability estimate over the 60 yr timeframe. The probabilities of first leakage (Figure 5-44) and rupture (Figure 5-45) show similar changes with the application of new uncertainty classification. An interesting consequence is that quantile curves are now completely different. As the time of crack occurrence is not fixed for each epistemic set, it is more likely to have at least one crack for each epistemic realization (but a smaller change that ALL realizations within an epistemic set leads to rupture). The interpretation of the quantiles curves changes considerably from one assumption to the other. In the second case, for instance, a median of 0.4 at 60 yr means that half of the epistemic realizations have a 40% chance of seeing a crack in the future, while in the base case, there was absolutely no chance of rupture for half of the epistemic realizations.

The results presented in Figure 5-46 show that the change of classification of the initiation parameter from epistemic to aleatory induces many more epistemic realizations with at least one crack, consequently with fractional surface area cracked. The median and 0.05 quantile are displayed indicating that at least 95% of the epistemic realizations lead to at least one crack occurring. Moreover, the aleatory uncertainty plays an important role in this characterization, as shown by the range covered by each CCDF individually. In the base case, most of the CCDFs are close to a vertical line indicating little effect of the aleatory uncertainty on the results. A comparison of CCDFs on Total Leak Rate (Figure 5-47) leads to the same conclusion.

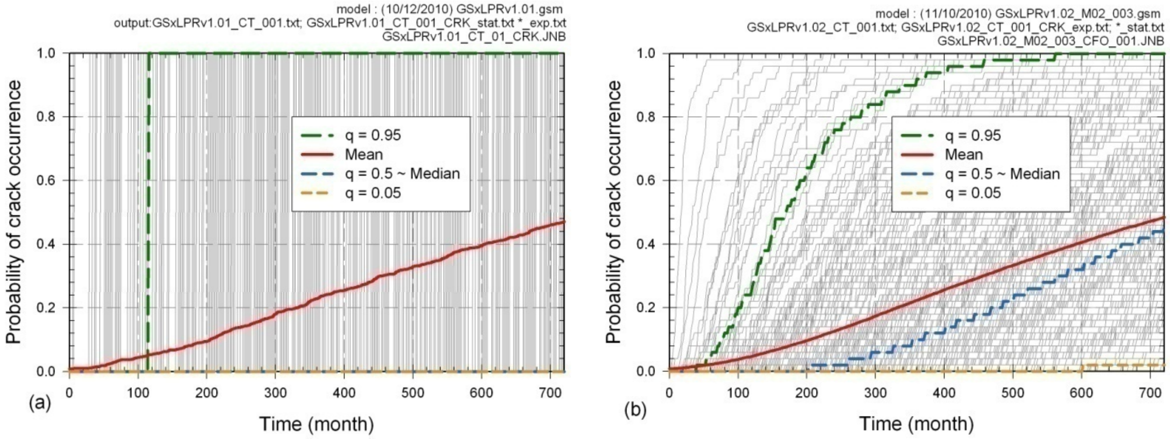


Figure 5-43. Probability of first crack for base case (a) and with crack initiation uncertainty changed from epistemic to aleatory (b)

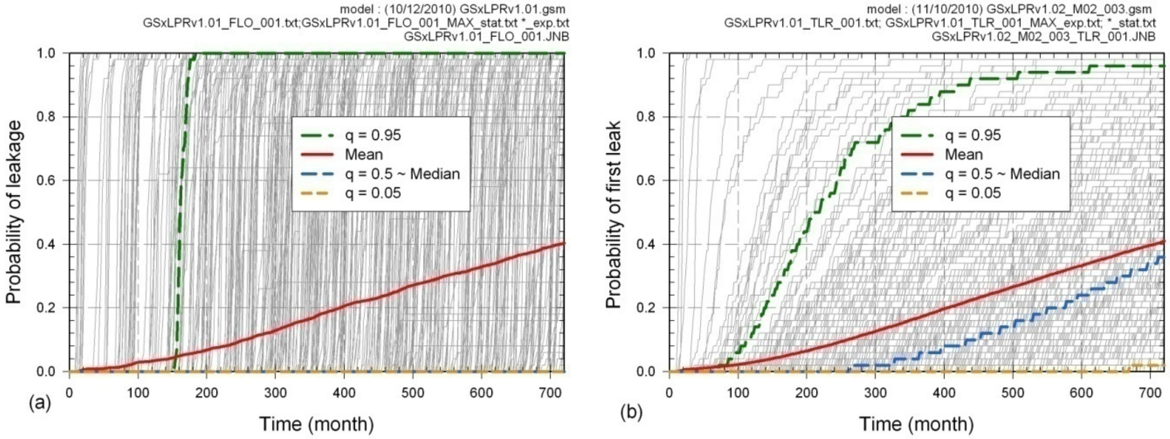


Figure 5-44. Probability of first leakage for base case (a) and with crack initiation uncertainty changed from epistemic to aleatory (b)

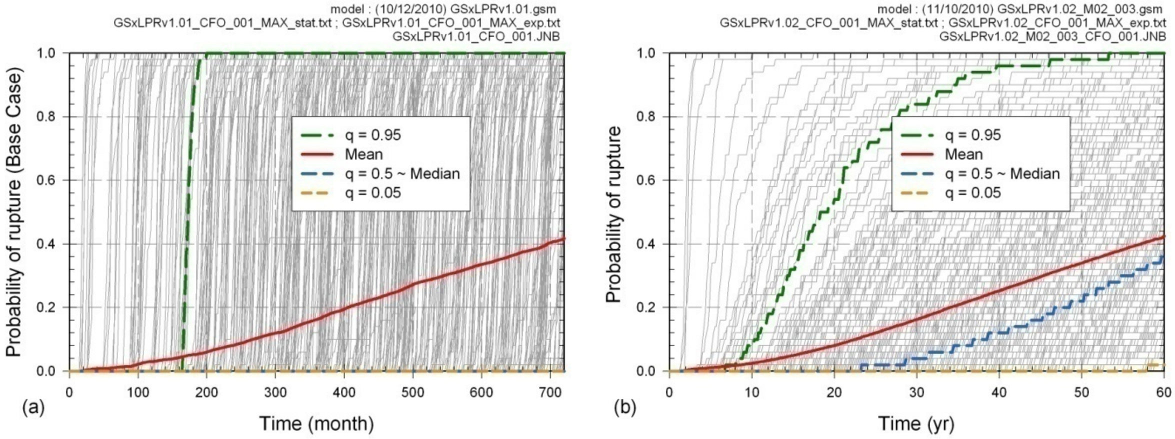


Figure 5-45. Probability of rupture for base case (a) and with crack initiation uncertainty changed from epistemic to aleatory (b)

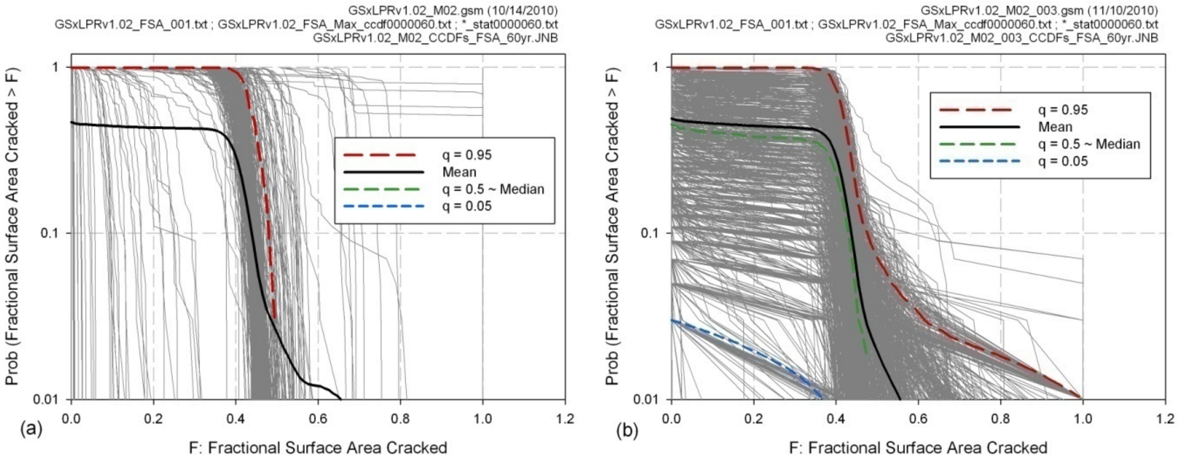


Figure 5-46. Distribution of CCDFs of fractional surface area cracked at 60 years for base case (a) and with crack initiation uncertainty changed (b)

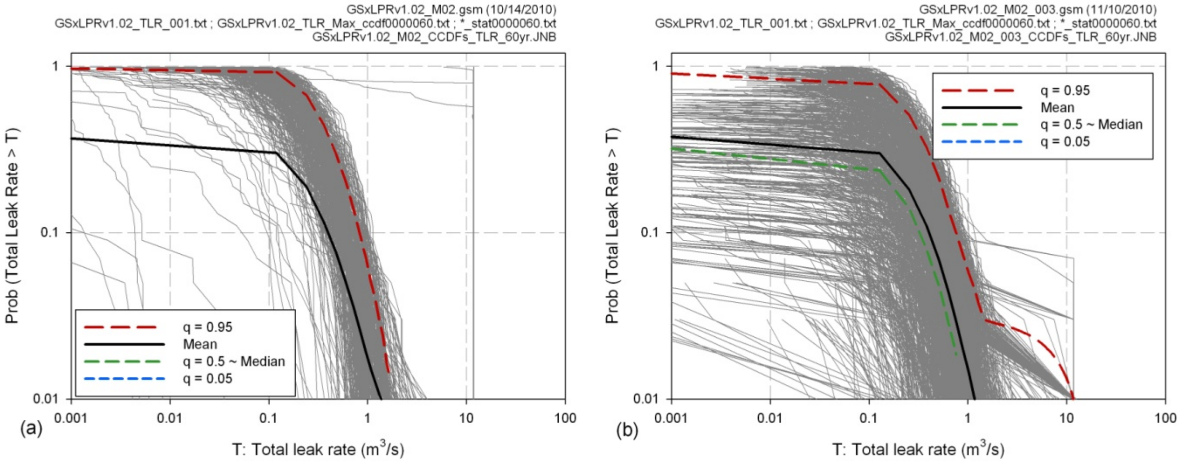


Figure 5-47. Distribution of CCDFs of total leak rate at 60 years for base case (a) and with crack initiation uncertainty changed (b)

A comparison of mean CCDFs of probability of rupture between the base case and the sensitivity case with crack initiation uncertainty changed to aleatory (Figure 5-48) shows a major difference brought from the re-classification of the crack initiation uncertain parameters. Finally the comparison of mean (i.e. expected over epistemic uncertainty) probabilities of first crack, first leak and rupture for both base case analysis and new uncertainty classification displayed in Figure 5-49 shows that the change in classification does not significantly affect the mean results.

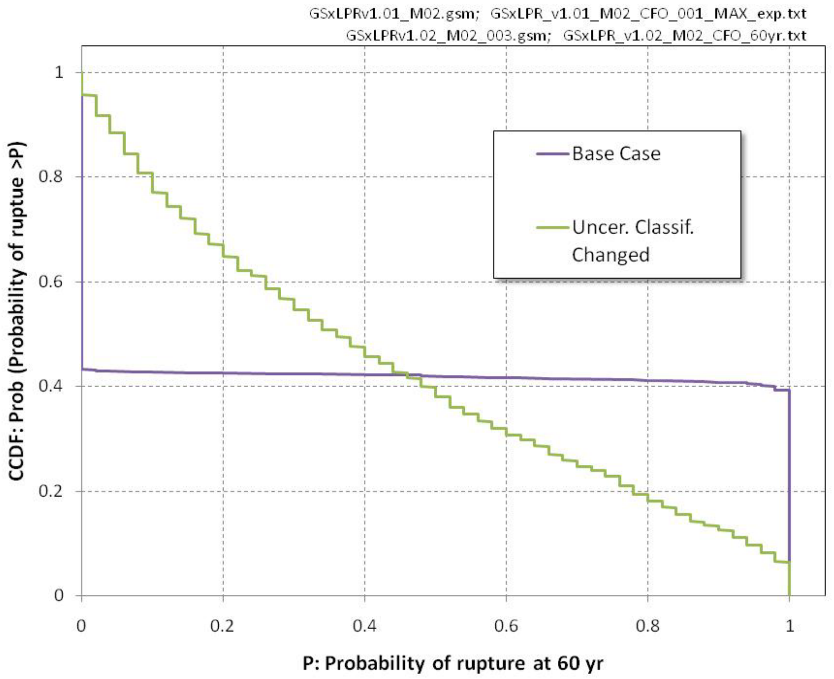


Figure 5-48. CCDF of probability of rupture at 60 yr for base case (purple) and with crack initiation uncertainty changed from epistemic to aleatory (green)

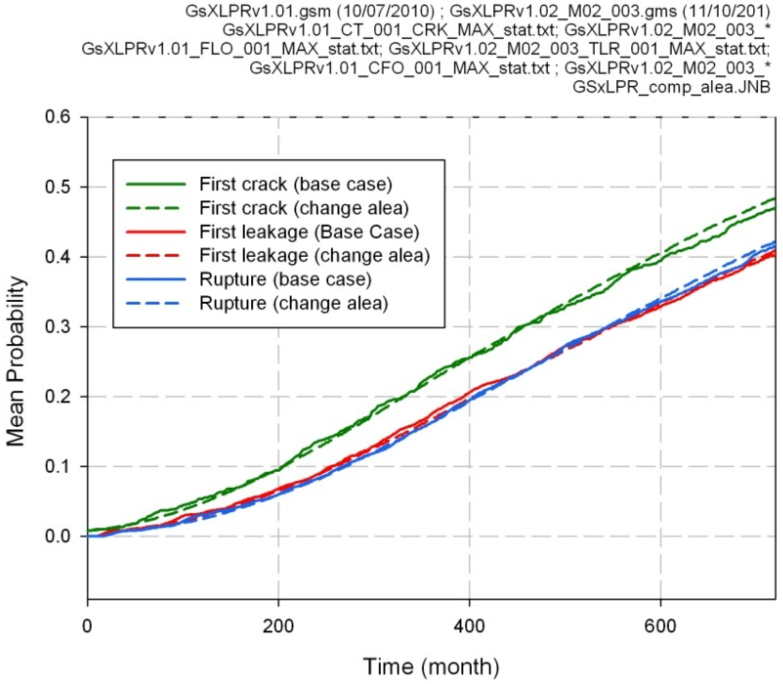
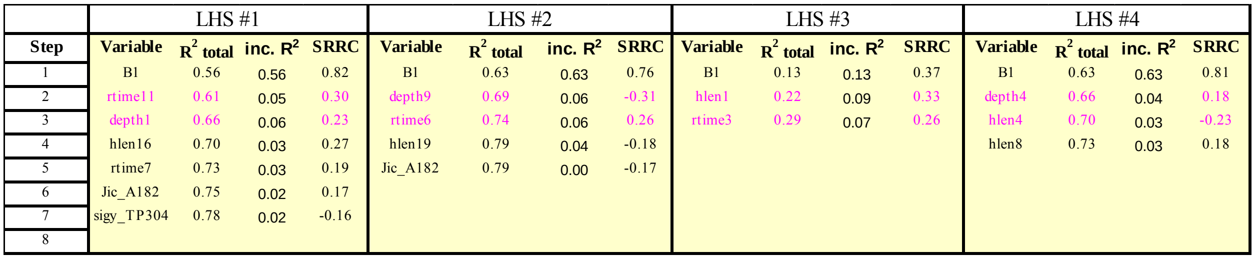


Figure 5-49. Comparison of mean probabilities of first crack (green) first leakage (red) and rupture (blue) between the base case analysis and with crack initiation uncertainty changed from epistemic to aleatory

A stepwise regression has been performed over aleatory uncertainty on the first four epistemic sets for rupture time (Table 5-9) and fractional surface area cracked (Table 5-10) for the sensitivity case with the crack initiation uncertainty changed to aleatory. The step wise regression over rupture time shows that there is not a clear influence on the time of rupture except for the B1 parameter. B1 is the most important parameter, often explaining up to 60% of the aleatory variability. All other parameters are not consistent and may be spurious relationships.

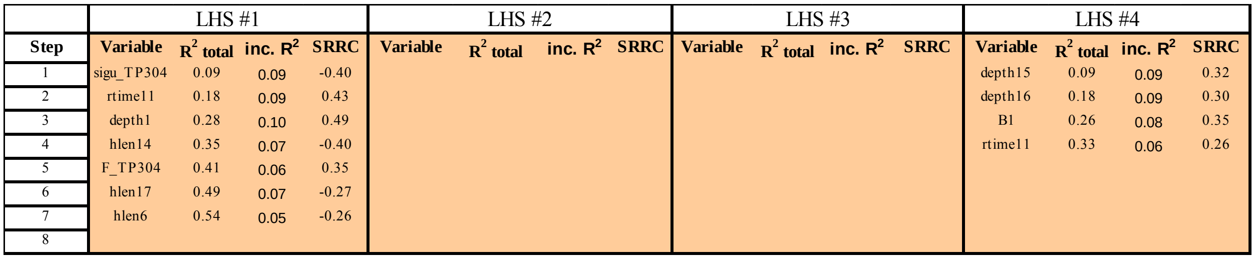
The stepwise regression over variability on fractional surface area cracked does not give consistent results (and gives no results at all for LHS#2 and LHS#3) indicating that no parameter has a monotonic influence on this variability. This result is identical to the one we expect to see on the base case, as the CCDFs have shown that the aleatory uncertainty did not play any role in the overall uncertainty of the output considered.

Table 5-9. Stepwise regression over aleatory uncertainty for pipe rupture time at 60 years for first 4 epistemic sets.



	LHS #1				LHS #2				LHS #3				LHS #4			
Step	**Variable**	**R^2 total**	**inc. R^2**	**SRRC**	**Variable**	**R^2 total**	**inc. R^2**	**SRRC**	**Variable**	**R^2 total**	**inc. R^2**	**SRRC**	**Variable**	**R^2 total**	**inc. R^2**	**SRRC**
1	B1	0.56	0.56	0.82	B1	0.63	0.63	0.76	B1	0.13	0.13	0.37	B1	0.63	0.63	0.81
2	rtime11	0.61	0.05	0.30	depth9	0.69	0.06	-0.31	hlen1	0.22	0.09	0.33	depth4	0.66	0.04	0.18
3	depth1	0.66	0.06	0.23	rtime6	0.74	0.06	0.26	rtime3	0.29	0.07	0.26	hlen4	0.70	0.03	-0.23
4	hlen16	0.70	0.03	0.27	hlen19	0.79	0.04	-0.18					hlen8	0.73	0.03	0.18
5	rtime7	0.73	0.03	0.19	Jic_A182	0.79	0.00	-0.17								
6	Jic_A182	0.75	0.02	0.17												
7	sigy_TP304	0.78	0.02	-0.16												
8																

Table 5-10. Stepwise regression over aleatory uncertainty for fractional surface area cracked at 60 years for first 4 epistemic sets.



	LHS #1				LHS #2				LHS #3				LHS #4			
Step	**Variable**	**R^2 total**	**inc. R^2**	**SRRC**	**Variable**	**R^2 total**	**inc. R^2**	**SRRC**	**Variable**	**R^2 total**	**inc. R^2**	**SRRC**	**Variable**	**R^2 total**	**inc. R^2**	**SRRC**
1	sigu_TP304	0.09	0.09	-0.40									depth15	0.09	0.09	0.32
2	rtime11	0.18	0.09	0.43									depth16	0.18	0.09	0.30
3	depth1	0.28	0.10	0.49									B1	0.26	0.08	0.35
4	hlen14	0.35	0.07	-0.40									rtime11	0.33	0.06	0.26
5	F_TP304	0.41	0.06	0.35												
6	hlen17	0.49	0.07	-0.27												
7	hlen6	0.54	0.05	-0.26												
8																

This sensitivity analysis demonstrates the effects of uncertainty classification on the uncertainty analysis and why the initial classification is important.

5.3.8 Summary Discussion of Uncertainty in the xLPR Model Results

An analysis outcome of fundamental importance in the xLPR project is the occurrence of pipe rupture. Three important questions associated with pipe rupture are: Q1, What controls the occurrence of pipe rupture?; Q2, What is the probability of pipe rupture?; and Q3, What is the uncertainty in an estimate of the probability of pipe rupture?.

The discussion of the answers to the three indicated questions is facilitated by the introduction of the following conceptual model for pipe rupture:

$$f\left(t \mid \mathbf{a}, \mathbf{e}_M\right) = \begin{cases} 1 \text{ if pipe rupture occurs before time } t \\ 0 \text{ otherwise,} \end{cases} \qquad \text{Eq.5-1}$$

where $\mathbf{a}$ is a vector of random quantities affecting pipe rupture (i.e., aleatory variables) and $\mathbf{e}_M$ is a vector of possible values for quantities affecting pipe rupture that have fixed but imprecisely known values (i.e., epistemic variables). In general, $f(t| \mathbf{a}, \mathbf{e}_M)$ would be a very complex mathematical model but the details of the mathematical structure of $f(t| \mathbf{a}, \mathbf{e}_M)$ are not needed for this discussion.

The vector $\mathbf{a}$ will have the form

$$\mathbf{a} = \left[n, t_1, \mathbf{p}_1, t_2, \mathbf{p}_2, \ldots, t_n, \mathbf{p}_n\right], \qquad \text{Eq. 5-2}$$

where n is the number of cracks that initiate in the time interval [0, t], t_i is the time at which crack i initiates with $0 \le t_1 \le t_2 \le \cdots \le t_n \le t$, and $\mathbf{p}_i$ is a vector of properties associated with crack i. For example, $\mathbf{p}_i$ might be of the form

$$\mathbf{p}_i = \left[l_i, d_i, s_i, m_i\right], \qquad \text{Eq. 5-3}$$

where l_i is the location of crack i, d_i is the initial depth of crack i, s_i is a measure of stress at the location of crack i, and m_i is a relevant material property at the location of crack i. The core idea is that the properties associated with each vector $\mathbf{p}_i$ vary randomly from crack to crack. In general, the vectors $\mathbf{p}_i$ could be more or less complex than indicated in Eq. (5-3). It is also possible that the properties of $\mathbf{p}_i$ could be correlated with the properties of $\mathbf{p}_j$ for j = 1, 2, …, i – 1. The set of possible values for $\mathbf{a}$ can be represented by a set $\mathcal{A}$, and the distribution of $\mathbf{a}$ can be represented by a density function $d_A(\mathbf{a})$ defined on $\mathcal{A}$.

The density function $d_A(\mathbf{a})$ derives from the distributions for the individual elements of $\mathbf{a}$. It is likely that many of these distributions will be poorly known, with the result that $d_A(\mathbf{a})$ actually has the form $d_A(\mathbf{a}|\mathbf{e}_A)$, where

$$\mathbf{e}_A = \left[e_{A1}, e_{A2}, \ldots, e_{A,nEA}\right] \qquad \text{Eq. 5-4}$$

is a vector imprecisely known (i.e., epistemic) quantities used in the definition of the distributions that characterize aleatory variability in the elements of $\mathbf{a}$ and the notation "$|\mathbf{e}_A$" is used to indicate the density function $d_A(\mathbf{a}|\mathbf{e}_A)$ is conditional on the epistemically uncertain quantities that constitute the vector $\mathbf{e}_A$. The set of possible values for $\mathbf{e}_A$ can be represented by a set $\mathcal{EA}$, and the distribution of $\mathbf{e}_A$ can be represented by a density function $d_{EA}(\mathbf{e}_A)$ defined on $\mathcal{EA}$. In general, $d_{EA}(\mathbf{e}_A)$ would derive from distributions defined to characterize the epistemic uncertainty associated with individual elements of $\mathbf{e}_A$.

The vector $\mathbf{e}_M$ has the form

$$\mathbf{e}_M = \left[e_{M1}, e_{M2}, ..., e_{M,nEM}\right], \tag*{Eq. 5-5}$$

where each element is an epistemically uncertain quantity required in the evaluation of $f(t|\ \mathbf{a}, \mathbf{e}_M)$. The set of possible values for $\mathbf{e}_M$ can be represented by a set $\mathcal{EM}$, and the distribution of $\mathbf{e}_M$ can be represented by a density function $d_{EM}(\mathbf{e}_M)$ defined on $\mathcal{EM}$. In general, $d_{EM}(\mathbf{e}_M)$ would derive from distributions defined to characterize the epistemic uncertainty associated with individual elements of $\mathbf{e}_M$.

The three indicated questions are now considered.

Q1, What controls the occurrence of pipe rupture? The issue addressed in this question is what determines whether the function $f(t|\ \mathbf{a}, \mathbf{e}_M)$ indicated in Eq. (5-1) has a value of 1 (i.e., pipe rupture before time t) or a value of 0 (i.e., no pipe rupture before time t). This question can be addressed at three different levels.

At the first level, the value for $\mathbf{e}_M$ is fixed and the possible values for $f(t|\ \mathbf{a}, \mathbf{e}_M)$ are analyzed conditional on this fixed value for $\mathbf{e}_M$. Most likely, this analysis would involve generating a sample

$$\mathbf{a}_j = \left[n_j, t_{1j}, \mathbf{p}_{1j}, t_{2j}, \mathbf{p}_{2j}, \cdots, t_{nj}, \mathbf{p}_{nj}\right], j = 1, 2, ..., m, \tag*{Eq. 5-6}$$

from $\mathcal{A}$ in consistency with the density function $d_A(\mathbf{a}|\mathbf{e}_A)$ and a corresponding mapping

$$\left[\mathbf{a}_j, f\left(t\,|\,\mathbf{a}_j, \mathbf{e}_M\right)\right], j = 1, 2, ..., m, \tag*{Eq. 5-7}$$

between the sampled values $\mathbf{a}_j$ and the resultant values $f(t|\ \mathbf{a}_j, \mathbf{e}_M)$. In turn this mapping would be analyzed to determine how the individual elements of $\mathbf{a}$ affect the value of $f(t|\ \mathbf{a}, \mathbf{e}_M)$. If a single element of $\mathbf{a}$ determines the value of $f(t|\ \mathbf{a}, \mathbf{e}_M)$, this analysis could be as simple as examining scatterplots to identify this influential element. As $\mathbf{a}$ and the influence of its elements become more complex, the necessary analysis also becomes more complex and has to be designed to appropriately match the properties of $\mathbf{a}$ and $f(t|\ \mathbf{a}, \mathbf{e}_M)$. For example, it might be determined that occurrence of a rupture is dominated by whether or not n is greater than zero. In this case, additional insights might be obtained by carrying out the analysis conditional on the assumption that n is greater than zero. In general, there is no universal method for analyzing complex relationships between the properties of $\mathbf{a}$ and $f(t|\ \mathbf{a}, \mathbf{e}_M)$. However, the analysis of binary data (i.e., 0, 1 data) has been widely studied and many techniques exist for the analysis of such data that have the potential to be applied to analyses of pipe rupture [42]. It is likely that analyses of complex relationships between $\mathbf{a}$ and $f(t|\ \mathbf{a}, \mathbf{e}_M)$ will involve some form of regression analysis.

To obtain appropriate insights with respect to the relationships between the properties of $\mathbf{a}$ and $f(t|\ \mathbf{a}, \mathbf{e}_M)$, the indicated analyses should be performed conditional on several different values for $\mathbf{e}_M$.

At the second level, the value for $\mathbf{a}$ is fixed and the possible values for $f(t|\ \mathbf{a}, \mathbf{e}_M)$ are analyzed conditional on this fixed value for $\mathbf{a}$. Conceptually, this analysis would be the same as the preceding analysis except that $\mathbf{e}_M$ is varied rather than $\mathbf{a}$. Most likely, this analysis would involve generating a sample

$$\mathbf{e}_{Mj} = \left[e_{M1j}, e_{M1j}, ..., e_{M1j} \right], j = 1, 2, ..., m, \qquad \text{Eq. 5-8}$$

from $\mathcal{EM}$ in consistency with the density function $d_{EM}(\mathbf{e}_M)$ and a corresponding mapping

$$\left[\mathbf{e}_{Mj}, f\left(t \mid \mathbf{a}, \mathbf{e}_{Mj} \right) \right], j = 1, 2, ..., m, \qquad \text{Eq. 5-9}$$

between the sampled values $\mathbf{e}_{mj}$ and the resultant values $f(t|\ \mathbf{a}, \mathbf{e}_{Mj})$. The indicated mapping can then be explored with procedures for the analysis of binary data to determine how the individual elements of $\mathbf{e}_M$ affect the value of $f(t|\ \mathbf{a}, \mathbf{e}_M)$. This exploration will probably be easier than the exploration of the effects of the elements of $\mathbf{a}$ on the value of $f(t|\ \mathbf{a}, \mathbf{e}_M)$ when $\mathbf{e}_M$ is fixed. This statement is made because, in general, the set $\mathcal{EM}$ of possible values for $\mathbf{e}_M$ will have less internal uncertainty structure than the set $\mathcal{A}$ of possible values for $\mathbf{a}$. In particular, the elements of $\mathcal{A}$ correspond to time series while the elements of $\mathcal{EM}$ do not have an equivalent temporal ordering.

To obtain appropriate insights with respect to the relationships between the properties of $\mathbf{e}_M$ and $f(t|\ \mathbf{a}, \mathbf{e}_M)$, the indicated analyses should be performed conditional on several different values for $\mathbf{a}$.

At the third level, the values for $\mathbf{e}$ and $\mathbf{a}$ would be simultaneously varied. This would involve generating a sample

$$\mathbf{e}_j = \left[\mathbf{e}_{Aj}, \mathbf{e}_{Mj} \right], j = 1, 2, ..., nE, \qquad \text{Eq. 5-10}$$

from $\mathcal{EA} \times \mathcal{EM}$ in consistency with the density functions $d_{EA}(\mathbf{e}_A)$ and $d_{EM}(\mathbf{e}_M)$ and samples

$$\mathbf{a}_{jk} = \left[n_{jk}, t_{1jk}, \mathbf{p}_{1jk}, t_{2jk}, \mathbf{p}_{2jk}, \cdots, t_{njk}, \mathbf{p}_{njk} \right], k = 1, 2, ..., m(j), \qquad \text{Eq. 5-11}$$

from $\mathcal{A}$ inconsistency with the density function $d_A(\mathbf{a}|\mathbf{e}_{Aj})$. Although not indicated in the notation in use, it is possible that the sample space $\mathcal{A}$ could change with changing values for $\mathbf{e}_{Aj}$. The result would be the mapping

$$\left[\mathbf{e}_{Mj}, f\left(t \mid \mathbf{a}_{jk}, \mathbf{e}_{Mj}\right)\right], k = 1,2,\ldots,m(j), j = 1,2,\ldots,m, \qquad \text{Eq. 5-12}$$

that combines aleatory and epistemic uncertainties. As before, this mapping could be explored with procedures for the analysis of binary data. On an a priori basis, it is difficult to tell how successful efforts to extract meaningful insights for this large amount of pooled information combining aleatory and epistemic uncertainties would be.

Q2, What is the probability of pipe rupture? The view taken here is that the probability of pipe rupture should be defined on the basis of random variability in the initiation and evolution of cracks (i.e., on the basis on aleatory uncertainty). Specifically, this leads to the following definition and approximation for the probability $p_R(t|\mathbf{e})$ of pipe rupture by time t conditional on the vector $\mathbf{e} = [\mathbf{e}_A, \mathbf{e}_M]$ epistemically uncertain analysis inputs:

$$\begin{aligned} p_R(t \mid \mathbf{e}) &= \int_{\mathcal{A}} f\left(t \mid \mathbf{a}, \mathbf{e}_M\right) d_A\left(\mathbf{a} \mid \mathbf{e}_A\right) \mathrm{d}\mathcal{A} \\ &\cong \sum_{j=1}^{m} f\left(t \mid \mathbf{a}_j, \mathbf{e}_M\right) / m, \end{aligned} \qquad \text{Eq. 5-13}$$

where $\mathbf{e}_j$, $j = 1, 2, \ldots, m$, corresponds to the sample in Eq. (5-6).

It is possible to go a step further and define an expected probability $E_E[p_R(t|\mathbf{e})]$ of pipe rupture by time t over both aleatory and epistemic uncertainties. Specifically, $E_E[p_R(t|\mathbf{e})]$ can be defined and approximated by

$$\begin{aligned} E_E[p_R(t \mid \mathbf{e})] &= \int_{\mathcal{E}} \left[\int_{\mathcal{A}} f\left(t \mid \mathbf{a}, \mathbf{e}_M\right) d_A\left(\mathbf{a} \mid \mathbf{e}_A\right) \mathrm{d}\mathcal{A} \right] d_E\left(\mathbf{e}\right) \mathrm{d}\mathcal{E} \\ &\cong \sum_{j=1}^{m} \left[\sum_{k=1}^{m(j)} f\left(t \mid \mathbf{a}_{jk}, \mathbf{e}_{Mj}\right) / m(j) \right] / m, \end{aligned} \qquad \text{Eq. 5-14}$$

where $\mathcal{E} = \mathcal{EA} \times \mathcal{EM}$, $d_E(\mathbf{e}) = d_{EA}(\mathbf{e}_A)\, d_{EM}(\mathbf{e}_M)$, and the indicated samples are defined in Eq. 5-10 and 5-11 . However, all the epistemic uncertainty information is lost in $E_E[p_R(t|\mathbf{e})]$.

Q3, What is the uncertainty in an estimate of the probability of pipe rupture? This question is answered by the epistemic uncertainty present in the pipe rupture probability $p_R(t|\mathbf{e})$ defined in Eq. (5-13). Specifically, the spread in the values for $p_R(t|\mathbf{e})$ that results from the possible values for $\mathbf{e}$ characterizes the uncertainty in an estimate of the probability of pipe rupture. This spread is approximated by the mapping

$$\left[\mathbf{e}_j, p_R\left(t \mid \mathbf{e}_j\right)\right], j = 1,2,\ldots,m, \qquad \text{Eq. 5-15}$$

between epistemically uncertain analysis inputs and the rupture probability generated with the sample in Eq. (5-10). Once the indicated mapping is generated, the epistemic uncertainty in $p_R(t|\mathbf{e})$ can be displayed with cumulative and complementary cumulative distribution functions, and the effects of the individual elements of $\mathbf{e}$ on the uncertainty present in the estimate of $p_R(t|\mathbf{e})$ can be investigated with a variety of sensitivity analysis procedures.

Examples. It should be possible to develop examples to accompany the above discussions of the indicated questions. The present example problem used to illustrate the xLPR analysis does not serve well to illustrate the separation of aleatory and epistemic uncertainty. In particular, the model for crack initiation time does not seem reasonable as it over constrains the times at which crack initiation occurs. As a result, there is almost not aleatory uncertainty in the times of crack initiation. Intuitively, it seems like a crack initiation model that is closer to a stationary, or perhaps nonstationary, Poisson process would be more reasonable.

6. Pilot Study Evaluation and Recommendations

The work presented herein supports the xLPR Pilot Project objective to evaluate the feasibility of the software tool for use in support of licensing, rulemaking, design, and regulatory decisions by both industry and NRC. The results of the pilot study in Section 5.0 address the specific issue of assessing the probability of rupture of DM, pressurizer surge nozzle welds degraded by PWSCC. The pilot study provided a short term learning experience can benefit the longer term program and code development by identifying areas requiring more focused effort.

The following discussion is a summary evaluation of the pilot study effort, including recommendations and a self assessment of the xLPR Framework developed in the pilot study. This feedback will be used to refine the planning and approach used in the long-term program. Later development phases will broaden the scope of xLPR to all primary piping systems in pressurized and boiling water reactors, using an incremental approach that incorporates the design requirements from previous iterations.

6.1 Evaluation of Uncertainty

The uncertainty in the system has been handled using a probabilistic approach, i.e. associating each uncertain parameter to a probability distribution and propagating the uncertainty through the system using sampling techniques. A more complete description can be found in Appendix D.

As for the studies cited in Appendix D, effort has been made to separate epistemic from aleatory uncertainty. This distinction is not trivial as the determination of one parameter into one or the other category is subject to interpretation and may be specific to the problem considered. However, the process has helped to define more precisely each uncertain input. We can also make the distinction between risk (i.e., inherent randomness) leading to calculation of probability of failure, and uncertainty (i.e., poorly known unique values) representing the state of knowledge.

A Simple Monte Carlo sampling was used over the aleatory loop. Stability analysis presented in Section 4.3.2 shows that a sample size of 20 is usually enough, but 50 seemed more appropriate. LHS was used over epistemic uncertainty to benefit from its dense stratification. The initial sample size, used for the base case has been set to 1000, for a total sample size (i.e., aleatory sample size x epistemic sample size) of 50,000. LHS is appropriate and usually used when the user wants to know where most of the uncertainty is, and is not intended to be accurate in the neighborhood of extreme quantiles of the output of interest, although it is usually better than SRS for defining these quantiles.

This initial study helps in understanding the behavior of the system as well as selecting amongst the input variables the ones whose uncertainty drives the uncertainty of the output of interest. The results, presented in Section 5.3, recognize the importance of sigma0_wrs and B1, respectively with a positive and negative influence.

While for the base case, a sample size of 1000 over epistemic uncertainty seems large enough to estimate the probability of rupture considering leak rate detection and inspection, limitations of such an approach is seen in Section 5.3.4 for the Safe End evaluation. Only a handful of realizations (6 out of 1000) lead to rupture before leak rate detection is applied. The addition of leak rate detection is beyond the accuracy of the method: only 6% of the realizations with rupture still lead to rupture when leak rate detection is considered. For the base case, 6% of 6 realizations means that not even one realization is expected to have such a condition.

In this situation, two approaches can be considered. The first one is to increase the sample size. However, this approach will result in a longer calculation time as well as larger output data files. When the original calculation time is on the order of hours and each output file size is on the order of one hundred MB, this can quickly become a limiting factor as increasing the sample size by an order of magnitude will have the same impact over the calculation time and file size.

The second approach consists of focusing on the area that matters (i.e., leading to probability of rupture) in the input hyperspace in order to have a better understanding of these regions (leading to a better estimate) while reducing the effort in the other areas of the input hyperspace. This is the purpose of importance sampling. Results of importance sampling are presented in Section 5.3 for the same problem (i.e., Safe End presented in Section 5.3.4).

While only 6 out of 1000 epistemic realizations lead to rupture in Section 5.3.4 (about half a percent of the runs), the use of importance sampling over Sigma0_wrs and Xc leads to almost 1000 runs out of 10,000 with rupture (about 10%). It is then necessary to correct with corresponding response weights to estimate correctly the probability of rupture. The probability based on 1000 realizations with rupture will be a lot more accurate than the one with only 6 realizations. Moreover, it is now possible to estimate the value of this probability when leak rate detection is considered (6% of 1000 realizations represents about 60 realizations in order to estimate probability of rupture with leak rate detection implemented).

In conclusion, while the use of the double loop with SRS and LHS to sample over aleatory and epistemic uncertainty is appropriate to understand and verify the system, as well as find the area of interest in the input hyperspace, for the xLPR program it is necessary to couple this method with some kind of importance sampling in order to estimate accurately some of the extremely low probability of rupture. DPD with importance sampling has been tested and found to be satisfactory.

While the feasibility and usefulness of importance sampling has been proven, it is recommended to pursue the effort for xLPR Version 2.0 to improve the efficiency and accuracy of the code. One of the difficulties remaining with this approach is that it is the responsibility of the user to analyze the results and select the variables which will benefit from importance sampling. Moreover, the user needs to find which region of the distribution needs to be densely sampled (low end? high end? middle?). DPD software allows adaptive importance sampling which may address this concern; however the module is not fully tested and was not available in Version 1.0 of the xLPR. Moreover, some reliability methods such as first order (FORM) and second order (SORM) should be able to find areas of interest with fewer runs than Monte Carlo, as long as such regions can be defined based on threshold values on the output set (for instance, the analysis performed in Section 5.3.3 seems to indicate that a fractional surface area cracked greater than 0.5 leads to pipe rupture even when leak rate detection is implemented).

6.2 Use of Commercial Software

As discussed in Section 1.4, several commercial software packages are marketed as framework codes. After a review of the available literature and a simplified evaluation, no commercial software alternative to the GoldSim Pro software was found to have the capabilities needed to build the xLPR framework.

When gauging the success or failure of the GoldSim Pro software for use as a framework for the xLPR code, one can simply say that the pilot study framework model developed in this program and the results presented in Section 5.0 of this report demonstrate that the software is capable. However, a more refined evaluation is necessary to pinpoint the strengths and weaknesses so that these can be compared with alternative approaches. In this section, we attempt to quantify both, based on our experience using the commercial software GoldSim to build a modular framework for the xLPR pilot study.

We have broken down the evaluation into several parts. Each part represents either a technical consideration or programmatic consideration. Some of the concerns like technical requirements (e.g., probabilistic capabilities), of the software must be met, others that are more programmatic (e.g., development time) are considerations that need to be included if all of the technical requirements have been met. Aside from the obvious technical requirements, the categories we have defined are based on our modeling experience and expertise as developers of complex models and insight to what an end user might need; they are not intended to be an exhaustive list.

6.2.1 Modular Framework

The GoldSim code was designed to be both a simulation environment and a simulation framework. The GoldSim Pro software was designed with the capability to connect to external modules for situations when complex functions cannot be readily implemented using the default features supplied with the software. GoldSim software was specifically designed such that you can develop separate program modules (written in C, C++, Pascal, Fortran or other compatible programming languages) which can then be directly coupled with the default GoldSim algorithms. These user-defined modules are referred to as external functions, and the elements through which they are coupled to GoldSim are called DLL elements (see Section 3.0). The software excels in this attribute, as the xLPR codes were integrating by adding a "wrapper" (or "shell") around the existing codes, and compiled into a DLL. In most cases, this required only a limited number of programming modifications.

Once the DLL interface is defined (e.g., input and output arrays) then modules can easily be switched. Another feature is the ability to activate or deactivate each module, such that a user can select from a number of available modules. The GoldSim software was designed for probabilistic analyses, so the framework has the default capabilities to meet this xLPR requirement.

6.2.2 Probabilistic Capabilities

GoldSim was designed from the ground up as a general-purpose, probabilistic simulation framework. Section 3.0 details and Section 5.0 demonstrates both the embedded Monte Carlo and LHS capabilities in the software as well as the versatility to substitute external sampling modules to the framework. This technical requirement is met with the commercial software, but as noted previously, due to the very low probability events, the native capabilities of the software will likely be augmented by a module with specialized focused sampling capabilities. The software has an extensive library of uncertain distribution types which facilitated the input interface during framework model development. One limitation is that the distribution type cannot be changed using the free GoldSim Player software. This may not be an issue given the likelihood of utilizing a supplemental sampling module as mentioned previously. Additionally, using the embedded SubModel features facilitated the separation of aleatory and epistemic uncertainties (see Sections 1.6 and 3.1) without requiring a customized module. One limitation encountered included the serial processing of the inner and outer looping. While the user can evaluate all outcomes including the mean response, the average over the inner loop (e.g., aleatory samples) cannot be viewed in the current pilot study model framework. A demonstration case using a customized module was developed to investigate the ability of the current architecture to be modified to accommodate this functionality if future design requirements mandate. This demonstration case confirms that it is indeed feasible.

6.2.3 Software Flexibility

When considering the software flexibility, both the GoldSim Pro software and the GoldSim Player Software must be considered. The pilot study framework model was constructed using only the stock functions (i.e., elements) available with the GoldSim Pro software. The free GoldSim Player software has a limited set of capabilities. The GoldSim Player software is end-user software, which allows the model to be run and the results analyzed. As discussed in Section 3.0 with the GoldSim Pro software, the development team was able to construct the xLPR framework model with the full range of options specified for the pilot study. The GoldSim Player user has the ability to switch between modules, turn them on and off, select alternative model options, and set up a number of different configurations of Monte Carlo and DPD settings. With the Excel interface the user has the option to change the input values. In addition the user can swap modules at any time given the new module meets a few basic requirements. Limitations (i.e., inflexibility) of the player software can quickly be reached when the model needs to be used for unanticipated situations or if the modules have changed significantly. Another limitation is that the user cannot add new parameters or change the distribution types with the current version, although some of this can be mitigated by coupling the module with a database instead of using Excel for the inputs (see Section 6.2.8). None of these limitations apply to the GoldSim Pro software, as a licensed user has the ability to add or delete parameters, make changes to the distribution types and values, add functionality and modify the model logic, basically full change control using the GUI interface. This is one of the selling points for the GoldSim software as users can walk through the model graphically and make changes. The influence links are checked automatically to ensure they are valid and redrawn so changes can be observed instantly, without having to recompile source code or navigate through many lines of source code.

6.2.4 Verified Software

All commercial software goes through some form of software V&V that it is operating as designed. The xLPR program will benefit from the industry standard software development and software CM program already in place. In terms of software QA, commercial software must meet specific criteria and the software vendor must be available for surveillance and auditing. The GoldSim software has previously been used in a quality affecting work and has successfully met the requirements under a software QA program.

6.2.5 Robustness of the Software (Stability)

The GoldSim software package is based on technology developed over a period of almost 20 years. GoldSim has been used by and/or for a diverse set of customers and clients, including government agencies in over 10 countries (such as the DOE, NASA, the NRC, and the California Department of Water Resources), research laboratories (including SNL, Los Alamos National Laboratory, the Paul Scherrer Institute, and Massachusetts Institute of Technology), and commercial organizations worldwide, including Amazon.com, Bechtel, BHP Billiton, Caterpillar, CH2M Hill, ConocoPhillips, Mitsubishi, Newmont Mining, Northrop Grumman and Rio Tinto.

The diverse user base and active software CM program, including bug reporting and software patches (see Section 6.2.6) over a period of many years has significant advantages over a new program. The losses to schedule and budget that result from tracking down and fixing software bugs can be substantial. There are many examples of complex model applications using the GoldSim software which attest to the robustness of the software to meet the xLPR program needs. The SNL development team for the xLPR Pilot program did not encounter any software issues. The software was robust and stable. A special feature of the software allows the user to run the modules as separate processes, so that module instability (e.g., module crashes) can be error trapped by the software framework (see Section 3.1.2). This runs the modules a bit slower, but is used extensively during the model development phase. In addition the software internally checks the model logic (causality sequence) to ensure there are no logical errors and the calculations are done in the correct sequence and automatically update the connections (links) between model elements when an element is added, modified, or deleted from the model framework. These development features contribute to the overall model stability.

6.2.6 Software Development and Software Technical Support

The GoldSim software is a commercial product and therefore the source code is not open source. Software development is the responsibility of the software vendor. Since several of our recommendations below may result in software modifications to enhance features to facilitate the operation of xLPR code (e.g., result file management) and other as yet unknown issues may arise, it is an important consideration.

Required changes and enhancements to the software will need to be conducted by the team of software developers at GoldSim Technology Group (GTG). SNL has extensive experience with the GTG software development team and have found them to be highly experienced in the latest object oriented programming methods and have always been able to meet our technical and program needs. Since GoldSim is a specialized product, the software vendor is able to meet individual and programmatic needs and have defined a business model around this objective.

Since the model logic and implementation are decoupled from the software development (see Section 6.2.7), this reduces the need for software development for xLPR. The successful completion of the pilot study did not require any software modifications. We believe this is an asset to meeting the program schedule and requirements, especially QA requirements (see Sections 6.2.4 and 6.2.5).

Being a commercial software company GTG provides extensive software technical support including: on call and email technical support to address installation issues, answer questions on how to use GoldSim, and help analyze and optimize models and GoldSim Training (webinars, periodic courses, and specialized on-site training). In addition, software patches and bug fixes are distributed to the users as needed and the user’s guide and documentation are continuously updated and available online. All of the technical support capabilities are maintained by the software vendor at no cost to the xLPR program.

6.2.7 Model Development Using the Software

As outlined in Section 3.0, using the GoldSim Pro software, models are constructed by linking together graphical elements with predefined functionality to create, document, and present models by creating and manipulating graphical objects representing the components of your system, data and relationships. The software elements include predefined and user defined functions and data. The construction of the complex model for the xLPR pilot study required hundreds of elements. In order to manage, organize and view such a model it is essential to group the elements into subsystems. A subsystem is simply a collection of elements. Subsystems are created in GoldSim by placing elements into Containers. A Container is simply a "box" into which other elements have been placed. Containers were placed inside other Containers, and a number of levels of containment were easily created. This ability to organize model elements into a hierarchy provides a powerful tool for creating "top-down" models, in which the level of detail increases into the containment hierarchy. The pilot study framework model was constructed using the software features available with the GoldSim Pro software and utilizes both basic and advanced software features. After the code was constructed, checked, and debugged using GoldSim Pro, a model is saved as a player file that can be freely distributed and used without a licensing fee.

Using a commercial software framework the model development environment is decoupled from the software development. The decoupling of the model development can be advantageous in a program that utilizes an incremental approach to meet the software and model QA requirements outlined in this report (Section 1.0).

6.2.8 Model Interface

The pilot study framework model developed for the xLPR program utilizes several of the software features to facilitate the model interface. Because GoldSim is graphical simulation software, the model interface is a focal point and offers a number of options. The Version 1.0 framework focuses on the interface functionality available with the GoldSim player software. As mentioned above and in Section 3.0, with the GoldSim Pro software the user has complete access to view, modify, or delete any model element or attribute. However, the GoldSim Player software primary utilizes the dashboard interface, and has limited functionality available to the user. However, the Version 1.0 framework developed for the pilot study was developed to maximize the functionality available with GoldSim Player software.

The model interface utilizes dashboard elements to select model settings, module options, to monitor the simulation and to view and post process the simulation results. In addition the interface with Microsoft Excel was established to facilitate the data input. All of the model logic (e.g., model elements), links, notes, and subsystem hierarchy is available for interrogation and viewing with the GoldSim Player software. Although the model cannot be modified with the GoldSim Player software, given the options included on the dashboard interface the software seems to be sufficient to support the end user requirements. Limitations of using the Excel spreadsheet interface for data input, which specifically include the inability to change the uncertain distribution type, can be eliminated by utilizing an ODBC compliant database (dB) for input data which allows the user to change the input types.

6.2.9 Documentation of the Model

The use of GoldSim software includes many options for model documentation. The software allows the developer to add graphics, explanatory text, notes and hyperlinks anywhere within the model file. The xLPR model is organized in a hierarchical manner, with added graphics, notes and hyperlinks inserted in key locations to demonstrate the documentation features of the software. Hyperlinks were inserted that connect the user to the xLPR CM SharePoint site and the xLPR Framework User's guide. Reference information was included for the MRP-216 report [40]. Any number of links, text, notes, and graphics can be added to the model file to facilitate the documentation. The software was specifically created with the tools to internally document a model such that the documentation becomes part of the model itself, and hence is immediately available to anyone viewing the model. Specific notes and descriptions can be added to any element or container in the model file. In addition, if a dB link is utilized, the reference information including the date that the data was downloaded from the dB can be transferred to each individual model input and saved with the model run.

GoldSim has powerful charting and display functions that allow plotting and viewing result data in a variety of ways, including time histories of the data, probability distributions, scatter plots, bar charts, and tables of results. The software creates a run log which contains the information about the run, including the version of the GoldSim software used as well as name and version model, of the DLL modules and any errors or warnings that may have occurring during a simulation. A set of customized error messages were added to the xLPR model to monitor the module calculations (Section 3.0).

6.2.10 User Base and Development Community

GTG maintains a web based user community which includes:

- A Video Library, with a number of short videos describing how to use various GoldSim features.

- An active online GoldSim Community, consisting of a Forum page, as well as a the GoldSim Wiki, which includes a Model Library, Knowledge Base and a wide variety of articles, all of which can be edited by our user community.
- Periodic email newsletters to keep users up to date on the latest GoldSim news.
- Periodic webinars to describe new GoldSim features and capabilities.
- Periodic GoldSim User Conferences and Specialized Training.

6.2.11 Use of Commercial Software Summary

GoldSim was designed to facilitate the construction of large, complex models. Models are built in a hierarchical, modular manner, such that the model can readily evolve and detail can be added as more knowledge regarding the system is obtained. Other powerful features, such as the ability to manipulate arrays, the ability to "localize" parts of the model, and the ability to assign version numbers to a model which is constantly being modified and improved, further facilitate the construction and management of large models.

Overall, the use of commercial software looked appropriate for the pilot study framework for several reasons. As members of a computational workgroup, a significant part of the time was spent studying different techniques that needed to be tested and refining the algorithms to implement them within the framework. For any complex system, it is common to have to test different algorithms in order to select the most appropriate and one of the main goals of the pilot study was to develop a modular and flexible framework allowing such tests and changes. GoldSim® revealed to be not only appropriate but often helpful during the development phase. As the framework is modular, most of the models were included in DLLs (and GoldSim allows DLL in languages such as C and Fortran) that were optimized at this level. GoldSim was used to sample the uncertain parameters and serve as an interface between models. The graphical representation of the model allows for a direct representation of the pilot study model flow chart clearly showing the relationship between modules and framework. Debugging is often as long as, if not longer than, writing the algorithms for such a complex system. A clear and easy to navigate framework is an advantage as it is not only simpler to understand but also helps finding module and framework bugs.

Another advantage we found in GoldSim was the embedded file management capabilities. Large sample sizes result in large output files. Most of the data is saved in the model as zipped binary files, taking less space than an uncompressed file (by about a factor of 10). Only the output directly used are automatically extracted (allowing post-processing analysis), but other information remains retrievable in the framework model file if needed. Additionally, the framework automatically checks to ensure the correct version of the DLL module files and EXCEL input file are available before running an analysis. If distributed processing is used, the software automatically copies the files to all of the nodes, ensuring that the analysis is performed correctly.

Of course, commercial software also has some drawbacks. The sampling strategy using a double loop requires an advanced approach which can affect model transparency. However, most of the features that were needed to match the project requirement took only a few days to implement and verify.

6.3 Self Assessment – xLPR Framework Acceptance Criteria

The xLPR acceptance task group is tasked with developing a protocol and metrics for comparing GoldSim software with an alternative approach developed by ORNL called SIAM. The comparison will be based on both the pilot study version and the perceived capabilities of a future version of the codes. The comparison will rank both alternatives with both quantitative and qualitative measures. The metrics were drafted and are included in this section. We have included, as part of our evaluation, a self assessment of the xLPR framework implemented in GoldSim using the draft acceptance criteria. The self assessment will attempt to identify both the strengths and weaknesses of the current version, as well as assess the potential for improvement that might be necessary in future versions. Likewise the ORNL team will be providing the xLPR program a self-assessment of SIAM, which will be documented in their report. The objective comparison between the two codes will be documented in a follow-on NRC report and include feedback from NRC and an independent evaluation of the two codes.

This self assessment includes a qualitative grading system (not met, met, or exceeded). A brief justification of the rating is given for each of the draft acceptance criteria.

Ease of use of the code

Availability of User's Manual – EXCEEDED. A Version 1.0 user's manual was created as part of this effort and has been transmitted separately. The user's guide contains a simple set of step by step instructions on how to download the GoldSim Player software, controlled model files, modules, and input from the xLPR CM web site. The user's guide contains the basic instructions necessary to run the model. Detailed instructions are contained within the model itself. Since the code was built using commercial software, an extensive user's manual and user's knowledge base already exists and is available online.

Ease of input

i. Organized Structure – EXCEEDED. As presented in Sections 3.0 and 6.2, the framework utilizes both a hierarchical designed and has the capability for extensive self documenting features (notes, graphics, links, etc).
ii. Save/Read input decks for editing – MET. The use of an Excel spreadsheet facilitates the input. The common interface enables the user to easily change and view the inputs without having to navigate through the model hierarchy. Using the dB interface would elevate this to exceed.
iii. Convenient for sensitivity studies – MET. The pilot study model includes a number of dashboard controls that enable the user to easily switch between various modeling options to create numerous sensitivity cases. All of the model options are located on one input screen. Graded on the features available with the GoldSim Pro software this criteria would be rated as exceeded. However the more limited functionality of the player software reduces the rating.

Ease of execution

iv. Batch runs possible – NOT MET. No batch function running is currently available. This is something we could request from the software developer, if desired. The software does have the capability to run on multiple processors when using the GoldSim Pro software (distributed processing) that significantly decreases the run times.

v. Run status displayed – EXCEEDED. As demonstrated with the Version 1.0 framework the software has the capabilities to add any result status for run time monitoring.

vi. Pertinent results displayed – EXCEEDED. As demonstrated with the Version 1.0 framework the software has the capabilities to add any result status for run time monitoring.

vii. Ease of installation – EXCEEDED. The software installation package distributed by the software vendor is simply downloaded from their web page and installed. Once the software has been installed the user must simply open the model file.

Ease of Post Processing

viii. File Size – MET. The GoldSim model file has a very good compression algorithm that reduces the model file size by up to a factor of 10. However, this requires some time during model run saving and later when the model is opened (the results need to be unzipped). Post processing makes use of several external modules developed by SNL. These external modules have to read ASCII text files that are automatically exported from the framework. These files are not compressed and can be quite large. Exporting the files can take a while. This area needs some improvement.

ix. Interface – MET. The interface between the framework and the post processor is facilitated through the use of the dashboard interface. The interface allows the user to browse the model file for desired results, export them, and post process the results all from a single interface. The complexity of the post processing codes however and length of time to export the files makes this an area that needs some improvement. The efficiency gains by using a two step approach to evaluate the affects of inspection and leak detection are quite a nice feature.

Presentation of results

x. Flexibility in presenting different results – EXCEEDED. The plotting and charting features available in the GoldSim software allow the user to view any saved results. In addition, the GoldSim Pro software allows the user to save all of the results for every element and calculations performed in the framework. The GoldSim Player software can also view or graph any results; however the selection of which or how much of the results are saved cannot be changed.

xi. Aesthetics – EXCEEDED. As discussed previously, the software has extensive capabilities to create a customized interface. The software allows the developer to add graphics, explanatory text, notes and hyperlinks anywhere within the model file. The xLPR model is organized in a hierarchical manner, with added graphics, notes and hyperlinks inserted in key locations to demonstrate the documentation features of the software.

Code efficiency (set up time, run time, post-processing time)

EXCEEDED. The software package is downloadable from the software vendor's web site and easily installs on a Microsoft Windows operating system. The model file is then simply opened. There are no special set up steps required. To run the code the only requirements are that the GoldSim player software is installed on a user's computer and that the model file set is available at the same location as the model file. Run time is relative criteria since the majority of the run time is devoted to the modules not the framework. However, the framework has been optimized to reduce the run times with logic that only runs the modules when they are needed, terminates a simulation when rupture occurs, and skips realizations where no cracks will occur. The GoldSim Pro software has the capability to run the framework using distributed processing, which automatically distributes the simulation over multiple central processing units (CPUs) and/or computer nodes. This requires no special installation other than the GoldSim Pro version of the software. The basic license allows for the run to access four additional nodes. An optional software add-on can be purchased that allows the run to be distributed to any number of additional nodes. This feature results in significant run time efficiency, an 8 hour model run can be run in 1 hour on eight CPUs. Given the current availability of multiple core CPUs it is not unthinkable that a dual quad-core CPU computer can be readily available. The distributing feature is an included capability of the GoldSim software, but is not available with the free GoldSim Player software. The only significant source of overhead is due to file sizes. Saving, opening, and exporting results from very large files can be time consuming. As mentioned previously this is a function of the number of samples needed to get a good result for the very low probability occurrence of rupture and as a result is not unique to this framework. However the software does automatically compress the files, so that the storage space is minimized, the compression ratio is about 10 to 1. Thus 10 GB of results can be stored in the GoldSim model file at a cost of only 1 GB of hard drive space.

Adaptability and Flexibility of the Code

EXCEEDED. GoldSim was designed from the ground up as a general-purpose, probabilistic simulation framework. Section 3.0 details and Section 5.0 demonstrates both the embedded Monte Carlo and LHS capabilities in the software as well as the versatility to substitute external sampling modules to the framework.

The pilot study framework presented in this report demonstrates the flexibility of the code to link to the external modules, implement the nested looping, and alternatively substitute between the internal Monte Carlo sampling and an external module using importance sampling. The software has distributed processing capabilities and the ability to link to an input database using an ODBC connection. Certain practical limitations are met when the file sizes are very large as a result of the large number of realizations necessary when Monte Carlo is used. However, this is not really a software issue, rather just a reality of creating so much data. This issue has not sufficiently been addressed in the pilot study and needs to be addressed in Version 2.0.

Post processing is accomplished using two additional modules developed by SNL specifically for the xLPR pilot study. Post processing occurs after the simulation has completed to give the user the ability to evaluate any number of various combinations of inspection times and leak detection rates without having to pay the expense of re-running model. This approach was approved by the computational group and is extremely efficient when considering the very large number of samples that are required to get stable results for the very low probability results.

Minimization of Technical /Computational Limitations of the Code

Ability to Implement Submodels with Technical Rigor – EXCEEDED. The software and framework were developed using an incremental approach that is integral with the CM process. The modular framework and software features, like the ability to make standalone models, cut and paste entire models into the framework, versioning, tracking changes internally, automatically testing links and causality sequences, and internal unit consistency checks available with the software are keys to ensuring the technical rigor. As mentioned previously there are mostly practical considerations in terms of the number of samples needed and the resulting run time and file sizes. There are limitations in ability of the software used to build the framework to handle these issues. While computationally the significant bulk of the calculations are done by the individual modules, the software framework is very efficient and does not represent a significant source of overhead in terms of run time. With distributed processing very large simulations can be completed. However, the software package is a 32-bit application and there is a limit for very large data results which cannot be exported to ASCII text files. During development we encountered this limit with simulations that had 720 time steps and over 100,000 realizations when trying to export a matrix with 19 cracks. This was mitigated by exporting the results into 19 different files, one for each crack. Again, this issue is an xLPR issue not so much an individual framework issue.

Ease of Code QA

This criterion is a function of both the model QA and software QA. As the xLPR framework was developed using a commercial software package, the evaluation focuses on the model QA. Commercial software QA requires much less effort than that which is required for developed software. The modules used in the framework need to be qualified and QA compliant software CM systems and processes for xLPR need to be in place. The CM process used for the pilot study is sufficient for model and module QA; however it is not optimized for software CM.

> *Ease of Version Control* – EXCEEDED. This criterion is strongly coupled with the CM process developed for xLPR by SNL and is not just framework or software dependent. However, the embedded capabilities of the GoldSim software to include versioning, version reporting, and to visually track the changes by color coding the graphical element within the GoldSim model file, make this criteria easily met. The software itself is maintained by the software vendor and therefore meeting QA requirements for the xLPR program can be easily met.
>
> *Ease of Document Control* – EXCEEDED. Again, this is a function of both the framework's internal documentation features (as noted in the previous section) and an effective CM process. The framework model automatically generates a run-log which documents the software and module versions, errors encountered and the date of the run.

The run log is saved with the model file and cannot be deleted or changed unless the model results are destroyed.

6.4 Lessons Learned

This section outlines some of the lessons learned from the xLPR the pilot study.

6.4.1 xLPR Project (Roles, Responsibilities, Goals)

- The project roles, responsibilities and goals were outlined well for the problem. The foresight to execute a pilot study to define the program requirements was essential to address this complex problem.
- Incongruent progress amongst the various task groups and organizations toward meeting the program goals has been an issue during the pilot study.
- As we have encountered in previous projects, the role of the computational group cannot be to simply plug the models altogether and generate the results. The computational group has an important role of integration to be sure that each model is used appropriately **within the context considered**. Moreover, they have an important role in implementing and testing uncertainty characterization and implementation.
- A poorly defined scope and incongruent schedule adherence within the Input and Models Task Groups during the pilot study had significant consequences on the pilot study model and analyses. The computational team had to fill the gaps when missing data or poorly tested models were discovered. A more involved project integration board and a formalized leadership role of the computational task group will be necessary moving forward to meet the xLPR long term goals.
- The lack of the availability of the Models/Input Task Group Version 1.0 report was significant in that there was not sufficient documentation and cross referencing available for the work completed and documented in this report.
- Without strong leadership from the computational group, the pilot study would not have been successful.

6.4.2 Evaluation of Commercial Software

- The availability of suitable commercial software for evaluation for the xLPR framework was extremely limited (Section 1.4).
- Most of the commercial software is either too simplistic or too specific to adapt well to the xLPR framework model requirements (Section 1.4).
- No suitable commercial software alternative to the GoldSim software was found (Section 1.4).
- SNL developed software was found to be suitable for the xLPR framework but would require a development team to enhance and customize the code for xLPR (Section 1.4, Appendix A).

6.4.3 Use of GoldSim

- The GoldSim software is easily adapted for use on the xLPR program (Section 3.0).
- The software is well documented with an extensive Users Guide, has an online knowledge base, and technical support.
- The free GoldSim Player software file has most of the available features needed for the end user and was carefully constructed to meet the pilot study framework requirements (Section 3.1.7).
- Although internal calculations are carried out using double precision numbers, results are only stored as single precision numbers (in order to reduce storage requirements). This means that when results are viewed in tables or charts, the range of values that can be displayed is between -1.2E-38 and 3.4E38. This is also true for exporting results. This is significant in that the overall DPD response weight could not be saved in the framework and had to be scaled so the exported weights could be passed to the post processor. The response weights are renormalized in the post processor.
- The use of a database with an ODBC connection to the model file would improve the capabilities available using the GoldSim Player Software, as well as enhance traceability and input control and configuration management.
- The availability of the distributed processing capabilities greatly reduced run times and enhanced the code efficiency.
- The customizable dashboard interface, the ability to embed graphics and text is necessary for documenting and operation of the model.
- The native Monte Carlo and LHS capabilities of the software worked well, but the importance sampling options did not seem to be aggressive enough for the low probabilities in xLPR. More work needs to be done to fully evaluate these software capabilities.
- The GoldSim Monte Carlo and LHS sampling was easily bypassed and replaced by the DPD module.
- The versioning feature was essential during model development and checking.
- It is not a viable option to use the Player Software for model development, although it was sufficient for model analyses and modules can be easily switched provided they have the same structure and file name.
- The software excels in the ease of coupling modules and modularity.
- The GUI interface, advanced browsing features (Function of and Affects views), and the ability to save all the data from intermediate calculations are extremely powerful capabilities for development, debugging, checking, and model analysis.
- The version of GoldSim used in this study (Version 10.11SP4) still used the old Excel standard. Thus exporting results directly to Excel files is limited to 255 columns, insufficient for the number of realizations for the analysis. The next version of the code, 10.5, will include an update to allow Export to Excel 2007 or later versions which can handle 1 million rows and 65,535 columns.

6.4.4 xLPR Model Development

- The choice to establish first an alpha version then a beta version of the framework model was a good one. The development process is inherently an iterative process. As new improvements or capabilities are added to the code or as previously undiscovered bugs are found and addressed, new versions are created.
- Redesign of the framework logic or modification to a module frequently occurred to correctly capture the phenomenon considered or appropriately represent the response in a downstream model. The framework development cannot be considered as a simple plugging of modules within a probabilistic loop.
- The modular approach worked well with the CM process, project roles and responsibilities and xLPR objectives.
- Using a stable and verified commercial software package facilitated the framework and model development and debugging effort, since there were no framework software bugs to address.
- Without the CM process and requirements to document and control the modules developed by the models group, it would not have been possible to construct a robust and stable xLPR model for the pilot study.
- Two approaches could have been taken to implement the nested loop approach for the xLPR framework model. The first approach would be to embed the aleatory parameters and time-loop within an epistemic outer loop. In the second approach the time loop would be contained in the outer model with the two separate epistemic and aleatory SubModels run at time=0 in order to generate the input sample set for each outer model realization. The first approach has advantages in terms of an intuitive design, however the functional outcome of embedding a complete GoldSim model of the complexity of the time-loop, greatly reduces the transparency of the calculations within the model calculations when the Player file is used. Therefore the second approach was selected for the xLPR pilot study (Section 3.0).
- A computational team member must be involved with the model and data task groups during module and data development.
- The construction of standalone subsystem models to develop and integrate the modules with the xLPR framework optimized the time required for model development. In addition this form of unit testing enhances the robustness of the integrated model (Section 3.1).
- The xLPR framework model was difficult to validate given only the deterministic test cases were available. While these deterministic test cases can be hand checked, sufficiently detailed field data from operating power plants is needed to validate the expected results.
- SNL did not have sufficient time or resources to develop a database for the xLPR pilot study model.
- The schedule must allow for iteration between the computational task group and the model and inputs task groups to refine and test the xLPR model components.

6.4.5 xLPR CM

- Establishment of a CM process including strict documentation, checking and review of the technical basis was essential.
- A CM program has to start before any model development.
- The CM process and guidelines were easily adopted by all the organizations.
- Having a web accessible electronic CM system was essential to the success of the CM process.
- The draft CM process required a CM lead role to provide guidance, training, review and approval necessary to keep the electronic CM system in order.
- The SharePoint software was not ideal for an electronic CM system as the software is designed for documents, thus the large file size and various file types were not recognized by the default settings. A considerable about of attention from the SNL CM lead and Battelle SharePoint administrator was required to maintain the electronic CM system.
- An electronic storage system with versioning and web access was essential to the pilot study success by enabling and encouraging collaboration and provided a comment document library.

6.4.6 Use of SharePoint

- SharePoint is primarily for document collaboration on a central server or server farm and is accessed by users either through a compatible web browser or directly via Microsoft® Office. However, while not ideal due to limits on file size and lack of desirable file level controls, the SharePoint site was sufficient to meet the needs of the SNL developed CM process for the xLPR pilot study.
- Site design and control was fairly easy to learn but takes some time to maintain. A continuous review and attention is required.
- The upload times were long and the file size limitations required assistance from a Battelle SharePoint administrator, via a Battelle FTP site, to post model files with results.
- The check out and check in feature in SharePoint was not available at all of the various organizations sites. This feature when enforced prevents simultaneous changes to the same document by multiple contributors. This is extremely important to maintain configuration management. Additionally, this feature only works on Microsoft Office files, Word Documents, Excel Spreadsheets, etc., and does not apply to source code files or GoldSim model files.
- The SharePoint system has limitations on the number of characters used in the directory names, thus when setting up the CM system directory names had to be shortened to get the desired levels on the directory structure.

6.4.7 Use of the Framework Model

- Very large model files were produced when all of the data were saved. It is not tractable to save all the data and this option was only used during model development, checking, and debugging.
- It currently takes the software quite a long time to export the very large result files.

- Once a modeling case has been run and the distribution of responses have been generated, the user can evaluate the effects of an unlimited number of combinations of inspection times and leak detection limits, using the post-processing tools (Section 3.2) without having to re-run the model case each time.
- Post Processing is extremely efficient when considering the very large number of samples that are required to get stable results for the very low probability results.
- Large sample sizes are not only slower to run but also generate large output that can become a problem. Time step sensitivity analyses have been performed to palliate part of the computational and file size burdens. A Subset of Outputs of interest has been selected amongst the output generated to reduce the amount of data saved.
- The use of importance sampling is necessary once the calculation times and disk space requirements become prohibitive when the sample size exceeds 100,000.
- The framework model was capable of utilizing alternative sampling and importance sampling to get stable results for the low probability events using the DPD method.
- In regard to run times and data file sizes, the analysis of leak rate detection and inspections in post-processing was a good tradeoff between added complexity and reduced transparency to avoid rerunning the code several times.
- The very low frequency rupture events and crack initiations were discovered to be extremely sensitive to the calibrated parameters used in the crack initiation model.
- The insights and recommendations in terms of the assessment of these very low probability ruptures is only a preliminary assessment of the problem.
- Computational limitations encountered in the pilot study were a result of the computational approach, which still needs some work to optimize the calculation of these extremely low probability events.

6.4.8 xLPR Uncertainty

- One of the purposes of this pilot study was to demonstrate the capability to estimate extremely low probabilities. In consequence, multiple options need to be tested in order to optimize the framework.
- The base case model for the xLPR pilot study does not serve well to illustrate the separation of aleatory and epistemic uncertainty. Only a preliminary effort has been made to separate epistemic from aleatory uncertainty. This distinction is not trivial as the determination of one parameter into one or the other category is subject to interpretation and may be specific to the problem considered.
- The stability of the analysis is difficult to assess due to the classification of the uncertainty, aleatory parameters have a negligible effect on the distribution of the results, presented in Section 5.2.1.
- The time of occurrence of cracks in the future is 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.
- The model for crack initiation time does not seem reasonable as it over constrains the times at which crack initiation occurs. As a result, there is almost no aleatory uncertainty in the times of crack initiation. Intuitively, it seems like a crack initiation model that is closer to a stationary or perhaps a nonstationary Poisson process would be more reasonable.

- The use of the double loop with SRS and LHS to sample over aleatory and epistemic uncertainty is appropriate to understand and verify the system, as well as find the area of interest in the input hyperspace. For the xLPR program it is necessary to couple this method with some kind of importance sampling in order to estimate accurately some of the extremely low probability of rupture.
- While for the base case, a sample size of 1000 over epistemic uncertainty seems large enough to estimate the probability of rupture considering leak rate detection and inspection, limitations of such an approach is seen in Section 5.3.4 for the Safe End evaluation. Only a handful of realizations (6 out of 1000) lead to rupture before leak rate detection is applied. The addition of leak rate detection is beyond the accuracy of the method: only 6% of the realizations with rupture still lead to rupture when leak rate detection is considered. For the base case, 6% of 6 realizations mean that not even one realization is expected to have such a condition.
- A quantitative measure to determine whether stability is achieved is desirable. However, any quantitative measure will have to be problem specific as well as its interpretation (Section 4.3.2.2).
- When DPD is used each realization is a random combination with replacement, the use of bootstrap is appropriate in order to estimate uncertainty due to sample size. Each subset of the original sample will be a valid sample by itself (Section 4.3.2.3).
- The application of importance sampling over LHS values can be very powerful. However, as for the regular LHS, it is not appropriate to use bootstrap estimates in order to construct the distribution of the mean. Re-sampling with replacement will break the LHS structure and may lead to "holes" in the sampled input distribution. Therefore a more traditional approach uses replicates and estimates confidence bounds over the distribution of the mean using t-distribution. However, bootstrap can still be applied in a qualitative way in order to see how much the distribution deviates from normal. (Section 4.3.2.4)
- Importance sampling is needed to focus on the region of the input sample space that generates more rupture when leak detection and inspection are evaluated.
- 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 choice of the distribution used and to a lesser extent the parameters selected, are mainly based on the 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 worse in some unlucky conditions.
- DPD with importance sampling has been tested and found to be satisfactory.
- Section 5.2.2.8 shows one of the limitations of using linear or monotonic regression when performing sensitivity analysis. While most of the important input-output relations follow a monotonic relation, it is not always the case and other non-monotonic relationships could have serious consequences when estimating low probabilities of rupture.
- Analysis in Section 5.3.7 has underlined the importance of a correct classification of uncertainty. While the change in classification is not likely to change the final result of interest (mean probability of rupture) the interpretation of intermediate results as well as quantiles can change significantly.

- 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 uncertainty analyses.

6.5 Recommendations

The following subsections include recommendations based upon the preliminary work done during development and analysis of the xLPR pilot study. Although it is outside the scope of this report, it should be noted that conclusions and recommendations contained herein are dependent on the robustness of the conceptual models used the calculation of PWSCC, crack stability, and leakage rates. Each of the modules used in Version 2.0 needs to be verified and validated using the best available in service data. The xLPR Models Task Group report documents the various modules and models used in this analysis.

6.5.1 Treatment of Uncertainty

Treatment of uncertainty is a key component of any major study (see Appendix D for more details). 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. Uncertainty is not trivial and needs involvement at all levels of development of a complex system. Of course the input group has a major role in describing the uncertainty of each input, but it has to work conjointly with the model group and 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 as, once sensitivity analysis is performed, the results should be communicated to the model group and input group in order to check that they make sense and each part of the code as well as the whole code perform as expected and give reasonable results. It is recommended strongly to follow such an approach as the time involved in such a task is wisely used, insuring that results makes sense and can be explained and that no uncertain quantity has been wrongly characterized. Indeed, 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 and invalidate the model. Moreover, conservatism at a subsystem (i.e., model) level can become a non-conservatism when implemented at the system level. Most of the uncertainty then should be revisited at least once.

Adaptive importance sampling needs to be explored as a possible method to enhance the confidence in the low probability outcome. Moreover, some reliability methods such as FORM and SORM may be able to find areas of interest with fewer runs than Monte Carlo, as long as such regions can be defined based on threshold values on the output set. Several improvements are considered as necessary in order to be able to offer efficient and easy to use importance sampling to the end user. One of the most desired properties would be an automatic selection of the regions of the input sample space that lead to the greatest probability of rupture conditional on selected assumptions. Many optimization and reliability methods have been developed to answer such questions and it is recommended to study the possibility of using some of these techniques in future versions of the xLPR code. In addition, the GoldSim code has importance sampling options within the software. Some discussion with the software vendor is recommended to evaluate modifying the software to include more aggressive or adaptive importance sampling techniques.

The sensitivity analysis techniques considered in this version of the xLPR framework should be extended to include more sophisticated techniques allowing the capture of non-monotonic relationships.

We were quite surprised to see the uncertainty parameters controlling the time and location of crack initiation classified as epistemic. Although we consider ourselves not necessarily qualified to choose a classification in the present situation, we strongly recommend for each uncertain parameter to be revisited for both its distribution and its classification. While there still may be disagreements in terms of a distribution and classification, a rationale should be written on the uncertainty characterization to support the analysis and interpret the results conditional on these assumptions.

6.5.2 Framework Development

The framework has to be incrementally developed and checked. The computational group development team should be involved at a high level and understand the whole purpose of the project. We strongly recommend planning enough time and an iterative approach be used to facilitate understanding each part of the model as well as the global model at a physical, mathematical and computational level. It is better to spend some (wisely spent) time understanding the purpose of the simulation first, as it will save a lot more time at the end.

The benefit of creating freeze points, or developmental versions, enables the periodic assessment and improvement of the code. The final Version 1.0 framework greatly benefited from this approach and it should be included in the Version 2.0 planning for both the model framework and module development.

It is recommended that the commercial software be used for Version 2.0 of the xLPR program. The results of this study indicate that the commercial software is sufficient to meet the needs of the xLPR modular framework. The commercial software can produce a customized model framework and greatly reduces the software QA requirements. Elimination of framework software development, software CM and QA will free up resources that can be applied to addressing the computational challenges. However, regardless of the approach, commercial or open source, the very large number of realizations necessary to produce stable results creates a set of very large data files. It currently takes the software quite a long time to export the very large files. Alternative options including the use of binary files and more aggressive importance sampling techniques need to be explored. This recommendation may require software modification.

6.5.3 Quality Assurance/Configuration Management

QA 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, 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., IS9000, 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. Section 1.5.2 outlines the CM process used in the pilot study, which were not linked to a QA process. A plan for the xLPR QA program for software development, model development, and input development are needed (e.g., NUREG/BR-0167).

Fundamentally all xLPR participants must participate in the QA program 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 then 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. For instance, SNL has corporate procedures in place that requires an evaluation and plan for work under a QA program. In the pilot study we demonstrated the use of program level guidance and a common CM system that was used by all of the participants regardless of their organizational affiliation. 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.

6.5.4 Representation and Analysis of Uncertainty

The pilot study has shown the crucial role of importance sampling in order to calculate correctly extremely low probabilities of rupture, specifically for the safe end sensitivity case. 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. This first analysis showed the importance of sigma0_wrs in this sense. However, some other parameters will probably influence the probability of rupture. In order to optimize the system and estimate more accurately low probabilities, a more complete sensitivity analysis must be performed on several outputs to thouroughly investigate the link between the inputs and the probability of rupture under defined conditions (for instance, linking the expected fractional surface area cracked above the threshold of 40% with the probability of rupture when leak detection is considered). This approach will help find potential relations between input and output that need to be studied. The use of adaptive importance sampling should also be tested as it can help find these kinds of relationships more precisely.

Additionally and perhaps more importantly, the evaluation and classification of uncertain parameters including a formal review of the distributions used for the uncertain parameters in the xLPR model are essential.

7. References

1. Software Platform Evaluation, Verifiable Fuel Cycle Simulation (VISON) Model, INL/EXT-06-01076. Jacobson, J.J., Shropshire, D.E, and West, W.B., November 2005.
2. Belcourt, K.N., BRISC Integration of Simplified xLPR Problem, (unpublished).
3. Software Platform Evaluation, Verifiable Fuel Cycle Simulation (VISON) Model, INL/EXT-06-01076. Jacobson, J.J., Shropshire, D.E, and West, W.B., November 2005.
4. Helton JC. Uncertainty and Sensitivity Analysis in the Presence of Stochastic and Subjective Uncertainty. Journal of Statistical Computation and Simulation 1997;57(1-4):3-76.
5. Helton JC, Burmaster DE. Guest Editorial: Treatment of Aleatory and Epistemic Uncertainty in Performance Assessments for Complex Systems. *Reliability Engineering and System Safety* 1996;54(2-3):91-94.
6. Paté-Cornell ME. Uncertainties in Risk Analysis: Six Levels of Treatment. *Reliability Engineering and System Safety* 1996;54(2-3):95-111.
7. Winkler RL. Uncertainty in Probabilistic Risk Assessment. *Reliability Engineering and System Safety* 1996;54(2-3):127-132.
8. Hofer E. When to Separate Uncertainties and When Not to Separate. *Reliability Engineering and System Safety* 1996;54(2-3):113-118.
9. Parry GW, Winter PW. Characterization and Evaluation of Uncertainty in Probabilistic Risk Analysis. *Nuclear Safety* 1981;22(1):28-42.
10. Helton JC. Probability, Conditional Probability and Complementary Cumulative Distribution Functions in Performance Assessment for Radioactive Waste Disposal. *Reliability Engineering and System Safety* 1996;54(2-3):145-163.
11. Hoffman FO, Hammonds JS. Propagation of Uncertainty in Risk Assessments: The Need to Distinguish Between Uncertainty Due to Lack of Knowledge and Uncertainty Due to Variability. *Risk Analysis* 1994;14(5):707-712.
12. Helton JC. Treatment of Uncertainty in Performance Assessments for Complex Systems. *Risk Analysis* 1994;14(4):483-511.
13. Apostolakis G. The Concept of Probability in Safety Assessments of Technological Systems. *Science* 1990;250(4986):1359-1364.
14. Haan CT. Parametric Uncertainty in Hydrologic Modeling. *Transactions of the ASAE* 1989;32(1):137-146.
15. Kaplan S, Garrick BJ. On the Quantitative Definition of Risk. *Risk Analysis* 1981;1(1):11-27.
16. Der Kiureghian A, Ditlevsen O. Aleatory or Epistemic? Does It Matter? *Structural Safety* 2009;31:105-112.
17. Helton JC. *Conceptual and Computational Basis for the Quantification of Margins and Uncertainty*.SAND2009-3055.Albuquerque, NM: Sandia National Laboratories 2009.
18. Helton JC, Johnson JD, Sallaberry CJ, Storlie CB. Survey of Sampling-Based Methods for Uncertainty and Sensitivity Analysis. *Reliability Engineering and System Safety* 2006;91(10-11):1175-1209.
19. Frey HC, Patil SR. Identification and Review of Sensitivity Analysis Methods. *Risk Analysis* 2002;22(3):553-578.
20. Saltelli A, Chan K, E.M. Scott (eds). *Sensitivity Analysis*. New York, NY: Wiley, 2000.

21. Cullen AC, Frey HC. *Probabilistic Techniques in Exposure Assessment: A Handbook for Dealing with Variability and Uncertainty in Models and Inputs*. London; New York: Plenum Press, 1999.
22. American Institute of Aeronautics and Astronautics. *AIAA Guide for the Verification and Validation of Computational Fluid Dynamics Simulations*. AIAA G-077-1998.Reston, VA: American Institute of Aeronautics and Astronautics 1998.
23. Roache PJ. *Verification and Validation in Computational Science and Engineering*. Albuquerque, NM: Hermosa Publishers, 1998.
24. Oberkampf WL, Trucano TG. Verification and Validation in Computational Fluid Dynamics. *Progress in Aerospace Sciences* 2002;38(3):209-272.
25. Oberkampf WL, Trucano TG, Hirsch C. Verification, Validation, and Predictive Capability in Computational Engineering and Physics. *Applied Mechanics Review* 2004;57(5):345-384.
26. Roache PJ. Building PDE Codes to be Verifiable and Validatable. *Computing in Science & Engineering* 2004;6(5):30-38.
27. Babuska I, Oden JT. Verification and Validation in Computational Engineering and Science: Basic Concepts. *Computer Methods in Applied Mechanics and Engineering* 2004;193(36-38):4057-4066.
28. Trucano TG, Swiler LP, Igusa T, Oberkampf WL, Pilch M. Calibration, Validation, and Sensitivity Analysis: What's What. *Reliability Engineering and System Safety* 2006;91(10-11):1331-1357.
29. Oberkampf WL, DeLand SM, Rutherford BM, Diegert KV, Alvin KF. Error and Uncertainty in Modeling and Simulation. *Reliability Engineering and System Safety* 2002;75(3):333-357.
30. 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.
31. 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.
32. Materials and Reliability Program Crack Growth Rates for Evaluating Primary Water Stress Corrosion Cracking (PWSCC) of Alloy 82, 182 and 132 Welds (MRP-115), EPRI, Palo Alto, CA: 2004. 1006696.
33. Materials Reliability Program: Materials Reliability Program: Development of Probability of Detection Curves for Ultrasonic Examination of Dissimilar Metal Welds (MRP-262, Rev. 1) EPRI, Palo Alto, CA: 2009. 1020451. MRP-262, Rev. 1.
34. 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.
35. 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.
36. Battelle, NRCPIPE User’s Guide (Windows Version 3.0), Contract NRC-04-91-063, April 1996.

37. "User's Guide for SQUIRT (Windows Version 1.1)," Battelle Memorial Institute, March 2003.
38. GoldSim Technology Group LLC, 2009, GoldSim User's Guide, Volumes 1 & 2, Version 10.0, GoldSim Technology Group LLC, Issaquah, Washington.
39. GoldSim Technology Group LLC, 2009, GoldSim Player User's Guide Version 10.0, GoldSim Technology Group LLC, Issaquah, Washington.
40. 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.
41. Mattie PD, Kalinich DA, Sallaberry CJ. *U.S. Nuclear Regulatory Commission Extremely Low Probability of Rupture Pilot Study: xLPR Framework Model User's Guide*. SAND2010-7131.Albuquerque, NM: Sandia National Laboratories 2010.
42. Cox DR, Snell EJ. *Analysis of Binary Data*, 2nd edn. New York, NY: Chapman and Hall, 1989.
43. Materials and 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. 10109082.

APPENDIX A. POTENTIAL FRAMEWORK SOFTWARE EVALUATION – SIMPLIFIED TEST PROBLEMS

A.1 Simplified xLPR Demonstration Case for Evaluation of Commercial off the Shelf (COTS) and Sandia Open Source (OS) Framework Software

The goal of this study is to test alternative software as a probabilistic framework for connecting existing codes and running them over a range of uncertainty in the inputs. The test case uses Alpha model version of the xLPR modules.

Description of the Probabilistic Framework

Uncertainty will be represented using a probabilistic approach. Each uncertain parameter will be associated with a probability distribution and classified as either epistemic or aleatory. The uncertainty will be propagated using Monte Carlo sampling. A first loop (outer loop) will be used to generate a sample size of n_E on the epistemic parameters. A second loop (inner loop) will be used to sample for each epistemic set n_A futures on the aleatory parameters. The mathematical model will then be run n_E x n_A times.

Description of the Deterministic Model

The model for this demonstration case will only evaluate the evolution of cracks on a single weld through time over a 60 year period. One or several cracks will appear (initiate) at different time steps following a Poisson process. Initially SCs will eventually grow to TWCs. The demonstration model will include three subroutines written in Fortran90. These routines are used to propagate cracks to an abstracted weld. In addition to these three subroutines, an initiation module will have to be implemented, following the description given in this document.

The code will be run deterministically (i.e., no uncertainty will be sampled in it), and the input (and sampled values) will be passed from the framework. Each code will be called as many times as necessary by the probabilistic framework (e.g., once per realization). The simulation will be performed for 720 months (60 yrs) with a time step of 1 month.

At every time step, the code will check for potential new cracks based upon the distribution calculated using a Poisson process (crack initiation module to be developed as part of the demonstration case) then, for **each existing crack** it will estimate the growth in half length and depth making the distinction between a SC and TWC (using stress intensity and grower module source code).

Crack Initiation Module

The initiation of cracks follows a Poisson process with a frequency of cracks equal to 0.6 per plan life (i.e. 60 yrs = 720 months). The annual frequency is then 0.01.

As there is no reason to simulate cases with no cracks, it is recommended either to ignore and resample cases with 0 cracks, or to use a discrete distribution conditional on at least one crack.

The discrete distribution with conditional probability (in column 3) is given below in Table A-1. For the purpose of illustration, GoldSim went up to 8 cracks. However, a maximum of 4 cracks have been observed for sample size of 10^3. A maximum of up to 6 or 8 or even 15 cracks can be used or directly the number from conditional Poisson distribution in Table A-1.

Table A-1. Conditional Probability Distribution for test case.

nb Cracks	Probability	Conditional Probability
1	3.29E-01	7.29E-01
2	9.87E-02	2.19E-01
3	1.97E-02	4.37E-02
4	2.96E-03	6.56E-03
5	3.56E-04	7.88E-04
6	3.56E-05	7.88E-05
7	3.05E-06	6.75E-06
8	2.29E-07	5.06E-07
9	1.52E-08	3.37E-08
10	9.14E-10	2.02E-09
11	4.99E-11	1.10E-11
12	2.49E-12	5.52E-11
13	1.15E-13	2.55E-12
14	4.93E-15	1.09E-13
15	1.97E-16	4.37E-15

For each generated crack, the time of occurrence, location, depth and half-length will be sampled from the distribution indicated in the aleatory uncertainty input parameters. All of these data can be implemented initially outside of the time loop, or during the time loop at the developer's convenience.

Stress Intensity Modules

Two Stress intensity modules are available: one for SC and the other for TWC. One of the two modules is called for each separate crack (depending whether this crack is a SC or a TWC) at each time step.

Ksurf.f90 generates two stress intensity results: K0 (tangent) and K90 (normal) (subroutine calcK)

The inputs of the routine are (in the following order):

- Rovert: Ratio of pipe inside radius (R) to pipe thickness (t)
- Covera: Ratio of half length (C) over crack depth (a)
- Aovert: Ratio of crack depth (a) over pipe thickness (t)
- Sig0: Axial stress component
- Sig1: Curve fit parameter for through-thickness distribution
- Sig2: Curve fit parameter for through-thickness distribution
- Sig3: Curve fit parameter for through-thickness distribution
- Sig4: Curve fit parameter for through-thickness distribution
- Sig5: Global bending stress
- a: Crack Depth

Ktwc.f90 generates only one stress intensity result: Ktwc (corresponding to K0) (subroutine calcTWCK).

The inputs of the routine are (in the following order):

- Sig0: Axial stress component
- Sig1: Curve fit parameter for through-thickness distribution
- Sig5: Global bending stress
- Pi: 3.1415926…
- Theatoverpi: Half crack length (C) divided by inside radius (R) then divided by Pi
- Rovert: Ratio of pipe inside radius (R) to pipe thickness (t)
- t: Pipe thickness

Grower Module

Subroutine Grower from **Grower.f90** is used for both SC and TWC. The module is called for each separate crack (with a distinction between SC and TWC for some inputs) at each time step. It estimates the updated crack depth (a) and half-crack length (C)

The inputs of the routine are (in the following order):

- PWSCC_C: Constant C value for PWSCC
- PWSCC_Q: Constant Q value for PWSCC
- PWSCC_Tref: Constant Tref value for PWSCC
- PWSCC_Kth: Constant Kth value for PWSCC
- PWSCC_Beta: Constant Beta value for PWSCC
- Temperature: temperature
- Time_interval: time step in yrs
- R: Inside radius
- t: Thickness
- a: Crack depth
- C: Half-crack length
- Ksurf90: normal stress (= 0 for TWC)
- Ksurf0: tangential stress (= Ktwc for TWC)

Change from SC to TWC

Once the depth of a SC becomes larger or equal to the pipe thickness, the crack becomes a TWC. The depth size is then equal to the thickness. The half length is corrected such as the area of the crack remains identical to the previous value for the SC. A SC area is considered as semi-elliptical. A TWC area is represented as a portion of the ring. The relation between half-length for TWC (C_2) and half-length for SC (C_1) is given below:

$$C_2 = \frac{R_i}{R_i + \frac{t}{2}} \frac{\pi C_1}{2}$$

Where R_i represents the inside radius of the pipe and t is the pipe thickness.

Inputs Characterization

Epistemic Uncertain Input

For Ksurf and Ktwc:

- Sig0 membrane (MPa): Uniform [10; 20]
- Sig0 wrs (MPa): Uniform [350; 400]
- Sig0 (MPa) = Sig0 membrane + Sig0 wrs
- Sig1 (MPa): Uniform [-3550, -3500]
- Sig2 (MPa): Uniform [7150, 7200]
- Sig3 (MPa): Uniform [-4015; -4000]
- Sig4 (MPa): Uniform [0.032; 0.036]

For Grower:

- Temperature (Degree Celsius): Normal (mean = 288; stdev = 16.7)
- Pressure (psi): Normal (mean = 2250; stdev = 50)

Aleatory Uncertain Input

For Crack Initiation

- n_{crack}: number of cracks, Poisson distribution
- τ_I : time of flaw (i.e. crack) i (months): Uniform [0; 720]
- d_i: initial depth for crack i (m): Uniform [0.00025, 0.0025]
- l_i: initial length for crack i (m): Uniform [0.000013, 0.0013]
- Θ_i: initial crack location (radian): Uniform [-π, π]

Constant Inputs

- R_i: inside radius (m): 0.075
- t: pipe thickness (m): 0.05

For Ksurf and Ktwc:

- Sig5 (MPa): 100
- Pi: 3.141592654

For Grower:

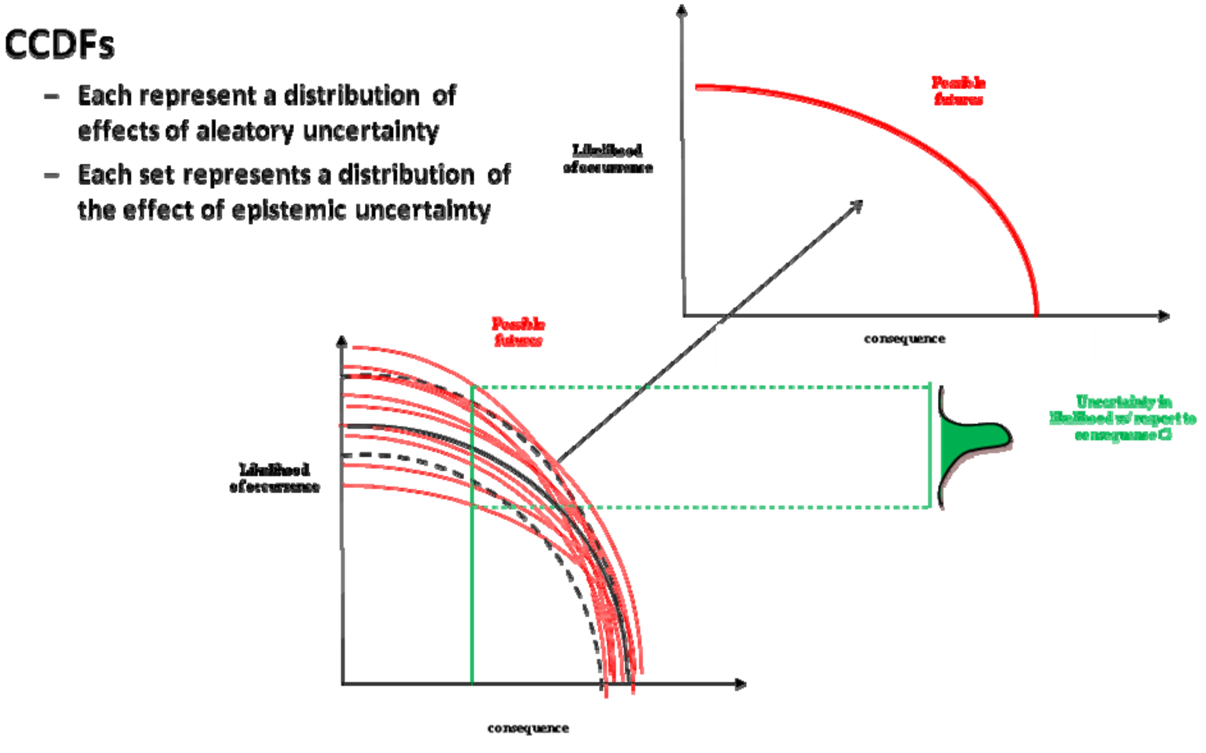
- PWSCC: 2.69998x10^{-8}
- PWSCC_Q (degree Kelvin): 15655.555
- PWSCC_Tref (degree Kelvin): 598
- PWSCC_Kth: 0
- PWSCC_Beta: 1.6

Results

Results expected for this demonstration case are:

- The number of cracks for each epistemic and aleatory realization;
- The status evolution for each crack (no crack → SC → TWC) for each epistemic and aleatory realization; and
- The time-dependent evolution of crack depth and half crack length for each of the cracks for each epistemic and aleatory realization.

For Example:



CCDFs

- Each represent a distribution of effects of aleatory uncertainty
- Each set represents a distribution of the effect of epistemic uncertainty

A.2 Evaluation of Powersim Studio

Introduction

SNL was asked to evaluate other COTS software as an alternative to the current GoldSim implementation of the xLPR software. SNL has considerable expertise in Powersim Studio, a system dynamics methodology COTS package. We were asked to complete the task by November 30, 2009 and began the task on November 3, 2009. The last set of information on the xLPR GoldSim solution was provided on November 16, 2009. SNL staff spent approximately 30 hours examining, learning about, and trying to reproduce the xLPR software code as actualized in GoldSim software with Powersim Studio. The overall opinion of the team is that Studio can simulate the xLPR problem as GoldSim did. Our caveats are:

1. The xLPR problem as stated is not a system dynamics problem and therefore does not make use of the methodology implemented in Studio;

2. The Studio software is capable of calling external routines but we were not successful in doing so due to lack of capability on our team; and

3. The team felt that we only scratched the surface in understanding the problem domain. This was due to a lack of expertise in this area and a lack of time.

Features of Studio

Powersim Studio is a system dynamics (www.systemdynamics.org) model building platform. Studio enforces the system dynamics methodology although one can build non-system dynamics models with Studio. A system dynamics model is useful for examining problems that exhibit delay, feedback, accumulation, and flows (of information or materials).

Studio implements the methodology of system dynamics with 4 basic constructs: levels, flow rates, constants, and information links in a graphical interactive development environment. In addition to the basic system dynamics methodology Studio has more than 100 built in functions, tools for building user interfaces, and the capability of embedding VBScript routines.

Requirements for xLPR

Understanding the requirements of xLPR in a matter of hours is not an easy task. After reading the provided material it is obvious that years of material science research have been applied to this model. Although the basic idea of propagating cracks in pipes is quite simple, none of the staff has any experience in materials or nuclear science that might aid a better understanding of the xLPR problem.

The requirements for the simulation of the xLPR problem were provided in the following documents:

1. Draft Program plan for alpha xLPR development - October 2009;

2. Program plan for alpha xLPR development;
3. xLPR_Activity_Diagram_10_01_09;
4. xLPR_Activity_Diagram_Time_Loop_10_01_09; and
5. xLPR Demonstration Case for Software Evaluation.

In addition, Fortran programs for crack propagation were provided. These programs were in the form of DLL. The Fortran codes were treated as a ‘black box’.

Matching Requirements to the Capabilities of Studio

The xLPR problem is not strictly a system dynamics problem. Although crack initiation and propagation have aspects of feedback, delay, accumulation, and flow rates those characteristics were hidden from us in the Fortran DLLs. Thus, Studio’s system dynamics methodological features were not employed.

Studio Implementation of Probabilistic Framework

Requirement: Dealing with Uncertainty

Uncertainty will be represented a using probabilistic approach. Each uncertain parameter will be associated with a probability distribution and classified as either epistemic or aleatory.

The uncertainty will be propagated using Monte Carlo sampling. A first loop (outer loop) will be used to generate a sample size of n_E on the epistemic parameters. A second loop (inner loop) will be used to sample for each epistemic set n_A futures on the aleatory parameters. The mathematical model will then be run n_E x n_A times.

Studio has built-in stochastic functions that can generate series of random numbers as sampled from several probability distributions. In this case the entire model needs to be run n_E x n_A times. Studio can accomplish this in one of two ways:

1. Using the ‘Runs’ feature and running the model run $n_E\ x\ n_A$ times and sending the results to an Excel spreadsheet; and
2. Using the built in Risk Assessment and Management Tools that permit Monte Carlo simulation of an existing model.

Studio Implementation of Deterministic Model

Requirement: Dealing with Deterministic Model

The model for this demonstration case will only evaluate the evolution of cracks on a single weld through time over a 60 year period. One or several cracks will appear (initiate) at different time steps following a Poisson process. Initially SCs will eventually grow to TWCs.

The demonstration model will include three subroutines written in Fortran90. These routines are used to propagate cracks to an abstracted weld.

This requirement is the closest, methodologically, to a system dynamics model. This is where the majority of the Studio modeling effort has been expended.

Four modules need to be implemented in Studio:

a. Crack Initiation Module;
b. Stress Intensity Modules;
c. Grower Module; and
d. Change from SC to TWC.

Of these four, two are calls to external DLLs (stress intensity and grower) and have not been implemented in Studio.

Studio DRAFT Model

The next two figures represent the modules created in Studio to address the xLPR problem. The first deals primarily with inputs and input preparation. The second deals with the Crack Initiation Module and the Change from SC to TWC. The time of crack initiation is determined in a VBScript subroutine corresponding to the initial condition phase outside of the time loop. The length, depth, and location of the SC are stored in stocks. These are accessed at every time step to calculate crack area and determine progress toward a TWC.

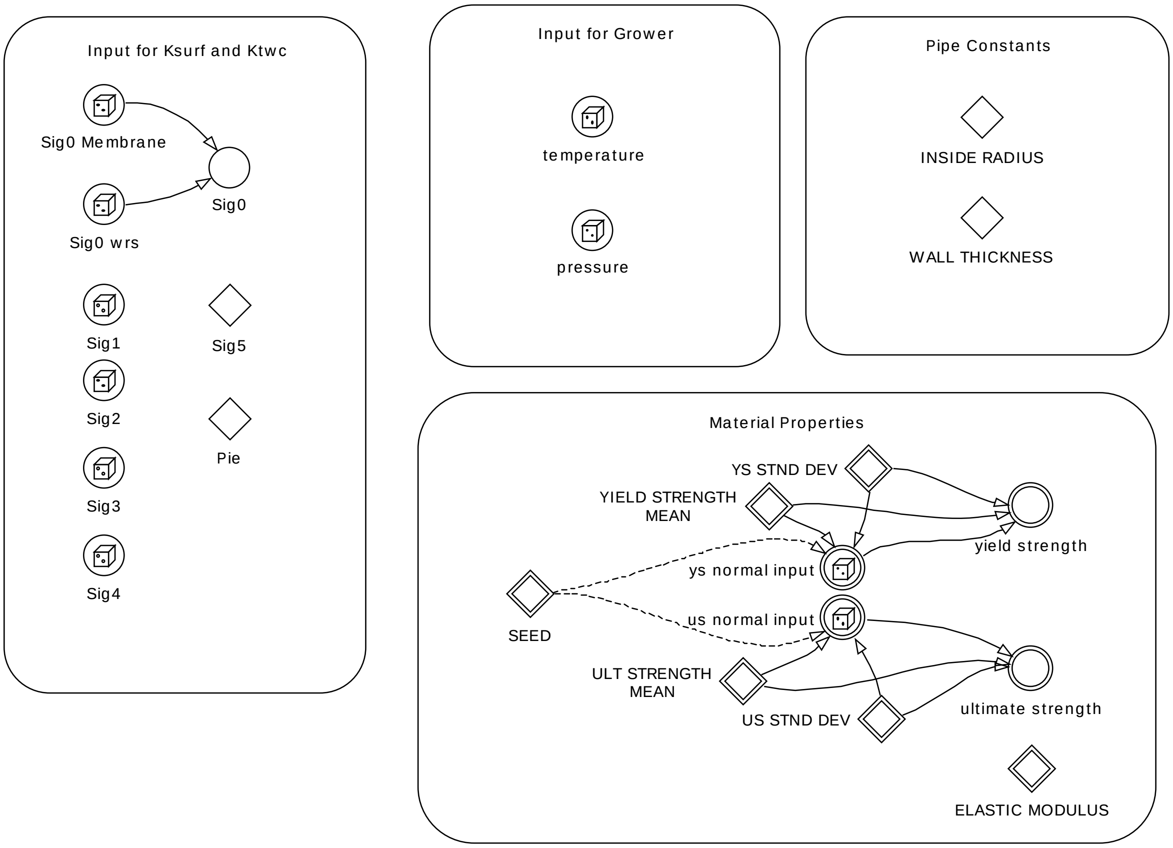


Figure A-1. Input data preparation.

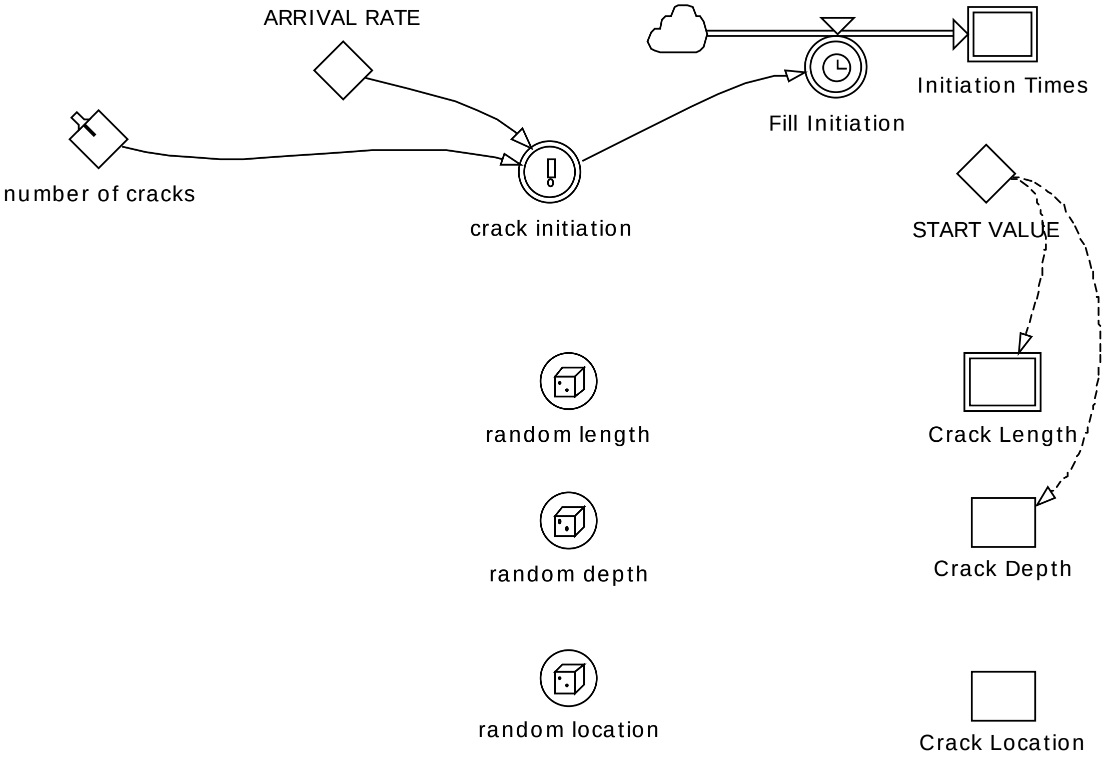


Figure A-2. Studio model.

Results

1. Studio can match the time-dependent nature of the xLPR crack propagation problem.

2. Studio can make calls to external DLL routines. This has been proven on other Studio projects with calls to Python and C#. We have not succeeded in making calls to the xLPR Fortran DLLs.

3. This is primarily due to our lack of familiarity with DLLs and an inability in VBScript to make direct DLL calls.

4. Studio has the necessary functions to produce series of random numbers. In the absence of a built in function, the VBFUNCTION() feature of Studio permits the construction of custom functions that can address this issue.

Table A-2. Powersim test case equation listing.

EQUATION LISTING			
Name	**Dimensions**	**Unit**	**Definition**
clock			NUMBER(TIME)
Crack Depth		cm	'START VALUE'
crack initiation	'Crack Number'		VBFUNCTION("initiate"='Crack Number' |"arrival"='ARRIVAL RATE', "cracks"='number of cracks'| //This is a routine to calculate time to crack initiation// "DIM P(720)" // fill the initial time series with zero// "for m = 0 to cracks-1" "initiate(m)=0" "next" //calculate factorial of crack number// "factorial = 1" "for i= 1 to cracks" "factorial = i*factorial" "next" //calculate probability matrix// "for k = 1 to 720" "P(k) = EXP(-1*arrival*k)*(arrival*k)^cracks/factorial" "next" //generate random time of initiation for each crack// "for j = 0 to cracks-1" "randomize" "k=0" "DO While k < 720 " "k = k+1" "random_numb = Int((720-2)*RND+1)" "value = P(random_numb)*k*0.01" "if value >= 1 then" "initiate(j) = k" "exit Do" "end if" "Loop" "next" //bubblesort the initiation times// "for i = 0 to cracks-2" "if initiate(i) >= initiate(i+1) then" "temp=initiate(i+1)" "initiate(i+1) = initiate(i)" "initiate(i) = temp" "end if" "next")*1<<mo>>
Crack Length	'Crack Number'	cm	'START VALUE'
Crack Location			0
ELASTIC MODULUS	'Material Type'	GPa	{186.3,177.1,203.1}
Fill Initiation	'Crack Number'		IF (NUMBER(TIME) <= 24001,'crack initiation'/1<<mo>>,0<<mo/mo>>)
Initiation Times	'Crack Number'	mo	0
INSIDE RADIUS		m	0.075<<m>>
number of cracks			1
Pi			Pi
pressure		psi	NORMAL(2250,50,0.5)*1<<psi>>
random depth		m	RANDOM(0.00025,0.0025, 0.5)*1<<m>>
random length		m	RANDOM(0.000013,0.0013,0.5)*1<<m>>
random location		rad	RANDOM(-PI, PI,0.5)*1<<rad>>
SEED	'Material Type'		{0.5,0.5,0.5}

Table A-2. Powersim test case equation listing (continued).

EQUATION LISTING			
Name	**Dimensions**	**Unit**	**Definition**
Sig0		MPa	'Sig0 Membrane'+'Sig0 wrs'
Sig0 Membrane		MPa	RANDOM(10,20,0.5)*1<<MPa>>
Sig0 wrs		MPa	RANDOM(350,400,0.5)*1<<MPa>>
Sig1		MPa	RANDOM(-3550,-3500,0.5)*1<<MPa>>
Sig2		MPa	RANDOM(7150,7200,0.5)*1<<MPa>>
Sig3		MPa	RANDOM(-4014,-4000,0.5)*1<<MPa>>
Sig4		MPa	RANDOM(0.032,0.036,0.5)*1<<MPa>>
Sig5		MPa	100<<MPa>>
START VALUE		cm	0
temperature		C	NORMAL(288,16.7,0.5)*1<<C>>
ULT STRENGTH MEAN	'Material Type'		{519.9,453.7,583}
ultimate strength	'Material Type'		('ULT STRENGTH MEAN' + 'US STND DEV' + 'us normal input')
us normal input	'Material Type'		FOR (i = 'Material Type' | NORMAL('ULT STRENGTH MEAN'[i],'US STND DEV'[i],SEED[i]))
US STND DEV	'Material Type'		{28.7,53.2,58}
WALL THICKNESS		m	0.05<<m>>
yield strength	'Material Type'		FOR (i= 'Material Type' | EXP(-1*(LN('ys normal input'[i]/1<<MPa>>-'YIELD STRENGTH MEAN'[i]/1<<MPa>>)^2/(2*('YS STND DEV'[i]/1<<MPa>>)^2))) / ('ys normal input'[i]/1<<MPa>>*'YS STND DEV'[i]/1<<MPa>>*SQRT(2*PI)))
YIELD STRENGTH MEAN	'Material Type'	MPa	{228.5,172.5,372}
ys normal input	'Material Type'	MPa	FOR (i = 'Material Type' | NORMAL('YIELD STRENGTH MEAN'[i],'YS STND DEV'[i],SEED[i])) //FOR (i = 'Material Type' |// // NORMAL('YIELD STRENGTH MEAN'[i],'YS STND DEV'[i],SEED[i])// //)
YS STND DEV	'Material Type'	MPa	{21.7,36.5,90.1}

A.3 Evaluation of FRAMES Software

FRAMES is a systems modeling software platform, developed by PNNL with funding from DOE, EPA, NRC, and DOD, for selecting and implementing environmental software models for risk assessment and management problems [A-1, A-2]. FRAMES is designed to dynamically introduce software modules representing individual components of a risk assessment (e.g., source release of contaminants, fate and transport in various environmental media, exposure, etc.) within a software framework, manipulate their attributes and run simulations to obtain results. A module contains one or more codes, models, or databases that meet the framework communication protocol. It allows legacy disparate models and databases to communicate in a plug and play atmosphere. FRAMES also allows users to develop a visual conceptualization of a specific scenario to be analyzed by constructing a pictorial display of the analysis using a drag-and-drop system of icons that represent each discrete part of analysis [A-3, A-4].

One of the major shortcomings of an early version (FRAMES 1.X) was its non-extensible architecture as well as the high number of bookkeeping tasks that module developers have to perform in order to incorporate individual modules in the system. This approach increased code maintenance and made correct file specification a critical aspect of the pluggable feature [A-4]. The improvements made in the most recent version (FRAMES 2.X) provide a highly extensible software system and reduce the overhead for module developers. In addition to the salient features of FRAMES 1.X, the improved system provides the ability to incorporate software modules representing individual components of a risk assessment process within a software framework. The software framework is designed using “object-oriented” design and allows for the decoupling of individual modules. This design greatly improves the ability of the module developers to “plug” the new module into the multimedia modeling system without the need to develop a complete modeling system from scratch. FRAMES 2.X allows a user to simulate contaminant-based exposure and risk in a multimedia environment, all at a single facility or at many facilities [A-4]. FRAMES 2.X has incorporated the modules of Multimedia Environmental Pollutant Assessment System (MEPAS), which is a suite of environmental models developed by PNNL to assess contaminated environmental problems. The MEPAS software integrates transport and exposure pathways for chemical and radioactive releases to determine their potential impact on the surrounding environment, individuals, and populations [A-5].

Most popular applications of FRAMES have been in the environmental arena, where it’s multiple "medium-specific" models (for example: air, water, and human impacts) as well as a database of chemical properties with associated environmental parameters have proven an effective way to solve risk analysis problems. Based upon the FRAMES 1.X platform, EPA developed a multimedia, multi-pathway, multi-receptor exposure and risk assessment modeling system (3MRA) in support of regulatory-based applications for hazardous waste management [A-6 to A-8]. This software system is an icon-driven, site-layout platform, which represents an interactive means by which the user graphically constructs a conceptualization of the problem by visually expressing the assessment, indicating sources of contamination, contaminant travel pathways through the environment, linkages between contamination and people or wildlife, and impacts associated with the contamination. The modeling system contains "sockets" for a collection of databases and computer codes that simulate elements of transport, exposure, and risk assessment, including contaminant source and release to and through overland soils, vadose and saturated zones, air, surface water, food supply, intake human health impacts, sensitivity/uncertainty, and ecological impacts [A-6 to A-8].

Adaptive Risk Assessment Modeling System (ARAMS^{TM}) was developed by the U.S. Army Corps of Engineers to conduct risk assessments to determine safe levels and cleanup target levels for military relevant compounds and to evaluate remediation alternatives [A-9 to A-11]. ARAMS is based on the FRAMES 1.X platform and incorporates various existing databases and models to simulate multimedia and multi-pathway fate/transport, exposure, intake/uptake, and effects of military relevant compounds and other constituents of potential concern to assess human and ecological health impacts/risks associated with chronic exposure [A-9 to A-11].

FRAMES 2.X provides improved ability for the users to "plug" disparate legacy software modules into the system, and allow broader usage of the framework outside the environmental science domain. This feature could be useful as a framework platform for a modular model system as envisioned for the xLPR application. However, one major drawback against potential application to the xLPR is that the software in its current version lacks the infrastructure for probabilistic analysis. For the xLPR application, modules to support probabilistic analysis, such as probability distribution functions, sampling schemes and necessary utility features need to be developed, implemented, verified and validated in the framework, along with appropriate wrappers for the modules conforming to the framework communication protocols. These may require substantial resources and time, which may make FRAMES 2.X not suitable for the xLPR application.

A.4 Evaluation of BRISC Software

Description of the Test Problem

The simplified xLPR test problem is designed to estimate the behavior of cracks in a theoretical weld on a pipe. The goal is to predict when cracks form, how the crack depth and length evolve over time, and when or if the pipe fails (ruptures) either because the crack length exceeds some maximum or the crack depth grows to the pipe thickness. Some of the test problem input parameters are not precisely known so they are represented by statistical distributions that constrain the range of possible values that the parameter can take. These so called uncertain parameters are grouped into either epistemic or aleatory uncertain parameters. Aleatory uncertain parameters govern the crack initiation time, length, depth and initial location as well as the number of cracks that form, while the epistemic uncertain parameters determine pipe stresses, pressures, and temperatures. For each epistemic or aleatory uncertain parameter, we sample the corresponding distribution using LHS, except for Poisson distributions, to ensure complete sampling of probability space.

The primary output from the test problem is the time evolution of each crack, for each aleatory realization, for each epistemic realization. Based on conversations with the customer the test problem was run with 100 epistemic and 100 aleatory realizations leading to approximately 10000 crack depth and length time evolution histories.

BRISC Implementation

The customer supplied Fortran 90 routines for computing the crack surface and through wall stresses, as well as the crack growth module. In addition customer documentation described the logic for computing the transition of a SC into a TWC as well as the maximum crack length computation.

Given this information, the BRISC framework was modified to support running the xLPR test problem in probabilistic fashion. Broadly the changes to BRISC included (1) extending our input file Extensible Markup Language (XML) syntax so we could support specifying parameters as distributions, mixed time-independent time-dependent transfers (e.g. from time-independent epistemic and aleatory physics to time-dependent crack initiation, stress and growth physics) and various new physics packages to sample epistemic and aleatory parameters and to compute crack initiation, surface stresses, TWC stresses, and crack growth, (2) adding code to parse xLPR input file constructs, construct model evaluators for each physics package, construct transfer operators for moving data from one physics to another physics package, and to construct the overall problem, and (3) added new model evaluators to both sample the epistemic and aleatory uncertain parameters as well as drive the nested realization loop (the aleatory realization loop calls the time integration loop in the Problem_Manager). The Fortran 90 routines are called through external C interfaces using standard C to Fortran calling conventions.

There is a single BRISC input file called Problem_Manager_setup.xml that is used to define the problem to run. It consists of logically coherent blocks (parameter lists) called Constants, Physics or Transfers. The Constants block was added so that users could change the value of various input quantities such as the pipe radius and thickness, the sigma5 stress, or the PWSCC quantities. The Physics block is used to define the name of the physics package and other input quantities used by the physics package. For the Epistemic and Aleatory Physics blocks, the input quantities include the sample size (number of realizations), the various distributions for the uncertain parameters and, for the Aleatory physics, a Boolean flag to indicate whether to resample the aleatory uncertainty for each epistemic realization (true) or only sample the aleatory uncertainty once (false). The four physics packages, one for crack initiation, surface stresses, through wall stress and crack growth are created so that we can specify how to transfer data between the various physics packages (i.e. they are only supplied with data transferred from other physics packages). The Transfer block describes the source and target of a data transfer (i.e. from Aleatory to crack initiation). The time integration loop is controlled either by time step or by a maximum number of iterations. The Max Iterations solver parameter was used and set it to 720 indicating that each step through the time integration loop advances one month or a total problem time of 60 years.

When built, the BRISC executable is called mpd.x (multi-physics driver executable). The executable should reside in the same directory as the input file Problem_Manager_setup.xml. By default the executable runs with no command line arguments and automatically looks for the input file in the current working directory. The executable will accept an optional single integer argument (N) on the command line, the number of cores to run the parallel simulation with. This integer value should always be less than or, at most; equal to the number of idle cores on the local system (it is not recommended to intentionally overloading your system by specifying more tasks than available free cores). Naturally if the integer command line argument is missing, the test problem is run in serial. The global xLPR test problem is partitioned into N tasks with each task computing M/N worth of epistemic realizations, where M is the total number of epistemic realizations. So, for example, if M is 100 epistemic realizations and N is 8 cores (or tasks), the multi-physics driver will round M up to 104, so that it is evenly divisible by the number of desired tasks (N), and then run 13 (104/8) epistemic realizations in each of the 8 tasks. The epistemic uncertain parameters are sampled once and then partitioned into M/N realizations so that we correctly sample the entire probability space independent of the size of the parallel run. Note that it could be chosen to run the same number of epistemic realizations on each core to eliminate load imbalance issues but it would not have easily supported other strategies.

Here is an example of the output when running the xLPR problem in parallel (note that there is no command line output from the multi-physics driver when run serially). If any errors are detected, they are reported to the command line and the simulation terminates.

```
kbelco$ ./mpd.x 8
waiting for child tasks to complete ... done
```

There are eight (8) output files that capture various computed quantities along with the time evolution of the crack history. The entire simulation output is located in the crack_history.txt file. It contains the complete history for the time evolution of each crack and is necessarily a rather large file.

The crack_avg.txt file contains the time step in column one, then the crack average depth for each epistemic realization, followed by the crack average lengths for each epistemic realization. There are no column labels to simplify the MATLAB script used to generate the plots. The average depth and length is computed over all the aleatory cracks during one epistemic realization.

The customer requires CCDF plots for the depth and length at time 10, 30 and 60 years. It was easiest to have the multi-physics driver compute these quantities and write them to separate files to simplify the amount of post-processing imposed on the customer. The files are called ccdf_depth_10.txt, ccdf_length_30.txt, and so on. A separate MATLAB script has been included in our transmittal that, if run in the same directory as the output files, will automatically post-process the output data files and produce the desired plots from these files. The MATLAB generated plots share the same file base name with a

*.pdf suffix (you can edit the post.m MATLAB script and change the plots from *.pdf to other formats).

Results

Included as part of this deliverable are the input file, the eight expected text output files, PDF files for each of the eight plots (average depth and length, and a CCDF plot for the depth and length at 10, 30 and 60 years) with the same base name as the text file with a *.pdf suffix, and a MATLAB script (called post.m) to post-process the output files to produce the *.pdf plot files.

BRISC Assessment

BRISC was developed as a research project to demonstrate the efficient solution of coupled multi-physics problems in parallel. As such, it was primarily a research code designed for knowledgeable developers without much support or focus to aid the end user. This means that there is no GUI, the input file syntax is extremely minimal and sparsely documented, there are no user guides or design documents and diagnosed errors may not be too meaningful. The BRISC development team is aware of these and other limitations but have not had sufficient time or funding to address them. Even with these usability issues, BRISC is a very capable multi-physics coupling framework that, combined with UQ capabilities, represents a very compelling computational engineering tool for engineers.

The xLPR test problem complements the BRISC model because we did not preclude supporting UQ type analysis when we did the BRISC design work. It was quite easy to drive the existing BRISC time integration loop by adding an outer loop over the number of aleatory realizations and an outermost loop over the number of epistemic realizations. This was accomplished with no substantive changes to our core Problem Manager module that drives the time integration. In addition, the epistemic realization loop was multi-threaded to facilitate parallel simulations on multi-core systems. Our transfer operators are very general and were extended to support transfers from the time-independent physics (epistemic and aleatory) to time-dependent physics for the crack routines. Even though the transfer operator details are not specified through the input deck, this is one possible extension that may be valuable to users.

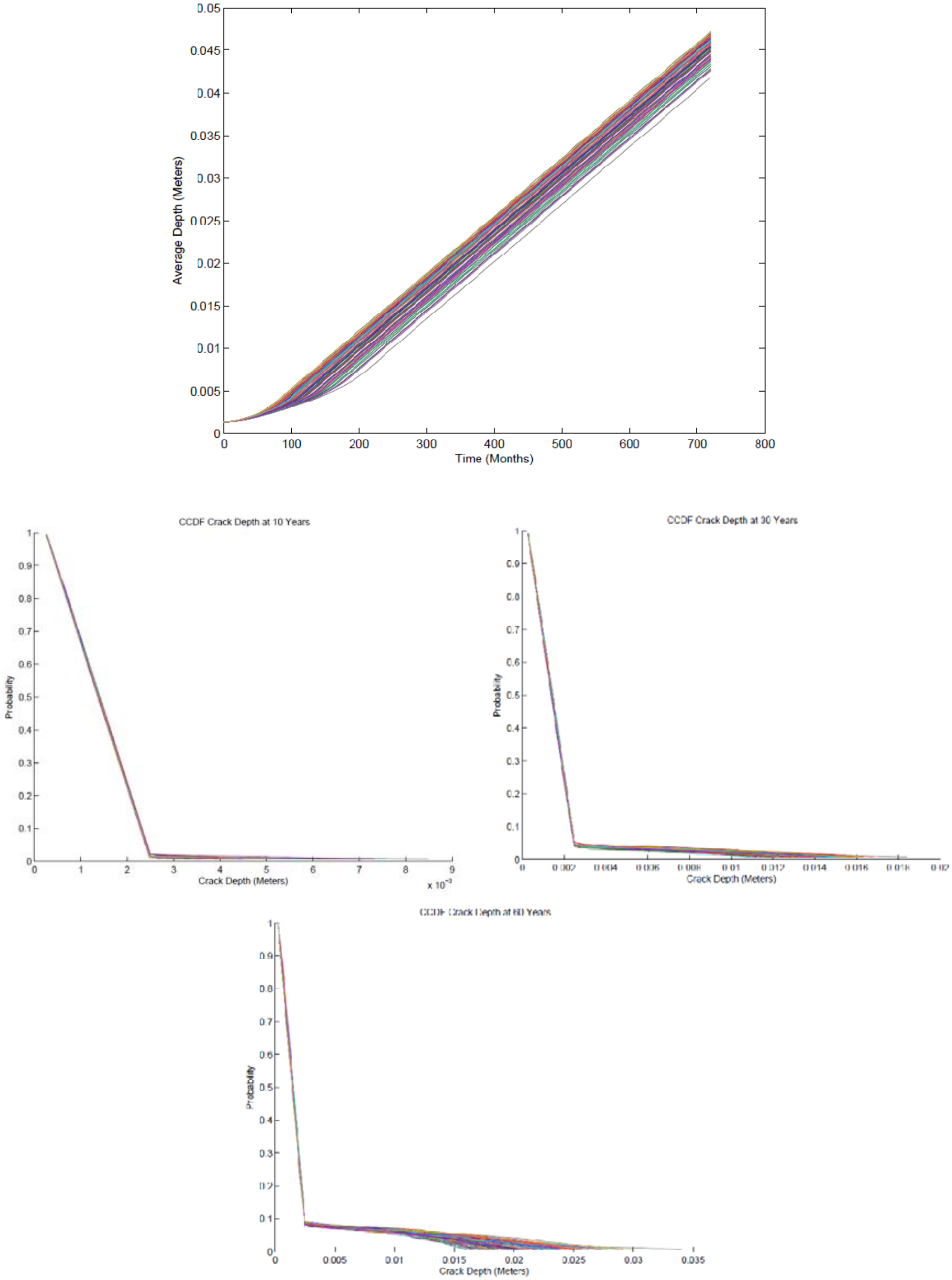


Figure A-3. Crack depth results from BRISC xLPR test case.

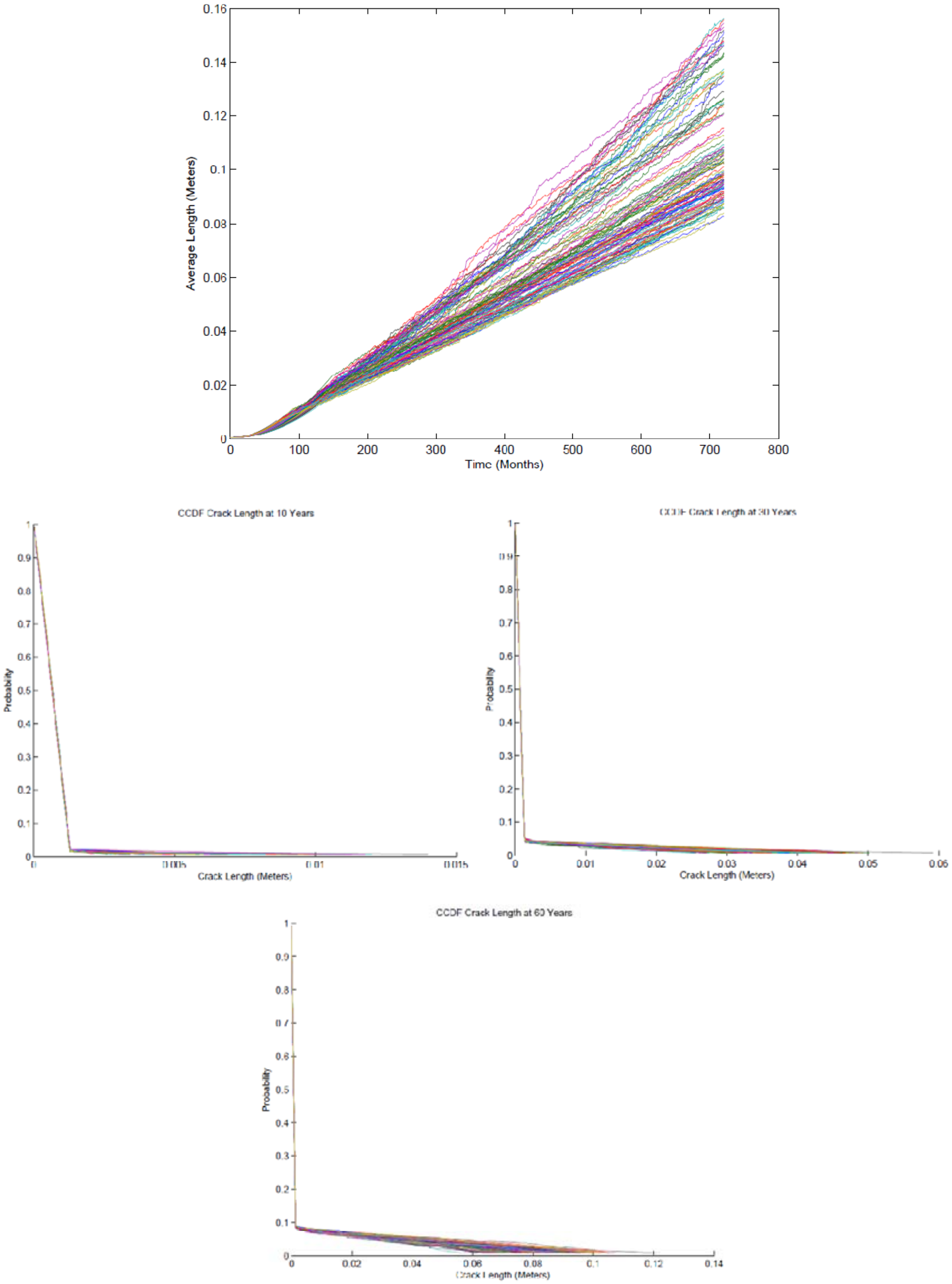


Figure A-4. Crack length results from BRISC xLPR test case.

Appendix A – References

[A-1] G. Whelan, K. J. Castleton, J. W. Buck, G. M. Gelston, B. L. Hoopes, M. A. Pelton, D. L. Strenge, and R. N Kickert. Concepts of a Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES). PNNL-11748, Pacific Northwest National Laboratory, Richland, Washington. October 1997.

[A-2] http://mepas.pnl.gov/FramesV1/index.stm

[A-3] M. A. Pelton, and B. L. Hoopes. "Multimedia-modeling Integration Development Environment." Proceedings of First International Conference on Brownfield Sites: Assessment, Rehabilitation and Development. September 2-4, 2002, Cadiz, Spain. Pp. 327-336. 2002.

[A-4] A. R. Shah, K. J. Castleton, M. A. Pelton, and B. L. Hoopes. "Framework for Risk Analysis in Multimedia Environmental Systems: Modeling Individual Steps of a Risk Analysis Process." Proceedings of International conference on Modeling, Simulation and Visualization Methods & International Conference on Algorithmic Mathematics and Computer Science. June 21-24, 2004, Las Vegas, NV. Pp. 38-44.

[A-5] http://mepas.pnl.gov/mepas/index.stm

[A-6] J. E. Babendreier, and K. J. Castleton. "Investigating Uncertainty and Sensitivity in Integrated, Multimedia Environmental Models: Tools for FRAMES-3MRA." Environmental Modeling & Software. Vol. 20, no. 8, pp. 1043-1055. 2005.

[A-7] Environmental Protection Agency (EPA). Multimedia, Multipathway, and Multireceptor Risk Assessment (3MRA) Modeling System. EPA530-D03-001a, U.S. Environmental Protection Agency, July 2003.

[A-8] http://www.epa.gov/athens/research/projects/3mra/

[A-9] M. S. Dortch and J. A. Gerald. "Army Risk Assessment Modeling System for Evaluating Health Impacts Associated with Exposure to Chemicals." Proceedings of First International Conference on Brownfield Sites: Assessment, Rehabilitation and Development. September 2-4, 2002, Cadiz, Spain. C. A. Brebbia, D. Almorza, and H. Klapperich (editors), WIT Press, Southampton, UK, pp. 353-361. 2002.

[A-10] M. S. Dortch and J. A. Gerald (2004). "Recent advances in the Army Risk Assessment Modeling System," in Brownfields, Multimedia Modeling and Assessment, G. Whelan (editor), WIT Press, Southampton, UK, pp. xxx-xxx. 2004.

[A-11] http://el.erdc.usace.army.mil/arams/

APPENDIX B. NRC xLPR ALPHA FRAMEWORK AND MODEL PROGRAM PLAN

The Computation Task Group developed an extensive pilot study program plan report for the Alpha xLPR framework development. This document is attached below and provides a brief description of the background, an overview of the framework and architecture, and a detailed description of the characterization of uncertainty. The program plan represents the basis for the xLPR pilot study code development and was developed by a collaborative effort by the computational task group. The program plan was not published and is provided here as a reference for the problem description in Section 2.0 and the framework description in Section 3.0. Sections 2.0 & 3.0 document the Beta development and Version 1.0 code and therefore any differences reflect changes from the alpha description presented below.

xLPR Alpha Model Program Plan

Computational Group

Background

10 CFR Part 50, Appendix A, 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 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. 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. Development of the xLPR methodology and the corresponding software tool will involve many challenging technical decisions, modeling judgments, and sensitivity analyses.

A pilot study will be completed to demonstrate the feasibility of the proposed NRC-industry cooperative process for developing a probabilistic software tool to address degradation mechanisms in piping system safety assessments. The pilot study will be a proof-of-concept effort to develop a simplified assessment tool for DM pressurizer surge nozzle welds, for which a considerable amount of publicly available information exists.

The analytical output of the pilot study will be a probabilistic assessment of surge nozzle DM weld leakage and rupture. The pilot study will provide relative, order-of-magnitude estimates of piping rupture probabilities; such analyses will identify areas requiring more focused attention in the long-term study.

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 be addressed in the long-term study.

The pilot study will be complete in the second quarter of 2010, and the long-term xLPR project will be complete by December 31, 2012.

Introduction

As part of the pilot study for the xLPR project, code framework and architecture development are required in order to assess the viability of the xLPR and modular code concepts. In addition, inputs and models relevant to the pilot study problem are required and are currently under development. The current schedule dictates that the architecture of the alpha xLPR framework be developed in parallel with the models and inputs development. Therefore, provisional inputs and models are needed for development of the framework architecture. The provisional models and input data may be replaced by other models once their choices have been finalized.

This document describes the program plan, logic, modules and architecture for the alpha xLPR code framework. It should be recognized that this is a living document and will be modified as the development of xLPR code continues. Changes will occur as the architecture, models (modules) and inputs are being developed. This initial version is meant only as a guide to allow the computational team to begin initial framework architecture development. Feedback from the development of the prototype framework using proxy xLPR modules and data will provide insights to the models and input groups in the parallel development of these components of the xLPR code.

Overall Flow

This section of the program plan provides an overview of the flow of the alpha xLPR framework. 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 prototype xLPR framework will handle uncertainties, followed by the process flow for xLPR. Note that the international system of units (SI units) is used throughout the xLPR code.

Implementation of Uncertainty and Handling Options

The appropriate treatment of uncertainty in analyses of complex systems is a topic of great importance and hence widespread interest. One pilot study goal is to ensure that consistent characterization and propagation of uncertainty can be implemented. This section focuses on uncertainty characterization, propagation of uncertainty through the analysis, and sensitivity study techniques that quantify the relationship between input and output uncertainties.

Characterization of Uncertainty

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

Daneshkhah [B-1] provides the following definitions for aleatory and epistemic uncertainty, which we will adopt here:

- 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.
- 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; expert judgments may be useful in its reduction. Epistemic, or internal, uncertainty reflects the possibility of errors in our 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.

Mosleh et.al. [B-2] claim: “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 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.

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 in consideration and the expert responsible for 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 of 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 risk level is still at an unacceptable level, one may want to treat epistemic uncertainty as aleatory. This allows a regulator or an operator to rank the epistemic uncertainty according to its contribution to the response uncertainty. A final use of the epistemic/aleatory construct is to identify the importance of different physical models.

Rather than having a standard PDF represent a random variable, different models can be used.[2] The end result is that the pilot study alpha framework must be able to handle four conditions that are listed according the classical method of modeling epistemic and aleatory uncertainty:

- A loop for the epistemic variables that are treated by standard sampling methods, i.e. PDFs
- An inner loop to the epistemic loop to handle aleatory uncertainties that are treated by standard sampling methods, i.e. PDFs
- An inner loop to the aleatory loop to handle aleatory uncertainties that are not treated by standard sampling methods, e.g. different models
- The ability for any variable to be assigned to any of the above three strategies

Currently, several 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 will only consider a probabilistic approach. Therefore, a probability distribution will be associated with each input parameter considered uncertain.[3]

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 may be introduced.

Finally, replacing continuous probability distributions with discrete distributions improves efficiency and allows any variable to be treated as either epistemic or aleatory. It is thus important to be able to (1) have the probabilistic characterization independent of the numerical model, and (2) create DPDs. The creation of DPDs is straightforward for

[2] 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.

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

standard distributions and is no different from standard Monte Carlo sampling. Non-standard distributions may require some development.

As a summary:

- Uncertainty in parameters will be handled using probability distributions, defined by the input group.
- Parameter properties will be classified as epistemic, aleatory or constant by being stored in a specific location.
- Parameters will have to be easily moved from one category to the other.
- It will be possible to correlate some inputs.
- Redefining distributions should be an easy task

Preferably, the probabilistic framework will be decoupled from the numerical (deterministic model) so that any change in the uncertainty treatment will not affect the physical model.

Propagation of Uncertainty

The uncertainty in the input will be propagated through the model using sampling based methods and, possibly, by numerical integration procedures. 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.

Based on the resources needed to perform a single deterministic run, it is possible to estimate the maximum number of runs that can be performed for an analysis. It is likely that this number will be used in order to maximize the sample size that is computationally practicable.

The nature of the output will also influence the sampling or discrimination used. The output of interest is an extremely low probability of rupture. It is thus likely that a "brute force" sampling will generate many runs without any rupture. The probability of rupture will be determined by a few runs and will, in consequence, be poorly estimated.

It seems thus appropriate to use some sort of importance sampling in order to cover with greater accuracy the regions (in the hypercube of the input space sampled) where the pipe may rupture. However, these regions are not known upfront and may have to be determined in an initial step. Finally it is critical to avoid the mistakes of the past in developing probabilistic fracture mechanics codes. Although many lessons have been learned one of the critical ones was a general lack of QA. In the pilot study it must be demonstrated that:

- Extremely low probabilities of rupture can be calculated
- A methodology for calculating the distribution of this probability can be developed
- The sources of the uncertainty in this distribution can be identified and quantified

- The appropriate documentation for the adoption of a model, input, PDF, etc. exists and is controlled

The level of QA for the entire process is suitable for use in a regulatory environment

The proposition for the pilot study involves a three prong approach. Each is briefly described in the next sections.

Object Oriented Commercial Framework

In this approach COTS software is used to construct two nested loops.

On the first step, a Monte Carlo approach will be used. The outer loop, capturing the epistemic uncertainty, would correspond to a sample size of n_E. This sample will be constructed using LHS technique [B-3]. The inner loop, capturing the aleatory uncertainty, would correspond to a sample size of n_A. This time a SRS will be used, as the stratification imposed by LHS is not desirable.

Once the first set of results will be available, they will be analyzed to determine which region is more likely to lead to failure of the pipe. Importance sampling can be used to cover the critical regions greater precision.

As a summary:

- Loops have to be defined in order to run the deterministic physical model several times. In the pilot study, up to three loops have been implemented although only two will be used in a first set of tests
- The sampling methods should include LHS and SRS.
- The structure should again be independent of the deterministic physical model in order to be able to change it easily without any impact on the model itself.

Object Oriented Open Source Framework

In this approach the only difference is that the commercial software is replaced by open source coding. There are several reasons for performing such studies, including i) the coding is available to everyone without licensing fees, and ii) the user is not limited to the commercial code limitations nor the possibility of losing software support.

OOC/Legacy Framework

Because all previous experience indicates that either an OOC or an OOOS framework will be, in a relative sense, very slow running, a third option exists to combine an OOC approach to the classic legacy code approach. Using an OOC approach provides the QA and control that older legacy codes did not have. In addition OOC frameworks can provide estimates of low probability events, especially when combined with LHS or discrete methods. However, their ability to perform extremely low probability calculations in hours in a personal computer environment remains a concern. Therefore, a third approach will combine the OOC framework with a computationally more efficient code that combines models and inputs from the OOC framework to efficiently produce rupture probabilities.

Sensitivity Analysis Techniques

The last step of uncertainty treatment involves analyzing the results and drawing conclusions. Several uncertainty and sensitivity analysis techniques are presented in [B-4]. In this pilot study, and supposing that traditional sampling techniques are used, it is recommended to use well known techniques. Sensitivity analysis refers to the determination of the uncertainty in the analysis result that derives from the uncertainty in analysis inputs. This corresponds essentially to a statistical analysis of the set of output resulting from the sample.

The separation of aleatory and epistemic uncertainty provides the basis for representing the output of interest as a set of CCDFs (see Figure B-1) at every time step.

Each CCDF (representing the effect of aleatory uncertainty) gives an answer to Kaplan and Garrick [B-5] risk triplet questions:

Q1. What can happen in the future?

Q2. How likely it is to happen?

Q3. What is the response if it happens?

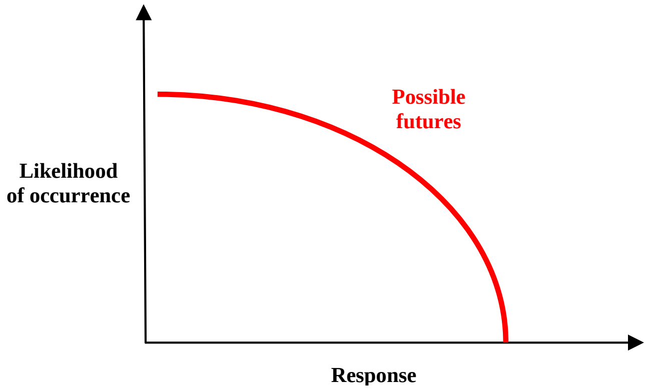


Figure B-1. Complementary cumulative distribution function illustration.

A set of these CCDFs (representing the effect of epistemic uncertainty) represents the state of knowledge on this “risk”.

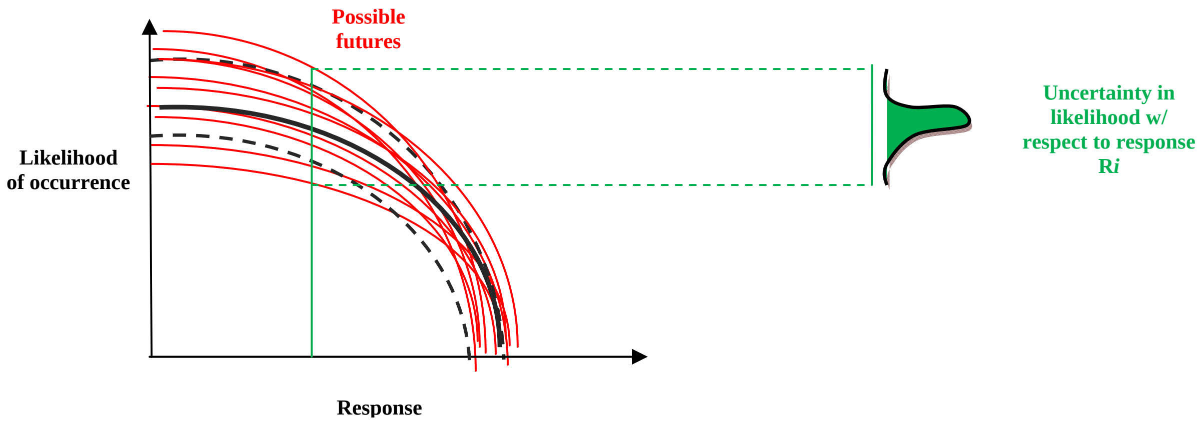


Figure B-2. CCDF illustration.

Each CCDF can be integrated to give a "representative value" (usually the mean with other possibilities including median or another quantile). Then, this set of "mean values" can be represented as a set of time-dependent expected values (Figure B-2).

In the pilot study, it seems reasonable to use (at least for a first step) simple and reliable methods. Scatterplots of output versus input remains one of the simplest and most useful techniques.

In order 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 seem the most appropriate. However, given the potentially large uncertainties, the known non-linear nature of the problem, and the possible fracturing of the response space (hypercube in the LHS terminology), it is necessary to also examine more sophisticated methods (such as non-linear maximum likelihood or non parametric methods). As rank regression often produces better results than linear regression with raw data, for essentially the same computing cost, analyses will be performed to obtain Partial Rank Correlation Coefficients (PRCCs) and SRRCs. The coefficient of determination for the regression model (R^2) will indicate the quality of the regression and consequently the quality of the sensitivity analysis. The approach that seems the most appealing graphically is:

- Calculating time-dependent PRCCs on time-dependent results: this shows graphically the evolution of the importance of the input parameters
- Estimating the importance of some parameters in a stepwise fashion (i.e., using stepwise rank regression) for non time-dependent parameters and at specific time step for time-dependent parameters

As a summary:

- For uncertainty analysis, CCDFs, time-dependent expected results and classical statistics (mean, standard deviation, quantile) will be obtained for the outputs of interest

- Sensitivity analysis will be performed using scatterplots, PRCCs, SRRCs and R^2 values
- Sensitivity and uncertainty analysis methods will be applied outside of the main model, once all data are available.
- Non-linear maximum likelihood analysis will be studied to determine if it is needed
- Other more sophisticated techniques based on nonparametric regression and other procedures could be applied if R^2 results are low.

xLPR Process

The alpha xLPR code framework overall flow chart is shown in Figure B-3, consisting of initial condition inputs, definition of variables, time history development and the time looping structure. Note that the looping structure shown represents only one option for uncertainty propagation as discussed in the previous section[4]. The time history development is described below.

After the initial conditions are defined, the time loop, shown in Figure B-4, begins. For each time increment, the code will check whether the analysis is beyond the predefined time period (described in Time History Section) for the analysis. If it is, the time loop is exited, if not, it will continue.

If a pre-emptive mitigation is to be performed at this time increment, the code will apply mitigation and continue.

The crack growth module is then used to calculate the crack growth increment for any existing cracks using the criteria. Within the crack growth module, the instantaneous loads, including the WRSs, are used along with the crack and pipe geometry to calculate stress intensity factors. These stress intensity factors are then used with the crack growth model to calculate the crack growth increment. For simplicity, only PWSCC will be considered in the pilot study. Each of the existing crack sizes will be updated, and if any SC has reached 100% through wall, it will transition to a TWC. In addition, a check will be made to determine if any SCs or TWCs coalesce. If they do, they will be combined.

Next, if the crack initiation model dictates that a crack initiates in this time step, a single, surface breaking crack will be placed in the model with the appropriate size and location. Note that, in the xLPR process time loop, the growth module precedes the initiation module to accommodate potential growth of cracks existing at time=0.

[4] The initial condition inputs are not directly discussed in this program plan. However, for each module, inputs and outputs are listed.

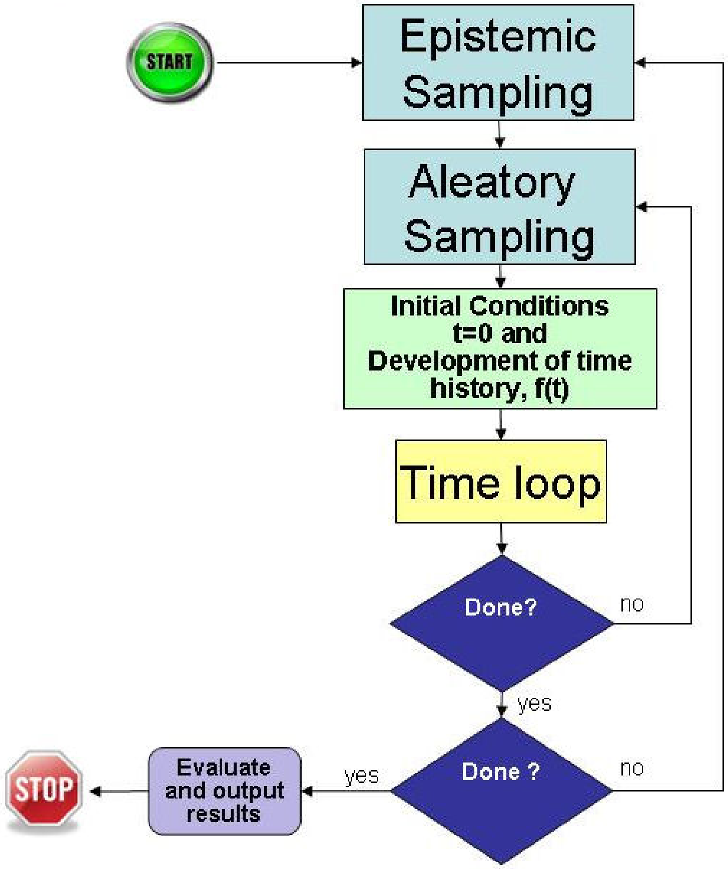

START
Epistemic Sampling
Aleatory Sampling
Initial Conditions t=0 and Development of time history, f(t)
Time loop
Done?
no
yes
Done ?
no
yes
Evaluate and output results
STOP

Figure B-3. Overall flow chart for xLPR.

Next, the crack stability module determines if any existing cracks have reached a critical size. At any time increment, TWCs or SCs may exist in the analyses. Note that in the final version of xLPR complex cracks (CC) may also be considered, but are neglected in the alpha xLPR code. For existing TWCs, if the instantaneous crack size is larger than the critical crack size, a double ended break (severance of the pipe) is assumed. For existing SCs, if net-section collapse failure is predicated at the operating loads, the crack transitions to a TWC. In this case, if the resultant TWC length is greater than the critical TWC length, a double ended break is assumed. The size of this opening at failure is recorded and the time loop is exited.

If a TWC is not critical, the leakage module is used to determine the level and acceptability of the leakage. First, a leakage calculation is performed for the TWC. If the calculated leak rate is greater than pre-defined LOCA rates, the leak rate and crack opening area are recorded. The time is then incremented and the analysis continues.

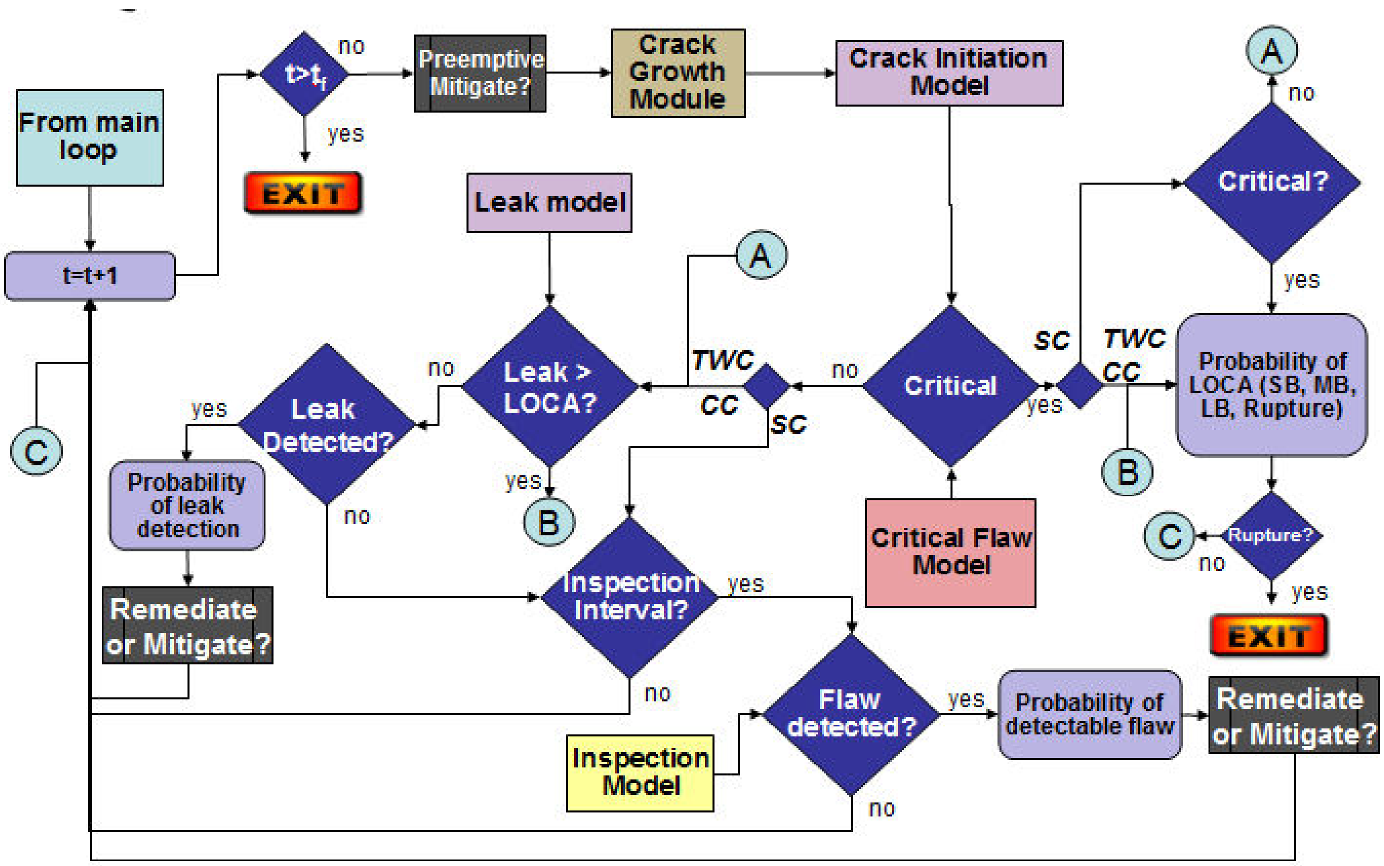


Figure B-4. xLPR process (time loop) details.

If the calculated leak rate is less than the predefined LOCA limits, it is compared against the input leak detection limit. If it is greater than this limit, the detected leak (crack size and leak rate) is recorded. The time is incremented and the analysis continues. If the leak is not detected or if net section collapse is not predicted for a SC, and if the current time corresponds to an inspection interval, an inspection takes place. The PND and the probability of repair (POR) are attached to the crack for each inspection interval and the final probabilities with no inspections are modified by this number to demonstrate the effect of inspections. This procedure always assumes a flaw found by inspection is completely removed, i.e., there are no mitigation or remediation options.

If the time does not correspond to an inspection interval, the time loop is incremented and the analysis continues.

Time History Development

A large portion of the xLPR computational effort is devoted to a time loop through the life of a plant. For a single realization of all random numbers, it is possible to develop a timeline of occurrence for all time-dependent variables before entering the time looping structure of the code. For instance, the loading history, the occurrence of transients and earthquakes, and the initiation of PWSCC cracks are all time based events and can be set into arrays as a function of time. However, the time-dependent growth of PWSCC cannot be determined prior to the time looping since the location and possible crack interaction are random. Therefore, for those variables that are xLPR process independent, a time line history is developed before the time loop phase of the program begins. For the alpha xLPR code, the following items will be set during the time history development:

- Time to PWSCC initiation
- All loads including transients but excluding crack face pressure

Geometry and Material Properties

It is envisioned that for the final version of the xLPR code, complete piping systems can be analyzed. However, for the alpha xLPR code, only one location will be analyzed. This location can be a single weld, or a section of pipe, but will only include one circumferential plane. To analyze an entire piping system with the alpha xLPR code, individual runs for each location are required and the probabilities summed to obtain the total failure probability. Typically, the location with the highest failure susceptibility is chosen, and the failure probability is conservatively estimated by multiplying the worst-case failure probability by the number of girth welds in the system.

The alpha xLPR framework is specifically focused on a pressurizer surge nozzle DM weld that is susceptible to PWSCC. Therefore, the geometry and material property options in this version will be limited.

It is assumed that the major pipe geometry features, i.e., diameter and wall thickness, are fixed. A side-view schematic of the surge nozzle geometry is given in Figure B-5. From MRP-216 [B-6], a survey of nine operating power plants suggests that surge nozzle pipe is NPS 14 with a 15-inch outer diameter at the DM weld. The wall thickness at the weld is typically 1.58 inches. These values will be used for alpha xLPR code.

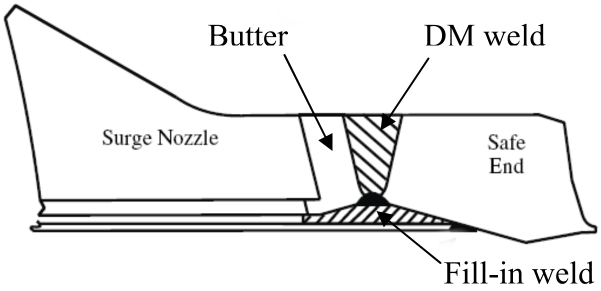


Figure B-5. Surge nozzle geometry schematic.

The material properties used will be specific to the pressurizer surge nozzle. It is envisioned that for the final version of the code, a material property database will be developed and the code will draw the necessary properties from that database. For the models chosen in the alpha xLPR code, both strength and fracture toughness properties are required. For the base materials, the yield and ultimate strength and the elastic modulus are required. The materials' constitutive response is assumed to follow the Ramberg-Osgood relationship:

$$\frac{\varepsilon}{\varepsilon_o} = \frac{\sigma}{\sigma_o} + \alpha\left(\frac{\sigma}{\sigma_o}\right)^n \qquad \text{(Eq. B-1)}$$

where

ε_o = σ_o/E
σ_o = reference stress, typically the yield strength, (MPa)
E = elastic modulus, (MPa)
α, n = curve fit parameters

Fit parameter data has been captured for the original form of the Ramberg-Osgood model, given by

$$\varepsilon = \frac{\sigma}{E} + \alpha\left(\frac{\sigma}{F}\right)^n \qquad \text{(Eq. B-2)}$$

where F and α are related as:

$$\alpha = \frac{\sigma_o^{\,n-1} E}{F^n} \qquad \text{(Eq. B-3)}$$

For the weld, in addition to the strength properties, fracture toughness properties are required. It is assumed that the materials' J-R curve can be represented with power law relationship and is given by:

$$J = J_i + C(\Delta a)^m \qquad \text{(Eq. B-4)}$$

where,

J = value of J at a given value of crack growth (Δa) (kJ/m^2),
J_i = value of J at crack initiation (kJ/m^2),
Δa = amount of crack growth (m),
m = curve fit exponent,
C = curve fit coefficient

The material properties assumed for the alpha xLPR code are given in Table B-1. The other material properties needed for the specific models, i.e., crack growth (WRSs and crack growth parameters), are given in the later sections of this program plan.

Table B-1. Initial pilot study material properties (at 300C).

Material	**Property**	**Mean**	**Stdev**	**Distribution type**	**Correlation**
A516 Gr 70	Yield strength, MPa	228.5	21.7	Lognormal	0.4866
	Ultimate strength, MPa	519.9	28.7	Lognormal	
	Elastic modulus, GPa	186.3	0	Constant	N/A
	F	915.2	82.3	Lognormal	-0.8565
	n	4.322	0.538	Lognormal	
TP304	Yield strength, MPa	172.5	36.5	Lognormal	0.6066
	Ultimate strength, MPa	453.7	53.2	Lognormal	
	Elastic modulus, GPa	177.1	0	Constant	N/A
	F	563.8	43.6	Lognormal	-0.6047
	n	4.298	0.571	Lognormal	
Alloy 182	Yield strength, MPa	372	90.1	Lognormal	0.5
	Ultimate strength, MPa	583	58	Lognormal	
	Elastic modulus, GPa	203.1	0	Constant	N/A
	$J_{Ic,}$ kJ/m^2	570.7	360	Lognormal	0.9
	C	292.34	150	Lognormal	
	m	0.62	0.1	Lognormal	N/A

Loads

For the alpha xLPR code, all of the loads will be input in a separate module. The load module will input the following:

- Load components as given in Table B-2
- ID WRSs (σ_{0WRS}) and X_c as described in the weld section below

and output the following:

- Membrane and bending stress components (see the Stress section below)
    - Pressure
    - Deadweight
    - Normal thermal (including stratification)
    - SSE
- WRSs coefficients as described in the Welding Residual Stress section below.

Table B-2. Average loads for surge nozzle location [6].

	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

Normal Operating Loads

The static normal operating load contributions are the result of pipe pressure, temperature, deadweight, and through-thickness weld residual stresses (see Welding Residual Stress section below). The axial stress components due to internal pressure, *P*, deadweight and normal thermal loading are calculated as:

$$\sigma_{oP} = \frac{F_P}{A},\ \sigma_{oDW} = \frac{F_{DW}}{A},\ \sigma_{oNTE} = \frac{F_{NTE}}{A} \qquad \text{(Eq. B-5)}$$

where:

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

$$F_P = P\left(\frac{\pi D_i^2}{4} + f_{cracked}\left(\frac{\pi\left(D_o^2 - D_i^2\right)}{4}\right)\right) \qquad \text{(Eq. B-6)}$$

where:

D_o = outer diameter (m)
D_i = inner diameter (m)

$f_{cracked}$ = crack area divided by pipe cross-sectional area. This term will be time-dependent and will have to be updated as the percentage of area cracked increases.

The global bending stresses resulting from deadweight or thermal expansion can either be specified directly or via their bending moment and torque components. If loads are input, an effective moment, M_{eff}, is calculated using Equation B-5.

$$M_{eff} = \sqrt{M_y^2 + M_z^2 + \left[\frac{\sqrt{3}}{2} M_x\right]^2} \qquad \text{(Eq. B-7)}$$

where,

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

The global bending stress, σ_B, is then calculated using the elastic relationship given by

$$\sigma_B = \frac{M_{eff} R_o}{I} \qquad \text{(Eq. B-8)}$$

where

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

For the analysis location of interest, the spatial distribution of stress must be calculated. The axial membrane stress will be constant around the circumference, but the bending stress can be calculated at the maximum location (Eq. B-8), and then scaled according to the azimuthal location. The scaled bending stress, $\sigma_{B\text{-}L}$, is given by

$$\sigma_{B-L} = \sigma_B \cos(\phi) \qquad \text{(Eq. B-9)}$$

where

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

For simplicity, in the alpha xLPR code, both primary and secondary stress components are treated as primary. This assumption will be revisited in the final xLPR code.

For the alpha xLPR code, the normal operating loads are taken from MRP-216 [B-6]. As shown in Figure B-6 the axial membrane stress does not vary significantly between the plants investigated.

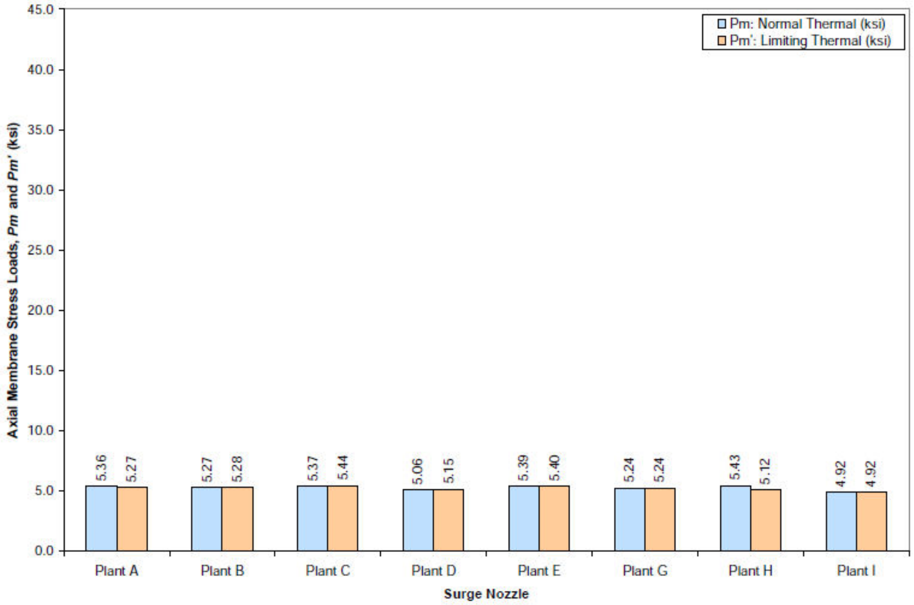


Figure B-6. Pressure, deadweight, and normal thermal axial membrane stress for the surge nozzle [B-6].

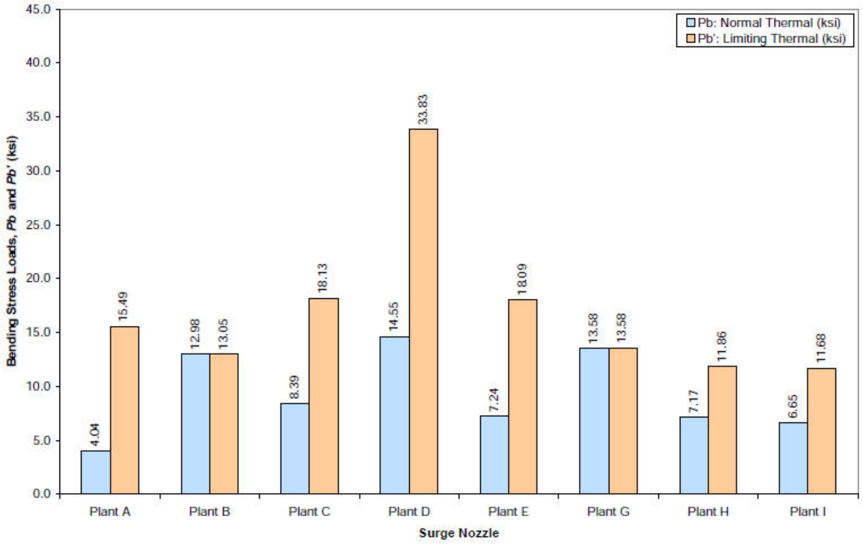


Figure B-7. Pressure, deadweight, and normal thermal global bending stress for surge nozzle [B-7].

The total global normal operating bending stresses from MRP-216 [B-6] are given in Figure B-7. As shown in this figure, the total bending stress ranges from 27.6 MPa (4 ksi) to 227.5 MPa (33 ksi) in the limiting case. According to MRP-216 [B-6], the limiting thermal loads for all surge nozzles are the maximum thermal load including all effects, such as piping expansion and thermal stratification. It is also noted that the limiting thermal load sometimes occurs during the normal operating condition (i.e., no thermal stratification).

Transient Loads

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

Thermal Stratification

A load condition that is unique to pressurizer surge nozzles is thermal stratification [B-6]. Thermal stratification occurs in the surge line of pressurized water reactors, due to the temperature difference between the pressurizer and the hot leg, which are connected by the surge line. The stratification produces a bending load that is carried through the surge nozzle safe end region. The thermal stratification loads during normal operation are approximately equal to thermal expansion loads without thermal stratification. 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 larger. The limiting thermal loads for all surge nozzles are reported as the entire maximum thermal load including all other effects, such as piping expansion.

Earthquake

Earthquake loads can be of high magnitude and therefore will be considered in the alpha 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 normal operating conditions. Since the event time for an earthquake is significantly shorter than the planned time increment for this initial xLPR version, it is assumed that the earthquake will occur within one time step. Also, since this initial version is focusing on PWSCC behavior, and neglecting contributions from fatigue, the low cycle fatigue aspect of earthquakes will be ignored.

The earthquake input loads are the maximum membrane and bending loads as well as the frequency of occurrence. The number of cycles that occur per earthquake is another input, but will be neglected in the alpha xLPR code. Note that this omission is slightly nonconservative, and will need to be considered for the final xLPR code. Since fatigue is not being considered, the earthquake loading will only affect crack stability (see Crack Stability below). It is typical for earthquake loading to be added to the normal operating loading (superposition) when conducting crack growth/stability calculations.

Load Summary

For the surge nozzle problem, typical loads can be taken from MRP-216 [B-6]. In that report, nine plants were investigated and the average loads from these plants are given in Table B-2. These loads will be used in the alpha xLPR code.

Time Loop

For the alpha xLPR code, it will be assumed that the analysis starts at time zero, and continues until the user input final time. This time may correspond to the end of design life, the end of extended life, or some other time specified by the user. For the alpha xLPR code, the assumed time increment will be 1 month. In the final xLPR code, there may be a need to refine or coarsen this number based on the models chosen.

Crack Initiation

Crack initiation models are an important driver to predicting probability of rupture in piping systems. Flaws that can lead to 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.). The following sections describe the initiation models for the alpha xLPR code.

Pre-existing Defects

Typical nuclear piping welds contain defects from the welding process that can act as initiation sites for subcritical cracking. Destructive examinations have been conducted to characterize the flaw frequency and depth distribution as a function of weld process and type. Since these flaws are often embedded within the weld, and since the initial pilot study is to focus on PWSCC, it was decided to make pre-existing defects not active in the alpha xLPR.

PWSCC Initiation – Poisson's Arrival Rate Model

Stress corrosion cracking in the primary water environment is a function of material susceptibility, stress magnitude and water chemistry. Despite extensive research on PWSCC of Alloy 600, primarily for steam generator tubing, no consensus has formed on the mechanism of PWSCC initiation, although the phenomenology has been fairly well characterized [B-7]. There is a strong influence of temperature with an activation energy of 40 to 52 kcal/mole. Stresses near yield are needed for initiation, and the time to initiation is sensitive to the stress level. The stress dependence of the time to initiation is frequently modeled as a power law, i.e., $\sim\sigma^4$. Furthermore, cold work has an important accelerating effect on initiation. In the case of control rod drive mechanism nozzles, this suggests that fabrication processes such as machining or surface grinding could have important consequences on susceptibility to cracking. Studies with steam generator tubing have shown that grain boundary carbides improve resistance to stress corrosion cracking (SCC). Even when stress level, fabrication, water chemistry, and other variables are carefully controlled, it is found that initiation is a statistical process [B-8].

Since there is great difficulty in developing a detailed model for PWSCC crack initiation, an arrival rate approach for crack initiation is being assumed in the alpha xLPR code. It is assumed that the probability of crack initiation, *P*, follows a Poisson's distribution given by:

$$P\left(N(t+\tau)-N(t)=k\right)=\frac{e^{-\lambda\tau}\left(\lambda\tau\right)^{k}}{k!} \qquad \text{(Eq. B-10)}$$

where:

$N(t+\tau)-N(t)$	=	number of events in time interval $(t, t+\tau)$.
τ	=	time interval
λ	=	arrival rate
k	=	number of cracks

This distribution allows the probability of multiple crack initiations to be developed as a function of time. For example, using an arrival rate of 0.01 cracks/year, the probability of occurrence of up to five cracks is shown in Figure B-8.

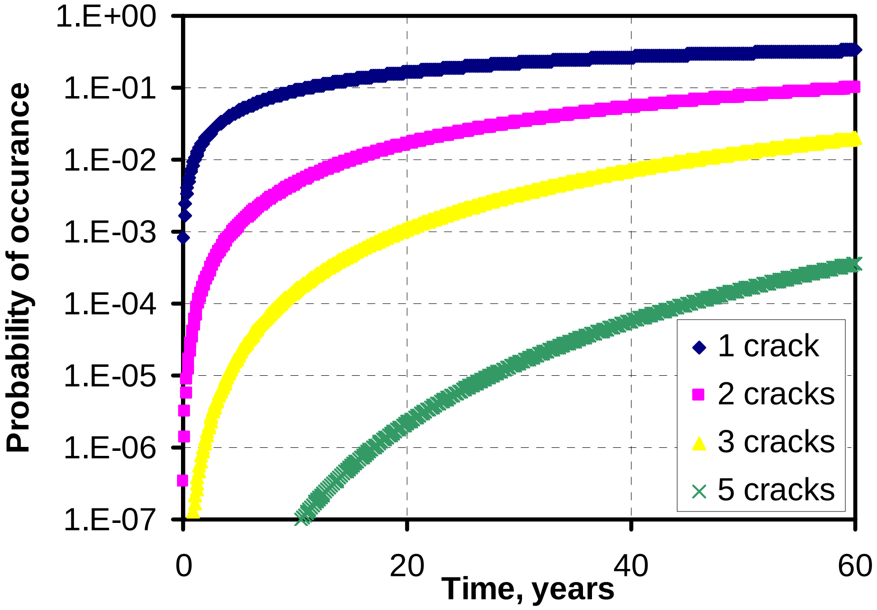


Figure B- 8. Crack initiation model with arrival rate of 0.01 cracks/year.

The Poisson arrival rate model discussed above is stress independent and therefore can only provide the time of crack initiation. The location of crack initiation is stress dependent and will be discussed in the Crack Placement Section below. Further, PWSCC initiation is clearly temperature dependent. However, for the pilot study, it is assumed that the arrival rate input is fixed at the operating temperature.

Using the equations and constraints above, the time at which cracks initiate can be determined by randomly sampling the probability of occurrence. This will allow the generation of a PWSCC initiation time history, which can be fully generated in the initial condition phase outside of the time loop. Within the time loop, when the current time matches an initiation time, a crack is placed into the analysis as described in the Crack Placement Section below.

The inputs and outputs for the PWSCC initiation model used in the alpha xLPR code are given in Table B-3.

Table B-3. PWSCC crack initiation inputs and outputs.

Model	Inputs	Outputs
Crack Initiation	Arrival rate	Time to initiation

Crack Placement

As stated above, the time history of PWSCC initiation is developed prior to the time looping, and therefore, for a particular realization, the number of SCCs and the times they initiate are known at the start of the simulation. A crack placement module is then used to locate the crack within an analysis. The criterion behind this model is stress based. Data by Amzallag [B-9] suggest that PWSCC initiates only when the stress nears the as-welded yield strength of the material. However, the applicability of this data to actual service conditions is unknown and should be considered random. Therefore, for the alpha xLPR code, it is assumed that for any location around the circumference:

- If the local ID stress is above the as-welded strength, this location will always be considered a crack initiation location.
- If the local ID stress is less than zero, this location will never be considered a crack initiation location.
- If the local stress falls between zero and the as-welded strength, a random sample (uniformly distributed) will determine if this location is to be considered for crack initiation.

Using the stress criteria above and a random number, the percentage of the circumference where crack initiation can occur can be determined. Within a realization, as the cracks initiate, they are placed randomly into the allowable circumferential area. The distance between two cracks will always be greater than two times the deepest SC depth. If a newly initiated crack resides in a location where a prior crack already exists, it is placed into another location. If the allowable circumferential area is filled with cracks, no further cracks are allowed in this realization. The table of inputs and outputs for the crack placement module is given in Table B-4.

The size of initiated defect is sampled from the user input length and depth distribution. For the alpha xLPR code, it will be assumed that initiating flaws will be 1.5 mm deep (Coefficient of variance (COV) = 5%) and have a length of 3 mm (COV = 5%). The flaw is assumed surface breaking and semi-elliptical in shape.

Table B-4. Crack placement inputs and outputs.

Model	Inputs	Outputs
Crack placement	$\sigma_o + \sigma_B$	New crack location
	Existing and new crack depths	
	Existing and new crack lengths	
	Existing crack locations	
	Uniform random number	

Bookkeeping

In these multiple crack analyses, proper bookkeeping of the crack information is essential. For the pilot study, the following information must be tracked for each crack:

- Initial crack length and depth
- Crack location (center of crack)
- Current crack length and depth
- Crack type, i.e., SC, TWC
- Current PND
- Current leak rate
- Other initiation mechanisms

Other Initiation Mechanisms

For the final version of xLPR, it is envisioned that fatigue, Intergranular Stress Corrosion Cracking (IGSCC), flow enhanced corrosion, and other piping specific initiation mechanisms will be included. However, these models will not be included in the alpha xLPR code.

Crack Growth

The crack growth module in the initial version of xLPR calculates a crack growth increment for the chosen cracking mechanism. The crack growth is controlled by the stress intensity factor and its relationship is fit to existing experimental data.

Stress Intensity 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 exceeds the materials critical value, crack advance occurs. For subcritical cracking, the process of crack advance is linked to the applied values of the stress intensity though curve fits with extensive experimental data. The stress intensity is not only a function of the loading, but also the global component geometry and local crack shape (see Table B-5). Over the years, many researchers have developed K-solutions for circumferential and axial surface and TWCs in cylindrical vessels based on finite element parametric analyses. In all cases, the K-solutions were developed using the principle of superposition. The principle of superposition states that the solution for a multiple load case is equal to the sum of the results from the individual load cases. These individual cases can then be combined to describe a more complicated loading history.

Table B-5. Inputs and outputs for the stress intensity model.

Model	Inputs	Outputs (SC)	Outputs (TWC)
Stress intensity	Pipe ID	Kdeep	KTWC
	Pipe wall thickness		
	Crack type - SC, TWC		
	Crack depth		
	Crack length	Ksurface	
	WRS		
	Operating Loads		
	Transient loads		

For the alpha xLPR code, the Anderson K-solutions for both SCs and TWCs in cylinders were used. Anderson's K-solutions for a circumferential SC on the inside pipe diameter are given in [B-10]. The solutions in this report were generated for R/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 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 [B-10]. For the case of a circumferential semi-elliptical SC, the crack growth at both the deepest (90 degrees) and surface (0 degrees) locations are calculated and applied to the initial crack sizes. The finite length SC 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 becomes a problem when trying to predict crack behavior from initiation to failure. Therefore, several assumptions are made. First, it is assumed that the influence functions can be extrapolated from a/t=0.8 to a/t=1.0. Secondly, a solution by Chapuliot [B-11] is used for a/t approaching 0. Linear interpolation is used between these values and Anderson's results at a/t = 0.2.

In addition to the elliptical SC results, Anderson also generated K solutions for a/c = 0 (infinitely long SC). Since long SC K-solutions are currently not available, it is assumed that for SCs with c/a greater than 32, the K solution at the free surface is equal to the K-solution at c/a = 32 and at the deepest point, the K-solutions equals that of the K-solution for a/c = 0. This assumption is 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 TWC in a pipe are given in [B-12]. 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 the TWC, however; only the G_0, G_1 and G_5 influence functions are available. In [B-12], the TWC K solutions were curve fit and the coefficients were presented for R/t values of 1, 3, 5, 10, 20, 60, and 100. These coefficients are 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 TWC are calculated, and then averaged to get the K-solution for through-wall-crack growth.

Loads for Crack Growth

Since the alpha xLPR code will only consider PWSCC, the loads defined for crack growth will be limited to this case. Since PWSCC grows under static load conditions, only the loads present during normal operation will be considered for subcritical crack growth. Therefore, the loads (membrane and global bending) considered (see Load section above) include:

- Pressure
- Deadweight
- Thermal expansion (including normal stratification)
- WRSs

Using this list and the definitions from the previous section, the total stress used for crack growth is given as:

$$\sigma_{CG} = \left(\sigma_{0DW} + \sigma_{0NTE} + \sigma_{0P} + \sigma_{0WRS}\right) + \sigma_{1WRS}\left[\frac{x}{t}\right] + \sigma_{2WRS}\left[\frac{x}{t}\right]^2 + \sigma_{3WRS}\left[\frac{x}{t}\right]^3 + \sigma_{B-L}$$

(Eq. B-11)

The WRS model for this initial version of xLPR is described in the next section.

Welding Residual Stress

WRS is one of the major drivers to SCC and must be included for proper predictions of subcritical crack growth. Issues such as weld repairs, grinding, etc., will all impact the probabilities of leakage and possible rupture. In the alpha xLPR code, the effects of local repair will be neglected and the WRS will be assumed to be axis-symmetric. In order to account for uncertainty in the WRS, the ID stress and through thickness location where the WRS crosses through zero will be considered random, see Figure B-9. The user is asked in input the ID stress (mean, standard deviation, and distribution type, i.e., uniform, normal, lognormal, etc.) and values for the term X_c (mean, standard deviation, and distribution type), where X_c is in terms of a fraction of the pipe wall (x/t). The code takes the sampled values (Table B-6) for inner diameter stress and X_c and fits a 3rd order polynomial with the following form:

$$\sigma_{WRS} = \sigma_{oWRS} + \sigma_{1WRS}\left[\frac{x}{t}\right] + \sigma_{2WRS}\left[\frac{x}{t}\right]^2 + \sigma_{3WRS}\left[\frac{x}{t}\right]^3 \qquad \text{(Eq. B-12)}$$

where

σ_{nWRS} = Curve fit coefficients,
x = distance from ID (m), and
t = wall thickness (m).

In addition to the inner diameter stress and X_c, the following restraints are used:

- The area under the curve must equal zero (stress equilibrates through thickness)
- The stress on the outer diameter is a uniform random number between 0 and the 0.5*σ_{0WRS}.
- If $X_c > 0.4$, the outer diameter stress has the opposite sign as σ_{0WRS}, if $X_c < 0.4$, the outer diameter stress has the same sign as σ_{0WRS}.
- If $X_c = 0.5$, the stress is linear through the wall with outer diameter stress equal but opposite in sign to σ_{0WRS}.

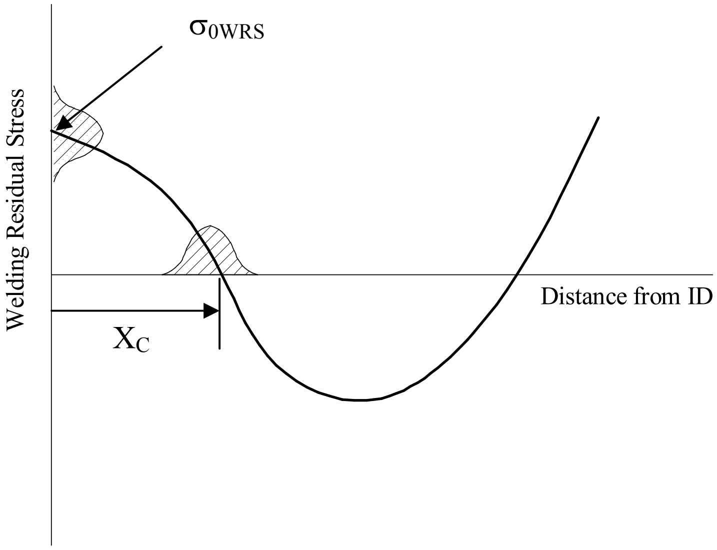


Figure B-9. WRS distribution schematic.

Table B-6. WRS model input and output.

Model	Inputs	Outputs
Weld residual stress	σ_{0WRS}	σ_{1WRS}
		σ_{2WRS}
	X_c	σ_{3WRS}

In order to test this model, data from Harris [B-13] on large diameter stainless steel boiling water reactor (BWR) coolant pipe was analyzed and predicted with this model. One hundred samples were taken and the results are shown in Figure B-10. The data points were taken from [B-13] and the curves were generated with this model using the calculated distribution[5] for σ_{0WRS} and $X_{c.}$ The distribution of curves shown captures the data relatively well.

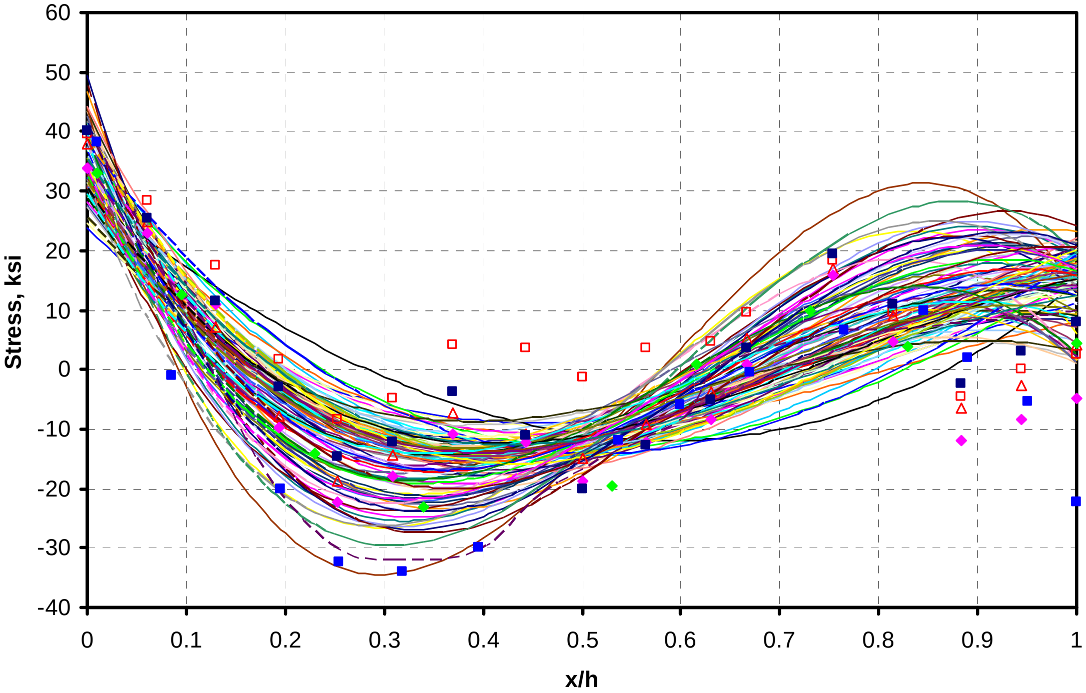


Figure B-10. Weld residual stress predictions.

PWSCC Growth for Each Active Crack – MRP-115 Model

For the pilot study, PWSCC of DM butt welds is the only subcritical cracking mechanism that will be considered. Over the recent years, researchers have conducted laboratory experiments to measure the growth rate of PWSCC in Alloy 82/182 DM welds. The collective experimental data has been gathered and analyzed by EPRI in MRP-115 [B-14]. In this effort, a distribution of crack growth rates as a function of temperature, alloy,

[5] For the data, σ_{0WRS} mean = 35.43 ksi, σ_{0WRS} stdev = 4.51 ksi, X_c mean = 0.146, X_c stdev = 0.035

and crack orientation were developed. These data were fit to the following functional form:

$$\dot{a} = \exp\left[-\frac{Q_g}{R}\left(\frac{1}{T}-\frac{1}{T_{ref}}\right)\right]\alpha f_{weld}\, f_{alloy}\, f_{orient}\, K^{\beta} \qquad \text{(Eq. B-13)}$$

where

$\dot{a}$ = crack growth rate at temperature T in m/s

Q_g = thermal activation energy for crack growth = 130 kJ/mole

R = universal gas constant = 8.314 x 10^{-3} kJ/mole-K

T = absolute operating temperature at the crack location in K

T_{ref} = absolute reference temperature to normalize data = 598.15 K

α = power law constant = 9.83 x 10^{-13}

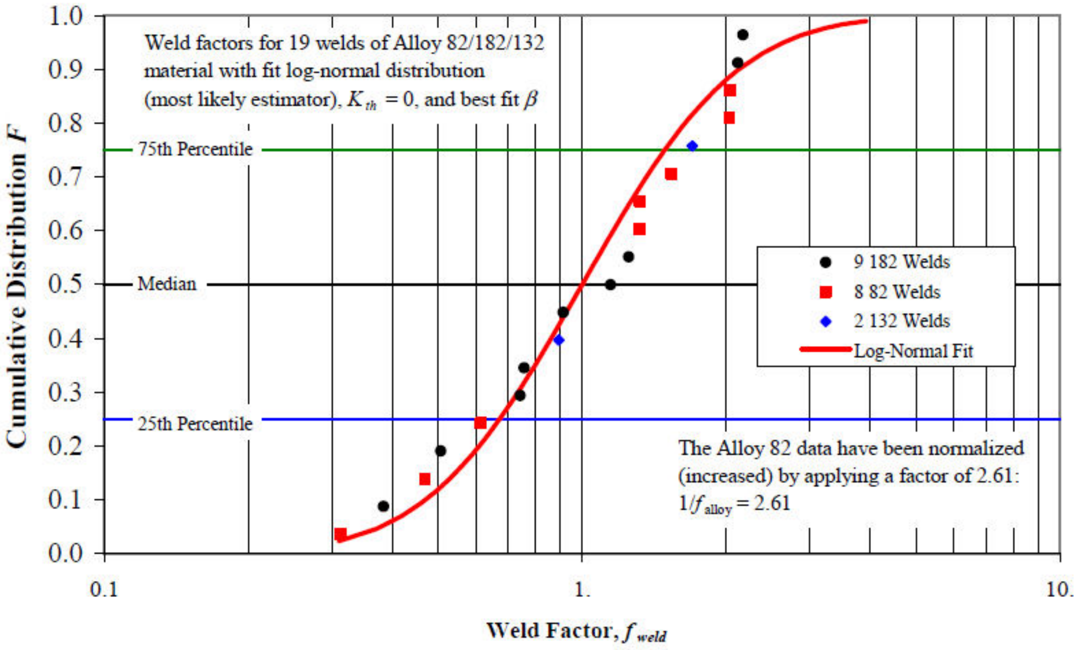
f_{alloy} = 1.0 for Alloy 182

f_{orient} = 1.0 for growth parallel to dendrite solidification direction

K = crack stress intensity factor, MPa-m$^{0.5}$

β = exponent = 1.6

The uncertainty in the crack growth rate was characterized by f_{weld} and the distribution is shown in Figure B-11. For this initial version of the xLPR code, the Eq. B-13 crack growth model with the FigureB-11 distribution will be used. Along with the constants given above, the inputs and outputs for this model are given in Table B-7.



Weld factors for 19 welds of Alloy 82/182/132 material with fit log-normal distribution (most likely estimator), K_{th} = 0, and best fit β

The Alloy 82 data have been normalized (increased) by applying a factor of 2.61: $1/f_{alloy}$ = 2.61

Figure B-11. Weld factor for Alloy182 crack growth rates per [B-14].

Table B-7. Inputs and outputs for crack growth model.

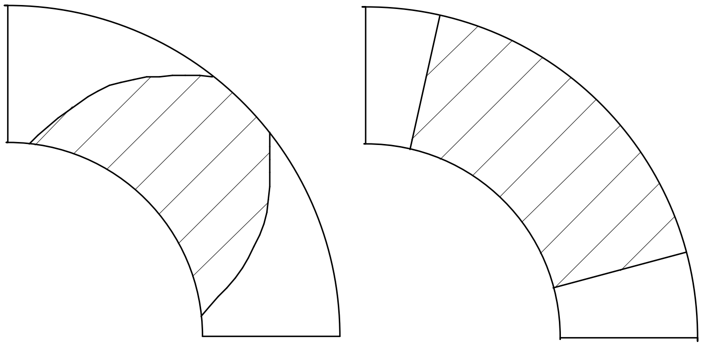
Model	Inputs	Outputs
Crack Growth	Crack growth coefficient	Updated crack depth
	Crack growth exponent	
	K_{deep} (or K_{TWC})	
	$K_{surface}$	Updated crack length
	Temperature	

Other Crack Growth Mechanisms

For the final version of xLPR, it is envisioned that fatigue, IGSCC, flow enhanced corrosion, and other piping specific initiation mechanisms will be included. However, these models will not be included in the alpha xLPR code.

Transition from SC to TWC

As an internal SC begins to penetrate the wall thickness, only a small breach of the pressure boundary is first observed. For an internal SC that becomes a leaking crack, the crack length on the OD is much smaller than that on the ID due to the previous crack growth. In idealized through-wall behavior, the crack front runs radially, and the actual OD crack length is longer than that on the ID, see FigureB-12.



(a) Penetrating crack (b) Idealized TWC

Figure B-12. Illustration of penetrating cracks and idealized TWC.

For the crack shapes in Figure B-12a, general stress intensity solutions do not exist. There are flat plate solutions in the WinPraise manual [B-15], and cylinder solutions for R/t=8 that were developed through the NURBIM program [B-16] but the accuracy of those solutions relative to the low R/t values for primary piping is unknown. Therefore, for the alpha xLPR code, it will be assumed that, as the SC penetrates the wall thickness, an idealized TWC with the same crack area will be formed. The assumption is conservative from a crack growth/stability standpoint, but may be non-conservative from a leakage perspective.

Coalescence

Crack coalescence is an important part of guaranteeing that the cracks that develop in this code are representative of the long SCs found in service. For circumferential SCs, as shown in Figure B-13, when the distance between the SCs becomes less than two times the deepest SC depth, the cracks will coalesce. The depth of the new crack is equal to the deepest SC and the length is equal to the sum of the lengths of each crack plus the distance between them. The inputs and outputs for this criterion are given in Table B-8 and based on Section XI, Article IWA-3000 of the ASME Boiler and Pressure Vessel code.

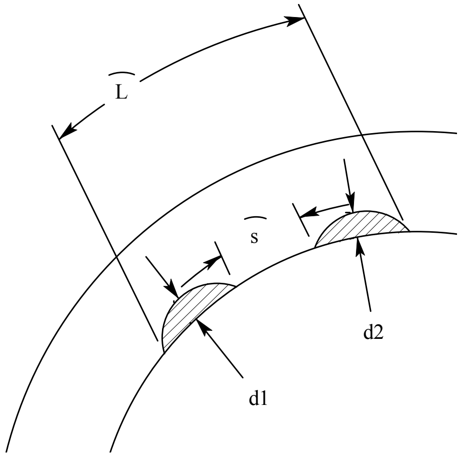


Figure B-13. Surface crack coalescence.

Another case of coalescence is when two TWCs interact. Realistically, this case has a low probability of occurrence, but if two TWCs, with total leakage less than the technical specification limit, are present, they will coalesce when the crack tips touch.

There is also a possibility that a TWC may interact with a SC. In this case, if the crack tips touch, a CC is formed. However, as with the penetrating crack, proper stress intensity solutions for CCs are unavailable. For the alpha xLPR code, it will be assumed that if a TWC and a SC interact, a TWC will be formed with a crack area equal to the sum of the two interacting crack areas.

Table B-8. Input and outputs for the crack coalescence model.

Model	Inputs	Outputs
Coalescence	Crack locations	Updated number of cracks
	Crack lengths	Updated crack lengths
	Crack depths	Updated crack depths
	Number of cracks	Updated crack locations

Crack Stability

The behavior of TWCs and SCs in nuclear grade piping has been the subject of many experimental programs conducted by the NRC. Many reports have been written with the majority of the past research summarized in [B-17]. This report deals with 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 [B-18] that recommends several methodologies for handling cracks in DM welds. The models chosen for the alpha xLPR code follow those recommendations.

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

$$\sigma_{CS} = \left(\sigma_{0DW} + \sigma_{0NTE} + \sigma_{0P}\right) + \sigma_{B-L} \qquad \text{(Eq. B-14)}$$

Surface Crack – Net Section Collapse

Since the pilot version of xLPR is focused on a high toughness Alloy 182 DM weld, it is appropriate to assume that the failure of a SC (low crack tip constraint) will be driven by net-section collapse. The inputs and outputs for the analyses are given in Table B-9. The methodology for net section collapse of circumferential SCs is described in detail in [B-19] and the level of uncertainty as compared to experiments is illustrated in Table B-10. For DM welds, the largest unknown in net-section collapse analyses is what material properties to use. Analyses have been conducted that suggest that a combination of the two base metal properties is appropriate for making critical SC predictions [B-20]. These analyses also suggest that if the crack is located near the stainless steel material, the stainless flow properties control the collapse.

Table B-9. Inputs and outputs for critical surface crack model.

Model	Inputs	Outputs
Critical Surface Crack	Diameter	Critical crack size
	Wall thickness	
	Crack type	
	Crack length	
	Crack depth	Critical bending moment
	Material flow stress (or yield and ultimate)	
	Operating loads	
	Transient loads	

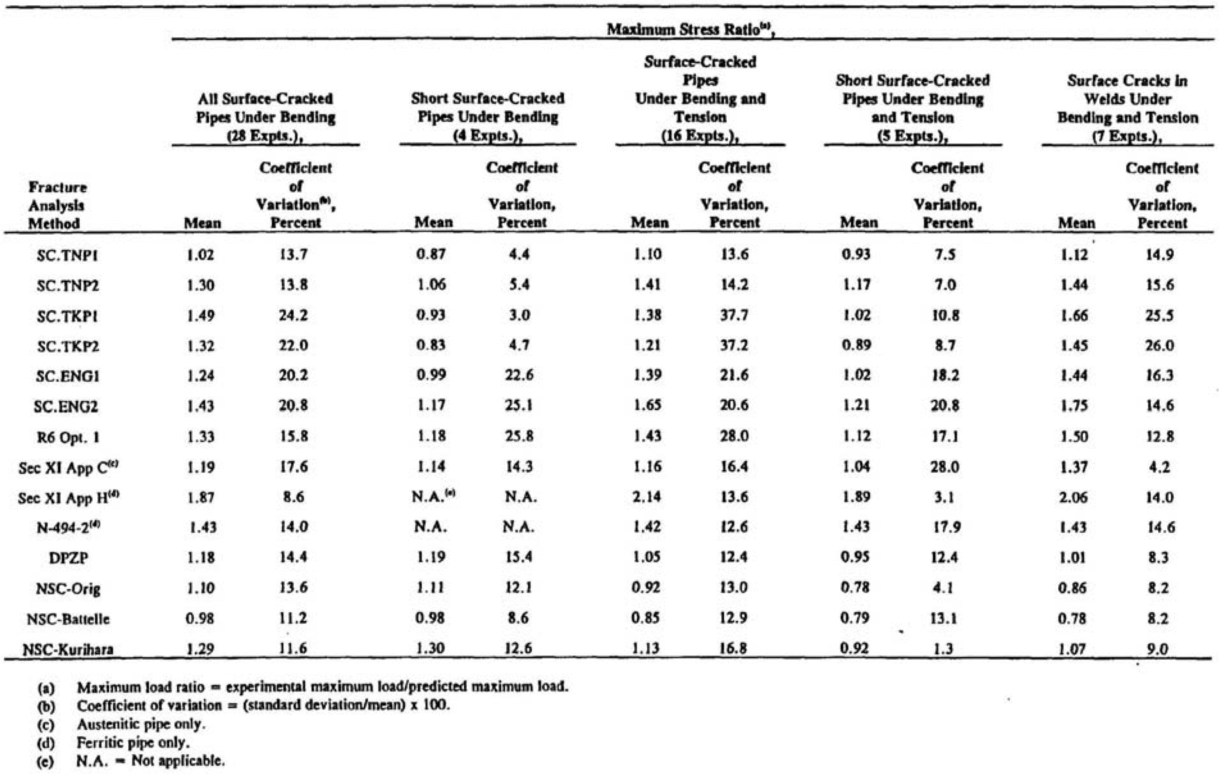
Since the net-section collapse analysis is relatively simple and inexpensive from a computational standpoint, for the alpha xLPR code, the maximum bending load will be predicted after each crack growth increment using the net-section collapse analysis given the applied axial loads (using superposition) and pipe and crack geometry. Two checks on operating loads are to be made

- Normal operating or normal + transients (excluding SSE)
- Normal + SSE

If the bending loads (from the first case) are greater than the calculated net section collapse bending load, the SC will transition to an idealized TWC with the ID length equal to that of the critical SC.

Another method of using the critical crack criteria is to run the analyses outside of the time loop and create a table of critical crack lengths and depths at each combination of loads. The code would then use this table within the time loop to assess criticality by comparing the current crack size to that of the critical crack size. Since the net-section collapse analysis is simple, for the alpha xLPR code, the first option will be utilized.

Table B-10. Uncertainty in maximum load predictions for surface cracked pipe [B-17].



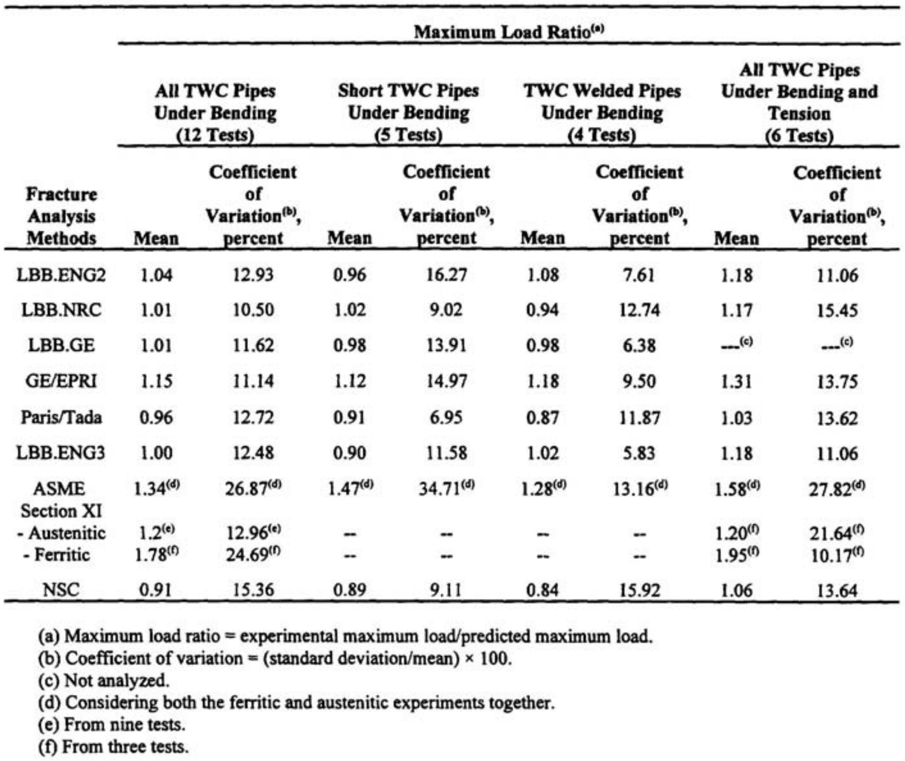
	Maximum Stress Ratio[a],									
	All Surface-Cracked Pipes Under Bending (28 Expts.),		Short Surface-Cracked Pipes Under Bending (4 Expts.),		Surface-Cracked Pipes Under Bending and Tension (16 Expts.),		Short Surface-Cracked Pipes Under Bending and Tension (5 Expts.),		Surface Cracks in Welds Under Bending and Tension (7 Expts.),	
Fracture Analysis Method	Mean	Coefficient of Variation[b], Percent	Mean	Coefficient of Variation, Percent	Mean	Coefficient of Variation, Percent	Mean	Coefficient of Variation, Percent	Mean	Coefficient of Variation, Percent
SC.TNP1	1.02	13.7	0.87	4.4	1.10	13.6	0.93	7.5	1.12	14.9
SC.TNP2	1.30	13.8	1.06	5.4	1.41	14.2	1.17	7.0	1.44	15.6
SC.TKP1	1.49	24.2	0.93	3.0	1.38	37.7	1.02	10.8	1.66	25.5
SC.TKP2	1.32	22.0	0.83	4.7	1.21	37.2	0.89	8.7	1.45	26.0
SC.ENG1	1.24	20.2	0.99	22.6	1.39	21.6	1.02	18.2	1.44	16.3
SC.ENG2	1.43	20.8	1.17	25.1	1.65	20.6	1.21	20.8	1.75	14.6
R6 Opt. 1	1.33	15.8	1.18	25.8	1.43	28.0	1.12	17.1	1.50	12.8
Sec XI App C[c]	1.19	17.6	1.14	14.3	1.16	16.4	1.04	28.0	1.37	4.2
Sec XI App H[d]	1.87	8.6	N.A.[e]	N.A.	2.14	13.6	1.89	3.1	2.06	14.0
N-494-2[d]	1.43	14.0	N.A.	N.A.	1.42	12.6	1.43	17.9	1.43	14.6
DPZP	1.18	14.4	1.19	15.4	1.05	12.4	0.95	12.4	1.01	8.3
NSC-Orig	1.10	13.6	1.11	12.1	0.92	13.0	0.78	4.1	0.86	8.2
NSC-Battelle	0.98	11.2	0.98	8.6	0.85	12.9	0.79	13.1	0.78	8.2
NSC-Kurihara	1.29	11.6	1.30	12.6	1.13	16.8	0.92	1.3	1.07	9.0

(a) Maximum load ratio = experimental maximum load/predicted maximum load.
(b) Coefficient of variation = (standard deviation/mean) x 100.
(c) Austenitic pipe only.
(d) Ferritic pipe only.
(e) N.A. = Not applicable.

TWC – LBB ENG2

There are several estimation schemes that have been developed for analyzing the elastic-plastic fracture behavior of circumferential TWCs in nuclear piping materials. For overall behavior, the LBB.ENG2 method has been shown to accurately predict the maximum moment for TWC pipe experiments, see Table B-11. Therefore, this method will be used in this initial pilot study version of xLPR. The LBB.ENG2 estimation method proposed by Gilles and Brust [B-21] for evaluating the J-integral of cracked tubular members subjected to combined tensile and bending loads is used for assessing the stability of TWCs. 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 inputs and outputs for this method are shown in Table B-12. 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 [B-17].

Table B-11. Uncertainty in TWC maximum moment predictions.



	Maximum Load Ratio[a]							
	All TWC Pipes Under Bending (12 Tests)		Short TWC Pipes Under Bending (5 Tests)		TWC Welded Pipes Under Bending (4 Tests)		All TWC Pipes Under Bending and Tension (6 Tests)	
Fracture Analysis Methods	Mean	Coefficient of Variation[b], percent	Mean	Coefficient of Variation[b], percent	Mean	Coefficient of Variation[b], percent	Mean	Coefficient of Variation[b], percent
LBB.ENG2	1.04	12.93	0.96	16.27	1.08	7.61	1.18	11.06
LBB.NRC	1.01	10.50	1.02	9.02	0.94	12.74	1.17	15.45
LBB.GE	1.01	11.62	0.98	13.91	0.98	6.38	---[c]	---[c]
GE/EPRI	1.15	11.14	1.12	14.97	1.18	9.50	1.31	13.75
Paris/Tada	0.96	12.72	0.91	6.95	0.87	11.87	1.03	13.62
LBB.ENG3	1.00	12.48	0.90	11.58	1.02	5.83	1.18	11.06
ASME Section XI	1.34[d]	26.87[d]	1.47[d]	34.71[d]	1.28[d]	13.16[d]	1.58[d]	27.82[d]
- Austenitic	1.2[e]	12.96[e]	--	--	--	--	1.20[f]	21.64[f]
- Ferritic	1.78[f]	24.69[f]	--	--	--	--	1.95[f]	10.17[f]
NSC	0.91	15.36	0.89	9.11	0.84	15.92	1.06	13.64

(a) Maximum load ratio = experimental maximum load/predicted maximum load.
(b) Coefficient of variation = (standard deviation/mean) × 100.
(c) Not analyzed.
(d) Considering both the ferritic and austenitic experiments together.
(e) From nine tests.
(f) From three tests.

Table B-12. Inputs and outputs for TWC stability model.

Model	Inputs	Outputs
Critical TWC	Diameter	Critical crack size
	Wall thickness	
	Crack type	
	Crack length	
	Crack depth	Critical bending moment
	Material flow stress	
	Material toughness	
	Operating Loads	
	Transient loads	

The LBB.ENG2 method is somewhat more complex and computationally expensive than the net-section collapse method due to the iterative process of the J-T methodology. Therefore, it is more efficient to run this algorithm outside of the time loop to generate a table of critical crack lengths as a function of applied loads. The instantaneous TWC lengths in the time loop portion of the analyses can then be checked against this table to assess crack stability.

When instability is reached, either by operating loads or transient loads, the code assumes a double ended break is formed. It is recorded and the program exits the time loop. For emergency and seismic loads, the code records the failure, but does not exit the time loop. The effects of SSE loads on the probability of rupture are developed as a post processing step.

Complex Crack

Although complex shaped cracks, i.e., TWC and SC combinations, may control the rupture behavior, the criterion for failure of these flaws in DM welds is not fully developed. In fact, it is difficult at this point to choose an appropriate model. Therefore, for the alpha xLPR code, CCs will be ignored. As stated earlier, if a SC and TWC interact, an equivalent area TWC will be generated.

Leak Rate Model

In order to calculate the fluid flow through the predicted TWCs in these pipes, a leak rate model is required. The leak rate model utilizes the instantaneous COD and crack morphology parameters, with a thermal-hydraulic model to predict the mass flow through the cracks. Extensive research in past NRC funded projects has developed these methodologies.

Crack Opening Displacement – GE/EPRI

As with the J-estimation schemes described above for maximum moment predictions, similar schemes have been developed for predictions of COD. These methodologies are described in detail in [B-22]. As shown in Table B-13, the original GE/EPRI method for calculating COD produces the best prediction of the experimental results and will be used in the alpha xLPR code. The uncertainties in these predictions are also given in this table.

Table B-13. Uncertainty in COD predictions.

Fracture Analysis Method	Experimental/Predicted COD	
	Mean	COV, %
Original GE/EPRI	1.01	72.8
Battelle-modified GE/EPRI	1.02	86.5
Tada/Paris	2.96	146

The prediction of COD will be made prior to the leak rate calculation using the applied instantaneous loads (assuming no WRS, see Eq. B-9), which may include transient conditions. At this point, the effect of weld residual stress on the COD will be ignored since that effect is not well understood. However, from a qualitative point of view, excluding WRS for this analysis will probably be conservative, since the residual stress field through thickness crosses into compression at the mid thickness which would reduce the COD. Table B-14 contains the inputs and outputs for this model.

Table B-14. Inputs and outputs for the COD model.

Model	Inputs	Outputs
Crack opening displacement	Diameter	Crack opening displacement
	Wall thickness	
	Crack length	
	Operating Loads	
	Transient loads	
	Material flow stress (stress-strain)	

Crack Morphology Parameters

As a fluid passes though a crack, significant pressure losses occur when the fluid changes direction along the flow path. The crack-face surface roughness (μ), the number of turns (n) along the flow path, and the actual crack path length (K) are the crack morphology parameters that need to be characterized in order to determine these pressure losses. For the standard subcritical crack mechanisms in these pipe welds, the crack morphology parameters have been measured [B-23, B-24] from detailed micrographs of cracks removed from service. The distribution of the parameters is shown in Table B-15.

Table B-15. Mean and standard deviation of crack morphology parameters.

Crack	Corrosion Fatigue	IGSCC	PWSCC – Base	PWSCC – Weld

Morphology Variable	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev
μ_L, µm	8.814	2.972	4.70	3.937	10.62	9.870	16.86	13.57
μ_G, µm	40.51	17.65	80.0	39.01	92.67	65.26	113.9	90.97
$n_{L,}$ mm^{-1}	6.730	8.070	28.2	18.90	8.043	2.043	5.940	4.540
K_G	1.017	0.0163	1.07	0.100	1.060	0.095	1.009	0.011
K_{G+L}	1.060	0.0300	1.33	0.170	1.327	0.249	1.243	0.079

Since the pressure losses across the crack are dependent on the surface roughness the fluid experiences, there is a relationship between the pressure losses and the tightness of the crack. For example, for a very wide crack, the fluid passes over the crack and is influenced by the global roughness of the crack face. However, for tighter cracks, the global roughness creates a tortuous flow path, and the fluid is influenced by the local grain roughness and number of turns. Empirical models have been developed, some of which have been supported by fluid mechanics calculations that describe this relationship. Figure B-14 illustrates one of these models [B-25].

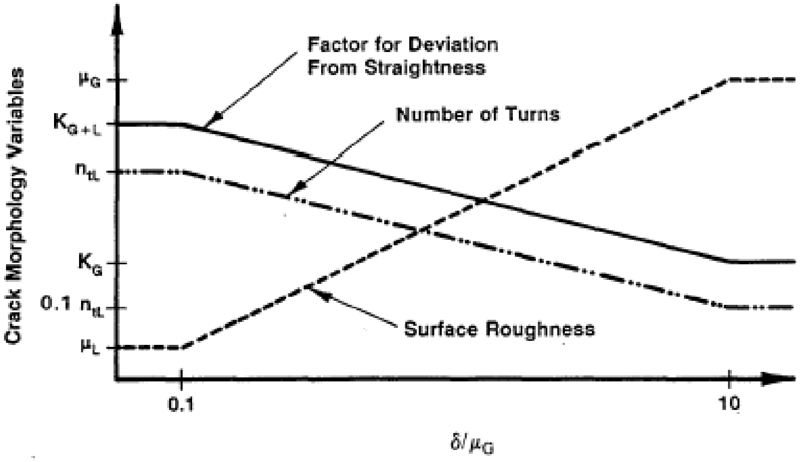


Figure B-14. Crack morphology variables versus normalized COD [B-25].

In this initial version of the xLPR code, the relationship between the crack opening and crack morphology parameters will be ignored.

Leak Rate Model – SQUIRT

A review [B-26] of existing thermal-hydraulic models 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 (CCs) determined from experimental data of Henry. As part of NRC funded research, this methodology was encoded in a computer code called SQUIRT.

SQUIRT is a computer program that 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 subcooled liquid, and transitions to two-phase as it flows 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 as it is discharged. 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. This module solves the flow equations associated with compressible gas flow.

SQUIRT has been validated against experiments ranging from flat plate to pipe tests with IGSCC flaws [B-27], with the distribution of predictions shown in Figure B-15.

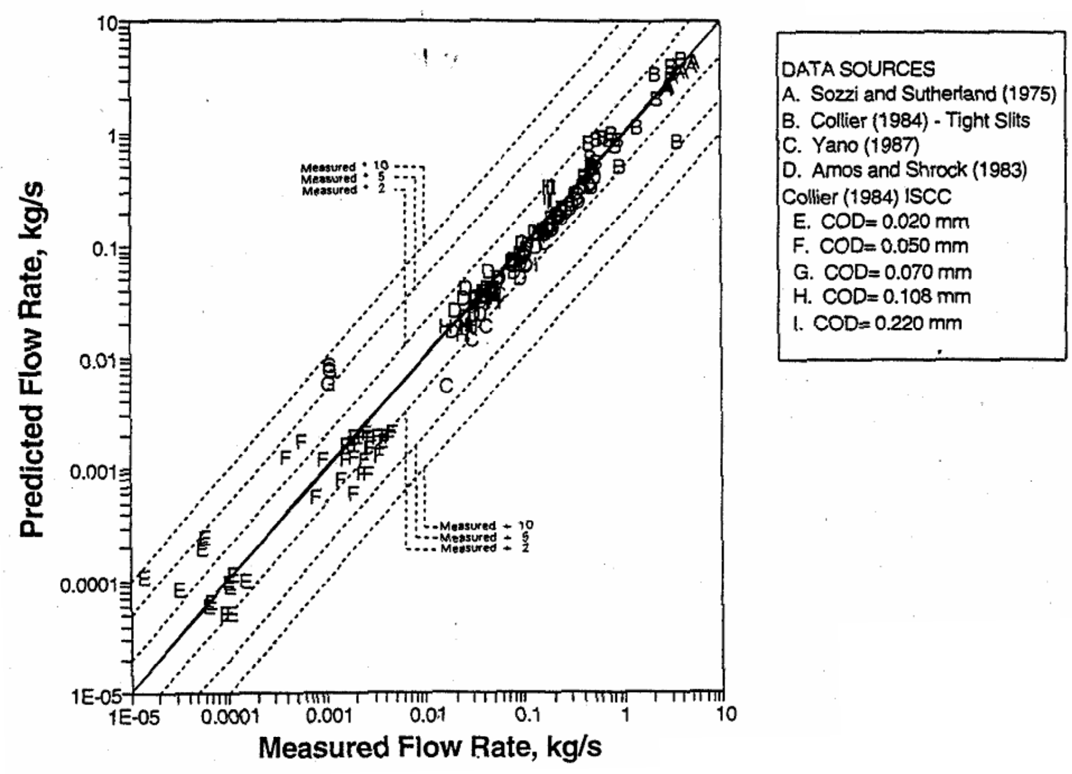


Figure B-15. Uncertainty bands for predictions with SQUIRT.

For the alpha xLPR code, an older version of the SQUIRT code will be implemented. This version does not contain the COD-crack morphology model and the single phase flow models. For the initial pilot study, it will be assumed that the crack morphology parameters are variable, and follow the distribution for PWSCC welds given in Figure B-14. The input and output for this model are given in Table B-16.

Table B-16. Inputs and outputs for leakage model.

Model	Inputs	Outputs
Leakage	Crack opening displacement	Leak rate
	Crack length	
	Crack opening shape	
	Fluid pressure	
	Fluid temperature	
	Crack morphology parameters	

In the alpha xLPR code, it will be assumed that if the leakage is found, i.e., greater than the technical specification limit (which is an input distribution), the data is recorded. However, the time loop is not exited until the pipe ruptures. The effects of the leak detection are calculated after the execution of the routine.

Inspection

One of the large drivers in predicting low probability of rupture is the proper handling of credit for in-service inspections. While there are many ways to handle the influence of inspection, the following is the plan for the alpha xLPR code. The debate on how the final version of xLPR will incorporate inspections is left for the Models Task Group.

First, an inspection schedule will be input. This schedule will include the time period (months) inspections are to occur relative to the start of the analysis.

Next, the probability of detection (POD) per inspection interval will be input. Based on the recent effort by EPRI [B-28], the POD for DMs welds will be represented by the following functional form:

$$POD\left(\frac{x}{t}\right)=\frac{e^{\beta_1+\beta_2\left(\frac{x}{t}\right)}}{1+e^{\beta_1+\beta_2\left(\frac{x}{t}\right)}} \qquad \text{(Eq. B-15)}$$

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 by:

β_1 = 2.7076, standard error = 0.2085
β_2 = 0.0031, standard error = 0.0045

The POD is a representation of the POD as a function of crack depth. From these curves, the PND can be calculated as (1-POD). Also included in this analysis is the POR. This allows the user to determine if a repair is carried out once an indication is found. For simplicity, the POR is set equal to 1.0 and the effects of sizing uncertainty will be omitted in the alpha xLPR code.

At each inspection time, the PND will be stored for each instantaneous flaw depth in the analysis. The user can then select either dependent or independent inspections. For subsequent inspections, the PND values are multiplied to simulate independent inspections, while only the most recent PND is used to simulate dependent inspections. The PND values as a function of time can be used to modify the probabilities of failure to account for inspections.

With this methodology, the flaws are assumed to be fully “repaired” after inspection. In addition, if one is found during an inspection, it is assumed that all flaws are repaired. The effects of other inspection driven mitigation are ignored.

Mitigation and Remediation

Two key aspects in the predictions of low probability of rupture are mitigation and remediation events. The final version of xLPR needs to be flexible enough to be able to handle the variety of possible mitigation and remediation techniques. However, for the alpha xLPR code, only two select techniques will be implemented.

Pre-emptive

In this initial pilot study version, mitigation techniques that modify the stress behavior of the cross section will be modeled by a change in the residual stress behavior. The user will input the new WRS distribution information (σ_{0WRS_u}, X_{c_u}) and the time at which the mitigation occurs. Within the code, this change will affect both the crack initiation and growth of PWSCC. For initiation, the mean arrival rate will be modified by

$$\lambda_{mitigated} = \left(\frac{\sigma_i}{\sigma_u}\right)^{-4} \lambda \qquad \text{(Eq. B-16)}$$

where

σ_i = original ID stress ($\sigma_{0DW}+\sigma_{0NTE}+\sigma_{0P}+\sigma_{0WRS}$),
σ_u = updated ID stress ($\sigma_{0DW}+\sigma_{0NTE}+\sigma_{0P}+\sigma_{0WRS\_u}$),
λ = original mean arrival rate.

The crack growth coefficients will remain the same, but the growth will be modified per the input residual stress distribution.

The second option is to completely mitigate the PWSCC. For this option, a mitigation time is input, and both crack initiation and growth will not occur in the future. A probability of mitigation effectiveness may be added, which will evoke a random number to determine if this full mitigation is effective.

Inspection Driven

An appropriate method to apply credit for inspection remains an open issue, and will not be included in the prototype xLPR code.

Outputs

The outputs from xLPR are going to depend highly on the recommendations from the Acceptance Group. For the alpha xLPR code, certain outputs are required to verify the code is working correctly. These outputs will include:

- An echo file that replicates the input

- Probability of particular crack opening sizes (initiation, leakage, and other crack opening area (COA)) as a function of time.
    - Without inspection, without SSE
    - With inspection, without SSE
    - With inspection, with SSE
    - With and without mitigation
- Crack length and depth distributions as a function of time
- Others? - This list will change as the code is developed.

Sample Problem

For alpha xLPR code verification, a sample analysis is being developed based on the pressurizer surge nozzle PWSCC pilot study problem.

Appendix B - References

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

[B-2] 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).

[B-3] SAND2001-0417, http://infoserve.sandia.gov/sand_doc/2001/010417.pdf

[B-4] Helton et. al., SAND2006-2901, http://prod.sandia.gov/techlib/access-control.cgi/2006/062901.pdf

[B-5] Kaplan, S., & Garrick, B. (1981). On The Quantitative Definition of Risk Analysis, 1 (1), 11-27.

[B-6] 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.

[B-7] P. M. Scott, "Prediction of Alloy 600 Component Failures in PWR Systems," Proceedings of the CORROSION/96 Research Topical Symposia 1996, National Association of Corrosion Engineers (NACE), Houston (1996).

[B-8] R. Staehle, "Bases for Predicting the Earliest Penetrations Due to SCC for Alloy 600 on the Secondary Side of PWR Steam Generators," NUREG/CR-6737 (2001).

[B-9] Amzallag, et al, "Stress Corrosion Life Assessment of 182 and 82 Welds Used in PWR Components," 10th International Symposium on Environmental Degradation of Materials in Nuclear Power Systems - Water Reactors, August 5 to 9, 2001.

[B-10] 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.

[B-11] 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.

[B-12] 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.

[B-13] O. Harris, D. Dedhia, E.D. Eason and S.D. Patterson, “Probability of Failure in BWR Reactor Coolant Piping: Probabilistic Treatment of Stress Corrosion Cracking in 304 and 316NG BWR Piping Weldments”, U.S. Nuclear Regulatory Commission Report NUREG/CR-4792, Vol. 3, Washington, D.C., December 1986.

[B-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).

[B-15] WinPraise Manual

[B-16] NURBIM report

[B-17] 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.

[B-18] 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.

[B-19] 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.

[B-20] 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.

[B-21] 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.

[B-22] Rahman, S., Brust, F., Ghadiali, N., Choi, Y., Krishnaswamy, P., Moberg, F., Brickstad, B., and Wilkowski, G., “Refinement and Evaluation of Crack-Opening-Area Analyses for Circumferential Through-Wall Cracks in Pipes,” NUREG/CR-6300, March 1995.

[B-23] Rahman, S., Ghadaili, N., Paul, D., and Wilkowski, G., “Probabilistic Pipe Fracture Evaluations for Leak-Rate-Detection Applications,” NUREG/CR-6004, April 1995.

[B-24] Rudland D., Wolterman, R. and Wilkowski, G., “ Impact of PWSCC and Current Leak Detection on Leak-Before-Break,” Emc2 final report to NRC, January 31, 2003.

[B-25] Rudland D.L., Wilkowski, G., and Scott, P., "Effects of Crack Morphology Parameters on Leak-rate Calculations in LBB Evaluations," International Journal of Pressure Vessels and Piping, 79 (2002) 99-102.

[B-26] 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.

[B-27] "User's Guide for SQUIRT (Windows Version 1.1)," Battelle Memorial Institute, March 2003.

[B-28] EPRI Presentation, "Results of POD Assessment for Dissimilar Metal Welds in LBB Scope," September 29, 2009.

APPENDIX C. xLPR VERSION 1.0 INPUT PARAMATERS AND UNCERTAINTY DISTRIBUTIONS

Appendix C lists the input values and files used in the analyses documented in this report. The controlled file set is contained on the xLPR CM site on the Battelle SharePoint server:
https://websps1.battelle.org/nrcnureg/home/xLPR_CM/default.aspx

List of Input Parameters and Distributions

Table C-1. Constant parameters and base case values.

Variable Name	Description	Source	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	n/a	yr	initiation model	60
Pipe Outer Diameter	pipe outside diameter	p. 12 of program plan (section 3)	m	orientation, load	0.381
Pipe Thickness	pipe thickness	p. 12 of program plan (section 3)	m	Grower, Ksurf, Ktwc, Prleak05b, load, orientation, coalescence, SC_Fail, ENG2_mp, COD	0.040132
Alpha	PWSCC power law constant	Table F-2 of MRP-263 and page 4-3 of MRP-115, equation (13) of program plan	$(m/s)/(MPa\text{-}m^{0.5})^{1.6}$	grower	2.01E-12
Beta	PWSCC exponent in equation (B-13) of program plan	MRP-263 and MRP-115, equation (13) of program plan	N/A	grower	1.6
Kth	reference crack stress intensity factor	MRP-263 and MRP-115, addition to equation (13) of program plan	$MPa\text{-}m^{0.5}$	grower	0

Variable Name	Description	Source	Unit	Input for Models	Value for Base Case
Tref	absolute reference temperature to normalize data	MRP-263 and MRP-115, equation (13) of program plan	K	grower	598.15
Fx_NT, Fx_NT_c	axial force for thermal expansion	table 2 of program plan (p. 18)	kN	load	3.87
Mx_NT, Mx_NT_c	moment in x direction for thermal expansion	table 2 of program plan (p. 18)	kN-m	load	65.3
My_NT, My_NT_c	moment in y direction for thermal expansion	table 2 of program plan (p. 18)	kN-m	load	-57.54
Mz_NT, Mz_NT_c	moment in z direction for thermal expansion	table 2 of program plan (p. 18)	kN-m	load	52.99
Fx_NTS, Fx_NTS_c	axial force for thermal stratification	table 2 of program plan (p. 18)	kN	load	17.39
Mx_NTS, Mx_NTS_c	moment in x direction for thermal stratification	table 2 of program plan (p. 18)	kN-m	load	2.51
My_NTS, My_NTS_c	moment in y direction for thermal stratification	table 2 of program plan (p. 18)	kN-m	load	-80.79
Mz_NTS, Mz_NTS_c	moment in z direction for thermal stratification	table 2 of program plan (p. 18)	kN-m	load	87.9
Fx_SSE, Fx_SSE_c	axial force for SSE	table 2 of program plan (p. 18)	kN	load	28.02
Mx_SSE, Mx_SSE_c	moment in x direction for SSE	table 2 of program plan (p. 18)	kN-m	load	32.39
My_SSE, My_SSE_c	moment in y direction for SSE	table 2 of program plan (p. 18)	kN-m	load	59.25
Mz_SSE, Mz_SSE,c	moment in z direction for SSE	table 2 of program plan (p. 18)	kN-m	load	94.89
Fx_DW, Fx_DW_c	axial force for deadweight	table 2 of program plan (p. 18)	kN	load	0.31

Variable Name	Description	Source	Unit	Input for Models	Value for Base Case
Mx_DW, Mx_DW_c	moment in x direction for deadweight	table 2 of program plan (p. 18)	kN-m	load	1.31
My_DW, My_DW_c	moment in y direction for deadweight	table 2 of program plan (p. 18)	kN-m	load	0.21
Mz_DW, Mz_DW_c	moment in z direction for deadweight	table 2 of program plan (p. 18)	kN-m	load	1.02
sig4	weld residual stress fitting parameter (4th)		MPa	Ksurf	0
Young’s Mod. for Alloy 600	Young's modulus for pipe material	table 1 of program plan	MPa	Crack initiation	2.07E+05
SigYS Alloy 600	Yield Strength (MPa) for alloy 600 (NOT 82/182) (used for Imethod = 2 only)		MPa	Initiation Model	*344.7*
SigUTS Alloy 600	Ultimate tensile strength (MPa) for alloy 600 (used for Imethod = 2 only)		MPa	Initiation Model	*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 Init	22000
SigTH	Threshold stress for initiation		MPa	crack Init	137.9
XN1	Exponent of Method 1 equation (used for Imethod=1 only)		unitless	crack Init	4
XN3	Exponent of Method 3 equation (used for Imethod=3 only) (dimensionless)		unitless	crack Init	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 Init	60

Variable Name	Description	Source	Unit	Input for Models	Value for Base Case
MTS_2	Duration of interval in the operating history		yr	crack Init	0
MTS_3	Duration of interval in the operating history		yr	crack Init	0
MTS_4	Duration of interval in the operating history		yr	crack Init	0
MTS_5	Duration of interval in the operating history		yr	crack Init	0
MTS_6	Duration of interval in the operating history		yr	crack Init	0
MTS_7	Duration of interval in the operating history		yr	crack Init	0
MTS_8	Duration of interval in the operating history		yr	crack Init	0
MTS_9	Duration of interval in the operating history		yr	crack Init	0
MTS_10	Duration of interval in the operating history		yr	crack Init	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, Elasticity_TP304		table 1 of program plan	MPa	RO_epso, COD	177100
elasticity_A516_Gr_70		table 1 of program plan	MPa		186300
elasticity_Alloy182		table 1 of program plan	MPa		203100
C1	Sample value for distribution of C1 (used for Imethod=3 only). The value of C1 (deterministic) is derived from data.				0.04

Table C-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	Initiation Model	Distribution Type	Normal
				Mean	3.00E-03
				Stdev	1.50E-04
				Determin	3.00E-03
				Type	Aleatory
crack_depth_init	crack_depth_init / crack_depth_init_random	m	Initiation Model	Distribution Type	Normal
				Mean	1.50E-03
				Stdev	7.50E-05
				Determin	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 distribution. The Heat-to-Heat sampled value is the median for within-Heat distribution.	Nunits+1	Crack initiation	Distribution Type	logN
				Geometric mean	1.20E-09
				second parameter (H-H) (stdev of log N B1)	1.607
				Determin	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.	unitless	crack Init	Distribution Type	logN
				Geometric mean	0
				Geometric Stdev	1.7419
				Determin	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]	unitless	Crack_init	Distribution Type	logN
				Geometric mean	3.1629
				second parameter (H-H) (stdev of log N A)	1.1595
				Determin	3.1629
				Type	Epistemic
AWH_Stdev	Standard deviation for the normal distribution of within heat distribution AmuWH [Nunits_Max], used with Imethod = 1 only.	unitless	crack Init	Distribution Type	logN
				Geometric mean	0
				Geometric Stdev	2.915
				Determin	0
				Type	Epistemic
random_placement (RandULoc)	placement of a new crack (cannot be changed by the user)	N/A	Initiation Model	Distribution Type	U[0,1]
				Min	0
				Max	1
				Type	2
random_number for time (RandU3)	time of crack initiation (cannot be changed by the user)	N/A	Initiation Model	Distribution Type	U[0,1]
				Min	0
				Max	1
				Type	Epistemic
sig0_wrs	axial stress component for wrs	MPa	load	Distribution Type	Normal
				mean	300.3
				Stdev	110
				Min	150
				Max	551
				Determin	150
				Type	Epistemic
sig0_wrs_mitigated	axial stress component for wrs - mitigated	MPa	load	Distribution Type	Normal

				mean	-344.75
				Stdev	34
				Min	-447
				Max	-242
				Determin	-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
				Determin	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
				Determin	0.38
				Type	Epistemic
OD_stress_random, Outer_Diameter_Stress	WRS in the outer diameter	N/A	load	Distribution Type	Uniform
				min	0.5
				max	1
				Determin	0.5
				Type	Epistemic
f_Weld	weld factor: common factor applied to all specimens fabricated from the same weld to account for weld	N/A	grower	Distribution Type	logN
				Geometric mean	0.99894

	wire/stick heat processing and for weld fabrication			Geometric Stdev	1.83475
				Min	0
				Max	2.71
				Determin	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
				Determin	15636
				Type	Aleatory
P	peak-to-valley ratio	N/A	grower	Distribution Type	Normal
				Mean	9.5
				Stdev	1.36
				Determin	9.5
				Type	Aleatory
c	characteristic width of crack growth rate curve	mV	grower	Distribution Type	Normal
				Mean	22.5
				Stdev	3.21
				Determin	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	Uniform
				Min	0
				Max	1
				Determin	0.5
				Type	Aleatory
β1 (POD_beta1)	parameter β1 for POD of a SC	N/A	ISI	Distribution Type	Normal
				mean	2.7076

				Stdev	0.2085
				Determin	2.7076
				Type	Epistemic
β2 (POD_beta2)	parameter β2 for POD of a SC	N/A	ISI	Distribution Type	Normal
				mean	0.0031
				Stdev	0.0045
				Correlated to POD_beta1	0.86
				Determin	0.0031
				Type	Epistemic
Pressure	pressure in the pipe	MPa	Leakage, TWCFail, SCFail	Distribution Type	Normal
				mean	15.5132
				Stdev	0.1551
				Determin	15.5132
				Type	Epistemic
Temperature	temperature in the pipe	C	Leakage, Grower	Distribution Type	Normal
				mean	344.9
				Stdev	0.0882
				Determin	345
				Type	Epistemic
Pipe Yield Stress - SS 304 (sigy_TP304)	yield stress of pipe material	MPa	TWC-FAIL	Distribution Type	logN
				Geometric mean	168.763
				Geometric Stdev	1.232792
				Determin	168.763
				Type	Aleatory
Pipe Ultimate Stress - SS 304 (sigu_TP304)	ultimate stress of pipe material	MPa	TWC-FAIL	Distribution Type	logN
				Geometric mean	450.6127
				Geometric Stdev	1.123960

				Correlate sigy_ TP304	0.6066
				Determin	450.6127
				Type	Aleatory
Ramberg-Osgood Fit Parameter, F - SS 304 (F_TP304, RO_F)	Ramberg-Osgood fit parameter, F	MPa	TWC-FAIL	Distribution Type	logN
				Geometric mean	562.1217
				Geometric Stdev	1.080277
				Determin	562.1217
				Type	Aleatory
Ramberg-Osgood Fit Parameter, n - SS 304 (n_TP304, RO_n)	Ramberg-Osgood fit parameter, n	N/A	TWC-FAIL	Distribution Type	logN
				Geometric mean	4.260565
				Geometric Stdev	1.141419
				Correlate F_TP304	-0.6047
				Determin	4.260565
				Type	Aleatory
J-resistance 182 (Jic_Alloy_182, Resist_Jic)	material initiation J-resistance	N/mm = kJ/m^2	TWC-FAIL	Distribution Type	logN
				Mean	482.7
				Stdev	1.783853
				Determin	482.7
				Type	Aleatory
J-resistance coefficient 182 (C_Alloy_182, Resist_C)	material initiation J-resistance coefficient	N/mm$^{(exponent+1)}$	TWC-FAIL	Distribution Type	logN
				Mean	260.1
				Stdev	1.621629
				Correlate Jic_Alloy_182	0.9
				Determin	260.1
				Type	Aleatory
J-resistance exponent 182	material initiation J-resistance exponent	N/A	TWC-FAIL	Distribution Type	logN

(m_Alloy_182, Resist_m)				Mean	0.612089
				Stdev	1.173811
				Determin	0.612089
				Type	Aleatory
sigy_Alloy_182	yield strength of Alloy 182	MPa	Leakage, TWCFail, SCFail	Distribution Type	logN
				Geometric mean	361.5464
				Geometric Stdev	1.269675
				Determin	361.5464
				Type	Aleatory
sigu_Alloy_182	ultimate strength of Alloy 182	MPa	Leakage, TWCFail, SCFail	Distribution Type	logN
				Geometric mean	580.1362
				Geometric Stdev	1.104332
				Correlated to sigy_Alloy_182	0.5
				Determin	580.1362
				Type	Aleatory
sigy_A516_Gr_70	yield strength of A516 Gr 70	MPa	Leakage, TWCFail, SCFail	Distribution Type	logN
				Geometric mean	227.4765
				Geometric Stdev	1.099388
				Determin	227.4765
				Type	Aleatory
sigu_A516_Gr_70	ultimate strength of A516 Gr 70	MPa	Leakage, TWCFail, SCFail	Distribution Type	logN
				Geometric mean	519.1096
				Geometric Stdev	1.056711
				Correlated to sigy_A516_Gr_70	0.4866
				Determin	519.1096
				Type	Aleatory

F_A516_Gr_70	Fit parameter for the Ramberg-Osgood model	MPa	Leakage, TWCFail, SCFail	Distribution Type	logN
				Geometric mean	911.5219
				Geometric Stdev	1.093895
				Determin	911.5219
				Type	Aleatory
n_A516_Gr_70	Fit parameter for the Ramberg-Osgood model for A516 Gr 70	N/A	Leakage, TWCFail, SCFail	Distribution Type	logN
				Geometric mean	4.288899
				Geometric Stdev	1.132017
				Correlated to F_A516_Gr_70	-0.8565
				Determin	4.288899
				Type	Aleatory

List Controlled Model Files Used in the Analysis

Table C-3. Framework files and descriptions.

Framework Files	Description
Beta_v2.02_GS10.11_M02.gsm	Final controlled copy of the beta phase of the model development and transformed to GSxLPRv1.0_M02.gsm.
GSxLPRv1.0_M02.gsm GSxLPRv1.01_M02.gsm GSxLPRv1.02_M02.gsm	GoldSim software version of the framework configured to run as a probabilistic case.
GSxLPRv1.0_M02.gsp GSxLPRv1.01_M02.gsp GSxLPRv1.02_M02.gsp	GoldSim Player software version of the framework configured to run as a probabilistic case.
GSxLPRv1.02_M02_Deterministic.gsm	GoldSim software version of the framework configured to run as a deterministic case.
GSxLPRv1.02_M02_Deterministic.gsp	GoldSim Player software version of the framework configured to run as a deterministic case.

Table C-4. Input files and descriptions.

Input File	Description
BETA_Inputs_AE_09_30_2010.xlsx	The input data for the framework model is contained in an Excel workbook file. A GoldSim "Spreadsheet" element is used to dynamically link to the Excel file. The uncertain parameters and constants are contained separate worksheets in the Excel workbook.

Table C-5. Fortran modules and descriptions.

Fortran Module (DLL)	Description
Coalescense_DLLx_v2.2.dllx	Model used to evaluate whether two adjacent cracks will coalesce.
COD_DLLx_v2.1.dllx	Model used to calculate the COD for TWCs.
crack_init_v2.1.dllx	Model used to calculate the number of cracks initiated, time, and location of the initiated crack.
grower_DLL_v2.1.dllx	Model used to calculate the crack growth.
ISI_DLL_v2.1.dllx	Model used to calculate the PND.
kSurf_DLL_v1.1.dllx	Model used to calculate the stress intensity at the crack surface.
kTWC_DLL_v1.1.dllx	Model used to calculate the stress intensity at a crack's deepest point
load_DLL_v1.1.dllx	Model used to calculate the total axial membrane stress and total bending moment.
SCFail_DLL.v2.1.dllx	Model used to assess the stability of a surface crack in a pipe subjected to combined tension and bending loading.

Fortran Module (DLL)	Description
SQURT_DLL_V1.1.dllx	Model used to calculate two-phase critical flow rate for water leaking from a TWC.
TWCFail_DLL_v2.1.dllx	Model used to assess the stability of a TWC in a pipe subjected to combined tension and bending loading.

Table C-6. Files and descriptions.

File	Description
TRANSFORMERS_v1.0.exe	Post Processor used to account for effects from leak rate detection and inspection for cracks.
EXPECTATION_v1.0.exe	Post processor used to average any result over the aleatory uncertainty and estimate the mean and selected quantiles.
Options.txt	input control file for TRANSFORMERS
Exp_Options.txt	input control file for EXPECTATION
inspection.txt	file containing the times at which inspections will be performed
quantiles.txt	file containing the quantiles of interest
times.txt	file containing the simulation time steps (in months)
variables_list.txt	file containing the names of the files to be post processed

APPENDIX D. UNCERTAINTY AND SENSITIVITY ANALYSES FOR COMPLEX SYSTEMS

D.1 Aleatory and Epistemic Uncertainty in the Analysis of Complex Systems

A performance assessment (PA) for a nuclear power plant, or in general any type of engineered facility, is an analysis intended to answer three questions about the facility (i.e., Q1, Q2 and Q3) and one question about the analysis itself (i.e., Q4): Q1, "What can happen?"; Q2, "How likely is it to happen?"; Q3, "What are the consequences if it does happen?"; and Q4, "How much confidence exists in the answers to the first three questions?". Two types of uncertainty are inherent in the answers to the preceding questions. Questions Q1 and Q2 relate to uncertainty with respect to future events (e.g., pipe failures, seismic events, etc.) at the facility under consideration whose occurrence, within the limits of our ability to predict the future, is assumed to be random. The descriptor aleatory is usually used for this type of uncertainty. Question Q4 relates to a lack of knowledge with respect to the appropriateness/correctness of the assumptions, models and parameter values that underlie answering questions Q1, Q2 and Q3. The descriptor epistemic is usually used for this type of uncertainty. The nature of aleatory and epistemic uncertainty and the importance of their separation in analyses of complex systems has been discussed by a number of authors [D-1 to D-7].

Answering the four indicated questions leads to an analysis based on three basic mathematical structures or entities: EN1, a probability space characterizing aleatory uncertainty; EN2, a function that predicts the physical behavior of the facility under consideration; and EN3, a probability space characterizing epistemic uncertainty [D-8, D-9]. The probability space corresponding to EN1 characterizes aleatory uncertainty and provides the basis for answering Questions Q1 and Q2. In practice, the function corresponding to EN2 is one or more very complex numerical models and provides the basis for answering Question Q3. The probability space corresponding to EN3 characterizes epistemic uncertainty and provides the basis for answering Question Q4. The nature of basic analysis components EN1, EN2 and EN3 is elaborated on in the remainder of this section.

Closely associated with the characterization of epistemic uncertainty provided by the probability space corresponding to EN3 and the answering of Question Q4 are the concepts of uncertainty analysis and sensitivity analysis. Uncertainty analysis designates the determination of the epistemic uncertainty in analysis results that derives from epistemic uncertainty in analysis inputs. Sensitivity analysis designates the determination of the contribution of the epistemic uncertainty in individual analysis inputs to the epistemic uncertainty in analysis results. Basically, uncertainty and sensitivity analysis are the means by which EN3 gives rise to the answer to Question Q4. A number of approaches to uncertainty and sensitivity analysis exist, including differential analysis, response surface methods, variance decomposition methods, and sampling-based (i.e., Monte Carlo) methods [D-10 to D-17].

The first entity underlying a PA, EN1, corresponds to a probability space ($\mathcal{A}$, $\mathbb{A}$, p_A), where $\mathcal{A}$ is the set of everything that could occur in the particular universe under consideration (i.e., over some specified time period for the facility under analysis), $\mathbb{A}$ is a suitably restricted set of subsets of $\mathcal{A}$ for which probability is defined, and p_A is the function that defines probability for elements of $\mathbb{A}$ (i.e., if $\mathcal{S}$ is an element of $\mathbb{A}$, then $p_A(\mathcal{S})$ is the probability of $\mathcal{S}$). In the usual terminology of probability theory, $\mathcal{A}$ is called the sample space or sometimes the universal set; elements of $\mathcal{A}$ are called elementary events; elements of $\mathbb{A}$ are called events; p_A is called a probability measure; and $p_A(\mathcal{S})$ is the probability of the event $\mathcal{S}$. In the terminology of radioactive waste disposal, elements of $\mathcal{A}$ are often called futures; elements of $\mathbb{A}$ are often called scenarios or scenario classes; and $p_A(\mathcal{S})$ is the probability of scenario $\mathcal{S}$.

Although the concept of a probability space is important conceptually and convenient notationally, calculations involving a probability space ($\mathcal{A}$, $\mathbb{A}$, p_A) are often described with a density function $d_A(\mathbf{a})$, where

$$p_A\left(\mathcal{S}\right)=\int_{\mathcal{S}} d_A\left(\mathbf{a}\right)\mathrm{d}S \qquad \text{(Eq. D-1)}$$

for $\mathcal{S} \in \mathbb{A}$, a $\in \mathcal{S}$, and dS corresponding to an increment of volume from $\mathcal{S}$. Then, the expected value, variance, CDF, and CCDF at time τ (yr) associated with a real-valued function $y = f(\tau|\text{a})$ defined on $\mathcal{A}$ are defined by

$$E_A\left[f\left(\tau\,|\,\mathbf{a}\right)\right]=\int_{\mathcal{A}} f\left(\tau\,|\,\mathbf{a}\right)d_A\left(\mathbf{a}\right)\mathrm{d}A, \qquad \text{(Eq. D-2)}$$

$$V_A\left[f\left(\tau\,|\,\mathbf{a}\right)\right]=\int_{\mathcal{A}}\left\{f\left(\tau\,|\,\mathbf{a}\right)-E_A\left[f\left(\tau\,|\,\mathbf{a}\right)\right]\right\}^2 d_A\left(\mathbf{a}\right)\mathrm{d}A, \qquad \text{(Eq. D-3)}$$

$$p_A\left[f\left(\tau\,|\,\mathbf{a}\right)\le y\right]=\int_{\mathcal{A}}\underline{\delta}_y\left[f\left(\tau\,|\,\mathbf{a}\right)\right]d_A\left(\mathbf{a}\right)\mathrm{d}A, \qquad \text{(Eq. D-4)}$$

and

$$p_A\left[y \le f\left(\tau \mid \mathbf{a}\right)\right] = \int_{\mathcal{A}} \bar{\delta}_y\left[f\left(\tau \mid \mathbf{a}\right)\right] d_A\left(\mathbf{a}\right) \mathrm{d}A, \qquad \text{(Eq. D-5)}$$

respectively, where

$$\underline{\delta}_y\left[f\left(\tau \mid \mathbf{a}\right)\right] = \begin{cases} 1 & \text{if } f\left(\tau \mid \mathbf{a}\right) \le y \\ 0 & \text{otherwise,} \end{cases} \qquad \bar{\delta}_y\left[f\left(\tau \mid \mathbf{a}\right)\right] = \begin{cases} 1 & \text{if} f\left(\tau \mid \mathbf{a}\right) > y \\ 0 & \text{otherwise,} \end{cases}$$

and dA represents an increment of volume from $\mathcal{A}$. A CCDF is defined in Eq. D-5 because of the typical usage of CCDFs rather than CDFs to represent uncertainty in risk assessments. In particular, a CCDF answers the question "How likely is it to be this bad or worse?" which is usually the question asked with respect to individual consequences in a risk assessment. However, conversion between CCDFs and CDFs is straightforward as a CDF is simply one minus the corresponding CCDF. In turn, the q quantile value (e.g., q = 0.05, 0.5 ~ median, 0.95) for $f(\tau|\mathrm{a})$ is the value y such that

$$q = p_A\left[f\left(\tau \mid \mathbf{a}\right) \le y\right] = \int_{\mathcal{A}} \underline{\delta}_y\left[f\left(\tau \mid \mathbf{a}\right)\right] d_A(\mathbf{a}) \mathrm{d}A. \qquad \text{(Eq. D-6)}$$

For notational purposes, the q quantile value for $f(\tau|\mathrm{a})$ can be represented by $Q_q[f(\tau|\mathrm{a})]$. The variance $V_A[f(\tau|\mathrm{a})]$ provides less information than the CDF and CCDF defined by the probabilities $p_A[f(\tau|\mathrm{a}) \le y]$ and $p_A[\mathrm{y} < f(\tau|\mathrm{a})]$, respectively, and is rarely used in the summary of results obtained in a PA.

In PAs for nuclear power plants, the probability space ($\mathcal{A}$, $\mathbb{A}$, p_A) for aleatory uncertainty is usually defined to characterize the occurrence of potential future events over some time period of interest (e.g., for a time period corresponding to one year plant operation or perhaps the intended operating life of the plant) that could affect the behavior/performance of the plant. Specifically, each element a of the sample space $\mathcal{A}$ is a vector of the form $\mathbf{a} = [a_1, a_2, \ldots, a_n]$, where the elements of $\mathbf{a}$ characterize the properties of one potential sequence of occurrences over the time interval under consideration.

For example, the future behavior of a nuclear power plant in a particular analysis might be assumed to be affected by a single class of disruptive events (e.g., large pipe failures) whose occurrence is characterized by a Poisson process with a rate constant λ (yr^{-1}). Each individual event is characterized by a time t (yr) of occurrence and a vector $\mathbf{p}$ of additional properties (e.g., size, location …). Then, for a specified time period [a, b] (e.g., [a, b] = [0, 40 yr]), each future a would be a vector of the form

$$\mathbf{a} = \left[t_1, \mathbf{p}_1, t_2, \mathbf{p}_2, \ldots, t_n, \mathbf{p}_n\right], \qquad \text{(Eq. D-7)}$$

where n is the number of occurrences in the time interval [a, b], $a \le t_1 \le t_2 \le \cdots \le t_n \le b$ are the times of the individual occurrences, p_1, p_2, …, p_n are vectors indicating the properties of the individual occurrences, and a_N =[0] represents the future in which no events occur. In turn, the sample space $\mathcal{A}$ would have the form:

$$\mathcal{A} = \{\mathbf{a} : \mathbf{a} = \left[t_1, \mathbf{p}_1, t_2, \mathbf{p}_2, \ldots, t_n, \mathbf{p}_n\right] \text{ for (i) } n = 0,1,2,\ldots, \text{ (ii) } a \le t_1 \le t_2 \le \cdots \le t_n \le b \text{ and (iii) } \mathbf{p}_i \in \mathcal{P} \text{ for } i = 1,2,\ldots,n\}$$

(Eq. D-8)

where $\mathcal{P}$ is the set of all possible values for the property vector $\mathbf{p}$ and scenario probabilities $p_A(\mathcal{S})$ are defined for subsets $\mathcal{S}$ of $\mathcal{A}$. For example,

$$p_A(\mathcal{A}_n) = \left\{\left[\lambda\left(b-a\right)\right]^n / n!\right\} \exp\left[-\lambda\left(b-a\right)\right] \qquad \text{(Eq. D-9)}$$

for

$$\mathcal{A}_n = \left\{\mathbf{a} : \mathbf{a} = \left[t_1, \mathbf{p}_1, t_2, \mathbf{p}_2, \ldots, t_n, \mathbf{p}_n\right] \text{ for } a \le t_1 \le t_2 \le \cdots \le t_n \le b \text{ and } \mathbf{p}_i \in \mathcal{P} \text{ for } i = 1,2,\ldots,n\right\}$$

The sets $\mathcal{A}_n$ are examples of the elements of $\mathbb{A}$. However, much more complex examples exist (e.g., subsets of $\mathcal{A}$ defined with restrictions involving elements of $\mathcal{P}$); in such cases, definitions of the corresponding set probabilities can become very complicated.

A potential representation of aleatory uncertainty in the context of a large pipe failure at a nuclear power station is shown in Table D-1, with only one aleatory set considered.

Table D-1. Potential representation of aleatory uncertainty.

half_crack_length_init: initial half length of a crack (m). *Distribution:* Normal. *Mean*: 3x10^{-3}. *Stdev:* 1.50x10^{-4}.
crack_depth_init: initial depth of a crack (m). *Distribution:* Normal. *Mean*: 1.5x10^{-3}. *Stdev:* 7.50x10^{-5}.
QoverR: ratio of thermal activation energy for PWSCC crack growth over universal gas constant (K). *Distribution:* Normal. *Mean*: 15636. *Stdev:* 631.
P: peak to valley ratio (unitless). *Distribution:* Normal. *Mean*: 9.5. *Stdev:* 1.36.
C: characteristic width of crack growth rate curve (mV). *Distribution:* Normal. *Mean*: 22.5. *Stdev:* 3.21.
POD_detection: random number determining whether a crack is detected or not (not used in current version but needed for ISI module) (unitless). *Distribution:* Uniform. *Min*: 0. *Max:* 1.
sigy_TP304: yield stress of material (Stainless Steel) TP 304 (MPa). *Distribution:* LogNormal. *Geometric Mean*: 168.763. *Geometric Stdev:* 1.232792.
sigu_TP304: ultimate stress of material (Stainless Steel) TP 304 (MPa). *Distribution:* LogNormal. *Geometric Mean*: 450.6127. *Geometric Stdev:* 1.123960. *Correlation*: 0.6066 with sigy_TP304.
F_TP304: Ramberg-Osgood Fit parameter F for material (Stainless Steel) TP 304 (MPa). *Distribution:* LogNormal. *Geometric Mean*: 562.1217. *Geometric Stdev:* 1.080277.
n_TP304: Ramberg-Osgood Fit parameter n for material (Stainless Steel) TP 304 (unitless). *Distribution:* LogNormal. *Geometric Mean*: 4.260565. *Geometric Stdev:* 1.141419. *Correlation*: -0.6047 with F_TP304.
Resist_Jic: material initiation J-resistance for Alloy 182 (N/mm=kJ/m^2) *Distribution:* LogNormal *Geometric Mean* : 482.7 *Geometric Stdev:* 1.783853
Resist_C: material initiation J-resistance coefficient C for Alloy 182 (N/mm=kJ/m^2). *Distribution:* LogNormal. *Geometric Mean*: 260.1. *Geometric Stdev:* 1.621629. *Correlation*: 0.9 with Resist_Jic.
Resist_m: material initiation J-resistance exponent for Alloy 182 (N/mm=kJ/m^2). *Distribution:* LogNormal. *Geometric Mean*: 0.612089. *Geometric Stdev:* 1.173811.
sigy_alloy182: yield stress for Alloy 182 (MPa). *Distribution:* LogNormal. *Geometric Mean*: 361.5464. *Geometric Stdev:* 1.269675.
sigu_alloy182: ultimate stress for Alloy 182 (MPa). *Distribution:* LogNormal. *Geometric Mean*: 580.1362. *Geometric Stdev:* 1.104332. *Correlation*: 0.5 with sigy_Alloy182.
sigy_A516_Gr_70: yield stress of material A516 Grade 70 (MPa). *Distribution:* LogNormal. *Geometric Mean*: 227.4765. *Geometric Stdev:* 1.099388.
sigu_A516_Gr_70: ultimate stress of material A516 Grade 70 (MPa). *Distribution:* LogNormal. *Geometric Mean*: 519.1096. *Geometric Stdev:* 1.056711. *Correlation*: 0.4866 with sigy_A516_Gr70.
F_A516_Gr_70: Ramberg-Osgood Fit parameter F for material A516 Grade 70 (MPa). *Distribution:* LogNormal. *Geometric Mean*: 911.5219. *Geometric Stdev:* 1.093895.
n_A516_Gr_70: Ramberg-Osgood Fit parameter n for material A516 Grade 70 (unitless). *Distribution:* LogNormal. *Geometric Mean*: 4.288899. *Geometric Stdev:*

1.132017. *Correlation*: -0.8565 with F_A516_Gr70.

The second entity underlying a PA, EN2, corresponds to a model, or more realistically a large system of interacting models, that predict the behavior of a nuclear power plant under accident conditions and various summary measures of this behavior (e.g., crack growth rate, water leak rate, etc.). Notationally, this model can be represented by a function of the form

$$\mathbf{f}(\tau\,|\,\mathbf{a}) = \left[f_1(\tau\,|\,\mathbf{a}), f_2(\tau\,|\,\mathbf{a}), \ldots, f_m(\tau\,|\,\mathbf{a})\right], \qquad \text{(Eq. D-10)}$$

where τ corresponds to time (yr), each element $f_j(\tau|\mathbf{a})$ of $\mathbf{f}(\tau|\mathbf{a})$ is a specific calculated result, and $\mathbf{a}$ is an element of the sample space $\mathcal{A}$ for aleatory uncertainty. In general, the value of $\mathbf{f}(\tau|\mathbf{a})$, and indeed the actual structure of the individual models that are combined to produce $\mathbf{f}(\tau|\mathbf{a})$, will change with changing values for $\mathbf{a}$.

As an example, a potential configuration of component models and associated connections that define $\mathbf{f}(\tau|\mathbf{a})$ for an analysis of pipe failure at a nuclear power station is illustrated in Figure D-1. An additional explanation of the model configuration in Figure D-1 is available in Section 3.0.

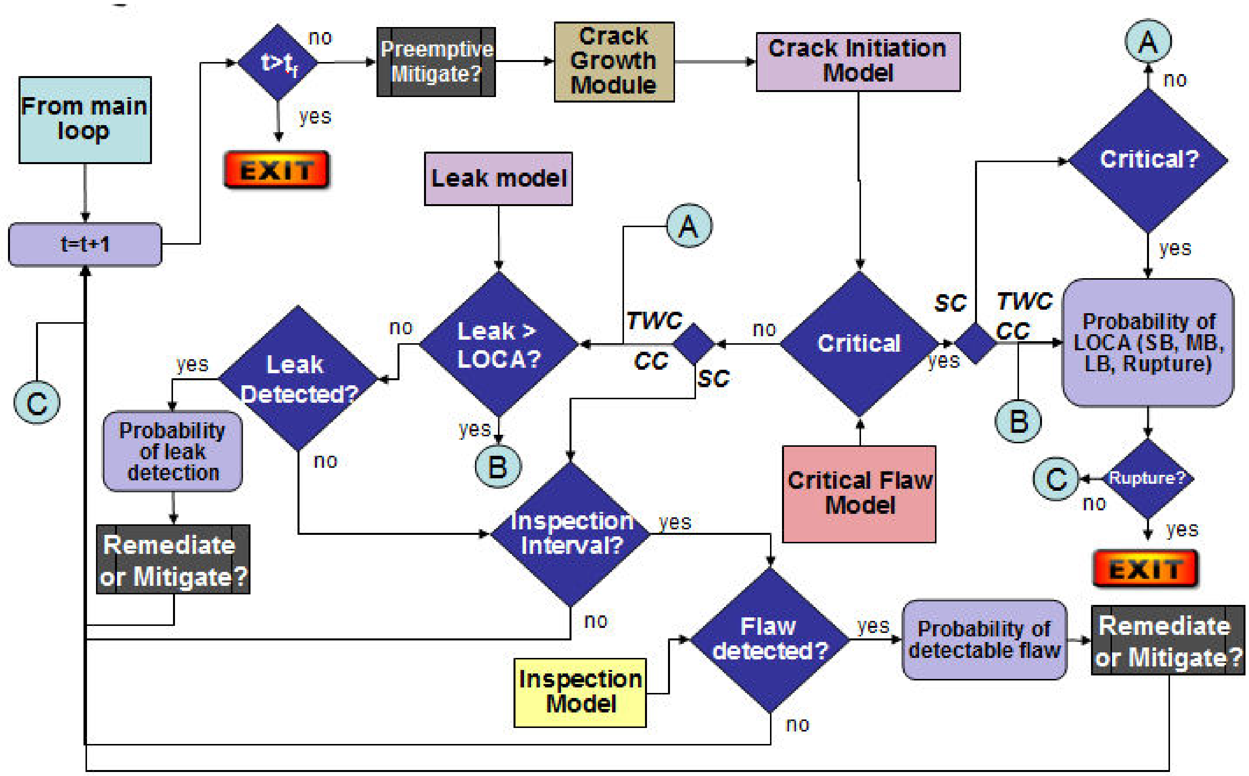


Figure D-1. Potential configuration of component models and associated connections that define f(τ|a) for an analysis of pipe failure at a nuclear power station.

Table D-2 displays an example of potential results that can be calculated and analyzed (i.e., element of $\mathbf{f}(\tau|\mathbf{a})$).

Table D-2. Example of potential results that can be calculated and analyzed.

CT_xx: time-dependent type for crack #xx (xx = 1...19) (unitless). Crack type is an indicator function that can be equal to 0 (no crack occurred), -1 (SC), -2 (TWC). xx = 1...19 (crack has coalesced with crack #xx) or 200 (pipe rupture)
HLA_xx: time-dependent half length for crack #xx (m).
DA_xx: time-dependent depth for crack #xx (m).
OA_xx: time-dependent crack orientation for crack #xx (radian).
FSA: time-dependent fraction of surface area cracked (unitless).
CFO: time-dependent function for critical failure (i.e., pipe rupture) (unitless). Equal to 0 before rupture and 1 after rupture, if rupture occurs.
CFT: critical failure time (month). Set to 9000 months if it does not occur.
TLR: time-dependent Total Leak Rate (m^3/s).
FLO: time-dependent function for first leakage occurrence (unitless). Equal to 0 before first leak and 1 after first leak if leak occurs.
FLT: first leak time (month). Set to 9000 months if it does not occur.

The third entity underlying a PA, EN3, corresponds to a probability space ($\mathcal{E}$, $\mathbb{E}$, p_E) for epistemic uncertainty. The conceptual properties associated with probability space ($\mathcal{E}$, $\mathbb{E}$, p_E) are the same as indicated in Eqs. D-1 to D-6 for the probability space ($\mathcal{A}$, $\mathbb{A}$, p_A) for aleatory uncertainty. In general, the elements of the sample space $\mathcal{E}$ are vectors of the form

$$\begin{aligned}\mathbf{e} &= [\mathbf{e}_A, \mathbf{e}_M] \\ &= [e_{A1}, e_{A2}, \ldots, e_{A,nEA}, e_{M1}, e_{M2}, \ldots, e_{M,nEM}] \\ &= [e_1, e_2, \ldots, e_{nE}], nE = nEA + nEM,\end{aligned} \qquad \text{(Eq. D-11)}$$

where $\mathbf{e}_A = [e_{A1}, e_{A2}, \ldots, e_{A,nEA}]$ is a vector of epistemically uncertain quantities used in the characterization of aleatory uncertainty (e.g., a rate term that defines a Poisson process), $\mathbf{e}_M = [e_{M1}, e_{M2}, \ldots, e_{M,nEM}]$ is a vector of epistemically uncertain quantities used in the evaluation of $\mathbf{f}(\tau|\mathbf{a})$ (e.g., a crack growth rate), and the concept of an uncertain quantity is interpreted broadly enough to include designators for possible values for poorly known functions or models. When notationally convenient, the probability space ($\mathcal{E}$, $\mathbb{E}$, p_E) can be represented with a density function $d_E(\mathbf{e})$.

Reference is made to “poorly known functions or models” in the preceding paragraph. This form of epistemic uncertainty is often given the designation model uncertainty [D-18] and involves a situation where there are multiple alternative models for a process and the analysts involved are not sure which is the appropriate model to use in the specific analysis context under consideration. Although there is no clear divide between where parameter uncertainty ends and model uncertainty begins, the designation model uncertainty is usually used in reference to a situation involving a finite number of structurally distinct models for a process. If probability is being used to mathematically characterize epistemic uncertainty, then a probability distribution would be defined to represent the analysts’ degree of belief with respect to which alternative model is the appropriate model to use. With this approach, the "model" would simply be one more epistemically uncertain variable that is sampled (i.e., an integer-valued pointer variable would be sampled, with the different values for this variable designating the use of different models). Most large analyses have a few pointer variables that identify different possible models. Usually, model uncertainty involves a very limited number of alternative models. In this situation, a possibility is to perform a ceteris paribus analysis in which the entire analysis is performed repeatedly with a different model used in each repetition. However, this approach has the potential to be very computationally demanding.

In practice, the probability space ($\mathcal{E}$, $\mathbb{E}$, p_E) is defined by assigning probability distributions to the individual elements of **e.** In addition, correlations and other restrictions involving the elements of **e** may also be specified. The specified distributions serve as mathematical summaries of all available information with respect to where the appropriate values for the elements of **e** are located and are often developed through expert review processes [D-19 to D-28]. Examples of variables (i.e., potential elements of **e**) that might be treated as being uncertain in an epistemic sense in an analysis of pipe failure at a nuclear power station are presented in Table D-3.

Table D-3. Input variables (i.e., elements of e) that might be treated as being uncertain in an epistemic sense in an analysis of pipe failure at a nuclear power station.

B1: Heat-to-Heat variability. Sample value for distribution of B1 (used for Imethod = 2 only). For each segment, sample from the Within-Heat distribution. The Heat-to-Heat sampled value is the median for Within-Heat distribution (unitless). *Distribution:* Lognormal. *Mean of log*: -20.54. *Stdev of log:* 1.607.
BWH_Stdev: Standard deviation for the normal distribution of Within-Heat distribution BmuWH [Nunits_Max], used with Imethod = 2 only (unitless). *Distribution:* Lognormal. *Mean of log*: 0. *Stdev of log:* 0.555.
A: Heat-to-Heat sampled value for distribution of A (used for Imethod = 1 only), used to generate distribution of AmuWH [Nunits_max] (unitless). *Distribution*: Lognormal. *Mean of log:* 0.141. *Stdev of log:* 0.148.
AWH_Stdev: Standard deviation for the normal distribution of Within-Heat distribution AmuWH [Nunits_Max], used with Imethod = 1 only (unitless). *Distribution:* Lognormal. *Mean of log*: 0. *Stdev of log:* 1.07.
random_placement (RandULoc): placement of a new crack - cannot be changed by the user (unitless). *Distribution:* Uniform. *Min:* 0. *Max:* 1.
random_number for time (RandU3): time of crack initiation - cannot be changed by the user (unitless). *Distribution:* Uniform. *Min:* 0. *Max:* 1.
sig0_wrs: axial stress component for WRS (MPa). *Distribution:* Truncated Normal. *Min:* 150. *Max:* 551. *Mean:* 300.3. *Stdev:* 110.
sig0_wrs_mitigated: axial stress component for WRS, mitigated (MPa). *Distribution:* Truncated Normal. *Min:* -447. *Max:* -242. *Mean:* -344.75. *Stdev:* 34.
Xc: location in the pipe where weld residual stress equal 0 (unitless). *Distribution:* Truncated Normal. *Min:* 0.125. *Max:* 0.5. *Mean:* 0.25. *Stdev:* 0.05.
Xc_Mitigated: location in the pipe where WRS equal 0, mitigated (unitless). *Distribution:* Truncated Normal. *Min:* 0.26. *Max:* 0.5. *Mean:* 0.38. *Stdev:* 0.038.
OD_stress_random, Outer_Diameter_Stress: WRS in the outer diameter (unitless). *Distribution:* Uniform. *Min:* 0.5. *Max:* 1.
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 (unitless). *Distribution:* Truncated Log-normal. *Min:* 0. *Max:* 2.71. *Mean of log:* -0.00106. *Stdev of log:* 0.607.
β1 (POD_beta1): parameter β1 for POD of a SC (unitless). *Distribution:* Normal. *Mean:* 2.71. *Stdev:* 0.2085.
β2 (POD_beta2): parameter β2 for POD of a SC (unitless). *Distribution:* Normal. *Mean:* 0.0031. *Stdev:* 0.0045. *Correlation*: 0.86 with β1.
Pressure: pressure in the pipe (MPa). *Distribution:* Normal. *Mean:* 15.51. *Stdev:* 0.1551.
Temperature: temperature in the pipe (C). *Distribution:* Normal. *Mean:* 344.9. *Stdev:* 0.0882.

With the introduction of the probability space ($\mathcal{E}$, $\mathbb{E}$, p_E) for epistemic uncertainty, the representation for the system model in Eq. D-10 becomes

$$\mathbf{f}(\tau\,|\,\mathbf{a},\mathbf{e}_M)=\left[f_1(\tau\,|\,\mathbf{a},\mathbf{e}_M),f_2(\tau\,|\,\mathbf{a},\mathbf{e}_M),\ldots,f_m(\tau\,|\,\mathbf{a},\mathbf{e}_M)\right] \qquad \text{(Eq. D-12)}$$

and the representation for the density function associated with the probability space ($\mathcal{A}$, $\mathbb{A}$, p_A) for aleatory uncertainty becomes $d_A(\mathbf{a}|\mathbf{e}_A)$. In turn, results of the form in Eqs. D-1, D-2, D-4 and D-5 become:

$$p_A(\mathcal{S}|\,\mathbf{e}_A)=\int_{\mathcal{S}}d_A(\mathbf{a}\,|\,\mathbf{e}_A)\,\mathrm{d}S, \qquad \text{(Eq. D-13)}$$

$$E_A\left[f(\tau\,|\,\mathbf{a},\mathbf{e}_M)\,|\,\mathbf{e}_A\right]=\int_{\mathcal{A}}f(\tau\,|\,\mathbf{a},\mathbf{e}_M)\,d_A(\mathbf{a}\,|\,\mathbf{e}_A)\,\mathrm{d}A, \qquad \text{(Eq. D-14)}$$

$$p_A\left[f(\tau\,|\,\mathbf{a},\mathbf{e}_M)\le y\,|\,\mathbf{e}_A\right]=\int_{\mathcal{A}}\underline{\delta}_y\left[f(\tau\,|\,\mathbf{a},\mathbf{e}_M)\right]d_A(\mathbf{a}\,|\,\mathbf{e}_A)\,\mathrm{d}A, \qquad \text{(Eq. D-15)}$$

and

$$p_A\left[y<f(\tau\,|\,\mathbf{a},\mathbf{e}_M)\,|\,\mathbf{e}_A\right]=\int_{\mathcal{A}}\bar{\delta}_y\left[f(\tau\,|\,\mathbf{a},\mathbf{e}_M)\right]d_A(\mathbf{a}\,|\,\mathbf{e}_A)\,\mathrm{d}A, \qquad \text{(Eq. D-16)}$$

where $f(\tau|\mathbf{a},\mathbf{e}_M)$ corresponds to one of the functions $f_j(\tau|\mathbf{a},\mathbf{e}_M)$ contained in $\mathbf{f}(\tau|\mathbf{a},\mathbf{e}_M)$. Similarly, the q quantile value for $f(\tau|\mathbf{a})$ defined in Eq. D-6 now depends on $\mathbf{e} = [\mathbf{e}_A, \mathbf{e}_M]$ and is appropriately represented by $Q_{Aq}[f(\tau|\mathbf{a},\mathbf{e}_M)|\mathbf{e}_A]$. As $\mathbf{e}$ changes, each of the preceding quantities also changes and has a probability distribution that derives from the probability space ($\mathcal{E}$, $\mathbb{E}$, p_E) for epistemic uncertainty.

In turn, the quantities in Eqs. D-13 to D-16 have distributions that derive from the probability space ($\mathcal{E}$, $\mathbb{E}$, p_E) for epistemic uncertainty. Similarly, elements $f(\tau|\mathbf{a},\ \mathbf{e}_M)$ of $\mathbf{f}(\tau|\mathbf{a},\ \mathbf{e}_M)$ for fixed values of $\mathbf{a}$ also have distributions that derive from the probability space ($\mathcal{E}$, $\mathbb{E}$, p_E) for epistemic uncertainty. For example, the expected value, CDF, CCDF and the q quantile $Q_{Eq}\{E_A[f(\tau|\mathbf{a},\ \mathbf{e}_M)|\ \mathbf{e}_A]\}$ for $E_A[f(\tau|\mathbf{a},\ \mathbf{e}_M)|\ \mathbf{e}_A]$ resulting from epistemic uncertainty are defined by

$$\begin{aligned}E_E\left\{E_A\left[f(\tau\,|\,\mathbf{a},\mathbf{e}_M)\,|\,\mathbf{e}_A\right]\right\}&=\int_{\mathcal{E}}E_A\left[f(\tau\,|\,\mathbf{a},\mathbf{e}_M)\,|\,\mathbf{e}_A\right]\mathrm{d}E\\&=\int_{\mathcal{E}}\left[\int_{\mathcal{A}}f(\tau\,|\,\mathbf{a},\mathbf{e}_M)\,d_A(\mathbf{a}\,|\,\mathbf{e}_A)\,\mathrm{d}A\right]\mathrm{d}E,\end{aligned} \qquad \text{(Eq. D-17)}$$

$$
\begin{aligned}
p_E\left\{E_A\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)|\,\mathbf{e}_A\right]\le\bar{y}\right\} &= \int_{\mathcal{E}}\underline{\delta}_{\bar{y}}\left\{E_A\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)|\,\mathbf{e}_A\right]\right\}d_E\left(\mathbf{e}\right)\mathrm{d}E \\
&= \int_{\mathcal{E}}\underline{\delta}_{\bar{y}}\left[\int_{\mathcal{A}} f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)d_A\left(\mathbf{a}\,|\,\mathbf{e}_A\right)\mathrm{d}A\right]d_E\left(\mathbf{e}\right)\mathrm{d}E,
\end{aligned}
$$

(Eq. D-18)

$$
\begin{aligned}
p_E\left\{E_A\left[\bar{y}<f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)|\,\mathbf{e}_A\right]\right\} &= \int_{\mathcal{E}}\bar{\delta}_{\bar{y}}\left\{E_A\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)|\,\mathbf{e}_A\right]\right\}d_E\left(\mathbf{e}\right)\mathrm{d}E \\
&= \int_{\mathcal{E}}\bar{\delta}_{\bar{y}}\left[\int_{\mathcal{A}} f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)d_A\left(\mathbf{a}\,|\,\mathbf{e}_A\right)\mathrm{d}A\right]d_E\left(\mathbf{e}\right)\mathrm{d}E,
\end{aligned}
$$

(Eq. D-
19)

and the value of $\bar{y}$ such that $q = p_E\{E_A[f(\tau\,|\,\mathbf{a},\mathbf{e}_M)\,|\,\mathbf{e}_A]\le\bar{y}\}$, respectively. Similarly, the expected value, CDF, CCDF and the q quantile $Q_{Eq}[f(\tau|\mathbf{a}, \mathbf{e}_M)]$ for $f(\tau|\mathbf{a}, \mathbf{e}_M)$ resulting from epistemic uncertainty are defined by

$$E_E\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)\right] = \int_{\mathcal{EM}} f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)d_{EM}\left(\mathbf{e}_M\right)\mathrm{d}EM, \qquad \text{(Eq. D-20)}$$

$$p_E\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)\le y\right] = \int_{\mathcal{EM}}\underline{\delta}_y\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)\right]d_{EM}\left(\mathbf{e}_M\right)\mathrm{d}EM, \qquad \text{(Eq. D-21)}$$

$$p_E\left[y<f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)\right] = \int_{\mathcal{EM}}\bar{\delta}_y\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)\right]d_{EM}\left(\mathbf{e}_M\right)\mathrm{d}EM, \qquad \text{(Eq. D-22)}$$

and the value of y such that $q = p_E\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)\le y\right]$, respectively. The preceding results associated with $f(\tau|\mathbf{a}, \mathbf{e}_M)$ are defined with respect to the probability space ($\mathcal{EM}$, $\mathbb{EM}$, p_{EM}) for $\mathbf{e}_M$ because $\mathbf{e}_M$ is the only epistemically uncertain quantity under consideration.

D.2 Propagation of Uncertainty

The propagation of aleatory uncertainty is considered first. Direct evaluation of the integrals in Eqs. D-13 to D-16 is usually not possible. As a result, some form of numerical procedure must be used. The two most widely used procedures are SRS and stratified sampling.

In SRS, a random sample

$$\mathbf{a}_j = \left[a_{1j}, a_{2j}, \ldots, a_{n_j,j}\right], j = 1, 2, \ldots, nSA, \qquad \text{(Eq. D-23)}$$

is generated from $\mathcal{A}$ in consistency with the definition of the probability space ($\mathcal{A}$, $\mathbb{A}$, p_A). In general, the properties of ($\mathcal{A}$, $\mathbb{A}$, p_A) and hence the resultant sample in Eq. D-23 will depend on $\mathbf{e}_A$. In turn, the results in Eqs. D-13 to D-16 are approximated by:

$$p_A\left(\mathcal{S}|\,\mathbf{e}_A\right) \cong \sum_{j=1}^{nSA} \delta_S\left(\mathbf{a}_j\right) / \, nSA, \qquad \text{(Eq. D-24)}$$

$$E_A\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)|\,\mathbf{e}_A\right] \cong \sum_{j=1}^{nSA} f\left(\tau\,|\,\mathbf{a}_j,\mathbf{e}_M\right) / \, nSA, \qquad \text{(Eq. D-25)}$$

$$p_A\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right) \le y\,|\,\mathbf{e}_A\right] \cong \sum_{j=1}^{nSA} \underline{\delta}_y\left[f\left(\tau\,|\,\mathbf{a}_j,\mathbf{e}_M\right)\right] / \, nSA, \qquad \text{(Eq. D-26)}$$

and

$$p_A\left[y < f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)|\,\mathbf{e}_A\right] \cong \sum_{j=1}^{nSA} \overline{\delta}_y\left[f\left(\tau\,|\,\mathbf{a}_j,\mathbf{e}_M\right)\right] / \, nSA, \qquad \text{(Eq. D-27)}$$

respectively, with $\delta_S(\mathbf{a}_j) = 1$ if $\mathbf{a}_j \in \mathcal{S}$ and $\delta_S(\mathbf{a}_j) = 0$ if $\mathbf{a}_j \notin \mathcal{S}$.

In stratified sampling, the sample space $\mathcal{A}$ is subdivided into a sequence of subsets $\mathcal{A}_j$, j = 1, 2, …, nSA, with the properties that $\cup_j \mathcal{A}_j = \mathcal{A}$ and $\mathcal{A}_i \cap \mathcal{A}_j = \varnothing$ for $i \neq j$. Then the results in Eqs. D-13 to D-16 are approximated by:

$$p_A\left(\mathcal{S}|\,\mathbf{e}_A\right) \cong \sum_{j=1}^{nSA} \delta_S\left(\mathbf{a}_j\right) p_A\left(\mathcal{A}_j\,|\,\mathbf{e}_A\right), \qquad \text{(Eq. D-28)}$$

$$E_A\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)|\,\mathbf{e}_A\right] \cong \sum_{j=1}^{nSA} f\left(\tau\,|\,\mathbf{a}_j,\mathbf{e}_M\right) p_A\left(\mathcal{A}_j\,|\,\mathbf{e}_A\right), \qquad \text{(Eq. D-29)}$$

$$p_A\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right) \le y\,|\,\mathbf{e}_A\right] \cong \sum_{j=1}^{nSA} \underline{\delta}_y\left[f\left(\tau\,|\,\mathbf{a}_j,\mathbf{e}_M\right)\right] p_A\left(\mathcal{A}_j\,|\,\mathbf{e}_A\right), \qquad \text{(Eq. D- 30)}$$

and

$$p_A\left[y < f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)|\,\mathbf{e}_A\right] \cong \sum_{j=1}^{nSA} \overline{\delta}_y\left[f\left(\tau\,|\,\mathbf{a}_j,\mathbf{e}_M\right)\right] p_A\left(\mathcal{A}_j\,|\,\mathbf{e}_A\right), \qquad \text{(Eq. D-31)}$$

respectively, where $\mathbf{a}_j$ is a representative element of $\mathcal{A}_j$. Event trees are often used to assess the effects of aleatory uncertainty and, in essence, are simply algorithms for implementing stratified sampling.

The widely used Kaplan/Garrick ordered triple representation for risk [D-29] corresponds to a summary description of an analysis based on stratified sampling. Specifically, this representation defines risk by the set

$$\left(\mathcal{S}_j, pS_j, \mathbf{cS}_j\right), j = 1, 2, \ldots, nS, \qquad \text{(Eq. D-32)}$$

where $\mathcal{S}_j$ is a set of similar occurrences, pS_j is the probability of $\mathcal{S}_j$, and $\mathbf{cS}_j$ is a vector of consequences associated with $\mathcal{S}_j$. In the context of the stratified sampling results in Eqs. D-28 to D-31, $\mathcal{S}_j = \mathcal{A}_j$, $pS_j = p_A(\mathcal{A}_j|\ \mathbf{e}_A)$, and $\mathbf{cS}_j = \mathbf{f}(\mathbf{a}_j|\mathbf{e}_M)$.

The propagation of epistemic uncertainty is now considered. As for the propagation of aleatory uncertainty, direct evaluation of the integrals that formally define the propagation of epistemic uncertainty is unlikely to be possible in a real analysis. SRS and stratified sampling are possibilities for the propagation of epistemic uncertainty. However, because of its efficient stratification properties, LHS is widely used for the propagation of epistemic uncertainty in complex and computationally demanding analyses [D-30, D-31].

LHS operates in the following manner to generate a sample size of nSE from the distributions D_1, D_2, …, D_{nE} associated with the elements of $\mathbf{e} = [e_1, e_2, \ldots, e_{nE}]$, where the distributions D_1, D_2, …, D_{nE} in effect define the probability space ($\mathcal{E}$, $\mathbb{E}$, p_E) for epistemic uncertainty. The range of each e_j is exhaustively divided into nSE disjoint intervals of equal probability and one value e_{ij} is randomly selected from each interval. The nSE values for e_1 are randomly paired without replacement with the nSE values for e_2 to produce nSE pairs. These pairs are then randomly combined without replacement with the nSE values for e_3 to produce nSE triples. This process is continued until a set of nSE nE-tuples

$$\mathbf{e}_i = \left[e_{i1}, e_{i2}, \ldots, e_{i,nE}\right], i = 1, 2, \ldots, nSE, \qquad \text{(Eq. D-33)}$$

is obtained, with this set constituting the LHS. If needed, a restricted pairing technique exists that can be used to induce a specified rank correlation structure in an LHS [D-32, D-33].

Once the LHS in Eq. D-33 is generated, the results in Eqs. D-17 to D-19 can be approximated by:

$$\begin{aligned} E_E\left\{E_A\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)|\,\mathbf{e}_A\right]\right\} &\cong \sum_{i=1}^{nSE} E_A\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_{Mi}\right)|\,\mathbf{e}_{Ai}\right]/\,nSE \\ &\cong \sum_{i=1}^{nSE}\left[\sum_{j=1}^{nSA} f\left(\tau\,|\,\mathbf{a}_j,\mathbf{e}_{Mi}\right)/\,nSA\right]/\,nSE, \end{aligned} \qquad \text{(Eq. D-34)}$$

$$\begin{aligned} p_E\left\{E_A\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)|\,\mathbf{e}_A\right]\le \bar{y}\right\} &\cong \sum_{i=1}^{nSE} \underline{\delta}_{\bar{y}}\left\{E_A\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_{Mi}\right)|\,\mathbf{e}_{Ai}\right]\right\}/\,nSE \\ &\cong \sum_{i=1}^{nSE} \underline{\delta}_{\bar{y}}\left[\sum_{j=1}^{nSA} f\left(\tau\,|\,\mathbf{a}_j,\mathbf{e}_{Mi}\right)/\,nSA\right]/\,nSE, \end{aligned}$$

(Eq. D-35)

and

$$\begin{aligned} p_E\left\{\bar{y} < E_A\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)|\,\mathbf{e}_A\right]\right\} &\cong \sum_{i=1}^{nSE} \bar{\delta}_{\bar{y}}\left\{E_A\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_{Mi}\right)|\,\mathbf{e}_{Ai}\right]\right\}/\,nSE \\ &\cong \sum_{i=1}^{nSE} \bar{\delta}_{\bar{y}}\left[\sum_{j=1}^{nSA} f\left(\tau\,|\,\mathbf{a}_j,\mathbf{e}_{Mi}\right)/\,nSA\right]/\,nSE, \end{aligned}$$

(Eq. D-36)

respectively, and the q quantile $Q_{Eq}\{E_A[f(\tau|\mathbf{a},\ \mathbf{e}_M)|\ \mathbf{e}_A]\}$ for $E_A[f(\tau|\mathbf{a},\ \mathbf{e}_M)|\ \mathbf{e}_A]$ resulting from epistemic uncertainty can be obtained from the approximation in Eq. D-35. Similarly, the results in Eqs. D-20 to D-22 can be approximated by:

$$E_E\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)\right] \cong \sum_{i=1}^{nSE} f\left(\tau\,|\,\mathbf{a},\mathbf{e}_{Mi}\right)/\,nSE, \qquad \text{(Eq. D-37)}$$

$$p_E\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)\le y\right] \cong \sum_{i=1}^{nSE} \underline{\delta}_y\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_{Mi}\right)\right]/\,nSE, \qquad \text{(Eq. D-38)}$$

$$p_E\left[y < f\left(\tau\,|\,\mathbf{a},\mathbf{e}_M\right)\right] \cong \sum_{i=1}^{nSE} \bar{\delta}_y\left[f\left(\tau\,|\,\mathbf{a},\mathbf{e}_{Mi}\right)\right]/\,nSE, \qquad \text{(Eq. D-39)}$$

respectively, and the q quantile $Q_{Eq}[f(\tau|\mathbf{a},\ \mathbf{e}_M)]$ for $f(\tau|\mathbf{a},\ \mathbf{e}_M)$ resulting from epistemic uncertainty can be obtained from the approximation in Eq. D-38. Distributional results for other quantities dependent on **e** and $\mathbf{e}_M$ are obtained in a similar manner.

The propagation of uncertainty is now illustrated with results from V1.0 framework analysis of the xLPR model. This analysis used an LHS of the form

$$\mathbf{e}_i = [\mathbf{e}_{Ai}, \mathbf{e}_{Ei}], i = 1, 2, ..., nSE, \qquad \text{(Eq. D-40)}$$

of size $nSE = 100$ from $nE = 16$ epistemically uncertain variables (i.e., from the sample space $\mathcal{E}$ for epistemic uncertainty as indicated in conjunction with Eq. D-11 and Table D-3).

A large number of analysis results conditional on specific realizations $\mathbf{a}$ of aleatory uncertainty (i.e., elements of the function $\mathbf{f}(\tau|\mathbf{a}, \mathbf{e}_M)$; see Table K3-4 [D-34]) were analyzed as part of the xLPR pilot model. As a single example, results for expected fractional surface area cracked $E_A[SFA(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]$ of a weld under nominal conditions (i.e., for the future $\mathbf{a}_N$ indicated in conjunction with Eq. D-7 involving no disruptions of any kind) are shown in Figure D-2. Specifically, Figure D-2(a) contains 100 individual curves corresponding to $E_A[SFA(\tau|\mathbf{a}, \mathbf{e}_{Mi})|\mathbf{e}_A]$ for each of the 100 LHS elements indicated in Eq. D-40. The spread in these curves provides a representation of the epistemic uncertainty present in the estimation of $E_A[SFA(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]$. The mean and quantile curves (i.e., q = 0.05, 0.5 ~ median, 0.95) in Figure D-2(a) provide additional summaries of the epistemic uncertainty present in the estimation of $E_A[SFA(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]$ and are defined as indicated in Eqs. D-20 and D-21 and estimated as indicated in Eqs. D-37 and D-38. As half of the 100 realization result in no crack at all, only 50 out of the 100 curves plotted are visible. Moreover lower quantiles (i.e., q = 0.05, 0.5 ~ median) are equal to 0.

The CDF and CCDF for $E_A[SFA(50\ \text{yr}|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]$ in Figure D-2(b) are defined and estimated as indicated in Eqs. D-21, D-22, D-38 and D-39 and provide a more detailed summary of the epistemic uncertainty in $E_A[SFA(50\ \text{yr}|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]$ than is provided by selected quantiles (e.g., by q = 0.05, 0.5 and 0.95 at 50 yr). Examples of many additional results of this type are presented in [D-34, Appendices J and K].

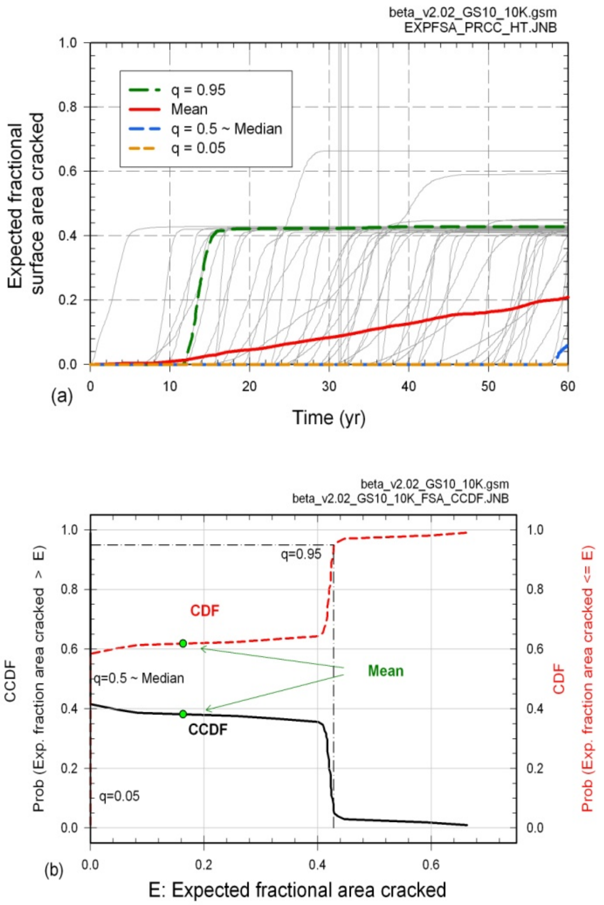


Figure D-2. Uncertainty analysis results for $E_A[SFA(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]$: (a) $E_A[SFA(\tau\ |\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]$ for all (i.e., nSE = 100) LHS elements, and (b) estimated CDF and CCDF $E_A[SFA(50\text{ yr}|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]$.

Distributions over aleatory uncertainty are now considered. As a single example, surface area cracked $SFA(\tau|\mathbf{a}, \mathbf{e}_M)$ in a weld in presence of nominal conditions is considered (Figure D-3).

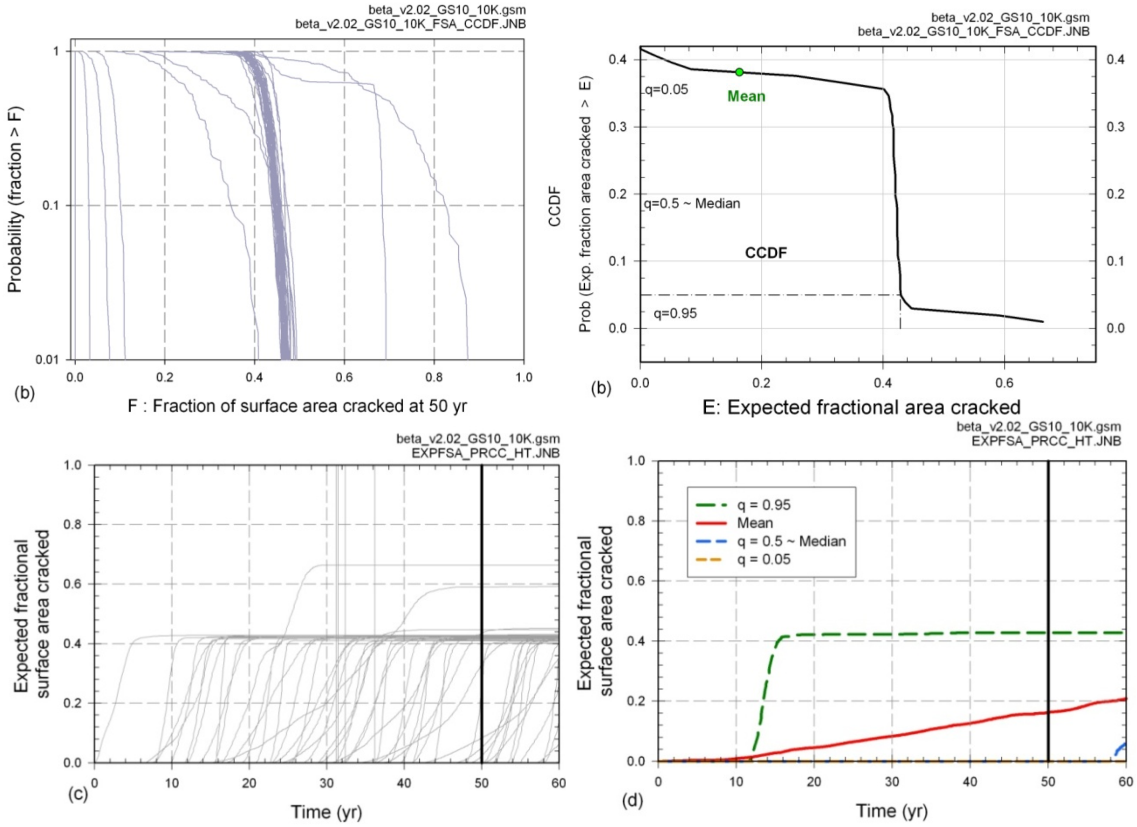


Figure D-3. Uncertainty analysis results for SFA(τ|a, $\mathbf{e}_M$): (a) CCDFs for SFA(50 yr|a, $\mathbf{e}_{Mi}$) for all (i.e., nSE = 100) LHS elements, (b) CCDF for E_A[SFA(50 yr|a, $\mathbf{e}_M$)|$\mathbf{e}_A$], (c) E_A[SFA(τ|a, $\mathbf{e}_M$)|$\mathbf{e}_A$] for all LHS elements, and (d) mean and quantile curves for E_A[SFA(τ|a, $\mathbf{e}_M$)|$\mathbf{e}_A$].

Individual CCDFs for SFA(50 yr|**a**, $\mathbf{e}_M$) conditional on $\mathbf{e}_{Mi}$ for each element of the LHS in Eq. D-40 are shown in Figure D-3(a). In concept, the CCDFs in Figure D-3(a) can be defined and approximated as indicated in Eqs. D-22 and D-27. However, in this analysis as is often the case in other analyses, this type of naïve sampling is very inefficient because of the large number of futures **a** that result in no consequences of interest (i.e., no crack occurring in this example). To avoid this inefficiency, the CCDFs in Figure D-3(a) will sometimes be estimated by using importance sampling to generate futures that involve cracks and then implementing a correction for this restricted sampling. Specifically, the CCDFs in Figure D-3(a) could be estimated from the exceedance probabilities

$$\boldsymbol{p_A}[\boldsymbol{FSA}(\mathbf{50\ yr}|\mathrm{a},\mathbf{e}_{\boldsymbol{Mi}}) > y|\mathbf{e}_{\boldsymbol{Ai}}] \cong \boldsymbol{p_A}(\tilde{\boldsymbol{A}}|\mathbf{e}_{\boldsymbol{i}})\sum_{\boldsymbol{j=1}}^{\boldsymbol{nSA}}\bar{\boldsymbol{\delta}}_{\boldsymbol{y}}[\boldsymbol{FSA}(\mathbf{50\ yr}|\mathrm{a},\mathbf{e}_{\boldsymbol{Mi}})]\,/\boldsymbol{nSA}$$

(Eq. D-41)

where (i) $\tilde{A}$ is the subset of $\mathcal{A}$ that contains only futures that cause at least one crack, (ii) the probability $p_A(\tilde{A}|\mathbf{e}_i)$ for $\tilde{A}$ depends in part on properties defined by elements of $\mathbf{e}_M$, and (iii) the individual futures $\mathbf{a}_j$ were sampled from $\tilde{A}$ (see Sect. J8.3 of Ref. [D-34] for additional details).

As indicated in Eqs. D-14 and D-25, CCDFs of the form appearing in Figure D-3(a) can be reduced to expected values $E_A[SFA(50\text{ yr}|\mathbf{a},\ \mathbf{e}_{Mi})|\mathbf{e}_{Ai}]$ over aleatory uncertainty. With use of the approximation procedure indicated in conjunction with Eq.D-41, this reduction becomes

$$\boldsymbol{E_A[SFA(}50\text{ yr}|\text{a, e}_{\boldsymbol{Mi}}\boldsymbol{)|}\text{e}_{\boldsymbol{Ai}}\boldsymbol{]}\cong \boldsymbol{p_A\left(\tilde{A}\middle|\mathbf{e}_i\right)\sum_{j=1}^{nSA}[FSA(50\text{ yr}|\mathbf{a},\mathbf{e}_{Mi})]/nSA}$$

(Eq. D-42)

and produces 100 expected values $E_A[SFA(50\text{ yr}|\mathbf{a},\ \mathbf{e}_{Mi})|\mathbf{e}_{Ai}]$ that can be summarized with a CCDF (Figure D-3(b)). The CCDF in Figure D-3(b) provides a representation of the epistemic uncertainty present in the estimation of $E_A[SFA(50\text{ yr}|\mathbf{a},\ \mathbf{e}_M)|\mathbf{e}_A]$ and corresponds to results indicated in Eqs. D-19 and D-36 with the estimation of $E_A[SFA(50\text{ yr}|\mathbf{a},\ \mathbf{e}_M)|\mathbf{e}_A]$ in Eq. D-36 modified to be consistent with the estimation procedure in Eq. D-42. Selected quantiles (i.e., q = 0.05, 0.5, 0.95) are also shown in Figure D-3(b). As indicated in conjunction with Eqs. D-18 and D-35, these quantiles are obtained by solving the equation $q = p_E\{E_A[f(\tau|\mathbf{a},\ \mathbf{e}_M)|\mathbf{e}_A] \le \bar{y}\}$ for $\bar{y}$, where $p_E\{\ldots\}$ represents probability with respect to epistemic uncertainty. However, once the CCDF in Figure D-3(b) is constructed, these quantiles can also be obtained by (i) starting at $1 - q$ on the ordinate, (ii) drawing a horizontal line to the CCDF, and then (iii) drawing a vertical line to the abscissa to obtain the quantile $Q_{Eq}\{E_A[f(\tau|\mathbf{a},\ \mathbf{e}_M)|\mathbf{e}_A]\}$.

The 100 values for expected surface area cracked $E_A[SFA(50\text{ yr}|\mathbf{a},\ \mathbf{e}_{Mi})|\mathbf{e}_{Ai}]$ summarized in the CCDF in Figure D-3(b) correspond to the 100 expected values above τ = 50 yr in Figure D-3(c). The spread of these curves provides a summary of the epistemic uncertainty in $E_A[SFA(50\text{ yr}|\mathbf{a},\ \mathbf{e}_M)|\mathbf{e}_A]$ as a function of time. A simple Monte Carlo analysis was used to calculate time-dependent expected values in Figure D-3(c).

The determination of mean and quantile results as indicated in Figure D-3b can be carried out for a sequence of times. Plotting these results as functions of time produces the mean and quantile curves in Figure D-3(d), with these curves providing an overall summary on the epistemic uncertainty in the expected dose $E_A[FSA(\tau|\mathbf{a},\ \mathbf{e}_M)|\mathbf{e}_A]$ as a function of time.

If desired, the epistemic uncertainty associated with the CCDFs in Figure D-3(a) can be summarized with mean and quantile curves in a manner analogous to that shown in Figure D-3(d) for expected curves (Figure D-4). The summary curves in Figure D-4 are defined by means and probabilities analogous to those defined by Eqs. D-17, D-18, D-34 and D-35 with $p_A[FSA(50\text{ yr}|\mathbf{a},\ \mathbf{e}_M) > y|\mathbf{e}_A]$ replacing $E_A[f(\tau|\mathbf{a},\ \mathbf{e}_M)|\mathbf{e}_A]$.

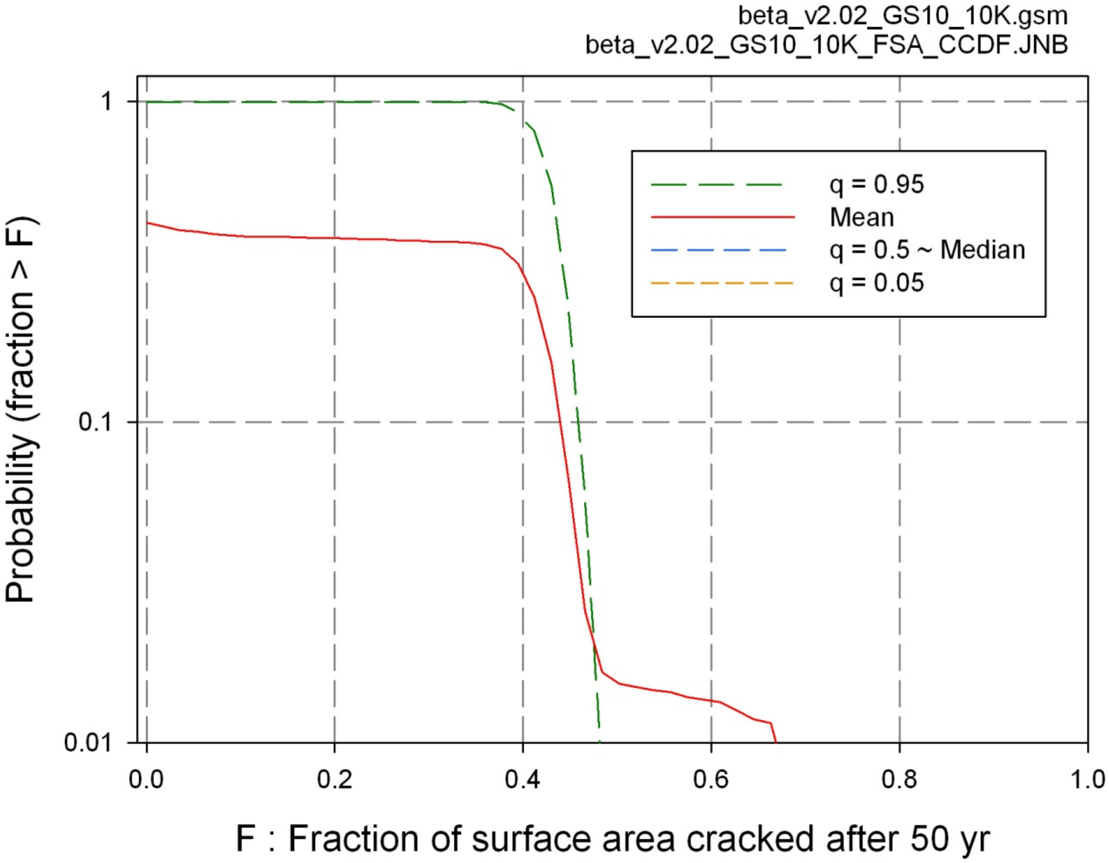


Figure D-4. Mean and quantile curves for $p_A[FSA(150\,\mathrm{yr}|\mathbf{a},\mathbf{e}_M) > y|\mathbf{e}_A]$.

The content of this section is adapted from a presentation written to describe PA for the geologic disposal of radioactive waste [D-38]. The possibility of this adaptation shows how similar PAs for nuclear power stations and PAs for radioactive waste disposal are at a conceptual level. Of course, the processes, models and time scales that must be considered are very different for these two types of facilities.

D.3 Procedures for sensitivity analysis

A number of approaches to sensitivity analysis that can be used in conjunction with a sampling-based uncertainty analysis are listed and briefly summarized below. In this summary, (i) x_j is an element of a vector $\mathbf{x} = [x_1, x_2, \ldots, x_{nX}]$ of epistemically uncertain analysis inputs, (ii) y_k is an element of $\mathbf{y}(\mathbf{x}) = [y_1(\mathbf{x}), y_2(\mathbf{x}), \ldots, y_{nY}(\mathbf{x})]$, (iii) $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{i,nX}]$, $i = 1, 2, \ldots, nS$, is a random or LHS from the possible values for $\mathbf{x}$ generated in consistency with the joint distribution assigned to the x_j, (iv) $\mathbf{y}_i = \mathbf{y}(\mathbf{x}_i)$ for $i = 1, 2, \ldots, nS$, and (v) x_{ij} and y_{ik} are elements of $\mathbf{x}_i$ and $\mathbf{y}_i$, respectively. This section uses examples from the 1996 PA for the Waste Isolation Pilot Plant [D-39, D-40] to illustrate sampling-based procedures for sensitivity analysis. In this PA, three replicated LHSs of size $nS = 100$

each (denoted R1, R2 and R3, respectively) were used in the propagation of the assessed epistemic uncertainty in $nX = 57$ analysis inputs [D-41].

Scatterplots. Scatterplots are plots of the points $[x_{ij}, y_{ik}]$ for $i = 1, 2, \ldots, nS$ and can reveal nonlinear or other unexpected relationships (Figure D-5). In many analyses, scatterplots provide all the information that is needed to understand the sensitivity of analysis results to the uncertainty in analysis inputs. Further, scatterplots constitute a natural starting point in a complex analysis that can help in the development of a sensitivity analysis strategy using one or more additional techniques. Additional information: Sect. 6.6.1 [D-42]; Sect. 6.1 [D-17].

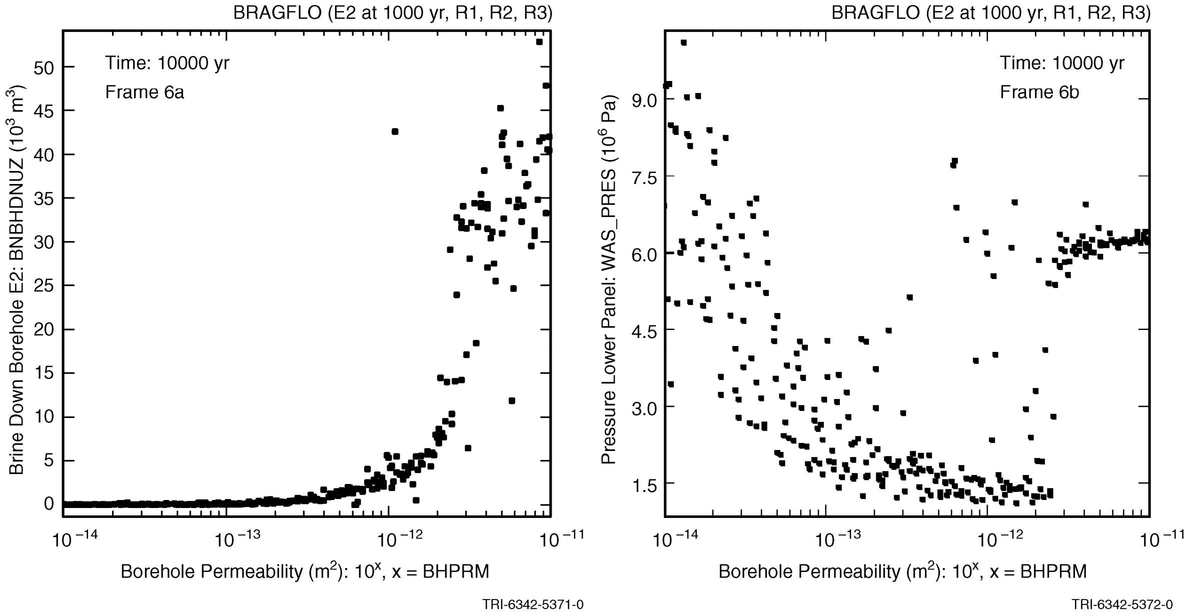


Figure D-5. Examples of scatterplots obtained in a sampling-based uncertainty/sensitivity analysis [D-43].

Cobweb Plots. Cobweb plots are plots of the points $[\mathbf{x}_i, y_{ik}] = [x_{i1}, x_{i2}, \ldots, x_{i,nX}, y_{ik}]$ for $i = 1, 2, \ldots, nS$ and provide a two-dimensional representation for $[\mathbf{x}_i, y_{ik}]$, which is a $nX + 1$ dimensional quantity. Specifically, values for the y_{ik} and also for the elements x_{ij} of x_i appear on the ordinate of a cobweb plot and the variables themselves are designated by fixed locations on the abscissa. Then, the values y_{ik}, $i = 1, 2, .., nS$, for y_k and the values x_{ij}, $i = 1, 2, \ldots, nS$, for each x_j are plotted above the locations for y_k and x_j on the abscissa and each $nX + 1$ dimensional point $[\mathbf{x}_i, y_{ik}]$ is represented by a line connecting the values for the individual components of $[\mathbf{x}_i, y_{ik}]$. Cobweb plots provide more information in a single plot frame than a scatterplot but are harder to read. Additional information: Sect. 11.7 [D-44].

Correlation. A CC provides a measure of the strength of the linear relationship between x_j and y_k. The CC between x_j and y_k is equal to the standardized regression coefficient (SRC) in a linear regression relating y_k to x_j and is also equal in absolute value to the square root of the R^2 value associated with the indicated regression. When calculated with raw (i.e., untransformed) data, the CC is often referred to as the Pearson Coefficient of Correlation (CC). Additional information: Sect. 6.6.4 [D-42]; Sect. 6.2 [D-17].

Regression Analysis. Regression analysis provides an algebraic representation of the relationships between y_k and one or more x_js. Regression analysis is usually performed in a stepwise fashion, with initial inclusion of most important x_j, then two most important x_js, and so on until no more x_js that significantly affect y_k can be identified. Variable importance is indicated by order of selection in the stepwise process, changes in R^2 values as additional variables are added to the regression model, and SRCs for the x_js in the final regression model (Table D-4). A display of regression results in the form shown in Table D-4 is very unwieldy when results at a sequence of times are under consideration. In this situation, a more compact display of regression results is provided by plotting time-dependent SRCs (Figure D-7). Additional information: Sects. 6.6.2, 6.6.3, and 6.6.5 [D-42]; Sect. 6.3 [D-17].

Table D-4. Example of stepwise regression analysis to indentify uncertain variables affecting the uncertainty in pressure at 10,000 yr in Figure D-6 (Table 8.6, [D-43]).

Step[a]	**Variable**[b]	**SRC**[c]	R^{2}[d]
1	*WMICDFLG*	0.718	0.508
2	*HALPOR*	0.466	0.732
3	*WGRCOR*	0.246	0.792
4	*ANHPRM*	0.129	0.809
5	*SHRGSSAT*	0.070	0.814
6	*SALPRES*	0.063	0.818

[a] Steps in stepwise regression analysis.
[b] Variables listed in the order of selection in regression analysis.
[c] SRCs for variables in final regression model.
[d] Cumulative R^2 value with entry of each variable into regression model.

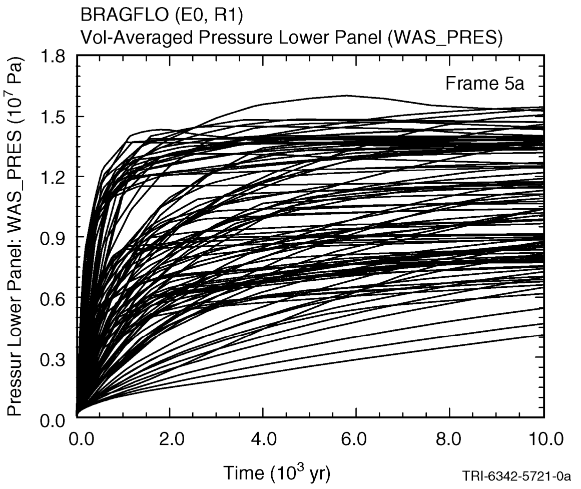


Figure D-6. Representation of uncertainty in analysis results that are functions: Pressure as a function of time (Figure 7.5 [D-43]).

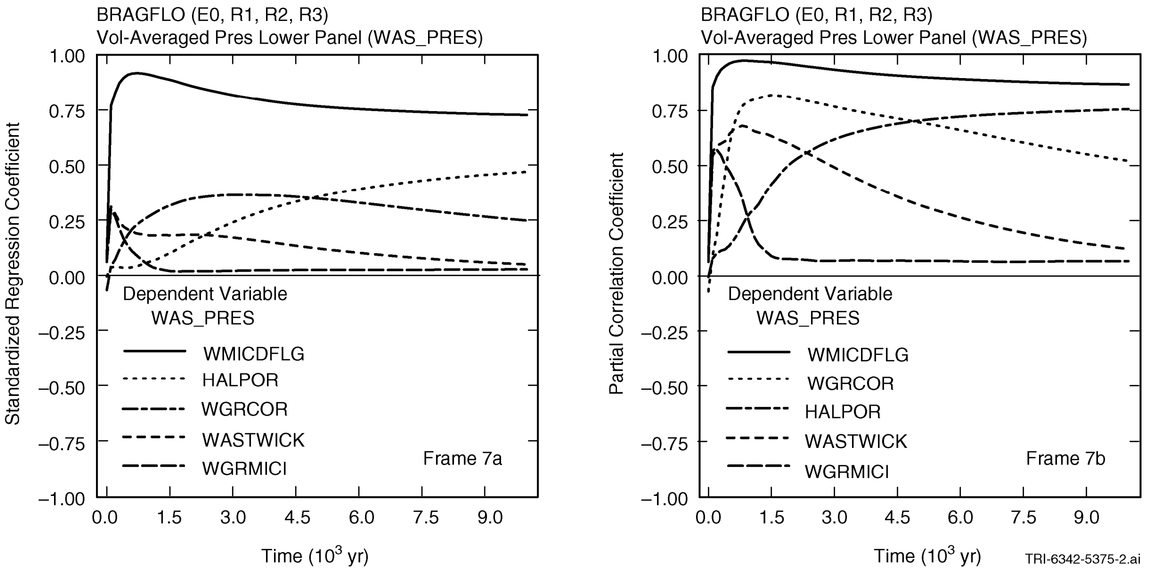


Figure D-7. Time-dependent sensitivity analysis results for uncertain pressure curves in Figure D-6, (a) SRCs as a function of time, and (b) PCCs as a function of time (Figure 8.3 [D-43]).

Partial Correlation. A partial correlation coefficient (PCC) provides a measure of the strength of the linear relationship between y_k and x_j after the linear effects of all other elements of $\mathbf{x}$ have been removed. Similarly to SRCs, PCCs can be determined as a function of time for time-dependent analysis results (Figure D-7(b)). Additional information: Sect. 6.6.4 [D-42]; Sect. 6.4 [D-17].

Rank Transformations. A rank transformation replaces values for y_k and x_j with their corresponding ranks. Specifically, the smallest value for a variable is assigned a rank of

1; next largest value is assigned a rank of 2; tied values are assigned their average rank; and so on up to the largest value, which is assigned a rank of *nS*. Use of the rank transformation converts a nonlinear but monotonic relationship between y_k and x_j to a linear relationship and produces rank (i.e., Spearman) correlations, rank regressions, SRRCs and PRCCs. In the presence of nonlinear but monotonic relationships between the x_j and y_k, the use of the rank transform can substantially improve the resolution of sensitivity analysis results (Table D-5). Additional information: Sect. 6.6.6 [D-42]; Sect. 6.6 [D-17]; [D-45].

Table D-5. Comparison of stepwise regression analyses with raw and rank-transformed data for variable *BRAALIC* in Figure D-8 (Table 8.8 [D-43]).

	Raw Data			**Rank-Transformed Data**		
Step[a]	**Variable**[b]	**SRC**[c]	***R*²**[d]	**Variable**[b]	**SRRC**[e]	***R*²**[d]
1	*ANHPRM*	0.562	0.320	*WMICDFLG*	–0.656	0.425
2	*WMICDFLG*	–0.309	0.423	*ANHPRM*	0.593	0.766
3	*WGRCOR*	–0.164	0.449	*HALPOR*	–0.155	0.802
4	*WASTWICK*	–0.145	0.471	*WGRCOR*	–0.152	0.824
5	*ANHBCEXP*	–0.120	0.486	*HALPRM*	0.143	0.845
6	*HALPOR*	–0.101	0.496	*SALPRES*	0.120	0.860
7				*WASTWICK*	–0.010	0.869

[a] Steps in stepwise regression analysis.

[b] Variables listed in order of selection in regression analysis.

[c] SRCs for variables in final regression model.

[d] Cumulative R^2 value with entry of each variable into regression model.

[e] SRRCs for variables in final regression model.

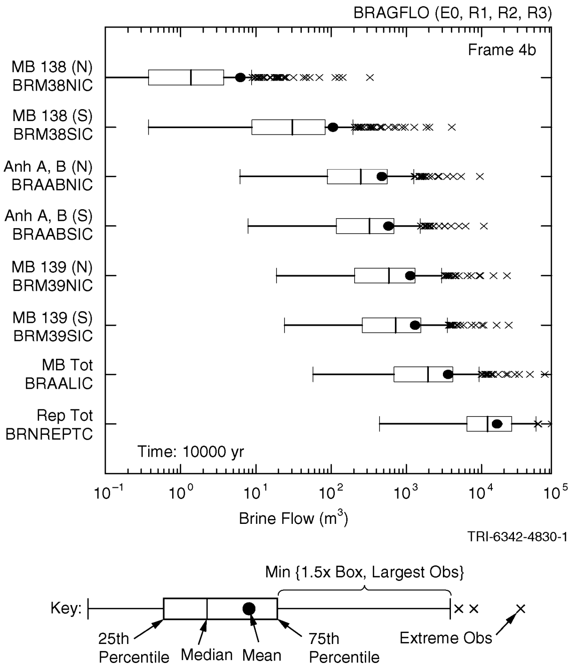


Figure D-8. Representation of uncertainty in scalar-valued analysis results with box plots (Figure 7.4 [D-43]).

Tests for Patterns Based on Gridding. Analyses on raw and rank-transformed data can fail when the underlying relationships between the x_j and y_k are nonlinear and nonmonotonic (Figure D-9). The scatterplot in Figure D-5(b) is for the pressure at 10,000 yr in Figure D-9(a) versus the uncertain variable *BHPRM*. The analyses with PRCCs summarized in Figure D-9(b) fail at later times because the pattern appearing in Figure D-5 is too complex to be captured with a regression analysis based on raw or rank-transformed data. An alternative analysis strategy for situations of this type is to place grids on the scatterplot for y_k and x_j and then perform various statistical tests to determine if the distribution of points across the grid cells appears to be nonrandom. Appearance of a nonrandom pattern indicates that x_j has an effect on y_k. Possibilities include (i) tests for common means and common distributions for values of y_k based on partitioning the range of x_j (Figure D-10(a)) and (ii) tests for common medians and no influence based on partitioning the ranges of x_j and y_k (Figure D-10(b)). Additional information: [D-46]; Sects. 6.6.8 and 6.6.9 [D-42]; Sects. 6.6 and 6.7 [D-17].

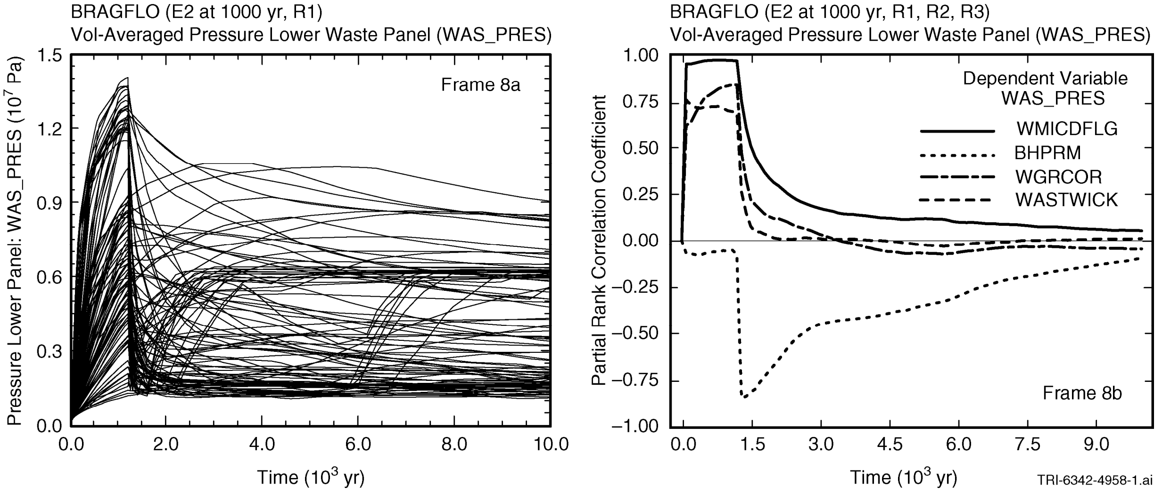


Figure D-9. Illustration of failure of a sensitivity analysis based on rank-transformed data: (a) Pressure as a function of time and (b) PRCC as a function of time (Figure 8.7 [D-43]).

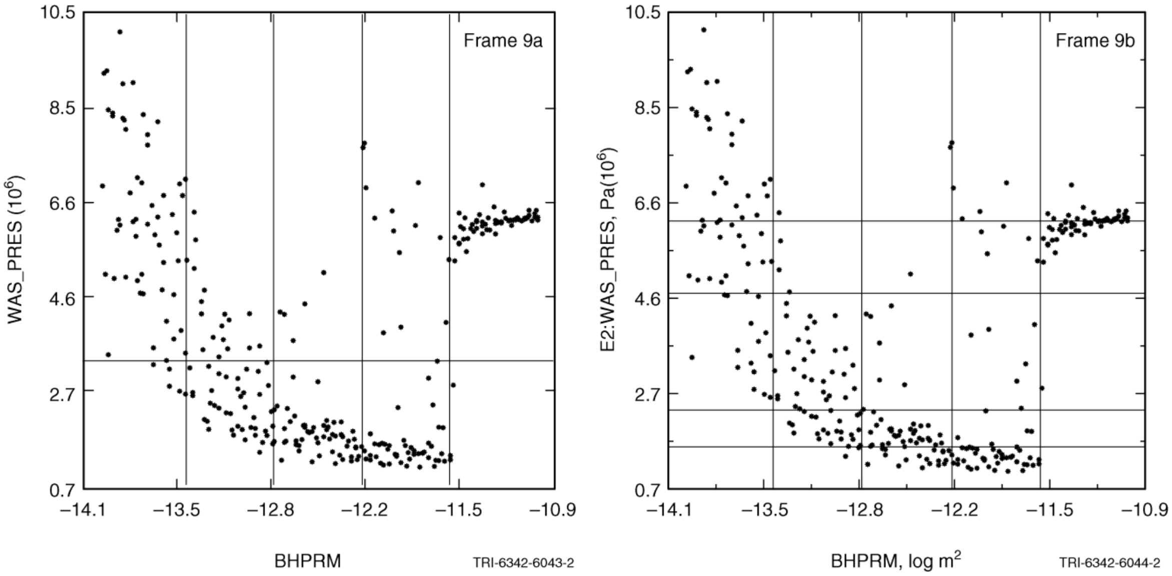


Figure D-10. Grids used to test for nonrandom patterns: (a) Partitioning of range of x_j for tests based on common means and common distributions and ranges of x_j and y_k for test based on common medians and common distributions (Figure 8.8 [D-43]), and (b) Partitioning of ranges of x_j and y_k for tests of no influence (Figure 8.9 [D-43]).

Nonparametric Regression. Nonparametric regression seeks more general models than those obtained by least squares regression and can succeed in situations such as the one illustrated in Figure D-5 where regression and correlation analysis based on raw and rank-transformed data fail. Nonparametric regression attempts to find models that are local in the approximation to the relationship between y_k and multiple x_js, and, as a result, are better at capturing complex nonlinear relationships than models obtained with traditional regression or rank regression. Nonparametric regression models can be constructed in a stepwise manner with incremental changes in R^2 values with the addition of successive variables to the model providing an indication of variable importance. Additional information: Sect. 6.8 [D-17]; [D-47] to [D-52].

Tests for Patterns Based on Distance Measures. Tests based on distance measures consider relationships within the scatterplot for y_k and x_j such as the distribution of distances between nearest neighbors and provide a way to identify nonrandom relationships between y_k and x_j. A positive feature of these tests is the avoidance of the problem of defining an appropriate grid as is the case with grid-based methods. Additional information: Sect. 6.11 [D-17]; [D-53] to [D-56].

Trees. Tree-based sensitivity analyses search for relationships between y_k and multiple x_js by successively subdividing the sample elements $\mathbf{x}_i$ on the basis of observed effects of individual x_js on y_k. Additional information: [D-57, D-58].

Two-Dimensional Kolmogorov-Smirnov Test. The two-dimensional Kolmogorov-Smirnov test provides a way to test for nonrandom patterns in the scatterplot for y_k and x_j that does not require the imposition of a grid. Additional information: Sect. 6.10 [D-17]; [D-59] to [D-61].

Squared Differences of Ranks. The squared difference of ranks procedure seeks to identify the presence of nonlinear relationship between y_k and x_j and is based on squared differences of consecutive ranks of y_k when the values of y_k have been ordered by the corresponding values of x_j. Additional information: Sect. 6.9 [D17]; [D-62].

Top-Down Concordance with Replicated Samples. This procedure uses the top-down coefficient of concordance and replicated (i.e., independently generated) samples for sensitivity analysis with some appropriate technique to rank variable importance for each sample. The top-down coefficient is then used to identify important variables by seeking variables with similar rankings across all replicates. Additional information: Sect. 6.12 [D-17]; [D-63, D-64].

Variance Decomposition. The variance decomposition procedure proposed by Sobol' and others is formally defined by high-dimensional integrals involving the x_j and $y_k(\mathbf{x})$. This procedure provides a decomposition of the variance $V(y_k)$ of y_k in terms of the contributions V_j of individual x_js to $V(y_k)$ and also the contributions of various interactions between the x_j to $V(y_k)$. In practice, the indicated decomposition is obtained with sampling-based methods. Two samples from $\mathbf{x}$ of size nS are required to estimate all V_j; $nX + 2$ samples of size nS are required to estimate all V_j and also the contributions of each of the x_js and its interactions with other elements of $\mathbf{x}$ to $V(y_k)$. This procedure is very appealing but can be computationally demanding as more samples and probably larger samples are required than with other sampling-based approaches to sensitivity analysis. Software for sampling-based variance decomposition is available as part of the SIMLAB package [D-65]. Additional information: Sect. 6.13 [D-17]; [D-65] to [D-70].

Appendix D - References

[D-1] Parry GW, Winter PW. Characterization and Evaluation of Uncertainty in Probabilistic Risk Analysis. *Nuclear Safety* 1981;22(1):28-42.

[D-2] Apostolakis G. The Concept of Probability in Safety Assessments of Technological Systems. *Science* 1990;250(4986):1359-1364.

[D-3] Helton JD. Treatment of Uncertainty in Performance Assessments for Complex Systems. *Risk Analysis* 1994;14(4):483-511.

[D-4] Hoffman FO, Hammonds JS. Propagation of Uncertainty in Risk Assessments: The Need to Distinguish Between Uncertainty Due to Lack of Knowledge and Uncertainty Due to Variability. *Risk Analysis* 1994;14(5):707-712.

[D-5] Helton JC, Burmaster DE. Guest Editorial: Treatment of Aleatory and Epistemic Uncertainty in Performance Assessments for Complex Systems. *Reliability Engineering and System Safety* 1996;54(2-3):91-94.

[D-6] Paté-Cornell ME. Uncertainties in Risk Analysis: Six Levels of Treatment. *Reliability Engineering and System Safety* 1996;54(2-3):95-111.

[D-7] Helton JD. Uncertainty and Sensitivity Analysis in the Presence of Stochastic and Subjective Uncertainty. *Journal of Statistical Computation and Simulation* 1997;57(1-4):3-76.

[D-8] Helton JD. Mathematical and Numerical Approaches in Performance Assessment for Radioactive Waste Disposal: Dealing with Uncertainty. In: EM Scott, ed. *Modeling Radioactivity in the Environment*. New York, NY: Elsevier Science, 2003:353-390.

[D-9] Helton JC, Anderson DR, Jow H-N, Marietta MG, Basabilvazo G. Conceptual Structure of the 1996 Performance Assessment for the Waste Isolation Pilot Plant. *Reliability Engineering and System Safety* 2000;69(1-3):151-165.

[D-10] Helton JD. Uncertainty and Sensitivity Analysis Techniques for Use in Performance Assessment for Radioactive Waste Disposal. *Reliability Engineering and System Safety* 1993;42(2-3):327-367.

[D-11] Hamby DM. A Review of Techniques for Parameter Sensitivity Analysis of Environmental Models. *Environmental Monitoring and Assessment* 1994;32(2):135-154.

[D-12] Saltelli A, Chan K, E.M. Scott (eds). *Sensitivity Analysis*. New York, NY: Wiley, 2000.

[D-13] Frey HC, Patil SR. Identification and Review of Sensitivity Analysis Methods. *Risk Analysis* 2002;22(3):553-578.

[D-14] Cacuci DG, Ionescu-Bujor M. A Comparative Review of Sensitivity and Uncertainty Analysis of Large-Scale Systems--II: Statistical Methods. *Nuclear Science and Engineering* 2004;147(3):204-217.

[D-15] Ionescu-Bujor M, Cacuci DG. A Comparative Review of Sensitivity and Uncertainty Analysis of Large-Scale Systems--I: Deterministic Methods. *Nuclear Science and Engineering* 2004;147(3):189-2003.

[D-16] Saltelli A, Ratto M, Tarantola S, Campolongo F. Sensitivity Analysis for Chemical Models. *Chemical Reviews* 2005;105(7):2811-2828.

[D-17] Helton JC, Johnson JD, Sallaberry CJ, Storlie CB. Survey of Sampling-Based Methods for Uncertainty and Sensitivity Analysis. *Reliability Engineering and System Safety* 2006;91(10-11):1175-1209.

[D-18] Mosleh A, Siu N, Smidts C, Liu D. *Proceedings of Workshop I in Advanced Topics in Risk and Reliability Analysis, Model Uncertainty: Its Characterization and Quantification*. NUREG/CP-0138. Washington, D.D.: U.S. Nuclear Regulatory Commission 1994.

[D-19] Hora SC, Iman RL. Expert Opinion in Risk Analysis: The NUREG-1150 Methodology. *Nuclear Science and Engineering* 1989;102(4):323-331.

[D-20] Meyer MA, Booker JM. *Eliciting and Analyzing Expert Judgment: A Practical Guide*. New York, NY: Academic Press, 1991.

[D-21] Keeney RL, Winterfeldt DV. Eliciting Probabilities from Experts in Complex Technical Problems. *IEEE Transactions on Engineering Management* 1991;38(3):191-201.

[D-22] Thorne MC, Williams MMR. A Review of Expert Judgment Techniques with Reference to Nuclear Safety. *Progress in Nuclear Safety* 1992;27(2-3):83-254.

[D-23] Thorne MD. The Use of Expert Opinion in Formulating Conceptual Models of Underground Disposal Systems and the Treatment of Associated Bias. *Reliability Engineering and System Safety* 1993;42(2-3):161-180.

[D-24] Budnitz RJ, Apostolakis G, Boore DM, Cluff LS, Coppersmith KJ, Cornell CA, Morris PA. Use of Technical Expert Panels: Applications to Probabilistic Seismic Hazard Analysis. *Risk Analysis* 1998;18(4):463-469.

[D-25] Ayyub BM. *Elicitation of Expert Opinions for Uncertainty and Risks*. Boca Raton, FL: CRC Press 2001.

[D-26] Cooke RM, Goossens LHJ. Expert Judgement Elicitation for Risk Assessment of Critical Infrastructures. *Journal of Risk Research* 2004;7(6):643-656.

[D-27] Cooke R. *Experts in Uncertainty: Opinion and Subjective Probability in Science*. Oxford; New York: Oxford University Press 1991.

[D-28] Garthwaite PH, Kadane JB, O'Hagan A. Statistical Methods for Eliciting Probability Distributions. *Journal of the American Statistical Association* 2005;100(470):680-700.

[D-29] Kaplan S, Garrick BJ. On the Quantitative Definition of Risk. *Risk Analysis* 1981;1(1):11-27.

[D-30] McKay MD, Beckman RJ, Conover WJ. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics* 1979;21(2):239-245.

[D-31] Helton JC, Davis FJ. Latin Hypercube Sampling and the Propagation of Uncertainty in Analyses of Complex Systems. *Reliability Engineering and System Safety* 2003;81(1):23-69.

[D-32] Iman RL, Conover WJ. A Distribution-Free Approach to Inducing Rank Correlation Among Input Variables. *Communications in Statistics: Simulation and Computation* 1982;B11(3):311-334.

[D-33] Iman RL, Davenport JM. Rank Correlation Plots for Use with Correlated Input Variables. *Communications in Statistics: Simulation and Computation* 1982;B11(3):335-360.

[D-34] SNL (Sandia National Laboratories). *Total System Performance Assessment Model/Analysis for the License Application*.MDL-WIS-PA-000005 Rev 00, AD 01.Las Vegas, NV: U.S. Department of Energy Office of Civilian Radioactive Waste Management 2008.

[D-35] Helton JC, Sallaberry CJ. Conceptual Basis for the Definition and Calculation of Expected Dose in Performance Assessments for the Proposed High-Level Radioactive Waste Repository at Yucca Mountain, Nevada. *Reliability Engineering and System Safety* 2009;94:677-698.

[D-36] Helton JC, Iuzzolino HJ. Construction of Complementary Cumulative Distribution Functions for Comparison with the EPA Release Limits for Radioactive Waste Disposal. *Reliability Engineering and System Safety* 1993;40(3):277-293.

[D-37] Helton JC, Breeding RJ. Calculation of Reactor Accident Safety Goals. *Reliability Engineering and System Safety* 1993;39(2):129-158.

[D-38] Helton JC, Sallaberry CJ. Treatment of Uncertainty in Performance Assessments for the Geological Disposal of Radioactive Waste. In: J Ahn, M Apted, eds. *Geological Repositories for Safe Disposal of Spent Fuels and Radioactive Materials: Advanced Technologies*. Cambridge, UK: Woodhead Publishing, 2010.

[D-39] Helton JC, Marietta MG. Special Issue: The 1996 Performance Assessment for the Waste Isolation Pilot Plant. *Reliability Engineering and System Safety* 2000;69(1-3):1-451.

[D-40] U.S. DOE (U.S. Department of Energy). *Title 40 CFR Part 191 Compliance Certification Application for the Waste Isolation Pilot Plant*.DOE/CAO-1996-2184, Vols. I-XXI. Carlsbad, NM: U.S. Department of Energy, Carlsbad Area Office, Waste Isolation Pilot Plant 1996.

[D-41] Helton JC, Martell M-A, Tierney MS. Characterization of Subjective Uncertainty in the 1996 Performance Assessment for the Waste Isolation Pilot Plant. *Reliability Engineering and System Safety* 2000;69(1-3):191-204.

[D-42] Helton JC, Davis FJ. Sampling-Based Methods. In: A Saltelli, K Chan, E.M. Scott, eds. *Sensitivity Analysis*. New York, NY: Wiley. pp. 101-153, 2000.

[D-43] Helton JC, Davis FJ. *Sampling-Based Methods for Uncertainty and Sensitivity Analysis*.SAND99-2240.Albuquerque, NM: Sandia National Laboratories 2000.

[D-44] Cooke RM, J.M. van Noortwijk. Graphical Methods. In: A Saltelli, K Chan, E.M. Scott, eds. *Sensitivity Analysis*. New York, NY: Wiley, 2000:pp. 245-264.

[D-45] Iman RL, Conover WJ. The Use of the Rank Transform in Regression. *Technometrics* 1979;21(4):499-509.

[D-46] Kleijnen JPC, Helton JD. Statistical Analyses of Scatterplots to Identify Important Factors in Large-Scale Simulations, 1: Review and Comparison of Techniques. *Reliability Engineering and System Safety* 1999;65(2):147-185.

[D-47] Storlie CB, Helton JD. *Multiple Predictor Smoothing Methods for Sensitivity Analysis*.SAND2006-4693.Albuquerque, NM: Sandia National Laboratories 2006.

[D-48] Hastie TJ, Tibshirani RJ. *Generalized Additive Models*. London: Chapman & Hall, 1990.

[D-49] Simonoff JS. *Smoothing Methods in Statistics*. New York, NY: Springer-Verlag, 1996.

[D-50] Bowman AW, Azzalini A. *Applied Smoothing Techniques for Data Analysis*. Oxford: Clarendon, 1997.

[D-51] Storlie CB, Helton JD. Multiple Predictor Smoothing Methods for Sensitivity Analysis: Description of Techniques. *Reliability Engineering and System Safety* To Appear.

[D-52] Storlie CB, Helton JD. Multiple Predictor Smoothing Methods for Sensitivity Analysis: Example Results. *Reliability Engineering and System Safety* To Appear.

[D-53] Ripley BD. Tests of Randomness for Spatial Point Patterns. *Journal of the Royal Statistical Society* 1979;41(3):368-374.

[D-54] Diggle PJ, Cox TF. Some Distance-Based Tests of Independence for Sparsely-Sampled Multivariate Spatial Point Patterns. *International Statistical Review* 1983;51(1):11-23.

[D-55] Zeng G, Dubes RD. A Comparison of Tests for Randomness. *Pattern Recognition* 1985;18(2):191-198.

[D-56] Assunçao R. Testing Spatial Randomness by Means of Angles. *Biometrics* 1994;50:531-537.

[D-57] Mishra S, Deeds NE, B.S. RamaRao. Application of Classification Trees in the Sensitivity Analysis of Probabilistic Model Results. *Reliability Engineering and System Safety* 2003;79(2):123-129.

[D-58] Breiman L, Friedman JH, Olshen RA, Stone CJ. *Classification and Regression Trees*. Belmont, CA: Wadsworth Intl, 1984.

[D-59] Peacock JA. Two-Dimensional Goodness-Of-Fit Testing in Astronomy. *Monthly Notices of the Royal Astronomical Society* 1983;202(2):615-627.

[D-60] Fasano G, Franceschini A. A Multidimensional Version of the Kolmogorov-Smirnov Test. *Monthly Notices of the Royal Astronomical Society* 1987;225(1):155-170.

[D-61] Garvey JE, Marschall EA, Wright RA. From Star Charts to Stoneflies: Detecting Relationships in Continuous Bivariate Data. *Ecology* 1998;79(2):442-447.

[D-62] Hora SC, Helton JD. A Distribution-Free Test for the Relationship Between Model Input and Output when Using Latin Hypercube Sampling. *Reliability Engineering and System Safety* 2003;79(3):333-339.

[D-63] Iman RL, Conover WJ. A Measure of Top-Down Correlation. *Technometrics* 1987;29(3):351-357.

[D-64] Helton JC, Davis FJ, Johnson JD. A Comparison of Uncertainty and Sensitivity Analysis Results Obtained with Random and Latin Hypercube Sampling. *Reliability Engineering and System Safety* 2005;89(3):305-330.

[D-65] Saltelli A, Tarantola S, Campolongo F, Ratto M. *Sensitivity Analysis in Practice*. New York, NY: Wiley, 2004.

[D-66] Li G, Rosenthal C, Rabitz H. High-Dimensional Model Representations. *The Journal of Physical Chemistry* 2001;105(33):7765-7777.

[D-67] Rabitz H, Alis OF. General Foundations of High-Dimensional Model Representations. *Journal of Mathematical Chemistry* 1999;25(2-3):197-233.

[D-68] Saltelli A, Tarantola S, Chan KP-S. A Quantitative Model-Independent Method for Global Sensitivity Analysis of Model Output. *Technometrics* 1999;41(1):39-56.

[D-69] Sobol' IM. Sensitivity Estimates for Nonlinear Mathematical Models. *Mathematical Modeling & Computational Experiment* 1993;1(4):407-414.

[D-70] Cukier RI, Levine HB, Shuler KE. Nonlinear Sensitivity Analysis of Multiparameter Model Systems. *Journal of Computational Physics* 1978;26(1):1-42.

[D-71] Helton JD. Uncertainty and Sensitivity Analysis for Models of Complex Systems. In: F Graziani, ed. *Computational Methods in Transport: Verification and Validation*. New York, NY: Springer-Verlag, 2008:207-208.

APPENDIX E. PILOT STUDY PROBLEM STATEMENT

Sandia developed a pilot study problem statement report for the xLPR computational task group for Version 1.0 xLPR framework. This document is attached below and provides a description of the background, the pilot study CM system, model verification, xLPR base case analysis, sensitivity analysis, post-processing analysis, uncertainty, and sensitivity analysis completed using Version 1.0 of the model framework.

xLPR Pilot Study Model Problem Statements

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 DM, pressurizer surge nozzle welds degraded by 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.

E.1. 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

E.1.1 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_002.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 spreadsheets 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 SharePoint 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 re-named 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 production runs directory on the xLPR CM SharePoint 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 can be or will be used in the pilot program reports. Due to file size limitations in SharePoint, 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: INDEPENDENT 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 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.

E.2. 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.

E.2.1. 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.

Deterministic Analysis #1: Single Crack at t = 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 is included with the controlled version of the inputs spreadsheet for xLPR.

Deterministic Analysis #2: Three Cracks at t = 0 years, with no mitigation. The same problem as the first deterministic analysis but with three cracks instead of just one. The three cracks are the same size as before. Their respective location are theta = 0 rad, theta = 0.6 rad and theta = -1 rad. The input deck for this case is included with the controlled version of the inputs spreadsheet for xLPR.

E.2.2. Stability Testing

Model stability testing activities for xLPR include two types of stability tests: statistical stability and temporal stability. Collectively, these three tests are referred to as model stability testing.

a. 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 seed 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 uncertainty.

b. 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.

E.3. xLPR Base Case Analysis

E.3.1. 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 size of 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 WRS, therefore for the base case the WRSs assumed are shown in Figure E-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 DM weld.

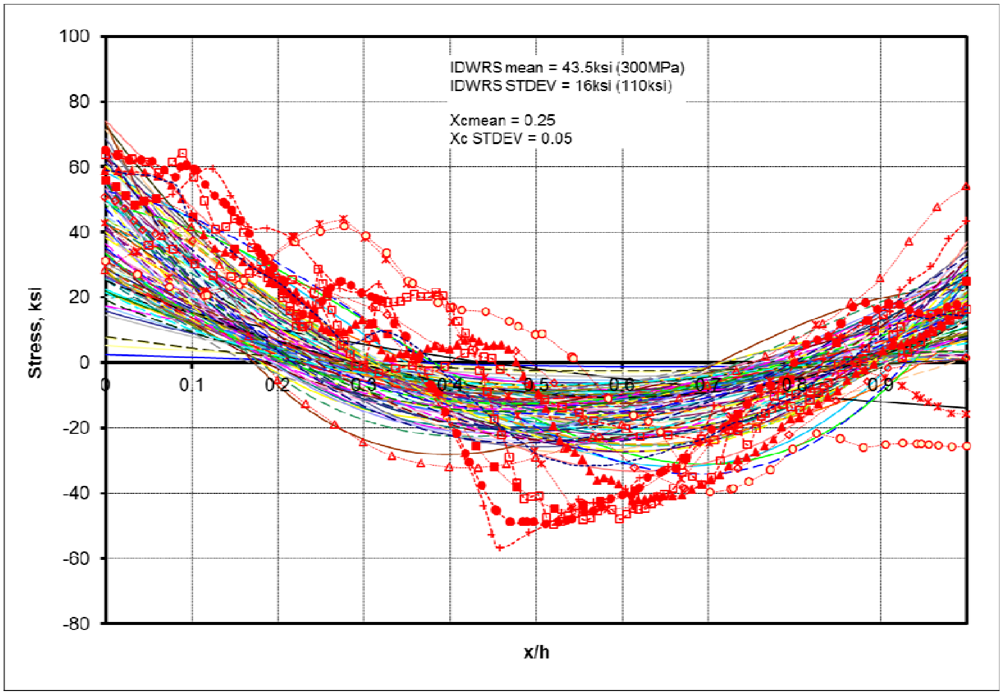


Figure E-1. Base case WRS.

E.3.2. xLPR Pilot Study Outputs

E.3.2.1. 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 not a single output but several intermediate outputs as well. Since each output depends on its aleatory and epistemic set, it can be analyzed in several ways, depending whether one integrates over aleatory uncertainty, epistemic uncertainty, both or neither:

a. Expected value over aleatory uncertainty: these values are obtained by averaging over aleatory uncertainty. As we use simple Monte Carlo sampling techniques for aleatory uncertainty, the averaging is a classical sum divided by the number of parameters, and then including correction due to conditionality. No critical failure will 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 the following way: Expected value = P(no crack)*Value_if_no_crack + P(at least one crack) * Value_if_at_least_one_crack.

b. 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.

c. Aleatory variation for a fixed epistemic set: Once one epistemic set is selected, aleatory uncertainty is represented. 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.

d. Epistemic variation for a fixed aleatory set: A symmetrical analysis of the previous one, it is less used as 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 a) and c) will be considered here. Method a) will be complemented with estimates of mean and quantiles of the expected values.

E.3.2.2. 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.

- Time-dependent crack depth (expected over aleatory uncertainty) for any relevant crack
- Time-dependent half crack length (expected over aleatory uncertainty) for any relevant crack
- Time-dependent fractional surface area cracked (expected over aleatory uncertainty)
- Time-dependent stress intensity (expected over aleatory uncertainty) for any relevant crack
- Scatterplot stress-intensity vs. crack area for specific time (10 yr, 30 yr, 60 yr) (expected over aleatory uncertainty)
- Time-dependent PND (expected over aleatory uncertainty) for any relevant crack
- Average duration of SC (over aleatory uncertainty and over all cracks)
- Time-dependent leak rate (expected over aleatory uncertainty) for any relevant crack.
- Total time-dependent leak rate (expected over aleatory uncertainty)
- First leakage probability as a function of time
- COA = 1-inch equivalent break diameter (506.71 mm^2) probability as a function of time
- COA = 3-inch equivalent break diameter (4,560.37 mm^2) probability as a function of time
- 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.

E.4. 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.

E.4.1. 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 ID stresses 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 that the safe end weld did not affect the stresses in the DM weld. This case will consider a short safe end length. For the safe end length considered, the distribution of WRS can be seen in Figure E-2. The symbols in the figure represent the detailed finite element analysis predictions of WRS, while the lines represent the fit to that data using the WRS model in xLPR. A unique distribution for axial stress component for the epistemic parameter S0_WRS and Xc are shown in Figure E-2. It is assumed that the distribution is normal and

- S0_WRS: Mean = -16.2 MPa, Stdev = 117 MPa, max = 300 MPa, min = -300 MPa
- Xc: Mean = 0.18, Stdev = 0.036, max = 0.5, min= 0.1

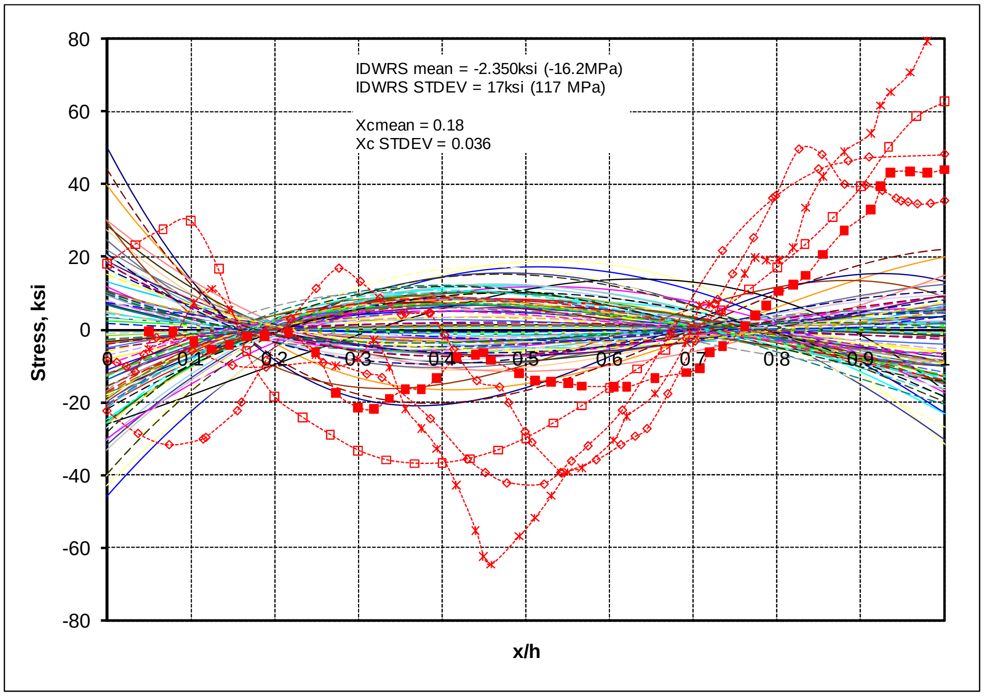


Figure E-2. WRS distribution for surge nozzle with safe end weld.

E.4.2. 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 10 years
2. Mitigation time 20 years
3. Mitigation time 40 years

The distribution of WRS to be used for the mitigation is shown in Figure E-3. For these cases a normal distribution should be assumed with:

- Sigma0_wrs_mitigated: Mean = -344.75 MPa, Stdev = 34 MPa, min = -447 MPa, max = -242 MPa
- Xc_mitigated: Mean = 0.38, Stdev = 0.038, min = 0.26, max = 0.5

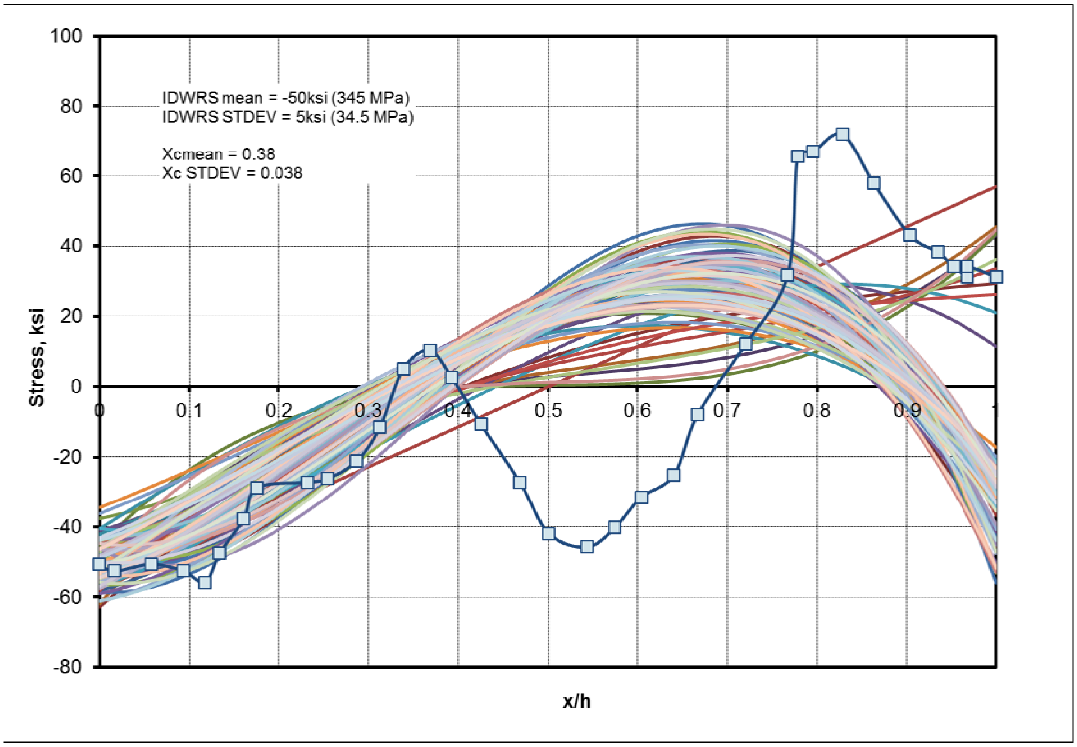


Figure E-3. Mitigated WRS for beta sensitivity analyses.

E.4.3. 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.

E.4.4. 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 effect.

E.4.5. 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 run using the safe end sensitivity case.

E.5. 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 failures. The desired output, defined in Section E.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 SNL for the xLPR Pilot Study.

E.5.1. Leak Detection Capability

In order to demonstrate the leak detection capability of the code, the output of select cases will be analyzed to demonstrate the effect on the output probabilities (last four bullets of Section E.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 supposed that once a leak is detected, the weld is replaced and will not fail again. Therefore all calculations are done supposing 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 SharePoint site at CM > Beta Model Dev > Modules > TRANSFORMERS v1.0. See the associated documentation for details on the features of this module.

E.5.2. 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 E.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 can be handled in a similar way. Once again, all calculations are done supposing that nothing is detected during inspection while, at each time step, a PND is estimated. As a post-processing task, results will then be corrected in consequence, as follows:

Once an inspection is schedule at a given time step, the PND is reported. It gives the probability that nothing changes in the analysis. The POD (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 same quality (randomly select another future, including future with no cracks at all),

Probability of failure, as well as other outputs of interest (last four bullets of Section E.3.2.2) will be corrected in consequence 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.

E.5.3. Uncertainty and Sensitivity Analysis

Uncertainty and sensitivity analysis 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 V&V tool, highlighting any strange behavior.

Uncertainty analysis will consist of classical statistical techniques such as CDF, CCDF representation and calculation of mean and quantiles and has been described in more detail in Section E.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 to detect monotonic relationship between input and output. It will include:

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

The parameter sensitivity analyses will be completed using the methodology and codes developed by SNL 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.

DISTRIBUTION

External Distribution

1 Aladar Csontos
Division of Engineering
Office of Nuclear Regulatory Research U.S. Nuclear Regulatory Commission
Mailstop: C-5A24M
Washington, DC 20555-0001
United States of America

10 David Rudland
Office of Nuclear Regulatory Research
Division of Engineering 21 Church Street, Room 5C04
Mail Stop: CSB-5CA24
Rockville, MD 20852
United States of America

2 Craig Harrington
Electric Power Research Institute (EPRI)
3010 LBJ Freeway, Suite 300
Dallas, TX 75234
Send to: 1110 Cherrywood Drive, Cleburne, TX 76033

Sandia National Laboratories Internal Distribution

1	MS0748	Randall O. Gauntt. Mgr.	6232
1	MS0748	Donald A. Kalinich	6232
15	MS0749	Patrick Mattie	6232
1	MS1369	Cedric Sallaberry	6224
1	MS0748	Jon Helton	1545
1	MS0899	Technical Library (electronic only)	9536

Sandia National Laboratories