Proceedings of the ASME 2010 Pressure Vessels & Piping Division / K-PVP Conference
PVP2010
July 18-22, 2010, Bellevue, Washington, USA

PVP2010-25963

DEVELOPMENT OF COMPUTATIONAL FRAMEWORK AND ARCHITECTURE FOR EXTREMELY LOW PROBABILITY OF RUPTURE (XLPR) CODE

D. Rudland[*†]
US Nuclear Regulatory Commission
Office of Nuclear Regulatory Research
Washington, DC 20555-0001

P. Mattie
Sandia National Laboratory
Albuquerque, New Mexico

R. Kurth
Battelle
Columbus, Ohio

H. Klasky
Oak Ridge National Laboratory
Oak Ridge, TN

B. Bishop
Westinghouse
Pittsburgh, PA

D. Harris
Structural integrity Associates
San Jose, CA

ABSTRACT

The Nuclear Regulatory Commission (NRC) Standard Review Plan (SRP) 3.6.3 describes Leak-Before-Break (LBB) assessment procedures that can be used to demonstrate compliance with the 10CFR50 Appendix A, GDC-4 requirement that primary system pressure piping exhibit an extremely low probability of rupture. SRP 3.6.3 does not allow for assessment of piping systems with active degradation mechanisms, such as Primary Water Stress Corrosion Cracking (PWSCC) which is currently occurring in systems that have been granted LBB exemptions.

Along with the existing qualitative steps to assuring safety in LBB lines with PWSCC, the NRC staff, working cooperatively with the nuclear industry through a memorandum of understanding, is developing a new, modular based, comprehensive piping system assessment methodology to directly demonstrate compliance with the regulations. This tool, called xLPR (eXtremely Low Probability of Rupture), would properly model the effects and uncertainties of both active degradation mechanisms and the associated mitigation activities. The tool will be comprehensive with respect to known challenges, vetted with respect to scientific adequacy of models and inputs, flexible enough to permit analysis of a variety of in-service situations and adaptable such as to accommodate evolving and improving knowledge.

A multi-year project has begun that will first focus on the development of a viable method and approach to address the effects of PWSCC as well as define the requirements necessary for a modular-based assessment tool. A prototype xLPR model and pilot study case is first being conducted 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 pilot study phase is focusing on PWSCC in pressurizer surge nozzles. Later development phases will broaden the scope of xLPR to all primary piping systems in pressurized and boiling water reactors (PWR and BWR), using an incremental approach that incorporates the design requirements and lessons learned from previous iterations.

This paper specifically examines the prototype xLPR model and includes the methods and approach used to couple existing models and software as modules within a probabilistic software framework. Since the pilot study is currently still ongoing, this

[*] Corresponding author, david.rudland@nrc.gov
[†] The views expressed herein are those of the author and do not represent an official position of the USNRC

paper provides a discussion of the current status and plans to move forward after the pilot study is complete.

INTRODUCTION

10 CFR Part 50, Appendix A, General Design Criteria (GDC) 4 states, in part, that the dynamic effects associated with postulated reactor coolant system pipe ruptures may be excluded from the design basis when analyses reviewed and approved by the 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. Based on the terminology of GDC 4, this effort is titled eXtremely Low Probability of Rupture (xLPR). Development of the xLPR methodology and the corresponding software tool will involve many challenging technical decisions, modeling judgments, and sensitivity analyses.

The development of a sophisticated probabilistic software tool that meets quality assurance (QA) and technical requirements is a daunting task. The management structure, the probabilistic framework, and data handling are just a few of the issues that need to be addressed early on in the software development effort. In order to meet this need, a pilot study is being conducted. This study will demonstrate the feasibility of the proposed NRC-industry cooperative process for developing a xLPR to address degradation in piping system. It will also be used to demonstrate the appropriate probabilistic framework for calculating the probability of rupture for a surge nozzle dissimilar metal weld. 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.

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.

This paper discusses the current status of the xLPR program from a computational point of view. Discussions from other xLPR groups can be found in companion papers [1, 2, 3]. After a description of the pilot study details, the deterministic modules and uncertainty characterization for the code will be briefly discussed. The computational framework will be discussed as well as the approach used to couple existing models and software as modules. After a discussion of the ongoing configuration management activities, the paper will conclude with plans for xLPR beyond the pilot study.

xLPR PILOT STUDY DESCRIPTION

As described above, a xLPR pilot study is being conducted to demonstrate the feasibility of the proposed developmental process and framework for a probabilistic code to address degradation mechanisms in piping system safety assessments. The pilot study will address the specific issue of assessing the probability of rupture of dissimilar metal (DM), pressurizer surge nozzle welds degraded by PWSCC, particularly those previously assessed [4] 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 pilot study consists of an alpha and beta phase of code development. Within the alpha phase, the computational group’s responsibility is to develop the probabilistic code framework from an open source and commercial code perspective. To do this task in a timely manner, the computational group also developed the modules to conduct the needed fracture mechanics analyses. In the beta phase, the experts within the models group [2], will be providing peer reviewed modules to replace those used by the computational group in the alpha phase. However, the overall framework structure will be consistent between the alpha and beta phases on the program.

The basic flow for the xLPR pilot study program is shown in Figure 1. The propagation of uncertainties is initially being handled by a nested loop structure where epistemic and aleatory uncertainties are propagated separately through the model. Details of the uncertainty propagation strategy are given later in this paper. The deterministic kernel for the initial xLPR code is embedded in a time loop structure whose details are shown in Figure 2. The time loop consists of deterministic models for crack initiation, crack growth, crack stability, leakage, in-service inspection, and PWSCC mitigation. The details of the alpha version of the models listed are given later in this paper. The details of the models used in the beta version are given elsewhere [2].

In developing the computational implementation for the initial xLPR version, both open-source and commercial software is being considered. Working in parallel, two unique codes are being developed to demonstrate the advantages of each framework. The commercial software GoldSim is being used to develop the commercial software version of the xLPR Model, while the code SIAM-PFM is being developed with only open source code to demonstrate this platform. The details of each framework code are given later in this paper.

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

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

These results will include a comparison of results with and without the effects of inspection and pre-emptive PWSCC mitigation. Sensitivity studies will also be carried out to exercise, verify and debug the code.

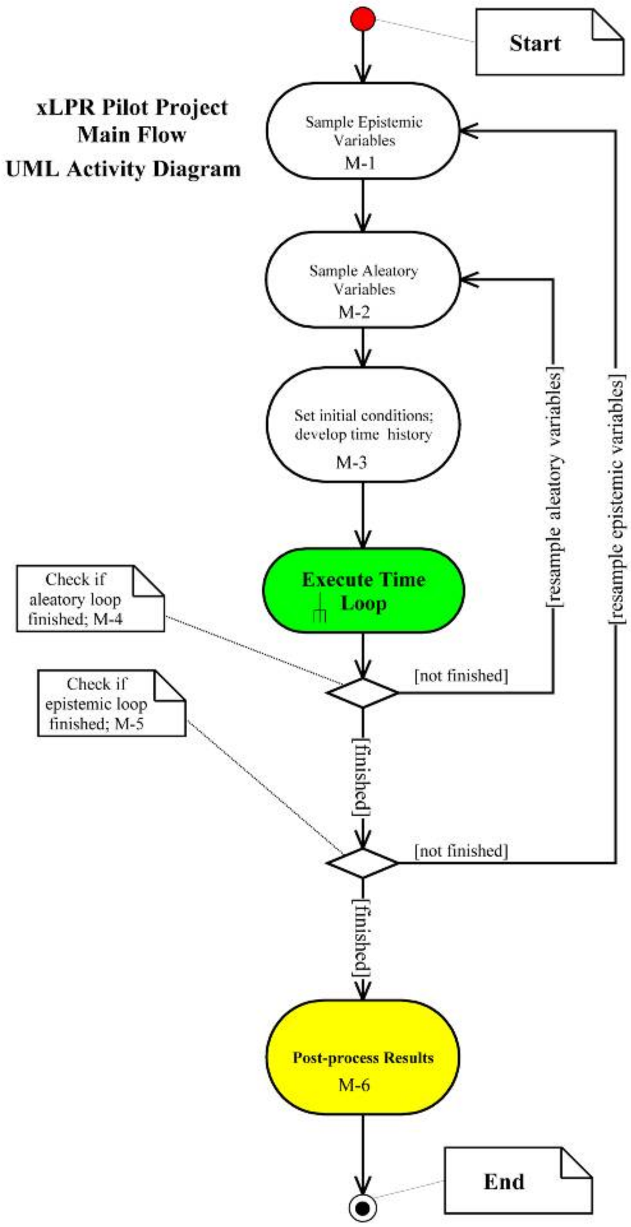

Figure 1 xLPR pilot study main flow

The programmatic outcome of the pilot study is intended to be an optimized development process for the general tool for assessing primary piping system safety. 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.

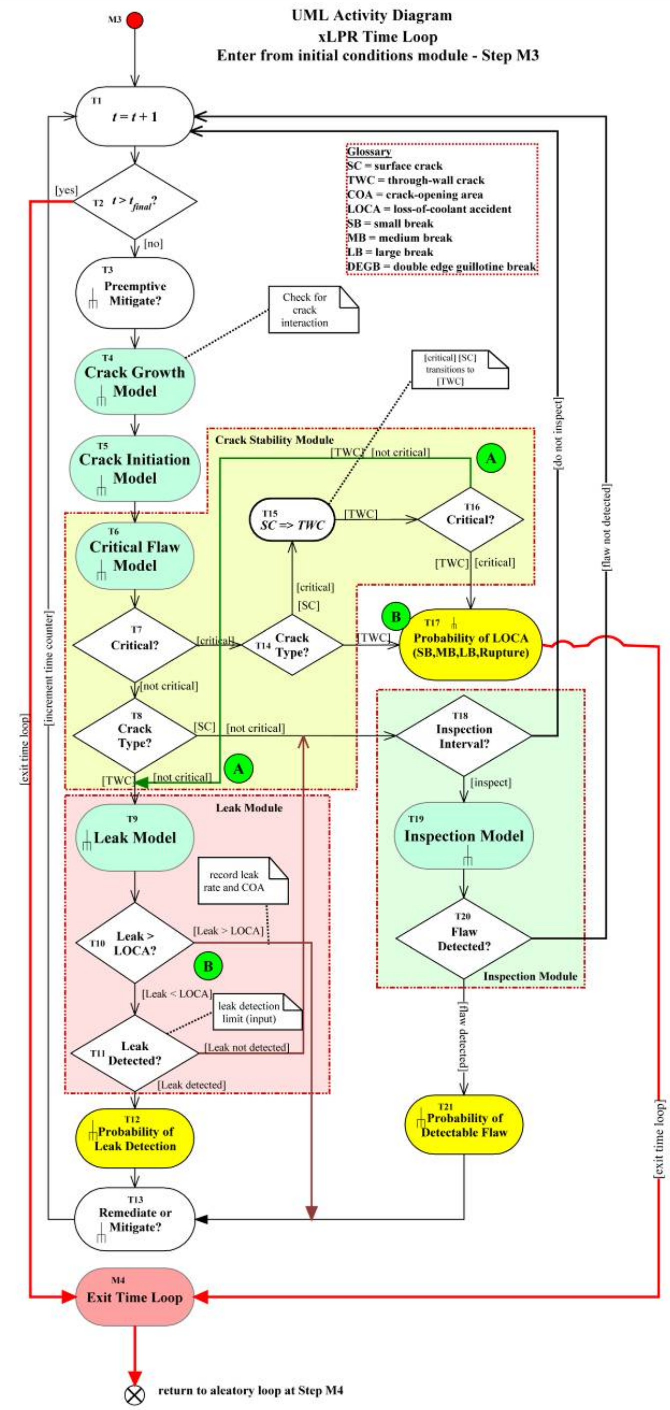

Figure 2 xLPR time loop details

MODULE DESCRIPTIONS FOR ALPHA PHASE

The xLPR software is being constructed as a collection of modules and this structure will be followed in the alpha version. Descriptions of the modules for the alpha version are provided below. The alpha version concentrates on initiation and growth of primary water stress corrosion cracks (PWSCC) in a dissimilar metal pressurizer surge nozzle weld. Hence the degradation mechanism, materials, environment and pipe size are fixed.

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 alpha 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 [4]. 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 and their respective times of initiation 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 number of cracks is modeled with a Poisson distribution, with the parameter of the distribution estimated from relevant service experience. The initiation module provides the number of initiated cracks and their time of initiation. Adjustments of the Poisson parameter for changes in temperature and stress are provided, so that effects of mitigation can be analyzed.

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

Crack Placement Module: The number of cracks has been defined in the crack initiation module, and the placement module prescribes the (circumferential) locations of initiated cracks. Three categories of possible locations are considered: (i) if the local ID axial stress is above the as-welded yield strength, this location will always be considered a crack initiation location; (ii) if the local ID axial stress is compressive, this location will never be considered for crack initiation; and (iii) if the local ID axial stress is between zero and the as-welded yield strength, this location will be considered for initiation with a uniformly distributed location within this region. The distance between two cracks will always be greater than two times the depth of the deeper crack. If it is attempted to place a newly initiated crack in a location where a crack already resides, it is placed in another location. If the allowable circumferential area is filled with cracks, then no further cracks are allowed.

Crack Coalescence Module: Initiated cracks can coalesce as the cracks are growing. The possibility of coalescence is checked after each time step. When the distance between two surface cracks is less than two times the depth of the deeper of the two cracks, the cracks will coalesce to form a single crack of depth equal to the deepest of the two and a length of the sum of the two lengths plus the distance between the closest tips. Coalescence criteria for two through-wall cracks and a part-through and through-wall crack are also included.

Stress Intensity Factors: Stress intensity factor solutions for part-through circumferential semi-elliptical cracks [5] and straight-fronted through-wall circumferential cracks [6] are included. The part-through solutions consider through-wall stress distributions described by a fourth order polynomial. The stresses vary only through the thickness. Local stress intensity factors for the deepest point and surface point are provided. For through-wall cracks, tension and through-thickness and global bending stresses are considered.

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 rate relation [7] is contained in the growth module and is based on the following functional form

$$\dot{a} = C \exp\left[-\frac{Q}{R}T\right]K^{\beta} \qquad (1)$$

where $\dot{a}$ is the crack growth rate, K is the stress intensity factor, T is temperature, Q is the thermal activation energy, and R is the universal gas constant. The constants C and β are estimated from literature data and depend on material, weld or base metal and orientation relative to dendrite direction. Crack coalescence, stability and leak rate for through-wall cracks are checked between each time step.

Inspection: The influence of inspection is treated through the probability of detection (POD) [8]. The POD is a function of crack size that is estimated from test data; therefore, there is uncertainty in the POD. At each inspection, the POD for each of the cracks is recorded, and the influence of inspection on leak probabilities (leaks of various sizes) is evaluated during post-processing. For cracks that grow to leak during the lifetime, the crack contributes (1-POD) leaks, rather than one leak. If more than one inspection takes place then the influence is the product of the (1-POD)s (independent inspections) or the (1-POD) of the last inspection (dependent inspections).

Crack Stability: The stability of part-through cracks is based on net-section collapse for tension and bending loading [9]. The stability of through-wall cracks is based on tearing instability that employs an 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 [10].

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

Crack Opening Displacement: The crack opening displacement for through-wall cracks is estimated for tension and bending loading using literature tabulations [11] that consider elastic-plastic material behavior. Load relaxation due to the presence of the crack is not considered. The crack opening displacement 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 through-wall cracks (complex cracks will be addressed in later versions of the code) are evaluated based on an early version of the SQUIRT software [12], 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 through-wall crack exceeds some specified limit, then this is recorded, but the time loop continues until the pipe ruptures. The effects of leak detection are analyzed during post-processing.

UNCERTAINTY CHARACTERIZATION AND PROPAGATION

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. The purpose of this section is to discuss methods for treating aleatory (irreducible uncertainty) and epistemic (lack of knowledge) uncertainties with a unified approach that allows consistent treatments to be developed regardless of the computer model being used.

The epistemic uncertainty can, in principle, be eliminated with sufficient study and, therefore, expert judgments may be useful in its reduction. From psychology point of view, epistemic (or internal) uncertainty reflects the possibility of errors in our general knowledge. For example, one believes that the population of city A is less than the population of the city B, but one is not sure of that. The treatment of uncertainty can be performed in any order and performing inner and outer loops for simulation methods that result in enormous computational times may not be the most efficient method for performing probabilistic analyses. Much depends upon the question asked and what results are desired.

Rather than enter into a philosophical discussion of these concepts we limit our discussion to simulation methods. Monte Carlo is not discussed in detail assuming the reader understands this concept. Detailed descriptions of two alternative strategies for focusing the calculations are provided. The first method discussed is the Latin Hypercube Sampling (LHS) method. The second is the Discrete Probability Distribution (DPD) method first proposed by Kaplan [13] and modified by Kurth and Cox [14,15] to allow random sampling of the discrete space. This technique allows for the contributions of epistemic and aleatory uncertainty to be defined post processing.

The purpose of LHS is to provide a "dense" sampling of the random inputs or processes to a physical model. The definition of dense is only applicable to the input space and in fact can be quite sparse when viewed from the response space as we shall see in the following discussion.

The LHS is constructed by dividing the input response distribution into N equal probability intervals. This is done for each of the inputs. The first interval for the first variable is then randomly paired with an interval from the second variable, leading to a couplet of (x1, xI) where I is the selected random interval for variable 2. If there is a third interval then this couplet is randomly paired with an interval from the third variable leading to a triplet, (x1, xI, xJ) where J is the random interval selected for the third variable. If there are M random variables then this process is repeated M-1 times leading to an M-tuplet (x1, xI, xJ, …, xK). To obtain the actual value of xL we would generate a random value according to the PDF of the variable selected from interval L. This M-tuplet then is the input that generates a single response. To obtain the next set of inputs the same process is repeated except that if a value has been previously selected it cannot be selected again. Thus a sampling without replacement scheme is used. This implies that there will be exactly N response generated. Therefore for M variables there are NM possible combinations of the inputs. There the LHS design will sample N1-M fraction of the response space.

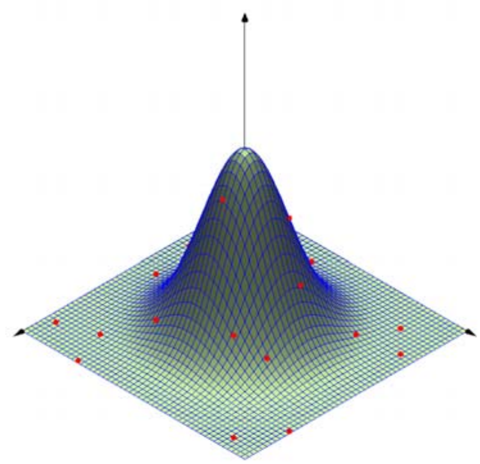

Figure 3 Latin Hypercube Sample for Continuous Response Space

Figure 3 shows a typical LHS of a continuous response function. In the analysis of actual physical models it is rarely the case that the functional form of the response is known since it is exactly what we are interested in determining. In fact experience has shown that there is rarely a standard functional form available to describe the response space. For the analyses being undertaken in xLPR we expect a non-analytic form which we call a fractured response space. This is because the piping system will remain safe from leaks if no cracks form. If more than one crack forms then the response space can be dramatically different than if a single crack forms. If the stress in the pipe remains less than a percentage of the material yield strength then SCC does not initiate. Above yield the damage mechanisms are dramatically different. Many other examples exist. For the purpose of illustration we adopt the "fractured" response surface shown in Figure 4.

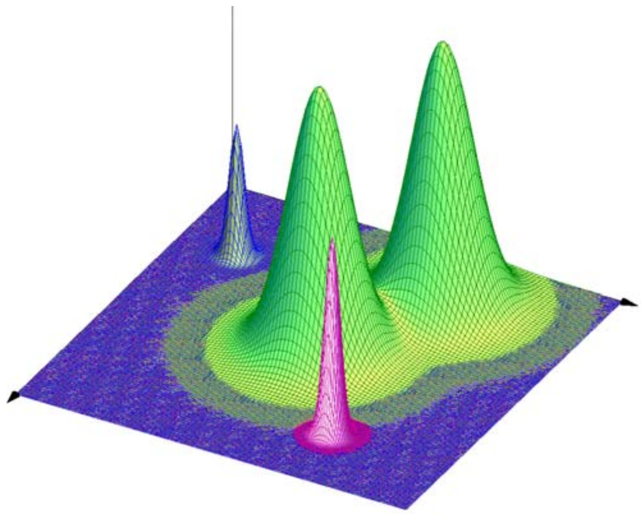

Figure 4 Fractured Response Space for Two Random Variables

The LHS method does not change simply because the response space is different. In fact the purpose of LHS[‡] is to determine the shape of the response space. So we impose the same LHS sample on this surface as was imposed previously. The results are shown in Figure 5.

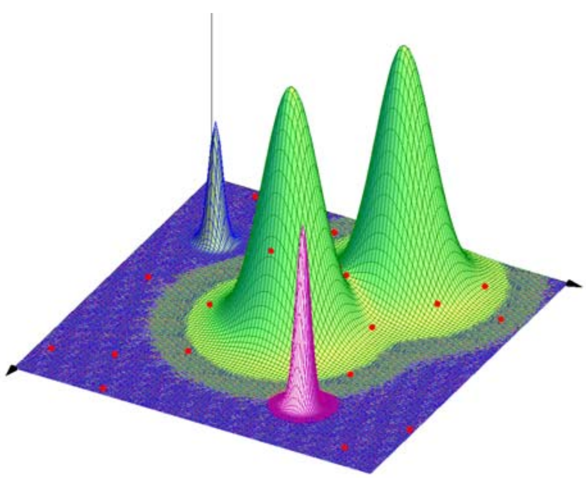

Figure 5 Latin Hypercube Sample for Fractured Response Space

Because of the various peaks and valleys it is difficult to see the LHS with respect to the response surface. Therefore we project it onto the X-Y plane in Figure 6.

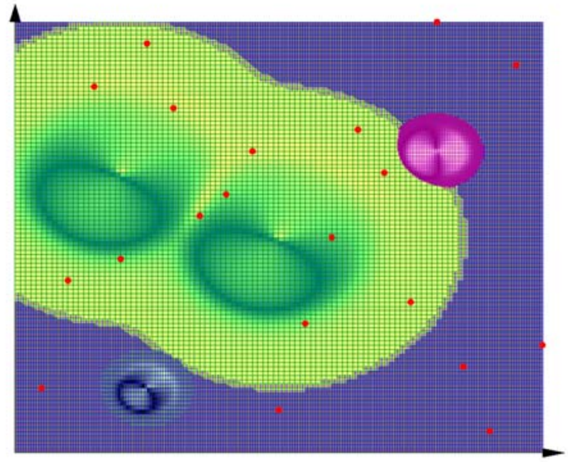

Figure 6 Projection of LHS and Response Surface onto X-Y Plane

What one immediately notices in Figure 6 is that the LHS sample points missed most of the areas of interest. The majority of the selected points are in the low value of the response space and thus without the a priori knowledge of the response surface it is virtually impossible to determine its characteristics.

One method for extracting more information from a LHS design is to run replicate designs. In Figure 7 we show a replicate 5 design. Even in this design we are unlikely to select a combination of points that provides sufficient sampling in the areas as illustrated in Figure 7.

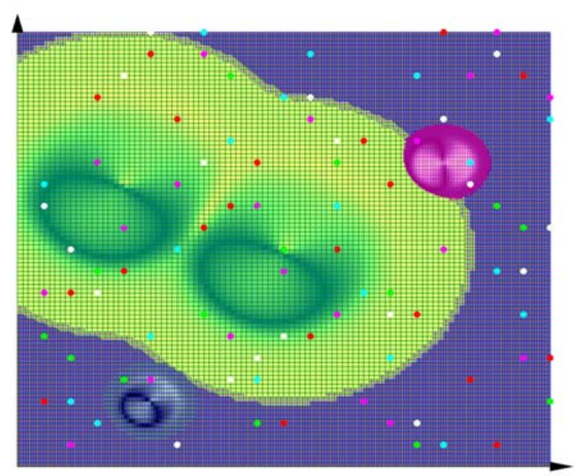

Figure 7 Replicate 5 LHS for Fractured Response Space

‡ Or any other method including Monte Carlo

Discrete Space Sampling

In the LHS sampling when an interval is selected it is sampled within the interval of the sample. A modified version of the code would employ the same strategy as the Discrete Probability Distribution (DPD) method and simply use the conditional mean of the interval. We employ this modified version for LHS comparisons since we do not have to run duplicate designs to compare similar strategies.

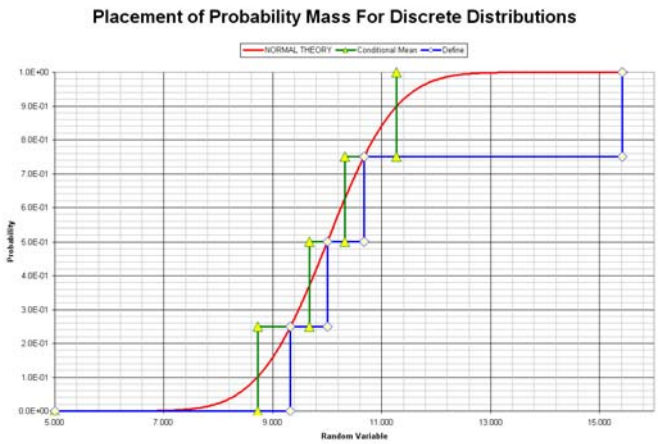

Figure 8 Probability Mass Placement for a 5 Discrete Point Distribution

In Figure 8 we show the conditional mean versus end point placement of a PDF where five points are used. The conditional mean of the interval is defined as:

$$C_M = \int_{x_{i-1}}^{x_i} xPDF(x)dx \tag{2}$$

where C_M is the conditional mean. This is the point at which the interval probability mass is placed giving an unbiased estimate of the CDF in discrete space. There are three methods for defining a DPD. The first is one in which equal probability intervals are used. The second is one in which unequal probability intervals are used. It can be shown that the optimal strategy for choosing these intervals is to use equal space intervals. The final method uses an adaptive strategy in which the DPD is constructed using unequal probability intervals near a peak value of the response. This strategy is illustrated graphically in Figure 9.

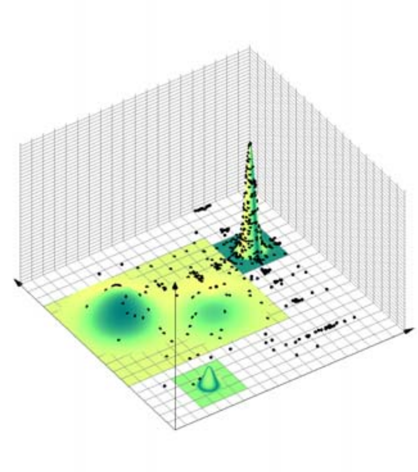

Figure 9 Adaptive Sampling Scheme for DPD

Summary of Uncertainty Analysis Methods for xLPR

There are four methods that will be examined for the propagation of modeling uncertainty in xLPR:

1. Monte Carlo (standard)
2. LHS
3. DPD
4. Adaptive DPD sampling

Monte Carlo is expected to be prohibitive to utilize because of calculation times, LHS will provide the dense sampling of the input space but may not find the regions of extremely low response probabilities, DPD and adaptive DPD can be coupled with LHS to focus the calculations on the response areas of most interest, e.g. crack depths greater than 75% of the wall thickness or leak rates less than 0.5 gpm. One of the major goals of xLPR is the most efficient design of these probabilistic methods. Other methods of propagating uncertainties may be investigated in later versions of the code, i.e., response function generation by high-dimensional model representation

COMMERCIAL FRAMEWORK

The prototype xLPR model and pilot study case was constructed using existing fracture mechanics software coupled to the commercial software framework. This approach leverages the intrinsic features of the commercial software package. A prototype xLPR model was constructed using the GoldSim software application. The GoldSim software is dynamic, probabilistic simulation software developed by GoldSim Technology Group, LLC. This general-purpose simulator is a hybrid of several simulation approaches, combining an extension of system dynamics with some aspects of discrete event simulation, and embedding the dynamic simulation engine within a Monte Carlo simulation framework [16]. The software is ideal for the prototype xLPR model and pilot study since it has off the shelf capabilities including:

- Ability to simulate deterministic or probabilistic model runs using the Monte Carlo method (including Latin Hypercube Sampling);
- 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 that are easy to understand and navigate;
- Dynamically link external programs (e.g., modules) directly to the GoldSim software;
- Directly exchange information between any EXCEL spreadsheet and GoldSim.

The modular-based GoldSim framework model for alpha xLPR model manages input variables (e.g., material properties) and model output (e.g., results) as well as the flow of information that includes the system level model logic. The GoldSim framework for xLPR was constructed with an option to use standard Microsoft Excel spread sheets to define the inputs as well as dynamically pass simulation results to Excel for advanced post-processing. The commercial framework simulation software serves as the integrating shell that links various modules used in the xLPR Model. The GoldSim xLPR model framework controls the order in which the modules are called and the passing of variables into and out of modules. The xLPR approach is to create all of the modules independently, so that the modules can be created by collaborators in any programming language. Both simple and complex calculations are coded as modules and then are directly coupled to the xLPR GoldSim framework using dynamic link libraries (DLLs) by wrapping the original module source code in a simple standard DLL shell [16].

The framework utilizes the GoldSim software libraries of probability distribution functions and the capability to use correlated variables that are used to perform multiple realization stochastic analyses in a Monte–Carlo approach. The framework benefits from the GoldSim software's ability to store simulation data from large numbers of realizations and generate statistics on global probability distributions. GoldSim permits each run to be saved in a single action, including all input data and results from Monte–Carlo analysis. Finally, GoldSim framework has built in GUI functions that allow the developer to quickly assemble specific model runs and to create interactive player files for end-users that allows for viewing, navigating, and even modify input values and model options to run the xLPR model without requiring a software license using the free GoldSim Player software application [16].

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

In addition, the GoldSim framework for xLPR includes the software’s ability to track changes that have been made to a model file. This feature (referred to as versioning) allows the differences between the current version and a previous version of a model file to be quickly determined [16]. The version history is an integral part of the model file, providing an easy to access history of all the changes that have occurred over the life of the model. Providing this configuration management capability is particularly useful for coordinating model changes when multiple people can access and modify the model file and as a Quality Assurance/Quality Control feature allowing for verification and documentation of where and when changes have been made to a model file.

OPEN SOURCE FRAMEWORK

To support the mission of the xLPR project, an Object Oriented Open Source (OOOS) version of the xLPR pilot study code is being developed. In this approach, only open source code and libraries are being used to construct two nested loops as described in Figure 1. With the inner loop and outer loops, the aleatory and epistemic uncertainty are being captured by using a Monte Carlo analysis with a variety of sampling schemes.
The open source xLPR application has been developed by creating Python bindings for the core Fortran models developed by the xLPR Pilot Project Computational Group. These models include the load-history generator, crack placement, and crack initiation procedures that are executed, as described in Figure 1, to construct the input data required for the Monte Carlo realizations. The realization records are stored in a database where they can be retrieved during a separate execution to carry out the time-loop deterministic kernel shown in Figure 2. The deterministic analysis includes the crack growth, crack stability, leak rate, inspection, and mitigation modules, all programmed in Fortran and linked through Python bindings.

The xLPR open source application will be integrated as a part of the SIAM-PFM framework developed by Oak Ridge National Laboratory (ORNL). The **S**tructural **I**ntegrity **A**ssessments **M**odular-**P**robabilistic **F**racture **M**echanics (SIAM-PFM) is a modular probabilistic fracture mechanics computer code. The objective of SIAM-PFM is to develop, validate, and maintain a configuration-controlled, modularly-designed, open-source computer code that can be used to assess the structural integrity of any passive pressure-bearing component in a nuclear power plant. SIAM-PFM is intended as a framework in which a wide range of problem classes in the area of nuclear power plant safety and reliability can be addressed in a systematic and consistent way using modern principles of probabilistic risk assessment. This problem solving environment is intended to be readily extensible to different problem classes with the level and methods of user interaction to be determined by discussions with the NRC and potential stake holders. A common feature of the different applications that come within the purview of SIAM-PFM is that they are all the subjects of probabilistic risk assessment and will, therefore, represent “risk-informed” analyses. Originally, in the demonstration phase of the SIAM-PFM project, the reactor pressure vessel and primary-water piping systems were chosen as initial test cases to show how two such disparate applications could be addressed within a common probabilistic framework. The SIAM-PFM framework is greatly complemented by the addition of the xLPR application into its target problem class.

CONFIGUATION MANAGEMENT PLAN

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

Each CM item (e.g., module, framework model, input set, etc.) is developed and controlled using a systematic process and includes documentation that CM item meets the design requirements and can be verified independently (e.g., without consultation with the originator). The CM process used for the xLPR program is based upon the concept of agile software development. Agile software development refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams[17]. The CM process incorporates the necessity for thorough documentation and issue tracking through the development process; this includes 'snap-shots' of the iterative model development. For example the xLPR pilot study includes an alpha model and a beta model. The alpha model will be used as the basis for the beta model as well as to help define its requirements. The beta model will be used to run the pilot test case and the results will be employed to define the longer term xLPR project requirements. The CM process for xLPR is a flexible paradigm that is adaptable to multiple configuration management software systems; a necessity when collaborating with teams distributed geographically and utilizing different platforms. The xLPR model framework and its associated modules, including source code and documentation, and inputs are controlled by storing them in a set of access controlled subdirectories on the xLPR file server. The electronic file server for controlled storage of xLPR model files uses the web accessible Microsoft SharePoint process and document management software. Modifications to the CM items (e.g., module source code and xLPR model inputs) are tracked and documented on the SharePoint server. Controlled versions are then downloaded when the xLPR model is run. This central repository enables the development of the modules and framework model independently, across organizational and geographic boundaries. The developer checks out a CM item from the SharePoint server (e.g., module source code) and makes the modifications and uploads the file version to be independently checked and verified. The documentation is also checked out, modified and checked back in to be independently verified. The central CM repository concept even flanges well with a standard SCM system software as needed to meet the software and QA requirements for the open-source development of the SIAM framework. The CM items are posted to the SharePoint repository and are updated at each iteration or control point defined in the xLPR program. Specifically, the development history, documentation, and issue tracking for the SIAM framework and modules are contained with a separate SCM system, but the controlled CM items, for the alpha and beta model versions, are posted to the SharePoint repository.

PLANS FOR xLPR BEYOND PILOT STUDY

After the pilot study, the plan for the xLPR Project is to develop a general purpose probabilistic fracture mechanics (PFM) computer program and programming system. The purpose of this program would be to compute the probability of failure of entire plant systems that comprise the primary pressure boundary for the coolant in both pressurized water reactor (PWR) plants and boiling water reactor (BWR) plants. The program should be able to evaluate the structural reliability of all passive components in each plant system, such as piping, their integral attachments for piping supports, tanks and vessels, heat exchanger tubing and pump and valve bodies. This plan does not require the capability to calculate the failure probabilities of each system component so they can be added to estimate the total system failure probability. Rather the plan is to be able to assess in detail the most limiting system components that control the overall system reliability.

For the detailed assessment using the planned PFM program, there are several areas of interest. The first area is the range of failure modes. This would potentially range from a limiting flaw size for approved repair procedures to a full pipe break or rupture. In between these extremes would be limiting leak rates that would either disable or significantly restrict the safety function of the plant system or would initiate events that would require plant systems to respond to safely shutdown the plant. Typical initiating events of concern are the small, medium and large break loss of coolant accidents (SBLOCA, MBLOCA and LBLOCA, respectively) that are evaluated in the probabilistic risk assessment (PRA) for the plant.

The second area of interest is the extent of degradation mechanisms and the effects, both positive and negative, of the associated mitigation actions. While the pilot plant study is limited to the degradation mechanism of primary water stress corrosion cracking (PWSCC) in the pressurizer surge line piping welds in a PWR plant and limited mitigation, such as replacement and structural weld overlay, the general PFM program must be able to evaluate all potential time-related damage mechanisms and all postulated mitigation actions. This would also include any synergistic effects between multiple mechanisms, such as combining fabrication flaws that were missed during pre-service inspections with flaws initiated later in life and combining the flaw growth due to stress corrosion cracking (SCC) with that due to fatigue from normal operating events like periodic system testing or plant heat up and cool down. Plants now have mitigation programs for all known active degradation mechanisms, such as stress corrosion cracking or flow assisted corrosion. However, the planned general PFM computer program would be especially useful in addressing the susceptibility to any emergent degradation mechanisms and assessing their associated mitigation actions in a realistic manner that includes the effects of the uncertainties inherent with limited amounts of information. Furthermore, it would be very beneficial if the affected plant utilities and the regulators at NRC were doing their evaluations using the same accepted PFM computer program.

The plans for the xLPR general PFM computer program do not include just one computer probabilistic program with all the capabilities described previously. What is more reasonable and feasible is the development of a general probabilistic programming system that would accept problem specific subroutines with all the capabilities needed for that specific problem, such as an emergent degradation mechanism, as discussed previously. The general programming system would contain standard modules for generating random numbers and different types of statistical distributions, such as postulated normal log-normal distributions, uniform and log-uniform distributions for expert input and a general purpose Weibull distribution for fitting any available data. A standard framework would also be provided to do the random trials and time looping with calls to the problem specific subroutines and the standardized input and output subroutines to start and end each computer run. Some sort of importance sampling would also be provided for quick calculation of low values of failure probabilities to assist in making sensitivity studies. Since all uncertainties (distributions) would be specified in the input, the sensitivity studies would also help to identify which input parameters and uncertainties had the greatest effect on the calculated probabilities. The standardized output format would allow for easy plotting of failure probability with operating time and graphical comparisons for each sensitivity study with a specified base case.

As shown in Figure 10, it is envisioned that the standard framework for the xLPR general PFM computer programming system would involve just five key problem specific subroutines. These include one for setting initial conditions, such as probability of having a fabrication flaw and its size, time to flaw initiation or both. Another subroutine would be for steady-state changes, such as high temperature creep crack growth, SCC, wall thinning or embrittlement of cast stainless steel. The third problem specific subroutine would be for transient changes, such as high cycle fatigue due to mechanical or flow-induced vibration or low cycle fatigue crack growth due to operating transients or design basis seismic events (operating basis earthquake). Another subroutine would evaluate if any of the failure modes of concern, as discussed previously would have occurred during each time step. The last subroutine would be for effects of an in-service inspection and could calculate the probabilities of detection and accurate sizing based upon the flaw size at that time. These five key subroutines could call any other supporting subroutines that would be needed for a specific problem, such as ones for calculating stress intensity factors, coalescence of adjacent flaws, crack opening displacement and leak rate.

To satisfy software quality assurance requirements the overall programming system libraries would only need to be verified once since it would not change from problem to problem. For each problem specific xLPR computer program, only the five key subroutines and any needed supporting subroutines would have to be verified.

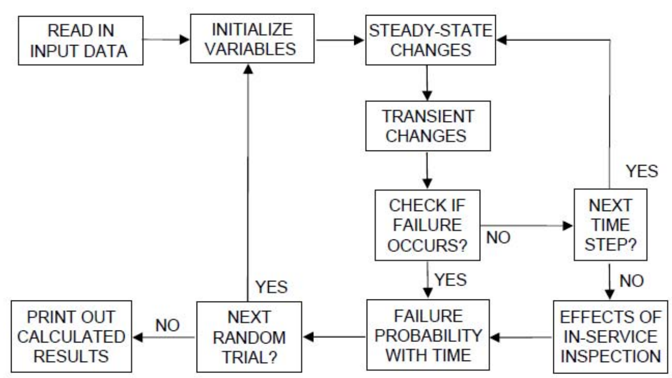

Figure 10 Flow Chart for an xLPR General PFM Computer Programming System

SUMMARY

In this paper, the initial computational framework development efforts for the xLPR program were discussed. The xLPR software code is being developed cooperatively between the US NRC staff and the nuclear industry through a memorandum of understanding to aid both the regulators and the industry in confirming that plants are in compliance with the current regulations.

Due to the complexity of such a project, a pilot study is first being conducted to determine the feasibility of both the computational approach and the management structure in developing such a code. This pilot study is focusing only on the problem of PWSCC in a pressuruzer surge nozzle. 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.

Current efforts by the computational group have focused on the alpha version of the xLPR code. In this version, existing code and methodologies have been used to develop computational frameworks that utilize both commercial and open source software in order to determine which is reasonable for further code development. In addition, a configuration management program has been developed to assure that a process is in place for establishing and maintaining consistency of the xLPR model and its functional attributes with its requirements, design, and operational information throughout its life.

Within the pilot study, models appropriate for solving the surge nozzle problem are still under development and stochastic processes that will be used to determine the extremely low probability of rupture are currently under investigation. In the long term, the plan for the xLPR Project is to develop a general purpose PFM computer program and programming system. The purpose of this program would be to compute the probability of failure of entire plant systems that comprise the primary pressure boundary for the coolant in both pressurized water reactor (PWR) plants and boiling water reactor (BWR) plants.

ACKNOWLEDGEMENTS

The authors would like to thank all of the members of the computational group team in the xLPR effort. These members include:
Cedric Sallaberry – Sandia National Laboratory
John Helton – Sandia National Laboratory
Don Kalnich – Sandia National Laboratory
Paul Williams – Oak Ridge National Laboratory
Dilip Dedhia – Structural Integrity Associates
Anita Gubbi - Structural Integrity Associates
Nathan Palm – Westinghouse

The authors would also like to thanks Craig Harrington from EPRI and Aladar Csontos from the NRC for their continued support of this program.

REFERENCES

1 Harrington, C., Csontos, A., “Development Of A Probabilistic Pipe Rupture Assessment Code,” PVP2010-26158, Proceedings of ASME-PVP 2010: 2010 ASME Pressure Vessels and Piping Division Conference, July 18-22, 2010, Bellevue, Washington, USA

2 EricksonKirk, M., Rathbun, H., Models for Use in Probabilistic Pipe Rupture Assessment Code Development,” PVP2010-25681, Proceedings of ASME-PVP 2010: 2010 ASME Pressure Vessels and Piping Division Conference, July 18-22, 2010, Bellevue, Washington, USA

3 Focht, E., and DeBoo, G., “Development of Initial Input Parameters for a Probabilistic Pipe Rupture Assessment Code for Nuclear Reactor Coolant Pressure Boundary Leak-Before-Break Analyses,” PVP2010-26021, Proceedings of ASME-PVP 2010: 2010 ASME Pressure Vessels and Piping Division Conference, July 18-22, 2010, Bellevue, Washington, USA

4 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.

5 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.

6 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.

7 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)

8 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.

9 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.

10 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.

11 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.

12 "User's Guide for SQUIRT (Windows Version 1.1)," Battelle Memorial Institute, March 2003.

13 Kaplan, S. On the Method of Discrete Probability Distribution in Risk and Reliability Calculations – Applications to Seismic Risk Assessment, Risk Anal., 2, 3 (1981)

14 Kurth, R.E, and Cox, D.C., An Investigation of Discrete Probability Distributions for Probabilistic Mechanics Analysis, Risk Anal.: Int. J., 5, 3 (1985)

15 Kurth, R.E. and Cox, D.C., A Random Sampling Discrete Probability Algorithm with Condensation for Probabilistic Analysis, Nuclear Technology, 92, 2 (1990)

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

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