xLPR PIB Meeting
Computational Framework and Alpha Program Plan

U.S.NRC
UNITED STATES NUCLEAR REGULATORY COMMISSION
Protecting People and the Environment

David L. Rudland, Ph.D
Senior Materials Engineer
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555

David.Rudland@nrc.gov

October 27, 2009

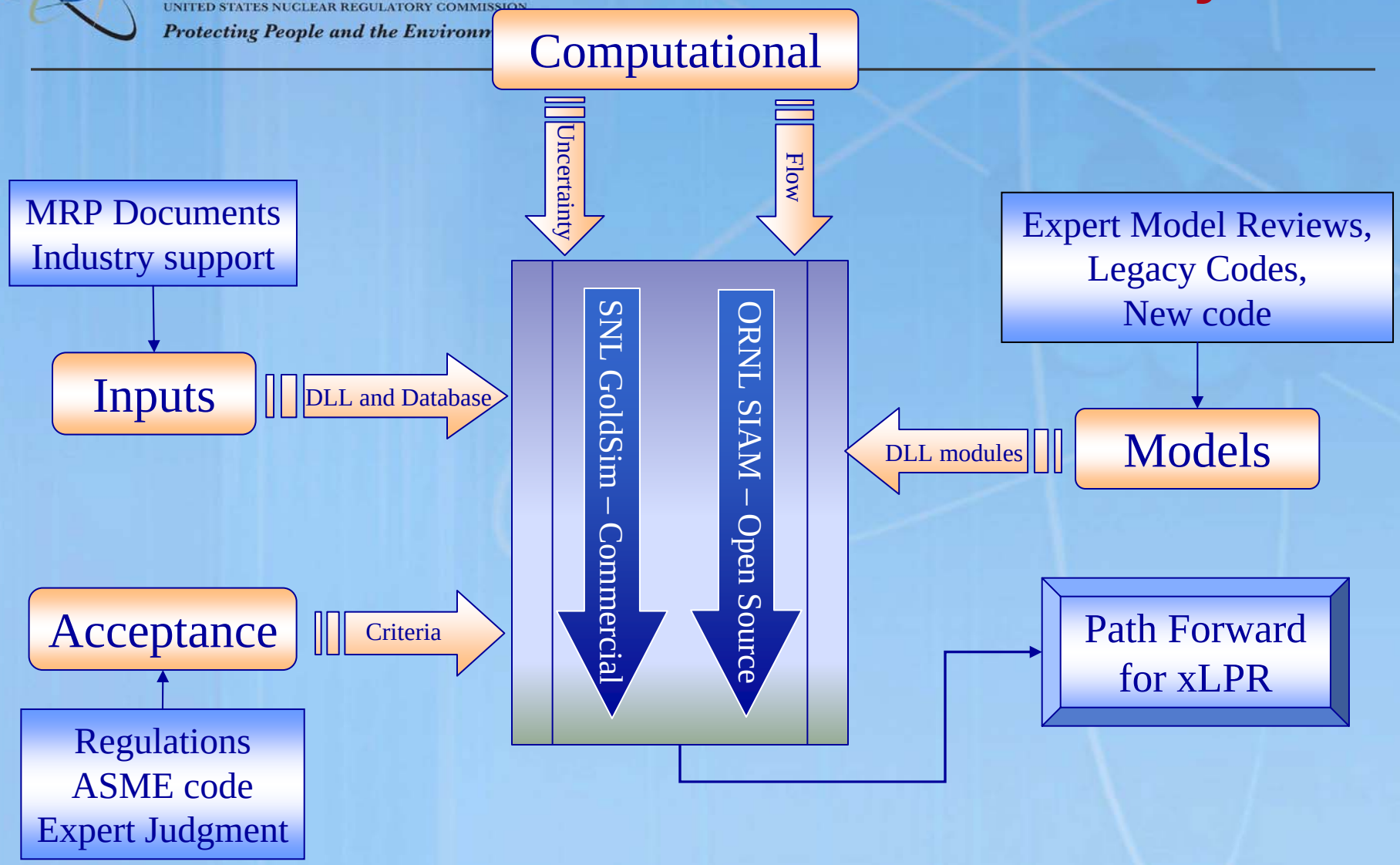
Computational Architecture

- We need to determine the required xLPR code architecture within the program timeline and the following constraints:
  - Comprehensive with respect to known challenges,
  - Vetted with respect to scientific adequacy of models & inputs,
  - Flexible to permit analysis of a variety of in-service situations
  - Adaptable to accommodate evolving and improving knowledge

Computational Architecture

- To meet this goal, we have decided to proceed down parallel paths to test two architectures
  - Commercial Software
  - Open Source software
- Each code will use the same “modules”, inputs, etc
- Each will be assessed based on the criteria in last slide
  - Probably need assessment document – PIB developed??
- It is envisioned both NRC contractor and Industry contractor will perform evaluation
  - Need to discuss

Pilot Study



Whole process guided by Project Integration Board

GoldSim

- GoldSim is a general-purpose simulation software program
- Some of GoldSim's key features include:
  - Quantitatively address the inherent variability and uncertainty
  - Superimpose the occurrence and consequences of discrete events
  - Build top-down models
  - Dynamically link external programs into GoldSim
  - Directly exchange data with databases
  - Insert graphics, notes and hyperlinks
  - Create custom designed graphical interfaces
  - Carry out sensitivity analyses
  - Save your models as player files using GoldSim Player (a free download)
  - Utilize powerful extension modules

GoldSim for xLPR

- GoldSim is a simulation software for modeling complex dynamic systems and supports decision and risk analysis by simulating future performance while quantitatively representing the uncertainty and risks inherent in all complex systems

- GoldSim software is an Object-oriented modeling environment with specialized functions, native Monte-Carlo capabilities and run controller features (manages all linked runtime software via DLL interface)

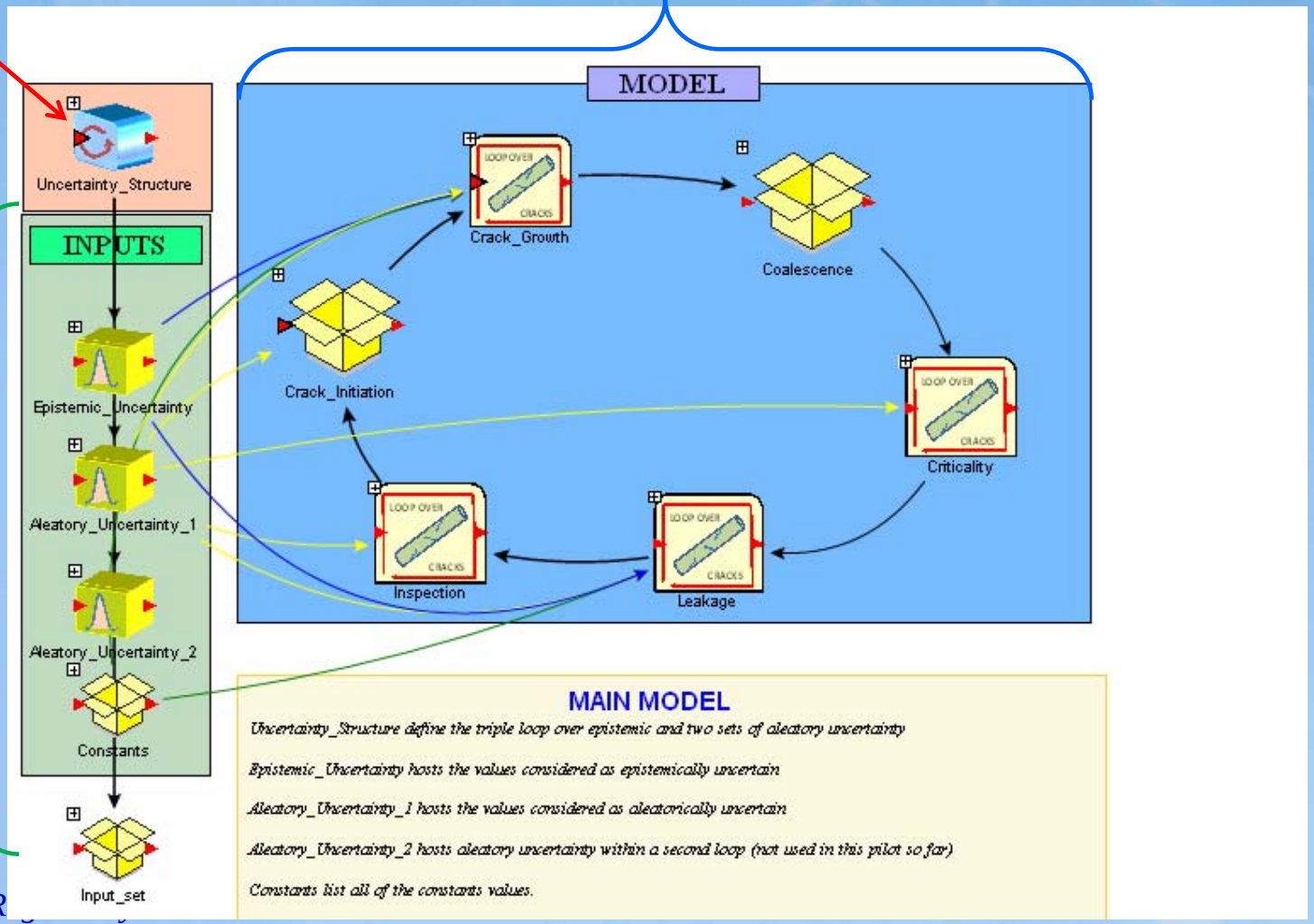
- Use GoldSim for Alpha xLPR Model and Pilot Study to
  - Develop a probabilistic framework
  - Sample uncertain parameters
  - Pass uncertain and constant parameters to modules (DLLs)
  - Receive and process output from module, e.g., link modules together
  - Compile, display and save model results

xLPR pilot - GoldSim

Uncertainty Structure

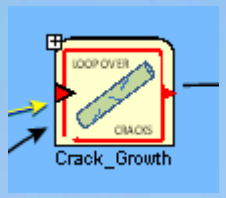
Model

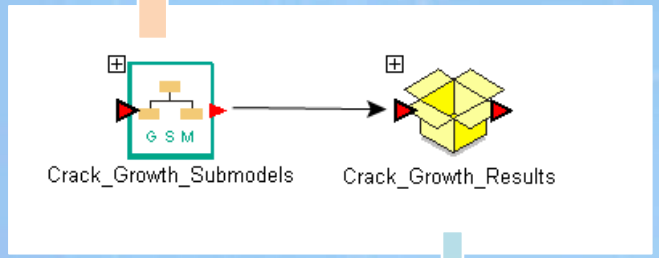
Inputs

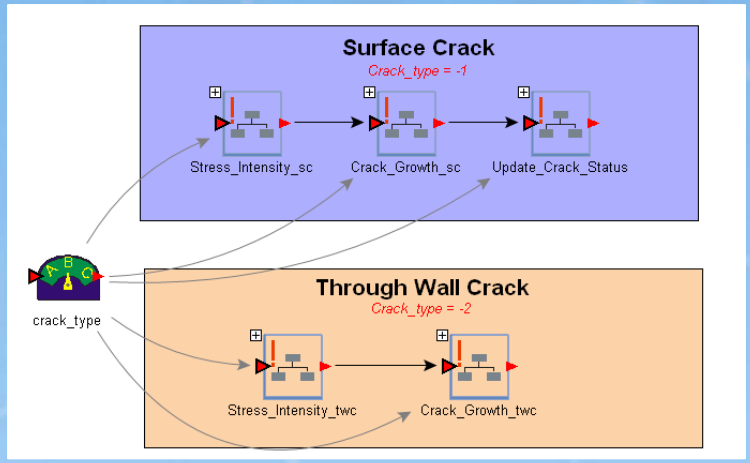


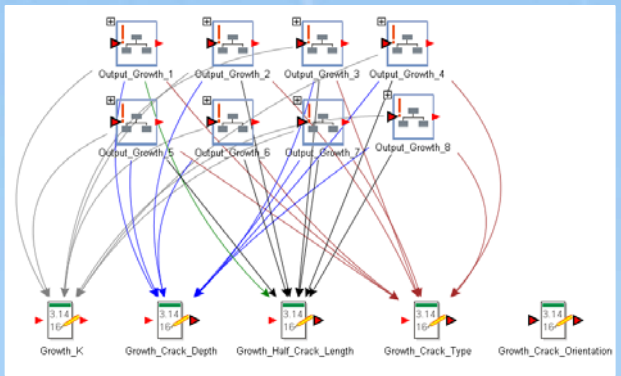
Crack Growth Modules

- Conditional Container – called for each crack
- Cracks treated differently according to their type (Surface Crack or Through wall crack)
- Results of properties saved for each crack









GoldSim Status

- Skeleton alpha version of xLPR complete and running
- Still incorporating temporary modules
  - COD
  - Inspection
  - Critical flaw size
- Alpha version complete by year end

Background on SIAM-PFM

Structural Integrity Assessments Modular - Probabilistic Fracture Mechanics (SIAM-PFM)

- **Develop a *general computational platform* for NPP primary circuit components**
- **Provide a *systematic basis* for risk-informed assessments**
- **Integrate *advanced fracture mechanics* techniques applicable to pressurized structures**
- **Enable contributions by multiple organizations within a *modular software architecture***

SIAM-PFM – Status

- The SIAM-PFM Development Team for the xLPR Pilot Project consists of:
  - Ms. Hilda Klasky – software engineer researcher
  - Dr. Kalyan Perumalla – computer science researcher
  - Dr. Paul Williams – computational fracture mechanics

ORNL will be adding a new member to the team in November:
Srikanth Yoginath – Computer Science Modeling and Simulation Group Computational Sciences and Eng. Div.

SIAM-PFM Status - Summary

- Front-end demonstration platform has been developed – available for review and comment.
- Established international interest in collaboration.
- Two new staff members have joined the SIAM-PFM team.
- Software QA tools have been put into place: Mercurial and Trac.
- Layered software architecture (MVC) has been implemented.
- QA-based coding practices for traceable documentation, code review, prototyping, and incremental unit testing have been established.
- The object-oriented design and analysis phase has begun.
  - Algorithmic decomposition for the xLPR pilot project is proceeding.
  - Model layer classes for the xLPR pilot project are now being coded.

SIAM-PFM Status - Schedule

- We expect to have an initial demonstrator by the end of January 2010.
- We expect to have an xLPR Pilot Project prototype ready for demonstration by the first of May 2010.
- The prototype will include some level of capability to also perform RPV beltline analysis.

Alpha Version of Program Plan

- Purpose and Scope
  - Written by computational group to aid in development of framework and architecture
  - Explanation of models/modules selected by computational group and used for framework development
    - These modules used as placeholders until models group finalizes modules
  - The document is “living” and will be modified as we go from alpha to beta to version 1.0
  - This document will also evolve into the technical basis for xLPR

Alpha Version Module Development

- Modules were either extracted from legacy code, or written for this purpose
- Computational group took the liberty of selecting alpha version modules
- Framework being written so alpha modules can be easily replaced with beta modules

Outline

- Section 1. Overall Flow
- Section 2. Time History Development
- Section 3. Geometry and Material Properties
- Section 4. Loads
- Section 5. Time Loop
- Section 6. Crack Initiation and Placement
- Section 7. Crack Growth
- Section 8. Crack Coalescence
- Section 9. Crack Stability
- Section 10. Leak Rate/Detection
- Section 11. Inspection
- Section 12. Mitigation and Remediation
- Section 13. Outputs
- Section 14. Sample Problem
- Section 15. References

May cause biggest challenges going from alpha to beta

Initiation and Crack Placement

- **Crack Initiation (Section 6.2)**
  - **Pre-existing defects not active in the alpha model**
  - **PWSCC initiation includes for uncertainty by using Poisson's arrival rate model**
  - **Other initiation mechanisms will not be included in alpha model**
    - **e.g., Fatigue, IGSCC, flow enhanced corrosion**

- **Crack Placement (Section 6.3)**
  - **Calculations done using a Crack Placement Module**
    - **Stress based criterion for crack placement (Amzallag, et al, 2001)**
    - **Includes uncertainty in crack placement model**

PWSCC Initiation

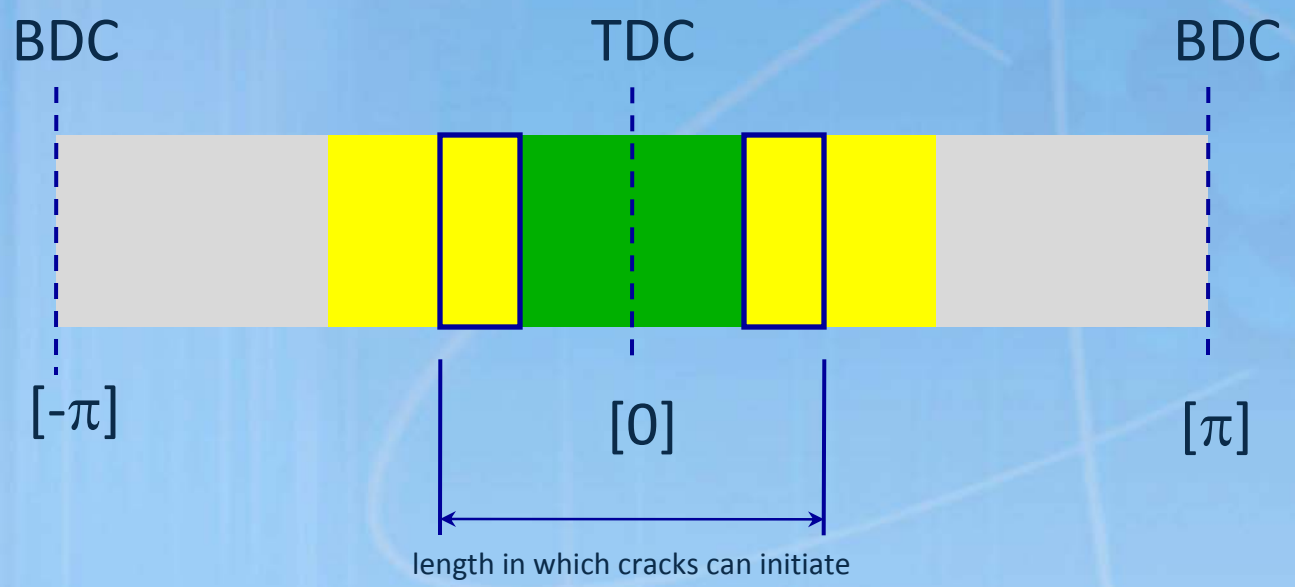
- **Crack Initiation will be determined at time=0**
  - **Inputs**
    - **Arrival Rate, λ**
    - **The arrival rate, λ, will be uncertain**
    - **Outputs - Time to initiation**

$$P\left(N(t+\tau)-N(t)=k\right)=\frac{e^{-\lambda\tau}\left(\lambda\tau\right)^{k}}{k!}$$

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

Crack Placement



Region 1: local ID stress > yield strength

Region 2: 0 <= local ID stress <= yield strength

Region 3: local ID stress < 0

fraction of region 2 in which cracks can initiate

Leak Rate and Detection

- In order to calculate leak rate the following information is needed
  - Crack size (length)
  - Crack opening displacement (at loads) and shape
  - Fluid conditions
  - Crack morphology parameters
  - A leak rate model

Leak Rate Model

- SQUiRT - Seepage Quantification of Upsets In Reactor Tubes has been developed to incorporate the Henry-Fauske two phase flow model and the single phase models.

- Most recent versions of SQUiRT also include COD-crack morphology models

- For the alpha version of xLPR, an older version of SQUiRT will be used that does not have single phase flow or COD-crack morphology models

Leak Detection and Mitigation

- For the alpha version of xLPR, the leak detection limit will be an input (sampled).
- The calculated leak rate will be compared against sampled leak detection limit.
- If the leak rate exceeds the limit, the data is recorded, but the time loop is not exited.
- The effect of leak detection will be calculated after the simulation is complete
- It is assumed that a leaking flaw will be completely removed from the analyses

Inspection

- Credit for in-service inspection will be included in alpha version of xLPR

- Probability of detection and inspection schedule will be input

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

- Probability of repair assumed equal to 1. Flaw sizing uncertainty is ignored

Inspection

- For the alpha version of the code, a simplified inspection scheme will be implemented
  - The probability of non-detection (as sampled from the POD) will be assigned to each flaw present in the analyses at the inspection times
  - The PND as a function of time is then used to modify the failure probabilities to credit for inspection
  - This process assumes that all flaws found by inspection are fully mitigated and removed from the analyses
  - Also, if one flaw is found, all flaws in the simulation are assumed repaired

Pre-emptive Mitigation

- Two options for pre-emptive mitigation – Stress based mitigation and full PWSCC mitigation
- Mitigation time will be a user input – deterministic
- A probability of mitigation effectiveness may be added, which will evoke a random number to determine if the mitigation is effective

Stress Mitigation

- Weld residual stress distribution for mitigated state would be input by user
- Using the “new” ID stress, the crack initiation arrival rate beyond the mitigation time would be

$$\lambda_{mitigated} = \left(\frac{\sigma_i}{\sigma_u} \right)^{-4} \lambda$$

- The crack growth after mitigation would be driven by the operating loads and the sampled mitigated weld residual stress

Full PWSCC Mitigation

- For this option, a mitigation time is input, and both crack initiation and growth will not occur in the future
- Probability of mitigation effectiveness will still apply

Summary

- Alpha version of code underway
  - Modules chosen by computational group, and used as placeholders until Models group modules complete
  - What are the topics which may cause pain in moving from Alpha to Beta?
- Program plan written to describe alpha version and guide computational group in developing framework and architecture
  - Detailed discussions tomorrow