FAVPRO

Fracture Analysis of Vessels - Probabilistic

NRC's New Probabilistic RPV Integrity Assessment Code

-

FAVPRO v1.0 Release - Public Meeting

U.S.NRC
United States Nuclear Regulatory Commission
Protecting People and the Environment

Meeting Agenda

Time	Topic	Presenters/Participants
1:00pm-1:15pm	Safety Brief and Introductory Remarks	NRC
1:15pm-2:00pm	FAVPRO v1.0 Overview	NRC
2:00pm-2:30pm	FAVPRO Demonstration	NRC
2:30pm-3:00pm	Open Q&A and Discussion	All

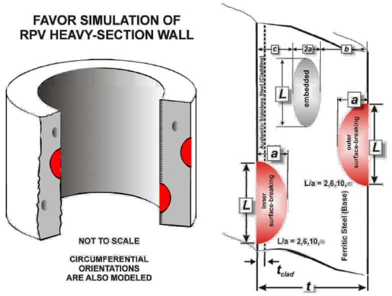
FAVPRO v1.0 Overview

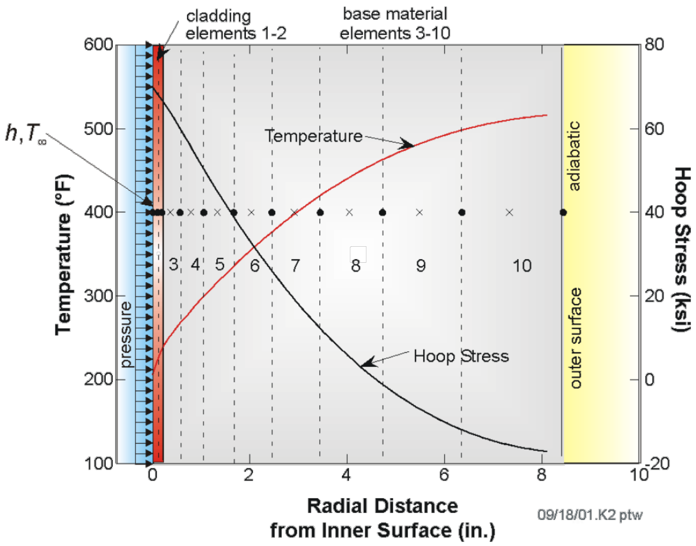
- Presentation Outline
  - Overview and Capabilities
  - A little bit of history: the FAVPRO journey
    - Modernization goals
    - Status update on FAVPRO
  - New Features
    - Embrittlement Trend Curves
    - Fracture Mechanics and Flaw Population Models
    - Parallelism and Performance
  - Software Quality Assurance (SQA) and Verification & Validation (V&V)
  - FAVPRO User Interface (UI) and JSON
  - Obtaining FAVPRO and FAVPRO User Group

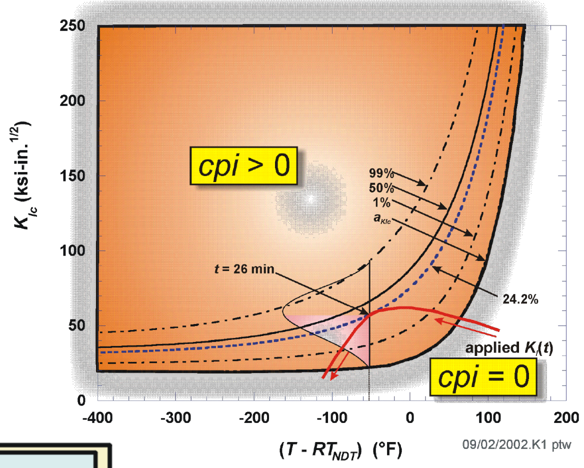
NRC wishes to have a more collaborative relationship with FAVPRO users, and have more regular information exchanges with the FAVPRO community

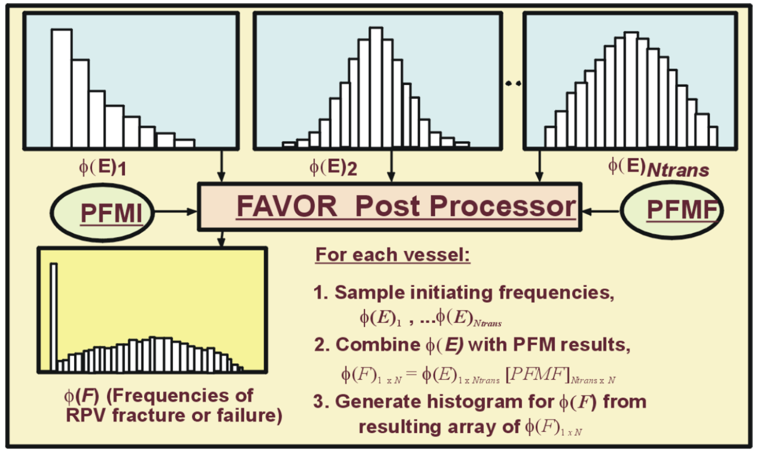
What is FAVPRO?

- Probabilistic Fracture Mechanics tool for RPV integrity assessment
- Focus on cylindrical beltline
- 1D finite element axisymmetric solver
  - Stresses and temperatures from any TH transient
  - Stress intensity factors (ID, OD, embedded flaws)
- Deterministic run modes
  - Through-wall profiles (T, σ, SIFs…)
  - Time histories
  - Critical RT_{NDT} (embrittlement) for crack growth
- Probabilistic run mode
  - Conditional probabilities of crack growth initiation (CPI) and vessel fracture (CPF)
- Combination of conditional probabilities and transient frequencies to generate frequencies of crack growth initiation (FCI) and through-wall crack failure (TWCF)





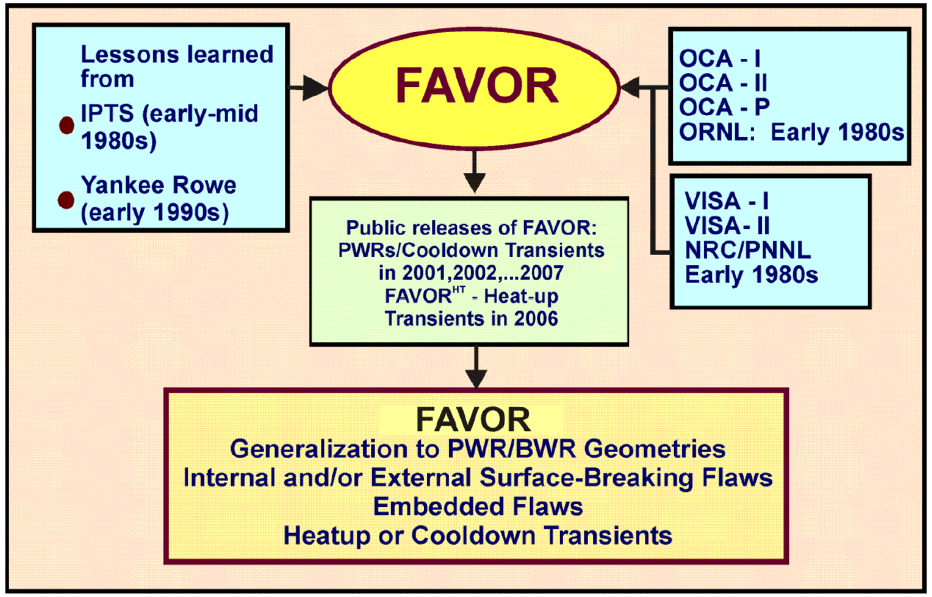




FAVPRO Validated Capabilities

- Heatup and cooldown transients
  - 1D finite element solution for temperatures and stresses
  - User specified material properties
  - Weld residual stress option
  - Crack-face pressure option
  - Stress-free temperature model for cladding residual stress
- Flaw populations
  - Semi-elliptical internal or external surface flaws
  - Elliptical embedded flaws within base metal
    - Cannot model semi-elliptical sub-cladding flaws
  - As-found flaw population or sampled population from specified distributions
- Stress intensity factor influence coefficients approach for K calculations
  - ASME solutions for base metal
  - Custom solutions for cladding (ID surface flaws)
- Warm prestress options
- Several embrittlement trend curves
- Ductile tearing and crack arrest options
- Vessel chemistry and fluence sampling
- Resampling option for crack growth

FAVPRO’s Ancestor: FAVOR

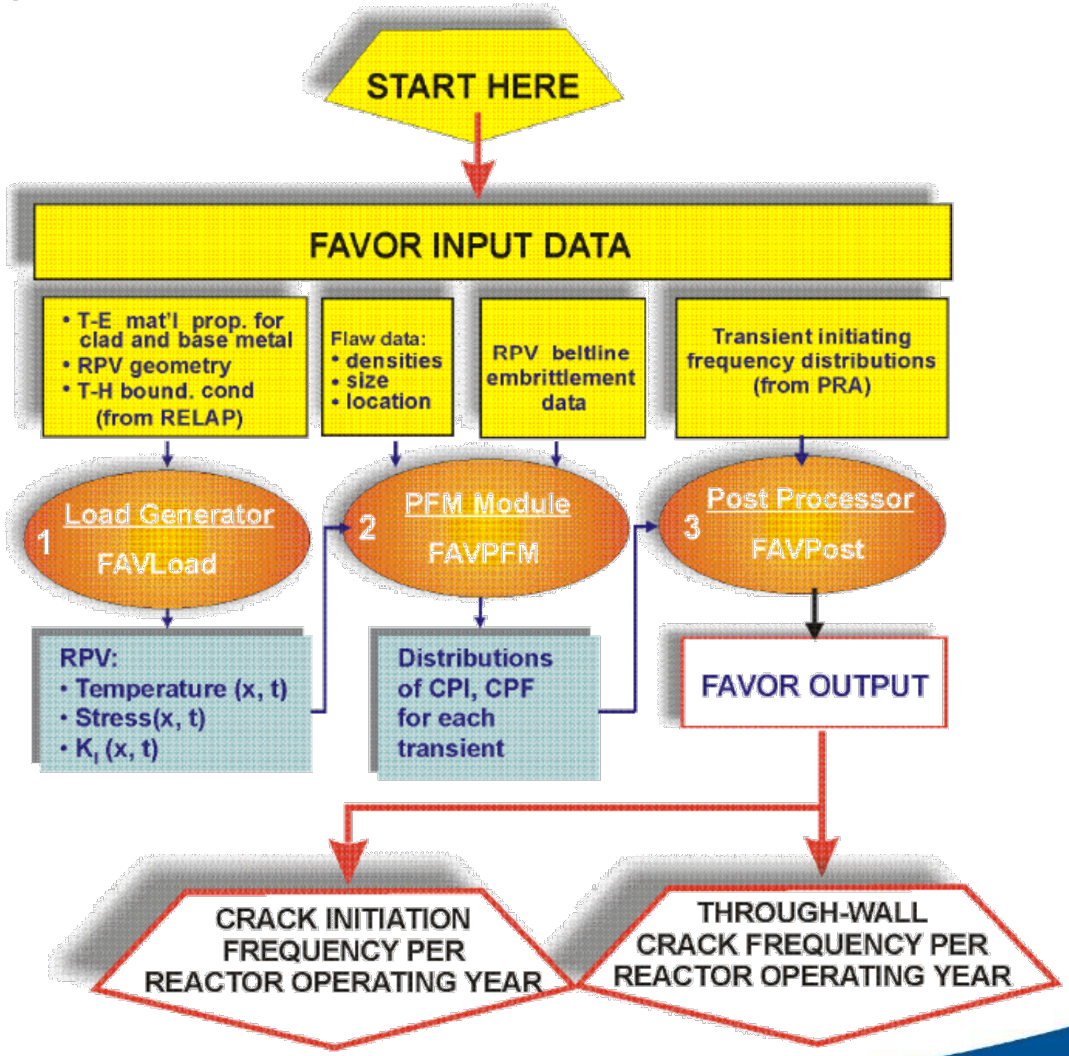


- Created in the 90s under the Heavy Section Steel Technology (HSST) program
  - Combined attributes of OCA-P (ORNL) and VISA-II (PNNL) codes
- Initially targeting Pressurized Thermal Shock (PTS) transients for PWRs
- Later expanded to all heat-up and cooldown transients, for both PWR and BWR
- Used for the PTS re-evaluation project which resulted in updating 10 CFR 50.61
- Used to develop the basis for alternate PTS rule 10 CFR 50.61a

- Recent uses include:
  - Shallow flaw issue disposition
  - Doel and Tihange laminar flaw evaluations
  - RG 1.99 Rev 2 re-evaluation
  - NuScale confirmatory calculations (FAVPRO)

FAVPRO's Beginning: FAVOR-v16.1

- Developed and issued by ORNL late 2016
- Was the final version of FAVOR issued by the Heavy Section Steel Technology (HSST) Program at Oak Ridge National Laboratory (ORNL)
- FAVOR = FAVLOAD + FAVPFM + FAVPOST
  - 3 sequential executable programs
  - Information passed via formatted text files
- Serial and sequential code
- Software Quality Assurance (SQA) gaps
  - [ML20017A171](ML20017A171)
  - [ML20017A170](ML20017A170)

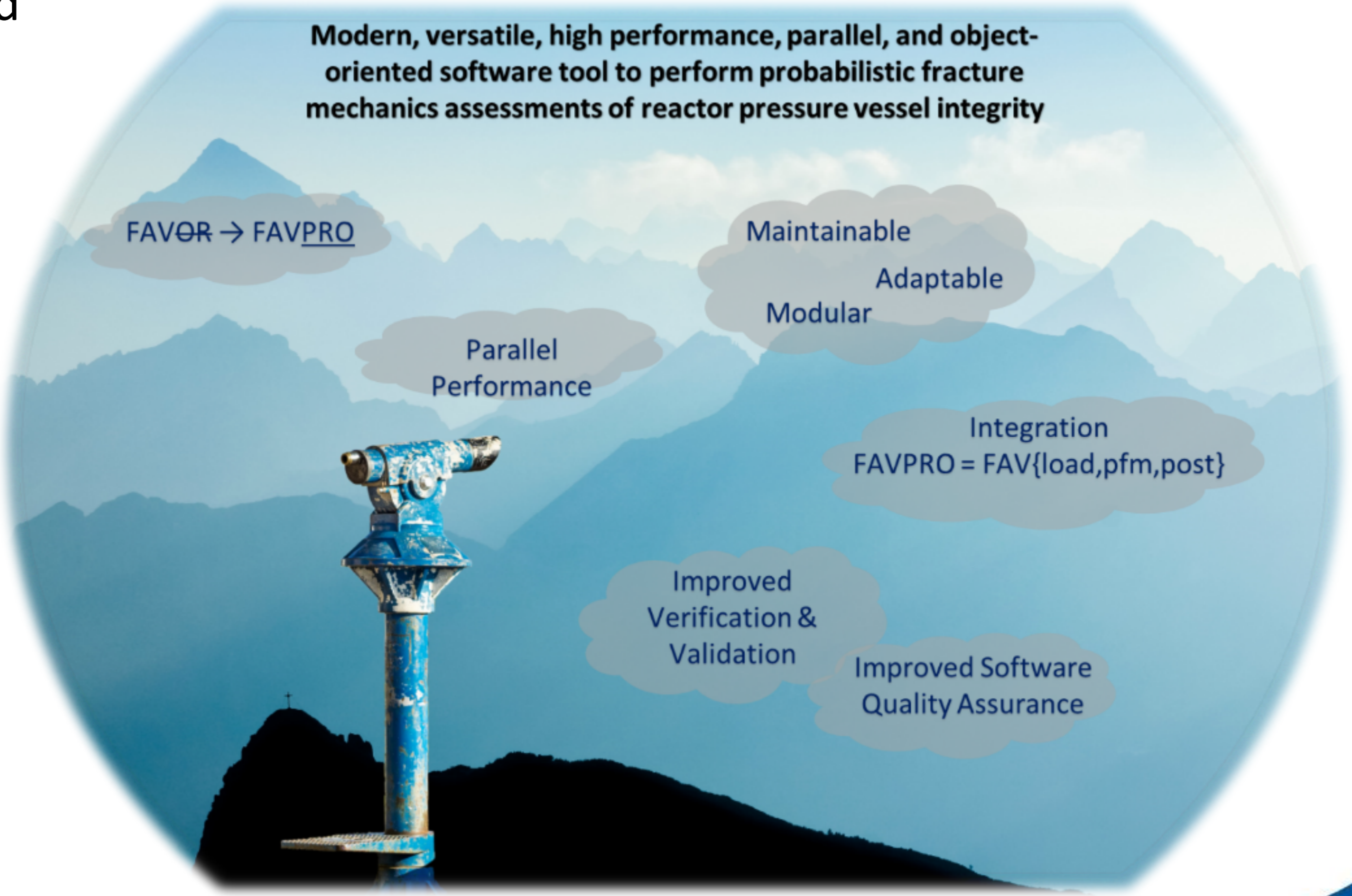


Final FAVOR Version: FAVOR-v20.1.12 Released on June 4, 2021

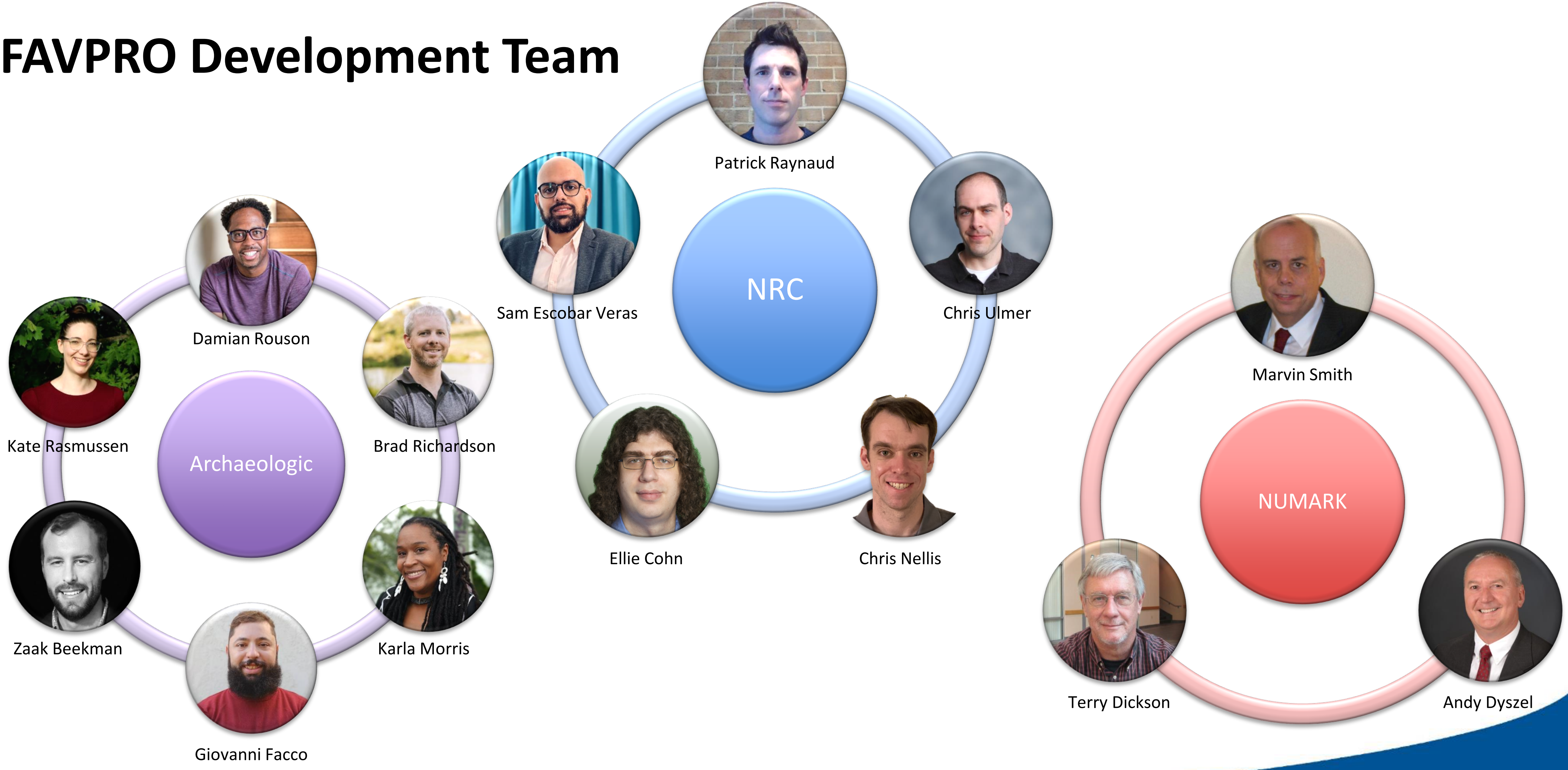
- CMake build system
- Source code improvements:
  - Convert to free form '.f90' files
  - Modularization
  - Begin removal of obsolete Fortran
- Testing improvements:
  - New integration tests
  - A few unit tests
  - Automatically run testing on GitHub for all code changes
- Documentation
  - Automatically generated developer documentation
    - FORD: Fortran Documenter
    - Detailed code descriptions from source parsing
  - Created new SQA documents:
    - SQAP: ML21180A161
    - CMMP: ML21180A167
    - SRD: ML21246A230
    - SDD: ML22132A068
  - Created new Manuals:
    - User Manual: ML21175A301
    - Theory Manual: ML21175A300

Vision and Goals for FAVPRO

- Completely refactor FAVOR to create an improved tool with equivalent capabilities, written in modern Fortran
- GOALS
  - Maintainability
  - SQA and V&V improvements
    - Testing
    - Documentation
  - Modularity, adaptability, easier feature development
  - Modern programming
    - Object-oriented code
    - Parallel code
  - Maximize automation for testing and documentation
  - Program integration: 3 FAVOR into 1 FAVPRO
  - Use State-of-Practice tools and libraries
    - GitHub: source control
    - State-of-practice build system
    - State-of-practice unit testing framework
    - Standardized I/O via Java Script Object Notation (JSON)



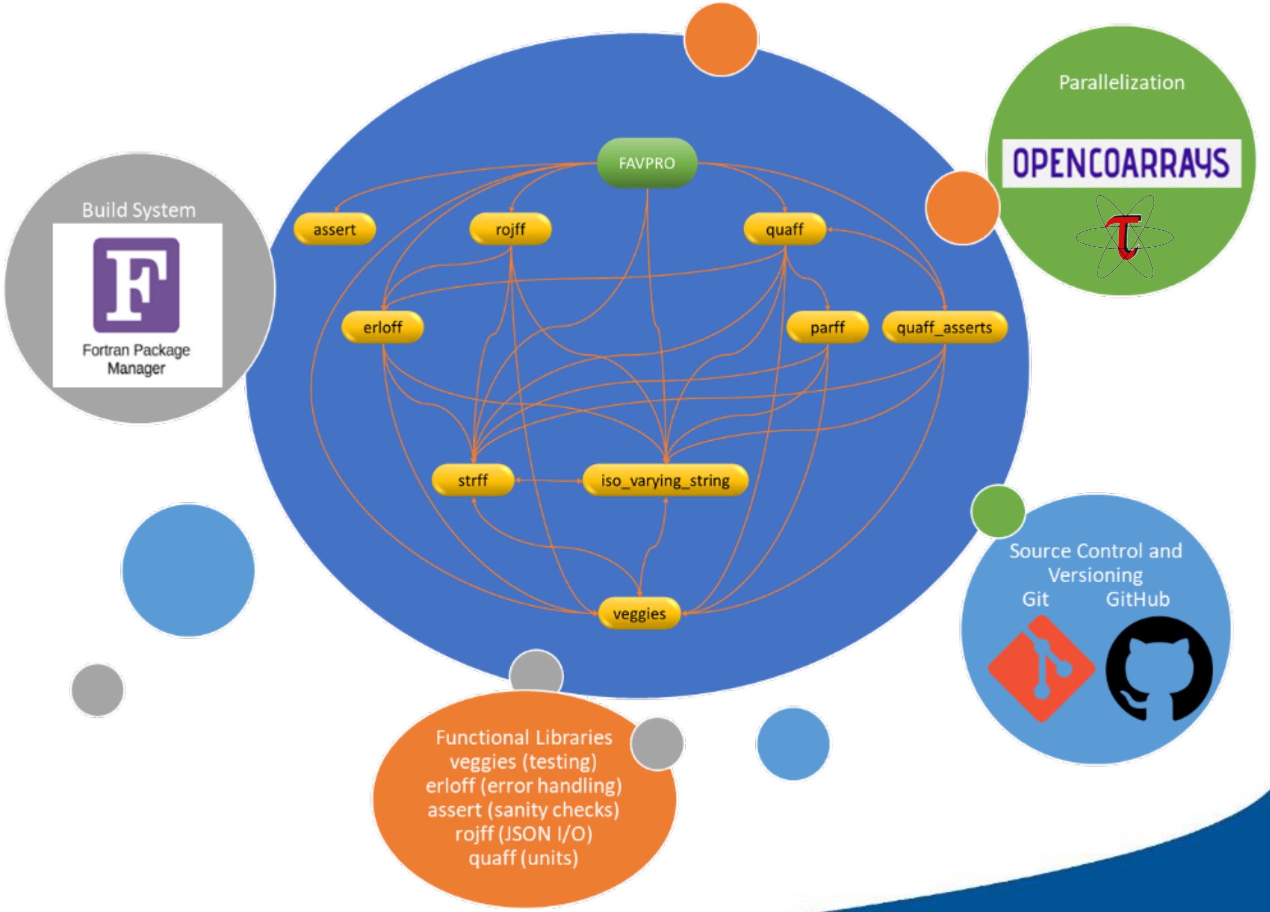
FAVPRO Development Team



U.S.NRC
United States Nuclear Regulatory Commission
Protecting People and the Environment

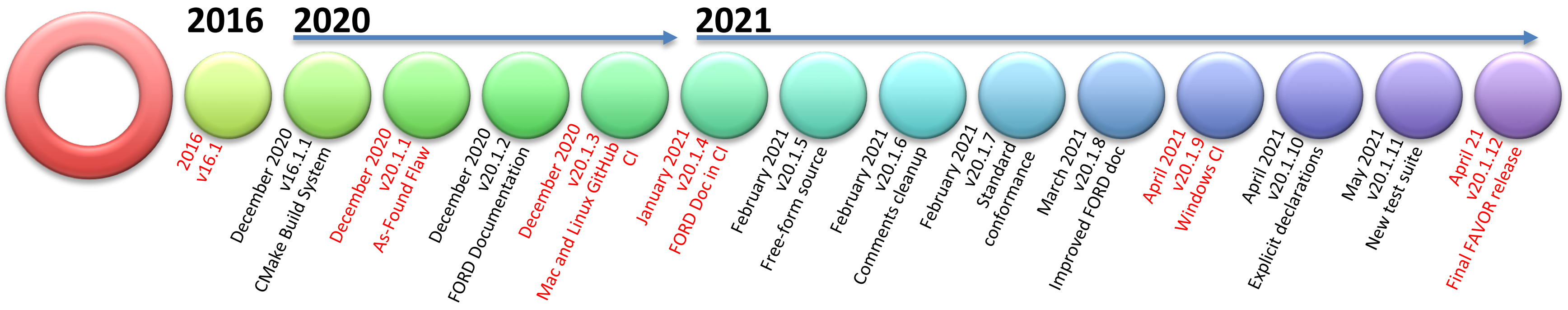
FAVPRO: Leveraging the Modern Fortran Ecosystem

- Fortran has a growing user community and there is renewed interest in the language
- Fortran 2018 is a modern object-oriented parallel programming language
  - High performance for scientific and engineering computations (one of the fastest)
- We use many open-source libraries
  - fpm: for building and testing
  - garden for unit test development
  - prune for object comparisons
  - assert: assertion utility -> prerequisite checks
  - quaff: quantities for Fortran -> unit tracking
  - rojff: return of JSON for Fortran -> JSON interface
  - rngff: random splitable number generator
  - erloff: errors and logging for Fortran
  - OpenCoarrays for parallelization



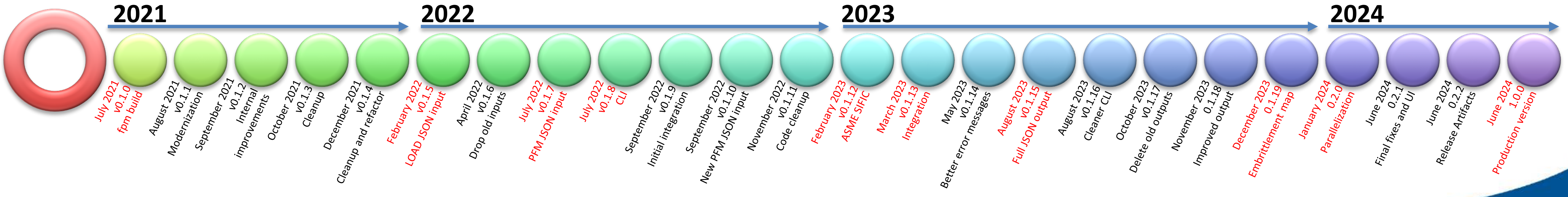
FAVPRO Timeline
Agile Development Pays Off!

FAVOR



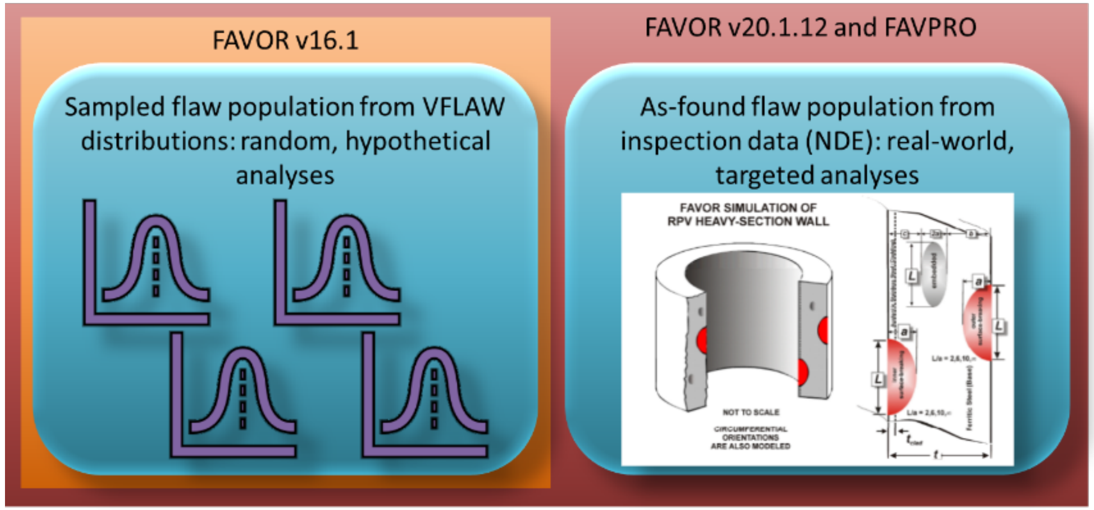
- 3-year development
- Intense collaboration
- Rapid release cycle
  - 90 days average
- Traceability

FAVPRO



FAVPRO Features: Flaw Treatment and Fracture Mechanics Update

- Flaw modeling options
  - VFLAW sampled flaw distributions
    - Relies on output from legacy VFLAW code
    - Can be produced ‘manually’ to force some flaw distributions
    - Cannot be used to specify actual flaws
  - As-found flaw specification
    - JSON input file format
    - Allows specification and placement of actual flaws
- Stress intensity factor (SIF) calculations
  - Use of ASME solutions wherever possible
  - Still use Abaqus-generated custom solutions for cladding contribution to SIF



Flaw type	FAVOR	FAVPRO
Inner Surface Flaw (cladding contribution)	ABAQUS	ABAQUS
Inner Surface Flaw (base metal contribution)	ABAQUS	ASME
Outer Surface Flaw	ABAQUS	ASME
Embedded Flaw	ASME	ASME

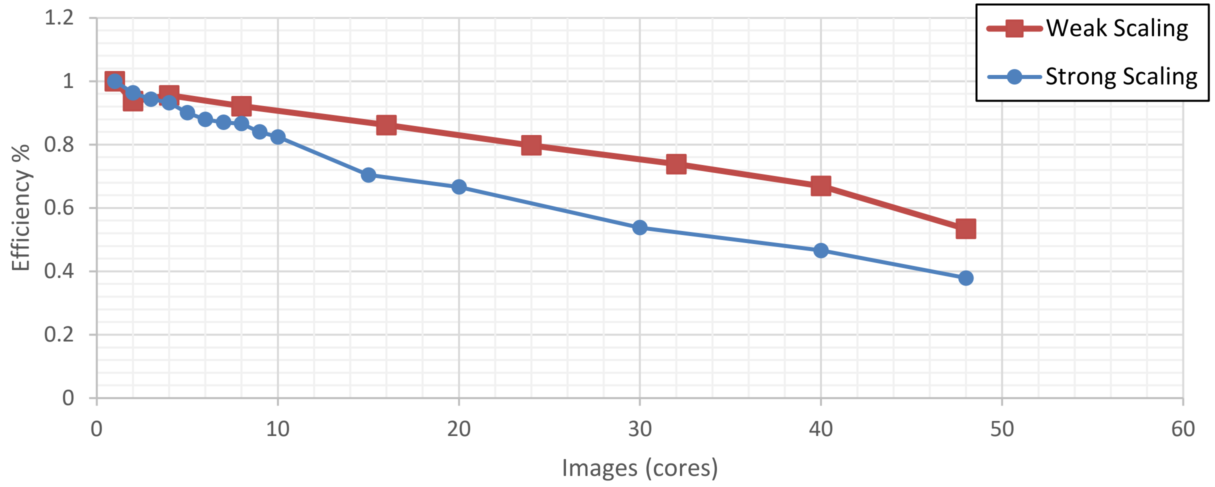
FAVPRO Features: New Embrittlement Trend Curves (ETC)

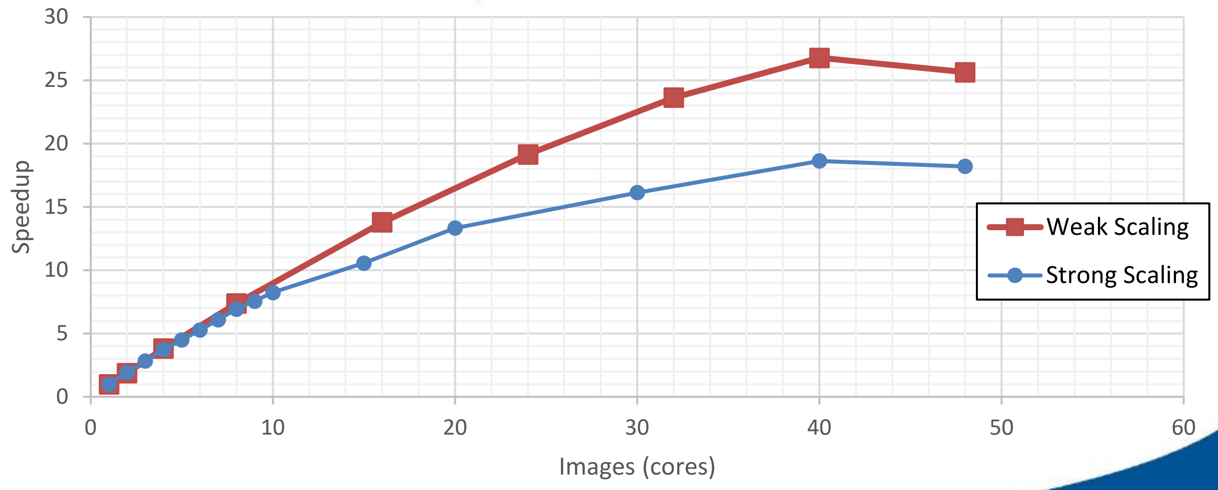
- Currently available ETC
  - RG-1.99 Rev. 2
  - EONY 2000 and 2006
    - Added newer EONY 2013 model
  - Kirk 2007, Radamo 2007, and Kirk+Radamo 2007
    - Early versions of ASTM model
    - Replaced by ASTM E-900
- Future: add non-US ‘mainstream’ embrittlement trend curves to the FAVPRO options?
  - Japanese model (update to JEAC4201, recently presented at FONTEVRAUD-10)
  - French model (2011: FONTEVRAUD-7, or more recent if available)

Embrittlement Trend Curves	
FAVOR	**FAVPRO**
RG-1.99 Rev. 2	RG-1.99 Rev. 2
EONY 2000	EONY 2000
EONY 2006	EONY 2006
	EONY 2013
Kirk 2007	ASTM E900
Radamo 2007	
Kirk + Radamo 2007	

FAVPRO Feature: Parallel Execution of Probabilistic Calculations

- Two FAVPRO executables are distributed
  – Serial executable: faster for deterministic calculations and small probabilistic problems
  – Parallel executable: faster for large probabilistic calculations using ‘mpiexec’
- Example for 48 core server
  – 100k simulations (strong scaling)
  – 12.5k simulations per processor (weak scaling)
  – Good scalability
  – Efficiency diminishes as machine resources are used up (as expected)





FAVPRO SQA and V&V

- Git/GitHub version control and independent tracking of changes
  - Transparency, critical reviews, automated unit and integrated testing
  - Code merging requirements must be met
- Updated SQA documentation (next slides)
- Increased V&V testing
  - Over 300 unit-tests
  - 78 integration-tests
  - Run automatically at every code change on Windows, Linux, and MacOS
  - Run for serial and parallel run modes

Summary of FAVPRO SQA Configuration Documents

SQA Document	Status
Software Quality Assurance Plan (SQAP)	Published ML24095A318
Configuration Management and Maintenance Plan (CMMP)	Published ML24095A319
Software Requirements Document (SRD)	Under development
Software Verification & Validation Plan and Results Report (SVVPR)	Published for FAVPRO v0.1.15 ML24102A185
Software Design Document (SDD)	Theory Manual and FORD Documentation
Software Test Plan(s) (STPs)	GitHub README file
Software Test Results Report(s) (STRRs)	GitHub Actions Log

Code Distribution Item	Status
Implementation Documentation	Ongoing
1. FAVPRO executables	Frequent internal releases on GitHub (current 1.0.0)
2. User's Manual	Published ML24113A237
3. FAVOR Theory Manual	Under final reviews
4. Acceptance Test Problems	GitHub CI

FAVPRO User Interface (UI) and JSON: What is JSON?

- JSON (JavaScript Object Notation) is a lightweight data-interchange format
  - Easy for humans to read and write
  - Easy for machines to parse and generate
- ➢ JSON is an ideal data-interchange language

```
{
  "vessel geometry" : {
    "internal radius" : "86.0 in",
    "wall thickness" : "8.75 in",
    "clad thickness" : "0.25 in"
  }
}
```

- JSON is built on two structures:
  1. A collection of name/value pairs
     - In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array
  2. An ordered list of values
     - In most languages, this is realized as an array, vector, list, or sequence
- These are universal data structures
  - Virtually all modern programming languages support them in one form or another
- It makes sense that a data format that is interchangeable with programming languages also be based on these structures

FAVPRO JSON Inputs and Outputs


```json
{
  "vessel geometry" : {
    "internal radius" : "86.0 in",
    "wall thickness" : "8.5 in",
    "clad thickness" :"0.156 in"
  },
  "material properties" : {
    "base" : {
      "temperature dependent" : true,
      "thermal expansion reference temperature" : "70.0 F",
      "density" : "489.0 lbm/ft^3",
      "thermal conductivity table" : {
        "T" : [
          70.0, 100.0, 150.0, 200.0, 250.0,
          300.0, 350.0, 400.0, 450.0, 500.0,
          550.0, 600.0, 650.0, 700.0, 750.0,
          800.0
        ],
        "k(T)" : [
          24.8, 25.0, 25.1, 25.2, 25.2,
          25.1, 25.0, 25.1, 24.6, 24.3,
          24.0, 23.7, 23.4, 23.0, 22.6,
          22.2
        ],
        "T units" : "F",
        "k(T) units" : "BTU/(h ft F)"
      },
```

```json
{
  "convergence table increment" : 10,
  "frequency PDFs" : [
    {
      "transient number" : 1,
      "thermal-hydraulic sequence ID" : 100,
      "PDF" : {
        "f" : [
          0.00000573, 0.00000738, 0.00000876, 0.0000101, 0.0000123,
          0.0000161, 0.0000177, 0.0000194, 0.0000227, 0.0000261,
          0.00003, 0.0000351, 0.0000381, 0.0000408, 0.0000543,
          0.0000687, 0.0000853, 0.000112, 0.000124
        ],
        "d(f)" : [
          0.5, 0.5, 1.5, 2.5, 5.0,
          10.0, 5.0, 5.0, 10.0, 10.0,
          10.0, 10.0, 5.0, 5.0, 10.0,
          5.0, 2.5, 1.5, 1.0
        ],
        "f units" : "1/yr",
        "d(f) units" : "%"
      }
    },
```

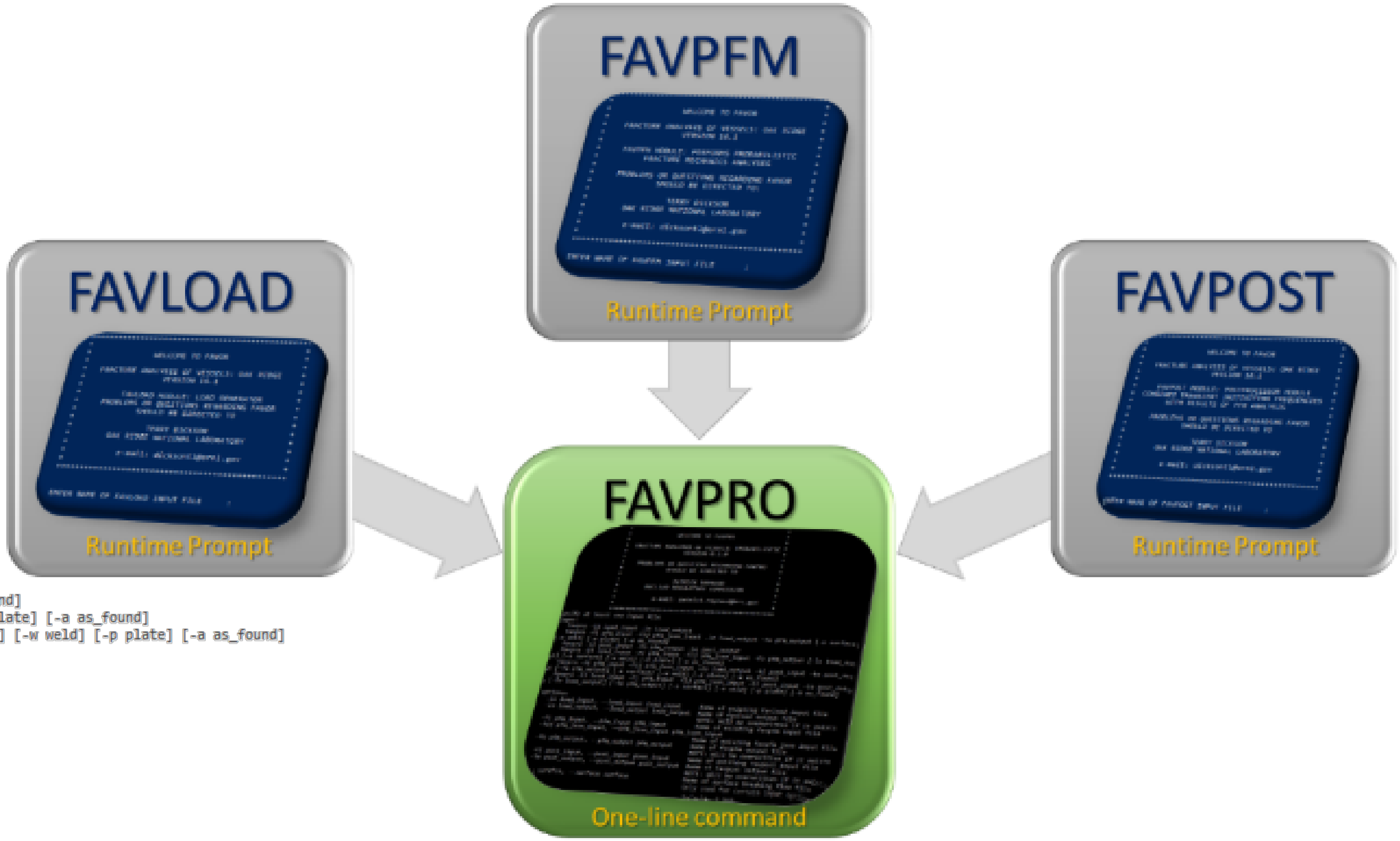
```json
{
  "probabilistic simulation" : {
    "probabilistic simulation properties" : {
      "number of RPV simulations" : 100,
      "flaw population model" : 2,
      "initiation growth arrest trials" : 100,
      "warm prestress model" : 1,
      "category-3 flaw analysis" : false,
      "child subregion reports" : false,
      "embrittlement model" : 2006,
      "normal coolant temperature" : "556.0 F",
      "plant operating time" : "60.0 yr",
      "ductile tearing" : true,
      "create ductile tearing reports" : false,
      "create detailed output report" : false,
      "flow stress" : "80.0 ksi",
      "max K" : "200.0 ksi in^0.5",
      "arrest model" : 2,
      "weld layer resampling" : false,
      "vessel failure normalized crack depth" : 0.9,
      "mean local fluence multiplier" : 0.118,
      "local fluence multiplier" : 0.056,
      "mean Cu multiplier" : 0.167,
      "stdev Ni weld" : 0.162,
      "stdev P weld" : 0.0013,
      "stdev Cu plate" : 0.0073,
      "stdev Ni plate" : 0.0244,
      "stdev P plate" : 0.0013
    },
    "embrittlement map" : {
      "weld major regions" : [
        {
          "major region ID" : "01",
          "major properties" : {
            "Cu" : 0.213,
            "Ni" : 1.01,
            "P" : 0.019,
            "Mn" : 1.32,
            "DT30 shift flag" : 1,
            "Cu saturation flag" : 3,
            "rtndt0" : "6.0 F",
```

```json
{
  "date" : "03-Jun-2024",
  "time" : "14:05:02",
  "version" : "0.2.1",
  "command line" : "build/bin/favpro -lo tests/integration/inputs/load/ver_1_ref.
  "random seed" : 1234567890,
  "load input" : {…
  },
  "pfm input" : {…
  },
  "probabilistic output" : {
    "CPI output" : […
    ],
    "CPF output" : […
    ],
    "analysis output" : {
      "number of transients" : 1,
      "number of RPV simulations" : 100,
      "thermal-hydraulic sequence numbers" : [
        1
      ],
      "number of subregions" : 13,
      "normalized maximum weld flaw depth (%)" : 15,
      "normalized maximum plate flaw depth (%)" : 15,
      "output by transient" : [
        {
          "transient number" : 1,
          "thermal-hydraulic sequence ID" : 1,
          "completed trials" : 100,
          "CPI" : 1.4141817e-3,
          "CPF" : 1.4141817e-3,
          "CPI and CPF fractions by major region" : {
            "RTndt mean value at crack tip" : "134.05346 F",
            "by parent major region" : [
              {
                "major region" : "01",
                "RTmax" : "277.64564 F",
                "number of simulated flaws" : 34,
                "percent of total flaws" : 1.6267599,
                "initiation" : {
                  "number of flaws" : 4,
                  "percent of total CPI" : 97.68034
                },
                "cleavage failure" : {
                  "number of flaws" : 4,
                  "percent of total CPF" : 97.68034
                },
                "ductile failure" : {
                  "number of flaws" : 0,
                  "percent of total CPF" : 0.0
                }
              },
```

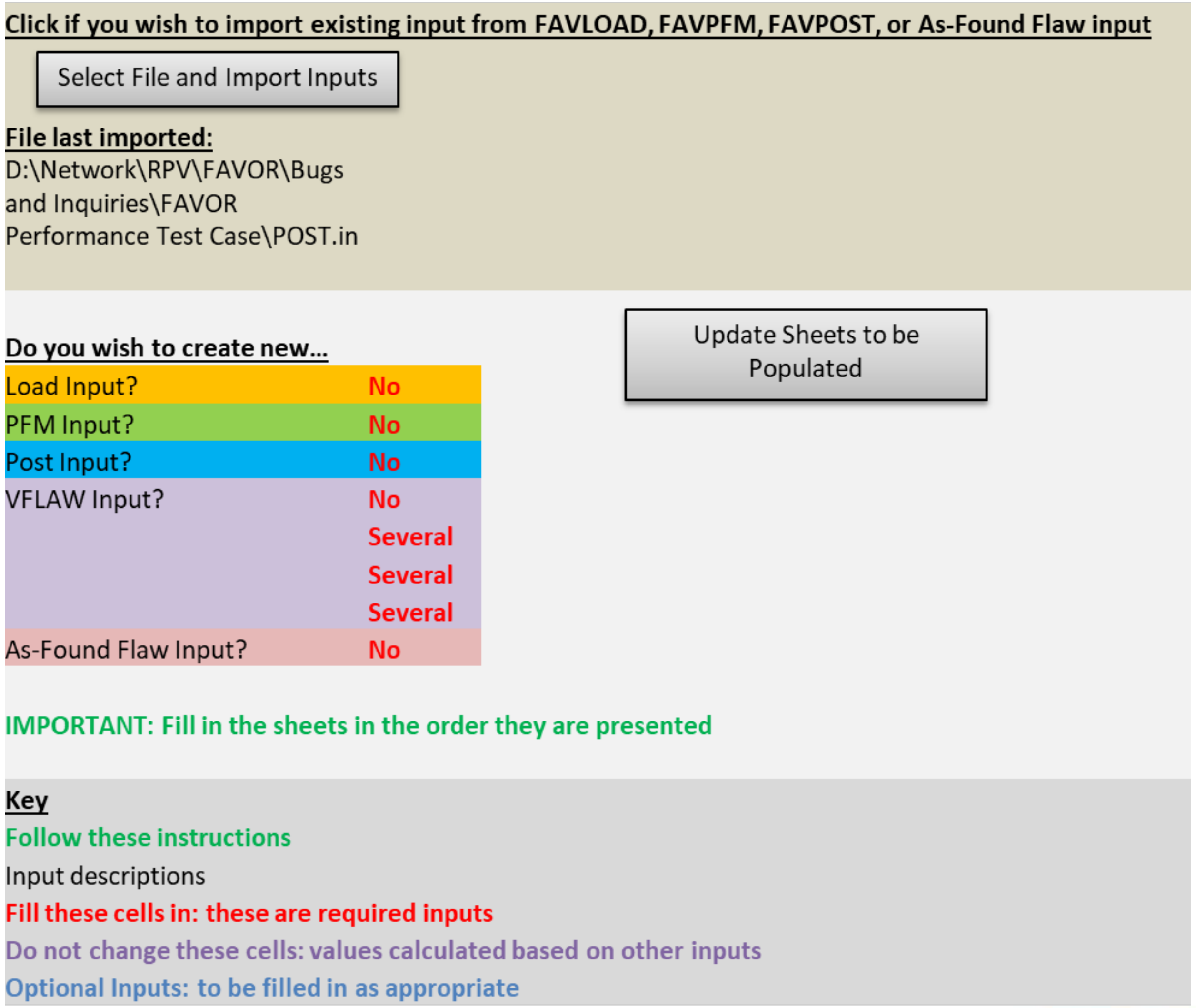

U.S.NRC
United States Nuclear Regulatory Commission
Protecting People and the Environment

FAVPRO Command Line Interface (CLI)

```
*************************************************
*                                               *
*               WELCOME TO FAVPRO               *
*                                               *
*  FRACTURE ANALYSIS OF VESSELS: PROBABILISTIC  *
*               VERSION 0.2.1                   *
*                                               *
*    PROBLEMS OR QUESTIONS REGARDING FAVPRO     *
*            SHOULD BE DIRECTED TO              *
*               PATRICK RAYNAUD                 *
*                 CHRIS ULMER                   *
*                     OR                        *
*                CHRIS NELLIS                   *
*                                               *
*  UNITED STATES NUCLEAR REGULATORY COMMISSION  *
*     Office of Nuclear Regulatory Research     *
*        e-mail: patrick.raynaud@nrc.gov        *
*       e-mail: christopher.ulmer@nrc.gov       *
*                     OR                        *
*       e-mail: christopher.nellis@nrc.gov      *
*                                               *
*************************************************
************************************************************
* This computer program was prepared as an account of      *
* work sponsored by the United States Government           *
* Neither the United States, nor the United States         *
* Department of Energy, nor the United States Nuclear      *
* Regulatory Commission, nor any of their employees,       *
* nor any of their contractors, subcontractors, or their   *
* employees, makes any warranty, express or implied, or    *
* assumes any legal liability or responsibility for the    *
* accuracy, completeness, or usefulness of any             *
* information, apparatus, product, or process disclosed,   *
* or represents that its use would not infringe            *
* privately-owned rights.                                  *
************************************************************
Specify at least one input file

Command line seen: "V:\Documents\Repos\FAVPRO\build\bin\favpro.exe"

Usages:
    favpro -li load_input -lo load_output
    favpro -fi pfm_input -lo load_output -fo pfm_output [-r seed] [-s surface] [-w weld] [-p plate] [-a as_found]
    favpro -ti post_input -fo pfm_output -to post_output [-r seed]
    favpro -li load_input -fi pfm_input -fo pfm_output [-lo load_output] [-r seed] [-s surface] [-w weld] [-p plate] [-a as_found]
    favpro -fi pfm_input -lo load_output -ti post_input -to post_output [-fo pfm_output] [-r seed] [-s surface] [-w weld] [-p plate] [-a as_found]
    favpro -li load_input -fi pfm_input -ti post_input -to post_output [-lo load_output] [-fo pfm_output] [-r seed] [-s surface] [-w weld] [-p plate] [-a as_found]

  options:
    -li load_input, --load_input load_input    Name of existing favload input file
    -lo load_output, --load_output load_output Name of favload output file
                                               NOTE: Will be overwritten if it exists
    -fi pfm_input, --pfm_input pfm_input       Name of existing favpfm input file
    -fo pfm_output, --pfm_output pfm_output    Name of favpfm output file
                                               NOTE: Will be overwritten if it exists
    -ti post_input, --post_input post_input    Name of existing favpost input file
    -to post_output, --post_output post_output Name of favpost output file
                                               NOTE: Will be overwritten if it exists
    -s surface, --surface surface              Name of surface breaking flaw file
                                               Only used for certain input options
                                               Default: S.DAT
    -w weld, --weld weld                       Name of weld flaw file
                                               Only used for certain input options
                                               Default: W.DAT
    -p plate, --plate plate                    Name of plate flaw file
                                               Only used for certain input options
                                               Default: P.DAT
    -a as_found, --as_found as_found           Name of as-found flaw file
                                               Only used for certain input options
                                               Default: F.DAT
    -r seed --random_seed seed                 Seed to be used to initialize random
                                               number sequence (optional)
                                               Specify 'random' to have the program
                                               choose one at random
                                               Uses default seed if unspecified
```



FAVPRO Automatic Input Generator (AIG)

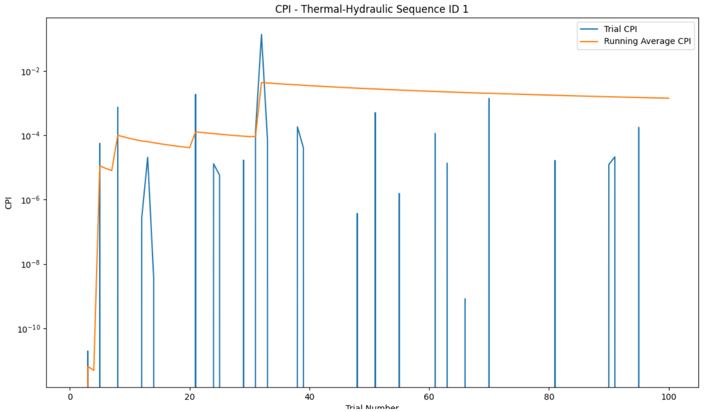


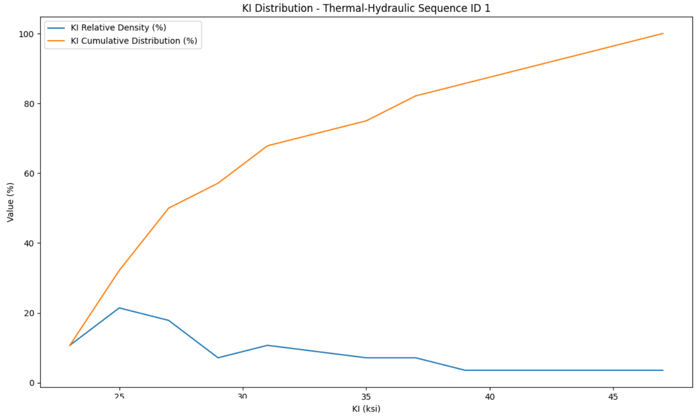
- Read in and convert old input files
- Produce new inputs
  - LOAD
  - PFM
  - POST
- Produce flaw inputs
  - VFLAW
  - As-Found Flaws

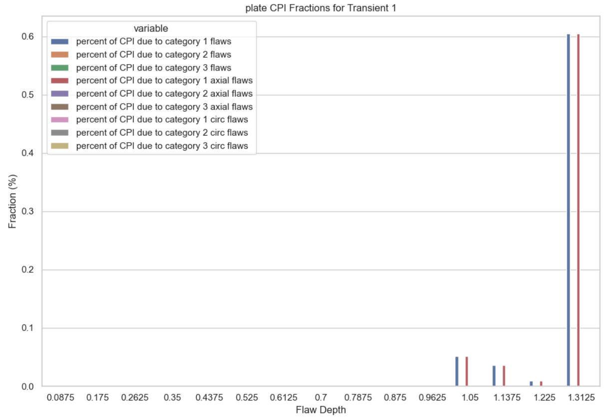
FAVPRO Output Visualization

- Python scripts
- Goal: Read in any FAVPRO output file and easily extract and visualize FAVPRO output data

```
PS V:\Documents\Repos\FAVPRO> python .\tests\integration\inputs\VT\FAVPRO_VT.py
Importing matplotlib (already installed)
Importing numpy (already installed)
Importing pandas (already installed)
Importing seaborn (already installed)
Importing xlsxwriter (already installed)

These are the JSON files available for visualization:

1. ver_5_ref.json

Which file would you like to visualize?
1

'ver_5_ref' directory and sub-directories created if they did not exist.

These are the keys available for visualization:
1. CPI output
2. CPF output
3. analysis output
4. All

Which key would you like to visualize?
4

Would you like to see the plots one by one? (y/n)
y

What would you like to do next?

1. I am done
2. I want to continue visualizing
1
PS V:\Documents\Repos\FAVPRO>
```







U.S.NRC
United States Nuclear Regulatory Commission
Protecting People and the Environment

FAVPRO User Group

- To obtain FAVPRO (or FAVOR):
  – Fill out the NRC Codes NDA
  – Once approved, the code executables (FAVPRO), input generator (FAVPRO-AIG), and visualization tool (FAVPRO-VT) are downloaded via NRC's BOX service
- All approved users automatically become members of the User Group
  – Annual meetings (hopefully more often in the future)
  – Newsletters (quarterly)
  – New code versions (as soon as they are available)
  – User input to the development team is strongly encouraged
    - Please tell us about bugs, desired new features, etc.
- Cost: free!
  – Could change at some point, but not in immediate future
- What about source code?
  – Can be obtained on special case-by-case basis
    - Need to show potential tangible benefit to NRC

Summary and Perspectives

- FAVPRO is a new modern tool to replace FAVOR
- Enhanced SQA pedigree and V&V testing
- Modern, modular, parallel code for enhanced performance, enhanced user experience, and enhanced adaptability
- New features:
  - As-found flaw modeling
  - Standard conforming K solutions where possible (ASME)
  - New embrittlement trend curves to reflect the latest standards and research
- **FAVPRO is a robust and resilient foundation that can be built upon to add new models, new probabilistic functionality, new materials, and new physical models to adapt to the rapidly evolving nuclear technology landscape**

FAVPRO Demonstration

- LOAD
- PFM deterministic
- PFM probabilistic
- POST
- AFF flaws

- LOAD+PFM+POST
- PFM deterministic
- PFM serial
- PFM parallel

- LOAD
- PFM deterministic
- PFM probabilistic
- POST