**MELCOR Code Development Status
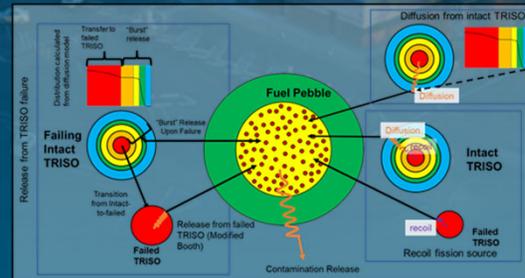EMUG 2023**

SAND2023-01791PE
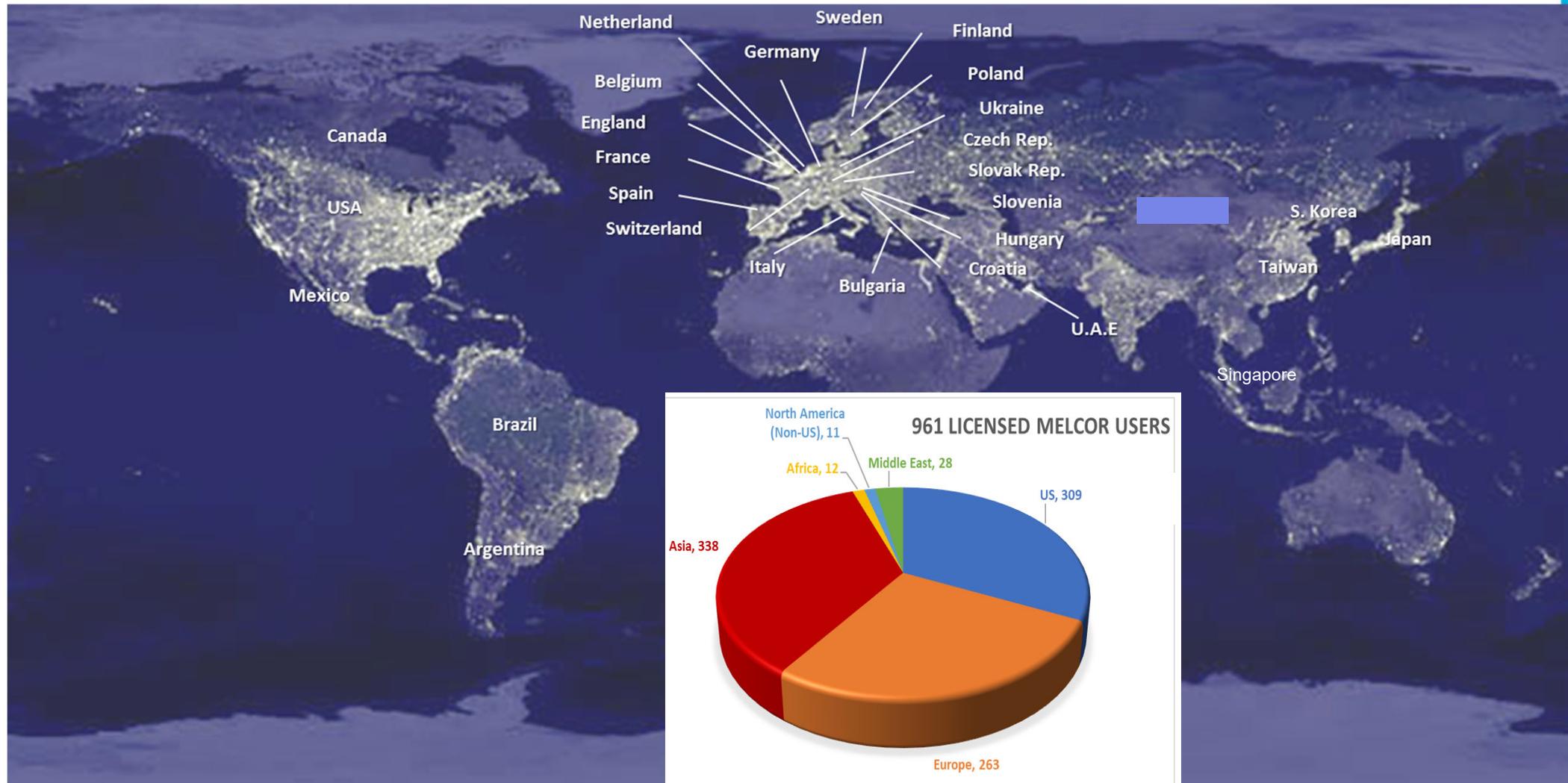
MELCOR
- New Modeling
- SQA
- Utilities
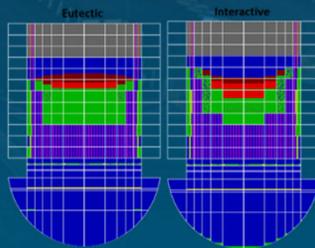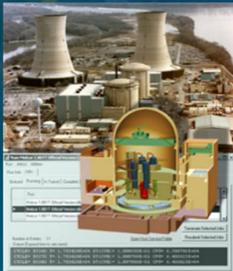
MELCOR Team, Sandia National Laboratories

PRESENTED BY

Larry Humphries, Sandia National Laboratories

International Use of MELCOR

961 LICENSED MELCOR USERS

- North America (Non-US), 11
- Africa, 12
- Middle East, 28
- US, 309
- Asia, 338
- Europe, 263

Recent MELCOR Workshops & Meetings

2021 Non-Nuclear facilities workshop
- One-week in-person
- May, New Mexico
- Another 1-week, in-person workshop scheduled for April 2022
- Funded by participants

2021 European MELCOR User Group (EMUG)
- Virtual – Hosted by NUBIKi, Budapest, Hungary
- April 14-16
- Two ½ day MELCOR workshop on MELCOR visualization

2021 CSARP/MCAP
- Virtual – U.S. Nuclear Regulatory Commission
- June 7-11, 2021

2021 Asian MELCOR User Group (AMUG)

Hosted by Singapore Nuclear Research and Safety Initiative - National University of Singapore
- November 2021
- Two ½ day MELCOR workshop on HTGR application

2022 Non-Nuclear facilities workshop
- 4-day in-person
- April, New Mexico
- Funded by participants

June 2022 in-person, week-long workshop in Bethesda, MD, USA
- 3 ½ days introductory material
  - Updated content
- 1 ½ days reactor applications
  - HTGRs, Spent Fuel Pools, Sodium Fire Modeling

2022 Asian MELCOR User Group (AMUG)
- Hosted by Singapore Nuclear Research and Safety Initiative - National University of Singapore
- November 2022
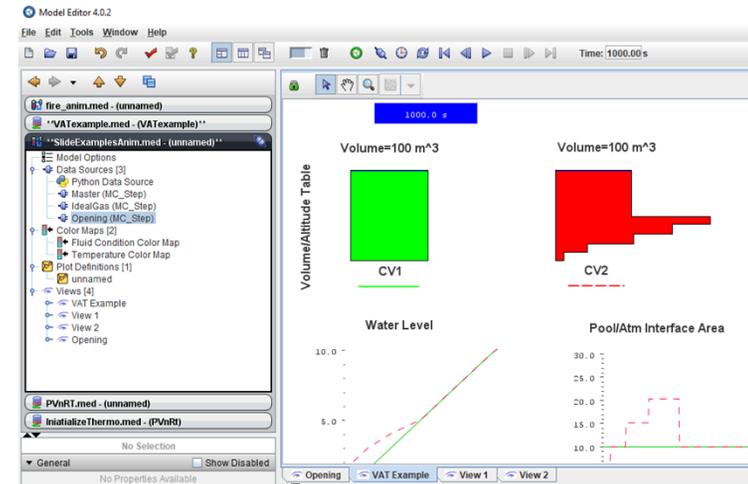- Two ½ day MELCOR workshop on HTGR application

Week-Long MELCOR Workshops

June 2022 in-person, week-long workshop in Bethesda, MD, USA

○ 3 ½ days introductory material
  - Updated content
    - MELCOR Overview
    - CF Package
      - Side-by-side comparison between SNAP & ASCII input
      - New CF exercises
        - integration, trips, Formula type CFs, Range, Vector CF
      - Updated for new features (ranges, vector CFs)
    - CVH Package
      - Simple 'SNAPlette' examples scattered throughout the presentation demonstrating basic concepts
        - VAT, Opening Heights, initializing thermodynamic conditions, static head
    - RN Package
      - Simple 'SNAPlette' examples scattered throughout the presentation demonstrating basic concepts
        - Deposition mechanisms, agglomeration, filters
    - COR Package
      - Simple 'SNAPlette' examples scattered throughout the presentation demonstrating basic concepts
        - DTDZ temperature model, quench model, oxidation model, flow blockage, virtual volumes, etc.

○ 1 ½ days reactor applications
  - HTGRs, Spent Fuel Pools, Sodium Fire Modeling

○ June 2023 in-person, week-long workshop in Bethesda, MD, USA

Volume-Altitude SNAPlette

Side-by-side SNAP/ASCII Input

SNAPlette: Molten Pool Modeling

- PWR Vessel
  - No penetrations modeled
- Finely nodalized lower plenum
  - 8 elevationa
  - 6 rings
- Lower Head Nodalization
  - 9 segments in hemisphere
  - 4 heads in cylindrical section
  - 10 layers thick
- 1 CV external with water pool
- 1 CV in lower plenum
- Initialized with PD in lower plenum at 2250 K
  - Effective UO2/ZrO2 Melt Temperature = 2200 K
- Uses 'model' to calculate heat transfer from particulate debris to lower head

- Things to observe:
- Temperature peak at metallic molten layer
- Thinning of vessel at bottom
- Things to try
  - This uses the interactive temperature model. Try using the eutectic model.
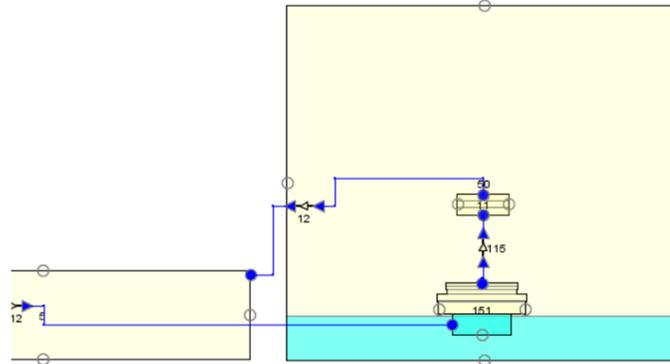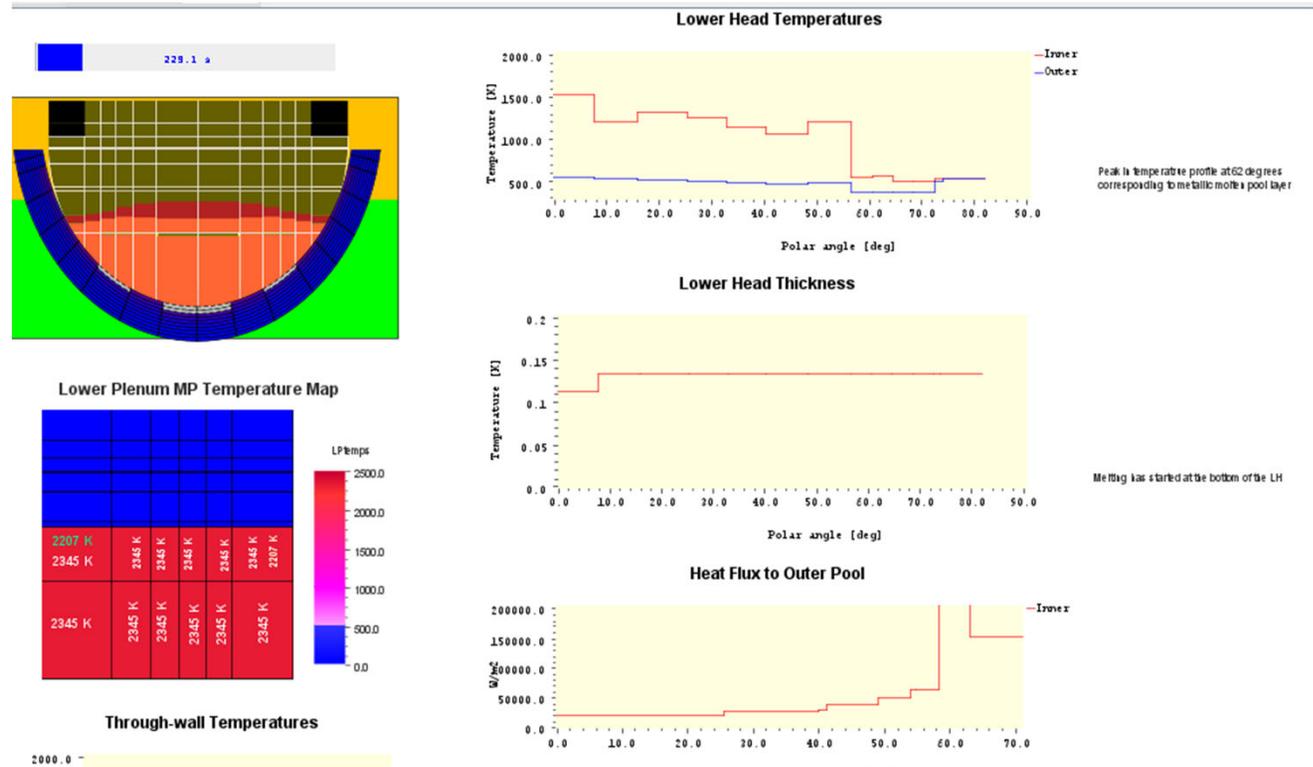
- Input Files
  - MoltenPool.med
  - LowerHead_anim.med
- Job Stream/Data Source
  - MPool

225.1 s

Lower Head Temperatures

Peak in temperature profile at 62 degrees corresponding to metallic molten pool layer

Lower Head Thickness

Melting has started at the bottom of the LH

Lower Plenum MP Temperature Map

2207 K 2345 K	2345 K	2345 K	2345 K	2345 K	2345 K 2207 K
2345 K	2345 K	2345 K	2345 K	2345 K	2345 K

LP temps

Heat Flux to Outer Pool

Through-wall Temperatures

2000.0

MELCOR Model Development

MELCOR Code Development

Models

- Convecting Molten Pool
- Curved Lower Head
- Stefan Model
- Point Kinetics
- Smart Restart
- H2 Production
- RN Activity (BONUS)
- Turbulent Deposition
- Mechanical Fan Cooler
- Multi-rod
- CORQUENCH
- LHC
- Eutectics
- Vector CFs
- Homologous Pump
- Resuspension
- Radiation Enclosure

Emphasis

- Conversion from F77 to F95
- MELCOR 2.X Robustness & User Flexibility
- HTGR Models
- SFP Models
- Molten Pool / Lower Head
- Code Performance Improvements
- Na Fire Models
- Non-LWR Models
- SMR Models

Official Release

- MELCOR 1.8.5
- MELCOR 1.8.6
- MELCOR 2.0 (beta)
- M2.1.1576
- M2.1.3649
- M2.1.4803
- M2.1.6342
- MELCOR 2.2

Year

2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020

Fusion

- Neutron Beam Injectors (LOVA)
- Li Loop LOFA transient analysis
- ITER Cryostat modeling
- Helium Lithium
- Helium Cooled Pebble Bed Test Blanket (Tritium Breeding)
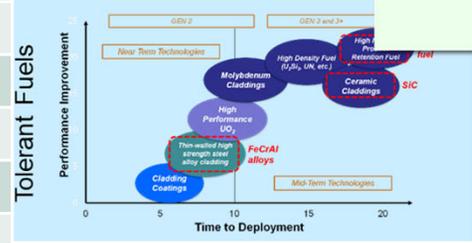
Spent Fuel

HTGR Reactors

- Helium Properties

Sodium Reactors

- Sodium Properties
- Sodium Equation of State
- Sodium Thermo-mechanical properties
- Containment Modeling
- Sodium pool fire model
- Sodium spray fire model
- Atmospheric chemistry model
- Sodium-concrete interaction

Molten Salt Reactors

- Properties for LiF-BeF2 have been added
  - Equation of State
  - Thermal-mechanical properties

M2x Official Code Releases

Version	Date
2.2.18019	December 2020
2.2.14959	October 2019
2.2.11932	November 2018
2.2.9541	February 2017
2.1.6342	October 2014
2.1.4803	September 2012
2.1.3649	November 2011
2.1.3096	August 2011
2.1.YT	August 2008
2.0 (beta)	Sept 2006

Tolerant Fuels

Non-Reactor Applications

Advanced Reactor Applications

LWR/Non-LWR/ATF Fuels Development

LWR and General MELCOR development

Advanced Technology Fuels (ATF)

Non-LWR Reactors
  ◦ **HTGR**
  ◦ **Sodium**
  ◦ **Molten Salts**

Spent Fuel Pools
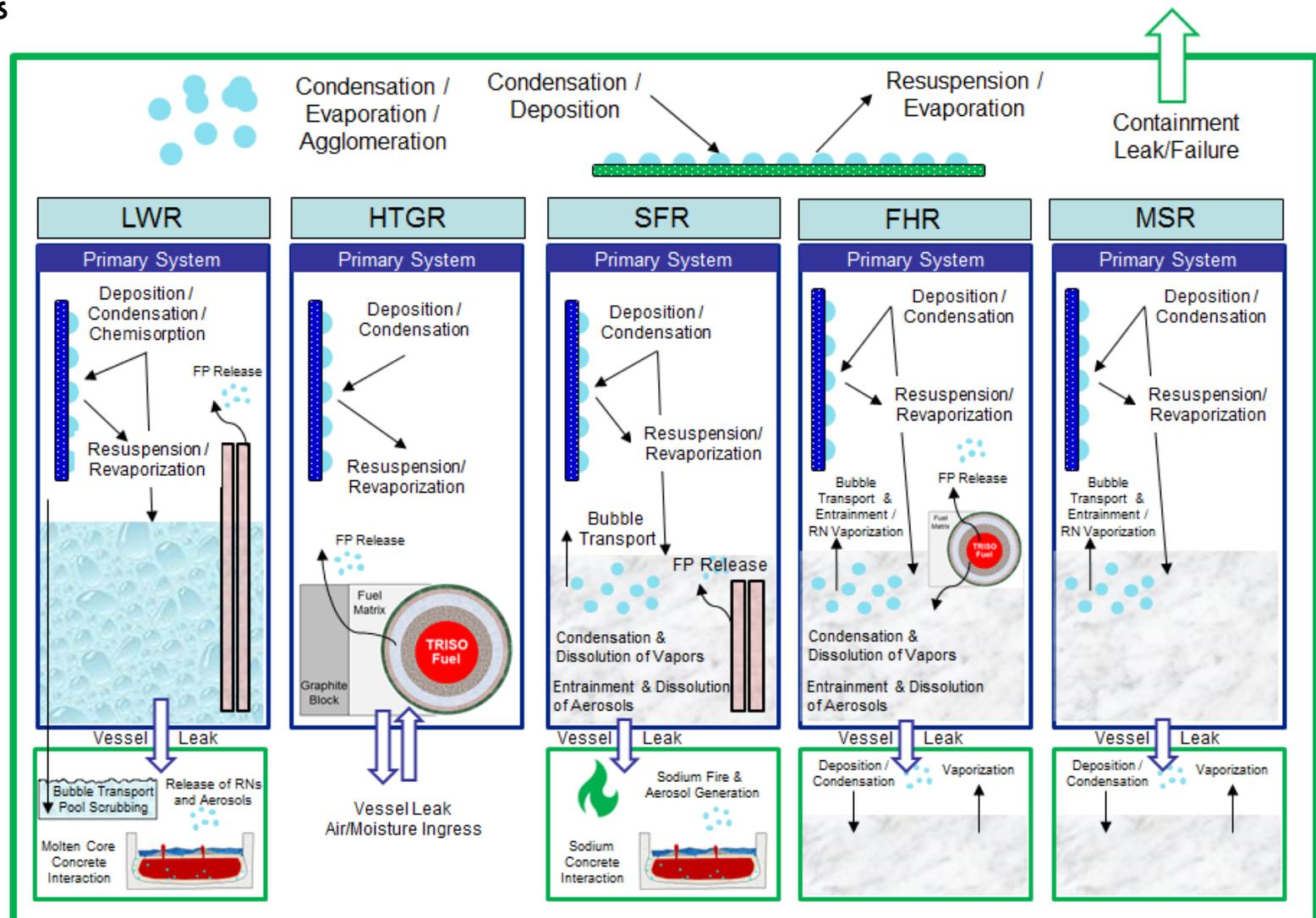
MELCOR HTGR modeling

Overall goals of the HTGR fission product release
model include:
- Predict radionuclide distributions within fuel
- Predict radionuclide release from fuel elements to coolant

Radionuclide release and transport model
- Diffusion transport of fission product species
- MELCOR transient/accident solution
  methodology
  - Steady-state diffusion
  - Steady-state transport
  - Transient diffusion/transport

HTGR Specific models
- Analytic release
- Energy/temperature for temperature-dependent
  diffusion
- Graphite oxidation
- Intercell and intracell conduction
- Convection & flow
- Point kinetics
- Dust generation and resuspension
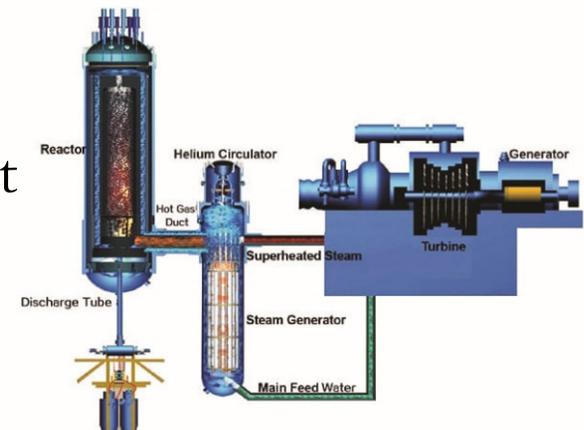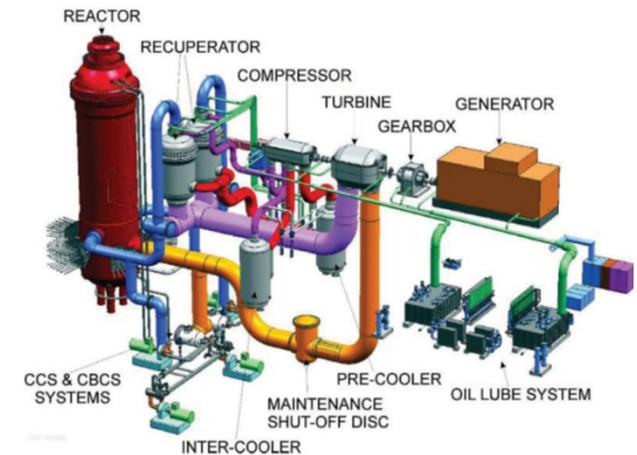
HTGR Components

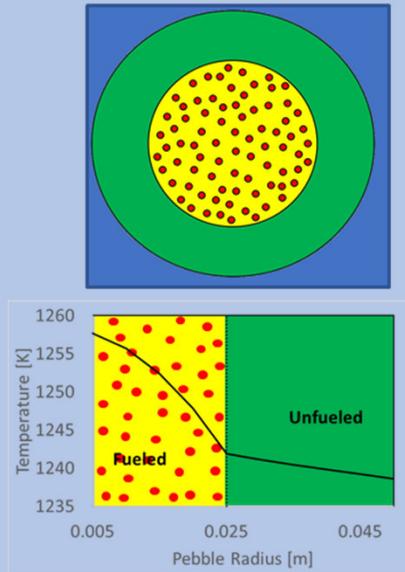## Pebble Bed Reactor (PBR) Fuel/Matrix Components

- Fueled part of pebble
- Unfueled shell is modeled as separate component (Matrix)
- Fuel radial temperature profile for sphere
  - Provides peak and surface pebble temperature
  - Modified for unfueled central core

Fueled pebble core

Unfueled pebble core

Prismatic Modular Reactor (PMR) Fuel/Matrix Components

- More "rod-like" geometry
- Fuel compacts represented as fuel component
- Part of hex block associated with a fuel channel is matrix component
- Fuel radial temperature profile for cylinder
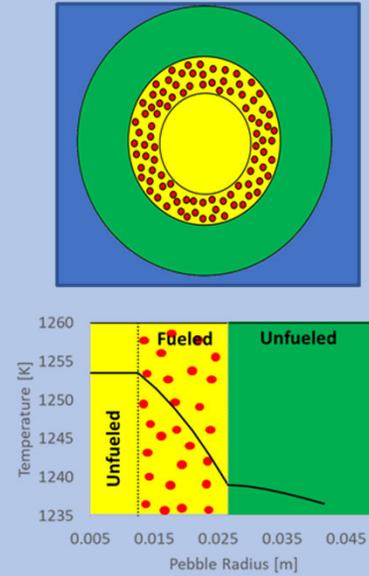
Legend

TRISO, GRAPHITE — Fuel (FU)

GRAPHITE — Matrix (MX)

Fluid B/C

TRISO (FU)
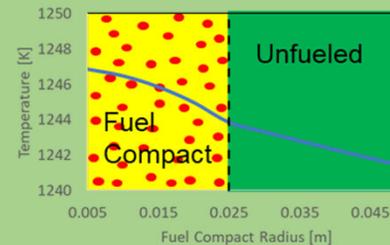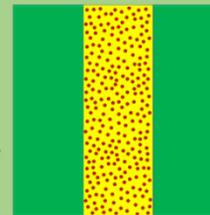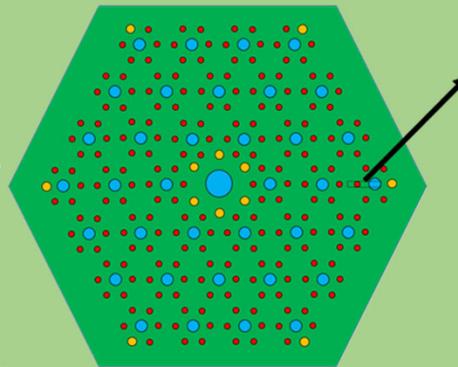
Sub-component model for zonal diffusion of radionuclides through TRISO particle

MELCOR/CONTAIN-LMR Implementation

Sodium Reactors

Implement sodium as replacement to the working fluid for a MELCOR calculation

- Implement properties & Equations Of State (EOS) from the fusion safety database
  - Na (tpfna), FLiBe (tpffi), Pb-Li (tpflipb), He (tpfhe), N2(tpfn2)
- Implement properties & EOS based on SIMMER-III

Implementation and Validation of:
- Sodium spray fires
  - Based on NACOM spray model from BNL
  - Input requirement: fall height, mean diameter and source
    - Droplet acceleration model
  - Internal droplet size distribution (11 bins) from Nukiyama-Tanasama correlation
  - Reactions considered:
    - (S1) $2\,Na + \frac{1}{2}O_2 \rightarrow Na_2O$,
    - (S2) $2\,Na + O_2 \rightarrow Na_2O_2$
- Sodium pool fires
  - Based on SOFIRE II code from ANL
  - Reactions considered:
    - $2\,Na + O_2 \rightarrow Na_2O_2$, 10.97 MJ/kg
    - $4\,Na + O_2 \rightarrow 2\,Na_2O$, 9.05 MJ/kg
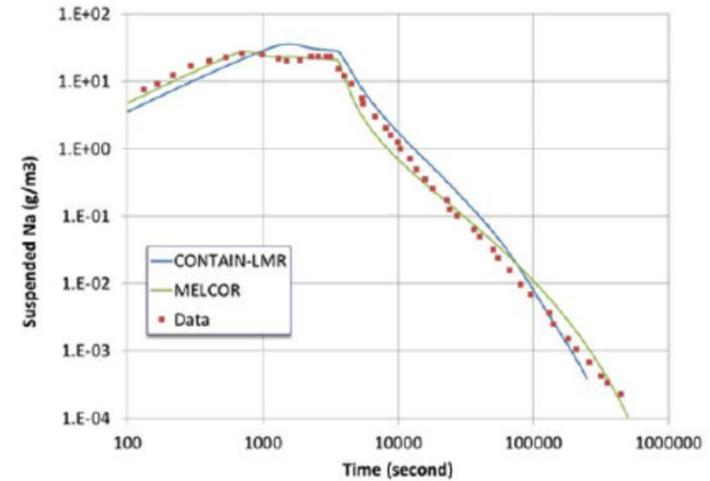
Figure 33. Suspended Na Aerosol Mass - AB1
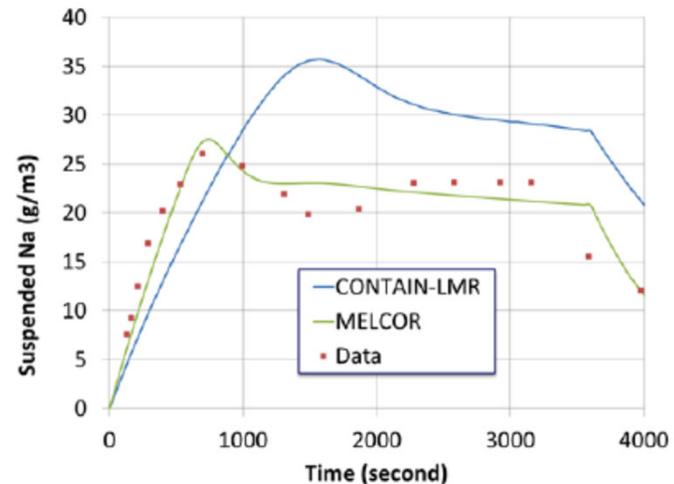
Figure 34. Suspended Na Aerosol Mass-AB1

SFR In-Pin Fuel/FP Modeling

Sodium Reactors

MELGEN

Burned Pin

Computed at transient time zero:
- Bond sodium contents and volume
- Pin plenum contents and volume
- Fuel porosity contents and volume
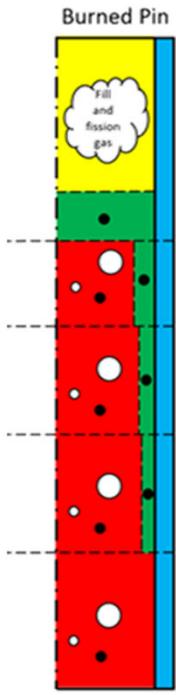
Initial condition specified by user in MELGEN

MELCOR

In-Pin Release

In-pin dynamics up to gap release, clad collapse:
- Fuel can melt, fission gas moves to molten cavity in fuel
- Fuel release model moves radionuclide mass into gap/plena
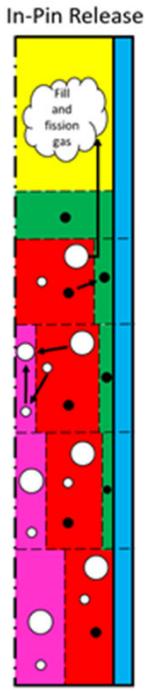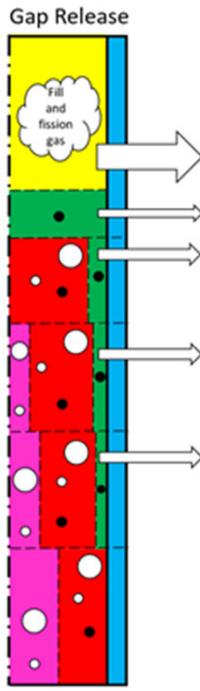
Gap Release

Gap release forces pin plenum release in corresponding core ring (assumption for now)

Pin Failure & Release

Fuel collapse coincides with clad collapse (assumption for now)

Legend:
- Solid Fuel
- Molten Fuel
- Sodium Bond
- Gas Plenum
- Cladding
- • Solid FP
- ○ Gaseous FP Closed Porosity
- ◯ Gaseous FP Open Porosity

Fill and fission gas
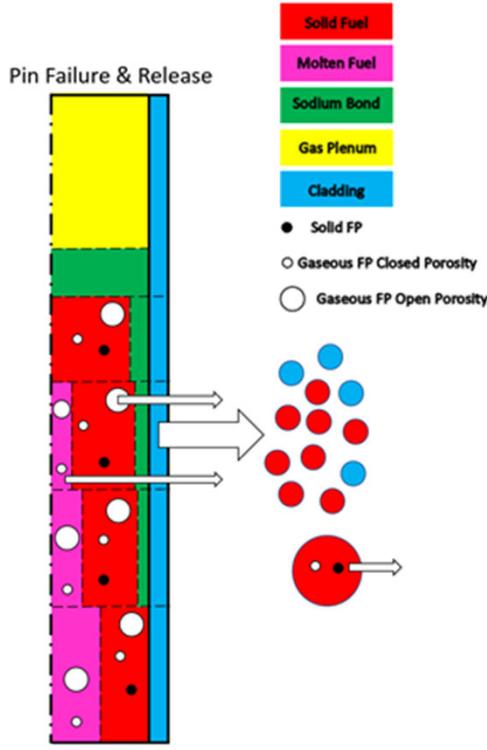
MELCOR Heat Pipe Reactor Modeling

Sodium Reactors

HPs replace conventional convective heat transfer between the fuel and coolant channel with the energy transfer from the fuel to the evaporative region of the HP. Heat rejection from the HP model at the condensation interface is transferred to the CVH package.

Distinct wall and working fluid region nodalization. MELCOR accommodates HP models of different fidelity through a common interface.

- Model 1: working fluid region modeled as high thermal conductivity material

- Model 2 (new in 2020 release): thermodynamic equilibrium approximation of working fluid (sodium or potassium EOS). P, T and liq/vap fraction evolve in time. Sonic, capillary and boiling limits enforced.

- If needed, higher fidelity models straight-forward to implement to interface.

MELCOR heat transfer paths enable radial (lateral) heat transfer in the core among multiple HP regions and to other MELCOR structures.

- Thermal resistance network approximation across heterogeneous domain

At failure, HP materials and regions transition to COR and CVH modeling.
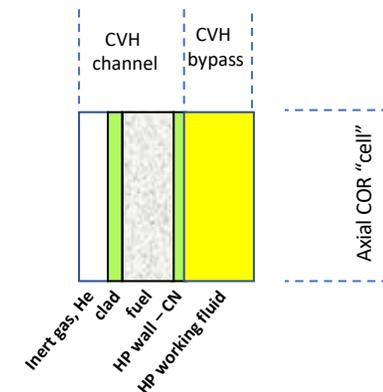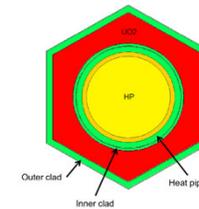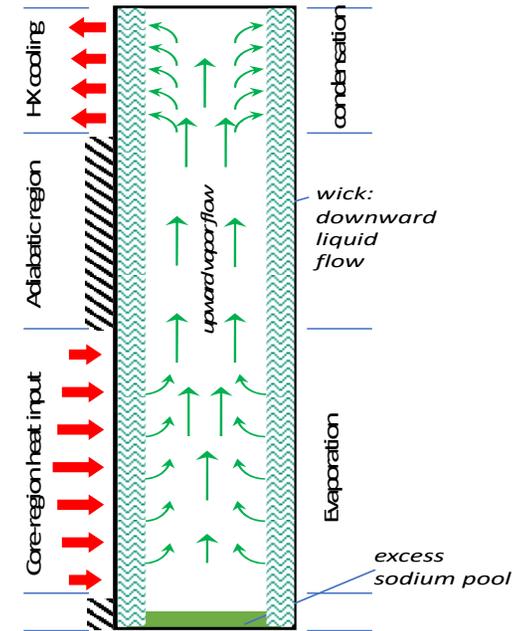
Molten Salt Reactors

Properties for LiF-BeF2 have been added
- Equation of State (current capability)
- Thermal-mechanical properties (current capability)
- EOS for other molten salt fluids would need to be developed

Two reactor types envisioned
- Fixed fuel geometry
  - TRISO fuel models (current capability)
- Liquid fuel geometry
  - MELCOR CVH/RN package can model flow of coolant and advection of internal heat source with minimal changes (current capability)
  - Framework for fission thermal power generation without COR package (current capability)
    - Control volumes defined as "vessel" type or "loop" type
    - Specify axial/radial power profile shape and magnitude (over "vessel" type CV's)
    - Specify flow paths constituting transitions between vessel and loop
    - Preserve decay heat deposition capabilities (CVH/RN1 tracks decay heat from aerosol/vapor already)
  - COR package representation no longer applicable, but structures can be represented by HS package (current capability)
  - Calculation of neutronics kinetics for flowing fuel (under development)

Radionuclide Transport

Molten salt fission product release models (under development)

Thermochemistry modeling using Thermochimica (under development)
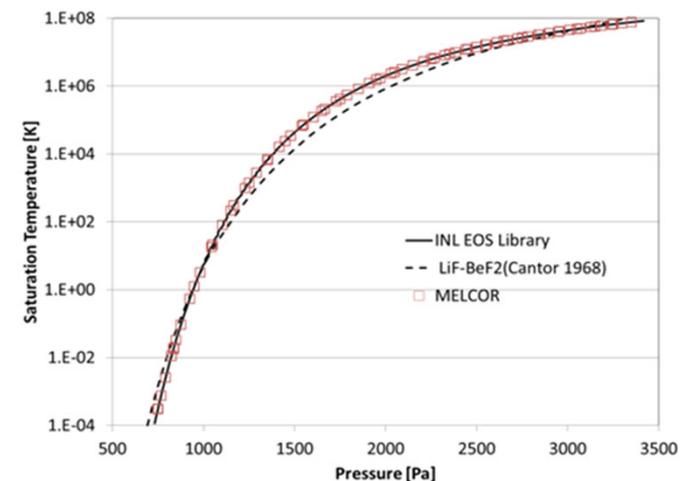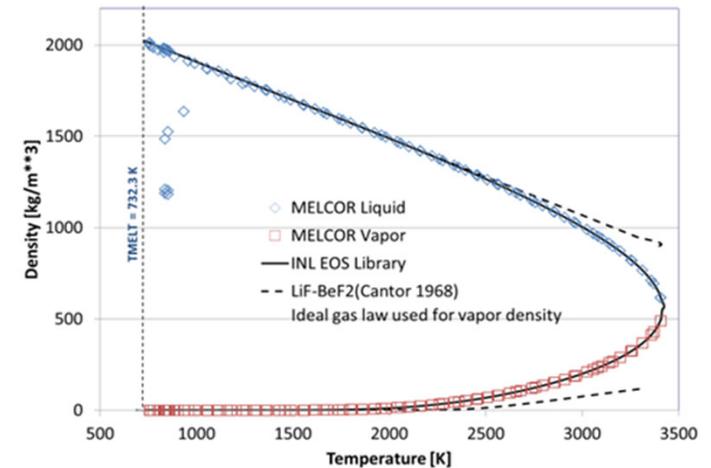
Generalized Radionuclide Transport and Retention (GRTR)

Molten Salt Reactor

Capability to define arbitrarily many new radionuclide forms that are:
- Sectionwise distributed (with size bins) or non-sectionwise distributed (no size bins), and
- Identified with either pool phase, atmosphere phase, or HS surface (deposition)

Built-in radionuclide forms are preserved (vapor and aerosol in the pool or atmosphere phase) as is heat structure deposition of built-in forms

User defines mass transfer processes applying between forms within classes:
- Between GRTR forms, or
- From GRTR forms to built-in forms, or
- From built-in forms to GRTR forms, but NOT from built-in forms to built-in forms

Unique rule for each class for a given mass transfer process, and the rule is either control function specified or some built-in modeling alternative

Molten Salt Chemistry and Radionuclide Release

Molten Salt Reactor

Model Scope

Evaluate thermochemistry

- Gibbs Energy Minimization with Thermochimica
- Provides solubilities and vapor pressures

Thermodynamic database

- Generalized approach to utilize any thermodynamic database
- An example is the Molten Salt Thermal Database
  - FLiBe-based systems
  - Chloride-based systems

Radionuclides grouped into forms found in the Molten Salt Reactor Experiment
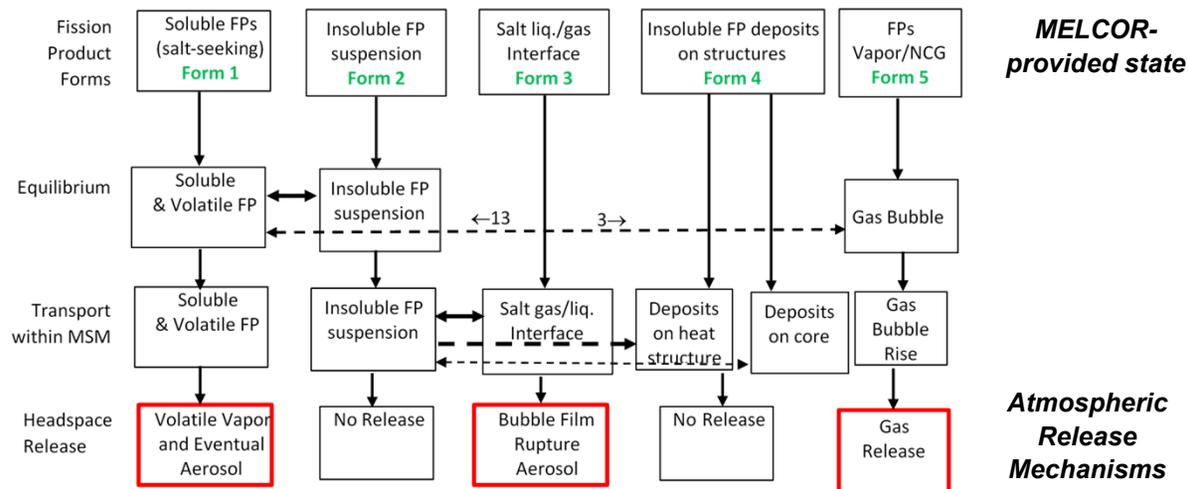
Initial Model Form

Solubility determined from empirical evidence (P. Britt ORNL 2017)

Solubilities mapped to 17 MELCOR fission product classes

Insoluble MELCOR classes are assigned to be colloidal

Fluid-Fuel Point Reactor Kinetics (FFPRK) Model

Molten Salt Reactor

Circulating fuel point kinetics model

$$\frac{dP(t)}{dt} = \left(\frac{\rho(t) - \bar{\beta}(t)}{\Lambda}\right)P(t) + \sum_{i=1}^{6} \lambda_i C_i^C + S_0$$

$$\frac{dC_i^C(t)}{dt} = \left(\frac{\beta_i}{\Lambda}\right)P(t) - \left(\lambda_i + \frac{2}{\tau_C}\right)C_i^C(t) + \left(\frac{V_L}{V_C}\right)\left(\lambda_i + \frac{2}{\tau_L}\right)C_i^L(t), \qquad i = 1 \dots 6$$

$$\frac{dC_i^L(t)}{dt} = \left(\frac{V_C}{\tau_C V_L}\right)C_i^C(t) - \left(\lambda_i + \frac{1}{\tau_L}\right)C_i^L(t), \qquad i = 1 \dots 6$$

$$\bar{\beta}(t) = \beta - \beta(t)_{lost} = \beta - \left(\frac{\Lambda}{P(t)}\right)\sum_{i=1}^{6} \lambda_i C_i^L(t)$$

Show Legend

MSRE Pump Coast-down Compensating reactivity.

Benchmark to MSRE pump coast-down
- Critical at "zero-power" (10 W) and full flow, then initiate a coast-down to zero flow
- Control system intervenes to preserve criticality, thus the control reactivity trace maps flow reactivity
- Pump flow data and control system reactivity response documented in ORNL-4233
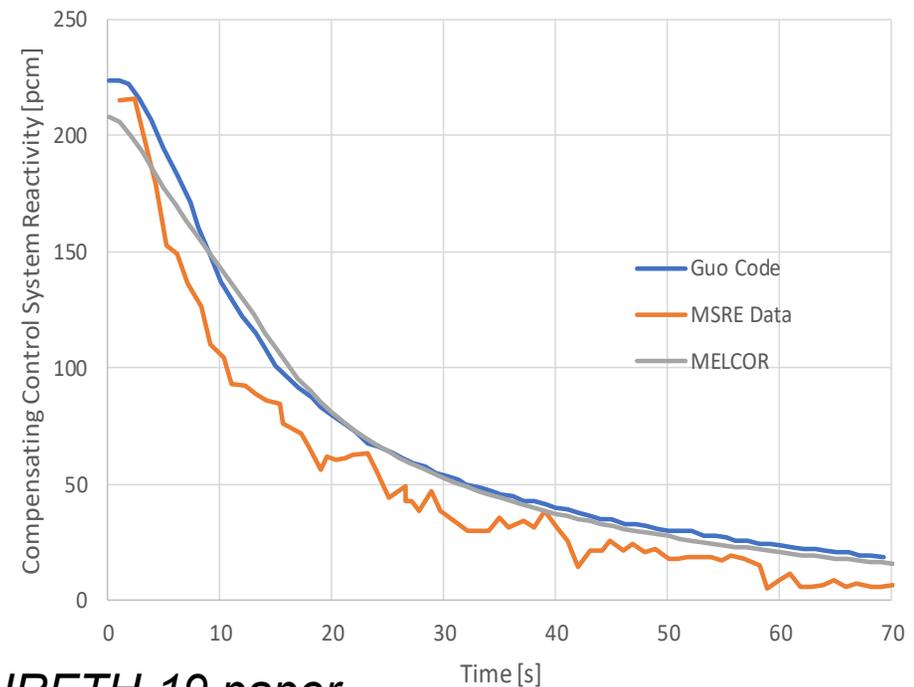
Documented in forthcoming NURETH-19 paper

Non-LWR Demonstrations Calculations

Develop an understanding of non-LWR beyond-design-basis-accident behavior

- Provide insights for enhancing source-term regulatory guidance
- Provide NRC staff with an understanding of non-LWR beyond-design-basis-accident behavior
  - Identify key accident characteristics influencing source terms
  - Highlight important uncertainties influencing accident progression and source terms

Demonstrate MELCOR for non-LWR beyond-design-basis accidents

- Distribution of non-LWR input models with MELCOR executable
- Shakedown the MELCOR modeling and identify missing models or data needs

MELCOR Applications

Non-LWR Demonstrations Calculations

MELCOR Applications

Demo Case Selection
◦ Demonstrate/exercise breadth of non-LWR physics modeled by MELCOR
◦ Consider NRC priority relative to timing of licensing submittal
◦ Utilize non-proprietary designs
◦ Build on existing decks where possible

Demo Cases

Heat Pipe Reactor - June 29, 2021
◦ Megapower Heat pipe reactor
  ◦ Transient Overpower (TOP)
  ◦ Loss-of-Heat Sink (LOHS)
  ◦ Anticipated Transient Without Scram (ATWS)

PBMR-400 - July 20, 2021
◦ Pebble bed HTGR
  ◦ Depressurized Loss of Forced Circulation (DLOFC)
  ◦ Anticipated Transient Without Scram (ATWS)

Flouride Salt Cooled High Temperature Reactor - September 14, 2021
◦ Pebble bed FHR
  ◦ Anticipated Transient Without SCRAM (ATWS)
  ◦ Station Blackout (SBO)
  ◦ Loss of Coolant Accident (LOCA)

Molten-salt-fueled Reactor – September 13, 2022
◦ ORNL Molten Salt Reactor Experiment (MSRE)
  ◦ Spill of molten salt into the reactor cell

Sodium-cooled fast reactor – September 20, 2022
◦ Advanced Burner Test Reactor (ABTR)
  ◦ Unprotected transient over-power (UTOP)
  ◦ Unprotected Loss of Flow (ULOF)
  ◦ Single blocked fuel assembly

◦ https://www.nrc.gov/reactors/new-reactors/advanced/nuclear-power-reactor-source-term.html

Electric Heater Element Modeling

Electrical Heater Elements are frequently used to simulate nuclear heat generation in fuel rods

- Voltage across bundle and resistance of fuel rods determines power generation
- Local resistivity dependent on local temperature.
- Temperature feedback determines the power distribution in the bundle

MELCOR modeling

- Historically, users would model with a specially compiled MELCOR executable
- With the release of MELCOR 2.0, an internal model was available not requiring special compilation.
  - Parameters were specified by input.
  - Model was severely limited since it was specifically based on one particular experiment.
  - Did not allow more than 2 rings for heater elements
- New model uses a high level control function that leverages ranges and vector control functions.
  - User specifies a range of COR cells and power functions for each ring in that range.
  - Allows multiple materials for elements (W, Cu, Mo)
  - Allows specification of electric losses external to elements
  - HTML expanded to include electrical heating element
    - resistivity as a function of temperature.
    - Video of temperature evolution.

LWR and General Development

Power Supply

Power Supply
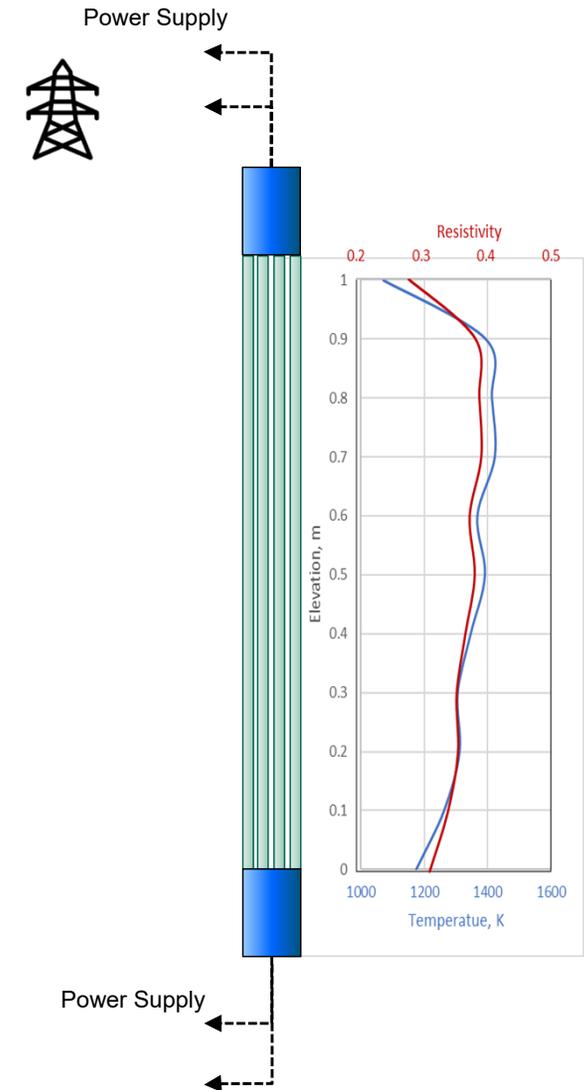
Improvements to Lower Head Model

Spheroidal Lower Head
- Generalization of hemisphere model to allow for an oblated lower cap
- Users specify two dimensions: a horizontal radius and a vertical radius

Spheroidal Lower Head

```
!      RCOR  RVESS  ILHTRN  DZRV  DZLH  ILHTYP      ELLIPA  ELLIPC
COR_VP 1.8   2.2    RVESS   0.24  0.24  SPHEROIDAL  2.5     1.25
```

Melting Lower Head
- Debris relocating to the lower head contains sufficient decay heat to lead to melting of the interior surface of the lower head.
- Though MELCOR already accounts for the reduction in load-bearing material as the lower head melts, it does not allow the melted material to become part of the COR package where it
  - can affect heat transfer (focusing effect) of molten materials,
  - can be oxidized (contributing to hydrogen production),
  - can be transferred to the CAV package for MCCI.
- This code modification will source steel into the calculation along with the associated thermal energy where the COR package then takes control for further relocation

Lower Head Melt Mass
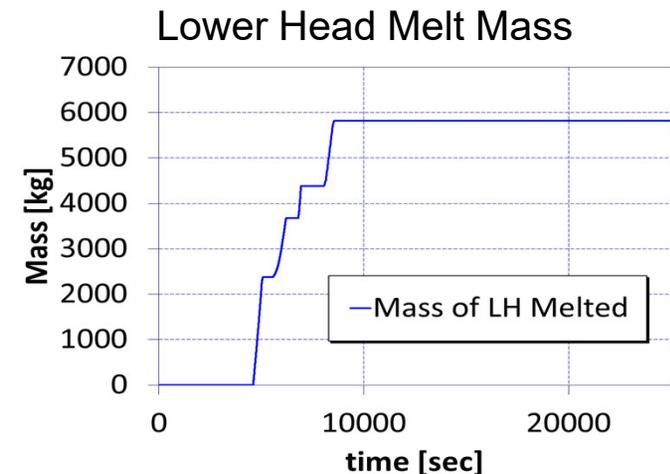
Limitations:

Currently only accounts for melting carbon steel

Mass, energy, and available volume sourced into the COR package

Currently only SS can receive the melting carbon steel

Material Property Development
(New in 2020 Release)

User-defined Materials

◦ Default material properties can be templated onto new materials

◦ Can be defined for COR with extra input

◦ Emissivity, Viscosity, Thermal expansion coefficient, Oxidation behavior

- COR User-Defined Material

```
MP_ID       NewZirc  COR-USER-METAL   NUZR
MP_BHVR    Zircaloy  METAL   OXIDATION-MODEL
EXEMPT
MP_BETMU   1.65E-5   3313.0  1.076E-3
MP_COREMIS CF  'NewEmissivity'  0.0001  0.9999
```

- Initialize Mass in component

```
COR_NMAT 1 ! IA    IR   IC  USRM1   USRM2   USRM3
USRM4
          1   ALL  ALL CL  0.0     0.0     64.2   0.0
```

- Associate with metal pool

```
COR_LAY 1 ! LAYMAT   MPLAY
          1   NUZR     UPPER  ! Associate with MP2
```

Generalized Oxidation Model
(New in 2020 Release)

Historically, MELCOR had a specific set of oxidizable material:

◦ Zirconium, Stainless-steel, Graphite, B_4C, Aluminum
◦ Now extended to use the user-defined materials (UDMs)

General Oxidation Model makes use of the new UDMs to create a new oxidizable material.

◦ Define a reactant core material, COR-USER-METAL, and its oxide product, COR-USER-OXIDE.User permitted to fully specify material properties
◦ May use templating or be wholly user-defined

```
MP_ID FeCrAl COR-USER-METAL UFCA
MP_BHVR     Itself  METAL   OXIDATION-MODEL
EXEMPT
MP_BETMU   3.1e-5 7.7 3313. 1.076e-3
MP_COREMIS linear - 0.0001 0.9999
0.042003702 0.0003474
MP_PRTF 4 ! NPAR PROPERTY DEFAULT/TF/CF(may
be only for THC)
           1 ENH FCA-IntEn
           2 CPS FCA-SpHeat
           3 THC FCA-Conduct TF
           4 RHO FCA-Density


MP_ID FeCrAl-Oxide COR-USER-OXIDE UFCAO
MP_BHVR ITSELF
MP_PRC 5180.0 1901.0 687463.0 193.4102395
MP_COREMIS linear - 0.0 1.0 0.7 0.0
MP_BETMU 3.1e-5 7.7 3313. 1.076e-3
MP_PRTF 4 ! NPAR PROPERTY DEFAULT/TF/CF(may
be only for THC)
           1 ENH FCAO-IntEn
           2 CPS FCAO-SpHeat
           3 THC FCAO-Conduct TF
           4 RHO FCAO-Density
```

Current Model Development

LWR and General Development

Metavolume approach to CVH modeling

Integration of Origen and MELCOR

Multiple condensable fluid option

Updated candling model for material in motion

MELCOR definition of a "meta-volume" region

"a grouping of interconnected CVs that share the same reference *thermodynamic* pressure $\overline{P}_q$"

Kinematic Pressure ($\check{P}_i$)
- Defined for each control volume "*i*".
  - As in current MELCOR
- Driving potential for flow
  - The kinematic pressure is used to enforce mass conservation between CVs within a meta volume region.

Thermodynamic Pressure, $\overline{P}_q$, is a function of:
- Used in EOS evaluations
- the total geometric volume $\overline{V}_q$,
- the internal energy $E_{i,\varphi}$ of each field φ ($\varphi = 1,2$) in the meta volume CVs "*i*", and
- the constituent masses "*m*" in each field in all of the CVs, $M_{i,m,\varphi}$.
- **Alleviates artificial pressure instabilities from vanishing volumes**

$\overline{P}_q$ Only one value for the entire meta-volume group of CVs.

$\check{P}_6$ CV 6

$\check{P}_7$ FP 8   FP 9   CV 7

$\check{P}_4$ CV 4   FP 4 $\check{P}_5$ CV 5

$\check{P}_8$

$\check{P}_2$ CV 2   FP 2 $\check{P}_3$ CV 3

FP 10   CV 8

$\check{P}_9$

$\check{P}_1$ CV 1   FP 1   FP 11   CV 9

Meta-volume 1
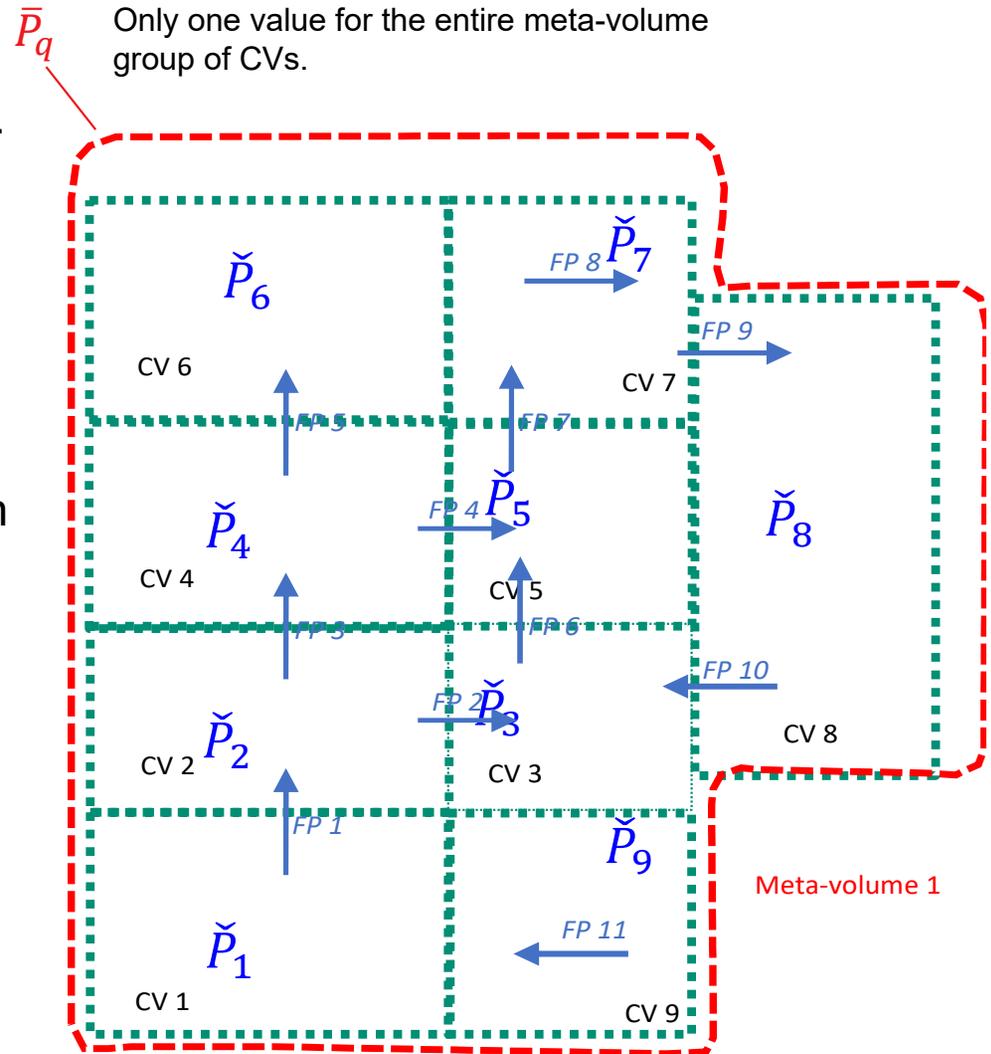
Schematic of Computational Coupling Scheme

LWR and General Development

ORIGEN | MELCOR

Transmutation of radionuclides step based on previous time step values	Element-wise masses and decay heats in ORIGEN regions

Elemental masses in MELCOR core cells, control volumes, heat structures, core structures

Partition elements across chemical classes
Instantaneous chemistry

Release and transport of chemical classes

"Read"-out elemental masses
Class to element mapping

Accumulate elemental mass *changes* across ORIGEN regions

System cycle time step
$t + \Delta t$

Subcycle time step to utilize ORIGEN interpolation

** Will explore computational efficiencies of interpolation over cycle rather than ORIGEN time advancement

MELCOR 2.2 r2023.0 Code Release (March, 2023)

Volume I: User Guide

SAND2023-10997O

Volume II: Reference Manual

SAND2023-10994 O

Volume III: Assessments

SAND2015-6693 R

MELCOR Modeling Guide (Volume IV)

Software Quality Assurance

- **User Guidance**
  - MELCOR has a steep learning curve and guidance is needed to help new users learn how to develop input decks.
    - Generate non-proprietary plant decks
      - BWR, PWR, SFP
    - Volume IV references these sample plant decks
  - Provide meaningful insights, recommendations, demonstrations of modeling methodology in a formal report for many commonly asked questions across much of the model space
  - Describe pitfalls and methods for troubleshooting and assessing results.
    - How to address code execution problems
    - How to review results to know the code is giving reasonable results
- **Best Practices**
  - Provide guidelines for appropriate use of the code in modeling severe accidents.
    - Recommended models and model options

Topics
- COR/CVH Nodalization
- HTGR Modeling
- Heat Pipe Modeling
- Molten Salt Modeling
- Sodium Reactor Modeling
- Spent Fuel Pool Modeling
- Containment DBA
- Numerical Variance
- Uncertainty Analysis
- FL/CVH Modeling
- Steady State Initialization
- Radionuclide Class Modeling
- MELCOR/MACCS Integration
- Troubleshooting MELCOR runtime issues
- Lower Head Modeling
- Heat Structure Modeling
- Cavity Related Modeling
- Fusion Applications

MELCOR Modernization Plan (5 years)

Modernization
- Improvements in code structure designed to strengthen
  - Performance – shorter run times and more robust calculation
  - Numerics – reduced numerical variance, improved accuracy, and improved characterization of uncertainty
  - Capability – Removal of limitations that hinder modeling of some effects
  - Maintainability – more extensible framework to allow code to adapt to new reactor designs., flexibility in future code/model development
- Maintain currently validated physics
  - Objective is not to add new physics models
  - This is not a re-write of the code (still FORTRAN though incorporating new FORTRAN features and possible other languages in new modules/routines)
- Incremental benefits in the existing code base

Five Year Plan
- Generalized Numerical Solution Engine Module
- MELCOR hydrodynamic modeling
- MELCOR In-Vessel Core Damage Progression
- Ex-Vessel Damage Progression
- Fission Product Release and Transport Modeling
- Rationalization of Current MELCOR Code Base with Modernized MELCOR Code Base / Finalization of the Modernized Code

MELCOR Code Modernization

COR Package Modernization

Improved Data Structure – array flattening

Develop Physics Manager – API to COR package models

Introduce Object Oriented program to facilitate implementation of new COR components and reactor types

Modernize and improve code testing

Update code to latest Fortran to access new Fortran features

Modernize physics models for lower head and candling
  Melting lower head

Modernization of MELCOR Development Environment

Objectives:
- Expand/enhance testing
- More efficient development – release cycle

Strategies
- Automation of work processes
- Utilize current industry standard software (avoid in-house tools)

Tools
- Git
- GitLab
- CMAKE
- Automation of assessment work

The overall intent of these efforts has been to migrate the MELCOR software development operations to an environment that allows the development process to be more readily update to current industry state-of-the-art in software development tools, from compilers to development and testing environment. The effort has automated a significant range of testing and code release processes that is enabling more extensive code testing and a more frequent code release cycle to ensure that new features and bug fixes are made available to users in a more timely manner.

Modernization of MELCOR Development Environment

MELCOR Code Modernization

1. **Moved from subversion to git for our revision control system,**

   Git replaces Subversion (SVN) as our version control system. The main distinction between Git and SVN is that Git uses a distributed repository while SVN uses a centralized repository. A distributed repository allows developers to check out a complete copy of the repository on a local system which then becomes independent in that it does not require online access for a complete history of code changes. While this can be advantageous, this was not the dominant reason for changing to Git. Git is more easily integrated with CMake and auto-testing and we foresee long-term support of git as a solution with greater support at Sandia National Laboratories.

2. **Moved to CMake for our build system,**

   CMake is used to build MELGEN/MELCOR executables in a consistent manner across multiple platforms using simple platform and compiler independent configuration files that are used to generate native makefiles and workspaces. CMake replaces in-house developed build scripts which are difficult to automate and maintain across multiple platforms.

3. **Updated to Intel OneAPI compilers from Intel 11.1 as the default compiler,**

   The latest version of the Intel compiler, v2021.6.0, fully supports the Fortran 2008 and Fortran 2018 standards. This enables features such as enumeration, submodules (faster more efficient compilation), polymorphic objects and classes, and other modern programming features. Before the changeover to the new compiler could take place it was necessary to update some existing code in order to obtain a version that compiled on all platforms, running the test cases in the repository with minimal impact on key test results.

4. **Moved our entire development operations to GitLab.**

Plan	Create	Verify	Secure	Package	Deploy	Monitor	Govern
Pages	GitLab CLI	Review Apps	Container Scanning	Dependency Proxy	Infrastructure as Code	Service Desk	Software Bill of Materials
Wiki	Web IDE	Merge Trains	Dependency Scanning	Helm Chart Registry	Auto DevOps	Incident Management	Dependency Management
Value Stream Management	Code Review Workflow	Code Testing and Coverage	Fuzz Testing	Container Registry	Environment Management	Metrics	Vulnerability Management
Design Management	Source Code Management	Continuous Integration (CI)	API Security	Package Registry	Release Orchestration	On-call Schedule Management	Compliance Management
Portfolio Management	Remote Development	Secrets Management	DAST	Dependency Firewall	Feature Flags	Error Tracking	Audit Events
Team Planning			Secret Detection		Continuous Delivery	Logging	Security Policy Management
Planning Analytics			SAST		Release Evidence	Tracing	
Requirements Management			Code Quality		Deployment Management		
Quality Management			License Compliance				
DORA Metrics							
DevOps Reports							

Hey there! Can I tell you about Gitlab for the Public Sector?

MELCOR Modernization - Git Branches

MELCOR Code Modernization

Developers work independently in branches
◦ Share development work back with other developers and the official release version.
◦ Allows developmental work to progress without interfering with other development tasks.
◦ Developers may work on multiple developmental branches or bugs simultaneously.

There is a hierarchy in branches from the official release version to intermediate merged versions, to local developmental versions.

Testing is performed automatically on individual branches and is required when merging from local branches back to the shared branch and ultimately to the official release branch

Location

Private
This branch exists only in the developer's environment and is deleted when it is merged back.

Feature Branch

Private/Public?
Can be cloned directly to a developer's local environment

Stage

Private/Public?
Can be cloned directly to a developer's local environment

Development

Public
Never cloned directly to a developer's local environment

Release/Main

Tagged Official Releases

Uploading Changes

- Merge Request initiated by developer?
- Short running tests only
- Code review not required
- No approval required

OR

hotfix

- Merge Request initiated by?
- Short & long running tests
- Code Review required
- No approval required

- Merge Request initiated by?
- Short & Long running tests
- Code Review
- Approval required

Usage

Use Case:
This is where development work resides. Changes here are only visible to developer until they are merged back into either Stage or Development

Use Case:
This is branch is where users' changes are merged back into the workflow. This branch is guaranteed to pass the short running tests but may contain code that is not suitable for general use. This branch is typically the base for developing new models.

Use Case:
This is where fully tested, fully reviewed code resides. This branch is always ready for release. For an official release, updates to documentation is also required. This branch is typically the base for debugging user issues.

Use Case:
This is where fully tested, fully reviewed code resides. This branch is always ready for release. However, For an official release, updates to documentation is also required.

Use Case:
This is not so much a branch as it is a tagged set of versions that reside in the Release Branch

MELCOR Modernization -Code Testing on Multi-Platforms

MELCOR Code Modernization

Multi-platform Testing
◦ Enabled by CMAKE

Introduction of CMAKE improves multi-platform releases
◦ Windows, Linux and macOS executables.
◦ Continuous testing of full test suite on targeted platforms.

Unit testing capabilities in addition to integral testing (new)
◦ Unit test harness, based on GoogleTest

Build	Test
✓ boca1_i21_3_debug	✓ boca1_debug_i21_3
✓ boca2_i21_3_release	✓ boca2_release_i21_3
✓ ceerws_i21_3_release	✓ ceerws_release_i21_3
✓ linux_i21_3_debug	✓ linux_debug_i21_3
✓ linux_i21_3_release	✓ linux_release_i21_3
✓ mac_i21_5_debug	✓ mac_debug_i21_5
✓ mac_i21_5_release	✓ mac_release_i21_5
✓ win_i21_5_debug	✓ win_debug_i21_5
✓ win_i21_5_release	✓ win_release_i21_5
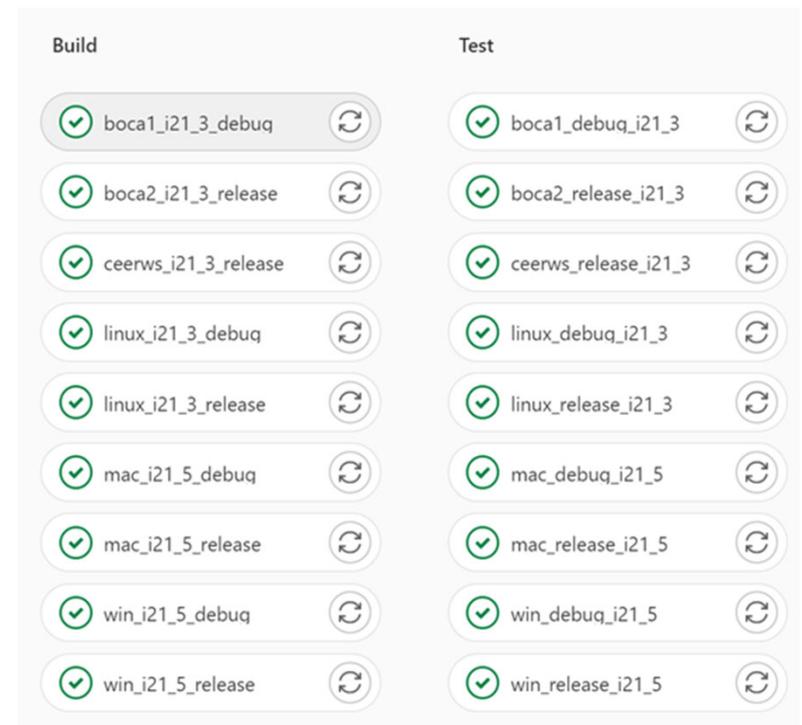
MELCOR Modernization - Array Flattening

Motivation:
- Simplifies Coding
- Easier to add new components
- Easier to add properties to all components
- [Facilitate debugging](#)
- Performance improvements due to localization
- Improve potential for achieving identical results from restarts
- Finding/reducing cliff-edge effects

As an example, consider the XMDC array as currently written:

Cor%Cell(naxl, nrad)%Cellcomp(kcmp)%New%Xmdc(ncrmat)

Our flattened stride one XMDC array becomes a rank five array dimensioned as follows:

Xmdc(ncrmat, kcmp, naxl, nrad, state)

The memory layout for XMDC is now contiguous in memory and stride one.

MELCOR Code Modernization

MELCOR Modernization - Physics Manager

Architectural change that manages the execution of physics algorithms

Flexible means by which developers can introduce new/alternate physics models

Enables the ability of external users to develop alternate physics models without access to source code.

An important benefit of the physics manager is that it enables advanced users of the code the option to introduce new algorithms without the need to access or change the existing code base. Potential applications would be simulator vendors, analysts seeking to utilize the code in highly specialized environments, or researchers developing models. This provides a flexible Application Programming Interface (API) that would not require distribution of source code.

Questions

MELCOR Code Development

- New Modeling
  - Non-LWR Models
    - HTGR
    - Heat Pipe Reactors
    - Molten Salt Reactors
  - LWRs
    - Melting Lower Head
    - New Defaults (MCCI, eutectics, etc.)
- SQA
  - Documentation
  - Validation
- Utilities
  - SNAP
    - Upgrade for new models
  - HTML
- Modernization