OpenFOAM을 이용한 격납건물 내 다상유동 연구

(Efforts to Understand Multiphase Flow Phenomena in an NPP Containment Using OpenFOAM)

김종태(KAERI), 정재훈(KAERI) , 김대희 (KAERI) , 최선락 (KAERI), 조용진 (KINS), 김균홍(OpenCAE)

2022. 9.22-23

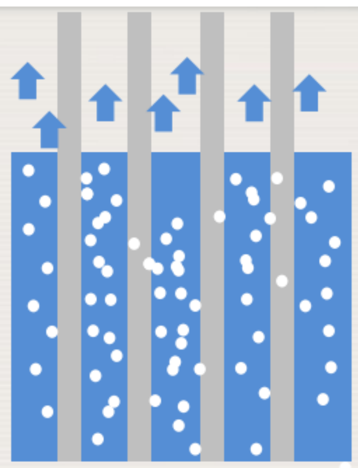
Contents

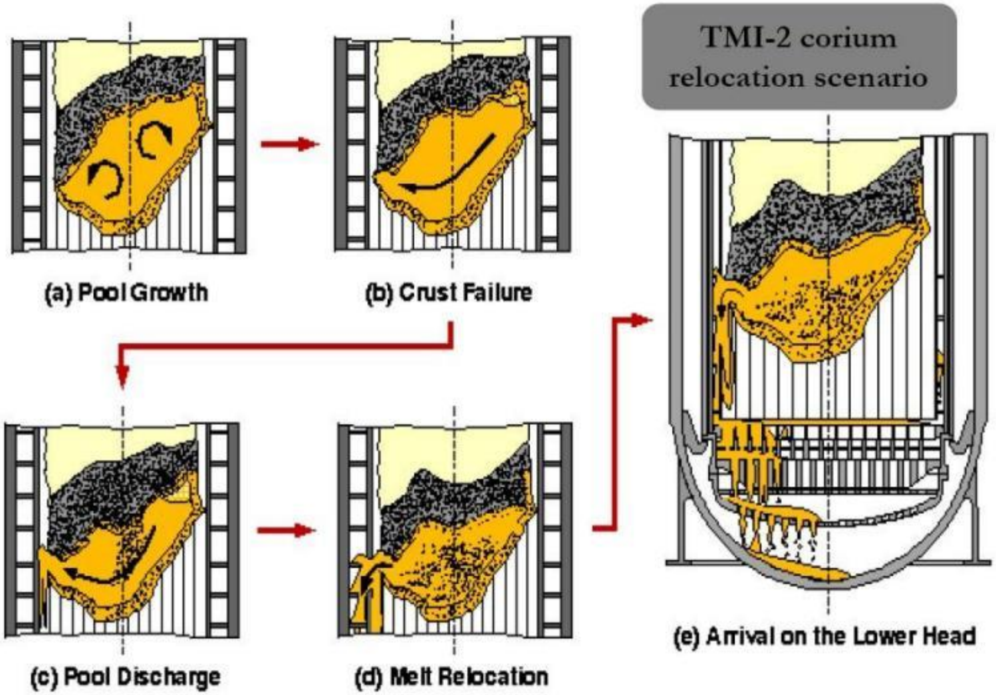
Examples of multiphase Flows

- **General multiphase flows**
  - boiling water flow: two-phase flow
  - air-water flow: two-fluid two-phase flow
  - water-oil flow: immiscible two-liquid flow
  - gas-aerosol/gas-droplet flow: two-fluid two-phase flow
  - fluidised bed: multicomponent two-phase flow
- **In-containment multiphase flow**
  - water spray in a containment
  - flashing water during a LOCA
  - condensate film flow on a containment wall
  - steam-fog/fission-aerosol in a containment atmosphere
  - molten corium jet-fragmentation, spreading
  - molten corium-concrete interaction
  - debris cooling in a cavity

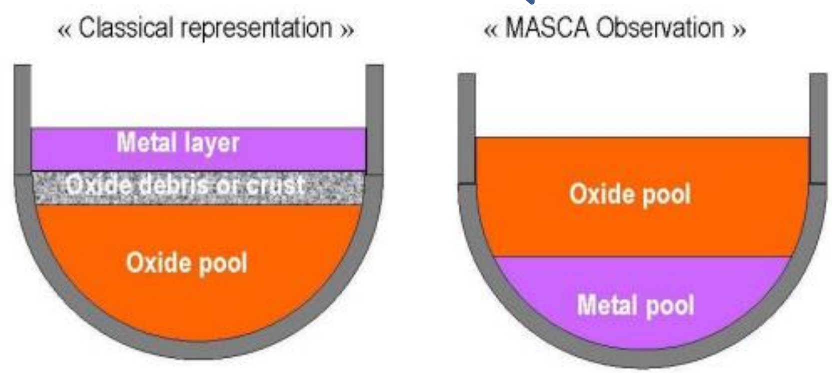
Examples of multiphase Flows In-vessel

water two-phase flow



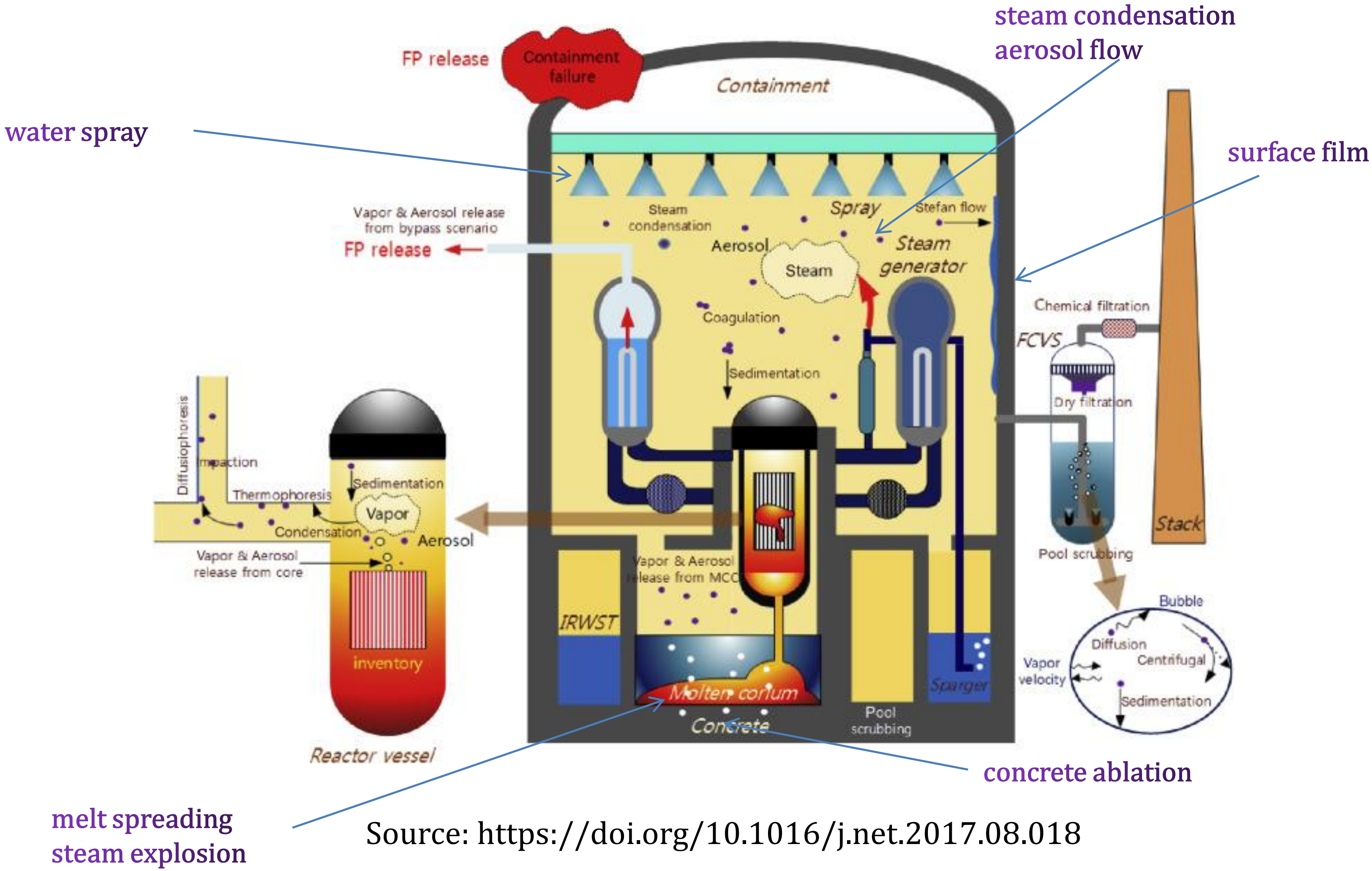


melting-solidification



Source: https://inis.iaea.org/collection/NCLCollectionStore/_Public/49/066/49066570.pdf

Examples of multiphase Flows in-containment



Source: https://doi.org/10.1016/j.net.2017.08.018

Multiphase Flows (physical point of view)

- **Multiphase flow**
  - Flows of ***immiscible*** fluids → fluid interface
    - different phases: gas phase, liquid phase and solid phase
    - immiscible fluids: such as water-oil flow
  - miscible fluids
    - multispecies mixture
  - Types of flows
    - Single -phase two-fluid: water-oil flow
    - Two-phase single-fluid: steam-water flow
    - Two-phase two-fluid: air-water flow
  - what happens on an interface
    - pressure jump by surface tension
    - mass, energy and momentum transfer
  - Interface scale
    - large scale interface: separated/segregated flow
    - small scale interface: two-phase mixture flow (homogeneous/inhomogeneous mixture)

Multiphase Flows (Numerical point of view)

- Multiphase flow
  - **interface of phases**
    - macroscopic interface (larger than cell-size)
      - interface capturing/tracking/fitting/representation: VOF, level-set, SLIC, PLIC
        - air-water flow with macroscopic interfaces: free surface flow
    - microscopic interface (smaller than cell-size)
      - partly occupying each mesh cells: Euler-Euler approach
        - air-water flow with microscopic interfaces: bubbly flow
      - comparatively small volumes occupied by a dispersed phase: Euler-Lagrange approach
        - air-water flow: droplet flow, aerosol flow
      - large-volume particle flow: Euler-DEM approach
  - **mixed with macro- and micro- interfaces**
    - VOF-Lagrange: VOF for macro, Lagrange for micro interface
    - VOF-Euler: VOF for macro, Euler for micro interface
    - examples of flows: liquid jet breakup, slug flow,

Numerical approach for multiphase flows

- Eulerian approach
  - VOF
    - 1-set of conservation equation (M-M-E)
    - application: large-scale interface capturing
    - cells on the interface have properties of two-fluid mixture
  - Homogeneous equilibrium mixture
    - 1-set of conservation equation (M-M-E)
    - application: micro-scale interface modeling (size-model with interfacial transport)
  - Inhomogeneous mixture
    - so-called two-fluid or 6-eqs. model
    - time-volume-ensemble averaging of the governing equations

Structure of OpenFOAM for Multiphase Simulation

- Code structure: applications + libraries
- Applications
  - solver and type
    - Homogeneous-mixture, VOF, Euler+Euler, Euler+Lagrage
    - hybrid (VOF+Euler, VOF+Lagrange)
  - Libraries
    - transport properties for multiphase mixture
      - mixture properties by mixing phasic properties
    - phaseSystems
      - basic type: twoPhaseSystem, multiphaseSystem
      - purpose: managing phases and modeling interfacial transports
    - phaseModels
      - phase itself
      - calculating phasic properties and transport
    - phasePair
    - interfacial models
    - size model

Multiphase solvers in Of v2006

- Solver: 17 solvers
  - VOF solver: 10 solvers
    - twoPhaseInter: interFoam, interIsoFoam,
      - cavitatingFoam, interPhaseChangeFoam, interCondensatingEvaporatingFoam
      - compressibleInterFoam, MPPICInterFoam
    - multiPhaseInter: multiphaseInterFoam, compressibleMultiphaseInterFoam.
      - icoReactingMultiphaseInterFoam
  - Euler solver: 5 solvers
    - TwoPhaseEuler: twoPhaseEulerFoam, reactingTwoPhaseEulerFoam,
    - chtMultiRegionTwoPhaseEulerFoam: CHT + reactingTwoPhaseEulerFoam
    - MultiPhaseEuler: multiphaseEulerFoam, reactingMultiphaseEulerFoam
  - Mixture: 2 solvers
    - driftFluxFoam, twoLiquidMixingFoam

OpenFOAM-8 VS OpenFOAM-v2012

- **OF-8**
  - twoPhase is merged to multiphase
  - reacting is generalized as a multiphase phase model.

OF-v2012	OF-8
twoPhaseEulerFoam	multiphaseEulerFoam
multihaseEulerFoam	
reactingTwoPhaseEulerFoam	
reactingMultiphaseEulerFoam	

3 basic classes for Eulerian multiphase modeling

- **Purpose**
  - Generalized multi-phase Eulerian modeling
  - Legacy approach: fixed number of phases
    - 2 phase: dispersed-gas and continuous liquid
    - 4 phase: dispersed/continuous-gas and dispersed/continuous liquid
- **Key classes for Eulerian multiphase modeling**
  - phaseSystem : phaseModel : phasePair
- **phaseSystem**
  - container class for phases. It construct phaseModels
  - It calculates interfacial transfer of mass, energy and momentum
- **phaseModel**
  - phaseModel is a phase
  - base is alpha (volume fraction of volScalarField)
- **phasePair**
  - manage interfacial transport modeling between 2 phases

Phase systems for Euler solvers

- Base class: **phaseSystem** (inheriting IOdictionary)
  - container of phases (**phaseModelList**)
    - **has subList of phases (movingPhaseModels_, stationaryPhaseModels_, anisothermalPhaseModels_, multi ComponentPhaseModels_)**
  - has phasePairs (**phasePairTable**),
  - includes virtual functions to calculate interfacial transfer sources
    - **momentumTransfer(), momentumTransferf(), phiFs(), phiFfs(), heatTransfer(), massTransfer()**
- Construction of phaseSystem
  - construct phase list, sub phase list,
  - generatePairsAndSubModels() add phasePair and create transfer model list
    - **surfaceTension, aspectRatio (from phaseSystem)**
    - **drag, virtualMass, lift, wallLubrication, turbulentDispersion (from MomentumTransferPhaseSystem)**
    - **heatTransfer (from TwoResistanceHeatTransferPS (pair) or oneResistanceHeatTransferPS)**
    - **phaseTransfer (from PhaseTransferPhaseSystem)**
    - **massTransfer (from InterfaceCompositionPhaseChangePhaseSystem) (Pair)**
    - **hash table (phasePairKey, modelType)**

Building blocks to construct phaseSystems

- File structure
  - directory: multiphaseSystem/PhaseSystems
- template phaseSystem: building blocks for real phaseSystems

template phaseSsyeme	featrue
InterfaceCompositionPhaseChangePhaseSystem	massTransferTable
MomentumTransferPhaseSystem	momentumTransferTable for all moving phases
OneResistanceHeatTransferPhaseSystem	heatTransferTable for all phases
TwoResistanceHeatTransferPhaseSystem	heatTransferTable for all phases
PopulationBalancePhaseSystem	massTransferTable for all species
ThermalPhaseChangePhaseSystem	massTransferTable for all species
PhaseTransferPhaseSystem	massTransferTable for all species phaseTransferModelTable

Template phaseSystems (1)

☐ MomentumTransferPhaseSystem

- Class which models interfacial **momenum transfer** between a number of phases. Drag, virt ual mass, lift, wall lubrication and turbulent dispersion are all modelled.

☐ OneResistanceHeatTransferPhaseSystem

- Class which models interfacial heat transfer between a number of phases. A **single heat tra nsfer model** is used for each interface

☐ TwoResistanceHeatTransferPhaseSystem

- Class which models **interfacial heat transfer** between a number of phases. **Two heat trans fer model**s are stored at each interface, one for each phase. This permits definition of an int erface temperature with which heat transfer occurs

☐ phaseTransferPhaseSystem

- Class which models **non-thermally-coupled mass transfers**; i.e., representation changes, r ather than phase changes

Template phaseSystems (2)

☐ ThermalPhaseChangePhaseSystem

- Class to provide **interfacial heat and mass transfer** between a number of phases according the interfacial temperature approximated by the saturation temperature

☐ PopulationBalancePhaseSystem

- Class which provides **population balance functionality**

☐ InterfaceCompositionPhaseChangePhaseSystem

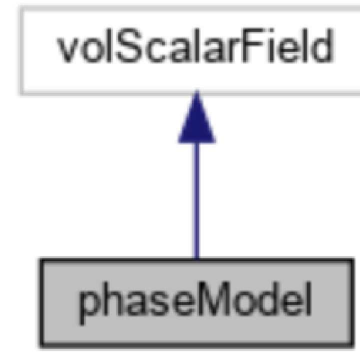
- Class to provide interfacial heat and mass transfer between a number of phases according to a interface composition model

Phase systems for multiphase

- multiphase sytems for object creation
  - **basicMultiphaseSystem**
    - *PhaseTransferPS<OneResistanceHeatTransferPS<MomentumTransferPS<multiPS>>>*
  - **interfaceCompositionPhaseChangeMultiphaseSystem**
    - **InterfaceCompositionPhaseChangePS***<PhaseTransferPS<TwoResistanceHeatTransferPS<MomentumTransferPS<multiPS>>>>*
  - **thermalPhaseChangeMultiphaseSystem**
    - **ThermalPhaseChangePS***<PhaseTransferPS<TwoResistanceHeatTransferPS<MomentumTransferPS<multiPS>>>>*
  - **populationBalanceMultiphaseSystem**
    - **PopulationBalancePS***<PhaseTransferPS<OneResistanceHeatTransferPS<MomentumTransferPS/multiPS>>>>*
  - **thermalPhaseChangePopulationBalanceMultiphaseSystem**
    - **ThermalPhaseChangePS<PopulationBalancePS***<PhaseTransferPS<TwoResistanceHeatTransferPS<MomentumTransferPS<multiPS>>>>>*

Phase Model

- phaseModel
  - **base class of phases**
  - **phase object created by phaseSystem's constructor**
  - **phaseModels_(lookup("phases"), phaseModel::iNew(*this))**
- List of the derived **template** classes (*unusable, capitalized*)
  - **heat transfer: IsothermalPhaseModel & AnisothermalPhaseModel**
  - **species: PurePhaseModel & MultiComponentPhaseModel**
  - **reaction: InertPhaseModel & ReactingPhaseModel**
  - **flow: StationaryPhaseModel & MovingPhaseModel**
  - **ThermoPhaseModel**



Phase Model (2)

- *ThermoPhaseModel*
  - **provides access to the thermodynamic variables**
- *IsothermalPhaseModel*
  - **Class which represents temperature (strictly energy) remains constant**
- ***AnisothermalPhaseModel***
  - **construct phasic energy equation without interfacial transprot**
- *StationaryPhaseModel*
  - **Class which represents a stationary (and therefore probably solid) phase**
  - **momentum related member functions return zero fields**
- ***MovingPhaseModel***
  - **Class which represents a moving fluid phase. Holds the velocity, fluxes and turbulence model and can generate the momentum equation.**
  - **return the phasic momentum equation without interfacial transport**

Phase Model (3)

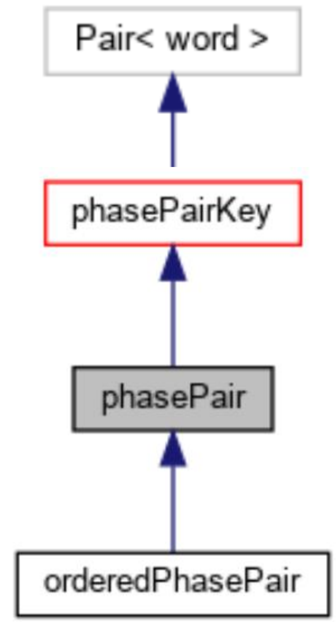
- *PurePhaseModel*
  - **Class which represents pure phases, i.e. without any species. Returns an empty list of mass f ractions.**
- ***MultiComponentPhaseModel***
  - **Class which represents a phase with multiple species. Returns the species' mass fractions, an d their governing equations**
- *InertPhaseModel*
  - **Class which represents an inert phase, with no reactions. Returns zero reaction rate and heat**
- ***ReactingPhaseModel***
  - **Class which represents phases with volumetric reactions. Returns the reaction rate and heat**

Phase Model (4)

- phaseModel type for object creation (phaseModels.C creats these real classes)
  - **purePhaseModel:** To consider momentum-energy transport
    - **AnisothermalPM**<PurePM<InertPM<**MovingPM**<ThermoPM>>>>
  - **pureStationaryPhaseModel:** To consider energy transport
    - **AnisothermalPM**<PurePM<InertPM<**StationaryPM**<ThermoPM>>>>
  - **pureIsothermalPhaseMode:** To consider momentum transport
    - **IsothermalPM**<PurePM<InertPM<**MovingPM**<ThermoPM>>>>
  - **pureStationaryIsothermalPhaseModel:** Nothing to consider
    - **IsothermalPM**<PurePM<InertPM<**StationaryPM**<ThermoPM>>>>
  - **multiComponentPhaseModel:** To consider species-momentum-energy transport
    - **AnisothermalPM**<**MultiComponentPM**<InertPM<**MovingPM**<ThermoPM>>>>
  - **multiComponentIsothermalPhaseModel:** To consider species-momentum transport
    - IsothermalPM<MultiComponentPM<InertPM<MovingPM<ThermoPM>>>>
  - **reactingPhaseModel:** To consider reactive-species-momentum-energy transport
    - AnisothermalPM<ReactingPM<MovingPM<CombustionModel>>>>

phasePair

- What is?: manage interfacial transport modeling between 2 phases
  - **structure**
    - **phasePair<phasePairKey<Pair<**
      - Pair is inheriting fixedList<T, 2>
    - Pair<word>
      - members: first(), last(), second(), other(), flip(),
    - **phasePairKey (name only)**
      - Pair + ordered_(default is false) + hash()
      - if "and" ordered_ = false
      - for phasePairKey it has two names of phases
    - **phasePair (including 2 phaseModels)**
      - contains 2 references of phaseModels
      - phase1(), phase2()
      - name() returns *first() + "And" + name2*
      - otherName() returns *Second() + "And" + name1*
    - **orderedPhasePair**
      - **phase1 is fromPhase, phase2 is toPhase**
      - name() returns *first() + "In" + name2*
      - otherName() returns *fatal error*
      - *dispersed() == phase1(), continuous() == phase2()*



Blending of interfacial transport

☐ Phase inversion

- gas-water two-phase flow: phase inversion from bubbly flow to droplet flow
  - bubbly flow: dispersed gas in continuous water
  - droplet flow: dispersed doprlet in continuous gas
- Method 1: 4-field model (nphase code by Podoski-Antal-Kunz)
  - dispersed/continuous gas fields and dispersed/continuous liquid fields
- Method 2: blending (OpenFOAM)
  - blending of dispersed-gas/continuous-liquid flow and dispersed-liquid/continuous-gas flow
  - $K = (1 - f_1 - f_2)K_{1,2} + f_1 K_{12} + f_2 K_{21}$

☐ OpenFOAM's blending method

- blending method: noBlending, hyperbolic, linear
- FullyContinuous = continuous phase, PartlyContinuous = dispersed phase

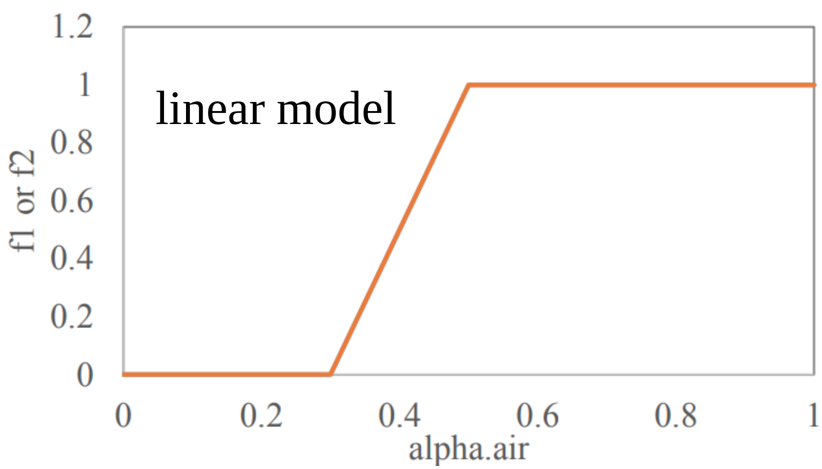
```
default
{
    type
    linear ;
    minFullyContinuousAlpha.gas 0.7;
    minPartlyContinuousAlpha.gas 0.5;
    minFullyContinuousAlpha.liquid 0.7;
    minPartlyContinuousAlpha.liquid 0.5;
}
```

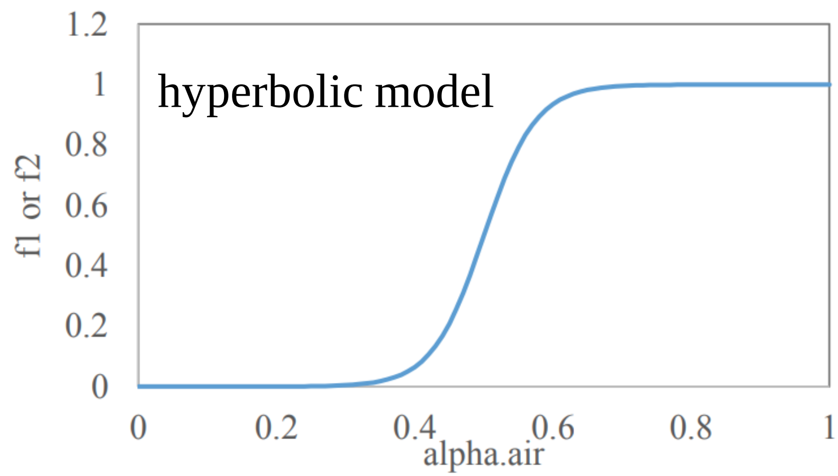
Blending of interfacial transport (2)

☐ BlendedInterfacialModel

- MinFullyContinuousAlpha (α_{full}) : Minimum fraction of phases which can be considered fully continuous
- MinPartlyContinuousAlpha (α_{part}): Minimum fraction of phases which can be considered partly continuous

Blending method	$0 < f_1$ and $f_2 < 1$
hyperbolic	$f_1 = 1/2[1+\tanh(4/\alpha_{trans}(\alpha_2 - \alpha_{min,2}))]$, $f_2 = 1/2[1+\tanh(4/\alpha_{trans}(\alpha_1 - \alpha_{min,1}))]$
linear	$f_1 = (\alpha_2 - \alpha_{part,2})/(\alpha_{full,2} - \alpha_{part,2})$, $f_2 = (\alpha_1 - \alpha_{part,1})/(\alpha_{full,1} - \alpha_{part,1})$
none	f_1 = (phase2.name() == continuousPhase), f_2 = (phase1.name() == continuousPhase)



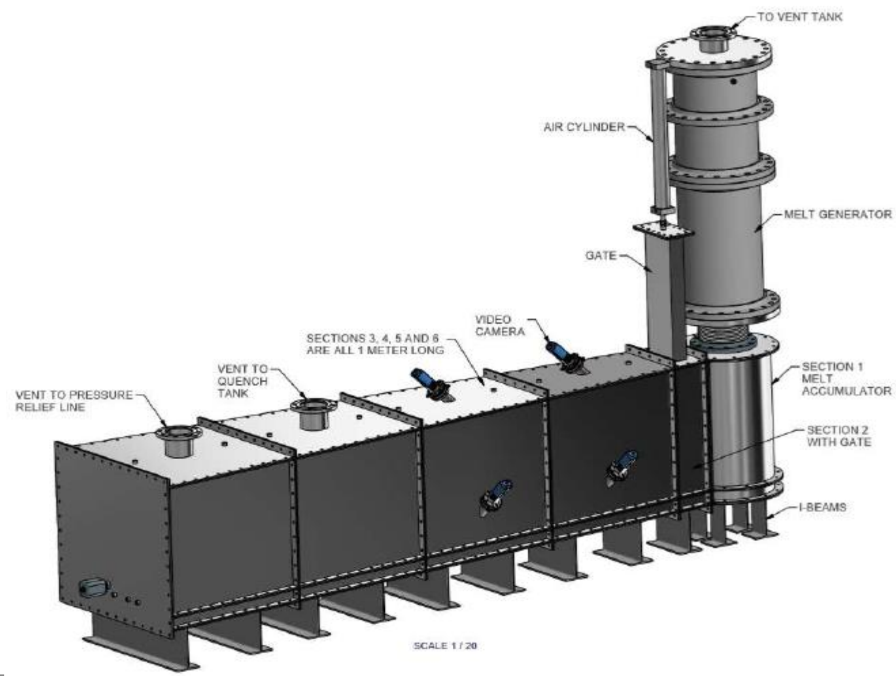


Multiphase flows in NPP Containment
And
Simulation by OponFOAM

Melt spreading in a reactor cavity

❑ Large-scale Underwater Melt Phenomena Spreading Experiment (LUMPS)

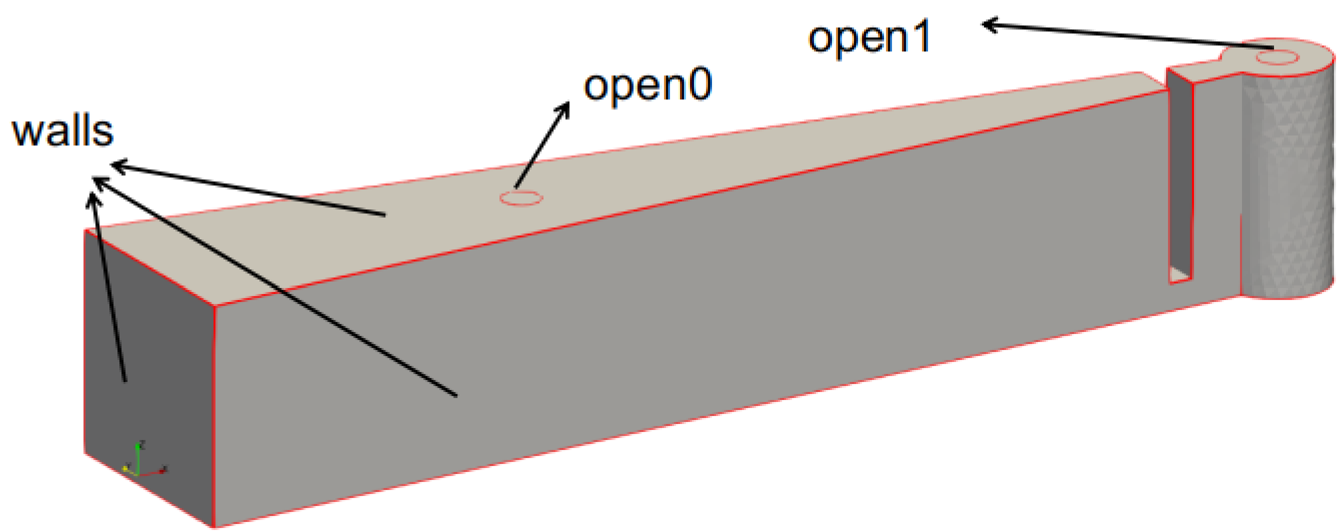
- ROSAU국제공동연구에서는 LUMP (large-scale underwater melt spreading experiment) 실험장치를 이용하여 용융물이 냉각수 내에서 퍼짐 특성을 실험하는 연구
- 구성
  - 용융물 생성장치(melt generator)
  - 용융물 저장장치(melt accumulator)
  - 방출 게이트(gate)
  - 퍼짐 채널(spreading channel)
  - 퍼짐 채널은 4개의 섹션



Melt spreading in a reactor cavity

❑ LUMPS analysis

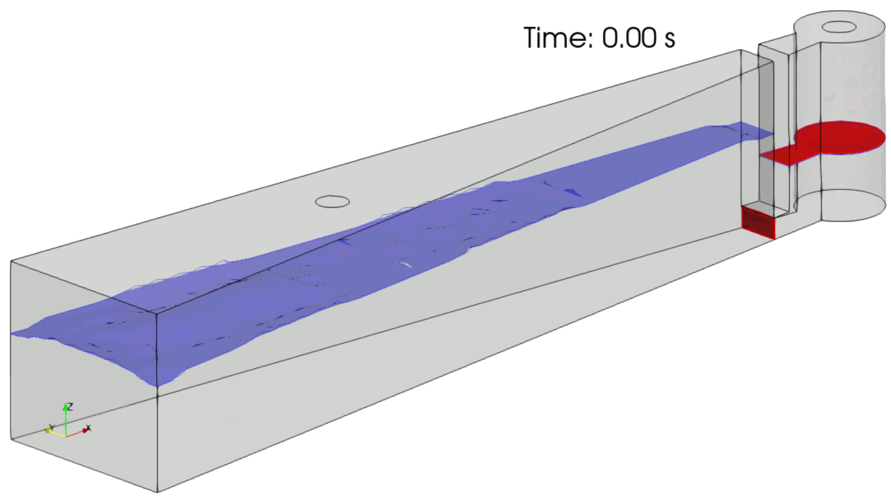
- **Pre-test analysis of LUMPS**
- **modeling for OpenFOAM analysis**
  - **solver: icoReactingMultiphaseInterFoam**
  - **surface tension considered**
  - **temperature-dependent melt viscosity**
  - **3 phases: gas, melt, water**



Pre-Test Analysis of LUMPS

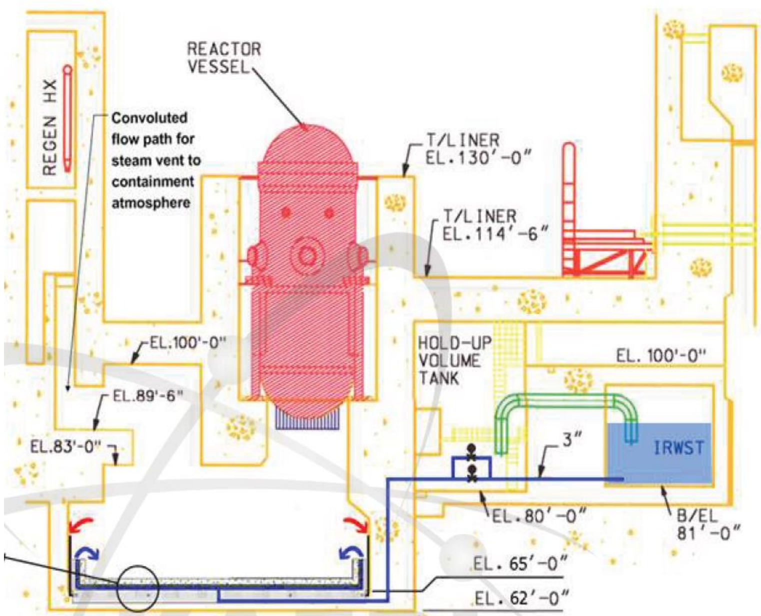
- OpenFOAM soler
  - icoReactingMultiphaseInterFoam
    - VOF for interface capturing with temperature-dependent viscosity model
    - air-water-melt three phase flow

```
type     massTransferMultiphaseSystem;
phases   (gas water melt);
gas
{
    type              pureMovingPhaseModel;
}
water
{
    type              pureMovingPhaseModel;
}
melt
{
    type              pureMovingPhaseModel;
}
surfaceTension
(
    (gas and water)
    {
        type              constant;
        sigma             0.074;
    }
    (gas and melt)
    {
        type              constant;
        sigma             0.62;
    }
    (water and melt)
    {
        type              constant;
        sigma             0.62;
    }
);
```

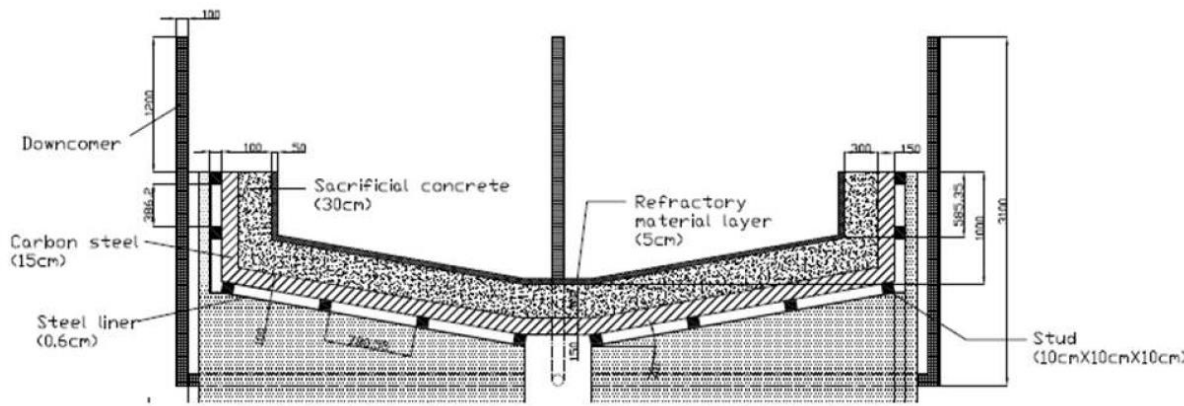


EU-APR1400 Core catcher

- PECS (Passive Ex-vessel Corium Retaining and Cooling System)
  - Corium cooling device to prevent MCCI
  - Cooling of corium spread on the dry corium catcher



EU-APR1400

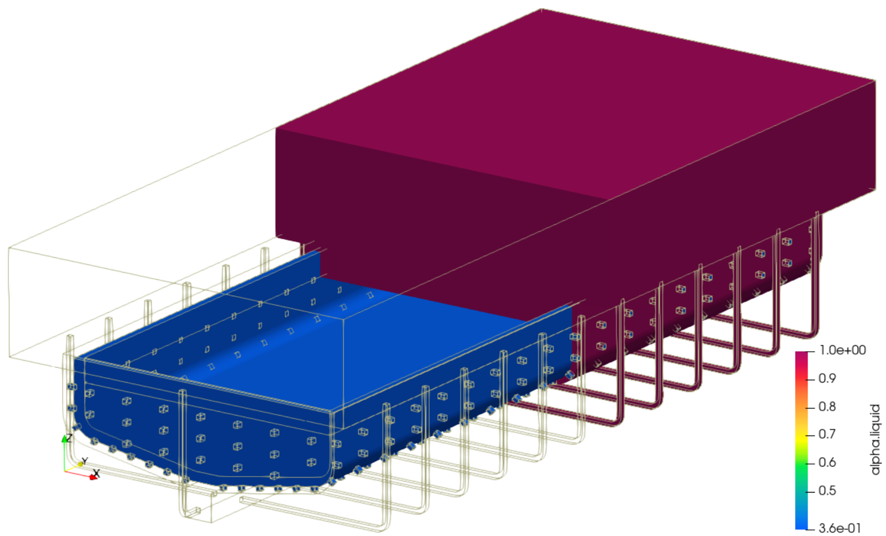


core catcher

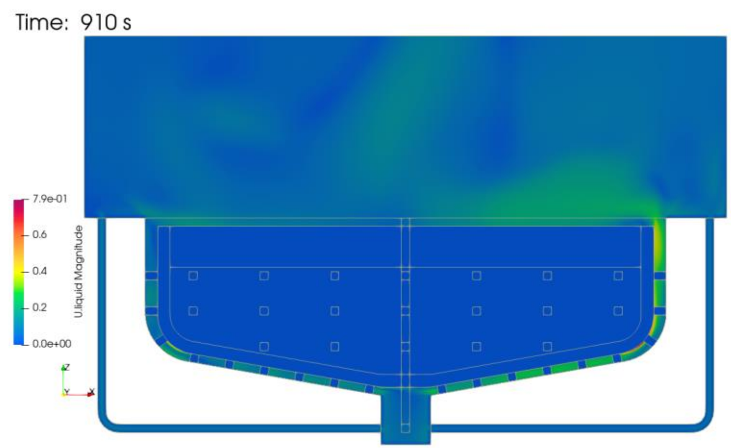
ref. KAERI/TR-5813/2014

EU-APR1400 Core catcher

- Simulation of natural circulative flow in the PECS flow system
  - PECS flow system: Upper pool ➔ downcomer ➔ channel ➔ upper pool
- OpenFOAM modeling
  - Solver: chtMultiRegionTwoPahseEulerFoam
    - solids (catcher, stud, corium crust) heat transfer + boiling two-phase flow
    - multi-component gas mixture (air + steam)



PECS geometry and pool



Liquid velocity

Steam Sparging to FCVS

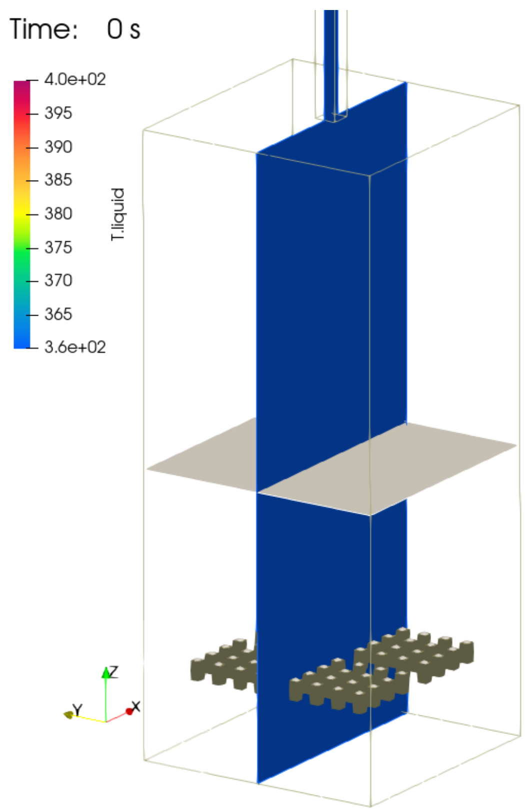
FCVS: Filtered containment vent system

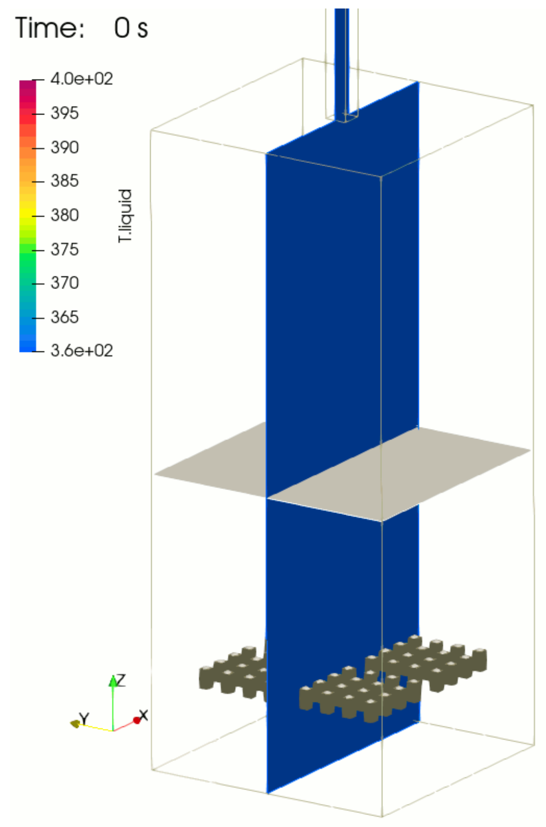
- Analysis of pool behavior condensing steam
- OpenFOAM modeling
  - solver: reactingTwoPhaseEulerFoam
    - multi-component gas/liquid two phase solver with phase change
  - Simplified FCVS modeling
    - sparger inlets: simple square channel
    - gas injection: steam and air injection with specified mass flow

Steam Sparging to FCVS

❑ FCVS: Filtered containment vent system

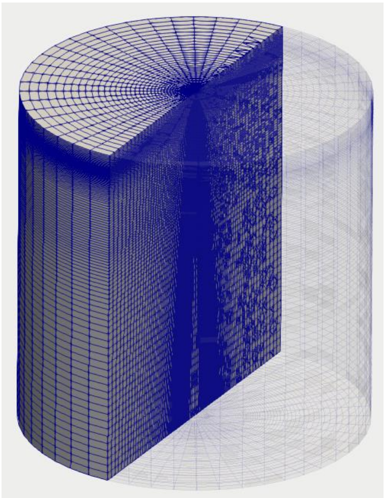
❍ preliminary results of pool behavior

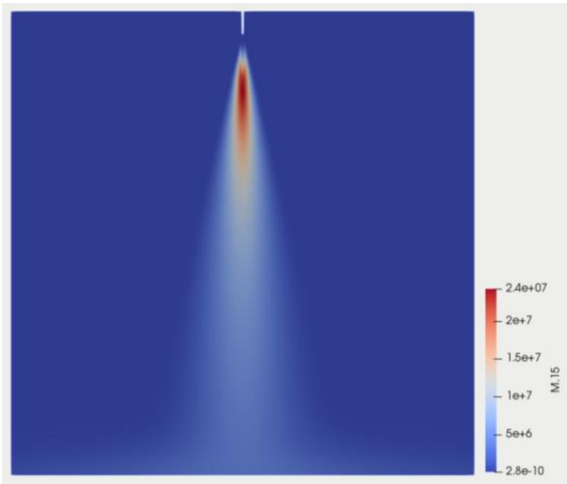




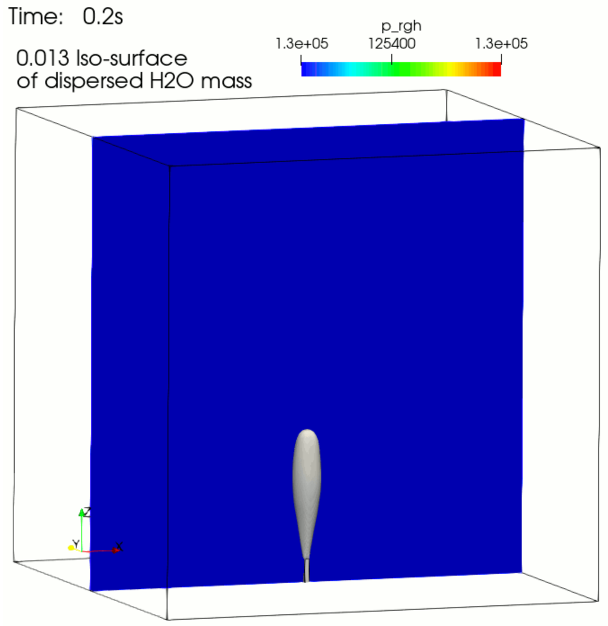
Steam Jet Condensation

- Steam jet in a containment during an accident
  - Bulk condensation includes nucleation from vapor and condensational growth from the nucleated seed or a droplet
  - Importance: hydrogen concentration governed by steam condensation
- Benchmark of steam jet experiment



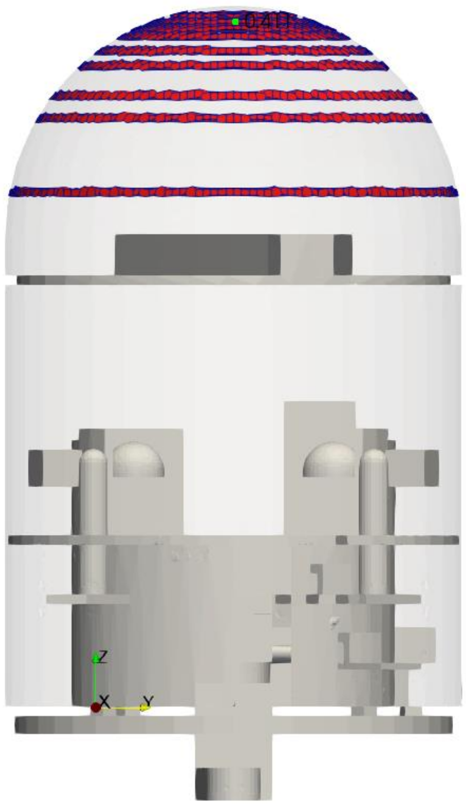


Number of particles (#/kg, 15^{th} section)



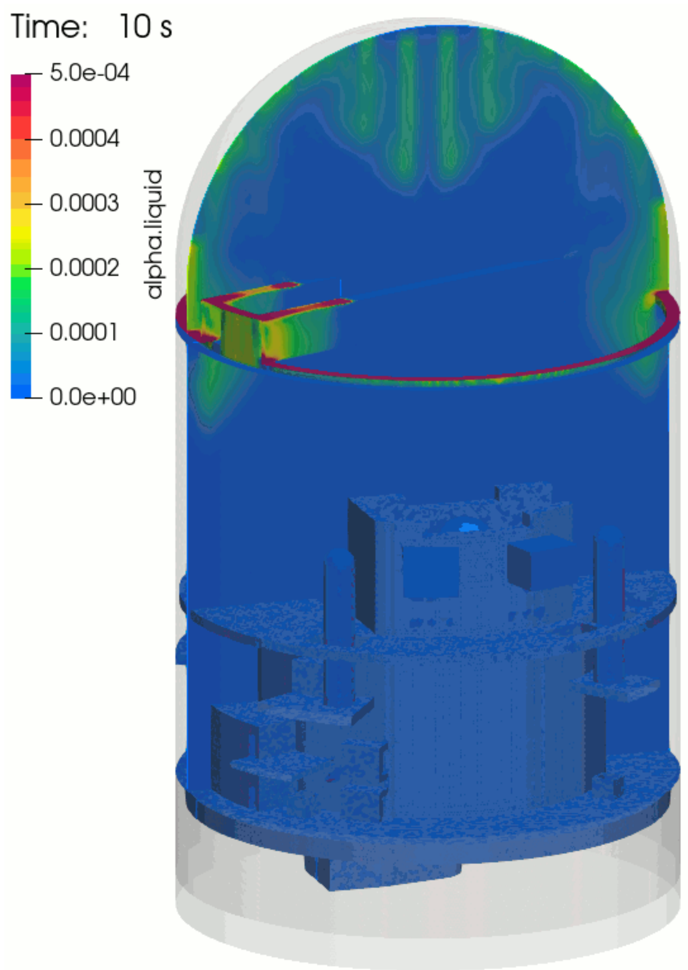
Spray and surface film flow in containment

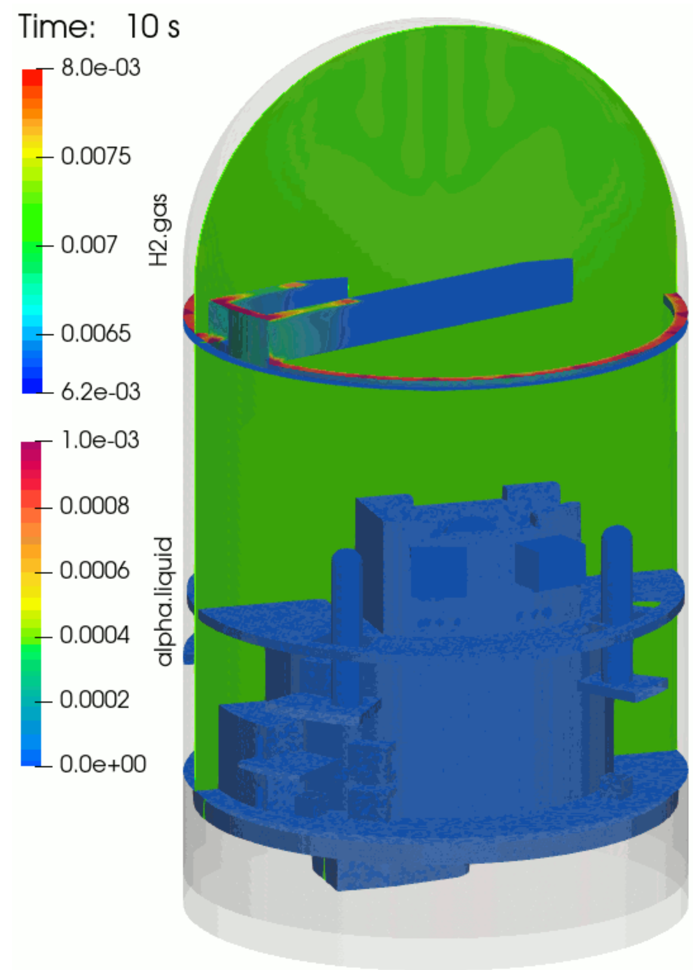
- Steam condensation and development of surface film
  - Importance: hydrogen concentration governed by steam condensation
    - steam condensation is dependent on surface film behaviors
- APR1400 spray analysis
  - OpenFOAM modeling
    - Solver: reactingTwoPhaseEulerFoam
    - multi-component gas mixture (air+steam+H2)
    - spray nozzle as mass-momentum-energy soruce
    - wall heat transfer/condensation
      - currently not modelled



Spray and surface film flow in containment

❑ Preliminary results of APR1400 spray analysis





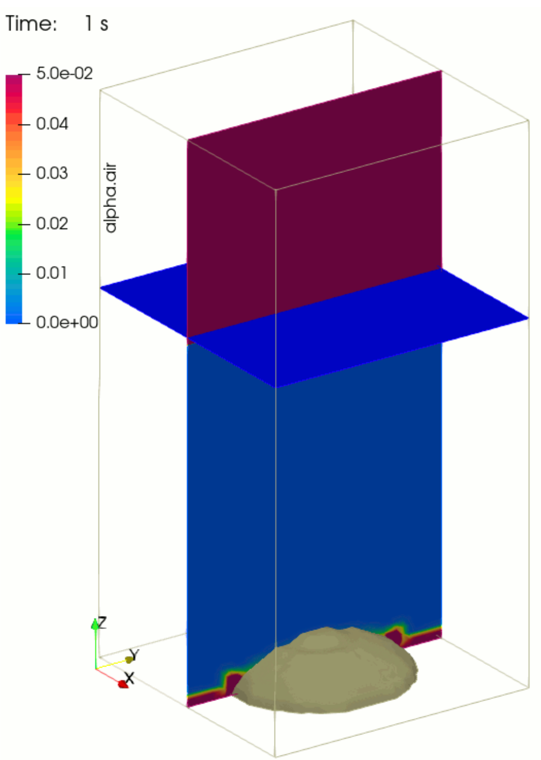
Debris Coolability

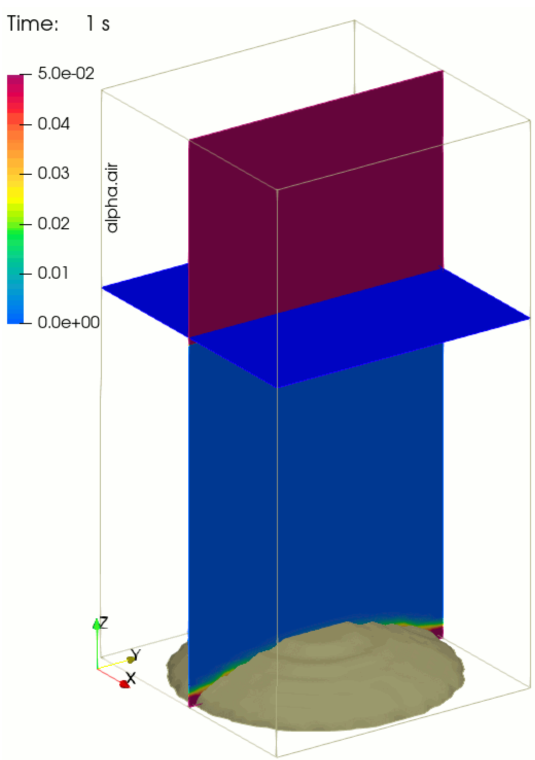
- Debris of core melt in the reactor cavity pool
  - Importance: Evaluation of debris coolability
    - thermal-hydraulic conditions of the cavity pool
      - Approximation of pool depth to stably cool the debris
    - Minimization of MCCI
- OpenFOAM modeling
  - Solver: multiphaseEulerFoam
    - debris as pureIsothermalStationaryPhaseModel
    - purePhaseModel for debris ➔ future work
- DEFCON (air test) 실험 해석

❑ DEFCON 공기 실험

❍ 단순 해석

- 코드 적용 가능성 검증
- 하부에서 공기 주입

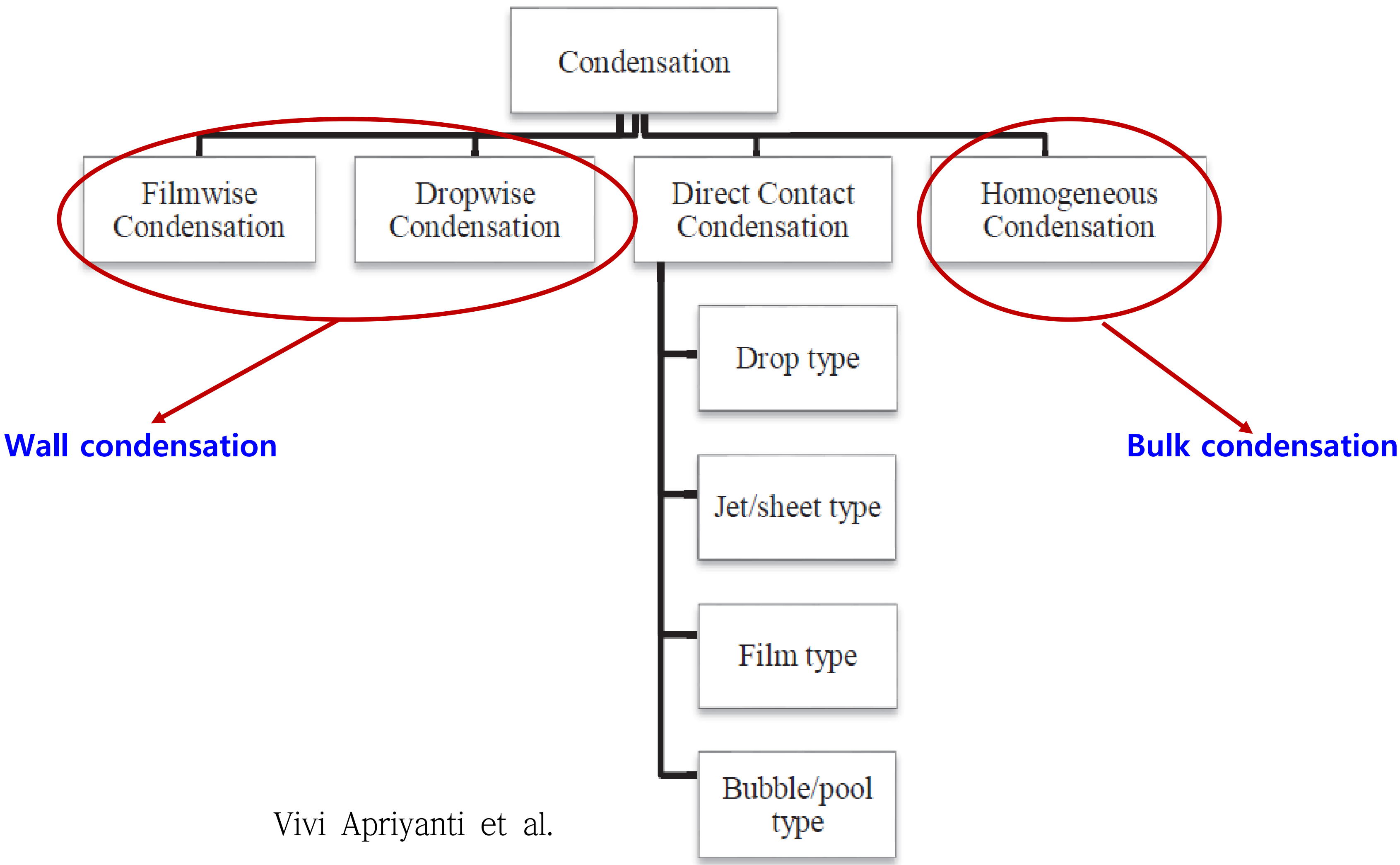




오픈폼을 이용한 격납건물 다상유동 해석

Development of a simple equilibrium model for wet-steam flow

❑ Condensation mechanism



Vivi Apriyanti et al.

Development of a simple equilibrium model for wet-steam flow (2)

❑ Assumptions

❍ Thermo-mechanical equilibrium

- Equal velocity and temperature for phases

Mixture–based approach (single–phase mixture model)	Source–term based approach (gas–liquid two–phase model)
no–source for mixture eqs. mass loss for vapor specie ➔ mass gain for fog	mass/momentum/energy loss in gas–phase
mass fraction of fog liquid $\alpha = \frac{m_f}{m}$	mass fraction of fog liquid $\alpha = \frac{m_f}{m_g}$
fog mass equation $\frac{\partial}{\partial t}(\rho\alpha) + \nabla(\rho\boldsymbol{U}\alpha) = S_\alpha$	fog mass equation $\frac{\partial}{\partial t}\left(\rho_g\alpha\right) + \nabla\left(\rho_g\boldsymbol{U}\alpha\right) - \nabla(\Gamma_t\nabla\alpha) = S_\alpha$ $\Gamma_t = \mu_t/Pr_t$
Using mixture properties	Using gas–phase and liquid–phase properties
fog volume fraction is negleced	fog volume fraction is negleced

Development of a simple equilibrium model for wet-steam flow (3)

❑ Ize modeling for fog aerosols

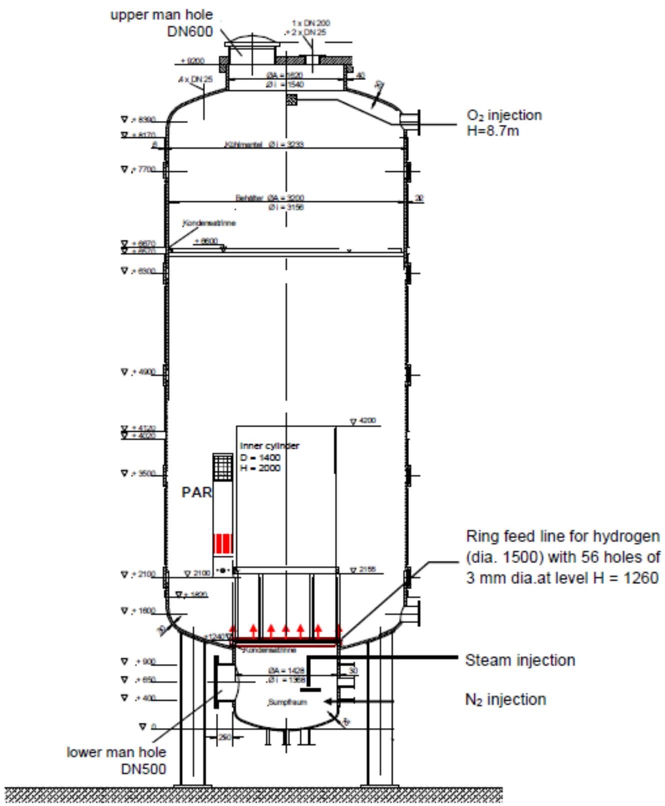
❍ No-model, mono-dispersed, poly-dispersed models

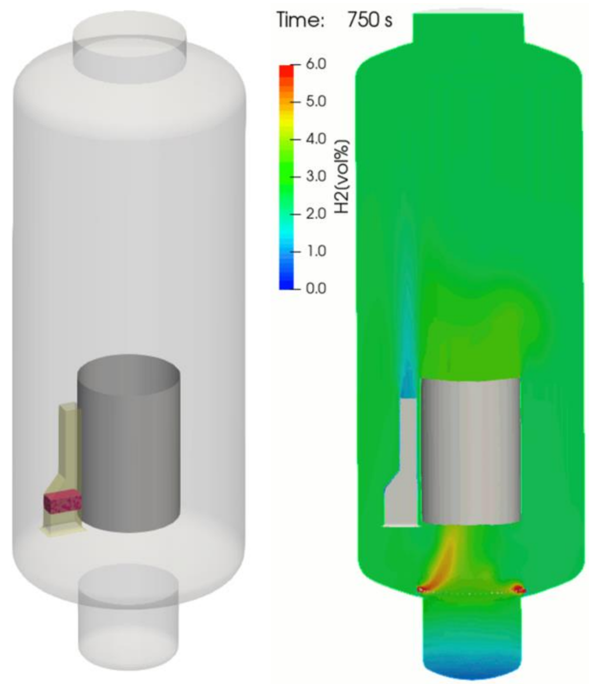
no–model approach	mono–dispersed approach
gas mass: $S_\rho = -S_\alpha,\ S_{h2o} = -S_\alpha$	gas mass: $S_\rho = -S_\alpha,\ S_{h2o} = -S_\alpha$
gas momentum: $S_u = -S_\alpha U$	gas momentum: $S_u = -S_\alpha U$
gas energy: $S_h = -S_\alpha h_f$	gas energy: $S_h = -S_\alpha h_f$
phase–change: $S_\alpha = C_{blk}(\rho_{sat} - \rho_{h2o}),\ \left[\frac{kg}{m^3 s}\right]$ No size model required	phase–change: $S_\alpha = S_{nu} + S_{gr}$ nucleation S_{nu} growth model required particle size
	particle number density model $(\mathrm{N} = \#/m_g)$ $\frac{\partial}{\partial t}\left(\rho_g N\right) + \nabla\left(\rho_g \boldsymbol{U} N\right) = S_N,\ m_p = \frac{\alpha}{N} = \rho_l \frac{4}{3}\pi r^3,\ \mathrm{r} = f(\alpha, N)$
$S_\alpha = S_{nu} + S_{gr} = C[\rho_{sat} - \rho_l]\ \left[\frac{kg}{m^3 s}\right]$	$S_{gr} = \rho_g N \frac{dm_p}{dt} = 4\pi r^2 \rho_f N \frac{dr}{dt},\ S_{nu} = \frac{4}{3}\pi r_c^3 \rho_f I$ $\frac{dr}{dt}$ is from heat transfer
fog particle temperature: $T_f = T_g$	fog particle temperature by capillarity effect: $T_f = T_s - T_{sc}\frac{r_c}{r}$

Test case for steam condensation

❑ THAI HR-14 test

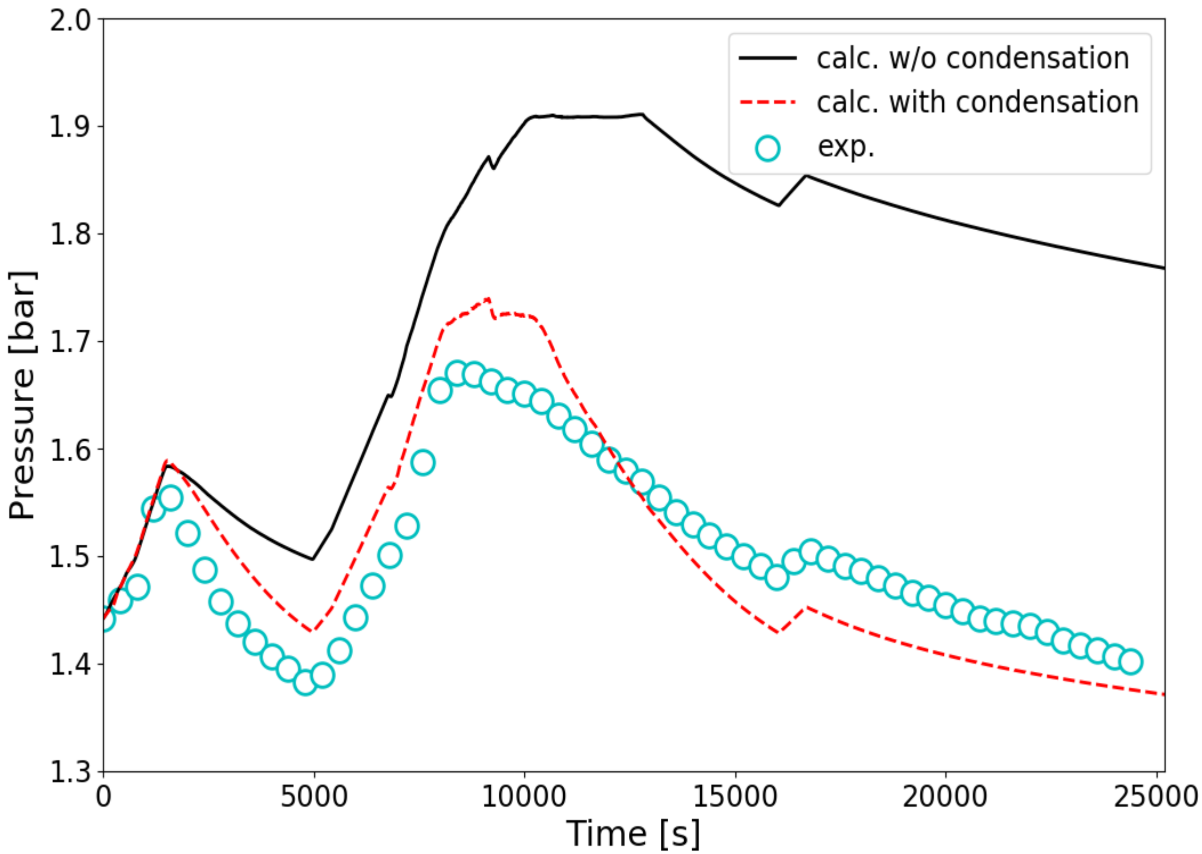
- PAR (passive auto-catalytic recombiner) test
  - Surface reaction mechanism: $H_2 + \frac{1}{2}O_2 \rightarrow H_2O + Q$
  - Condensation of PAR exhaust steam

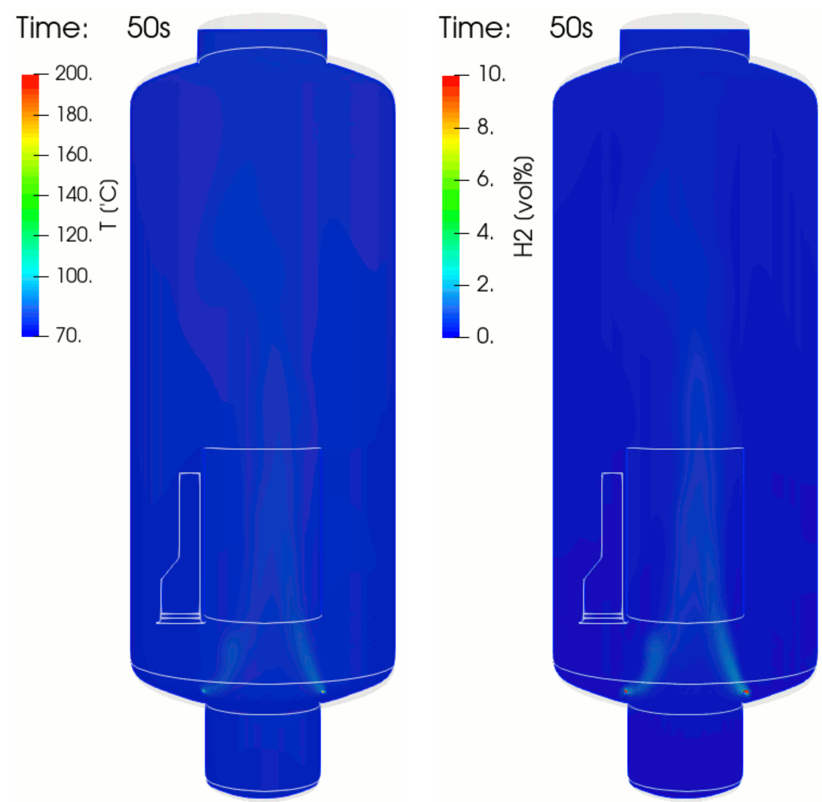




Test case for steam condensation (2)

❑ THAI HR-14 test





결언

❑ OpenFOAM 기반 다상유동 해석

❍ 어려운점

- 모델과 코드의 복잡성
  - C++ 의 특성 상 이해하기 여러운 부분이 많으나 다른 다상유동 코드와 비교하여 충분히 경쟁력 있음
- 수치적 불안전성
  - 이상유동/다상유동의 본질적인 문제로 앞으로 개선될 필요가 있음

❍ 좋은 점

- 코드의 역량
  - 그 어떤 코드보다 구조화 되어 있으며 따라서 발전 가능성이 매우 높음
- 컴뮤니티 중심 사용/개발 협업
  - 오픈폼 기반 다상유동 모델링 관련 많은 학위논문과 코드가 발표되고 있으며 개방적 협력관계를 통해 연구 목표를 달성할 수 있음

❑ 향후 연구

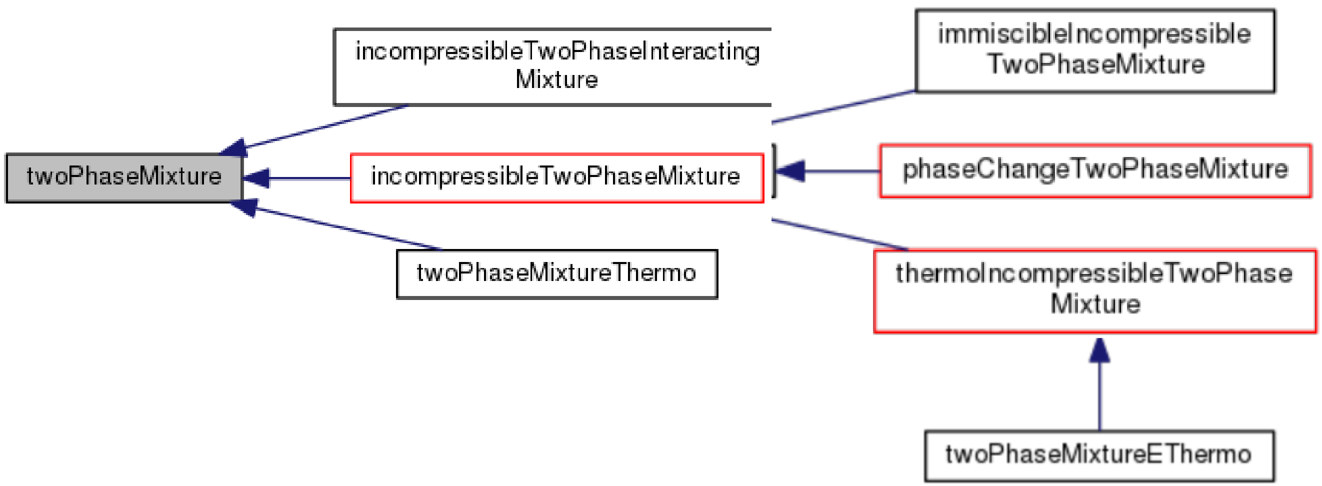
❍ 격납건물 내 다상유동 해석을 위한 오픈폼 모델의 검증 및 개선

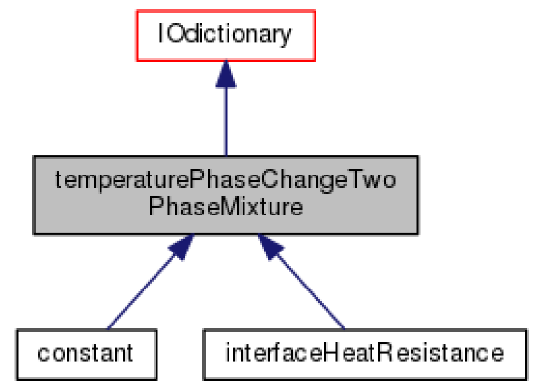
Multiphase solvers and their phase systems

Solver	approach	• Phase system	• EOS:thermo
interFoam	VOF	immiscibleIncompTwoPhaseMixture	Incom:isotherm
interIsoFoam	VOF	immiscibleIncompTwoPhaseMixture	Incom:isotherm
interPhaseChangeFoam	VOF	phaseChangeTwoPhaseMixtures	Incom:isotherm
interCondensatingEvaporatingFoam	VOF	twoPhaseMixtureEThermo	Incom:thermalT
cavitatingFoam	VOF	incompressibleTwoPhaseMixture	Barotropic:isotherm
compressibleInterFoam	VOF	*twoPhaseMixtureThermo*	comp:thermalT
MPPICInterFoam	VOF	immiscibleIncompTwoPhaseMixture	Incom:isotherm
multiphaseInterFoam	VOF	multiphaseMixture	Incom:isotherm
icoReactingMultiphaseInterFoam	VOF	multiphaseInter/multiphaseSystem	Incom:thermalT:specie
compressibleMultiphaseInterFoam	VOF	multiphaseMixtureThermo	comp:thermalT
twoPhaseEulerFoam	Euler	twoPhaseEuler/twoPhaseSystem	comp:thermalE
reactingTwoPhaseEulerFoam	Euler	reactingEuler/twoPhaseSystem	comp:thermalE:specie
chtMultiRegTwoPhaseEulerFoam	Euler	reactingEuler/twoPhaseSystem	comp:thermalE:specie:cht
multiphaseEulerFoam	Euler (+VOF)	multiphaseEuler/multiphaseSystem	comp:thermalE
reactingMultiphaseEulerFoam	Euler (+VOF)	reactingEuler/multiphaseSystem	comp:thermalE:specie
twoLiquidMixingFoam	Mixture	incompressibleTwoPhaseMixture	Incom:isotherm
driftFluxFoam	Mixture	incomTwoPhaseInteractingMixture	Incom:isotherm

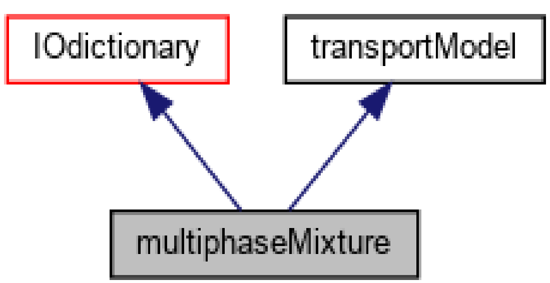
PhaseSystem of VOF/mixture solvers

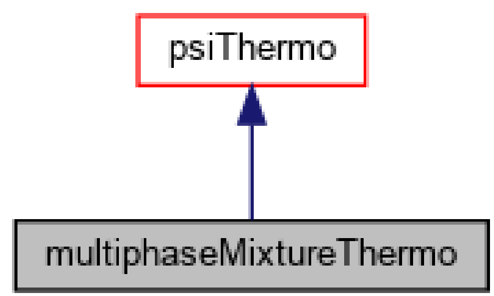
- VOF solver
  - mixtureModel = phaseSystem + phaseModel + phasePair
  - **twoPhaseMixture: without phasePair**





  - **multiphaseMixture: with phasePair**





PhaseSystem of VOF/mixture solvers

Mixture phase system	Used solvers	characteristics
twoPhaseMixture	-	base class to have alpha1, alpha2
incompTwoPM	twoLiquidMixingFoam	2 const.densities, 2 viscosity models
twoPhaseMixtureThermo	compressibleInterFoam	thermal proerties from thermo1, thermo2, inheriting interfaceProerties interface-phenomena: surface-tension
incompTPInteractingMixture	driftFluxFoam (Not VOF)	mixture of dispersed phase1 and continuous phase2, interface-transport by mixture viscosity
immiscibleIncompTwoPM	interFoam, cavitatingFoam	interfaceProperties model, inheriting interfaceProerties interface-phenomena: surface-tension
phaseChangeTwoPM	interPhaseChangeFoam (surface-tension)	pSat (pr-induce phase change), interface-phenomena: volume change
thermoIncompTwoPM	-	const. thermal propertoes
tempPhaseChangeTwoPM	interCondensatingEvaporatingFoam (surface-tension)	mixture const. thermal propertoes interface-phenomena: volume change
multiphaseMixture	multiphaseInterFoam	interfacePair for surface-tension interface-phenomena: surface-tension
multiphaseMixtureThermo	compressibleMultiphaseInterFoam	interfacePair for surface-tension interface-phenomena: surface-tension

PhaseSystem for VOF solvers (2)

- PhaseSystem in icoReactingMultiphaseInterFoam
  - incompressible multi-phase mixture with interface capturing
  - multiphaseSystem: phaseModel table, phasePair table
  - mixing rule for mixture density and thermo-physical properties

$$\varphi = \sum \alpha_i \varphi_i$$

  - mass, momentum (surface tension), heat transfer at interfaces
  - solve alpha equations

TwoPhaseSystem and multiphaseSystem

- File location
  - multiphaseSystem: reactingEuler/multiphaseSystem/multiphaseSystem
  - twoPhaseSystem: reactingEuler/twoPhaseSystem
- twoPhaseSystem: inherit **phaseSystem**
  - contains phase1 (phaseModels_[0]) and phase2 (phaseModels_[1])
  - **member function solve() solves alpha equation**
- **multiphaseSystem: inherit phaseSystem**
  - **member function solve() solves alpha equations**

TwoPhaseSystem and multiphaseSystem (2)

- phaseSystem
  - Class to represent a system of phases and model interfacial transfers between them.
  - inheriting Iodictionary (“phaseProperties”)
  - base class of twoPhaseSystem and multiphaseSystem
  - member: phaseModelList, phasePairTable, phaseModelPartialList, ...
- twoPhaseSystem
  - Class which solves the volume fraction equations for two phase
  - inheriting phaseSystem
  - includes 2 phaseModels (phase1, phase2)
  - object created by “twoPhaseSystem::New(mesh)”
- multiphaseSystem
  - Class which solves the volume fraction equations for phases
  - inheriting phaseSystem
  - object created by “multiphaseSystem::New(mesh)”