2026 한국원자력학회 춘계학술발표회 원자력 인공지능 강습회

(진흥1) 원자력 인공지능 이해하기:

PINN으로 열수력코드 만들기

전준구

Assistant Professor, NINE Lab.,

Division of Advanced Nuclear Engineering,

POSTECH

Numerical
Investigation for
Nature &
Energy Lab.

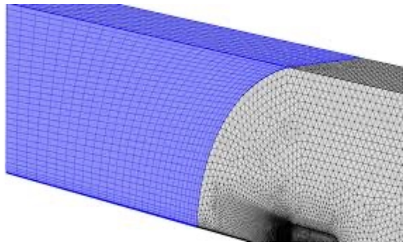
Introduction of MELCOR code

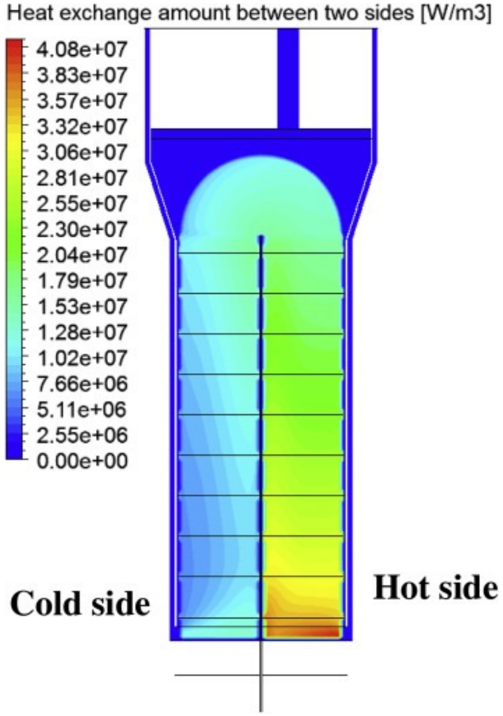
- Computational fluid dynamics (CFD) vs Thermal-hydraulics (TH) system code

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \nabla \cdot (\nu \nabla \mathbf{u}) = -\nabla p$$

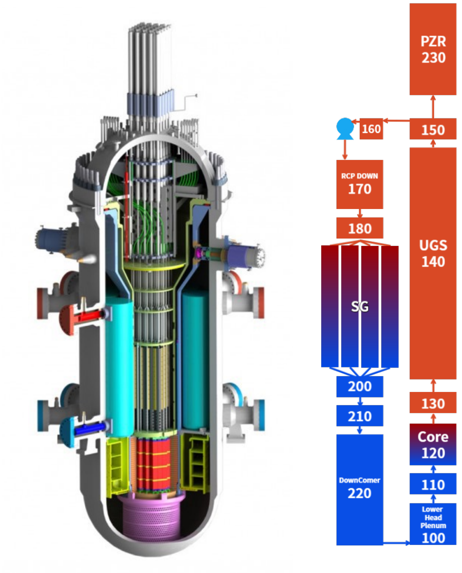
$$\alpha_{j,\varphi}\rho_{j,\varphi}L_j \frac{\partial v_{j,\varphi}}{\partial t} = \alpha_{j,\varphi}(P_i - P_k) + \alpha_{j,\varphi}(\rho g \Delta z)_{j,\varphi} + \alpha_{j,\varphi}\Delta P_j$$

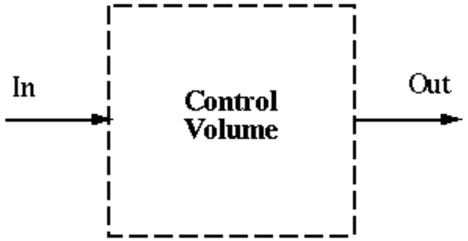
$$+\alpha_{j,\varphi}\rho_{j,\varphi}v_{j,\varphi}(\Delta v_{j,\varphi}) - \frac{1}{2}K^{*}_{j,\varphi}\alpha_{j,\varphi}\rho|v_{j,\varphi}|v_{j,\varphi} - \alpha_{j,\varphi}\alpha_{j,-\varphi}f_{2,j}L_{2,j}(v_{j,\varphi} - v_{j,-\varphi})$$





Zhao et al., 2021

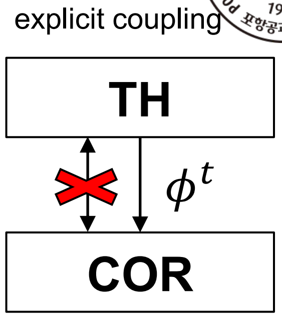


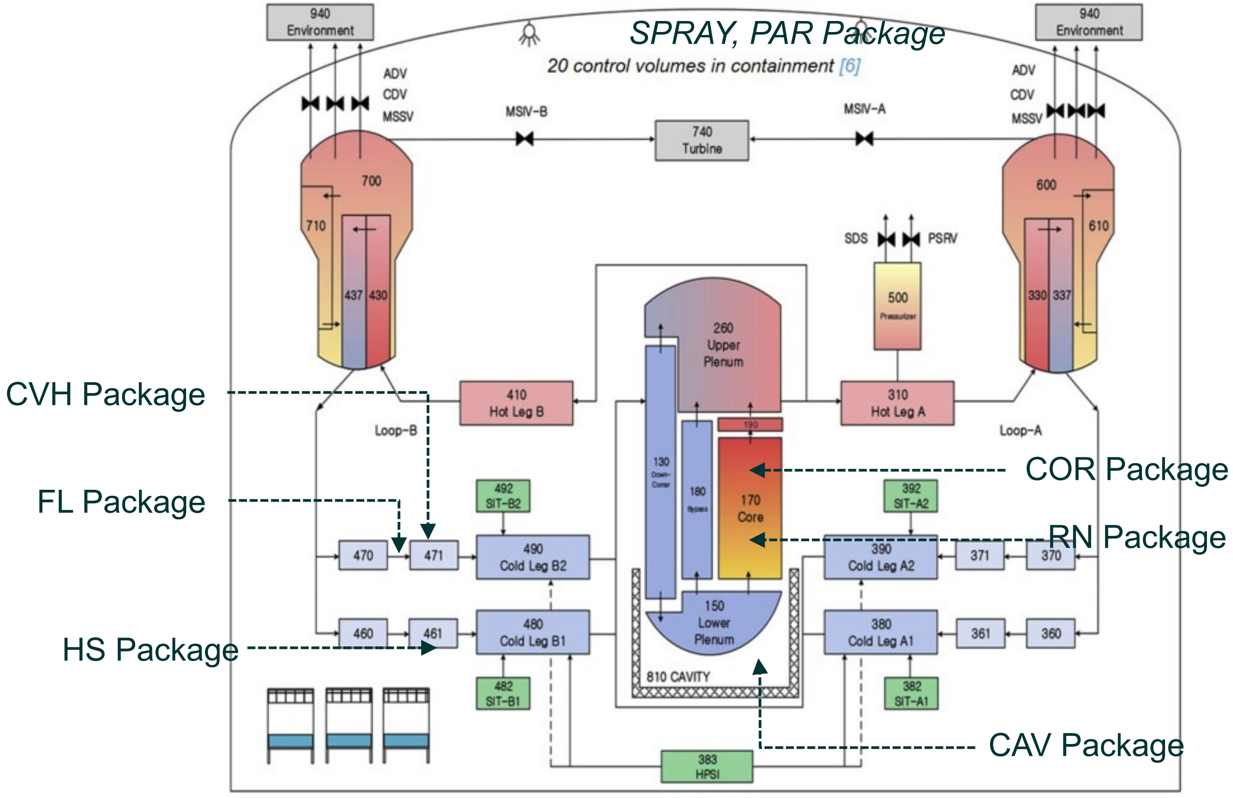


iSMR MELCOR Nodalization by NINE

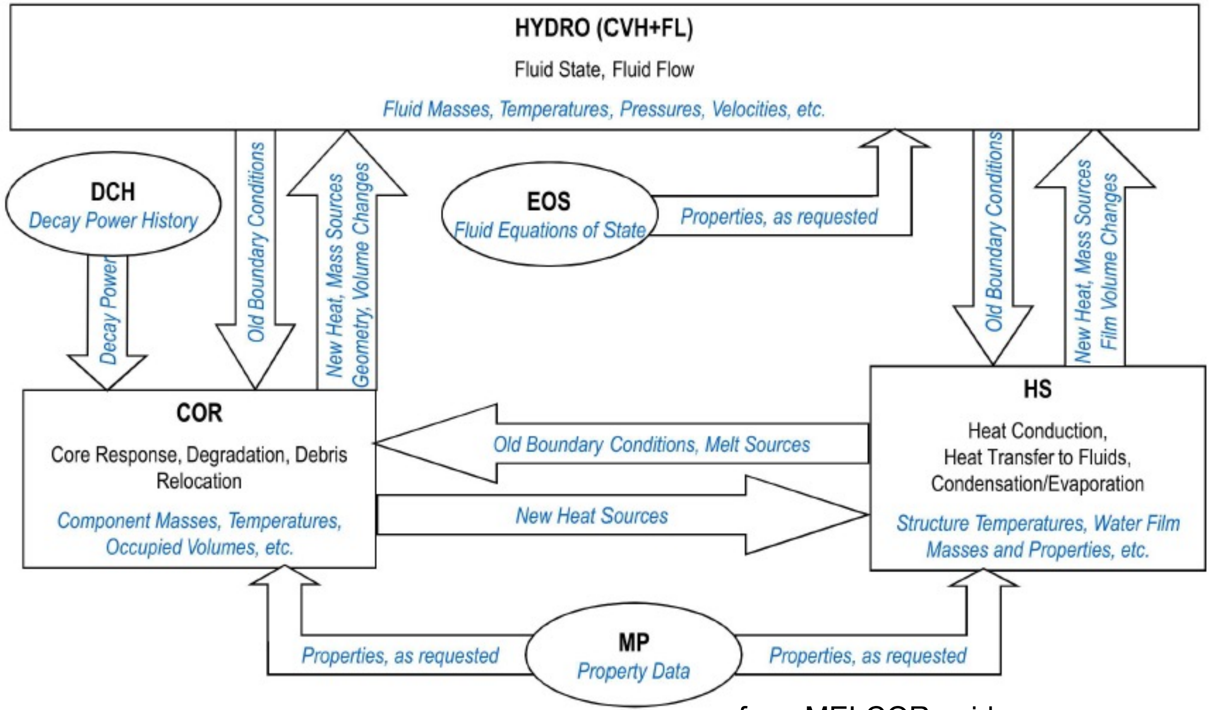
Introduction of MELCOR code

- **MELCOR**: a system code to analyze nuclear reactor severe accident (SNL/US NRC)
- **Rougher, faster, multi-physics code** (but, explicit coupling).
- Explicit coupling limits accurate **modeling of fission product behaviors**.





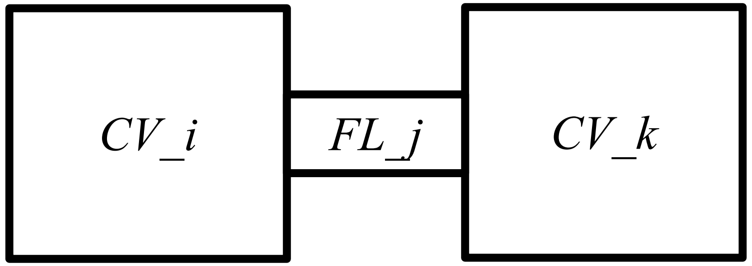
OPR1000 Nodalization by HYU



from MELCOR guide

Introduction of MELCOR code

- First principles



Mass:

$$\frac{\partial M_{i,m}}{\partial t} = \sum_j \sigma_{i,j}\alpha_{j,\phi}\rho^d_{j,m}v_{j,\phi}F_jA_j + \dot{M}_{i,m}$$

Momentum:

$$\overset{1)}{\alpha_{j,\phi}\rho_{j,\phi}L_j\frac{\partial v_{j,\phi}}{\partial t}} = \overset{2)}{\alpha_{j,\phi}(P_i - P_k)} + \overset{3)}{\alpha_{j,\phi}(\rho g\Delta z)_{j,\phi}} + \overset{3)}{\alpha_{j,\phi}\Delta P_j}$$

$$\underset{4)}{-\frac{1}{2}K^*_{j,\phi}\alpha_{j,\phi}\rho_{j,\phi}|v_{j,\phi}|v_{j,\phi}} - \underset{5)}{\alpha_{j,\phi}\alpha_{j,-\phi}f_{2,j}L_{2,j}(v_{j,\phi} - v_{j,-\phi})} + \underset{6)}{\alpha_{j,\phi}\rho_{j,\phi}v_{j,\phi}(\Delta v)_{j,\phi}}$$

Energy:

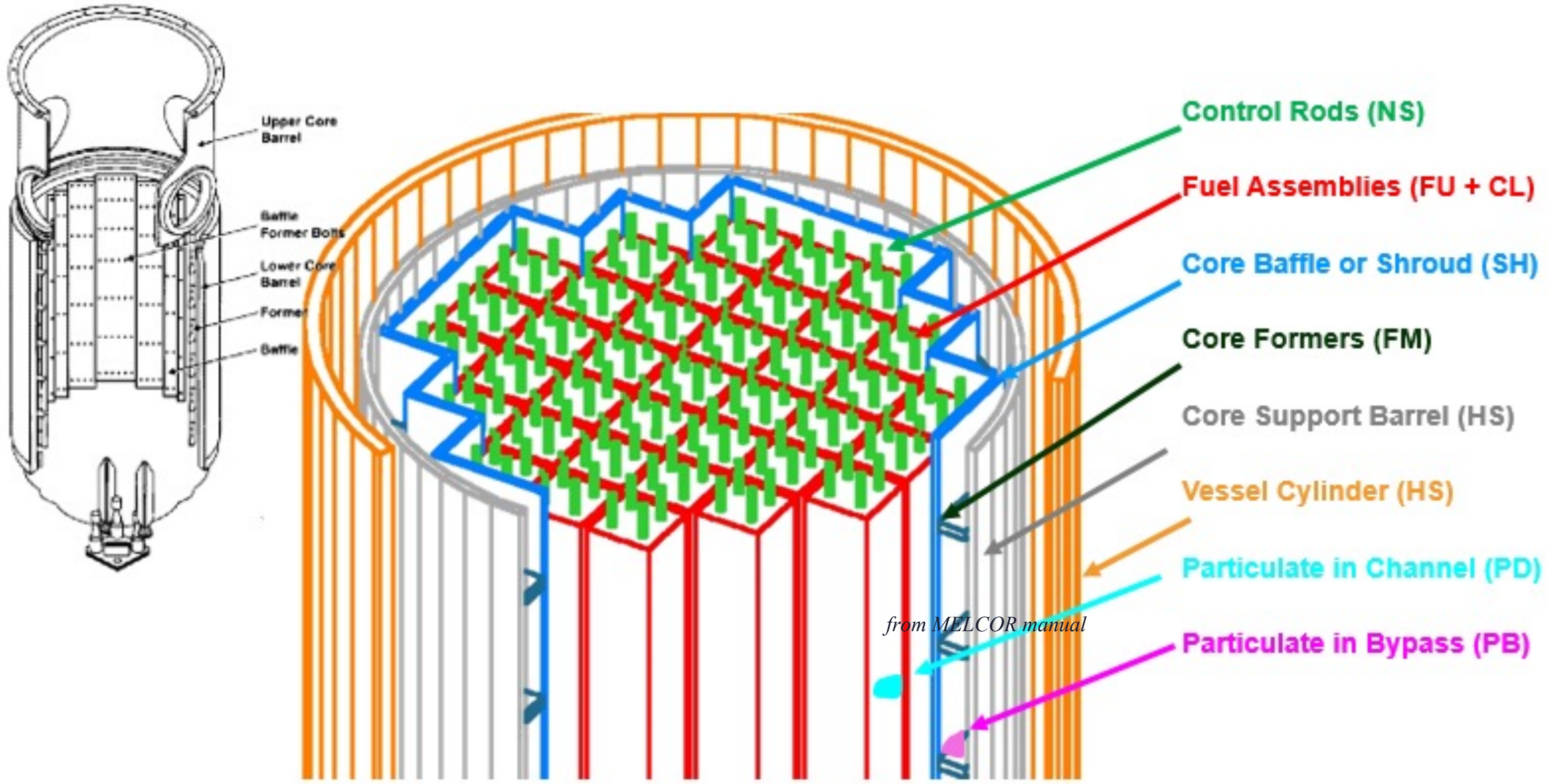
$$\frac{\partial E_{i,\phi}}{\partial t} = \sum_j \sigma_{i,j}\alpha_{j,\phi}\left(\sum_m \rho^d_{j,m}\, h^d_{j,m}\right)v_{j,\phi}F_jA_j + \dot{H}_{i,\phi}$$

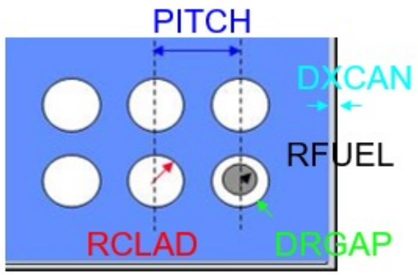
1) Transient term
2) Pressure gradient
3) Body force
4) Wall fraction
5) Interface frictional drag
6) Acceleration force

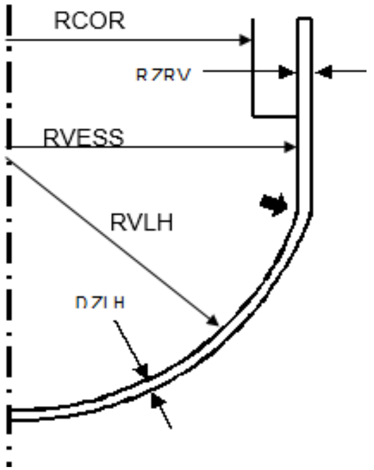
MELCOR manual, Sandia National Laboratories

Introduction of MELCOR code

- MELCOR CORE modeling

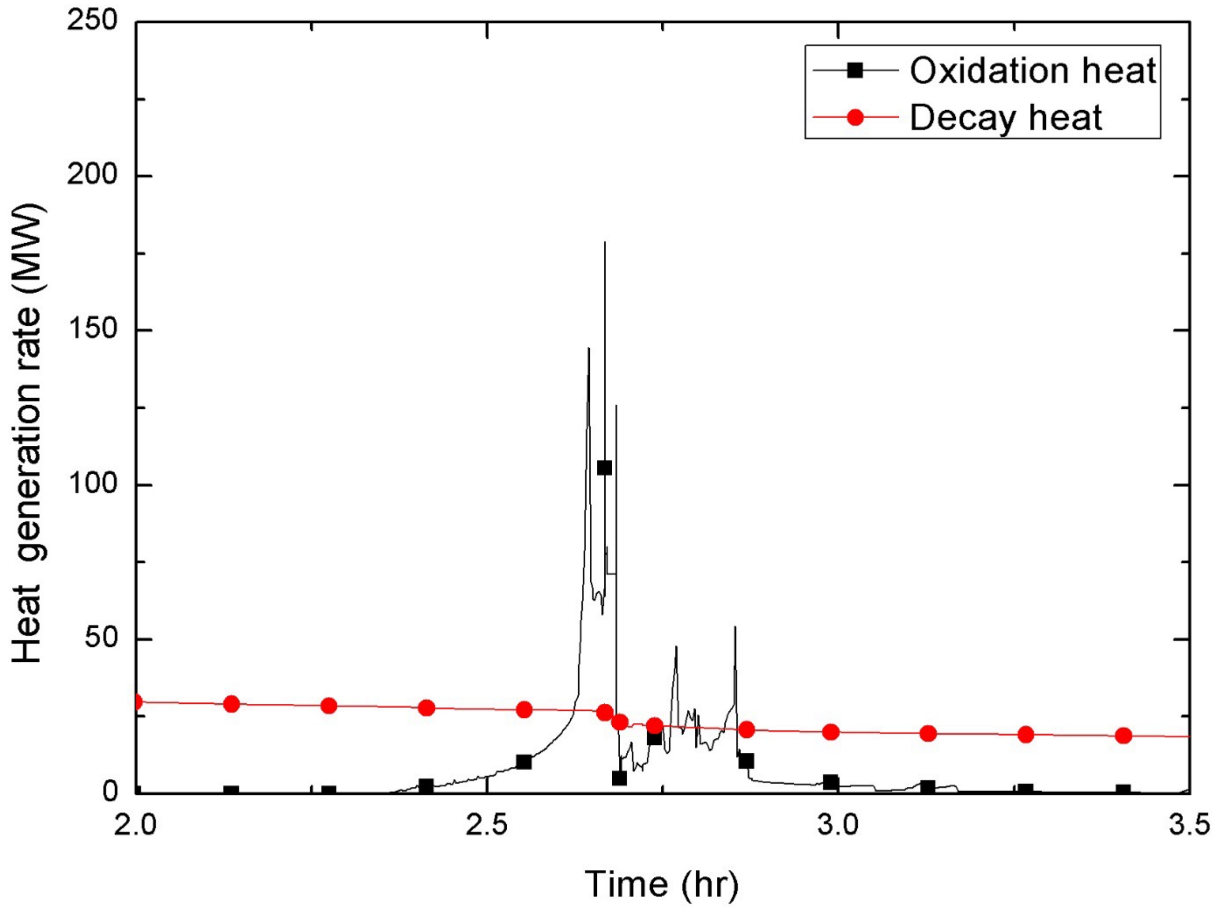






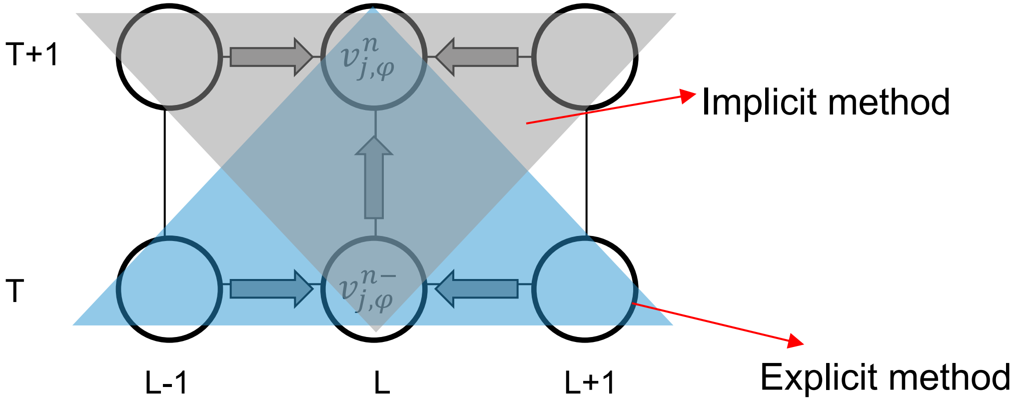
Introduction of MELCOR code

- Example of accident analysis



Semi-implicit Numerical Scheme

- Logic of momentum calculation



Scheme for semi-implicit method

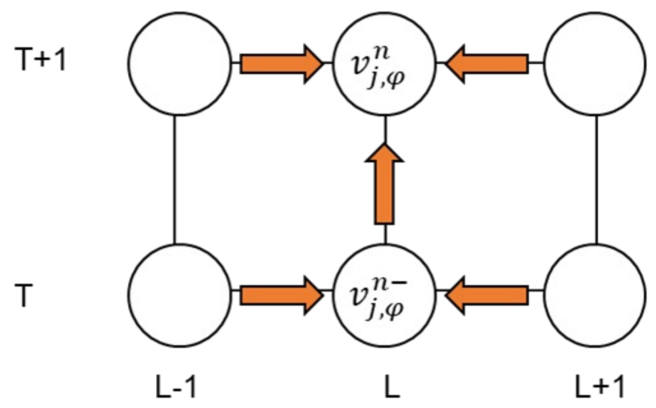
$$v^{n}_{j,\varphi} = v^{o+}_{j,\varphi} + \frac{\Delta t}{\rho_{j,\varphi} L_j}\left(P^{n}_{i} + \Delta P_j - P^{n}_{k} + (\rho g \Delta z)^{n}_{j,\varphi} + v^{o}_{j,\varphi}(\rho \Delta v)^{o}_{j,\varphi}\right)$$
$$- \frac{K *_{j,\varphi} \Delta t}{2L_j}\left(\left|v^{n-}_{j,\varphi} + v'_{j,\varphi}\right| v^{n}_{j,\varphi} - \left|v'_{j,\varphi}\right| v^{n-}_{j,\varphi}\right) - \frac{\alpha_{j,-\varphi} f_{2,j} L_{2,j} \Delta t}{\rho_{j,\varphi} L_j}\left(v^{n}_{j,\varphi} - v^{n}_{j,-\varphi}\right)$$

Momentum equation

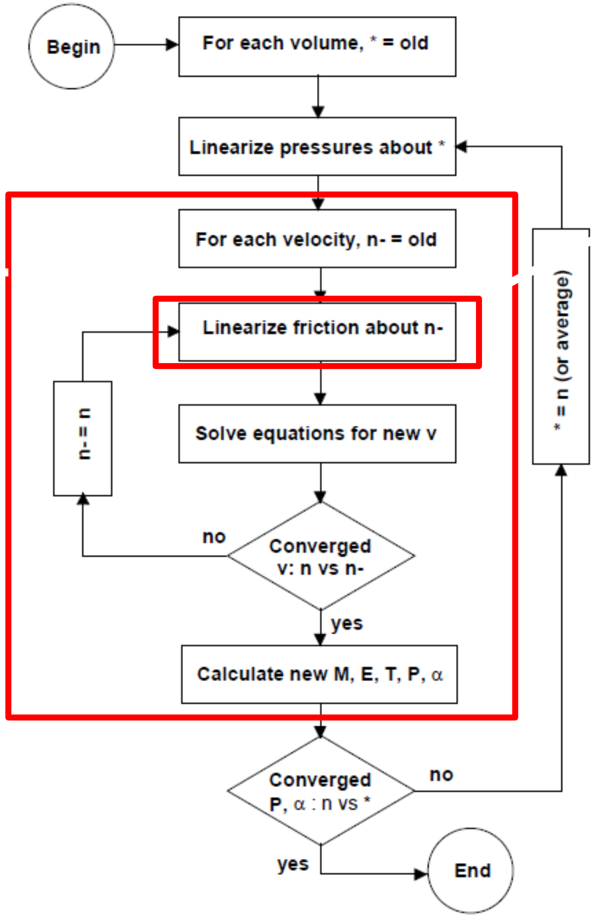
Implicit method	predicting the future value with current value and the future surrounding values
+	
Explicit method	calculating the future value through information given in the present status.
⇩	
Semi-implicit method	adopts both methods by treating some values as implicit method and other in explicit method thereby bring the stability and accuracy in the calculation

Semi-implicit Numerical Scheme

- Logic of momentum calculation: that's why MELCOR uses old boundary method



Scheme for semi-implicit method



Logic of the MELCOR Calculation

$$\alpha_{j,\varphi}\rho_{j,\varphi}L_j\frac{\partial v_{j,\varphi}}{\partial t} = \alpha_{j,\varphi}(P_i - P_k) + \alpha_{j,\varphi}(\rho g\Delta z)_{j,\varphi} + \alpha_{j,\varphi}\Delta P_j$$

$$+\alpha_{j,\varphi}\rho_{j,\varphi}v_{j,\varphi}(\Delta v_{j,\varphi}) - \frac{1}{2}K_{j,\varphi}^{*}\alpha_{j,\varphi}\rho|v_{j,\varphi}|v_{j,\varphi} - \alpha_{j,\varphi}\alpha_{j,-\varphi}f_{2,j}L_{2,j}(v_{j,\varphi} - v_{j,-\varphi})$$

$$v_{j,\varphi}^{n} = \left(v_{j,\varphi}^{o+} + \frac{\Delta t}{\rho_{j,\varphi}L_j}\left((\rho g\Delta z)_{j,\varphi}^{n} + v_{j,\varphi}^{o}(\rho\Delta v)_{j,\varphi}^{o}\right) + \frac{K_{j,\varphi}^{*}\Delta t}{2L_j}\left(|v'_{j,\varphi}|v_{j,\varphi}^{n-}\right) + \frac{\alpha_{j,-\varphi}f_{2,j}L_{2,j}\Delta t}{\rho_{j,\varphi}L_j} * v_{j,-\varphi}^{n}\right)$$
$$/\left(1 + \frac{\alpha_{j,-\varphi}f_{2,j}L_{2,j}\Delta t}{\rho_{j,\varphi}L_j} + \frac{K_{j,\varphi}^{*}\Delta t}{2L_j}\left(|v_{j,\varphi}^{n-} + v'_{j,\varphi}|\right)\right)$$

Equation variation for adequate style

MELCOR VS PINN

How to solve governing Equations

	How to solve Momentum Conservation Equation	Mass Conservation Equation
PINN	$\alpha_{j,\varphi}\rho_{j,\varphi}L_j\frac{\partial v_{j,\varphi}}{\partial t} = \alpha_{j,\varphi}(P_i - P_k) + \alpha_{j,\varphi}(\rho g\Delta z)_{j,\varphi} + \alpha_{j,\varphi}\Delta P_j$ $+\alpha_{j,\varphi}\rho_{j,\varphi}v_{j,\varphi}(\Delta v_{j,\varphi}) - \frac{1}{2}K^*_{j,\varphi}\alpha_{j,\varphi}\rho\lvert v_{j,\varphi}\rvert v_{j,\varphi} - \alpha_{j,\varphi}\alpha_{j,-\varphi}f_{2,j}L_{2,j}(v_{j,\varphi} - v_{j,-\varphi})$	$\frac{\partial M_{i,m}}{\partial t} = \sum_j \sigma_{ij}\alpha^n_{j,\varphi}\rho^d_{j,m}v^n_{j,\varphi}F_jA_j$
MELCOR	$v^n_{j,\varphi} = \left(v^{o+}_{j,\varphi} + \frac{\Delta t}{\rho_{j,\varphi}L_j}\left((\rho g\Delta z)^n_{j,\varphi} + v^o_{j,\varphi}(\rho\Delta v)^o_{j,\varphi}\right) + \frac{K^*_{j,\varphi}\Delta t}{2L_j}\left(\lvert v'_{j,\varphi}\rvert v^{n-}_{j,\varphi}\right) + \frac{\alpha_{j,-\varphi}f_{2,j}L_{2,j}\Delta t}{\rho_{j,\varphi}L_j} * v^n_{j,-\varphi}\right)$ $/\left(1 + \frac{\alpha_{j,-\varphi}f_{2,j}L_{2,j}\Delta t}{\rho_{j,\varphi}L_j} + \frac{K^*_{j,\varphi}\Delta t}{2L_j}\left(\lvert v^{n-}_{j,\varphi} + v'_{j,\varphi}\rvert\right)\right)$	$M^n_{i,m} = M^o_{i,m} + \sum_j \sigma_{ij}\alpha^n_{j,\varphi}\rho^d_{j,m}v^n_{j,\varphi}F_jA_j\Delta t$

Crucial!

☑ Equation Comparison

- Two of the main equations used in this scenario displayed.
- MELCOR utilizes the FDM, whereas PINNs employ the continuous form of PDEs.
- FDM is alternative form of PDE equation with linear approximation and MELCOR's experience
- Future PINN validation and sensitivity analysis requires understanding FDM-form equations.

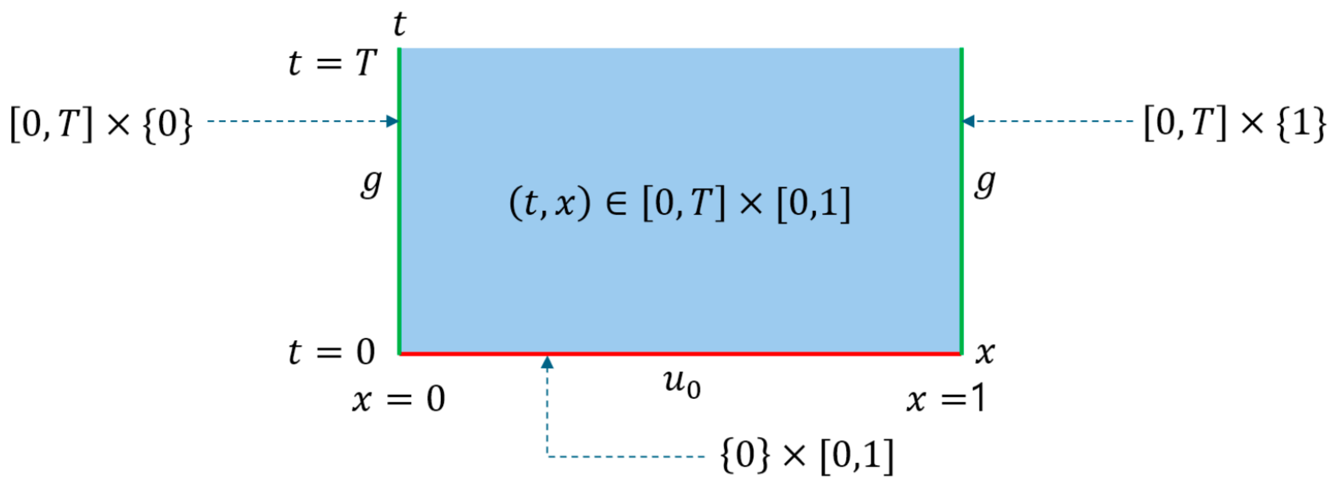
Physics-informed neural networks

Initial-boundary value problem (heat equation)

$$\frac{\partial u(t,x)}{\partial t} - \kappa \frac{\partial^2 u(t,x)}{\partial x^2} = f(t,x) \quad (t,x) \in [0, T] \times [0,1], \qquad \text{PDE}$$

$$u(t,x) = g(t,x) \quad (t,x) \in [0, T] \times \{0,1\}, \qquad \text{Boundary Condition}$$

$$u(x,0) = u_0(x) \quad x \in [0,1]. \qquad \text{Initial Condition}$$



Physics-informed neural networks

$$\frac{\partial u}{\partial t} - \kappa \frac{\partial^2 u}{\partial x^2} = f \quad \Longleftrightarrow \quad R_1(u) = \left| \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - f \right|^2 = 0 \qquad (t, x) \in [0, T] \times [0, 1],$$

$$u = g \quad \Longleftrightarrow \quad R_2(u) = |u - g|^2 = 0 \qquad (t, x) \in [0, T] \times \{0, 1\},$$

$$u(x, 0) = u_0(x) \quad \Longleftrightarrow \quad R_3(u) = |u - u_0|^2 = 0 \qquad x \in [0, 1].$$

A way to find a solution:

$$\underbrace{R_1(h) + R_2(h) + R_3(h) \to 0}_{\text{minimize}} \quad \Longrightarrow \quad h \to u \quad \text{(solution of the initial-boundary value problem)}$$

Physics-informed neural networks

To embed physics laws into neural networks (physics-informed):

- For all $(t, x) \in [0, T] \times [0, 1]$ and a neural network $u(t, x; \theta)$, we should minimize

$$R_1(u(t,x;\theta)) + R_2(u(t,x;\theta)) + R_3(u(t,x;\theta)).$$

- Set a (population) loss function as

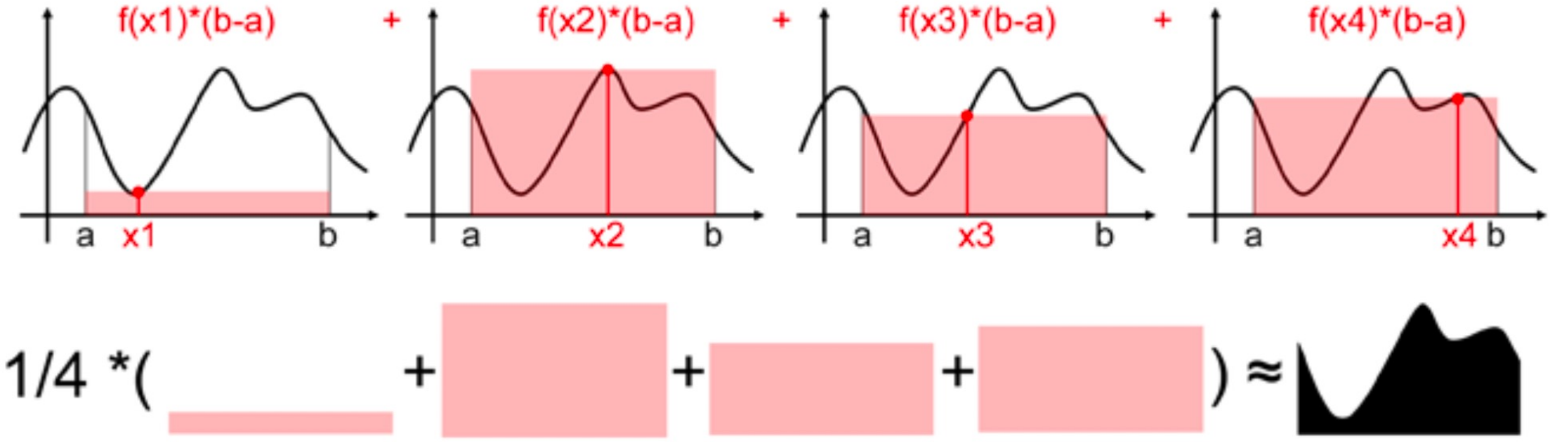
$$\mathcal{L}(\theta) = \underbrace{\int_0^T \int_0^1 R_1(u(t,x;\theta))\,\mathrm{d}x\mathrm{d}t}_{\text{PDE loss}} + \underbrace{\left(\int_0^T R_2(u(t,0;\theta))\,\mathrm{d}t + \int_0^T R_2(u(t,1;\theta))\,\mathrm{d}t\right)}_{\text{B.C. loss}} + \underbrace{\int_0^1 R_3(u(0,t;\theta))\,\mathrm{d}x}_{\text{I.C. loss}}.$$

- This population loss function is ideal but has some drawbacks:
  - Computing the integral causes additional errors.
  - High-dimensional PDEs: the curse of dimensionality.

Physics-informed neural networks

Monte-Carlo Integration

$$\int_{\Omega} f(x)\,\mathrm{d}x \quad \approx \quad \frac{1}{N}\sum_{j=1}^{N} f(x_j)|\Omega| \quad \text{with } x_j \sim \mathcal{U}(\Omega)$$



Use Monte-Carlo integration to compute the population loss function for PINNs

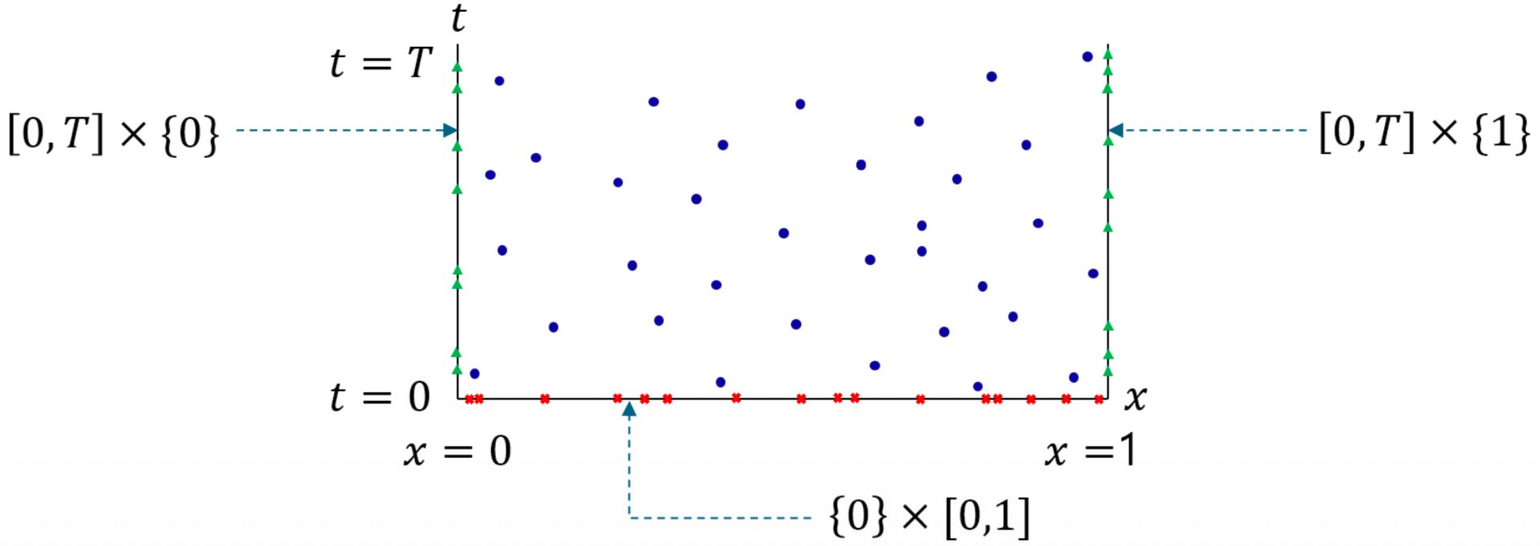
Physics-informed neural networks

(Empirical) loss function for Physics-Informed Neural Networks:

$$\mathcal{L}(\theta) = \frac{1}{N_1}\sum_{i=1}^{N_1}\left|\frac{\partial u(t_i, x_i;\theta)}{\partial t} - \frac{\partial^2 u(t_i, x_i;\theta)}{\partial x^2} - f\right|^2 + \frac{1}{N_2}\sum_{j=1}^{N_2}\left|u(t_j, x_j;\theta) - g(t_j, x_j)\right|^2 + \frac{1}{N_3}\sum_{k=1}^{N_3}\left|u(0, x_k;\theta) - u_0(x_k)\right|^2$$

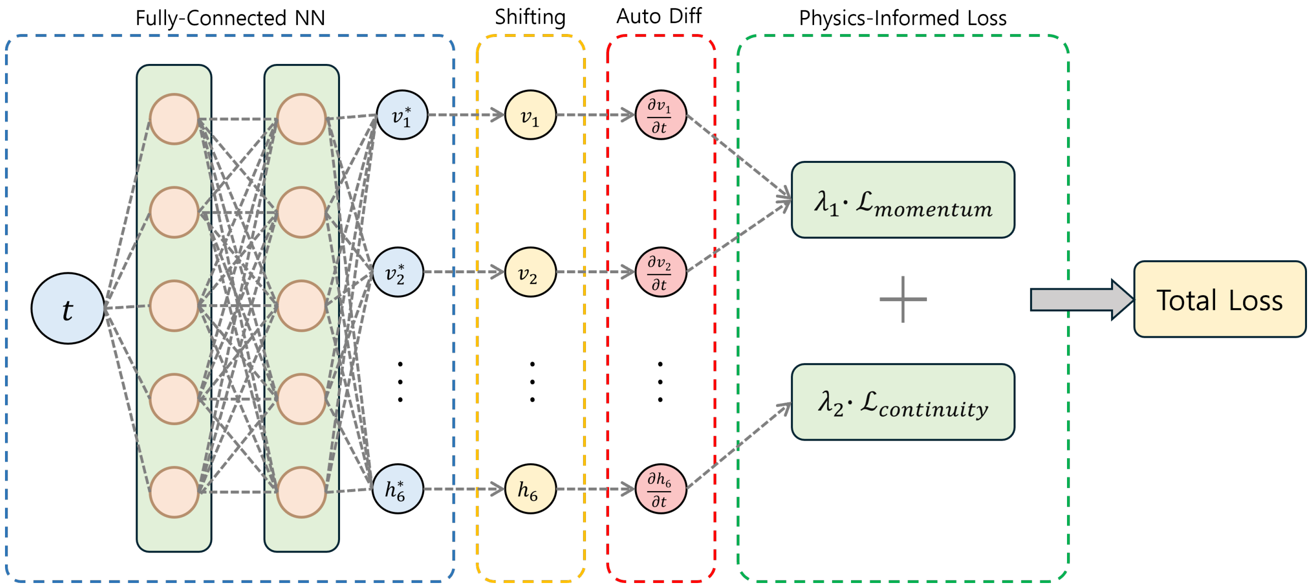
where

$$\underbrace{(t_i, x_i) \sim \mathcal{U}([0,T]\times[0,1])}_{\text{PDE collocation points}}, \quad \underbrace{(t_j, x_j) \sim \mathcal{U}([0,T]\times\{0,1\})}_{\text{boundary collocation points}}, \quad \underbrace{x_k \sim \mathcal{U}([0,1])}_{\text{initial collocation points}}.$$



Vanilla PINN

Vanilla PINN Architecture



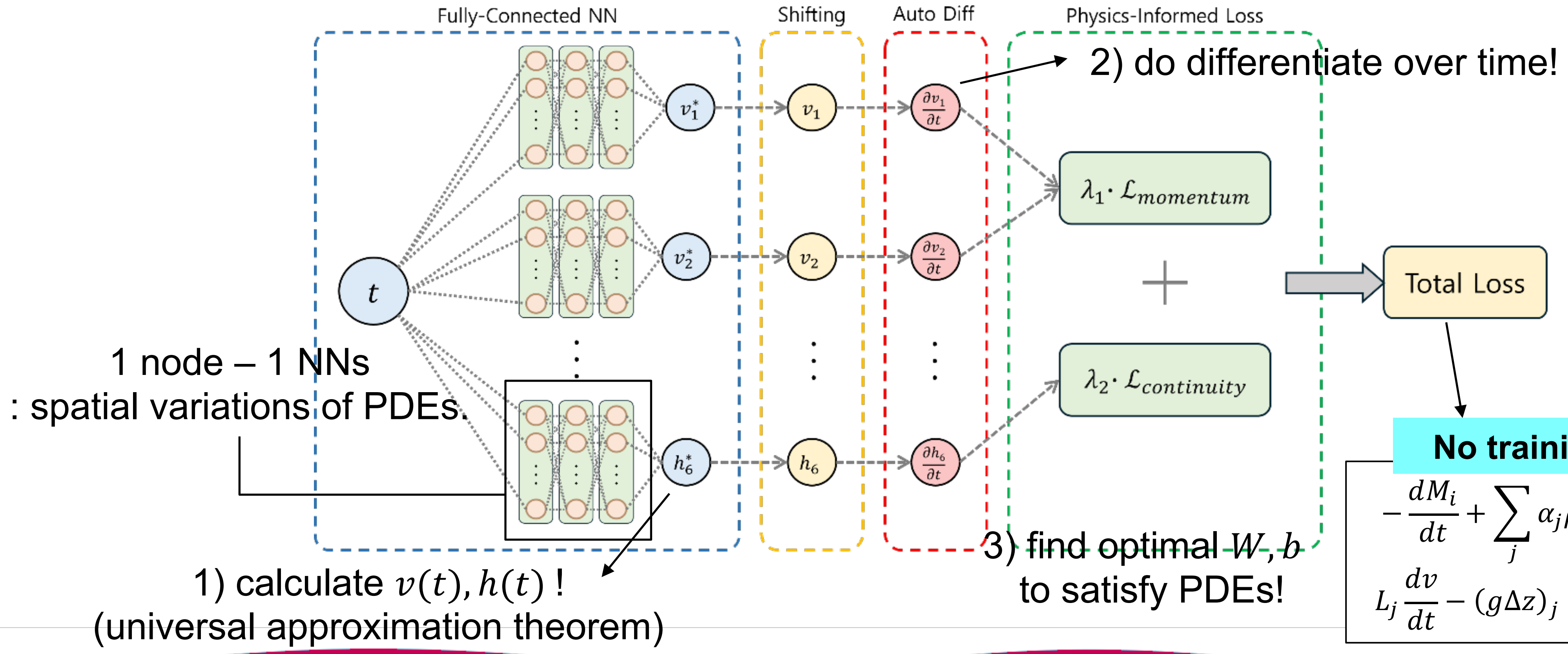
Based on MELCOR characteristics, the input is 1D (time), and the output is $2N_i - 1$ (N_i = number of CV)

There are N_i continuity equations for the CVs and $N_i - 1$ momentum equations for the FLs.

Shin et al., Physics-informed neural networks for thermal-hydraulic system simulation: CVH/FL/HS module, Progress in Nuclear Energy (2026).

NA-PINN (Node-assigned PINN)

- We are developing **AI-based MELCOR code for implicit coupled multi-physics analysis.**
  - ❖ Physics-informed neural networks (PINNs)
  - ❖ We proposed a node assigned PINNs (**NA-PINN) suitable for control volume approach TH code.**
  - ❖ **This is the first study to develop TH code in a multi-node environment (without data).**



$$-\frac{dM_i}{dt} + \sum_j \alpha_j \rho_j v_j A_k = 0$$

$$L_j \frac{dv}{dt} - (g\Delta z)_j + \frac{1}{2} K_j v_j^2 = 0$$

Shin et al., Physics-informed neural networks for thermal-hydraulic system simulation: CVH/FL/HS module, Progress in Nuclear Energy (2026).

NA-PINN (Node-assigned PINN)

NA-PINN Training Process

1. Initialize parameters of all neural networks in the set, $\boldsymbol{\Theta} = \{\theta_{\hat{v}_1}, \dots, \theta_{\hat{v}_{n-1}}, \theta_{H_1}, \dots, \theta_{H_n}\}$

2. Collocation points are sampled within the physical domain.

3. Predict $\{\hat{v}_j(t')\}_{j=1}^{n-1}$ and $\{H_i(t')\}_{i=1}^{n}$ using each neural network

4. Compute temporal derivatives via automatic differentiation

5. Compute residuals of governing equations, then construct the loss function $\mathcal{L}_{PDE}$.

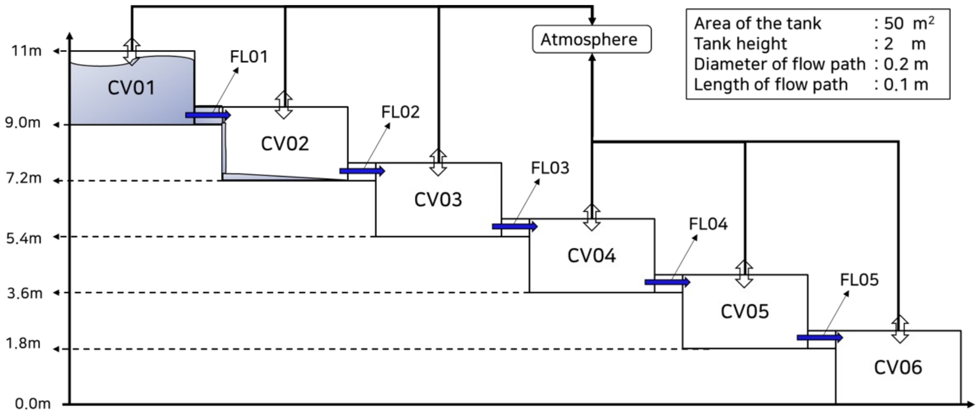
6. Update parameters using a learning rate: $\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} - \eta_{sched} \nabla_{\Theta} \mathcal{L}_{PDE}$

Shin et al., Physics-informed neural networks for thermal-hydraulic system simulation: CVH/FL/HS module, Progress in Nuclear Energy (2026).

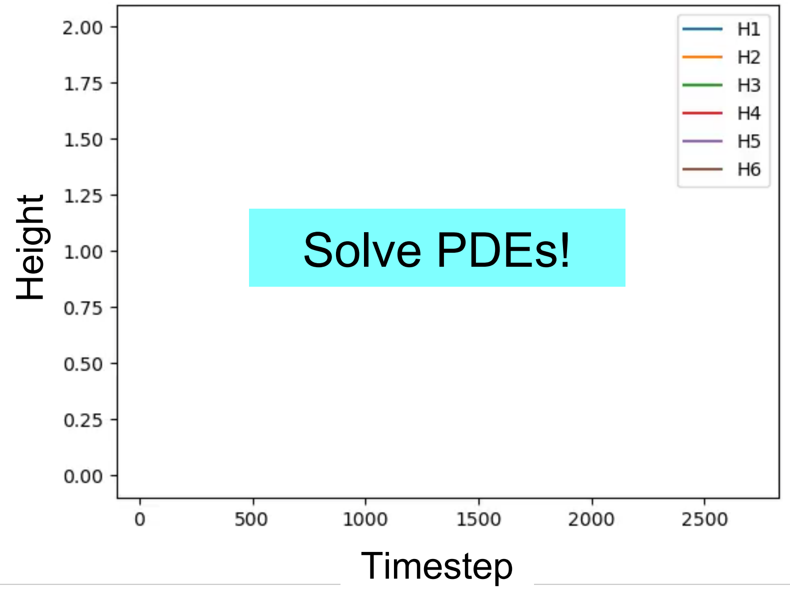
NA-PINN (Node-assigned PINN)

- Super fast (no inference time).
- How to solve the **single instance problem**?

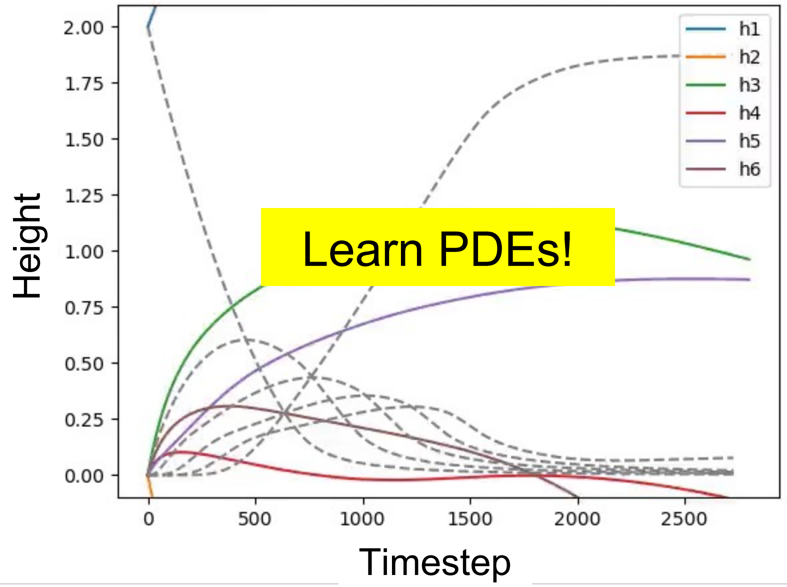
  - use surrogate manner



FDM: time stepping



PINN: epoch stepping



Check the paper!

Shin et al., Physics-informed neural networks for thermal-hydraulic system simulation: CVH/FL/HS module, Progress in Nuclear Energy (2026).