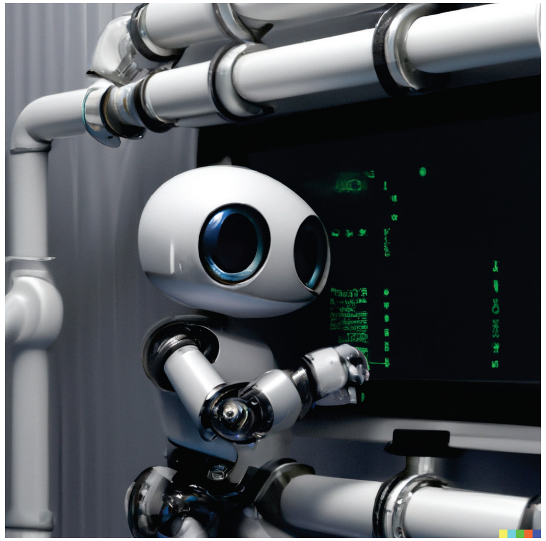
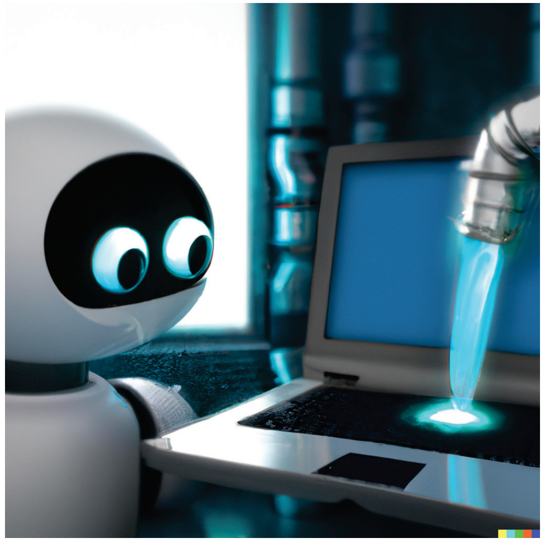
한국원자력학회 워크샵

CFD 해석 기술 향상을 위한 인공지능 활용 전략 논의

인공지능 믿을 수 있을까?

by DALLE2





전준구

Assistant Professor, NINE Lab,

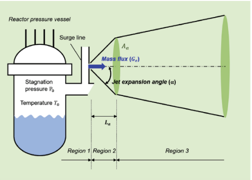
Graduate School of Integrated Energy-AI

Jeonbuk National University

2023.05.17.

My research overview

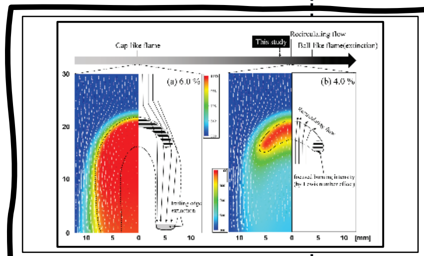
[1] J. Jeon et al., *Ann. Nucl. Energy*, 2018.
[2] J. Jeon et al., *Nucl. Eng. Technol.*, 2019.
[3] J. Jeon et al., *Energies*, 2020.
[4] J. Jeon et al., *Int. J. Heat Mass. Transf.*,2021
[5] J. Jeon et al., *Nucl. Eng. Technol.*, 2019.
[6] J. Jeon et al., *Nucl. Eng. Technol.*, 2021.
[7] J. Jeon et al., *Int. J. Energy Res.*, 2022.
[8] J. Jeon et al., arXiv preprint arXiv:2206.06817

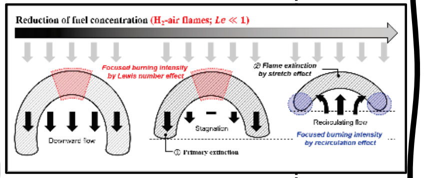


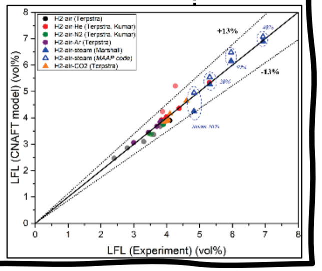
2016 — **2018** — **2020** — **2022** — **2023**

- *Numerical simulation of **severe accident** scenarios [1].*
- ***Turbulent gas behavior** at pipe rupture accident [2].*

"The need for hydrogen-LFL model"



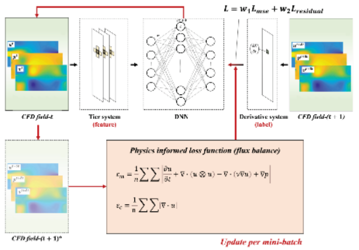


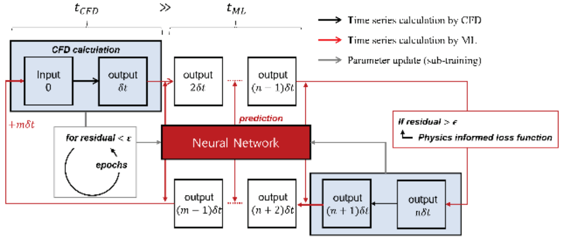


- *Identification of the **hydrogen flame extinction mechanism** with CFD simulation [3, 4].*
- *Analytical modeling and validation of **hydrogen flammability limit model** [5, 6].*
- *Development of **hydrogen combustion risk prediction code** for containment building (flammability, flame acceleration, DDT evaluation).*

"Need to accelerate CFD simulations"

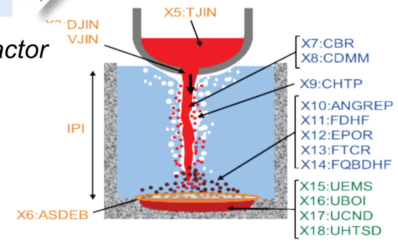
- *Development of a **new concept of network model** by introducing CFD principles [7].*
- *Validation of the model using **non-reacting and reacting flows**.*
- ***Physics-informed transfer learning strategy** to accelerate unsteady simulations [8].*





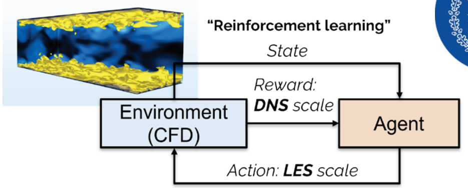
AI application for nuclear safety

- *Development of nuclear reactor severe accident.*
- *Surrogate model for corium heat transfer*
- *Accident prognosis AI*



"Turbulent near-wall region modeling"

- *Limitation of algebraic (logarithmic) wall model and TBLE model*
- *Development of DRL method for turbulent near-wall region modeling*
- *Sod2d (spectral element method code) + TensorFlow (DRL library)*



Contents

① Background and objective

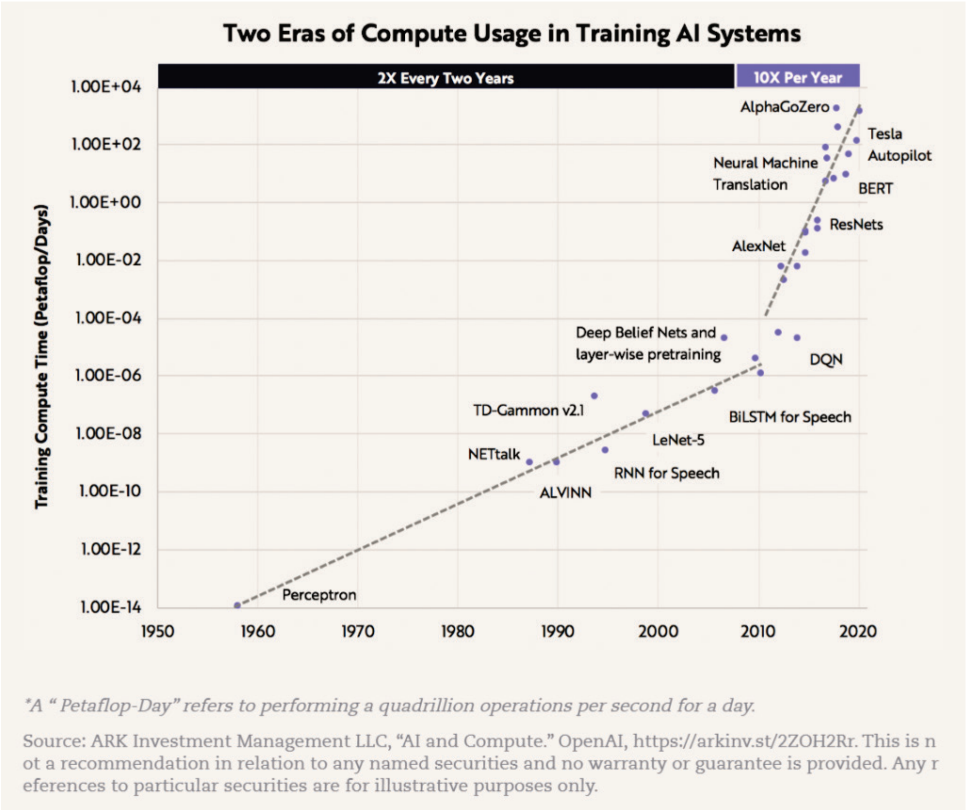
② Achilles heel of CFD

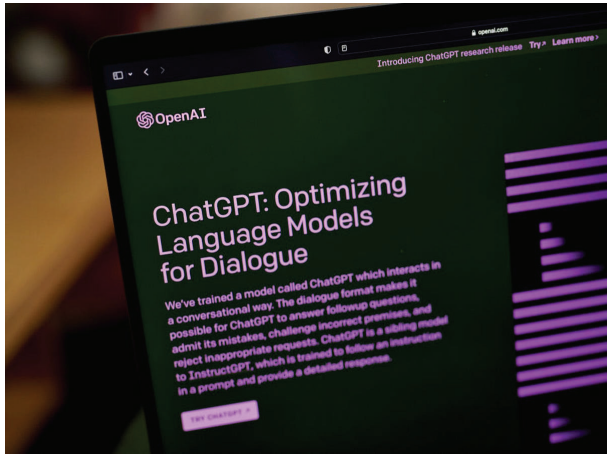
③ What is the role of AI in CFD?

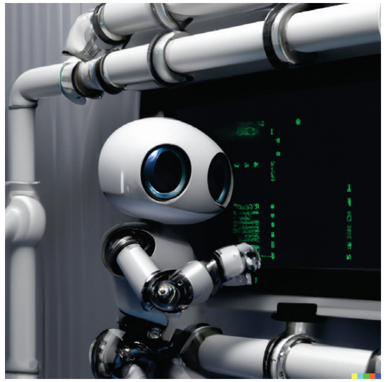
④ Summary and conclusions

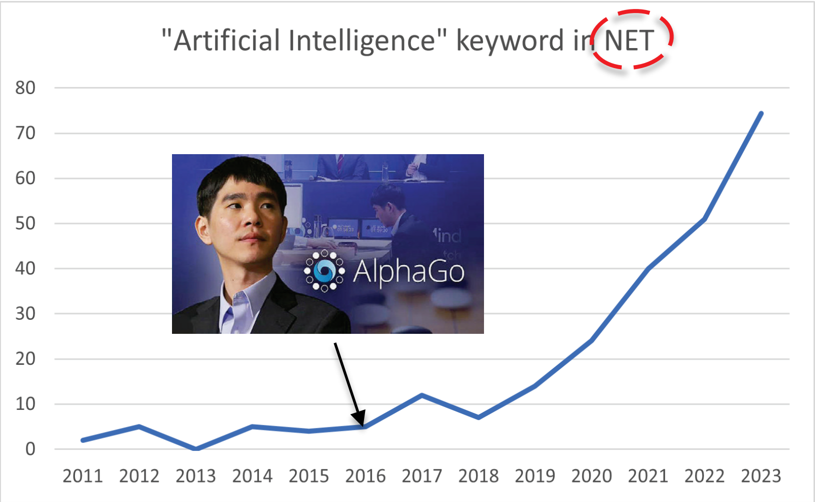
Background and objective

- A modern **big wave of machine learning** has propagated to all industries.

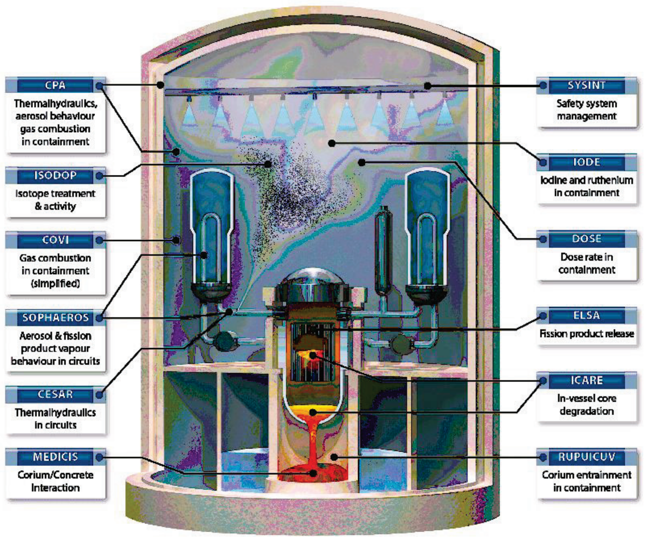




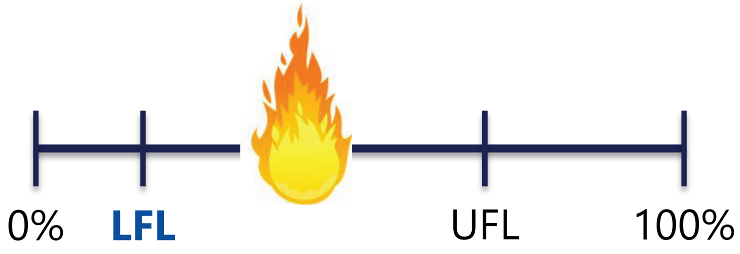


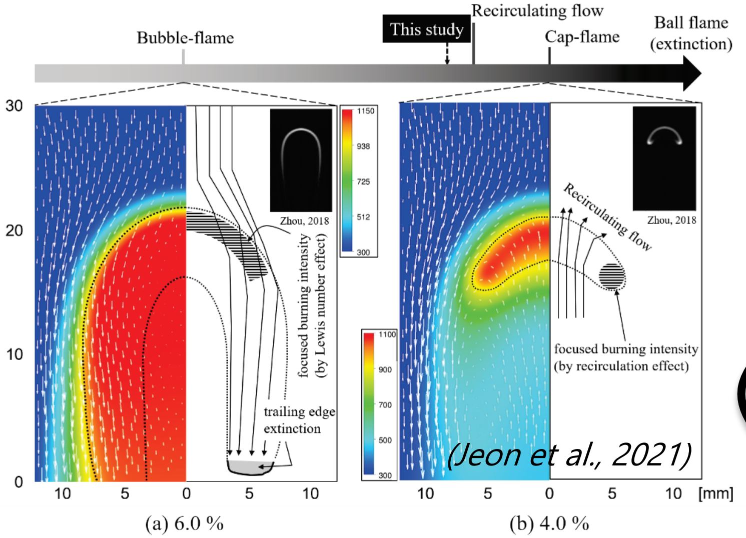


Background and objective



Nuclear reactor severe accident



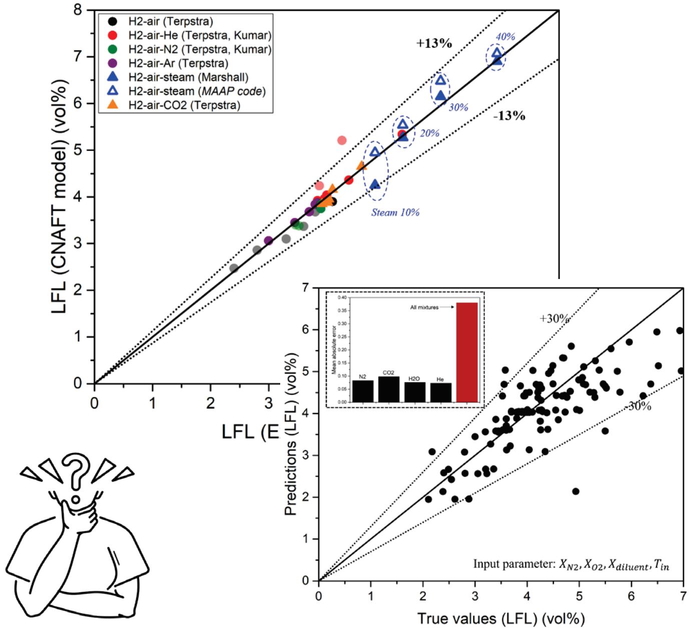


Hydrogen LFL model was developed by elucidation of flame extinction mechanism:

$$0.207(\pi - \pi_{ref}) = \sum_{reactants} n_i[\Delta H^0_{f,i} + \bar{c}_{p,i}(T_i - T_{ref})] - \sum_{products} n_i[\Delta H^0_{f,i} + \bar{c}_{p,i}(T_{CNAFT} - T_{ref})]$$

now being used by national research institute

(Jeon et al., 2021)



Mechanistic model vs AI regression model

Where AI can do best & most needed!

① Massive data

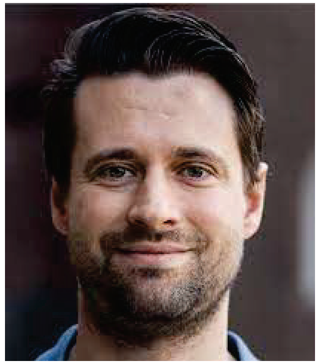
② Multi-dimensional data

③ Non-analytical (non-linear) data

Background and objective

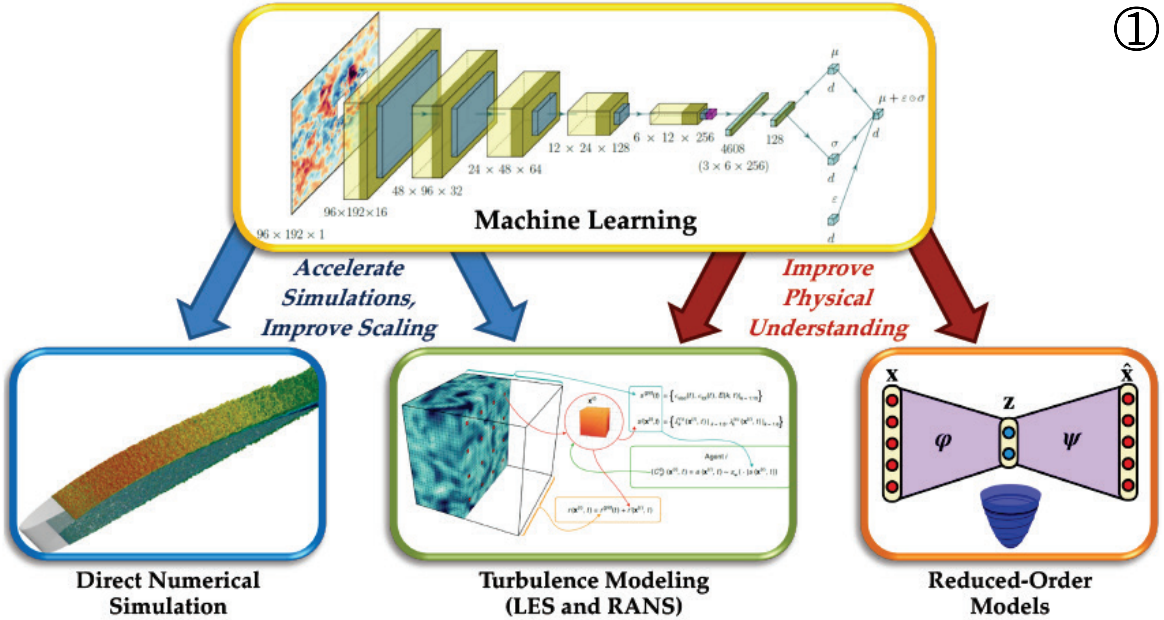
- Big wave of machine learning to fluid dynamics community

“Enhancing computational fluid dynamics with machine learning (2022)
Nature Computational Science”

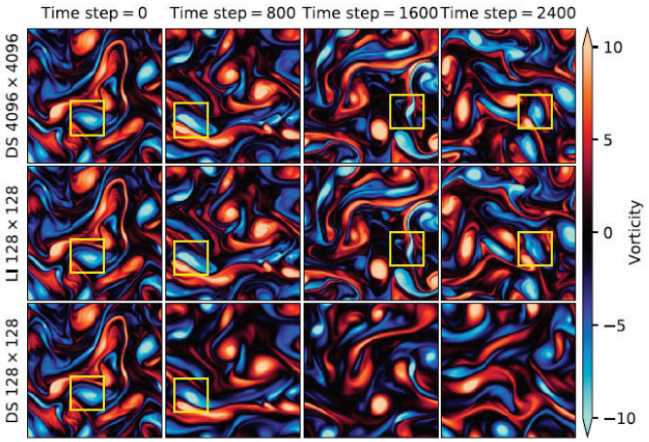


Prof. R. Vinuesa

Prof. S.L. Brunton



① **Increasing the speed of computational fluid dynamics**
- Finding spatial derivatives in low-resolution grids
- finite volume discretization scheme neural networks
- solving Poisson equation with deep learning



$$\frac{\Delta t}{\rho}\nabla^2 p = -\nabla \cdot u^*$$

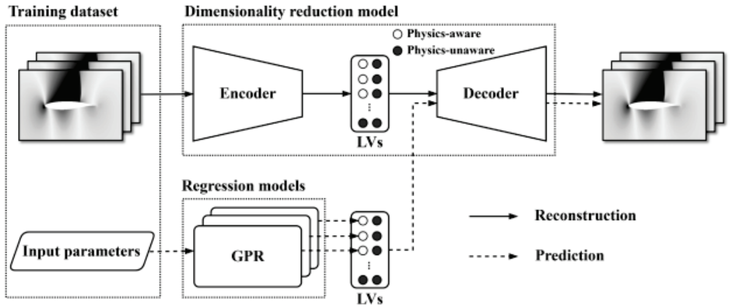
Background and objective

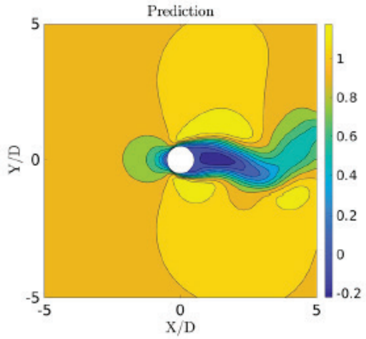
- Big wave of machine learning to fluid dynamics community

② **Turbulent modeling**
- DNS quantities → supervised learning to LES simulation
- Sub grid scale (SGS) turbulent modeling with reinforcement learning

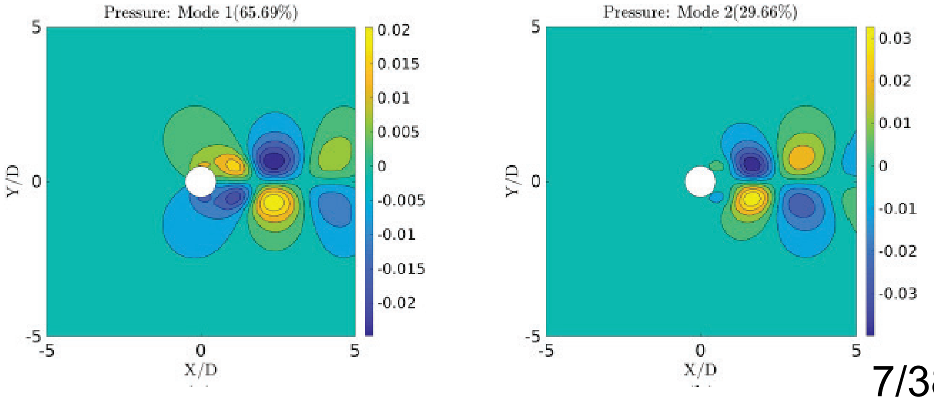
③ **Reduced order models (ROMs)**
- even complex flows often exhibit a few dominant coherent structures.
- to extract flow mode for flow control
- for more efficient data-driven methods.

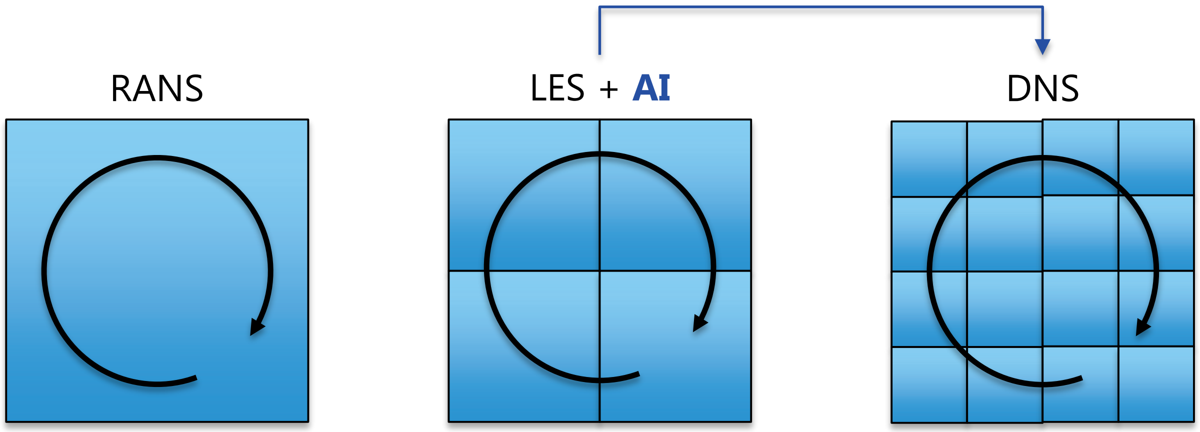
(Y.E. Kang, 2022)



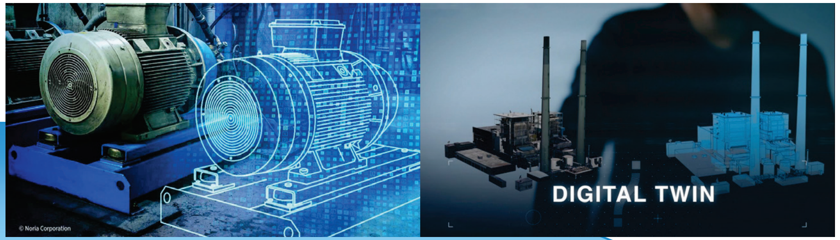


(S.R. Bukka, 2021)





Background and objective



Digital twin

Energy safety

Energy efficiency

...

Fluid mechanics

...

System Engineering

...

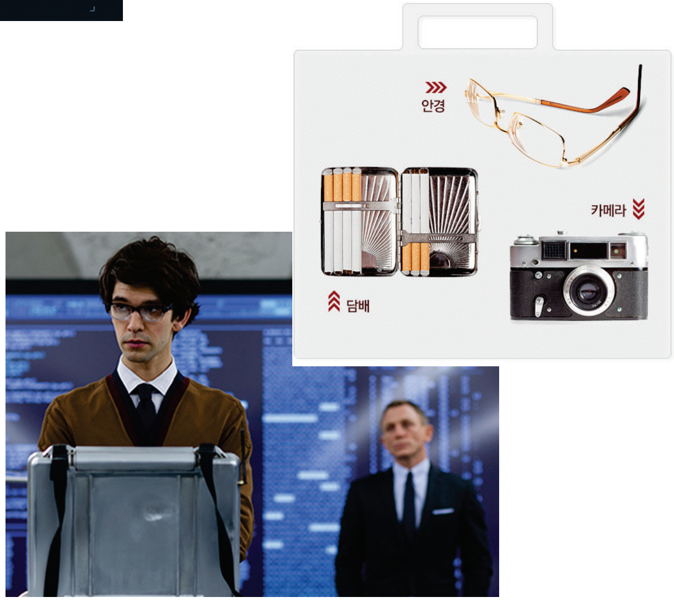
CFD

System code

Missing parts
(ex. turbulent models, computation speed)

Missing parts
(ex. accident prognosis, model accuracy)

State-of-the art AI techniques



"CFD= 007, AI= Agent Q"

Contents

① Background and objective

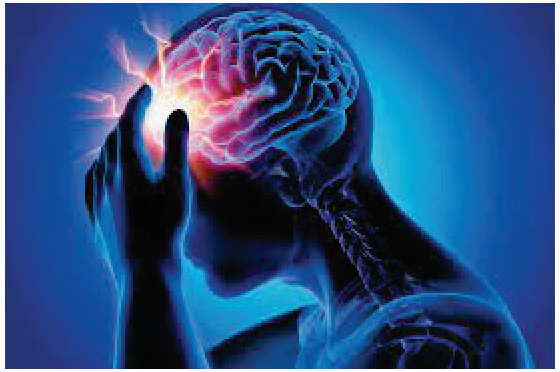
② Achilles heel of CFD

③ What is the role of AI in CFD?

④ Summary and conclusions

What is CFD?

mass, momentum, energy, species equations



- How it simulates fluid flow.

① **Derivation of mathematical equations**

② **Numerical method** for solving partial differential equations (PDEs)

③ **Mesh generation**

④ **Simulation**

mass conservation by finite difference method

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho z)}{\partial z} = 0$$

$$\frac{\rho^{t+1} - \rho^{t}}{\delta t} + \frac{(\rho u)^{t}_{i+1jk} - (\rho u)^{t}_{ijk}}{\delta x} + \frac{(\rho v)^{t}_{ij+1k} - (\rho v)^{t}_{ijk}}{\delta y} + \dots$$

$$\begin{aligned}\binom{mass\ change}{rate} = &\binom{x - inflow}{rate} - \binom{x - outflow}{rate} \\ &+ \binom{y - inflow}{rate} - \binom{y - outflow}{rate} \\ &+ \binom{z - inflow}{rate} - \binom{z - outflow}{rate} \\ &+ \binom{source}{term}\end{aligned}$$

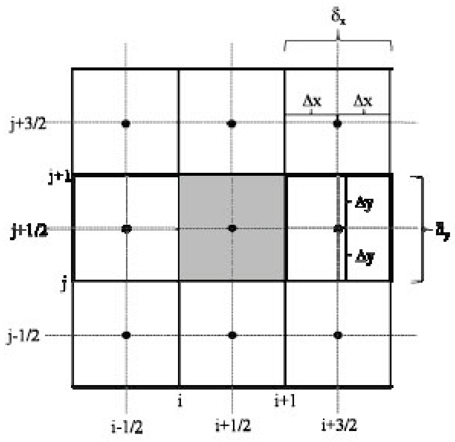
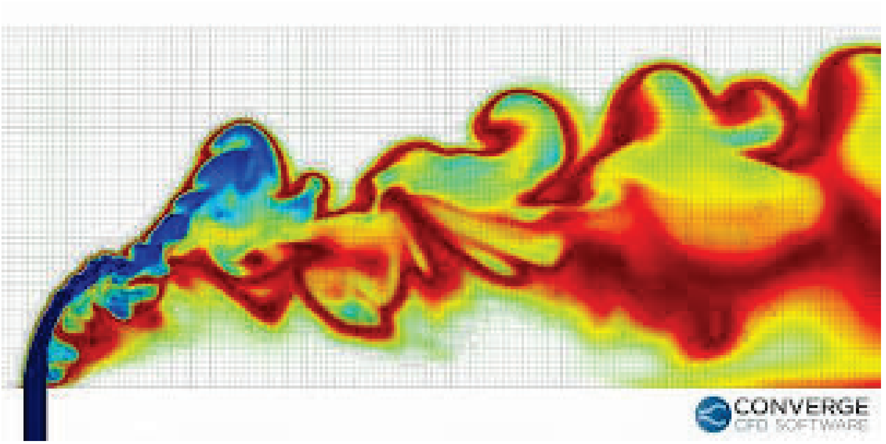


Figure 1: Representation of the control volume.



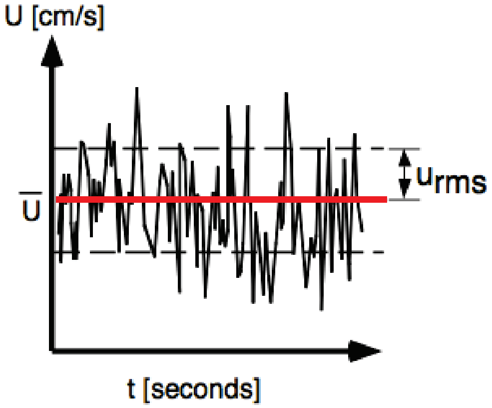
What is CFD?

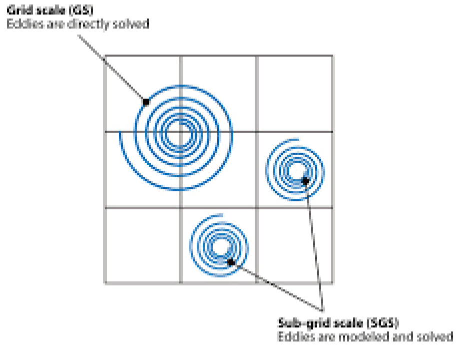
- Turbulent flow is more and more complex…

Laminar: $$\frac{\partial u}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} - \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + g$$

Adorable laminar flow

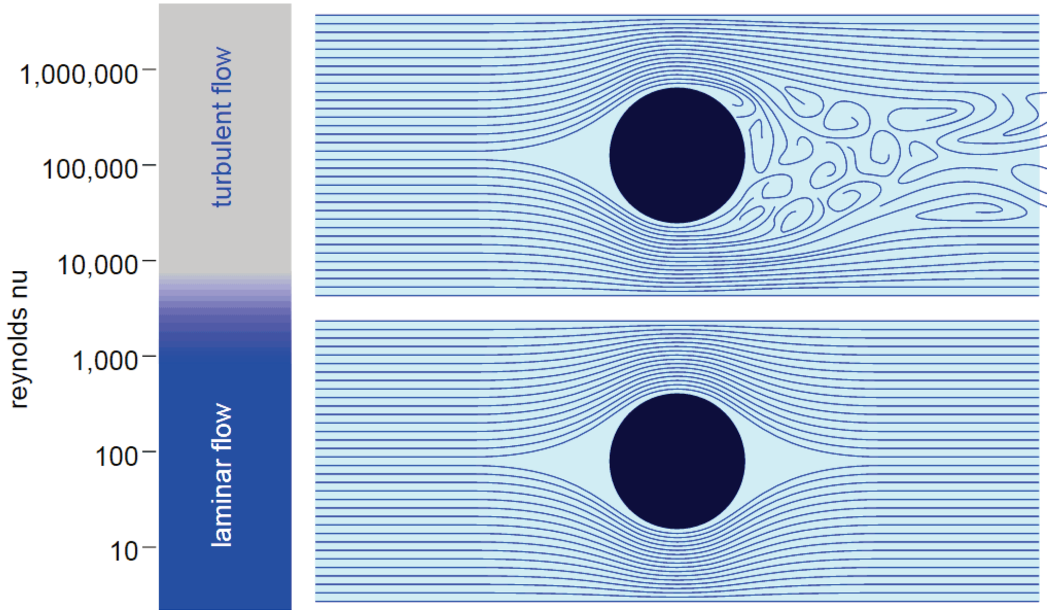
Terrible turbulent flow





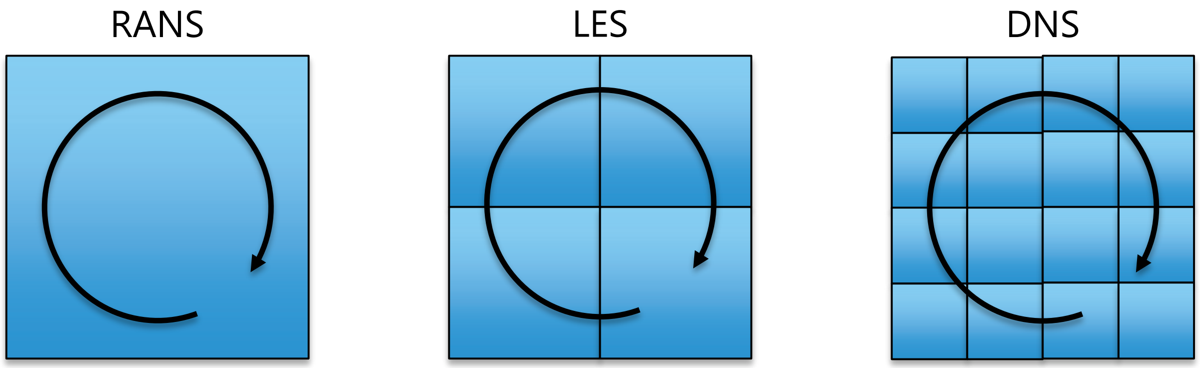
Turbulent: $u(x,t) = \bar{u}(x) + \boldsymbol{u}'(\boldsymbol{x},\boldsymbol{t})$

$$\frac{\partial \bar{u}}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} - \nu \frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_i} + \bar{g} - \frac{\partial \overline{u_i' u_j'}}{\partial x_j}$$

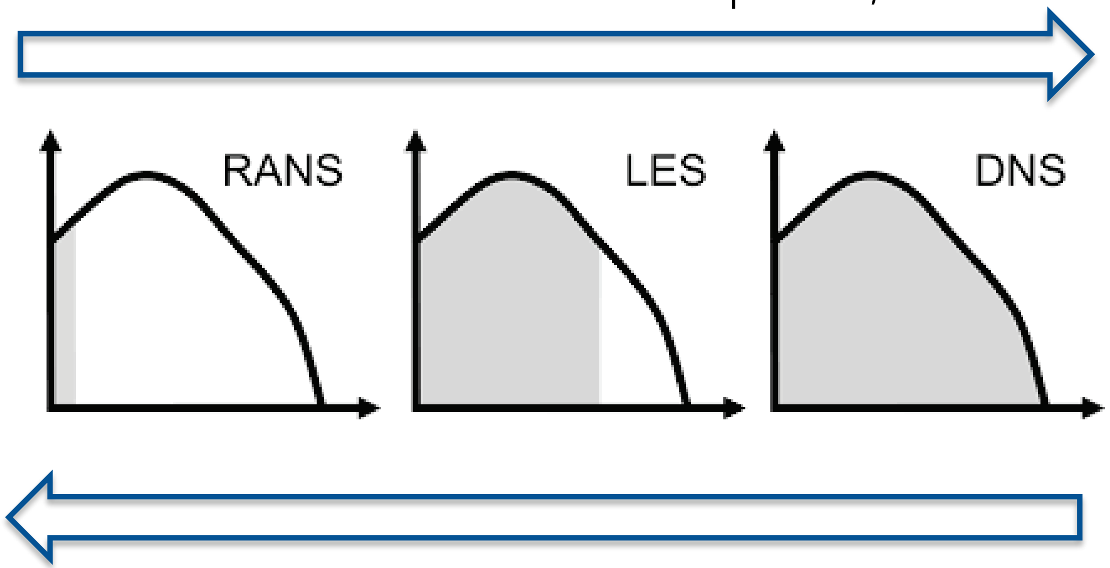


What is CFD?

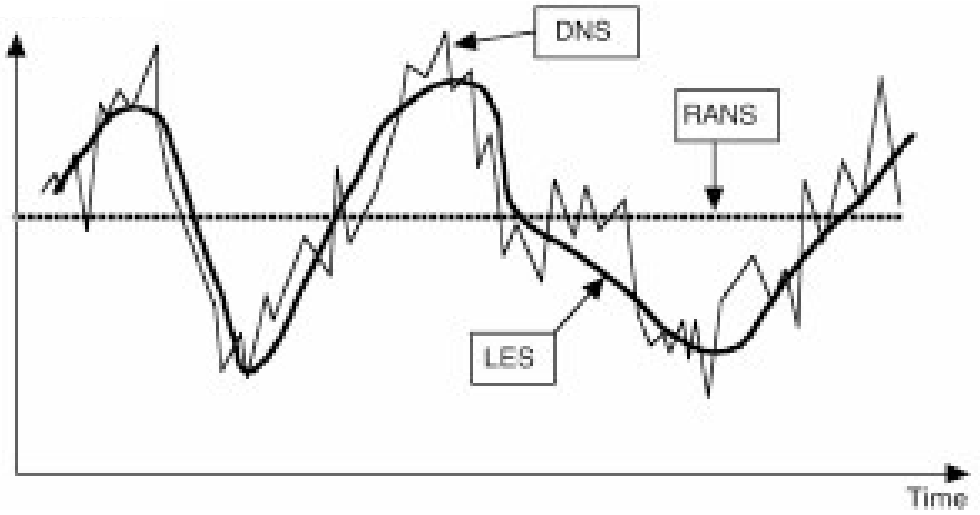
- **From RANS to DNS**
  - Reynolds averaged Navier Stokes (RANS)
  - Large eddy simulation (LES)
  - Direct numerical simulation (DNS)



expensive, accurate



cheep, rough



(G. Staffelbach, 2008)

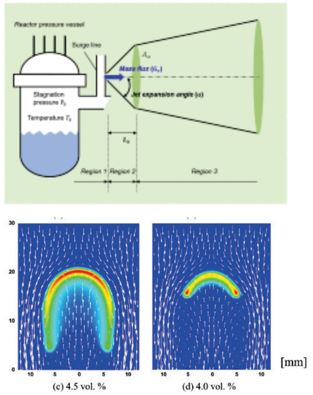
Achilles heel of CFD

① **Unrealistic computation costs** (especially for turbulent, reacting, multiphase flows)

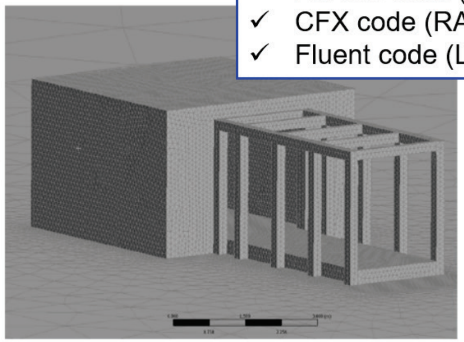
- my experiences…

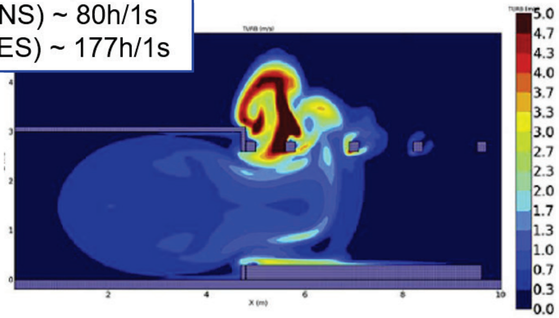
Hydrogen explosion simulation (**~100h/1s**)

- ✓ FLACS code (RANS) ~ 6h/1s
- ✓ CFX code (RANS) ~ 80h/1s
- ✓ Fluent code (LES) ~ 177h/1s



(Jeon et al., 2022)

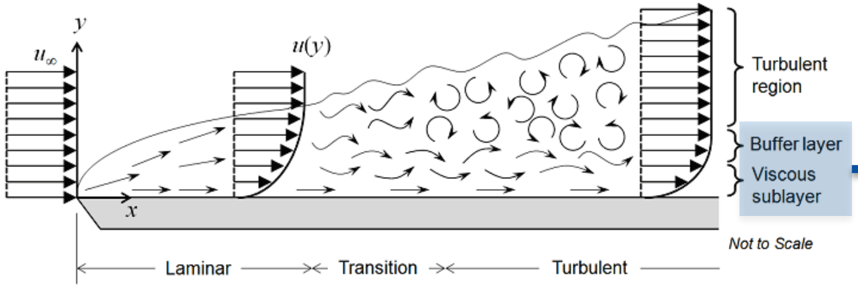




(Tolias et al., 2018)

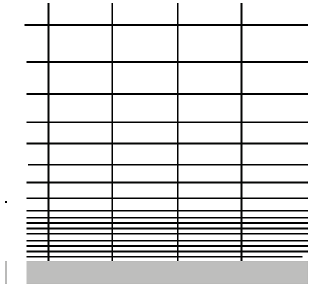
"Nuclear reactor severe accident simulation: 72 h"

- scaling studies for near-wall region

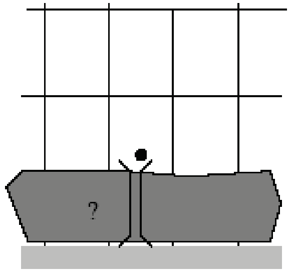


too expensive!

$\sim O(Re^{13/7})$



$\sim O(Re)$



Based on LES

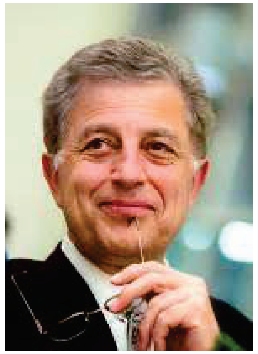
(S.T. Bose, 2019)

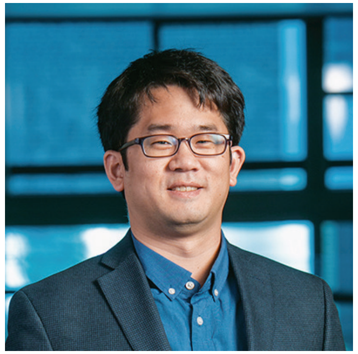
Achilles heel of CFD

"Wall-modeled large-eddy simulation for complex turbulent flows, Annual Review of Fluid Mechanics"

Prof. P. Moin

Prof. G.I. Park

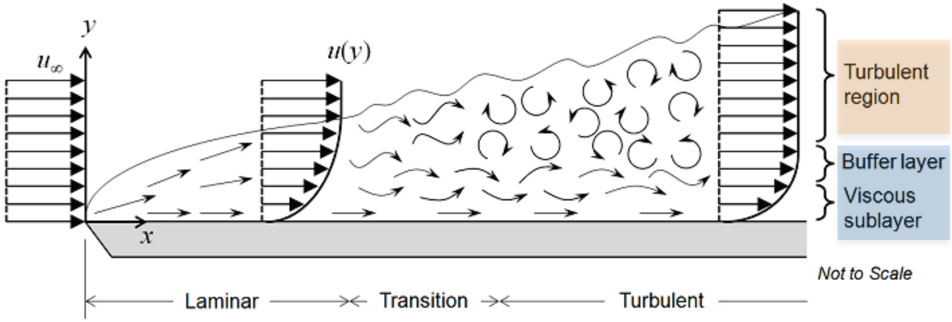




turbulent flame model, boiling heat transfer, etc.

② Turbulent models: a lot of progress, but still hungry

- With the current CPU performance, industrial application of DNS is 'usually' impossible.
- **Wall-modeled LES (WMLES)** is being considered as the next best option.
- However, sub-grid scale Reynolds stress model and turbulent wall model are still highly dependent on **empirical constants.**



Sub-grid scale viscosity model

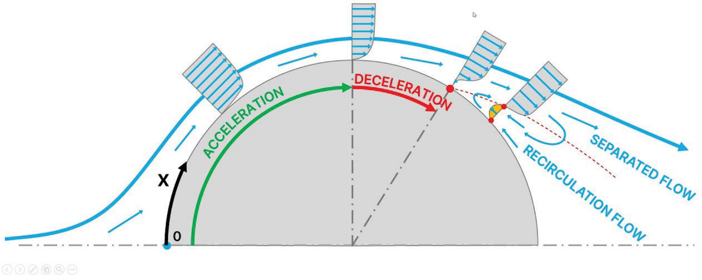
$$\nu_{sgs} = (C_s \Delta)^2 \sqrt{2 S_{ij} S_{ij}}$$

Turbulent wall model

$$u^+ = \frac{1}{k} \ln y^+ + B$$

Empirical constants

Unreliable accuracy for complex geometry!



Contents

① Background and objective

② Achilles heel of CFD

③ Can AI improve CFD?

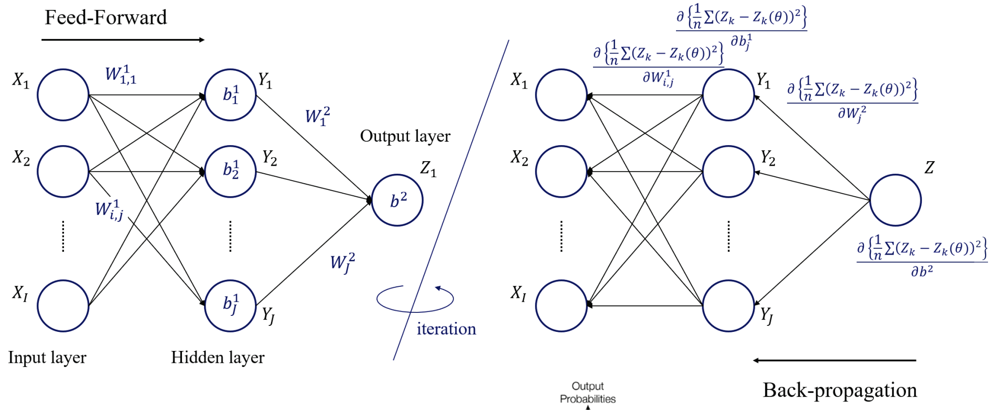
- Part 1: acceleration (supervised/unsupervised learning)

- Part 2: accuracy (reinforcement learning)

④ Summary and conclusions

Neural networks

- The deep neural network algorithms were inspired by biological neural network .
- Below figure shows **the feed-forward algorithms in two-layer network model.** :I-dimensional input matrix and J unit number of a hidden layer
- The back-propagation allows to optimize parameter values.
  - Eq. (4) shows the representative loss function (mean square error)



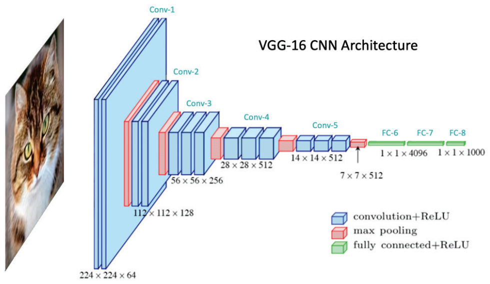
$$Y_j = \sum_i W_{i,j}^1 X_i + b_j^1 \quad (1)$$

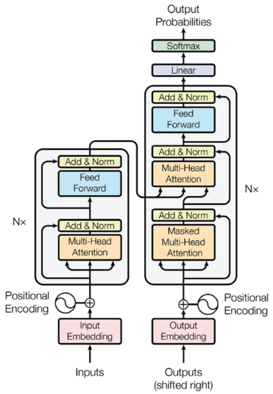
$$Z = \sum_j W_j^2 Y_j + b^2 = \sum_j \left(W_j^2 \left(\sum_i W_{i,j}^1 X_i + b_j^1 \right) \right) + b^2 \quad (2)$$

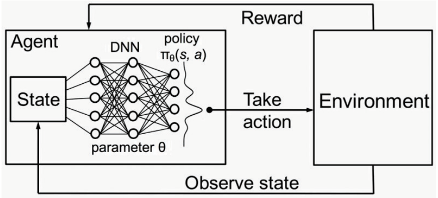
$$Z = \sum_j \left(W_j^2 \cdot \mathrm{relu}\left(Y_j\right) \right) + b^2 = \sum_j \left(W_j^2 \cdot \mathrm{relu}\left(\sum_i W_{i,j}^1 X_i + b_j^1 \right) \right) + b^2 \quad (3)$$

$$J(\theta) = \frac{1}{n} \sum_{k=1}^{n} \left(Z^k - Z^k(\theta) \right)^2 \quad (4)$$

$$\frac{\partial J}{\partial W^1} = \left(\frac{\partial J}{\partial Y} \cdot \frac{\partial Y}{\partial W^1} \right)^{\mathsf{T}} = \left(\frac{\partial J}{\partial Y} \cdot X^{\mathsf{T}} \right)^{\mathsf{T}}$$
$$= \left(W^2 \cdot \left((W^2)^{\mathsf{T}} \cdot \mathrm{relu}\left((W^1)^{\mathsf{T}} X + b^1 \right) + b^2 - Z \right) \cdot X^{\mathsf{T}} \right)^{\mathsf{T}} \quad (5)$$





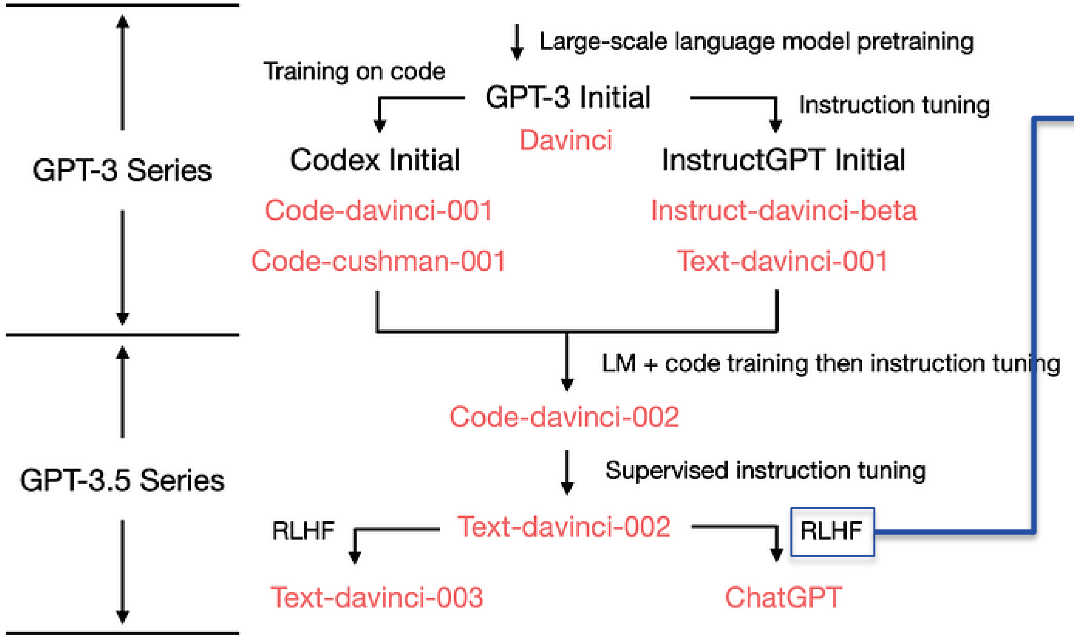


Neural networks - domain knowledge

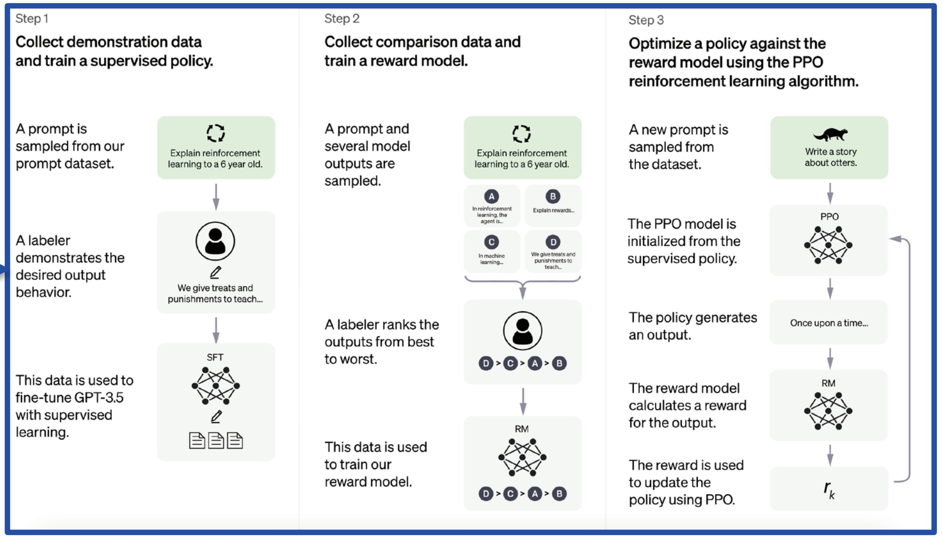
- Paradoxically, the success of ChatGPT highlighted the importance of domain knowledge.

(source: OpenAI)

※Reinforcement learning from human feedback (RLHF)



(source: Matt Richard's blog)



1) Supervised fine tuning (**human labeler**)
2) Reward modeled (**human evaluator**)
3) Reinforcement learning with the reward model

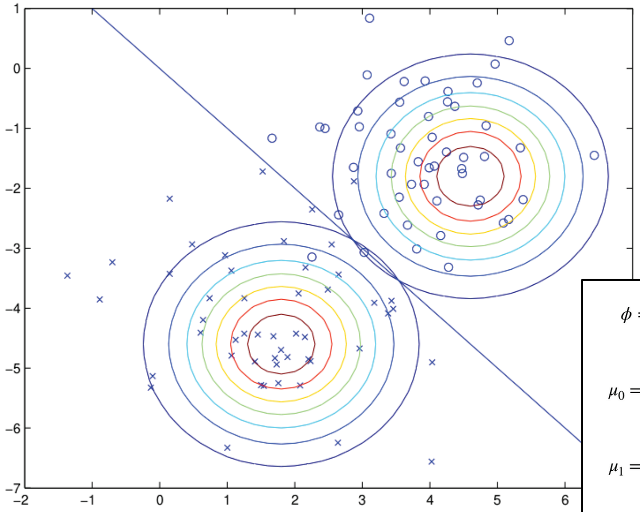
"Insert domain knowledge in AI models"

Neural networks - domain knowledge

- In fact, we can see the power of domain knowledge even from machine learning fundamentals

Gaussian discriminant analysis (GDA)

logistic regression analysis



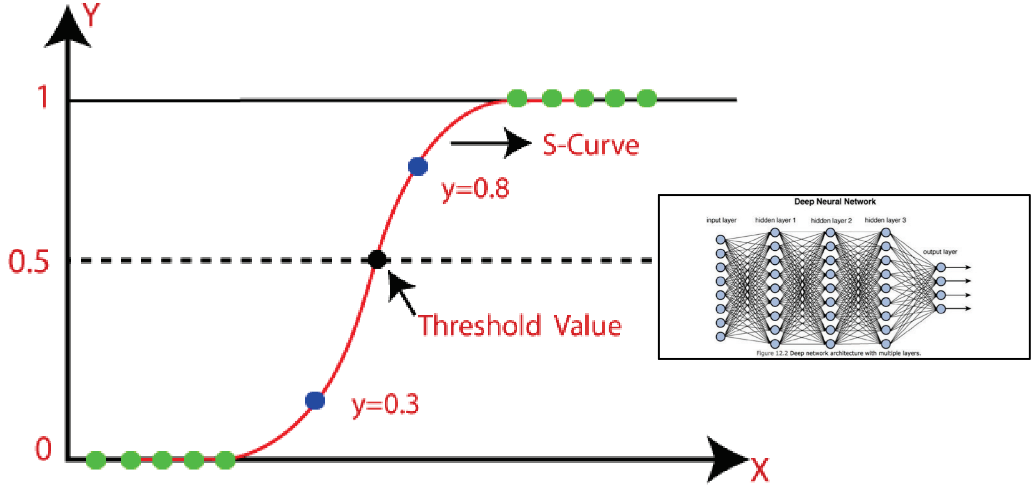
$$\phi = \frac{1}{m}\sum_{i=1}^{m} 1\{y^{(i)} = 1\}$$

$$\mu_0 = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}}$$

$$\mu_1 = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}$$

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m} (x^{(i)} - \mu_k)(x^{(i)} - \mu_k)^T \quad \text{where } k = 1\{y^{(i)} = 1\}$$

required data amount, parameters



If data is expensive and special: CFD

If data is rich and general: NLP

Domain knowledged AI

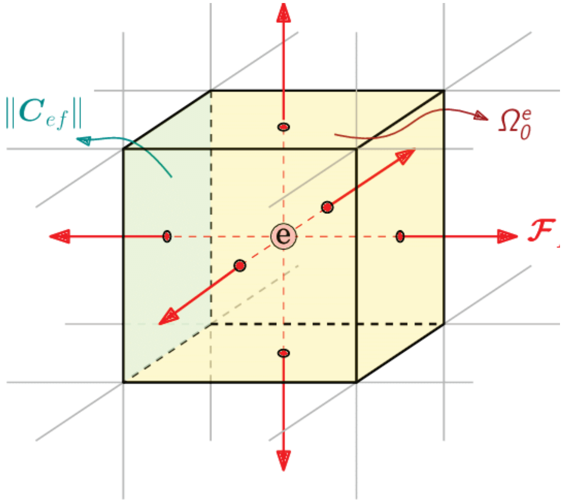
general AI

Commonality

Specialized

Neural networks - domain knowledge

- Principles of the finite volume method (FVM)
  - It is important to thoroughly understand the principles of CFD simulation.
  - General transport equation can be expressed by Eq. (1) with the discretized control volumes.
  - **The basic idea of the FVM** is the divergence terms can be converted to surface integrals (Eq. (3))
    : by Gauss's theorem (Eq. (2)).
  - **The quantity of the neighboring grid as well as the main grid** determines the next timestep field
  - We paid attention to these principles of the FVM, the **tier (stencil) /derivative** system.



$$\frac{\partial}{\partial t}\int_V \rho\phi\, dV + \int_V \nabla\cdot(\rho u\phi)\, dV - \int_V \nabla\cdot\left(\rho\Gamma_\phi\nabla\phi\right) dV = \int_V S_\phi(\phi)\, dV \qquad (1)$$

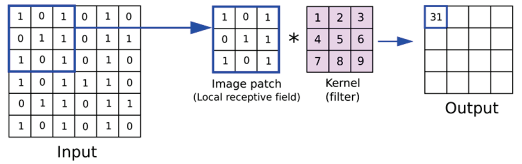
$$\int_V \nabla\cdot a\, dV = \oint_A dA\cdot a \qquad (2)$$

$$\frac{\partial}{\partial t}\int_V \rho\phi\, dV + \oint_A(\rho u\phi)\, dA - \oint_A\left(\rho\Gamma_\phi\nabla\phi\right) dA = \int_V S_\phi(\phi)\, dV \qquad (3)$$

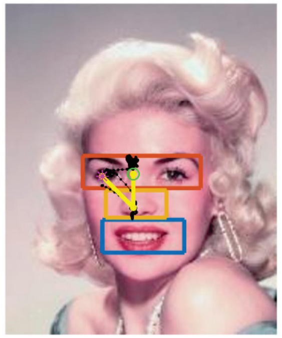
$$\frac{\partial\rho}{\partial t} + \frac{\partial}{\partial x}(\rho v_x) + \frac{\partial}{\partial y}\left(\rho v_y\right) = 0 \qquad (4)$$

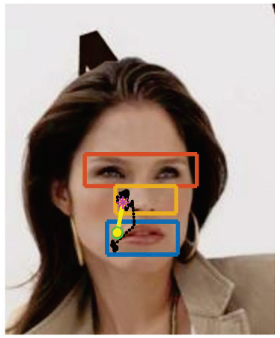
Can AI improve CFD? - Part 1. acceleration

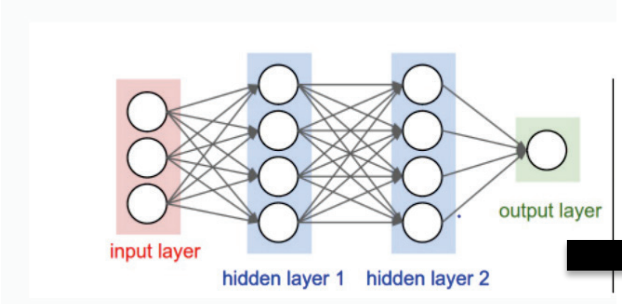
For best performance, we should develop a CFD fitted-network model!



- Idea of CNN: image has the stationarity of statistic

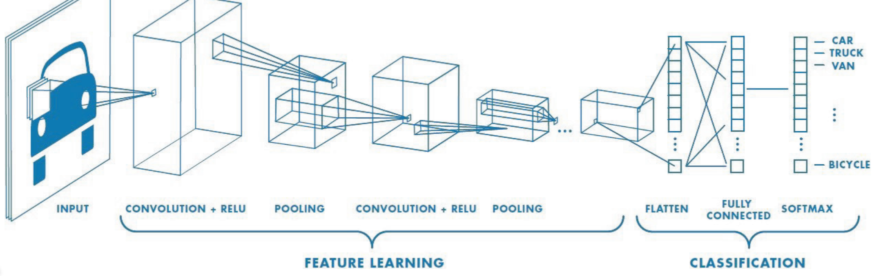






MLP

Convolutional Neural Network

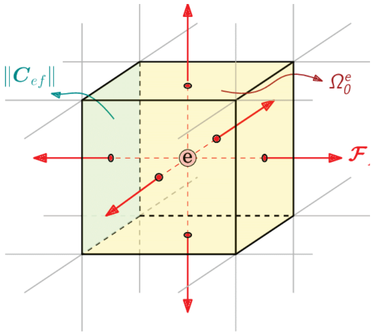


CNN: 1 image = 1 dataset
Our: 1 grid = 1 dataset

100,000 images
vs 10 images

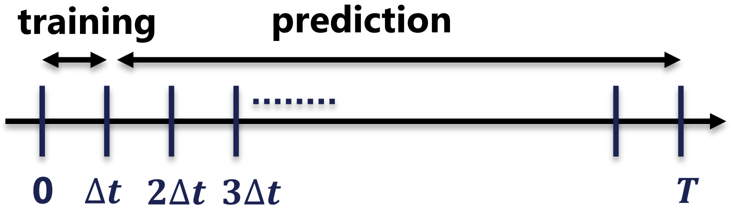
CFD acceleration!!

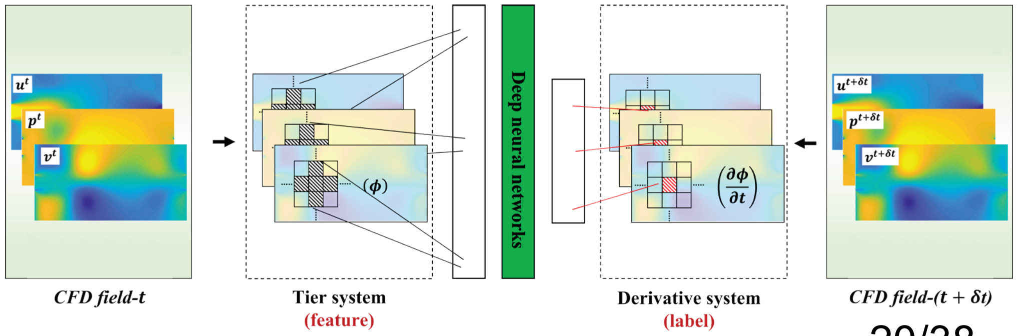
- Idea of our network model: all CFD nodes has the same rules



All nodes must be satisfied with near nodes:

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

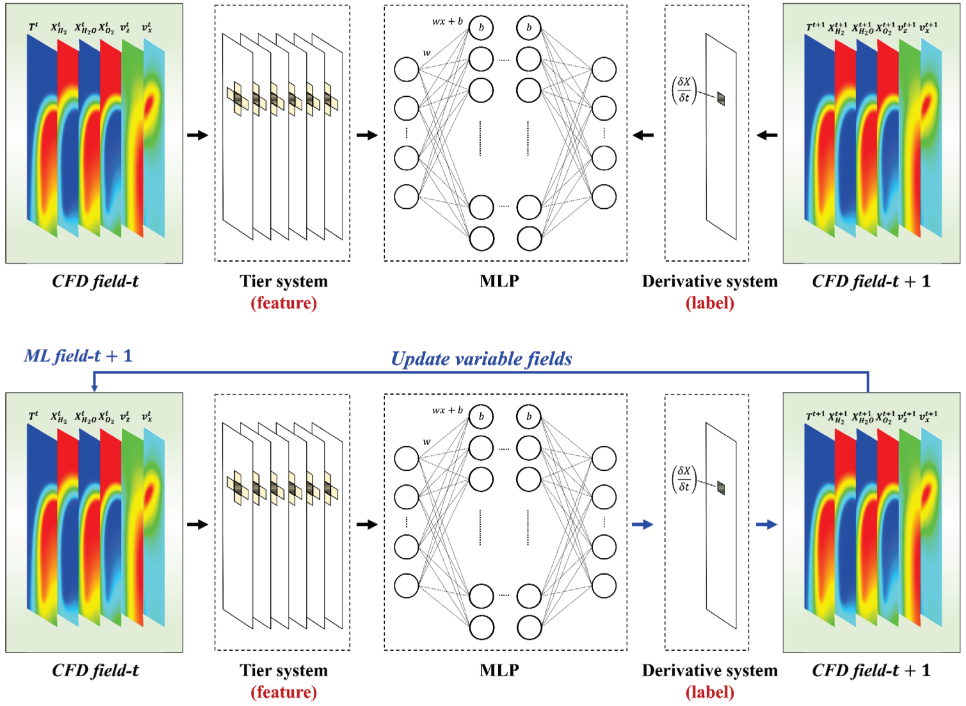


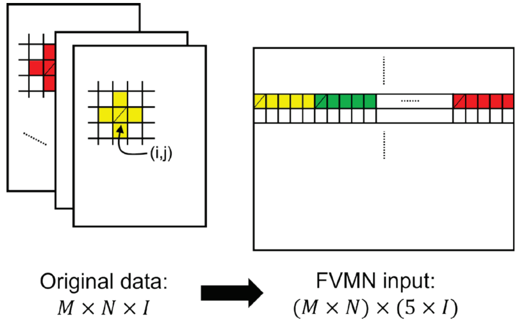


(Jeon, 2022)

Can AI improve CFD? - Part 1. acceleration

- novel concept of network model: **FVMN**





(Jeon, 2022)

- In this study, **a finite volume method network (FVMN) was proposed** considering the principles of the FVM in the network input/output system.
- Although the tier system have been already suggested by previous studies, we additionally design the derivative system.

$$X^t = \left[x_{i,j}^t\right] \text{ where } X^t \in R \tag{1}$$

General model

$$Z^t = \left[x_{i,j}^{t+1}\right] \text{ where } X^{t+1} \in R \tag{2}$$

FVMN model

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

$$X_t^t = \left[x_{i,j}^t, x_{i-1,j}^t, x_{i+1,j}^t, x_{i,j-1}^t, x_{i,j+1}^t\right]^{\mathsf{T}} \text{ where } X_t^t \in R^5 \tag{3}$$

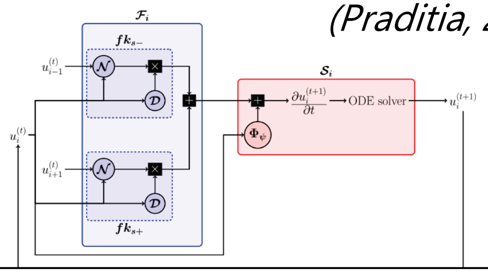
$$Z_d^t = \left[\left(\frac{\delta x}{\delta t}\right)_{i,j}^{t+1}\right] \text{ where } Z_d^t \in R \tag{4}$$

- CNN: $\#\ of\ training\ samples = \#\ of\ CFD\ snapshots$ (**image-based**)
- $FVMN$: $\#\ of\ training\ samples = \#\ of\ CFD\ grids$ (**grid based**)

Can AI improve CFD? - Part 1. acceleration

Physics-informed neural networks (PINNs)

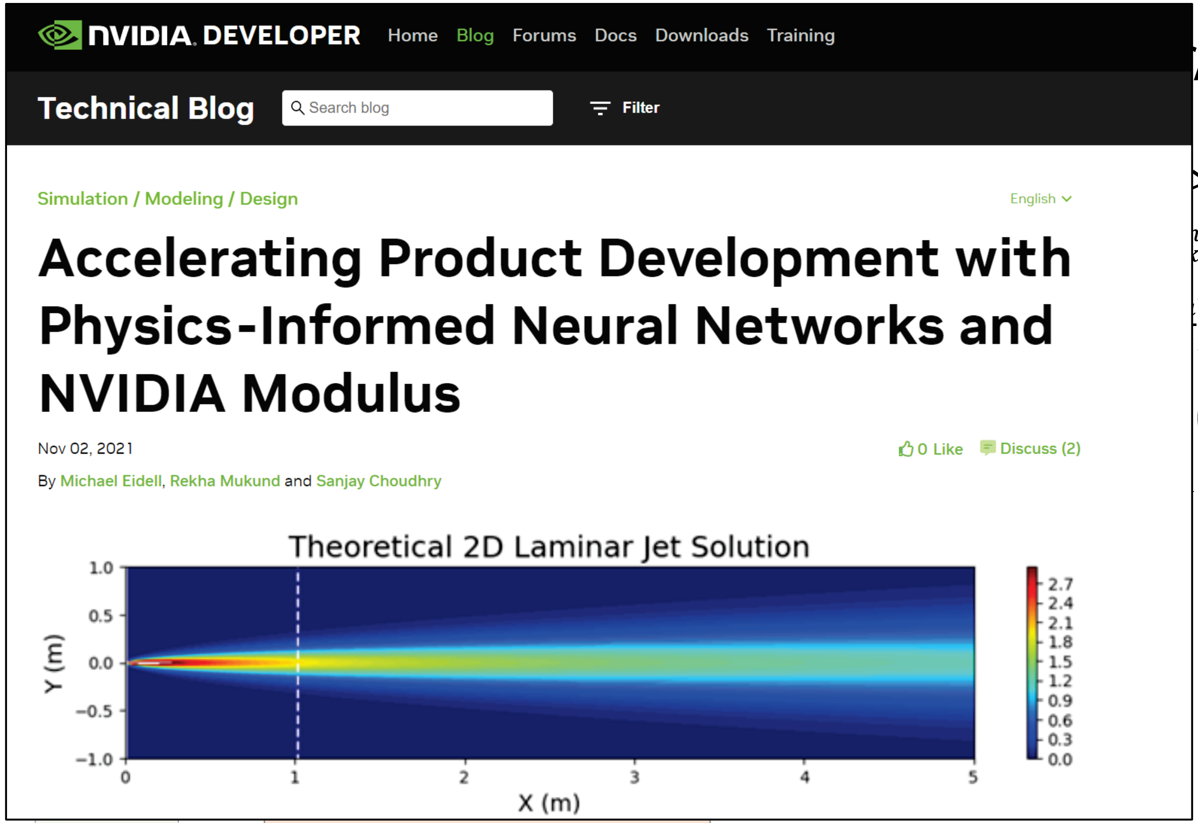
(Praditia, 2021)



What is PINNs?

(a)

(b)


NVIDIA DEVELOPER Home Blog Forums Docs Downloads Training

Technical Blog Search blog Filter

Simulation / Modeling / Design English

Accelerating Product Development with Physics-Informed Neural Networks and NVIDIA Modulus

Nov 02, 2021 0 Like Discuss (2)

By Michael Eidell, Rekha Mukund and Sanjay Choudhry


SAMSUNG SDS 클라우드 물류 | 업종 무료체험

인사이트 | 인사이트 리포트 클라우드 기술 백서 고객사례 리소스 동영상

인공지능

AI 기반 물리 정보 신경망을 이용한 시뮬레이션

2023-01-18 | 문기효

인쇄하기

이번 글에서는 시뮬레이션에서 최근 화두가 되고 있는 물리 정보 신경망(Physics-Informed Neural Network, 이하 PINN)에 대해서 다루어 보겠습니다. 여기서 시뮬레이션이라는 것은 유체역학, 전자기학 등 공학 문제를 슈퍼컴퓨터를 이용한 수학적인 방법으로 푸는 것을 말합니다. 자동차 형상, 휴대폰 열설계, 반도체 소자 시뮬레이션 등이 있습니다.

+ 풀이과정
+ 풀이과정
+ 풀이과정

*CFD field-(t + 1)**

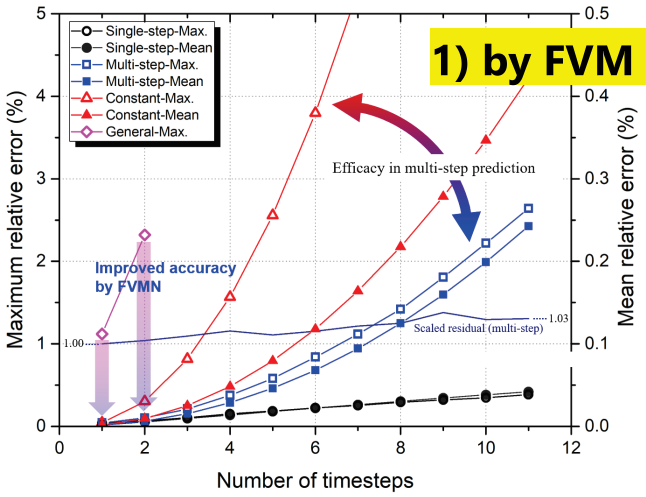
Update per mini-batch

(Jeon, 2022)

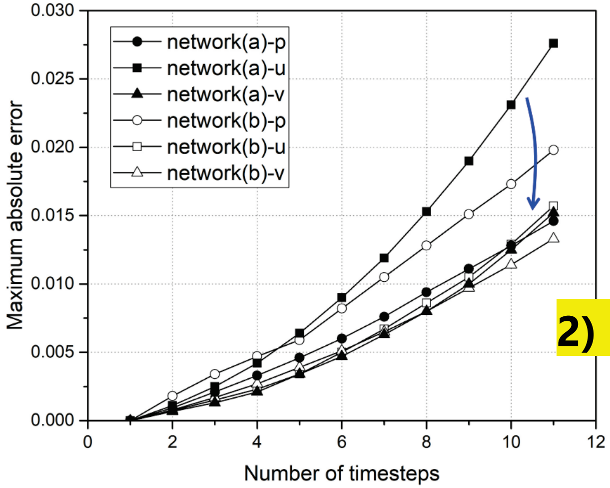
1. Improve synchronization of FVM method and NNs
2. Prevention "non-physical overfitting"

Can AI improve CFD? - Part 1. acceleration

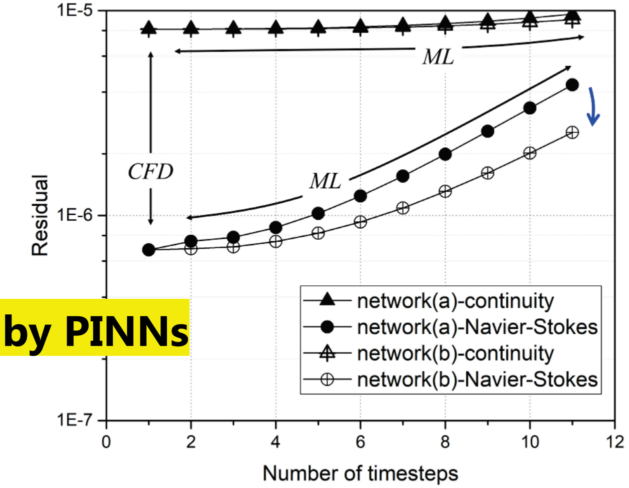
- Improved performance of FVMN



1) by FVM architecture



2) by PINNs



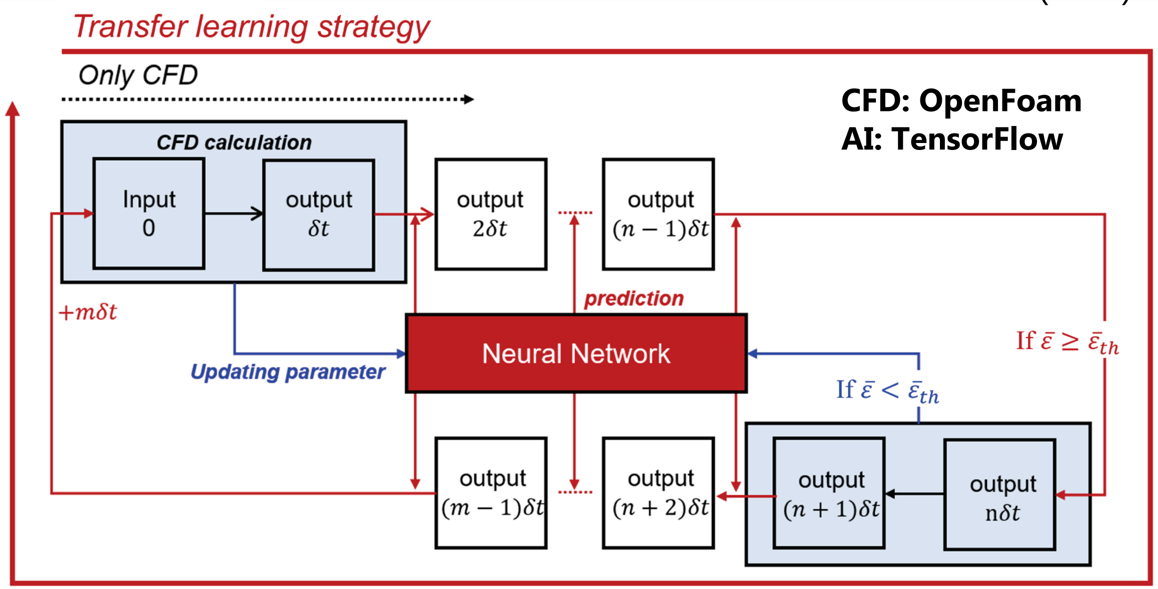
- Improved network performance
- **Reduced residuals** in prediction time series
- Still error growing...

Can AI improve CFD? - Part 1. acceleration

Mr. J. Lee (HYU)

- Physics-informed transfer learning

Car: AI
Repairman: CFD

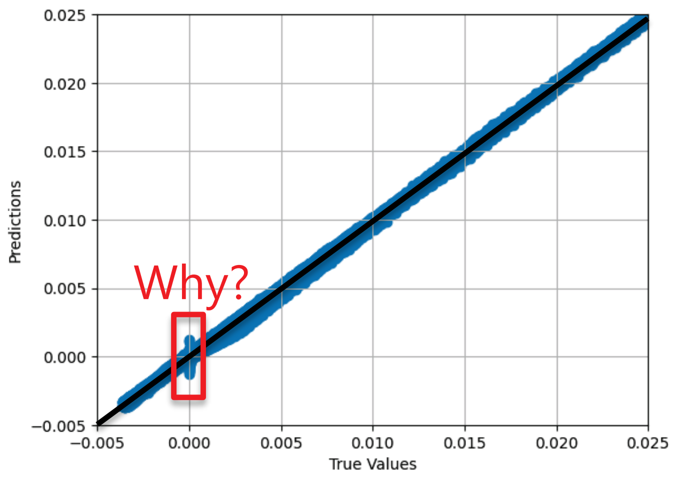


$$\frac{\mathrm{d}(\mathbf{u})}{\mathrm{d}t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \nabla \cdot (\nu \nabla \mathbf{u}) + \nabla p = \boldsymbol{\varepsilon}$$

(Jeon, 2022)

Can AI improve CFD? - Part 1. acceleration

- **The boundary layer issue** is also the reason why the transfer strategy is needed.

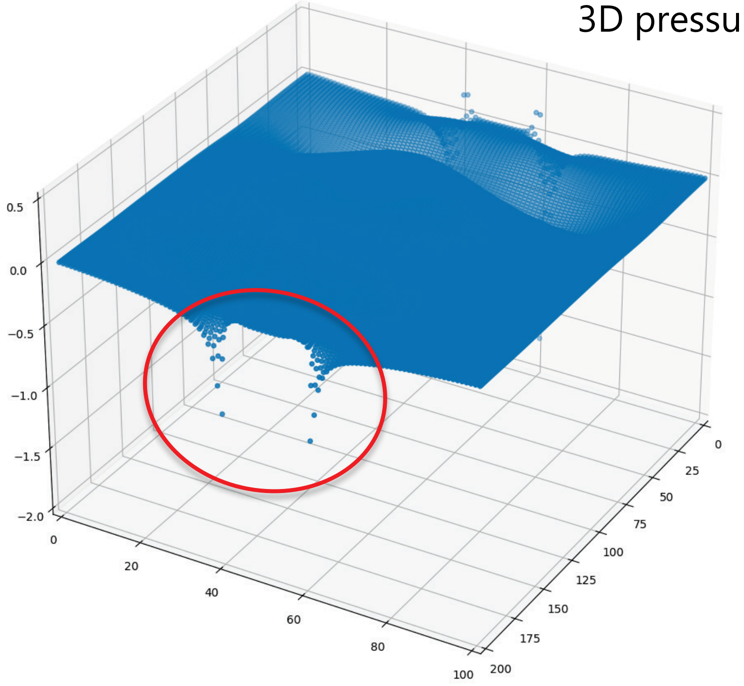


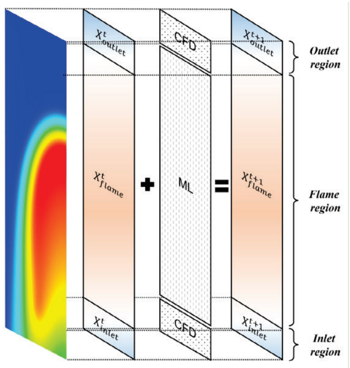
Two solutions

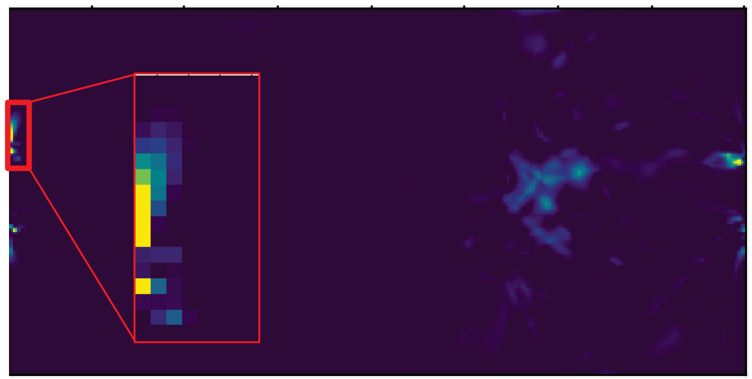
1) Large AI model (~175 billion parameter numbers, now x 10^7)

2) Physics-informed transfer learning

3D pressure contour



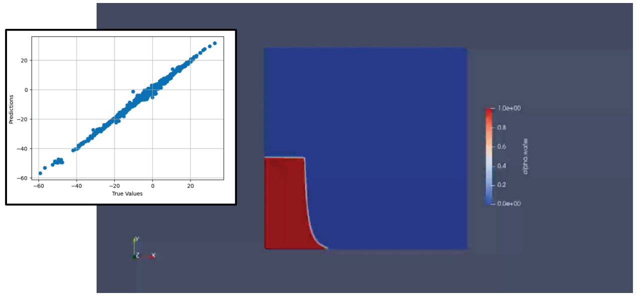




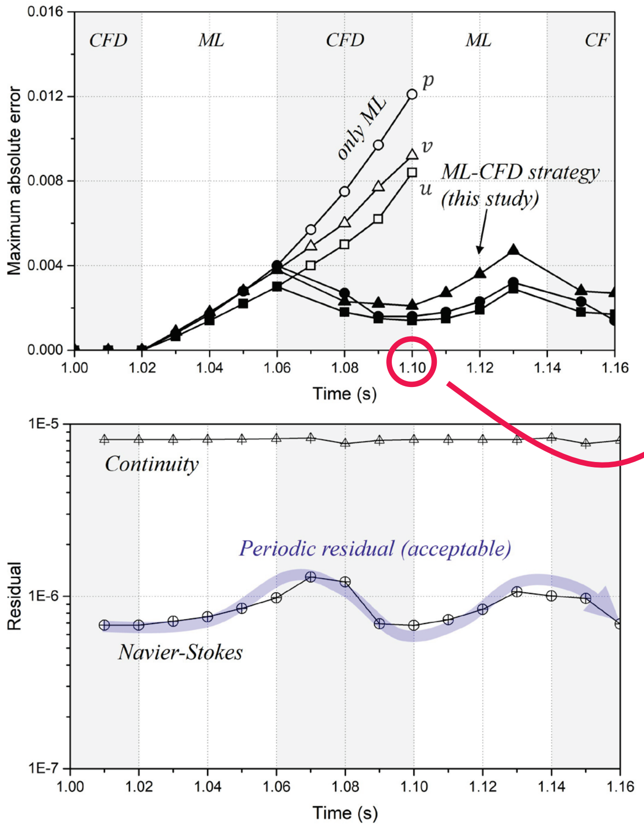
2D error contour

Can AI improve CFD? - Part 1. acceleration

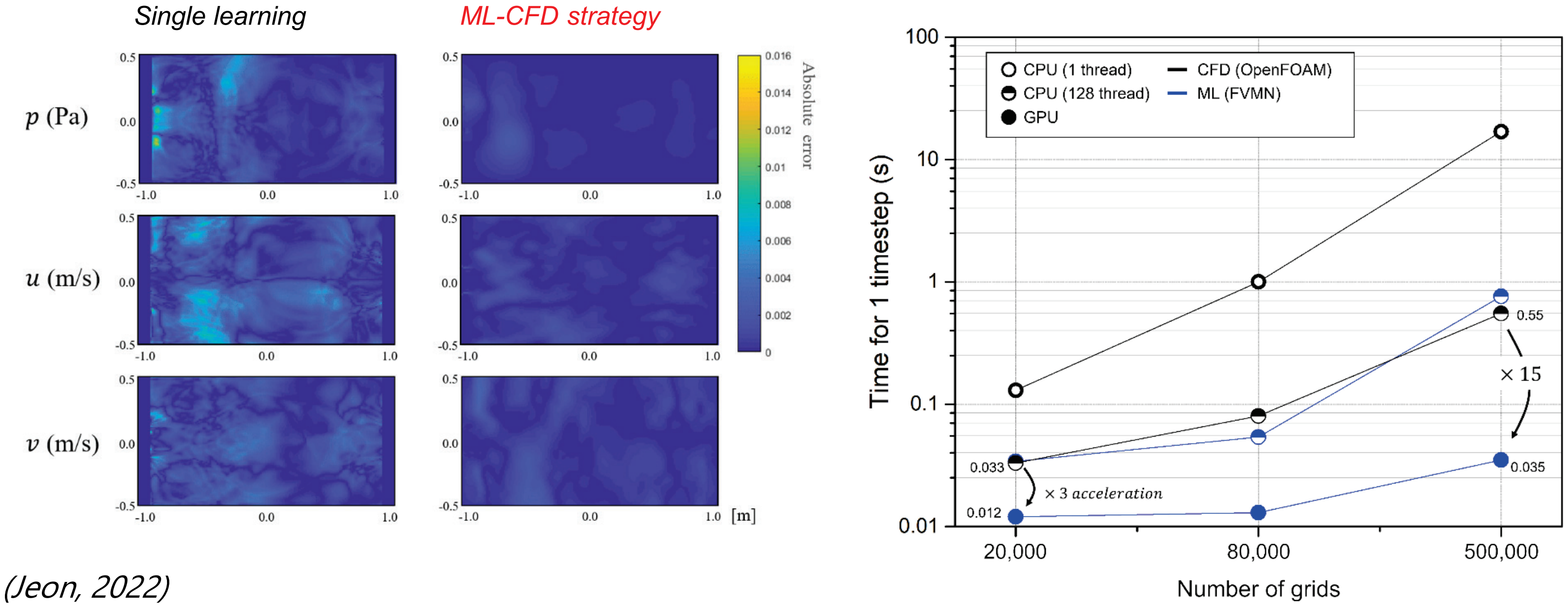
Multiphase flow (ongoing)



- Feasibility study of the physics-informed transfer learning strategy



- **Error recovery** was observed in the CFD zone.
- Effect by **correction of flux balances** (residuals)



(Jeon, 2022)

Contents

① Background and objective

② Achilles heel of CFD

③ What is the role of AI in CFD?

- Part 1: acceleration (supervised/unsupervised learning)

- Part 2: accuracy (reinforcement learning)

④ Summary and conclusions

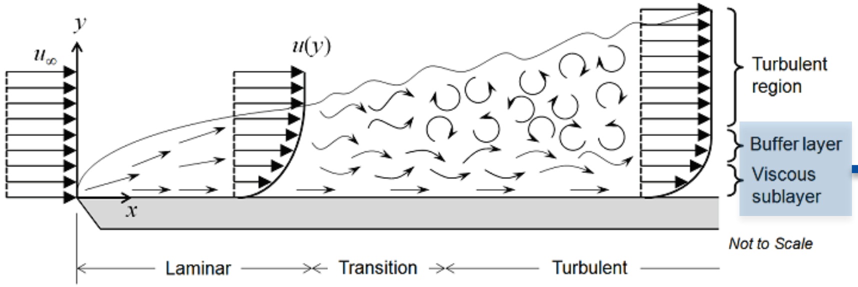
Can AI improve CFD? - Part 2. accuracy

- **Background**
- Why we need Large eddy simulation (LES)?

- The NASA 2030 CFD Vision Report demonstrated **deficiency of lower fidelity (RANS-based) solution** approaches and suggests that unsteady simulation techniques such as **LES may provide sufficient accuracy.**
- Especially, in nuclear field, RANS-based solution is not suitable for analyzing complex flow in accident conditions.

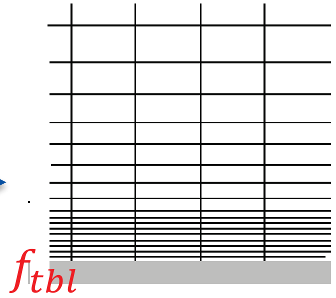
- Why we need wall modeled LES (WMLES)?

- In high Reynolds numbers, the grid requirements for wall resolved large eddy simulations are not feasible.
- Grid requirement = computational cost

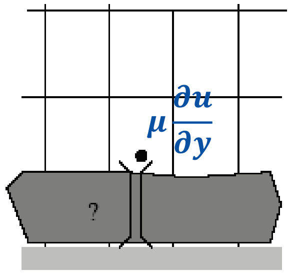


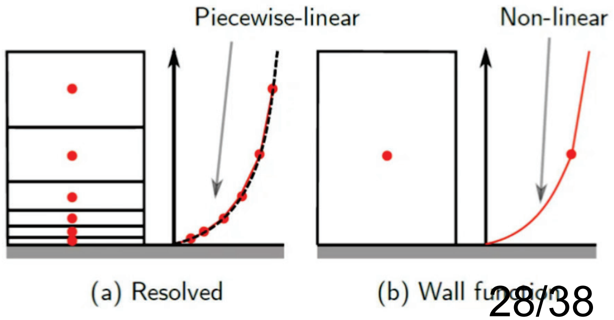
too expensive!

$\sim O(Re^{13/7})$

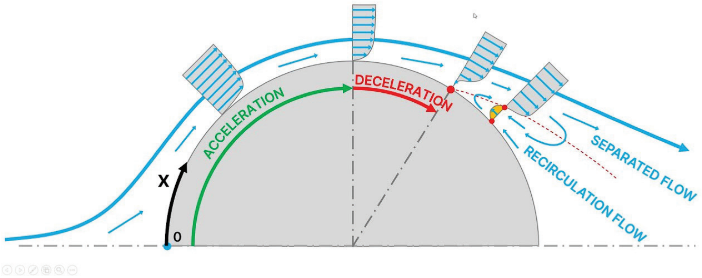


$\sim O(Re)$





Can AI improve CFD? - Part 2. accuracy



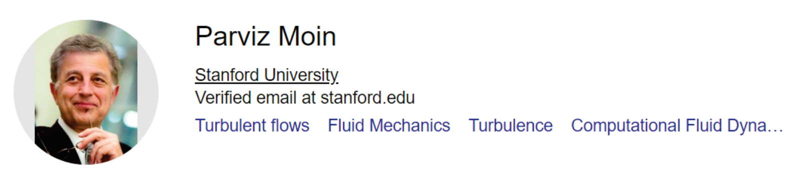
- **Background**
- Historical attempts in WMLES

(1) Algebraic wall model: effective in the absence of pressure gradient conditions.

$$u^+ = \frac{1}{\kappa}\log y^+ + B$$

$$u^+ = \frac{1}{0.41}\log(1 + 0.4y^+) + 7.8\left(1 - e^{-\frac{y^+}{11}} - \frac{y^+}{11}e^{-0.33y^+}\right)$$

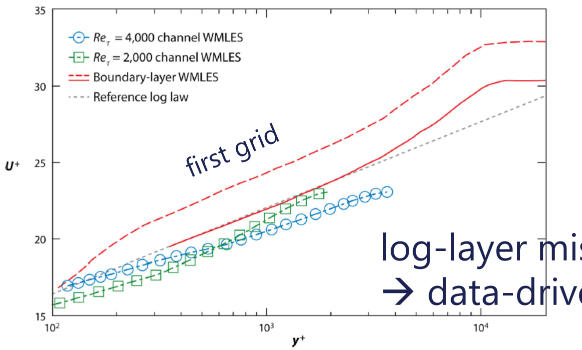
Many studies on the non-equilibrium wall model are still in progress!



(2) Thin boundary-layer equation (TBLE)

$$\frac{\partial \tilde{u}_i}{\partial t} + \frac{\partial \tilde{u}_i \tilde{u}_j}{\partial x_j} + \frac{1}{\rho}\frac{\partial \tilde{p}}{\partial x_i} = \frac{\partial}{\partial y}\left[(\nu + \tilde{\nu}_t)\frac{\partial \tilde{u}_i}{\partial y}\right]$$

→ with equilibrium assumption*: $\frac{d\tau}{dn} = \frac{d}{dn}\left(\mu\frac{dU}{dn} - \overline{\rho u'v'}\right)$



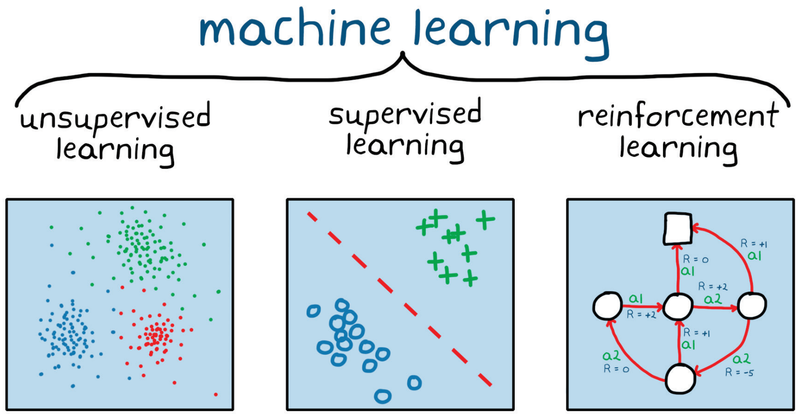
log-layer mismatch
→ data-driven method

* assuming a local balance between the pressure gradient/streamwise convection and that the spatiotemporal resolution of the LES is large compared with viscous length and time scales such that the near-wall cell

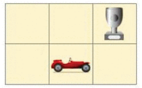
Can AI improve CFD? - Part 2. accuracy

- Reinforcement learning

"Reinforcement learning differs from **supervised learning** in a way that in supervised learning the training data has the answer key with it so the model is trained with the **correct answer itself** whereas in **reinforcement learning**, **there is no answer** but the reinforcement agent decides what to do to perform the given task."



Game Board:



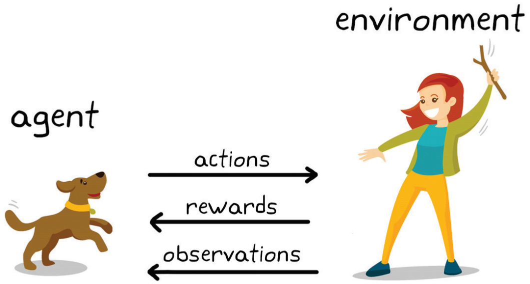
Current state (s): 000 010

Q Table: γ = 0.95

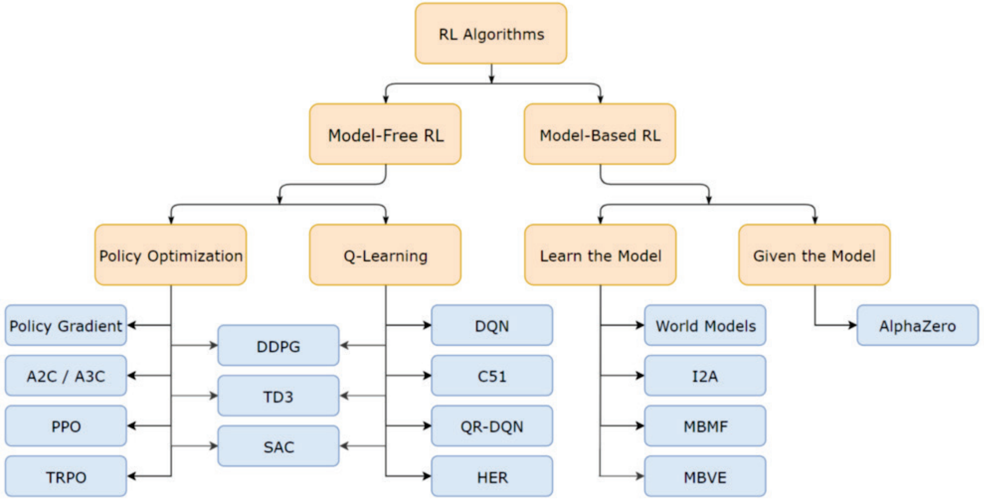
	000 100	000 010	000 001	100 000	010 000	001 000
⇧	0.2	0.3	1.0	-0.22	-0.3	0.0
⇩	-0.5	-0.4	-0.2	-0.04	-0.02	0.0
⇨	0.21	0.4	-0.3	0.5	1.0	0.0
⇦	-0.6	-0.1	-0.1	-0.31	-0.01	0.0

$$J(\theta) = \frac{1}{n}\sum_{k=1}^{n}\left(Z^k - Z^k(\theta)\right)^2$$

$$Q^{new}(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t + \gamma \max Q(s_{t+1}, a) - Q(s_t, a_t))$$

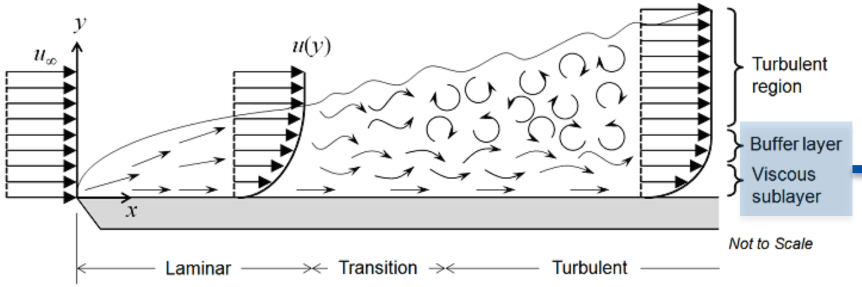


What is reinforcement learning?
by mathworks.com



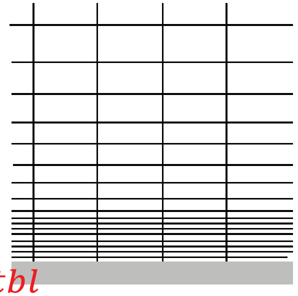
Can AI improve CFD? - Part 2. accuracy

- **Overview**



too expensive!

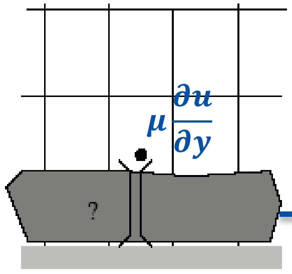
$\sim O(Re^{13/7})$



f_{tbl}

Wall resolved

$\sim O(Re)$

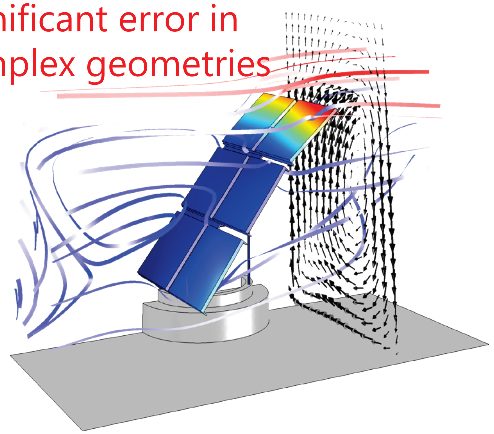


Wall modeled

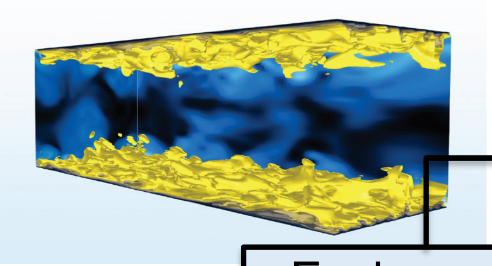
Algebraic wall model

$$u^+ = \frac{1}{\underline{k}} \ln y^+ + \underline{B}$$

Significant error in complex geometries



Idea: There is no ground truth for f_{wm}, **But we know $\mu \frac{\partial u}{\partial y}$!**



"Reinforcement learning"

State

Environment (CFD)

Reward

Agent

Action

$$\frac{d(u)}{dt} + \nabla \cdot (u \otimes u) - \nabla \cdot (\nu \nabla u) = -\nabla p + f_{wm}$$

State: flow variables: $u\,(x, z, t_n), \frac{\partial u}{\partial y}(x, z, t_n)$

Action: volumetric force term: $f_{wm}(x, z, t_n)$

Reward: shear stress: $\mu \frac{\partial u}{\partial y}(x, z, t_n)$

KTH : AI

BSC : CFD

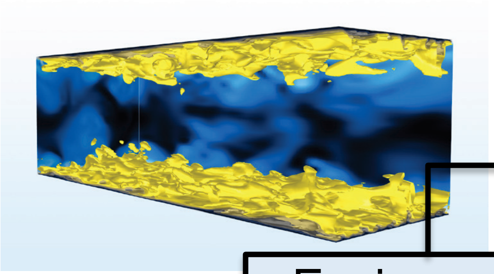
Can AI improve CFD? - Part 2. accuracy

- **Method: deep reinforcement learning**
- Why not supervised learning?

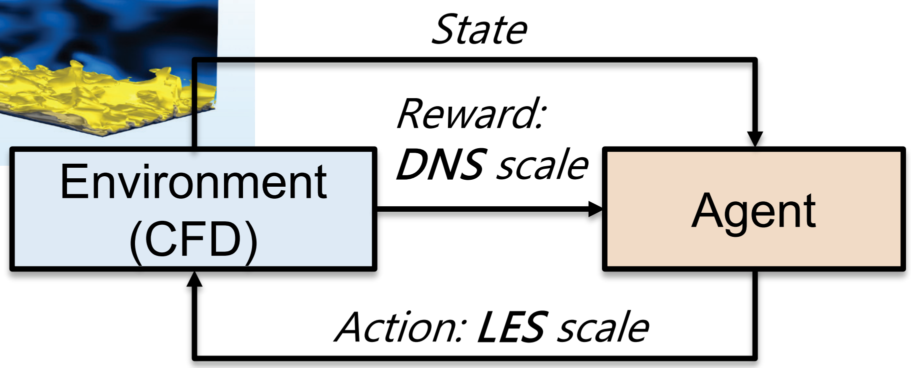
- For implicitly filtered large eddy simulation (LES), this approach is infeasible,… As a consequence, **the closure terms for implicitly filtered LES cannot be computed from high-fidelity DNS data**, since the filter that would have to be applied is unknown (*A. Beck, Int. J. Hear Fluid Flow, 2023*)

- How about reinforcement learning?

- Can avoid inconsistency by training **not on a previously obtained training dataset, but by deterministic reward function.**

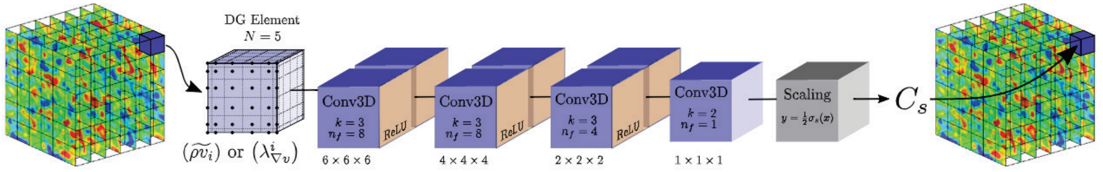


"Reinforcement learning"



Can AI improve CFD? - Part 2. accuracy

- **Method: deep reinforcement learning**
- A. Beck, University of Stuttgart, 2023
  - DRL for **dynamic** Smagorinsky's model parameter (PPO)

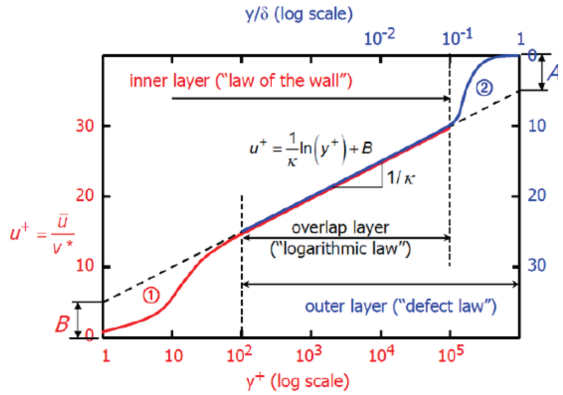


  - $\mu_t = \rho(\boldsymbol{C_s}\Delta)^2\sqrt{2\tilde{S}_{ij}\tilde{S}_{ij}},\ \tilde{S}_{ij} = \frac{1}{2}\left(\frac{\partial\tilde{v}_i}{\partial x_j} + \frac{\partial\tilde{v}_j}{\partial x_i}\right)$

Content	Function
Action	Smagorinsky's model parameter within $C_s \in [0, 0.5]$
Reward	$R(s) = 2\exp\left(-\frac{1}{\alpha k_{max}}\sum_{k=1}^{k_{max}}\left(\frac{\boldsymbol{\bar{E}_{DNS}(k) - E_{LES}(k)}}{\boldsymbol{\bar{E}_{DNS}(k)}}\right)^2\right) - 1$

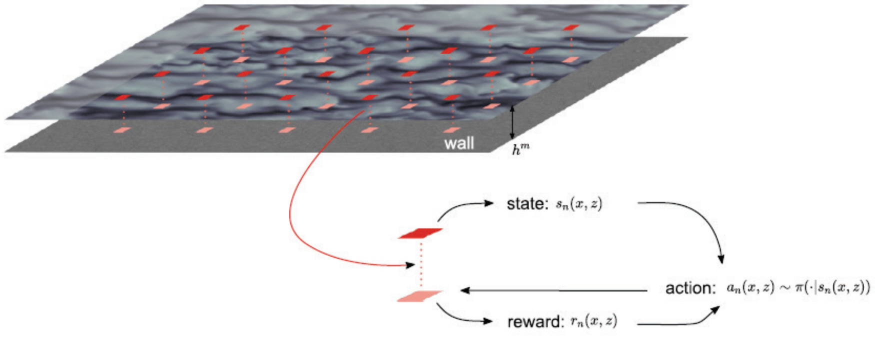
- If a similar method is used for turbulent wall modeling,

Content	Function
Action	Wall model parameter: $u^+ = \frac{1}{k}\ln y^+ + B$
Reward	Turbulent energy spectrum, wall shear stress, etc.

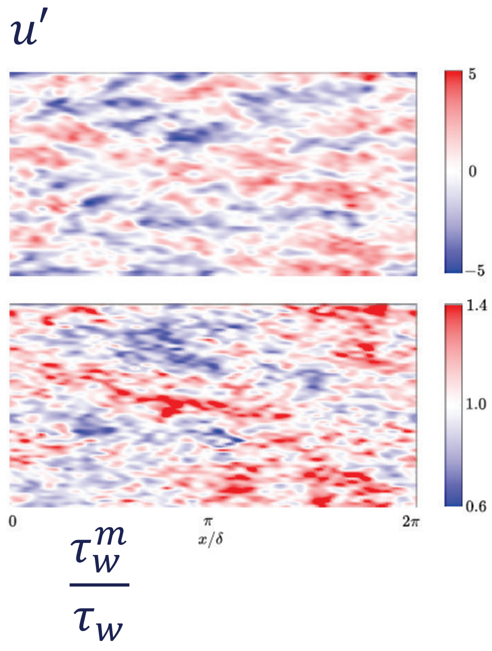


Can AI improve CFD? - Part 2. accuracy

- **Method: deep reinforcement learning**
- H. Jane Bae, California Institute of Technology 2022
  - DRL for multiplication factor of wall shear stress
  - $\tau_w^m(x,z,t_{n+1}) = a_n(x,z)\tau_w^m(x,z,t_n)$
  - The initial wall-shear stress is set to $\pm$ 20% of correct wall-shear stress



Content	Function
Action	Multiplication factor of wall shear stress $a_n(x,z) \in [0.9, 1.1]$
State	$u^*(x,h^m,z,t_n), \frac{\partial u^*}{\partial y^*}(x,h^m,z,t_n), y^* = (h^m)^*$
Reward	$\mathrm{r}_n(x,z) = \left(\frac{\lvert\tau_w - \tau_w^m(x,z,t_n)\rvert - \lvert\tau_w - \tau_w^m(x,z,t_{n-1})\rvert}{\tau_w}\right) + \mathbb{I}\left(\frac{\lvert\tau_w - \tau_w^m(x,z,t_n)\rvert}{\tau_w} < 0.01\right)$



Can AI improve CFD? - Part 2. accuracy

- **Method: deep reinforcement learning**
- Limitations of recent studies by A. Beck and H. Jane Bae

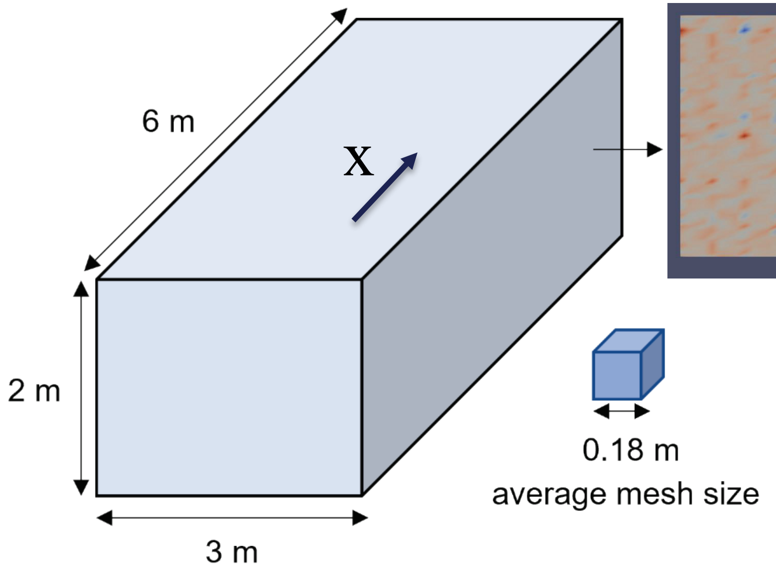
  (1) Dependence on empirical formula

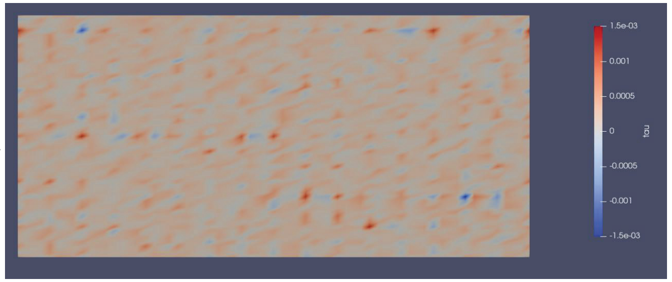
  (2) Non-physical meaning of action variable: multiplication, parameter sensitivity
- Our idea: DRL for volumetric momentum source

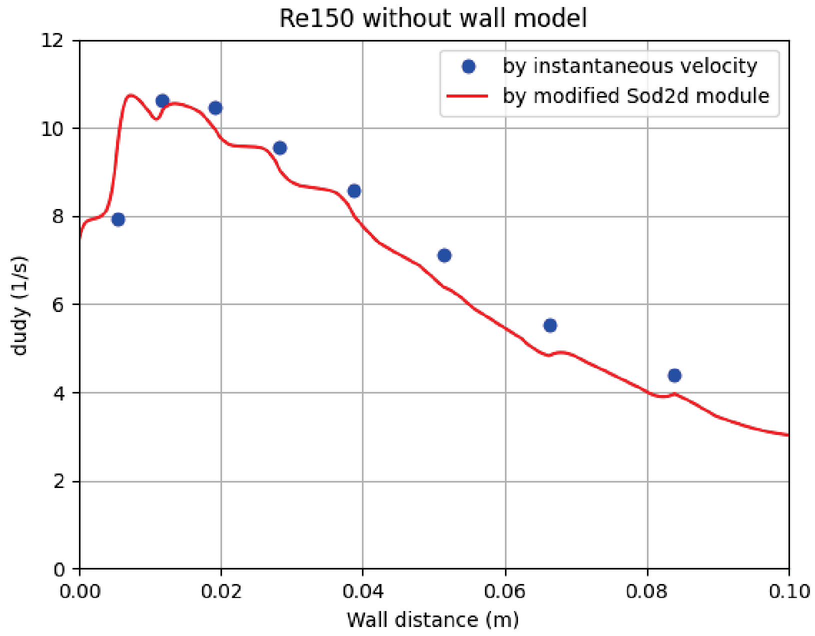
Content	Function
Action (example)	$\frac{\mathrm{d(u)}}{\mathrm{d}t} + \nabla \cdot (\mathrm{u} \otimes \mathrm{u}) - \nabla \cdot (\nu \nabla \mathrm{u}) = -\nabla p + \boldsymbol{f}_{wm}$
State (example)	$u^*(x, h^m, z, t_n), \frac{\partial u^*}{\partial y^*}(x, h^m, z, t_n), y^* = (h^m)^*$
Reward (example)	(1) $R = R\left(\sum_{h=1}^{H}\left(\frac{\boldsymbol{u}_{LES}^{+}(x,z,t_n)-\boldsymbol{u}_{ref}^{+}(x,z,t_n)}{\boldsymbol{u}_{ref}^{+}(x,z,t_n)}\right)^2\right)$ (2) $R = R\left(\frac{\lvert\tau_W^{LES}(x,z,t_n)-\tau_W^{DNS}(x,z,t_n)\rvert}{\tau_W^{DNS}(x,z,t_n)}\right)$

Can AI improve CFD? - Part 2. accuracy

- **Now developing DRL framework**







Parameter	Value
Geometry	2×3×6
Friction Reynolds number (Re_τ)	100 (diverged at 950)
Average mesh size	18 × 18 × 18 cm
Wall model	Reichardt's wall model
Number of mesh	6,656
Initial timestep	~1e-4 (adaptive with Coruant)
LES model	Vreman SGS model

: DRL-WMLES project

: Reference DRL code

: CFD source code

Contents

① Background and objective

② Achilles heel of CFD

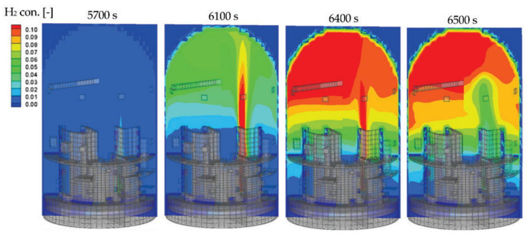
③ What is the role of AI in CFD?

- Part 1: acceleration (supervised/unsupervised learning)

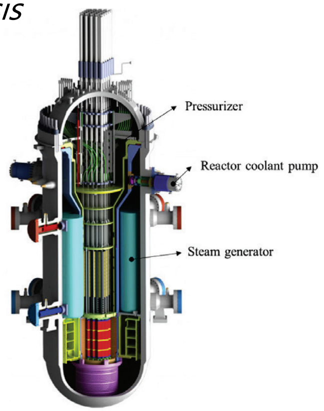
- Part 2: accuracy (reinforcement learning)

④ Summary and conclusions

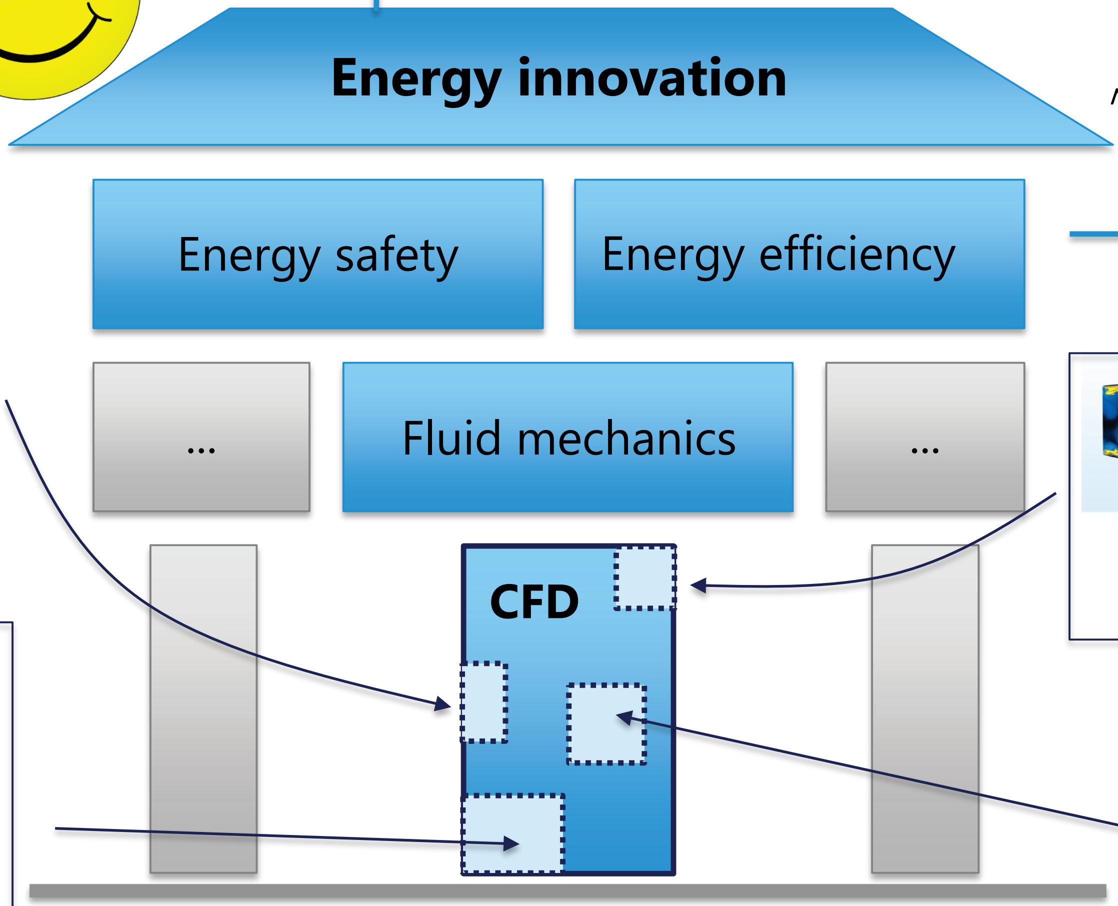
Summary and conclusion

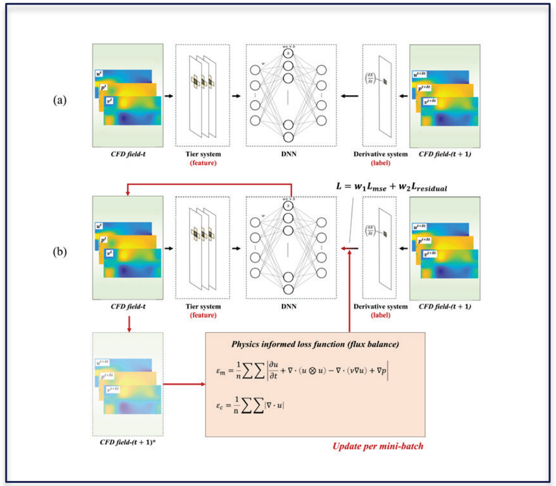


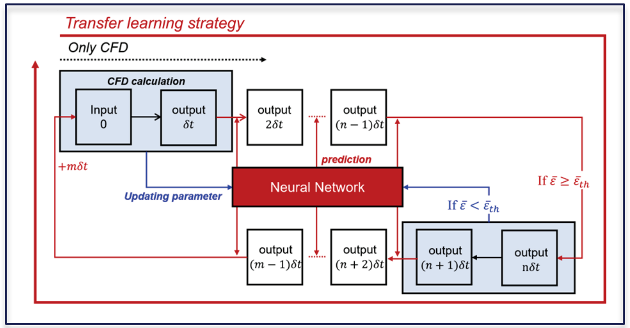
Accident analysis
(Kang, 2022)

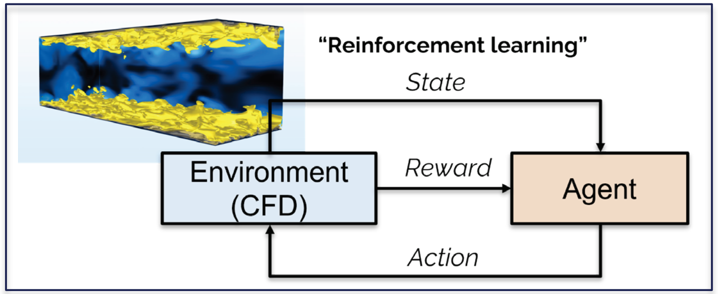


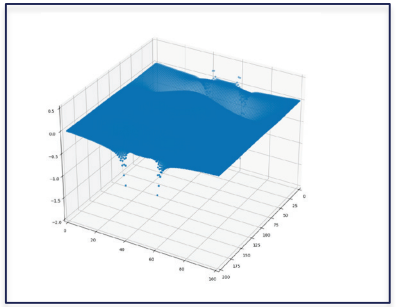
new reactor design









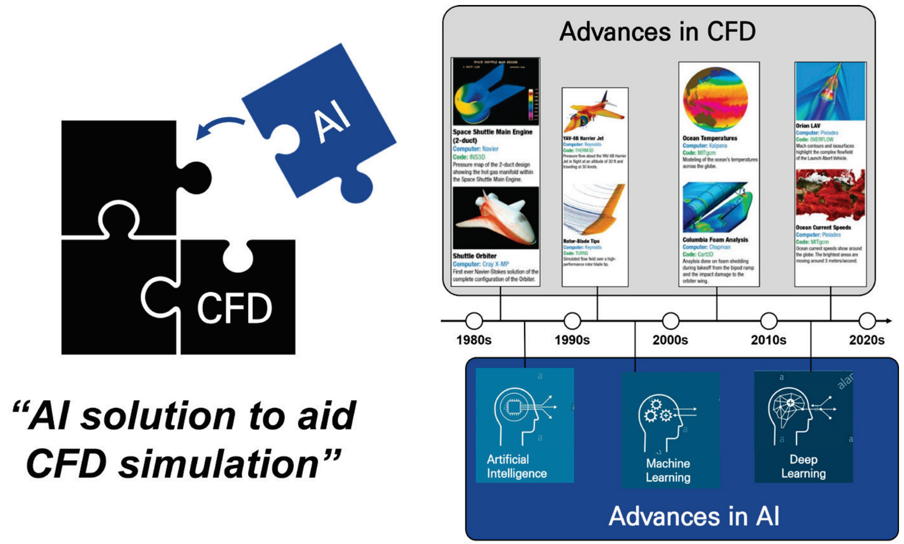


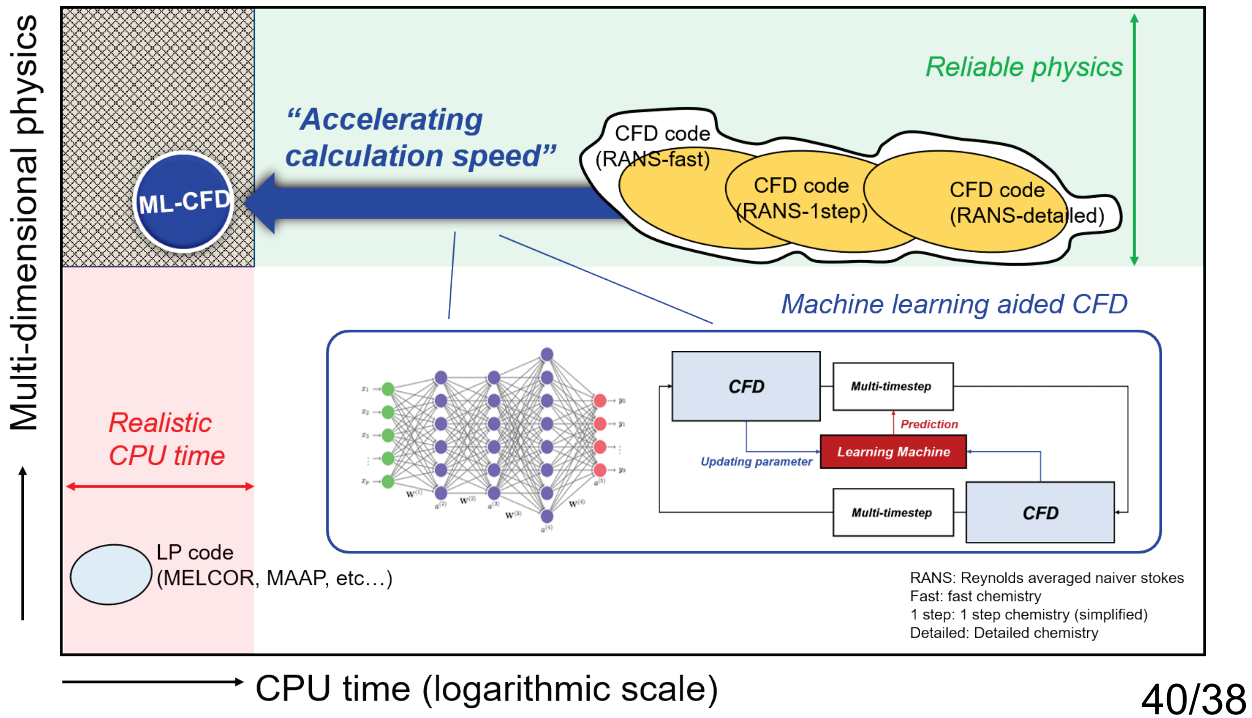
Thank you for listening!

jgjeon41@jbnu.ac.kr

Appendix – why we need to accelerate the CFD

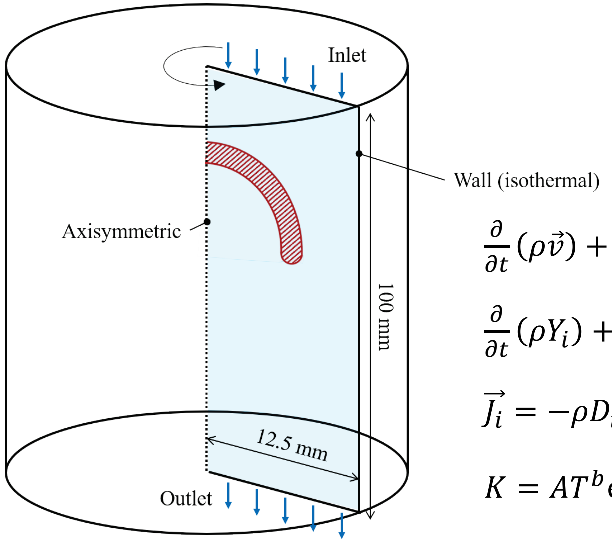
- Overview
  - To develop a **novel concept of network model** by understanding CFD and ML principles.
  - To evaluate the performance of the developed network model
  - To suggest a **computational framework** of the AI aided CFD simulation.





Appendix – hydrogen combustion dataset

- Unsteady CFD simulation datasets
  - In this study, **we produced datasets by the stabilized flame simulation**.
    * 5% H_2-air flame simulation to investigate the flammability of the hydrogen flames.
  - The need for DNN to deal with the non-linearity was highlighted by the governing equations.
  - Right figure shows **the entire timeline of the stabilized flame generation process.**
    * $t_{total} = 1000\,ms, \Delta t = 1\,ms$
    * The flame continues to expand until 0.5 s when the ignition energy is in effect.
    * After 0.5 s, the flame begins to stabilize through the balance of heat loss mechanisms and combustion heat.
  - This flame stabilizing period was selected as the subject of this study (0.600-0.611 s)



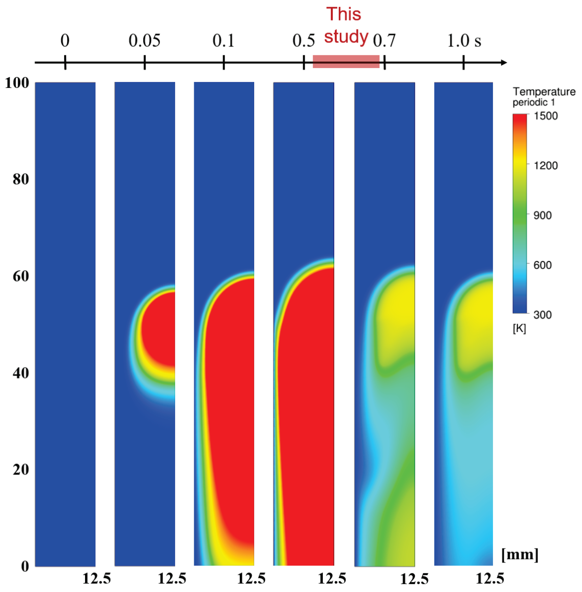
(Jeon, 2022)

$$\frac{\partial}{\partial t}(\rho\vec{v}) + \nabla\cdot(\rho\vec{v}\vec{v}) = -\nabla p + \nabla\cdot(\bar{\bar{\tau}}) + \rho\vec{g} \quad (1)$$

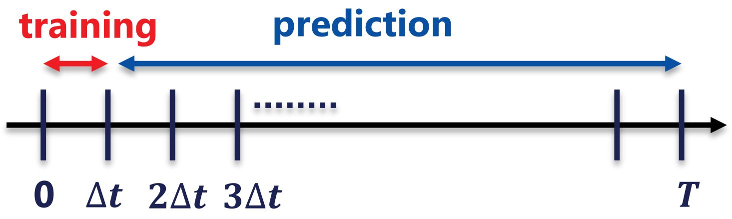
$$\frac{\partial}{\partial t}(\rho Y_i) + \nabla\cdot(\rho\vec{v}Y_i) = -\nabla\cdot\vec{J_i} + R_i \quad (2)$$

$$\vec{J_i} = -\rho D_{i,m}\nabla Y_i \quad (3)$$

$$K = AT^b\exp(-\frac{E_a}{RT}) \quad (4)$$



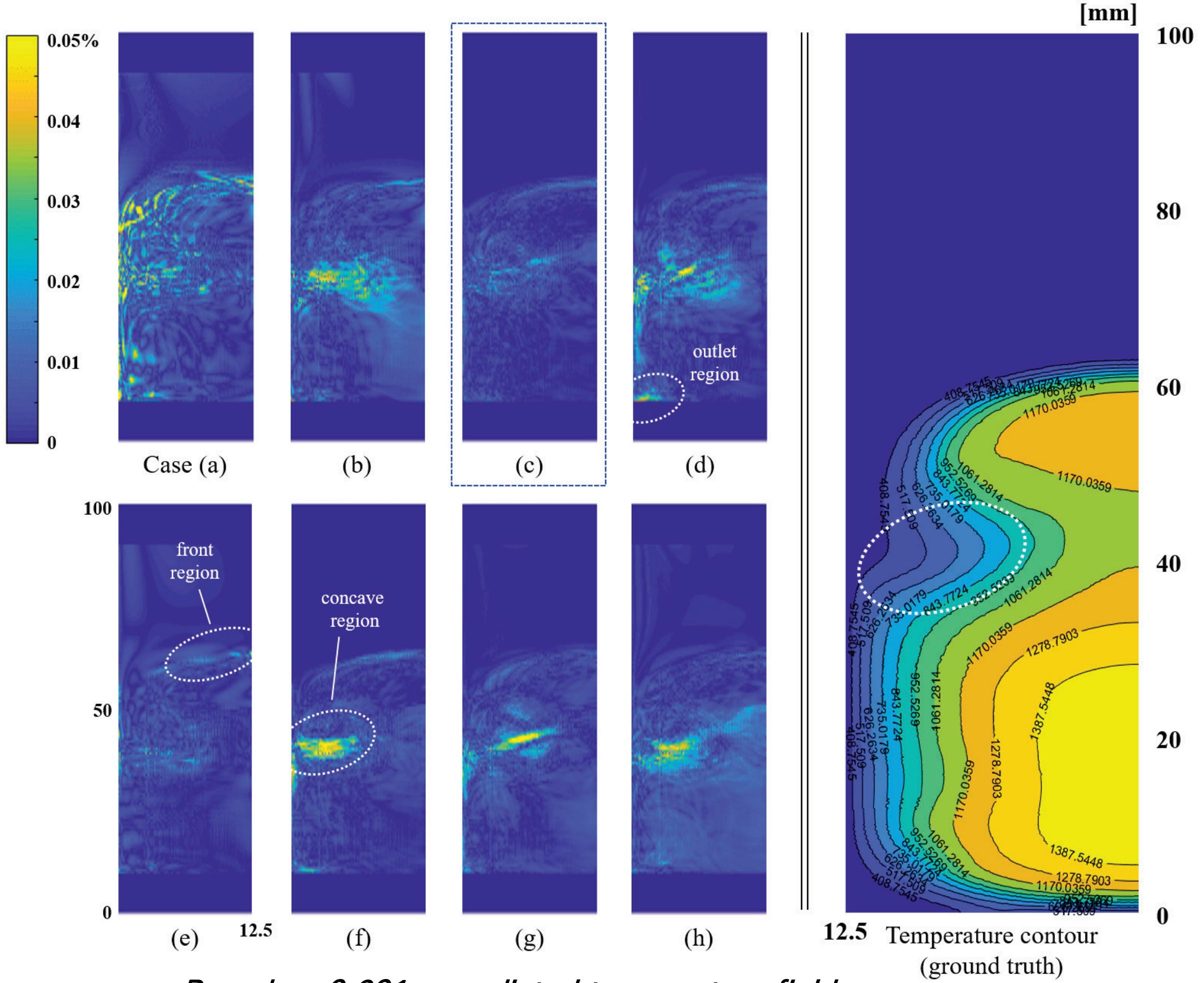
Appendix – hyperparameters



- Optimization of hyperparameters
  - The number of training examples is $n = M \times N$ (100,926).
  - The network models were **trained with 80%** of 0.600-0.601s data (loss function: **MSE for 20% validation dataset**).
  - After training, the trained networks predicted all 0.601 time series data by 0.600 data.
  - Right figure shows the **error distribution** between the ground truth data (CFD) and the predicted value.
  - **Overfitted networks** caused soar of errors in the stiff region

Test matrix for optimization of FVMN

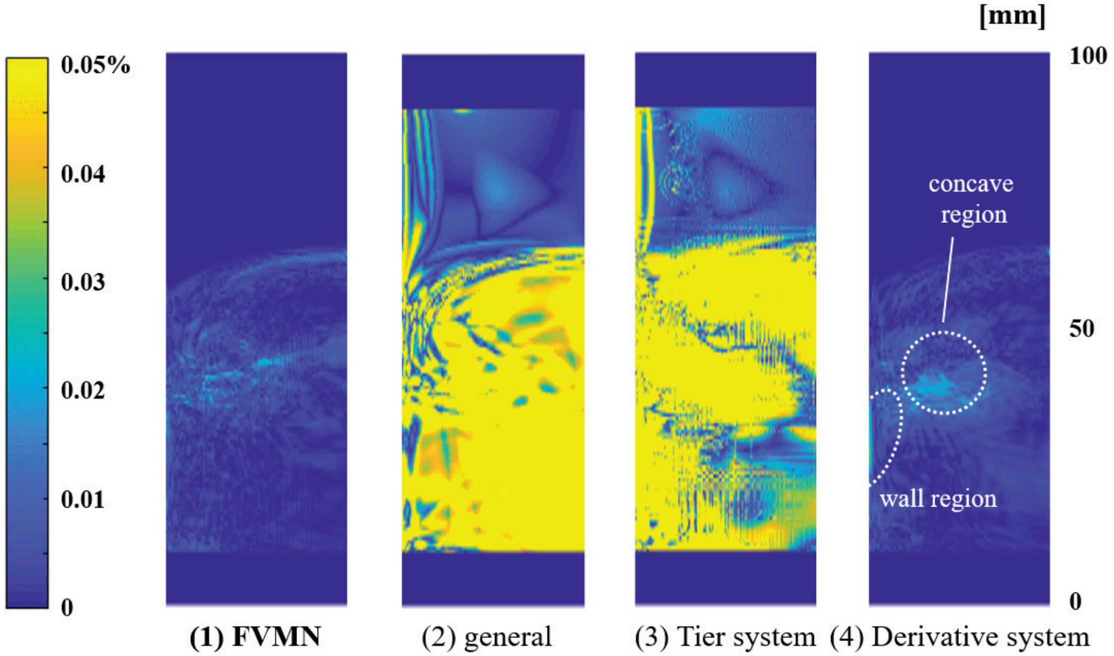
Case	Hidden layers	Number of parameters	Activation function	Learning rate	Loss function
a	64	2,049	ReLU	0.001	MSE
b	64, 64	6,209	ReLU	0.001	MSE
c	64, 64, 64	10,369	ReLU	0.001	MSE
d	64, 64, 64, 64	14,529	ReLU	0.001	MSE
e	64, 64, 64	10,369	Sigmoid	0.001	MSE
f	128, 128, 128	37,121	ReLU	0.001	MSE
g	256, 256, 256	139,777	ReLU	0.001	MSE
h	64, 32, 16	4,609	ReLU	0.001	MSE



Based on 0.601 s predicted temperature field

Appendix – tier/derivative systems

- Performance evaluation of the FVMN: (1) training/validation dataset
  - Below figure **shows the effect of the tier and derivative system** application on the error reduction.
  - The only difference in case (1-4) is the form of input and output variables during training/prediction process.
  - **The efficacy of the derivative system is much more noticeable- scale separation effect:**
  - Although the maximum relative error is similar between cases (1) and (4), the local error at the wall boundary region is more pronounced in case (4).



$$I = 6\ (T, v_x, v_r, X_{H_2O}, X_{H_2}, X_{O_2})$$

$$X_t^t = \left[x^t_{1_{i,j}}, x^t_{1_{i-1,j}}, x^t_{1_{i+1,j}}, x^t_{1_{i,j-1}}, x^t_{1_{i,j+1}}, \cdots x^t_{I_{i,j}}, x^t_{I_{i-1,j}}, x^t_{I_{i+1,j}}, x^t_{I_{i,j-1}}, x^t_{I_{i,j+1}}\right]^{\mathsf{T}}, X_t^t \in R^{(5\times I)} \quad (1)$$

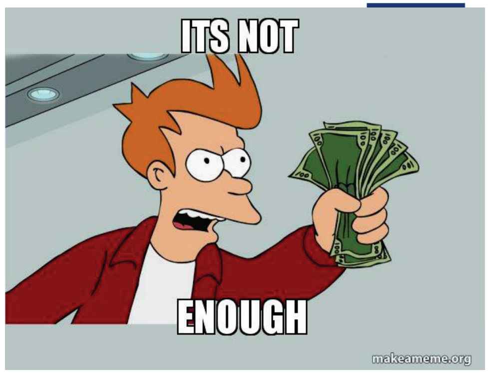
$$Z_d^t = \left[\left(\frac{\delta x_1}{\delta t}\right)^{t+1}_{i,j}\right] \text{ where } Z_d^{t+1} \in R \quad (2)$$

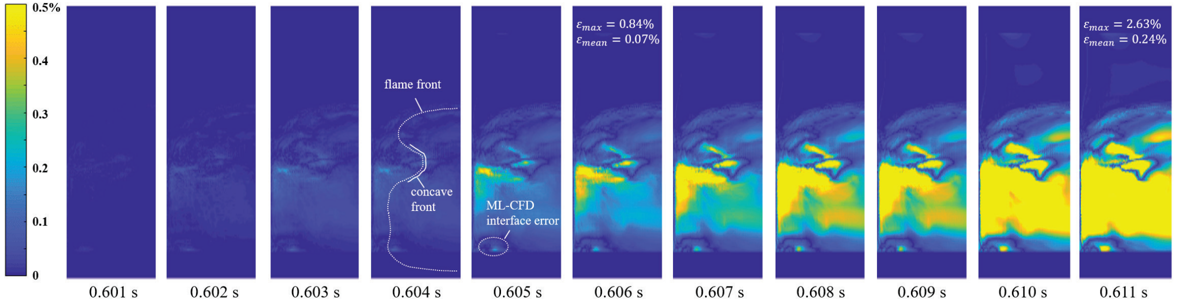
$$X^t = \left[x^t_{1_{i,j}}, \cdots x^t_{I_{i,j}}\right]^{\mathsf{T}}, X^t \in R^{(I)} \quad (3)$$

$$Z^t = \left[x^{t+1}_{1_{i,j}}\right] \text{ where } Z^t \in R \quad (4)$$

Appendix – FVMN performance

- Performance evaluation of the **FVMN**: (2) test dataset
  - We evaluated the **performance of the FVMN in multi-step prediction** (0.601 – 0.611 s).



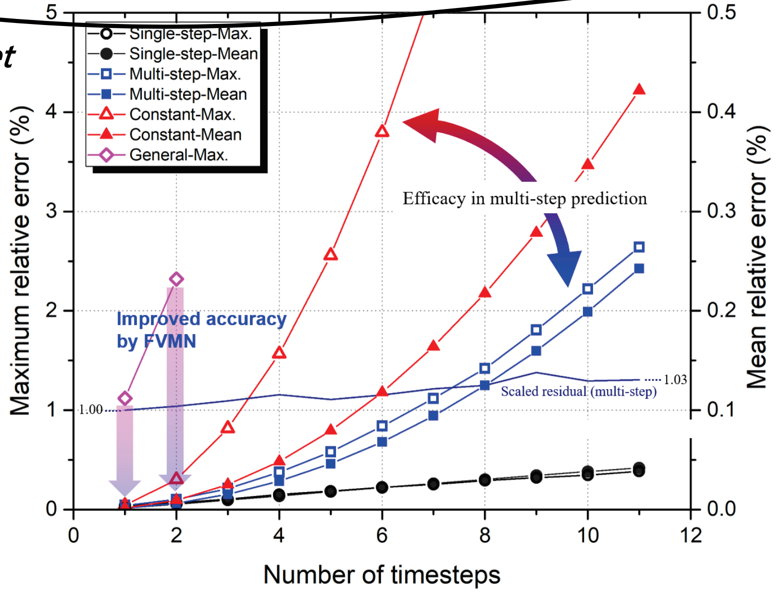


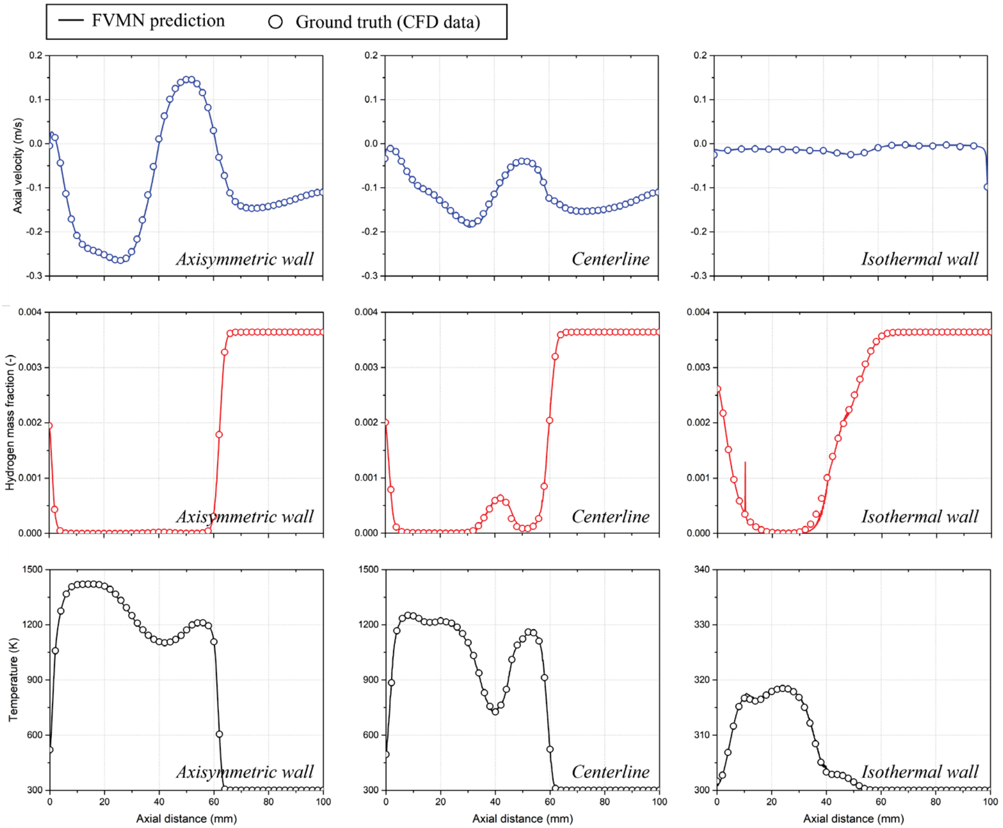
Training dataset

Test dataset

CNN: >10,000 snapshots
FVMN: only 2 snapshots

In CFD, each grid has a unified relationship with the neighboring grid (governing equations).

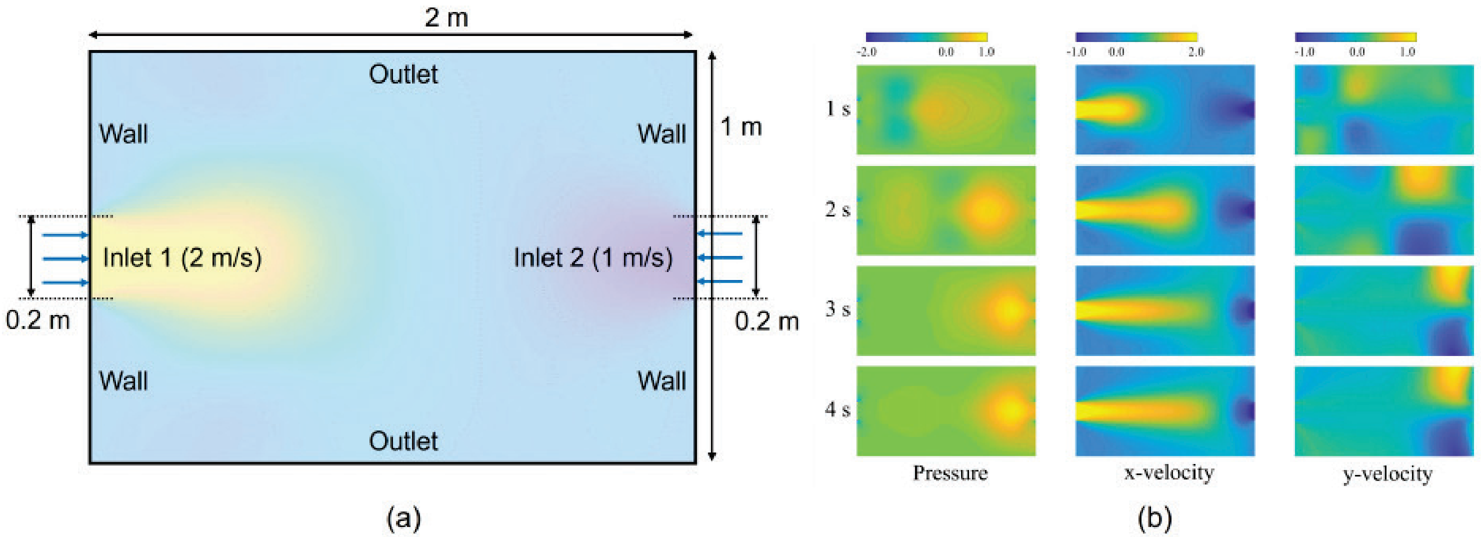




Appendix – PINNs dataset

- Physics-informed ML strategy

  1. Development of physics-informed neural networks
  2. ML-CFD cross-coupling computation strategy (transfer learning)



(Jeon, 2022)

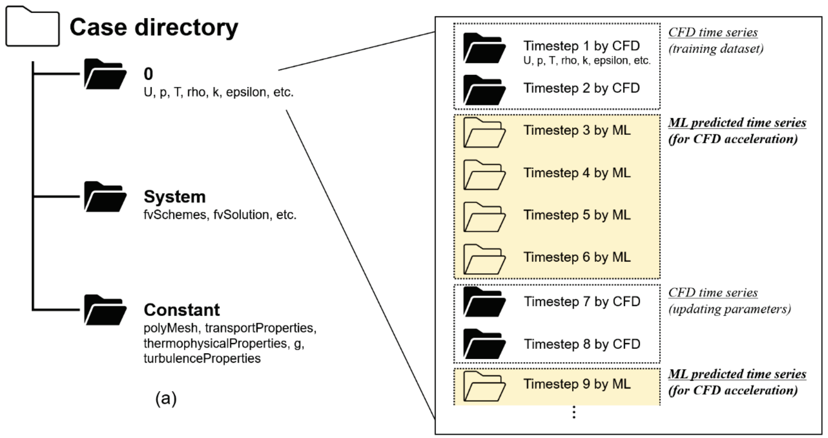
Summary of CFD dataset: counterflow simulation: time 4 s, timestep 0.01 s

Appendix – OpenFOAM-TensorFlow

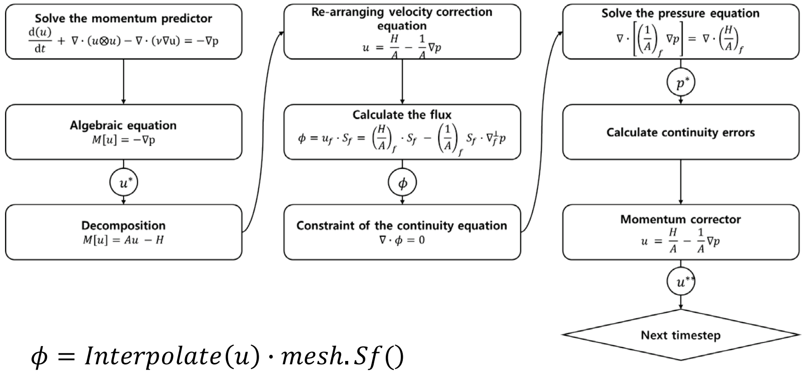
- (2) Cross-coupling strategy

Mr. J. Lee (HYU)

- Computational framework: OpenFOAM (CFD) – TensorFlow (ML)
- OpenFOAM solver: icoFOAM (ϕ-calculation algorithm)



(a) (b)



$$\phi = Interpolate(u) \cdot mesh.Sf()$$

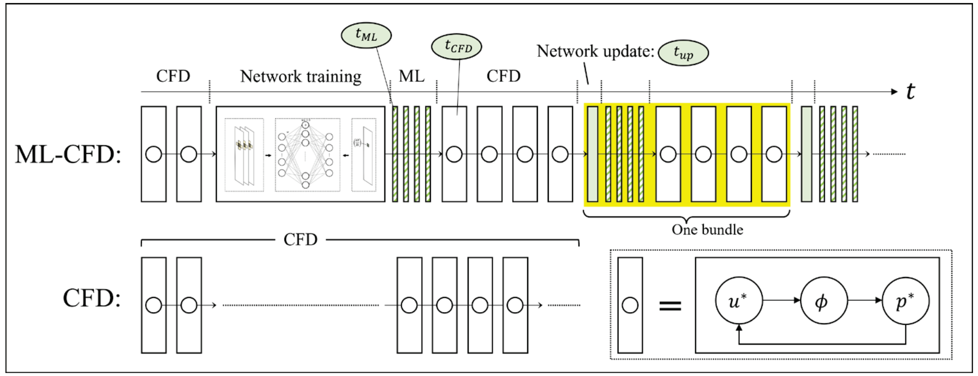
$$\phi = u_f \cdot S_f = \left(\frac{H}{A}\right)_f \cdot S_f - \left(\frac{1}{A}\right)_f S_f \cdot \nabla_f^{\perp} p$$

Appendix – Performance comparison

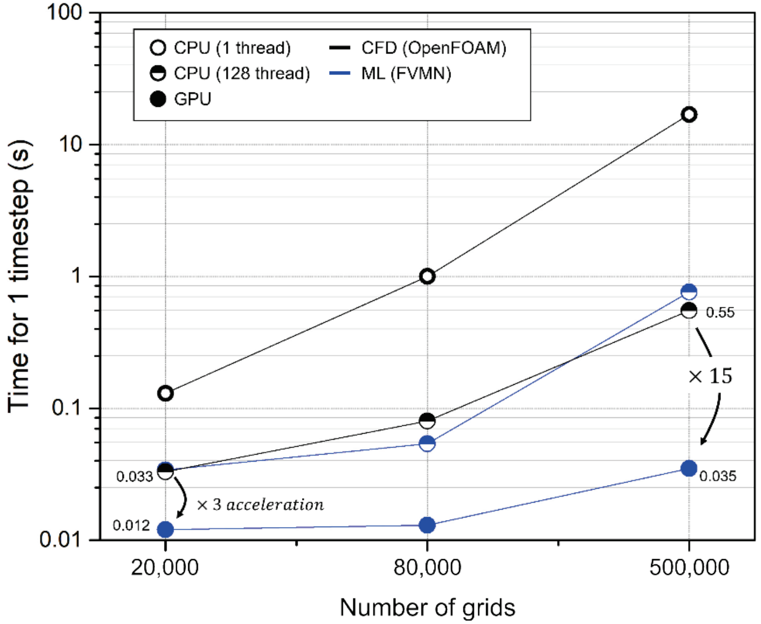
- (2) Cross-coupling strategy

(Jeon, 2022)

- **Required time: t_{ML}, t_{CFD}, t_{tr}, t_{up}**
- About 1.8 times acceleration performance



Computational timeline



Acceleration performance: $$\psi = \frac{n_{CFD}t_{CFD}}{n_{CFD}t_{CFD} + t_{up} + n_{ML}t_{ML}} = 1.8$$