「 Quantum Machine Learning 」

# Solving Differential Equation in Multiphysics Application with **DQC**(Differentiable Quantum Circuit)

포스코홀딩스 미래기술연구원 AI연구소
박찬신 수석연구원

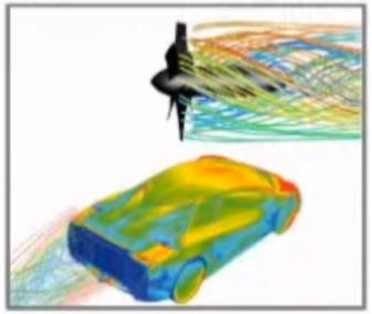
# **Introduction** Multiphysics

## Multiphysics

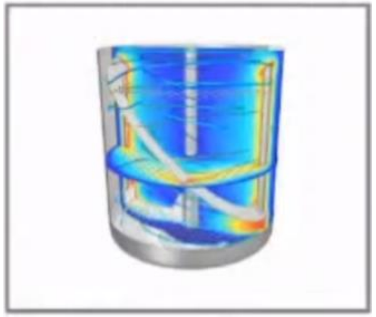
- 여러가지 역학 현상(열, 유체, 고체 등)이 복합적으로 일어나는 시스템으로 자연현상의 다양한 문제를 해석하고자 하는 방법
- 다양한 시/공간계(temporal/spatial scales)가 동시에 상호작용하는 복잡한 물리적 현상(phenomena)을 묘사
  → 다양한 미분방정식(DE: Differential Equations)의 집합으로 이루어짐

### Potential use-cases for multiphysics simulations

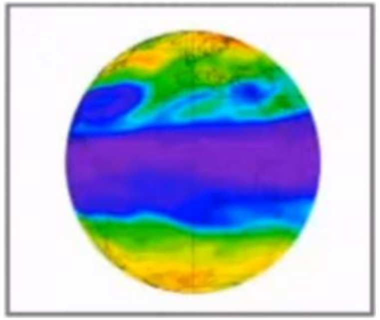
Mechanical Engineering



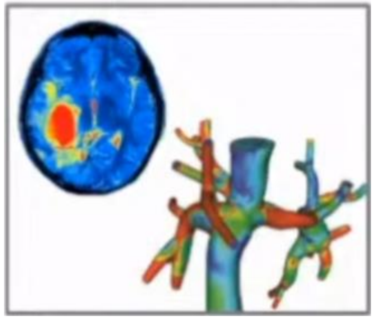
Chemical Engineering



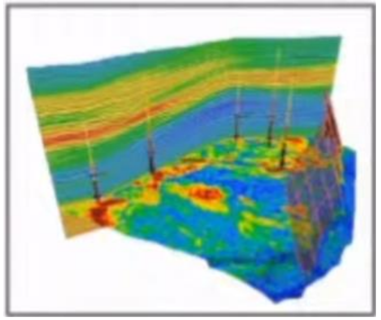
Metrology & Climate



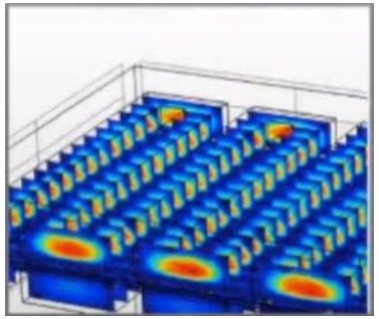
Biomedical



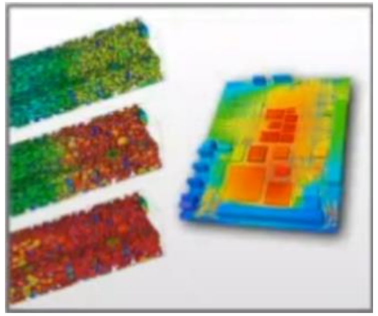
Oil & Gas



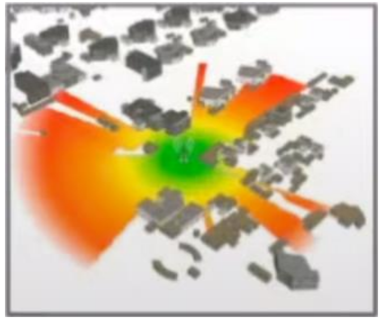
Fuel cell design



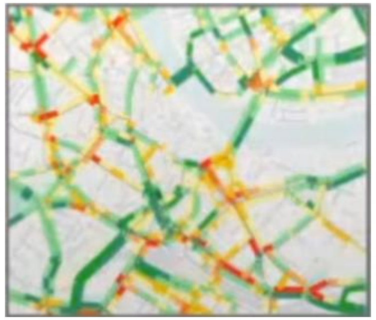
Batteries & Electronics



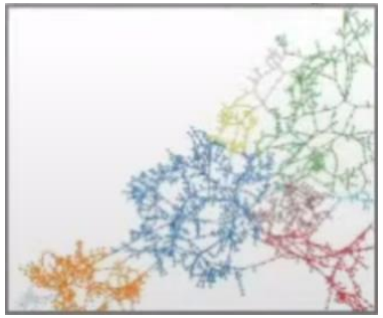
Wireless networks



Traffic & Logistics



Powergrids



# **Problem** Differential Equations

## 미분방정식: "변화"에 대한 수학적 모델

- **Mechanics and Fluid Dynamics**: describing the equation of motion of waves or a pendulum
- **Medical science**: the growth or spread of certain diseases in the human body
- **Ecology**: description of various exponential growths and decays
- **Finance**: market dynamics, calculation of optimum investment strategies to assist the economists
- **Epidemiology**: disease spreading, description of various exponential growths and decays

Examples

| | | | |
|---|---|---|---|
| $\frac{d^2s}{dt^2} = -g$ | Free Falling Body<br>자유낙하 운동 | $\frac{dP}{dt} = \text{kP}$ | Population Growth<br>인구학 |
| $\frac{d^2\theta}{dt^2} + \frac{g}{l}\theta = 0$ | Harmonic Oscillator<br>조화 진동운동 | $\frac{dP}{dt} = P(\ a\text{-}bP\ )$ | Population Growth<br>(limited resources)<br>인구학 |
| $\frac{dT}{dt} = \text{k}(\text{T - }T_m)$ | Newton's Law of Cooling<br>뉴턴의 냉각법칙 | $\frac{dx}{dt} = kxy$ | Spread of Disease<br>전염병학 |
| $\frac{d^2y}{dx^2} = k\sqrt{1 + (\frac{dy}{dx})^2}$ | Shape of a hanging string<br>역학 | $\frac{dy}{dt} = \text{y}(\alpha \text{ - } \beta\text{x})$<br>$\frac{dx}{dt} = \text{x}(\text{-}\gamma\text{+}\delta\text{y})$ | Predator-Prey<br>전염병학 |

# **Solution** Classical Approach

## 고전 미분방정식의 종류와 풀이 기법

- Grid/Mesh기반 기법(Local Method): rely on discretization of the space of variables, with derivatives being approximated with numerical differentiation techniques fine grid for multivariable functions require increasing computational cost
- Spectral 기법(Global Method): represent the solution in terms of a suitable basis set. Finding spectral solutions for complex  problems require ever-increasing basis sets to achieve high accuracy.

| | | |
|---|---|---|
| **Macroscopic** Eulerian approach, starting from the equations of motion | **Spectral methods** Use spectral basis functions for truncated series expansion | Fourier series |
| | | Chebyshev series |
| | **Grid-/mesh-based methods** Discretize the governing equations and solve on a grid (mesh) or point-cloud (meshfree) | Finite Element |
| | | Finite Volume |
| | | Finite Difference |
| | | Meshfree |
| | | Immersed Boundary |
| **Micro-/mesoscopic** Lagrangian approach, directly modeling particle motion and collisions | **Molecular dynamics** Calculate motions of individual atoms acted upon by interatomic potentials | DSMC |
| | **Lattice Gas methods** Solve motion of particles with discretized momenta on discretized positions (lattice) | LGCA |
| | **Lattice Boltzmann** Simulate fluid density on lattice with streaming with collision/relaxation processeses | LB-BGK |
| | | Entropic LB |

# **Limitation** Classical Approach

## 고전적인 풀이 기법(Solver)의 한계

- Grid/Mesh 기반의 수치해석적(Numerical Method) 기법이 가장 많이 활용되고 있는 방법이다.
- 높은 정확도나 복잡한 문제를 해석하기 위해서는 Grid/Mesh의 크기가 늘어나야하고 이에 따른 필요 계산량이 늘어나기 때문에, 확장이 어려운 단점이 있다.

### <1 minute

Thermal fatigue in T-junction

Example simulation
- 3D k-epsilon model
- 2$^{nd}$ order in time/space
- Eulerian approach
- Structured mesh
- $10^4$ cells

Classical runtime
- <1 minute
- $10^3$ cores
- 5 Gb memory

### ~50 hours

Golf  ball aerodynamics

Example simulation
- DNS model
- Supercritical flow
- 3$^{rd}$ order in time/space
- Eulerian approach
- Structured mesh
- $10^9$ cells

Classical runtime
- ~50 hours
- $10^3$ cores

### Weeks

Nuclear fuel assembly

Example simulation
- LES turbulence model
- 2$^{nd}$ order in time/space
- Eulerian approach
- Unstructured mesh
- $10^{10}$ cells

Classical runtime
- Weeks
- $10^6$ cores
- 25 Tb memory

### Intractable

Aerospace modeling

Example simulation
- Turbulence model
- LES: 1012 cells
- DNS: 1020 cells
- DNS level accuracy is needed by the industry within next 10 years

Classical runtime
- DNS: intractable

# **Solution** New Approach(Classical/Quantum)

## 새로운 미분방정식 풀이 기법(Solver)

- Deterministic Classical Solver: 고전적인 수치해석적 계산 기법(유한요소법, 유한차분법, 경계요소법 등)
- Variational Classical Solvers(PINN): 많은 양의 basis 함수에 대한 학습이 필요하고 이로인한 계산복잡도와 긴 학습시간이 요구될 수 있다.
- Variational Quantum Solvers: 양자컴퓨팅의 병렬연산을 활용하여 PINN의 한계인 처리가능 basis함수의 크기를 늘리는 양자회로기반 학습 기법
- Deterministic Quantum Solvers(Fault-tolerant QC): HHL[1)]알고리즘

### Deterministic Classic Solvers
(Spectral & Grid/Mesh-based methods)

- Examples include grid-based methods (finite elements) or discrete spectral methods
- Grid-based methods typically require a very large number of grid-points, while discrete spectral methods are more efficient, but struggle dealing with complex boundary conditions
- One downside of all deterministic methods is that they are not variational in nature, which means one may only hope to improve the result by increasing discretization resolution further.

### Variational Classical Solvers
(Physics Informed Neural Networks)

- Nerual network(NN) solvers are variational in nature: NN nodes are used to represent basis functions and are trained to represent a function that approximately satisfies a set of differential equations and boundaries.
- These methods are slowly coming out of academia to industry, because they show good convergence for smooth functions, can deal with high degree of non-linearity and can handle sharp gradient
- However, they typically require a large number of basis functions which increases computational complexity and their training time.

### Deterministic Quantum Solvers
(HHL kind of algorithms)

- Many proposed quantum solvers typically employ some version the so-called HHL quantum subroutine, which can be used to solve linear systems efficiently
- However, HHL type algorithms are often only suitable for long-term fault-tolerant quantum processors
- Data is assumed to be encoded in amplitudes, posing a data input- and data output-problem
- HHL-type algorithms typically need to linearize any nonlinearity in the problem before solving it
- Derivatives are estimated using finte difference

### Variational Quantum Solvers
(Differentiable Quantum Circuits)

- In 2019 a first proposal was made for a variational quantum algorithm for solving nonlinear DEs
- However, similar to the deterministic quantum solvers, this proposal requires the efficient conversion of a large classical dataset into the amplitudes of a quantum wave function, which is not(yet) possible
- Additional downsides of this algorithm include inaccuracies due to numerical differentiation, and while in-principle it is NISQ-compatible, the circuit coherence requirements are unfeasible for near-term hardware.

1) Harrow, Hassidim, and Lloyd, PRL 103.15 (2009)

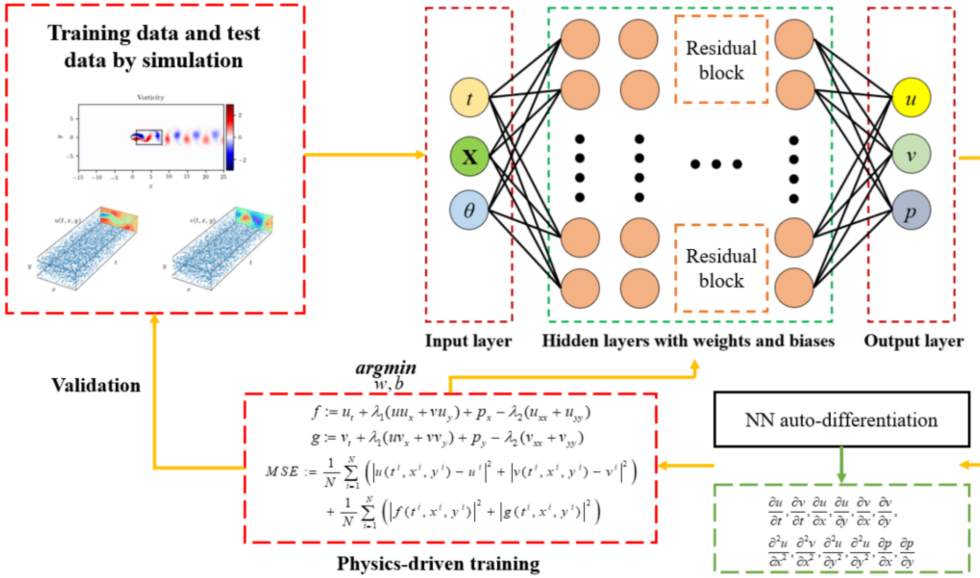
# New Approach(Classical) PINNs

## Physics-informed Neural Networks

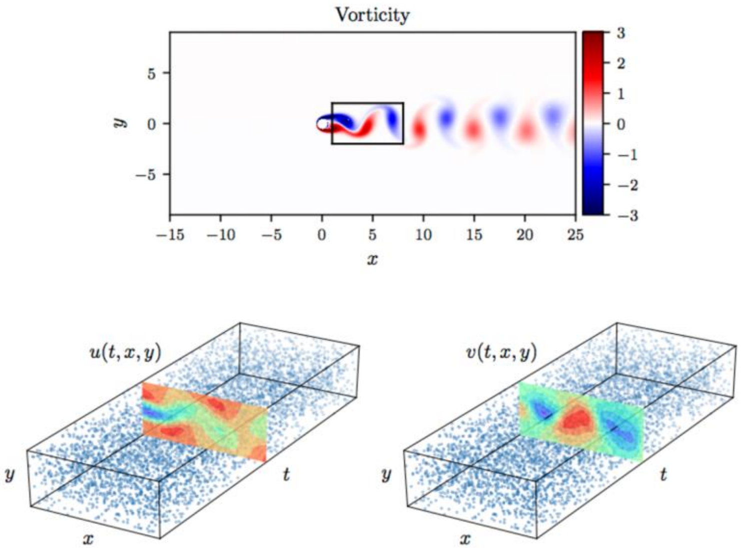
- 기존의 전통적인 수치해석적 방법과 달리 메쉬가 필요없는 mesh-free기법으로 복잡한 형상에도 효과적으로 적용 가능하고, 비교적 빠른 시간에 정확한 PDE해를 도출할 수 있다.
- 기존의 신경망이 데이터 중심 접근 방식인데 비해 PINN은 PDE(편미분방정식)에 내재되어 있는 물리 정보를 신경망에 도입함으로 학습 데이터를 줄여 학습 속도를 빠르게 한다.

  ※ 기존 FEM 방법에 비해 Nonlinear Dynamics를 정확히 계산할 수 있음이 Bifurcation Theory 연구에서 보고됨[1].

  ※ Paremeter가 충분하다면 Universal Approximation Theorem[2]과 Universal Differential Equation Theorem[3]에 의해서 복잡한 Geometry도 학습 가능




<PINN with Resnet Block for solving Fluid Flow Problems[5]>




<Navier-Stokes equation4)>

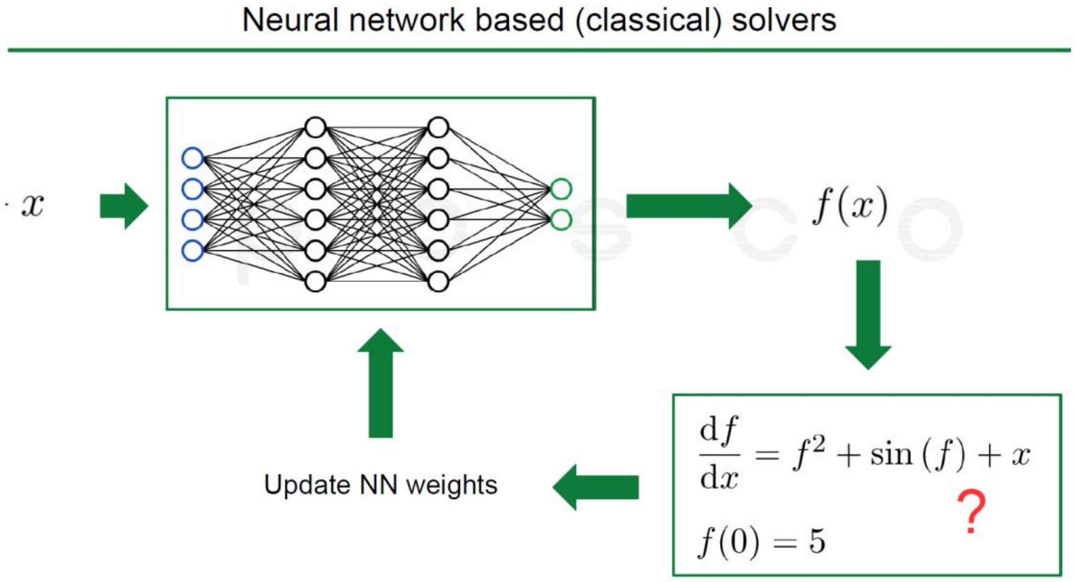

1) Sandor Beregi et al., Using scientific machine learning for experimental bifurcation analysis of dynamic systems, *arXiv*. 2110.11854, 2022
2) Hornik Kurt et al., Multilayer Feedforward Networks are Universal Approximators, Neural Networks, 2, 359–366
3) Christopher Rackauckas et al., Universal Differential Equations for Scientific Machine Learning, arXiv. 2001.04385, 2020
4) M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics, Volume 378, 2019
5) Cheng, C.; Zhang, G.-T. Deep Learning Method Based on Physics Informed Neural Network with Resnet Block for Solving Fluid Flow Problems. Water 2021

# New Approach(Quantum) DQC

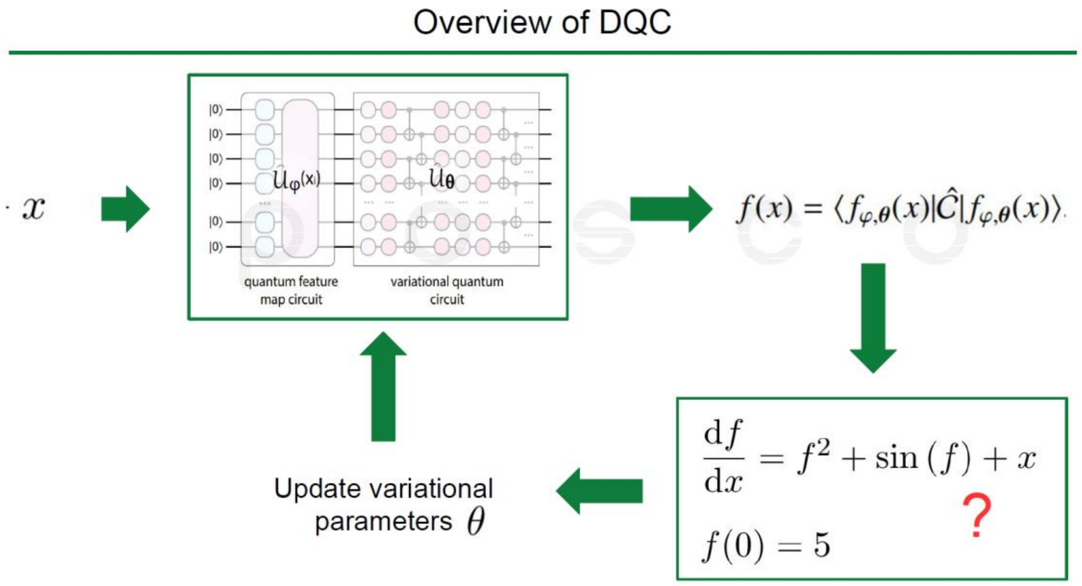
## Differentiable Quantum Circuits

- 양자컴퓨팅을 활용한 PINN기반의 미분방정식 풀이 기법(Solver)
- Expressivity: 양자컴퓨팅의 장점인 지수적인 크기의 양자 상태공간(state space)을 활용한 계산으로 넓은 범위의 문제들에 대해 UFA*를 만족할 수 있다.
- Differentiablity: 지수적인 크기의 양자 상태공간으로 다양한 형태의 입력값과 파라미터의 조합에 대해서 미분가능성을 보장할 수 있다.

* UFA(Universal Function Approximator) [1)]




<PINN>




<DQC>

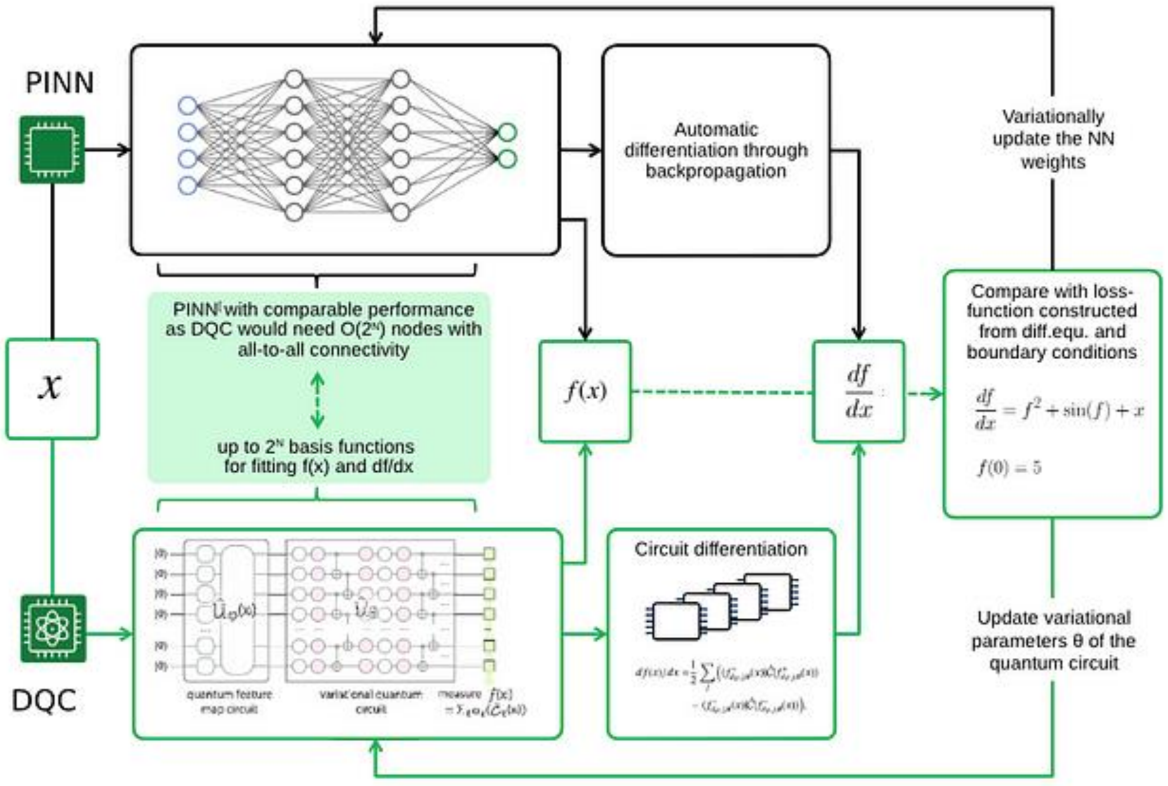
1) Hornik Kurt et al., Multilayer Feedforward Networks are Universal Approximators, Neural Networks, 2, 359–366

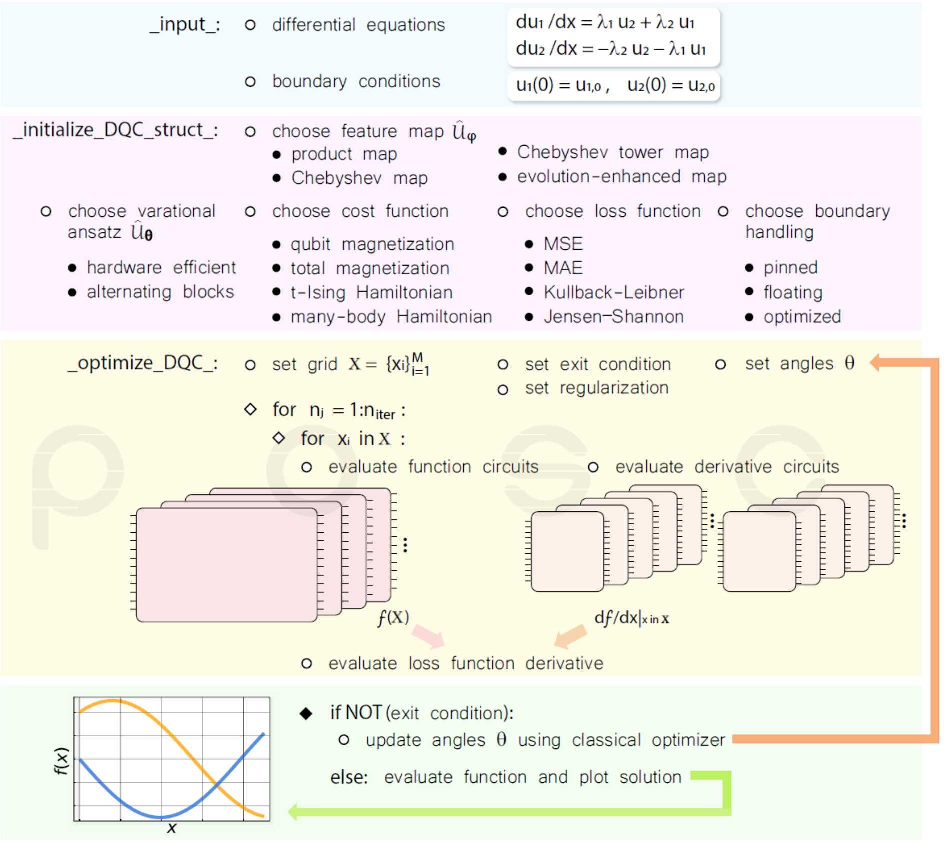
# New Approach(Quantum) DQC

## DQC workflow [1] (PASQAL)

- Choose the quantum circuit composition

1) quantum feature map 2) ansatz of variational quantum circuit 3) define cost function 4) define loss function 5) boundary handling strategy




1) Kyriienko, Oleksandr and Paine, Annie E. and Elfving, Vincent E, Solving nonlinear differential equations with differentiable quantum circuits, Phys. Rev. A 103, 052416 – Published 17 May 2021

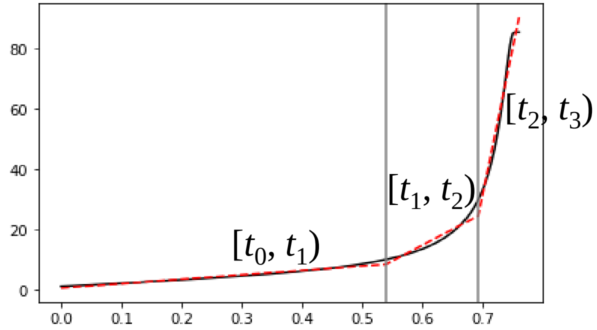
# **Advantage** Convergence

## PINNs기법은 복잡한 시나리오의 문제에 수렴이 어려움

- DQC를 통해 양자컴퓨팅이 가지고 있는 이점을 활용하여 수렴(Convergence)을 가속(Boosting)할 수 있다.

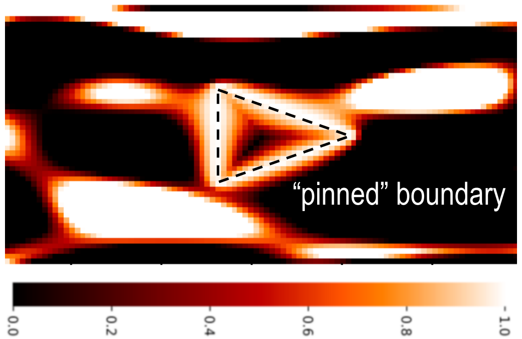
### Domain decomposition

- When the profile of the target solution to learn is hard, domain decomposition can lower the required expressivity of the QNN used to parameterize it.
- It decomposes the problem within different subdomains that are solved independently. The solutions are then combined together by appropriate interface conditions, so that a solution in the full domain is available.




### Boundary Pinning

- PINNs can be greatly accelerated by incorporating in the architecture knowledge of (some of) the properties of the solution.
- Here, we applied this general principle by replacing the variational penalty against solutions that do not satisfy the boundary conditions, with a "pinning" strategy hence avoiding expressivity to be wasted.
- It replaces a simple universal function approximator $NN(x, \theta)$ with a smooth function designed to automatically satisfy a certain boundary condition.




$\boldsymbol{NN'(x,\theta)}$

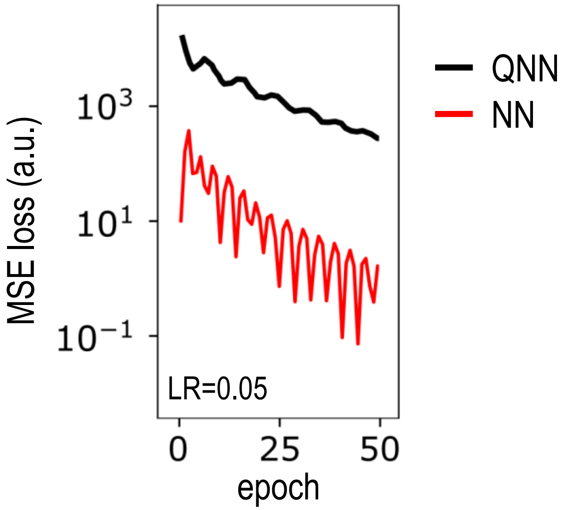
$$NN'(x,\theta) = ce^{-kd(x)} + \left(1 - e^{-kd(x)}\right)NN(x,\theta)$$

for a given boundary condition $f(x)|_{x\in\partial\Omega} = c$.
Here, $d(x)$ is set to represent a certain distance from boundaries

# **Benchmark** PINNs vs DQC

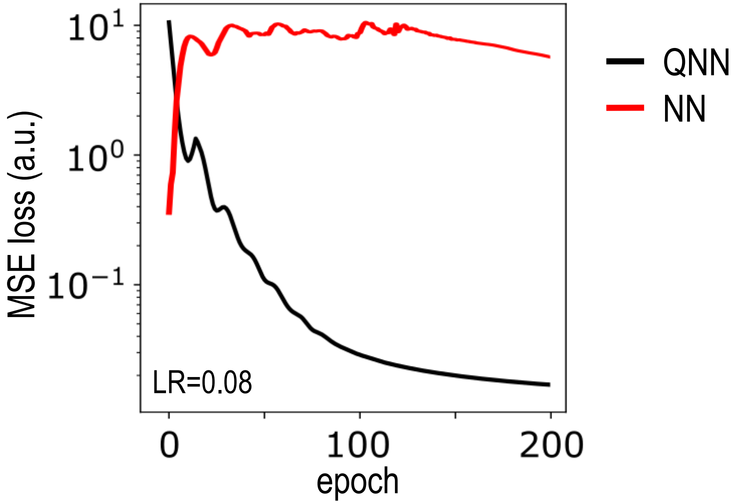
### Stability

When training the model with global optimizers, QNNs appears typically more tolerant to high learning rates (LR) than classical NNs, as can be seen by smoother evolution of the loss



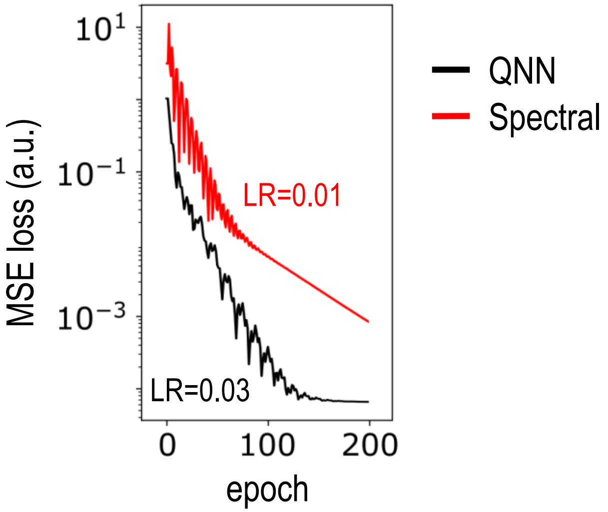

### Trainability

In extreme, problem-specific cases, we can observe the learning being disrupted for classical architectures by LR values where QNN can still be meaningfully trained




### Convergence rate

When compared with a more similar architecture (a trainable spectral decomposition), QNNs can use higher LR to converge faster to the correct solution



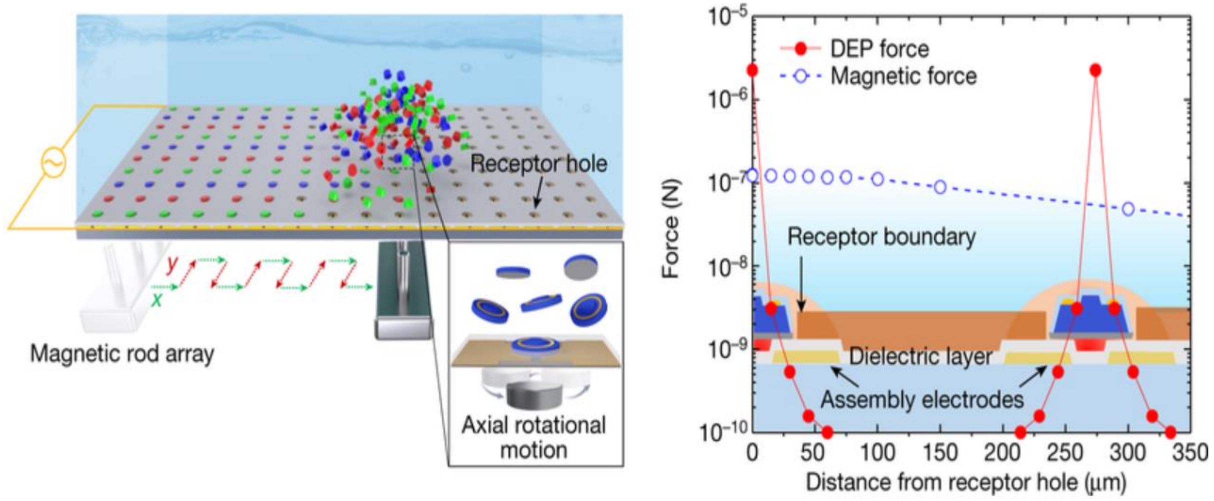

**Caveats** : These comparisons are exemplary behaviour, and one must bear in mind that the training instances shown adopt fundamentally different paradigms.

- The initialization procedure is different across NNs and QNNs, hence for the same instance we often observe different starting loss.
- The collocation strategy of points is typically optimized against the *quality of the final solution*, which is correlated with the *loss* shown here, yet not interchangeable with the latter.
- Equivalently, the number of free parameters leading to similar final solutions can differ between NNs and QNNs, impacting also the convergence in the two cases

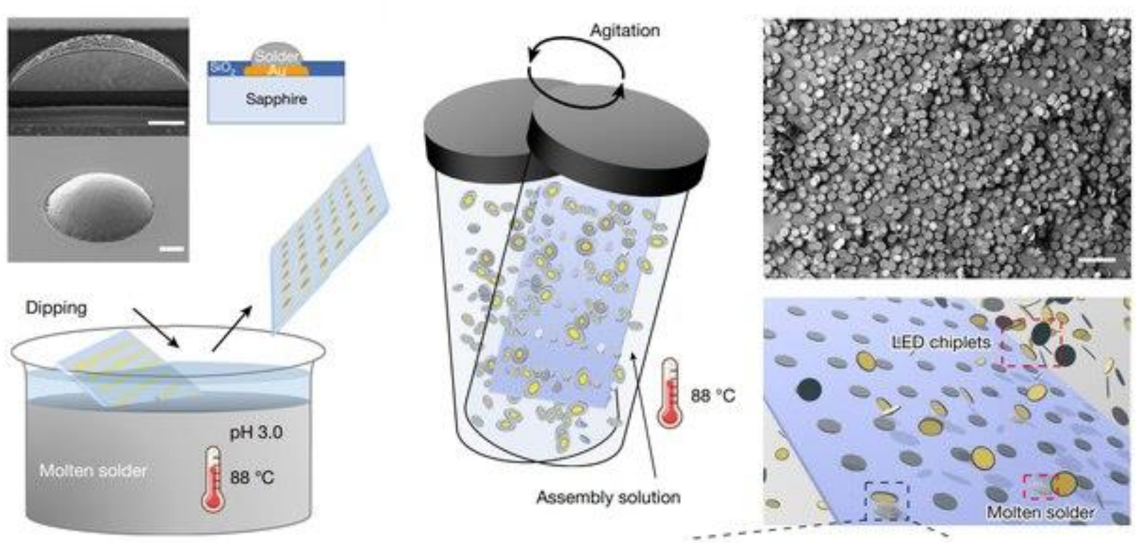
# **Application** Micro-LED

## Micro-LED

- 마이크로 LED는 유기발광다이오드(OLED) 디스플레이에 비해 긴 수명, 높은 밝기 등 다양한 장점으로 차세대 디스플레이로 각광받고 있다.
- 마이크로 LED가 주류 제품 시장에 진입하여 다른 상용디스플레이와 경쟁하기 위해서는 높은 처리량, 높은 수율, 유리 크기까지의 생산 확장성 등 전사 기술의 장애를 극복해야 한다.
- **자기력 보조 유전영동 자가조립(MDSAT)**: 자석을 이용해 마이크로LED 소자가 스스로 조립하는 기술로 15분 이내에 99.99%의 RGB LED 동시 전사 수율을 달성하는 기술




MDSAT 유체 조립 공정의 모식도 및 COMSOL 시뮬레이션에서 계산된 DEP 및 자력 프로파일




유체자기조립 마이크로LED를 만드는 세부 과정. 기판에 납땜을 하고 88도 물에 넣어 납을 살짝 녹인 뒤 특수 용액, 마이크로LED 부품과 함께 통에 넣고 흔들면 부품들이 자기 자리를 알아서 찾아가면서 LED가 조립된다.

Nature, DOI: https://doi.org/10.1038/s41586-023-06167-5

# **Micro-LED** particle motion simulation

## Simulating particle motion

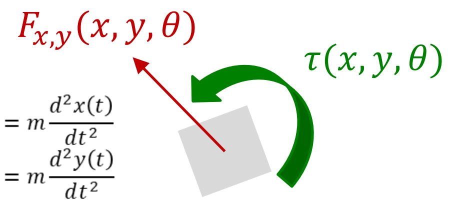
- 입자의 무게(m), 시작 위치(x,y), 각속도, 역장과 토크를 통해 일정 시간 후의 입자의 위치를 시뮬레이션하여 전사 수율을 높이도록 제어.

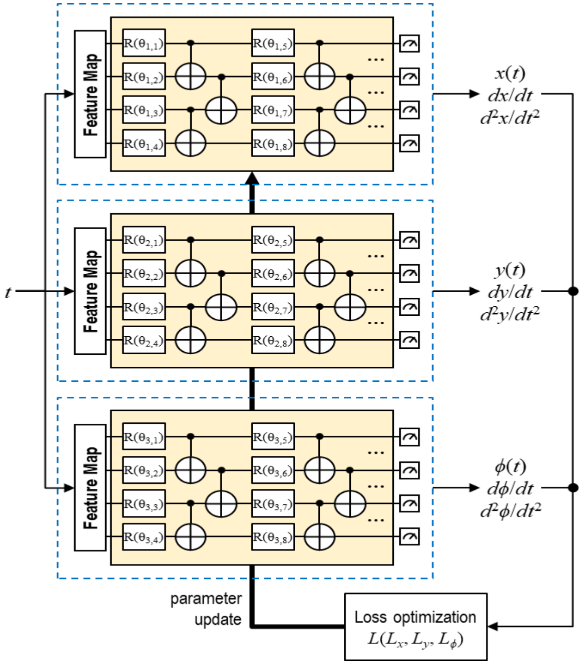
### Problem setting

- Given a force field $F_i(x,y,\theta)$, a torque field $\tau(x,y,\theta)$, a particle mass $m$ with moment of inertia $I$, initial positions and velocities corresponding to the $x$, $y$ and angular dimensions

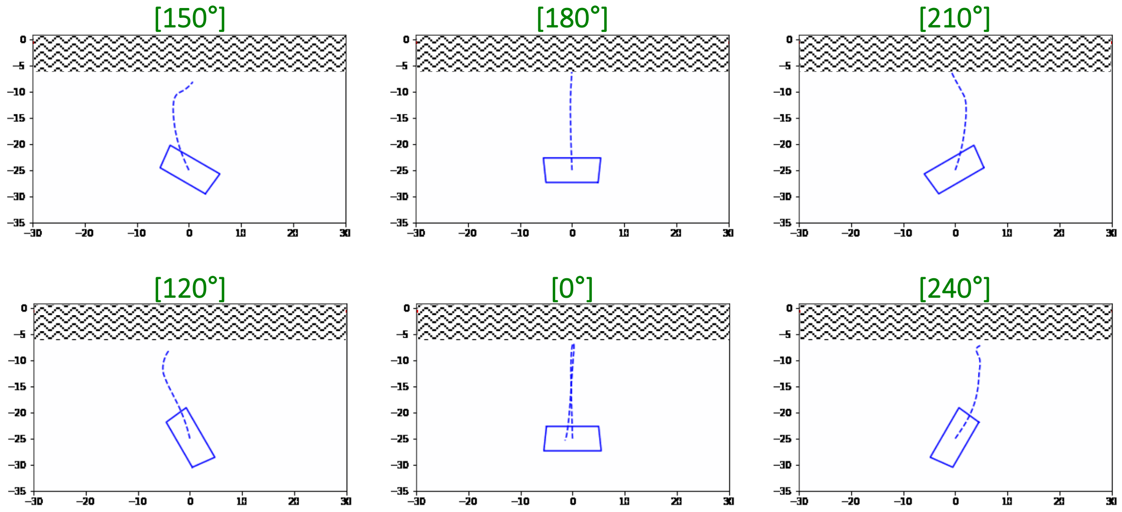
⇒ Computed the position and orientation of the particle at following times adopting 3 QNNs and domain decomposition approaches

$$F_x\big(x(t),y(t),\phi(t)\big) = m\frac{d^2x(t)}{dt^2}$$
$$F_y\big(x(t),y(t),\phi(t)\big) = m\frac{d^2y(t)}{dt^2}$$
$$\tau\big(x(t),y(t),\phi(t)\big) = I\frac{d^2\phi(t)}{dt^2}$$




### Simulated particle motion



# **Summary** QML/DQC

## QML의 기대효과

- Quantum Neural Network의 Degree of Freedom이 부족해도 학습이 되는 이유는 Analytic Continuation[1)]으로 설명
- Quantum Neural Network의 Universal Approximation Theorem이 Width방향으로 증명함[2)]
- No Cloning Theorem[3)]에 의해 Fan Out Gate가 존재할 수 없기 때문에 Neural Network의 width를 늘리기 위해서는 큐빗수를 늘려서 입력차원을 늘려야함
- Quantum Neural Network에서 같은 입력을 하는 큐빗을 늘리는 것으로 Expressivity가 지수적으로 증가하는것이 확인됨[4)]
- Quantum Neural Network의 지수적 Expressivity에 의해 복잡한 Geometry를 학습하는데 필요한 큐빗 수가 고전방식에 필요한 파라미터 대비 지수적으로 적음[2)]
- Quantum Probably Approximately Correct Learning[5)] 관점에서 Worst case의 입력 분포를 가질 때 학습 필요수에 양자이득이 증명되지 않았으나 가능성이 확인됨
- Quantum Machine Learning[6)]에서 얽혀있는 큐빗수가 증가함에 따라 학습 필요 수가 선형적으로  증가됨이 확인됨
- 양자컴퓨팅을 이용하면 복잡한 Geometry를 학습하는데 필요한 연산수가 지수적으로 줄어들 것으로 기대함

## **Conclusion** DQC

- DQC exploits NN based solvers by replacing the classical NN by a trainable quantum circuits
- In DQC, a quantum feature map is used to encode a trial function, then based on a bi-partite loss function that depends of the expectation value of a cost function, the parameters of a variational form are updated so as to improve the solution
- DQC is suitable for application on current noisy intermediate scale noisy devices
- DQC is a very versatile tool as it comes with different strategies to initialize its structure
- DQC uses analytical differentiation rather than numerical differentiation
- DQC doesn't linearize the problem, but solves nonlinearity directly
- DQC is compatible with a wide variety of differential equation types

1) M. D. Kruskal, Maximal Extension of Schwarzschild Metric, Phys. Rev. 119, 1743, 1960
2) Takahiro Goto et al., Universal Approximation Property of Quantum Machine Learning Models in Quantum-Enhanced Feature Spaces, Phys. Rev. Lett. 127, 090506, 2021
3) James Park, The concept of transition in quantum mechanics, Foundations of Physics. 1, 23–33, 1970
4) Yadong Wu et al., Expressivity of Quantum Neural Networks, Phys. Rev. Research 3, L032049, 2021
5) Ryan Sweke et al., On the Quantum versus Classical Learnability of Discrete Distributions, Quantum 5, 417, 2021
6) Hsin-Yuan Huang et al., Information-theoretic bounds on quantum advantage in machine learning, Phys. Rev. Lett. 126, 190505, 2021