PASQAL

QUANTUM DISCOVERY

PASQAL Differentiable Quantum Circuits (DQC)

PASQAL
www.pasqal.com
office@pasqal.com
7 rue Léonard de Vinci
91300 Massy
France

Solving nonlinear differential equations

Context

Many industry relevant problems can be described using differential equations:

$\frac{d^2s}{dt^2} = -g$ Free Falling Body

$\frac{d^2\theta}{dt^2} + \frac{g}{l}\theta = 0$ Harmonic Oscillator

$\frac{dT}{dt} = k(T - T_m)$ Newton's Law of Cooling

$\frac{d^2y}{dx^2} = k\sqrt{1 + \left(\frac{dy}{dx}\right)^2}$ Shape of a hanging string

$\frac{dP}{dt} = kP$ Population Growth

$\frac{dP}{dt} = P(a - bP)$ Population Growth (limited resources)

$\frac{dx}{dt} = kxy$ Spread of Disease

$\left.\begin{aligned}\frac{dy}{dt} &= y(\alpha - \beta x)\\ \frac{dx}{dt} &= x(-\gamma + \delta y)\end{aligned}\right\}$ Predator-Prey

Solving nonlinear differential equations

Approximation techniques

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

Solving nonlinear differential equations

Macroscopic solvers

Deterministic Classical Solvers (Spectral & Grid-/mesh-based methods

Variational Classical Solvers (Physics informed neural networks)

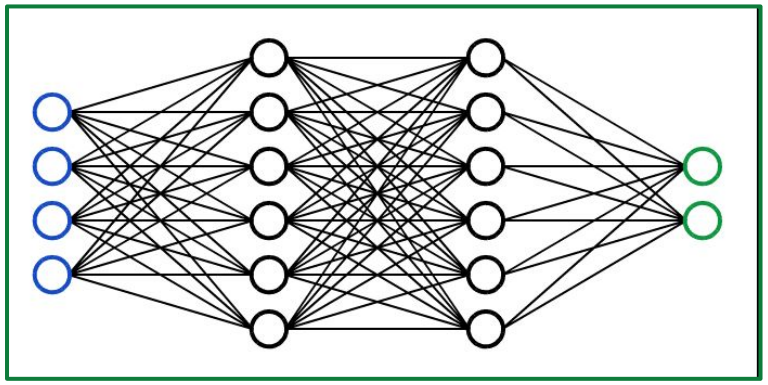
Deterministic Quantum Solvers (HHL kind of algorithms)

Variational Quantum Solvers (Differentiable quantum circuits → **This lesson**)

Solving nonlinear differential equations

Neural network based (classical) solvers

x



$f(x)$

$$\frac{\mathrm{d}f}{\mathrm{d}x} = f^2 + \sin(f) + x$$

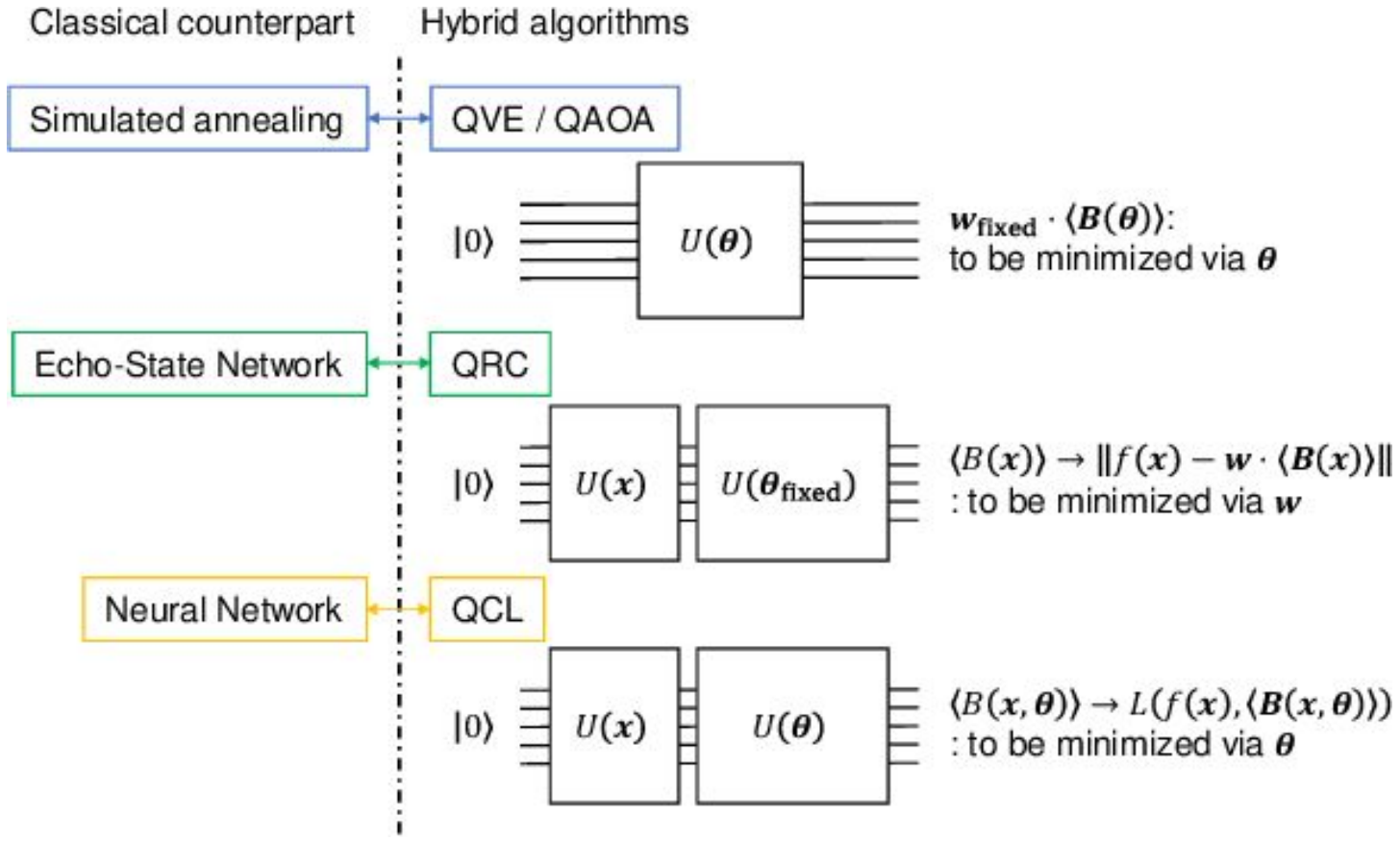
$$f(0) = 5$$

?

Update NN weights

Solving nonlinear differential equations

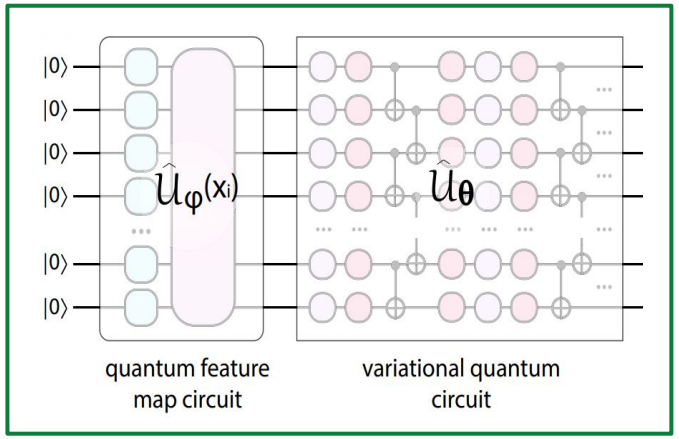
Quantum Circuit Learning



Solving nonlinear differential equations with DQC

Overview of DQC

x



$$f(x) = \langle f_{\varphi,\theta}(x)|\hat{C}|f_{\varphi,\theta}(x)\rangle$$

$$\frac{\mathrm{d}f}{\mathrm{d}x} = f^2 + \sin(f) + x$$

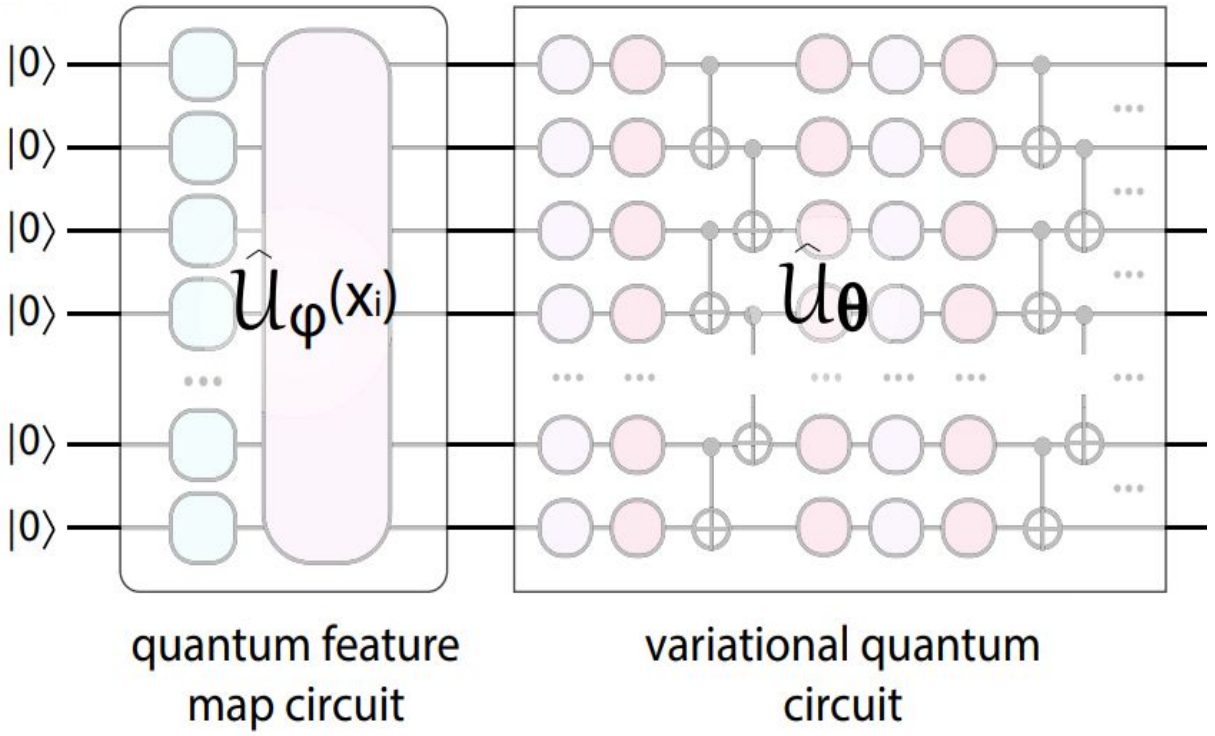
$$f(0) = 5$$

?

Update variational parameters θ

Solving nonlinear differential equations with DQC

Overview of DQC

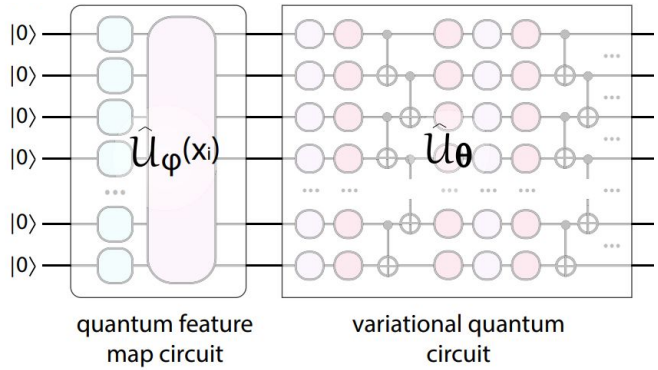


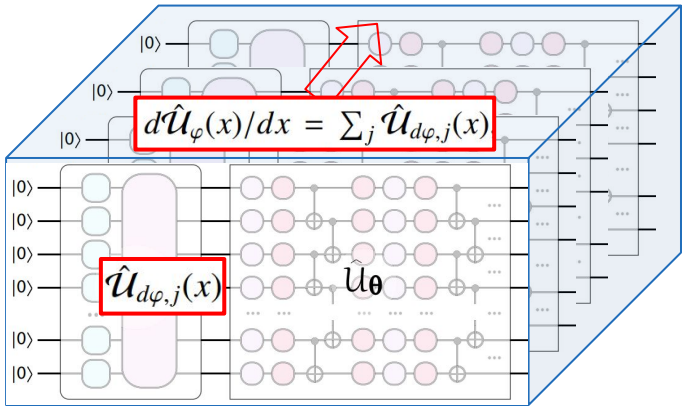
Solving nonlinear differential equations with DQC

Circuit differentiation

DQC aims applies circuit differentiation to both the quantum feature and the variational form (the former represents the function derivatives analytically, the later helps for efficient training the quantum neural network):

$$f(x) = \langle f_{\varphi,\boldsymbol{\theta}}(x)|\hat{C}|f_{\varphi,\boldsymbol{\theta}}(x)\rangle$$





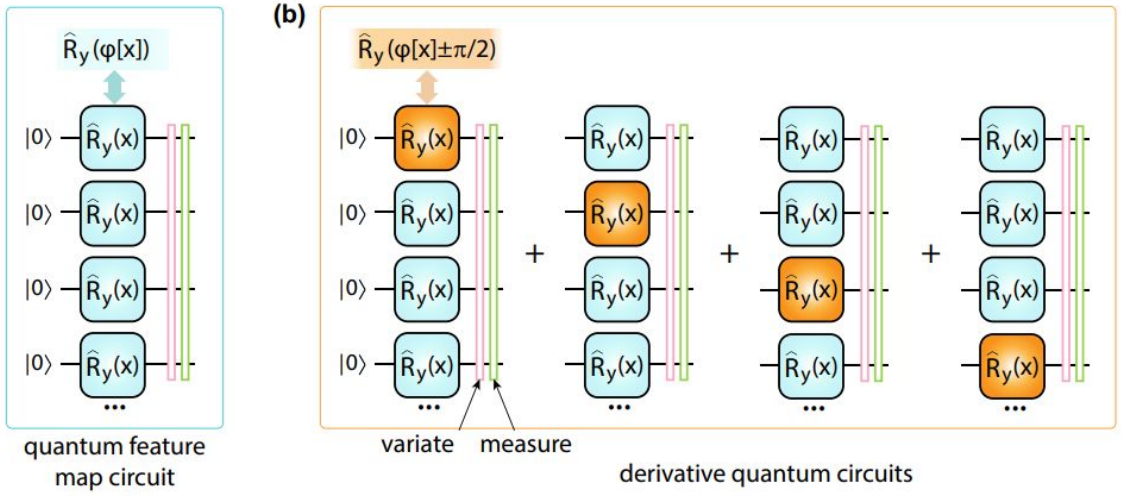
Circuit differentiation (example: parameter shift rule):

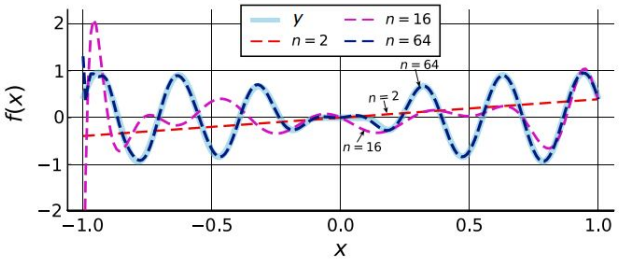
$$df(x)/dx = \frac{1}{2}\sum_j \Big(\langle f^{+}_{d\varphi,j,\boldsymbol{\theta}}(x)|\hat{C}|f^{+}_{d\varphi,j,\boldsymbol{\theta}}(x)\rangle - \langle f^{-}_{d\varphi,j,\boldsymbol{\theta}}(x)|\hat{C}|f^{-}_{d\varphi,j,\boldsymbol{\theta}}(x)\rangle\Big),$$

Solving nonlinear differential equations with DQC

Quantum feature map

Example(Chebyshev-type quantum feature map):





$$\hat{\mathcal{U}}_\varphi(x) = \bigotimes_{j=1}^{N} \hat{R}_{y,j}(2n[j]\arccos x)$$

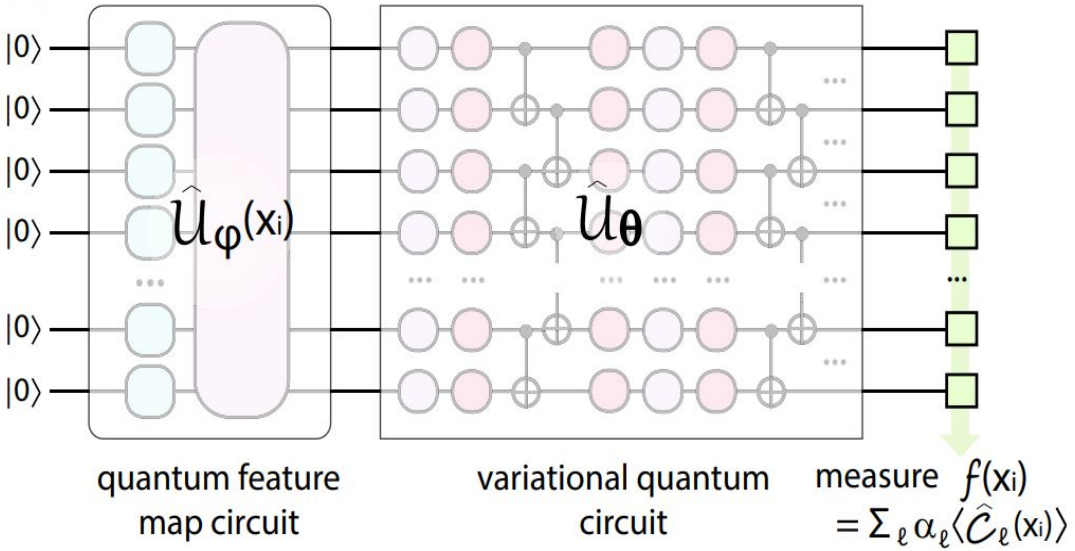
Chebyshev polynomials of first and second kind of order n

$$\begin{aligned}\hat{R}_{y,j}(\varphi[x]) &= \exp\left(-i\frac{2n\arccos(x)}{2}\hat{Y}_j\right)\\ &= \cos(n\arccos(x))\mathbb{1}_j - i\sin(n\arccos(x))\hat{Y}_j\\ &= T_n(x)\mathbb{1}_j + \sqrt{1-x^2}U_{n-1}(x)\hat{X}_j\hat{Z}_j.\end{aligned}$$

Solving nonlinear differential equations with DQC

Cost function

To access f(x) we devise a cost function C which uniquely and discriminately maps distinct quantum states to distinct expectation values:



To make sure each x maps to a unique f(x) via a rich feature map, it is best to use a cost function that contains many distinct eigenvalues, i.e. a rich spectrum

Solving nonlinear differential equations with DQC

Loss function

Like classical neural network solvers, DQC employs a loss function to quantify how good the trial solution 'solves' the differential equation optimization:

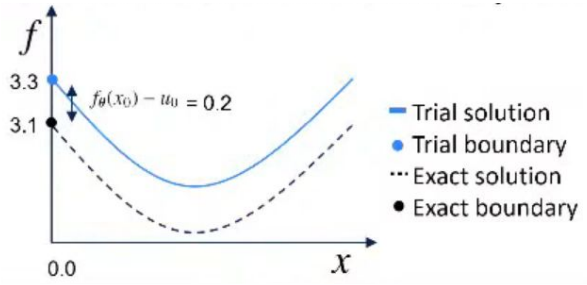
$$\mathcal{L}_{\boldsymbol{\theta}}[d_x f, f, x] = \mathcal{L}_{\boldsymbol{\theta}}^{(\text{diff})}[d_x f, f, x] + \mathcal{L}_{\boldsymbol{\theta}}^{(\text{boundary})}[f, x]$$

- Loss function from matching the differentials:

$$\mathcal{L}_{\boldsymbol{\theta}}^{(\text{diff})}[d_x f, f, x] = \frac{1}{M}\sum_{i=1}^{M} L(F[d_x f(x_i), f(x_i), x_i], 0)$$

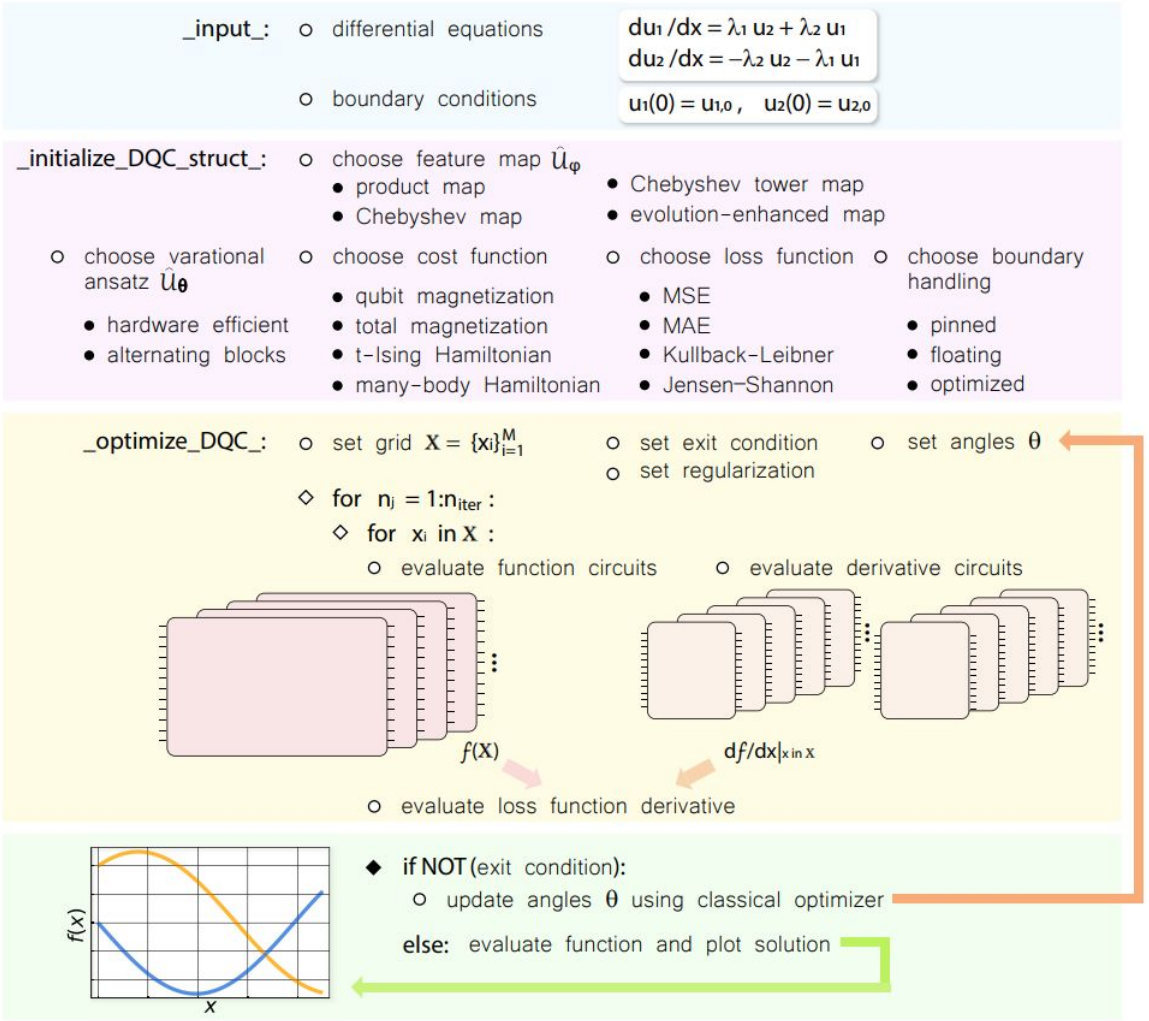
- Loss contribution from satisfying the boundary conditions:

$$\mathcal{L}_{\boldsymbol{\theta}}^{(\text{boundary})}[f, x] = \eta L(f(x_0), u_0)$$



Solving nonlinear differential equations with DQC

Summary of DQC:



Conclusion

- ➔ 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