# Computational battery modelling with differentiable quantum circuits

Ignacio Fernández Graña, Nicolò Toscano, Smit Chaudhary, Lorenzo Cardarelli, Andrea A. Gentile

Pasqal SAS, 7 Rue Léonard de Vinci,91300 Massy, France

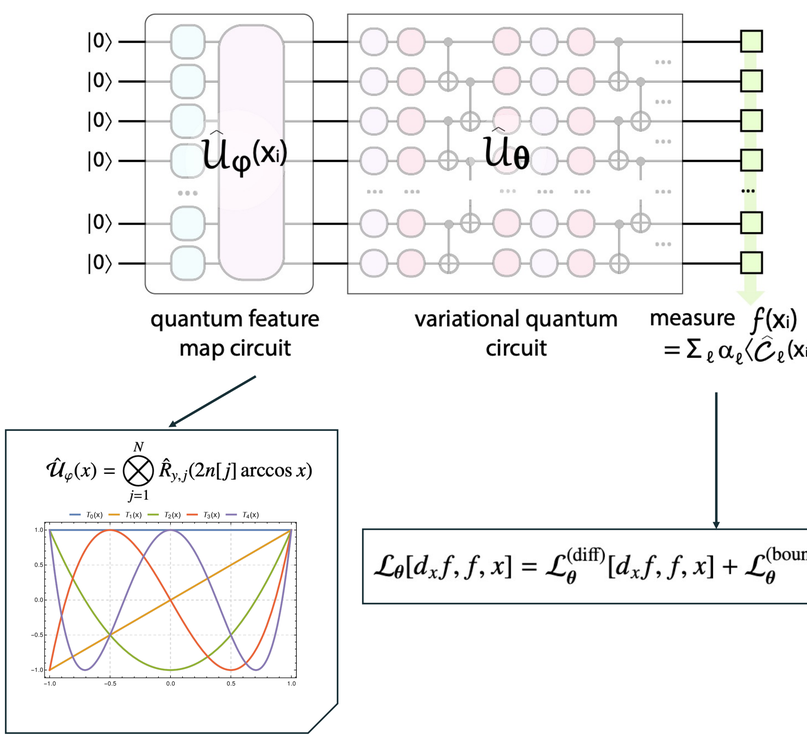
## Introduction

Many scientific problems are formulated in the form of **Partial Differential Equations** (PDEs). The most popular approach to solve these systems are mesh-based methods, which are powerful but complex and with severe scaling issues. Physics-Informed Neural Networks (PINNs) are part of the emerging field of **Scientific Machine Learning** (SciML) and are proposed as an alternative paradigm, using a NN as universal function approximator (UFA) to model the solution of the target PDE. The key advantages of this approach are its **flexibility** and **inference capabilities**, and they offer the opportunity to directly embed system properties and use experimental data for regularization purposes.

This research showcases the use of Differentiable Quantum Circuits (DQC), a **hybrid quantum-classical algorithm** inspired from PINNs, applied to solving systems of PDEs relevant to battery simulation. We validate the algorithm's accuracy in generating **solutions for multiple variables simultaneously**. Given the **high expressivity** provided by its large basis-set (which scales exponentially with the register size), and the **low depth** required to converge, this algorithm can perform well on **complex tasks** that are challenging for its classical counterparts while remaining NISQ-compatible.

## Differentiable quantum circuits

DQC[1] is a hybrid quantum-classical algorithm that can be used to solve systems of nonlinear differential equations. The trial solution is encoded in the expectation value of an observable in a parametrized quantum circuit.



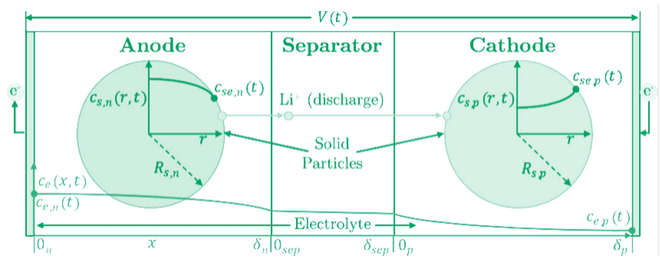

Its main components are:

• The **feature map** (FM) encoding. Input data is encoded in the quantum circuit via a non-linear quantum feature map. The FM derivatives can be exactly calculated through automatic differentiation. The choice for a specific FM depends on the nature of the problem and leads to different spectral representations. Trainable parameters can be used to improve flexibility and convergence[2].

• **Variational quantum circuit**. Used to manipulate the latent space basis function and improve the circuit's expressive power.

• The **loss function**. Measures the quality of the solution generated by the quantum circuit. When solving systems of differential equations, the loss function measures how well the PDE and boundary conditions terms are satisfied.

## SPMe battery model




***Solid phase***

$$\mathcal{C}_{\mathrm{k}} \frac{\partial c^0_{\mathrm{s,k}}}{\partial t} = -\frac{1}{r_{\mathrm{k}}^2} \frac{\partial}{\partial r_{\mathrm{k}}} \left( r_{\mathrm{k}}^2 \frac{\partial c^0_{\mathrm{s,k}}}{\partial r_{\mathrm{k}}} \right), \qquad \mathrm{k} \in \{\mathrm{n}, \mathrm{p}\}$$

***Electrolyte phase***

$$\mathcal{C}_{\mathrm{e}} \epsilon_{\mathrm{k}} \gamma_{\mathrm{e}} \frac{\partial c^1_{\mathrm{e,k}}}{\partial t} = -\gamma_{\mathrm{e}} \frac{\partial N^1_{\mathrm{e,k}}}{\partial x} + \begin{cases} \frac{I}{L_{\mathrm{n}}}, & \mathrm{k} = \mathrm{n}, \\ 0, & \mathrm{k} = \mathrm{s}, \\ -\frac{I}{L_{\mathrm{p}}}, & \mathrm{k} = \mathrm{p}, \end{cases} \qquad \mathrm{k} \in \{\mathrm{n}, \mathrm{s}, \mathrm{p}\},$$

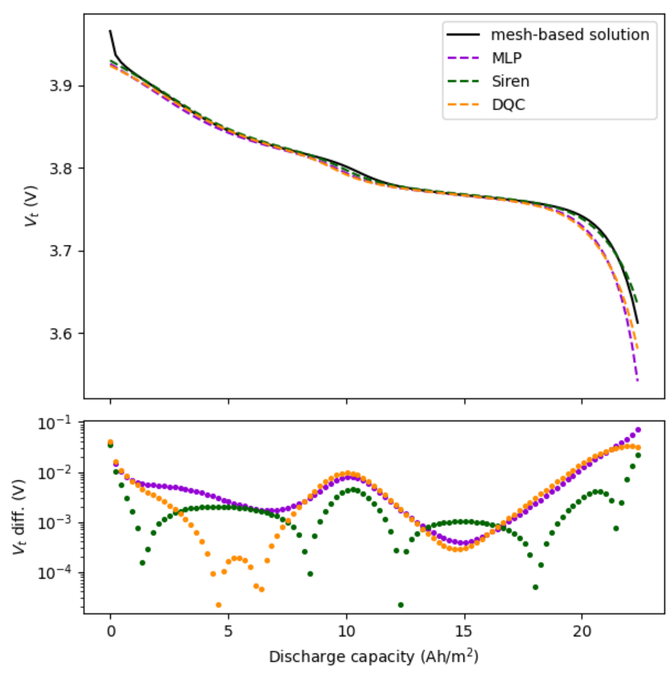
$$N^1_{\mathrm{e,k}} = -\epsilon^{\mathrm{b}}_{\mathrm{k}} D_{\mathrm{e}}(1) \frac{\partial c^1_{\mathrm{e,k}}}{\partial x} + \begin{cases} \frac{x t^+ I}{\gamma_{\mathrm{e}} L_{\mathrm{n}}}, & \mathrm{k} = \mathrm{n}, \\ \frac{t^+ I}{\gamma_{\mathrm{e}}}, & \mathrm{k} = \mathrm{s}, \\ \frac{(1-x) t^+ I}{\gamma_{\mathrm{e}} L_{\mathrm{p}}}, & \mathrm{k} = \mathrm{p}, \end{cases} \qquad \mathrm{k} \in \{\mathrm{n}, \mathrm{s}, \mathrm{p}\}.$$

Both systems require a mix of Dirichlet and Neumann boundary conditions.

Solid- and electrolyte-phase concentrations are then injected into a simple algebraic expression to obtain the terminal voltage

***Terminal voltage at constant current discharge***

$$V = \overline{U}_{\mathrm{eq}} + \overline{\eta}_r + \overline{\eta}_c + \overline{\Delta\Phi}_{\mathrm{Elec}} + \overline{\Delta\Phi}_{\mathrm{Solid}}$$




We solve the Single Particle Model with electrolyte[3] (SPMe) for a Lithium-ion (Li-ion) battery. The SPMe is a **full-battery model** that captures the basic physics of the battery, simulating the behavior of lithium-ion concentration in the negative and positive electrodes as well as in the electrolyte.

We benchmark our DQC implementation against mesh-based solutions (produced with PyBamm) and classical PINNs built with different underlying architectures (MLP and Siren[4]).

Each **DQC** model is built with:

- 4-qubits register
- In-series Chebyshev tower FM

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

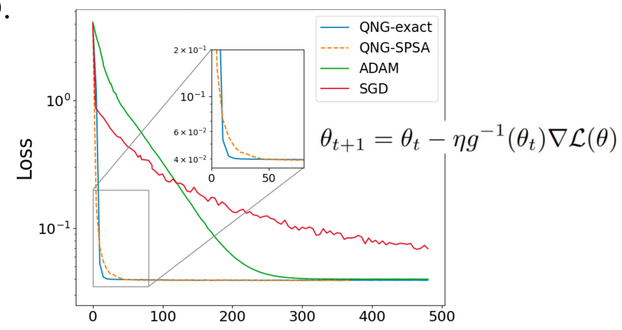
- 4 layers of hardware efficient ansatz

## DQC trainability

To investigate whether the loss landscape of DQC can be explored in an **efficient** way, we train the solid-phase model with different optimizers.

We find that standard gradient-based (GD) methods navigate very slowly compared to the **Quantum Natural Gradient** (QNG), which takes in account geometry of the parameter space.

We also test a SPSA approximation of the QNG which only requires and overhead of O(1) circuit evaluations compared to vanilla GD.




## Next steps

- Apply DQC to **more complex scenarios**, where classical mesh-based solvers and PINNs struggle to converge (e.g. high-dimensionality, multi-physics simulations).
- Enhance the capabilities of DQC by developing **novel feature maps encoding** techniques and improve **inductive bias** techniques for specific families of problems.
- Investigate **transfer learning** possibilities between different domains and problem specifications.

## References


- [1] Phys. Rev. A 103, 052416
- [2] Phys. Rev. A 109, 042421
- [3] J. Electrochem. Soc. 166 A3693
- [4] NeurIPS 33, 7462-7473