REUNIÓN: CIRCUITOS CUÁNTICOS & ECUACIONES DIFERENCIABLES ORDINARIAS

Dr. Carolina Perdomo
Quantum Fellow-QuantumQuipu

CONTENIDOS

01 **INTRODUCCIÓN**
Puntos principales.

02 **ALGORITMOS**
HHL y otros, implementación.

03 **DQC**
Circuitos cúánticos diferenciables, implementación.

01

INTRODUCCIÓN

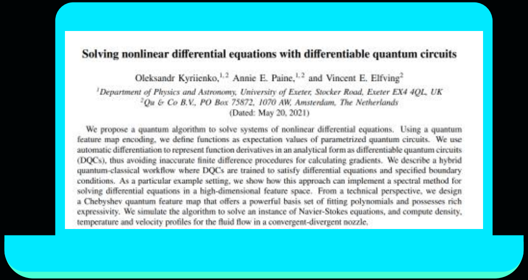
NISQ y el desafío de las ED.

Noisy Intermediate-Scale Quantum (NISQ) computer
Ecuaciones Diferenciables (ED)

Puntos principales

Hoy:

1. Flujo de trabajo.
2. Algoritmos (HHL, VQA, DQC, etc).
3. Diferentes ejemplos para mostrar la adaptabilidad del algoritmo DQC.



Solving nonlinear differential equations with differentiable quantum circuits

Oleksandr Kyriienko,[1,2] Annie E. Paine,[1,2] and Vincent E. Elfving[2]

[1]*Department of Physics and Astronomy, University of Exeter, Stocker Road, Exeter EX4 4QL, UK*
[2]*Qu & Co B.V., PO Box 75872, 1070 AW, Amsterdam, The Netherlands*

(Dated: May 20, 2021)

We propose a quantum algorithm to solve systems of nonlinear differential equations. Using a quantum feature map encoding, we define functions as expectation values of parametrized quantum circuits. We use automatic differentiation to represent function derivatives in an analytical form as differentiable quantum circuits (DQCs), thus avoiding inaccurate finite difference procedures for calculating gradients. We describe a hybrid quantum-classical workflow where DQCs are trained to satisfy differential equations and specified boundary conditions. As a particular example setting, we show how this approach can implement a spectral method for solving differential equations in a high-dimensional feature space. From a technical perspective, we design a Chebyshev quantum feature map that offers a powerful basis set of fitting polynomials and possesses rich expressivity. We simulate the algorithm to solve an instance of Navier-Stokes equations, and compute density, temperature and velocity profiles for the fluid flow in a convergent-divergent nozzle.

DQC: Differential Quantum Circuit [arxiv.org/abs/2011.10395

¡GUAU!

- Era de los primeros ordenadores cuánticos disponibles.
- Imperfectos: NISQ.
- **Computación híbrida cuántica/clásica.**

EJEMPLOS

Boosting advancements in quantum chemistry

Pasqal and Rahko will jointly develop algorithms to solve advanced chemistry problems, and fully implement them on Pasqal's upcoming

Accelerating High-Performance Computing

Through a collaboration with Atos, the global leader in digital transformation, Pasqal is striving to incorporate its quantum processors into high-performance computing environments, thereby setting the stage for an era of hybrid quantum-HPC systems with potential for short-term, real-world applications.

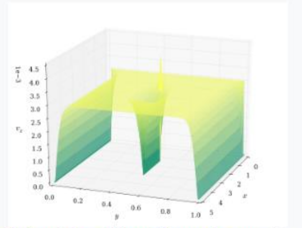
¿POR QUÉ QUEREMOS ESTUDIAR ED?

Muchos modelos industriales pueden modelarse con ED.

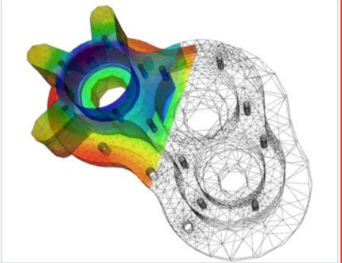
- ¿Cuáles? Tasas de cambios, leyes de conservación, etc.

Las ED describen la física y las reglas del sistema.

Simulaciones físicas



Navier–Stokes differential equations used to simulate airflow around an obstruction



Visualization of heat transfer in a pump casing, created by solving the heat equation. Heat is being generated internally in

Procesos gobernados por ED

Transferencia de calor,
flujos de aire o fluidos,
tensiones, esfuerzos,
reacciones químicas, etc.

¡Vamos a practicar!

$$I(\tau) = Cg\,\frac{\mathrm{d}V_g}{\partial\tau} + g_k n^4\left(V_g - V_k\right) + g_{N_a} m^3 h\left(V_g - V_{N_a}\right) + g_L\left(V_g - V_L\right)$$

¿Por qué es relevante?
¿Por qué me interesa su solución?

SOBRE ED:

ED lineales

Relación lineal.
Ejemplo: oscilador armónico.

ED no lineales

No lineal, más complejo.

EDO

Una sola variable independiente.

$$LV = *$$

operador diferencial

$$L(AV_1 + V_2) = ALV_1 + LV_2$$

$$L = a_n(x)\frac{d^n}{dx^n} + a_{n-1}(x)\cdot\frac{d^{n-1}}{dx^{n-1}} + \cdots$$

$$L(Y_1 + Y_2) = LY_1 + LY_2$$

¿$Y'' + KY$? ¿$Y'' + \boxed{XY}$?

a_0

LXY

$L[Y'' + KY]$

$= L[(Y_1 + Y_2)''$

$+ K(Y_1 + Y_2)]$

$= L(Y_1 + Y_2)''$

$+ KL(Y_1 + Y_2)$

¡Vamos a practicar!

$$I(\tau) = Cg\frac{\mathrm{d}V_g}{\partial\tau} + g_k n^4(V_g - V_k) + g_{N_a} m^3 h(V_g - V_{N_a}) + g_L(V_g - V_L)$$

¿Lineal o no?
¿Por qué?

COMPUTACIÓN CUÁNTICA Y ED

1

Encontrar soluciones a sistemas no lineales de ED es todo un reto.

2

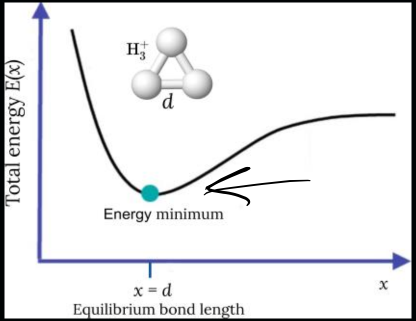
Los ordenadore cuánticos ofrecen un enfoque fundamentalmente diferente para realizar.

3

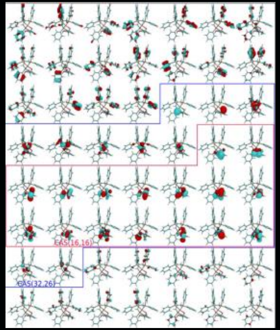
Encontrar protocolos NISQ que puedan ofrecer ventajas para tareas de relevancia industrial representa un reto abierto.

USOS POTENCIALES DE QC PARA ED

INGIENERÍA QUÍMICA



CLIMA



Quantum computing enhanced computational catalysis arXiv:2007.14460

Muchos problemas relevantas para la industria pueden modelizarse mediante ecuaciones diferenciales. Algunas de estas ecuaciones suelen ser más complejas de resolver con los ordenadores clásicos.

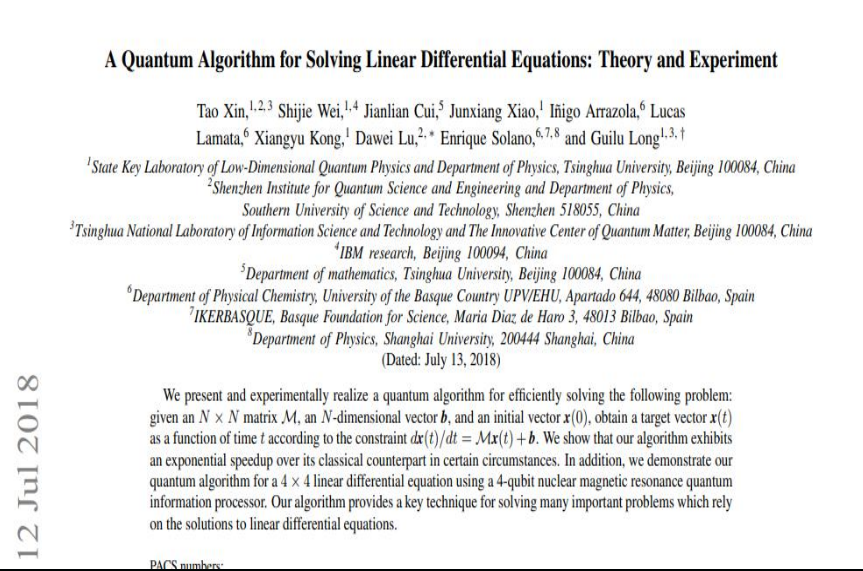
—**RESUMEN**

02

ALGORITMOS

¡Hora de aventura!

Algoritmo para ecuaciones lineales



A Quantum Algorithm for Solving Linear Differential Equations: Theory and Experiment

Tao Xin,[1,2,3] Shijie Wei,[1,4] Jianlian Cui,[5] Junxiang Xiao,[1] Iñigo Arrazola,[6] Lucas Lamata,[6] Xiangyu Kong,[1] Dawei Lu,[2,*] Enrique Solano,[6,7,8] and Guilu Long[1,3,†]

[1]*State Key Laboratory of Low-Dimensional Quantum Physics and Department of Physics, Tsinghua University, Beijing 100084, China*
[2]*Shenzhen Institute for Quantum Science and Engineering and Department of Physics, Southern University of Science and Technology, Shenzhen 518055, China*
[3]*Tsinghua National Laboratory of Information Science and Technology and The Innovative Center of Quantum Matter, Beijing 100084, China*
[4]*IBM research, Beijing 100094, China*
[5]*Department of mathematics, Tsinghua University, Beijing 100084, China*
[6]*Department of Physical Chemistry, University of the Basque Country UPV/EHU, Apartado 644, 48080 Bilbao, Spain*
[7]*IKERBASQUE, Basque Foundation for Science, Maria Diaz de Haro 3, 48013 Bilbao, Spain*
[8]*Department of Physics, Shanghai University, 200444 Shanghai, China*

(Dated: July 13, 2018)

12 Jul 2018

We present and experimentally realize a quantum algorithm for efficiently solving the following problem: given an $N \times N$ matrix $\mathcal{M}$, an N-dimensional vector $\boldsymbol{b}$, and an initial vector $\boldsymbol{x}(0)$, obtain a target vector $\boldsymbol{x}(t)$ as a function of time t according to the constraint $d\boldsymbol{x}(t)/dt = \mathcal{M}\boldsymbol{x}(t) + \boldsymbol{b}$. We show that our algorithm exhibits an exponential speedup over its classical counterpart in certain circumstances. In addition, we demonstrate our quantum algorithm for a 4×4 linear differential equation using a 4-qubit nuclear magnetic resonance quantum information processor. Our algorithm provides a key technique for solving many important problems which rely on the solutions to linear differential equations.

PACS numbers:

arXiv:1807.04553v1 [quant-ph]

01

¿Qué hicieron?

Solucionador de ecuaciones lineales a pequeña escala.

02

¿Cómo?

4-qubits en un sistema de resonancia magnética nuclear.

$$\frac{d\boldsymbol{p}(\boldsymbol{t})}{d\tau} = M\mathbf{p}(\mathbf{t}) + b$$

lo conoces

No lo conoces

Lo conoces

¿Cuál es la intuición detrás de estos problemas?

Queremos encontrar $\underline{p}(\tau)$

¿Cómo encuentras la **solución analítica**?

1. Separar variables. → $\frac{d\underline{p}(\tau)}{d\tau} - M\underline{p}(\tau) = b$

2. Factor de integración: $\mu(t) = e^{-M\tau}$ → $e^{-M\tau}\frac{d}{d\tau}\underline{p}(\tau) - e^{-M\tau}M\underline{p}(\tau) = e^{-M\tau}b$

3. Multiplicamos ambos lados por ese factor.

4. Regla del producto de una derivada.

$\int d_\tau\left[e^{-M\tau}\cdot\underline{p}(\tau)\right] = \int e^{-M\tau}\, b\, d\tau$ Integramos ambos lados

Notación Carolina $\vec{A} = \underline{A}$

$$\boxed{\frac{d\boldsymbol{p}(\boldsymbol{t})}{d\tau} = M \times \mathbf{p}(\mathbf{t}) + b}$$

$$\rightarrow \int_0^\tau d_\tau \left[e^{-M\tau} \cdot \underline{p}(\tau) \right] = \int_0^1 e^{-M\tau}\, b\, d\tau$$

$$\rightarrow e^{-M\tau}\, \underline{p}(\tau) \Big|_0^\tau = -\frac{1}{M}\, e^{-M\tau}\, b \Big|_0^\tau$$

$$\left(\cancel{e^{-M\tau}}\, \underline{p}(\tau) - 1 \cdot \underline{p}(0) = -M^{-1} \cancel{e^{-M\tau}}\, b + M^{-1}\, b \right) \times e^{M\tau}$$

$$\rightarrow \boxed{\boldsymbol{p}(\boldsymbol{\tau}) = e^{M\tau} \boldsymbol{p}(\mathbf{0}) + (e^{M\tau} - I) M^{-1} \boldsymbol{b}}$$

$$\frac{d\boldsymbol{p}(\boldsymbol{t})}{d\tau} = M \times \mathbf{p}(\mathbf{t}) + b$$

$$\boldsymbol{p}(\boldsymbol{\tau}) = e^{M\tau}\boldsymbol{p}(\mathbf{0}) + (e^{M\tau} - I)M^{-1}\boldsymbol{b}$$

$\underline{P} = A^{-1}\underline{b}$

for N variables classically $O(N^2)$ by Gaussian elimination

Solución por expansión de taylor

orden aprox.

$$\underline{P}(\tau) \approx \sum_{m=0}^{K} \frac{(M\tau)^m}{m!} \underbrace{\underline{P}(0)} + \sum_{n=1}^{K} \frac{M^{n-1}\tau^n}{n!} \underline{b}$$

Operator (lo conocemos

Describir como un estado cuántico

$|P(0)\rangle ; |b\rangle$ (los conocemos)

Encode

¿Cuál es la intuición para **construir la solución** de estos problemas?

$$\boxed{\frac{\mathrm{d}\boldsymbol{p}(\boldsymbol{t})}{d\tau} = M \times \mathbf{p}(\mathbf{t}) + b}$$

Objetivo

Construir: $|p(t)\rangle \approx \sum_{m=0}^{K} \frac{\|p(0)\| (\|M\| A t)^m}{m!} |p(0)\rangle + \sum_{n=1}^{K} \frac{\|b\| (\|M\| A)^{n-1} t^n}{n!} |b\rangle$

operador

proceso

encoding

¿Es posible?

¿Cómo es A? ¿Hermítico? ¿Unitario? ¿Cómo leemos el output?

Vamos a actuar en $|x(0)\rangle$ y $|b\rangle$ por un proceso dinámico relacionado con el operador A

Para leer el output necesitamos evaluar los valores de expectación

evaluación

$$\frac{d\mathbf{p}}{d\tau} = \underbrace{M}_{\text{Matriz (arbitraria } N\times N)} \times \underbrace{\mathbf{p0}}_{\text{vector en el estado inicial}} + \mathbf{b}$$

$\mathbf{b}$ → otro vector

$p(\tau)$

Vector representando el estado de un sistema en un tiempo τ

No lo conocemos.

¿Qué significa esta ecuación?

Tasa de cambio del Vector p.

(1) Encoding: $\log_2 N$ qubits son necesarios para N vectores dimensionales.

$|p(0)\rangle$; $|b\rangle$

Notación Carolina $\vec{A} = \underline{A}$

→ Prepara qubits en un estado específico para representar el problema.

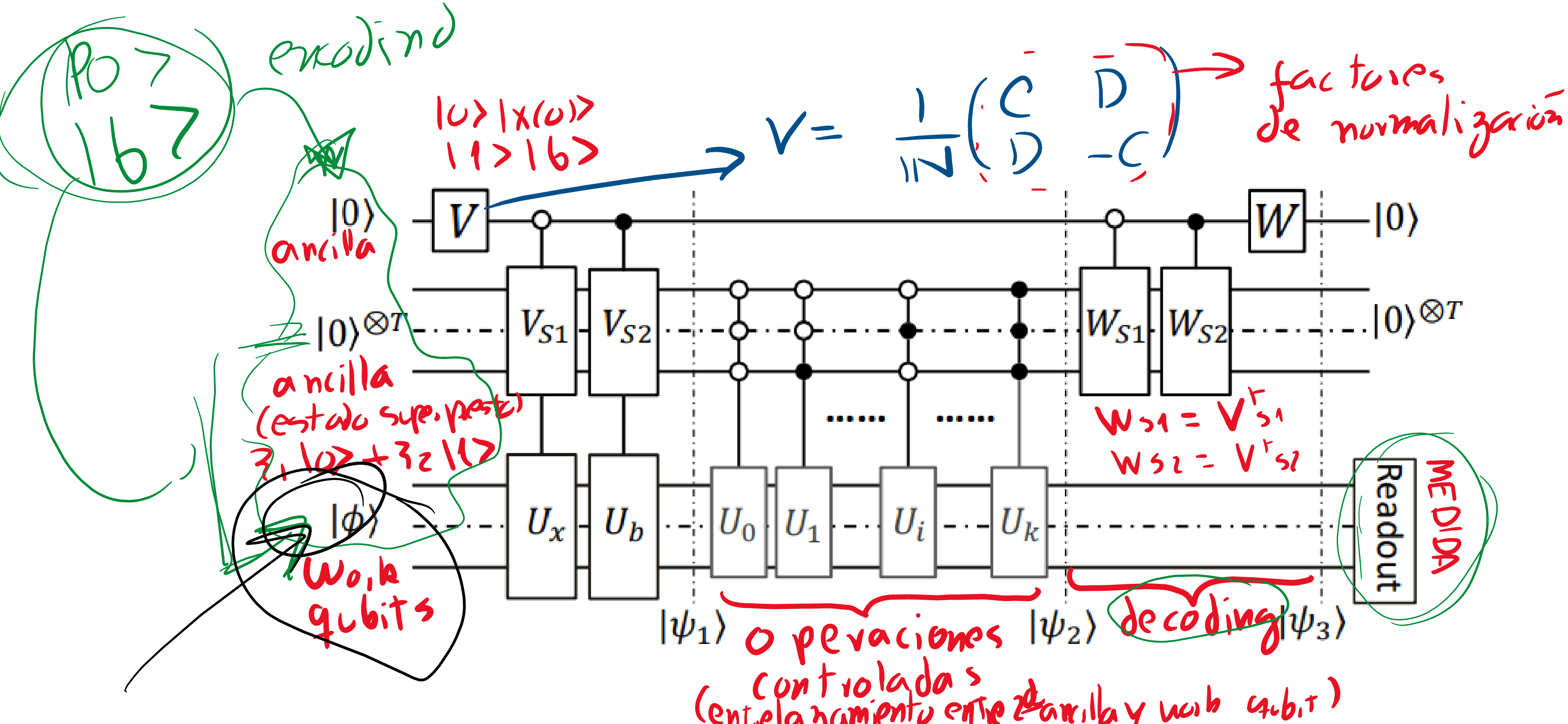


Figure 1: Quantum circuit for solving LDEs when A is unitary. $|\phi\rangle$ denotes the initial state of the work system, and $T = \log_2(k+1)$. The controlled operations U_x and U_b are used to create $|x(0)\rangle$ and $|b\rangle$, respectively. The

¿De dónde vienen los errores?

EJEMPLO
IMPLEMENTACIÓN

$$\frac{dp}{d\tau} = \underbrace{M}_{\text{Matriz}} \times \underbrace{p0}_{\text{vector en el estado inicial}}^{N} + b_{N}$$

otro vector $b = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$

$p(\tau)$

Vector representando el estado de un sistema en un tiempo τ

$p0 \rightarrow |p_0\rangle$

$b \rightarrow |b\rangle$

$M = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$

$p_0 = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$

¿Qué significa esta ecuación?

Tasa de cambio del vector p, determinada por M, comenzando en el estado p_0 con la adición de b.

(i) Encoding. $\log_2 N$ work qubits are needed to encode the N-dimensional vectors. $|x(0)\rangle$ and $|b\rangle$ are prepared

A is unitary: In order to solve an LDE where matrix A is unitary, we need a composite quantum system with a $(1+T)$-qubit ancilla register and a $\log_2(N)$-qubit work system. Suppose the input state of the work system is $|\phi\rangle$ and all ancilla qubits are

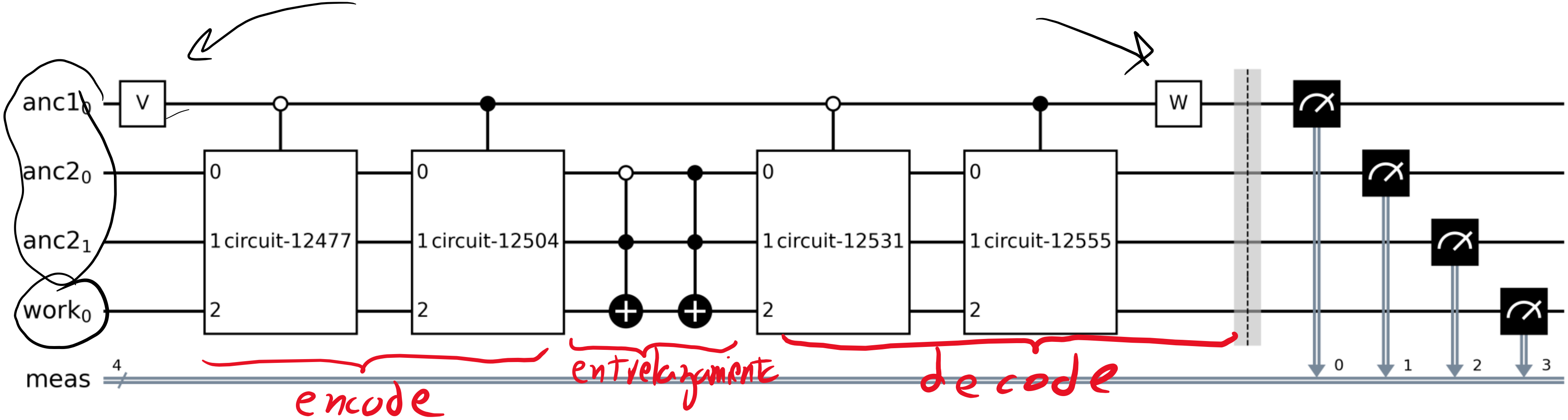
$\rightarrow 1+k$ (Taylor)

$\log_2(2) = 1$

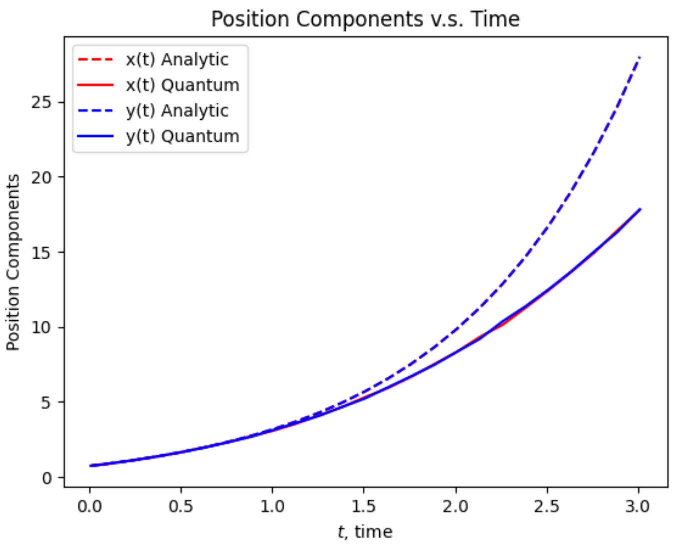
3

$1 + T \rightarrow 1 + 1 + k$

$\rightarrow 2$



CONSTRUCCIÓN



¿QUÉ ES IMPORTANTE RECORDAR?

1

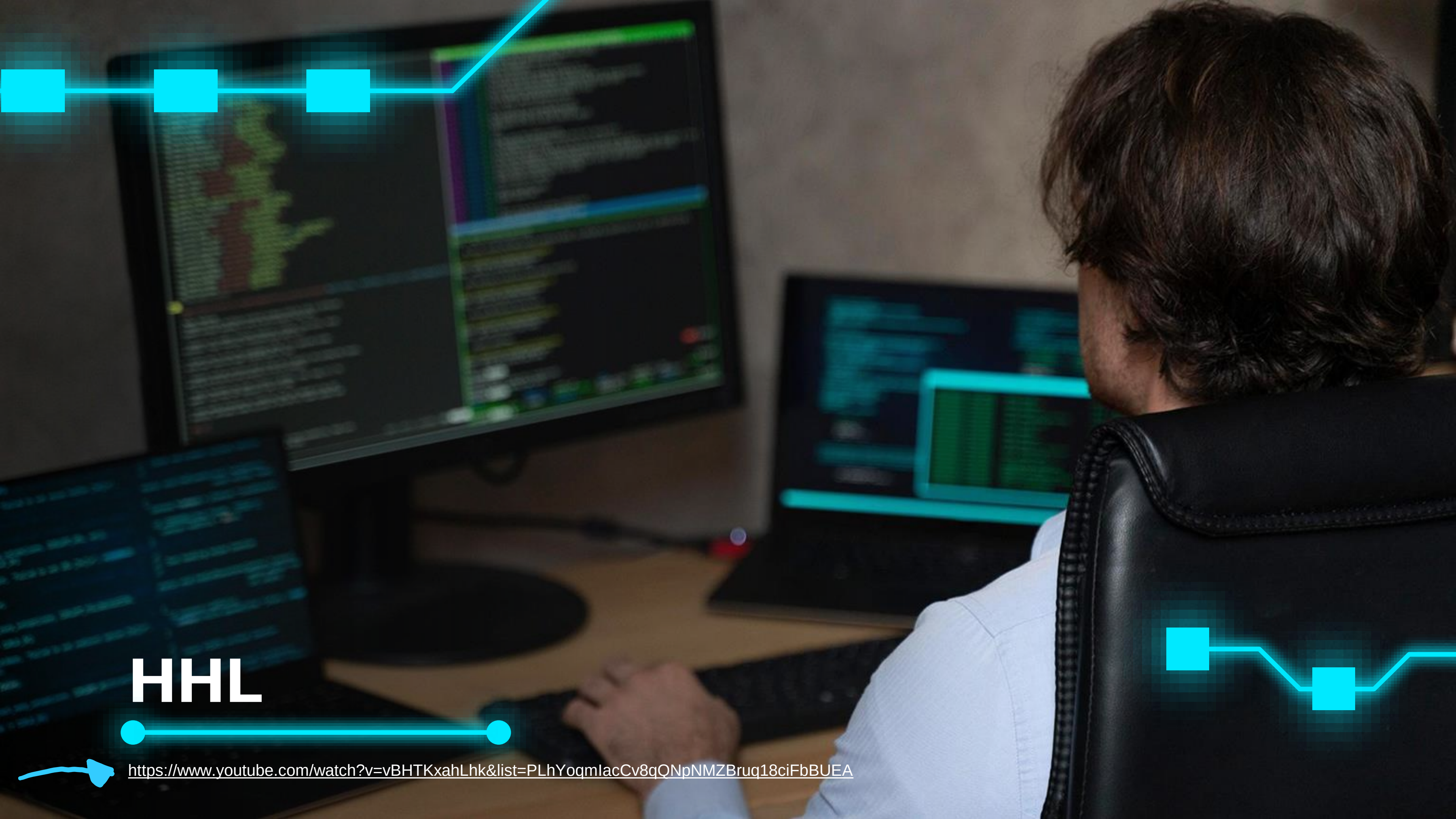
Construyendo una solución con la computadora cuántica.

2

Encoding: cómo introducir los vectores que ya conozco al circuito cuántico.

3

El uso de las compuertas para mapear la expansión de Taylor.



HHL

https://www.youtube.com/watch?v=vBHTKxahLhk&list=PLhYoqmIacCv8qQNpNMZBruq18ciFbBUEA

Sistema de Ecuaciones Lineales

$$\begin{cases} x' = 2x \\ y' = 2x + y \end{cases}$$

$$\begin{pmatrix} x \\ y \end{pmatrix}' = \begin{pmatrix} 2 & 0 \\ 2 & 4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

$$A\,\underline{x} = \underline{b}$$

La conoces

La conoces

$$A\,|x\rangle = |b\rangle$$

SISTEMAS DE ED LINEALES

1

Recordemos:

A|x>=|b>

$$|b\rangle = \sum_i b_i |u_i\rangle$$

$$A = \sum_i \underbrace{\lambda_i}_{\text{auto-valores}} \underbrace{|u_i\rangle\langle u_i|}_{\text{autovectores}}$$

2

¿Cuál es el objetivo?

Construir (basado en las restricciones):

$$\boxed{|x\rangle = \sum_i \frac{b_i}{\lambda_i}|u_i\rangle}$$

3

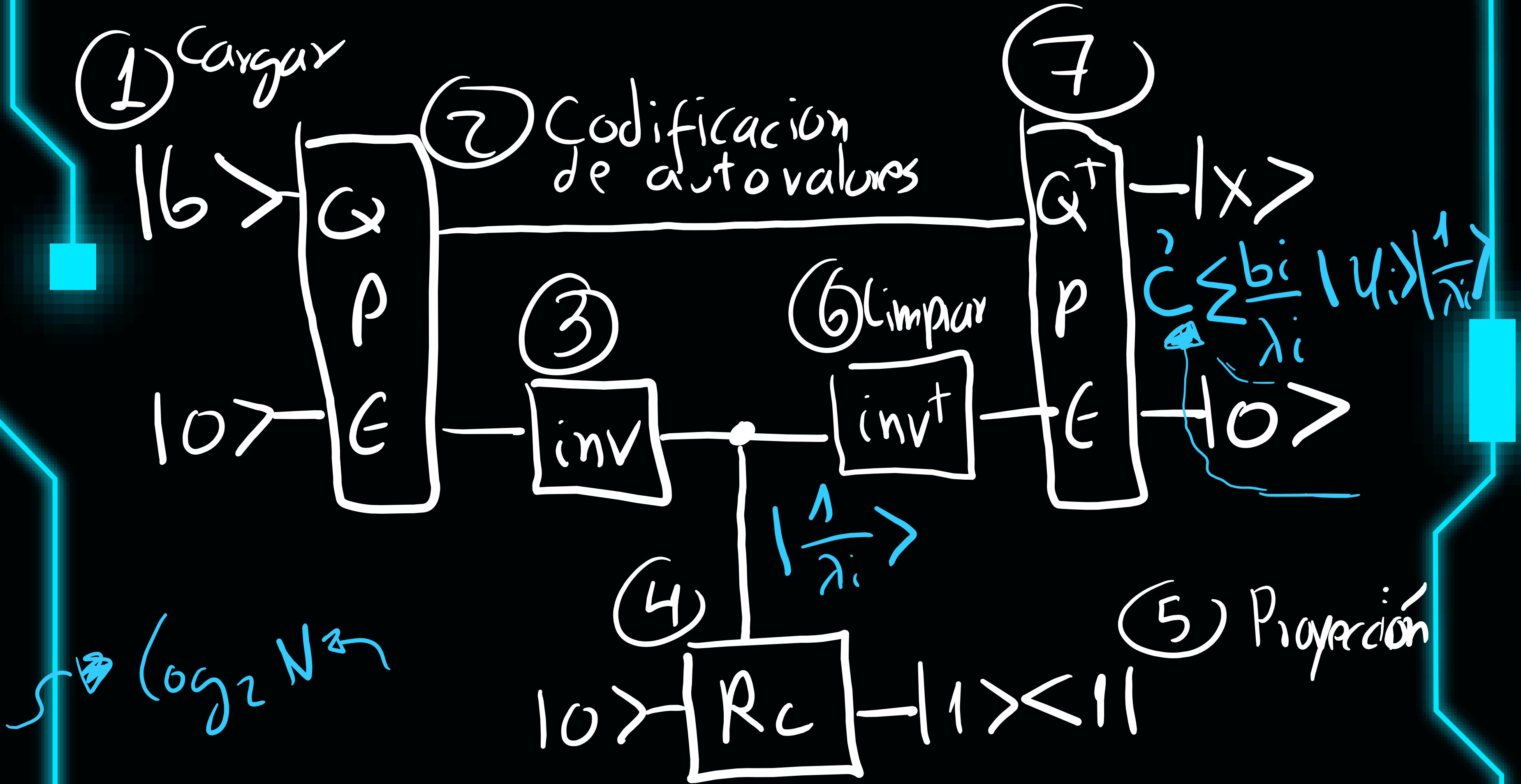
Restricciones:

A tiene que ser invertible (x es la solución única) y hermítica.

4

Pasos:

- Introducir |b> (**input**).
- Usar **Quantum Phase Estimation** sobre el estado |b>.
- Aplicar operador de inversión.
- Rotación condicionada sobre un qubit auxiliar.
- Preparar solución.



1 Cargar
2 Codificación de autovalores
|b>
QPE
|0>
3
inv
|0>
|1/λi>
4
Rc
|1><1|
5 Proyección
6 Limpiar
inv†
7
Q†PE
|x>
c Σ bi/λi |ui>|1/λi>
|0>
log2 N

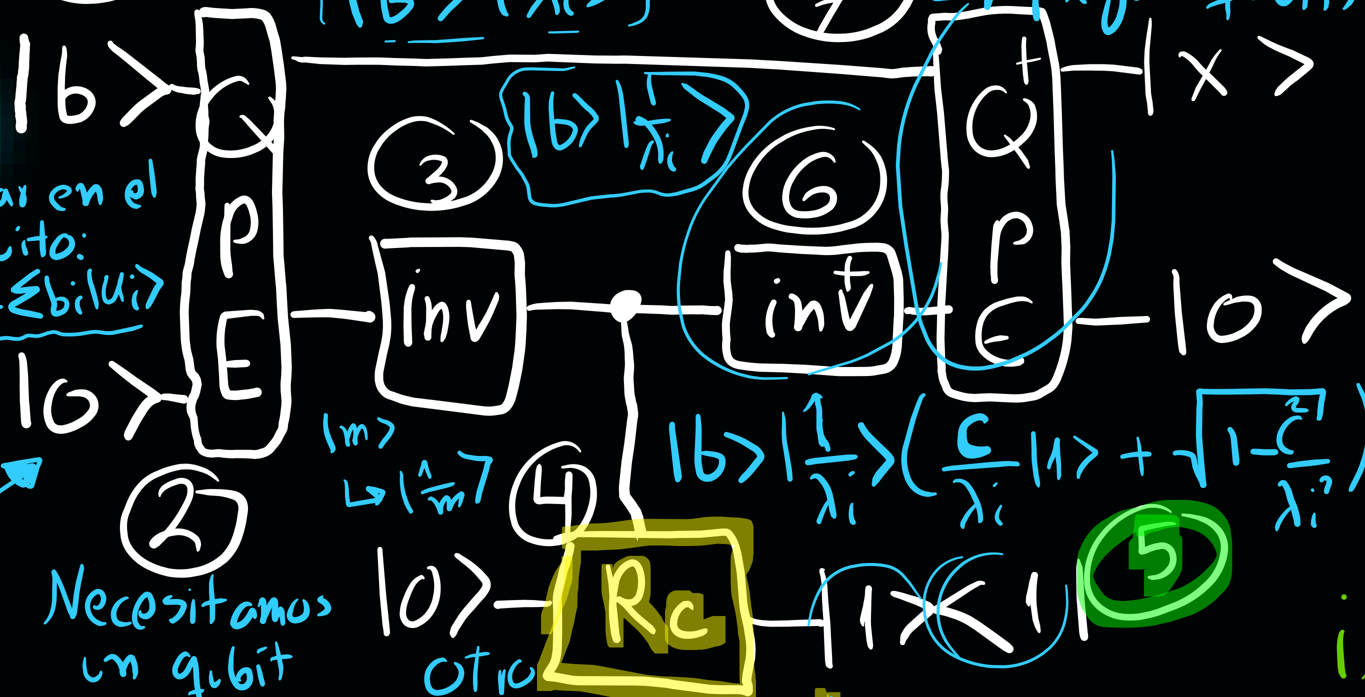
$$V = e^{iA} \Rightarrow \{e^{i\lambda_1}, \dots, e^{i\lambda_r}\}\ \{|u_1\rangle \dots |u_n\rangle\}\ (A = A^\dagger)$$

1

$|b\rangle|\lambda_i\rangle$

7 Limpieza qubits auxiliares.

Cargar en el circuito: $|b\rangle = \sum b_i |u_i\rangle$



$|b\rangle|\frac{1}{\lambda_i}\rangle$

$|m\rangle \hookrightarrow |\frac{1}{m}\rangle$

4

$$|b\rangle|\tfrac{1}{\lambda_i}\rangle\left(\frac{C}{\lambda_i}|1\rangle + \sqrt{1-\frac{C^2}{\lambda_i^2}}\,|0\rangle\right)$$

2 Necesitamos un qubit auxiliar $|b\rangle|0\rangle$

otro qubit auxiliar.

5

$|b\rangle|\tfrac{1}{\lambda_i}\rangle|0\rangle$

¡Medir!

Nos quedamos solo con los $|1\rangle$

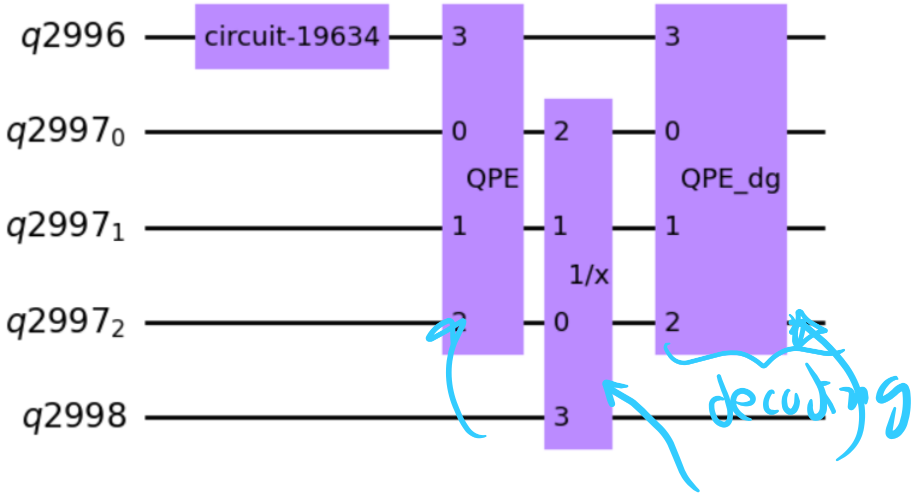
EJEMPLO
IMPLEMENTACIÓN

```python
hhl_solution = HHL().solve(matrix, vector) # We have HHL in qiskit, yay!
```

```
Classical solution: [ 0.86023308 -1.48719142]
Quantum solution:   [ 0.87786206 -1.46956242]
```

```python
problems = [
    {
        "A": np.array([
            [0.86540304, 0.35951908],
            [0.35951908, 0.86540304]
        ]),

        "b": np.array([ 0.20977463, -0.97774977])
    },
```

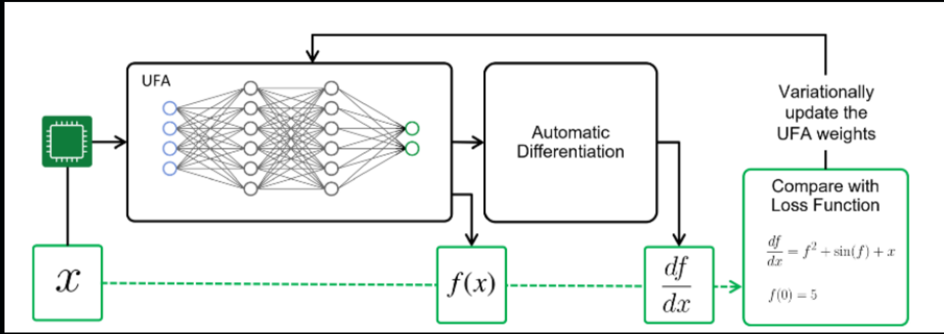


ED NO lineales

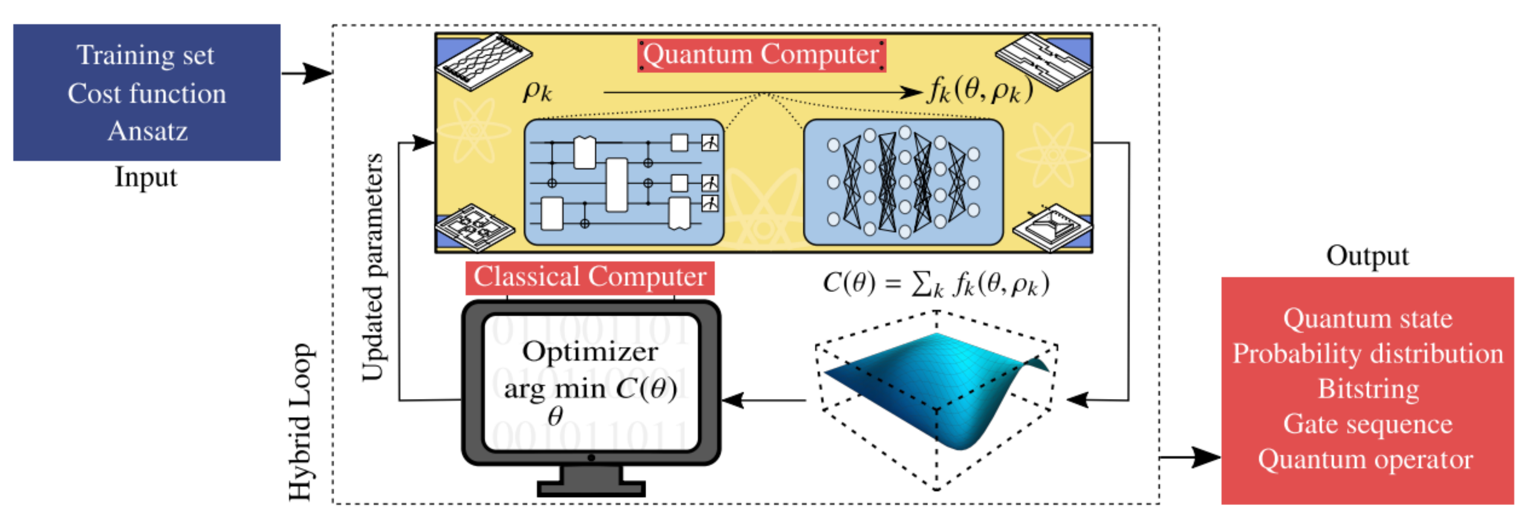
ALGORITMO DQC

CLÁSICO

Redes neuronales.



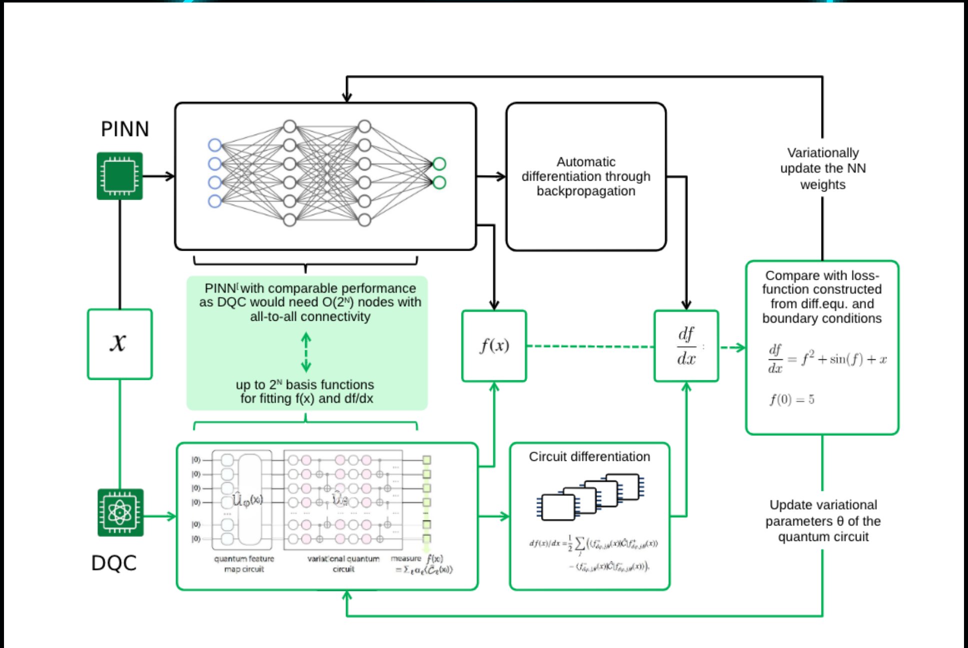
CONTEXTO TÉCNICO



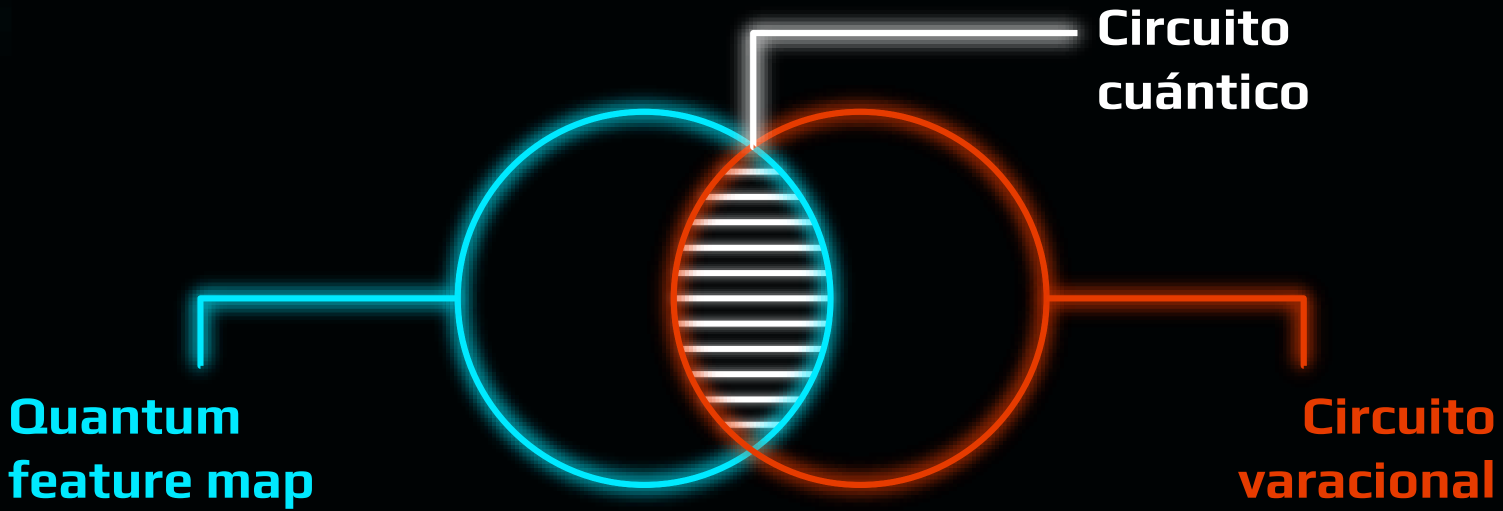
VQA

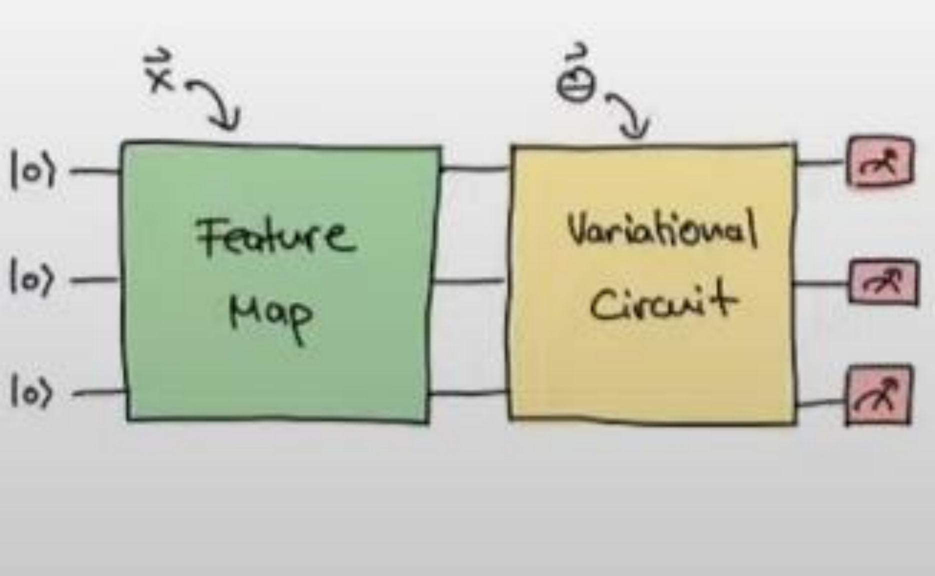
¿QUÉ PODEMOS APRENDER DE LAS REDES NEURONALES?

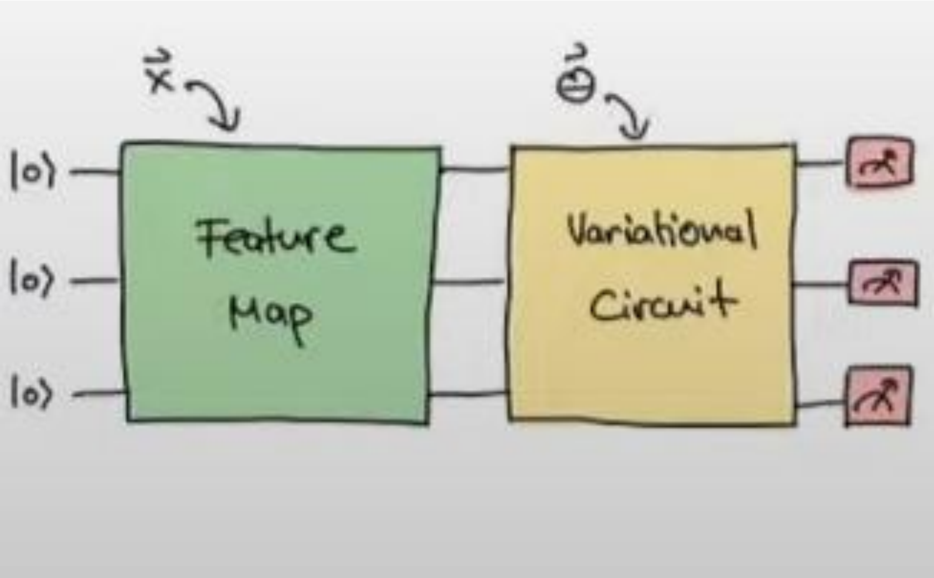
Algoritmos híbridos.



PASOS DE IMPLEMENTACIÓN

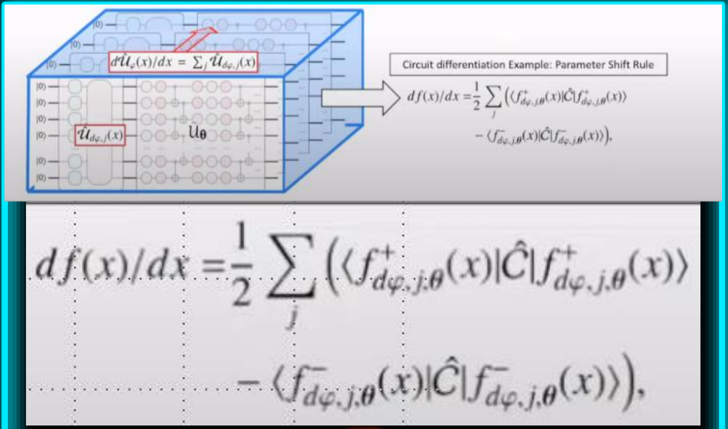



$\vec{x}$
$\vec{\Theta}$
$|0\rangle$
$|0\rangle$
$|0\rangle$
Feature Map
Variational Circuit



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

Uniquely maps distinct wavefunctions to distinct expectation values.



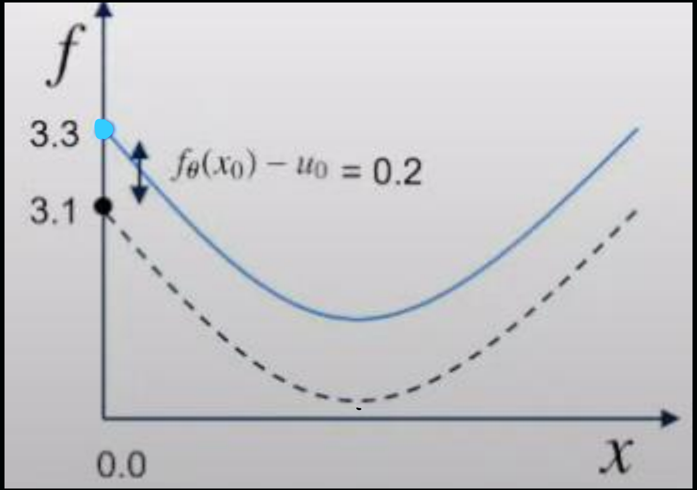
¿Y cómo hacemos las derivadas?

"The parameter-shift rule": Da el gradiente correcto hasta la precisión numérica posible.

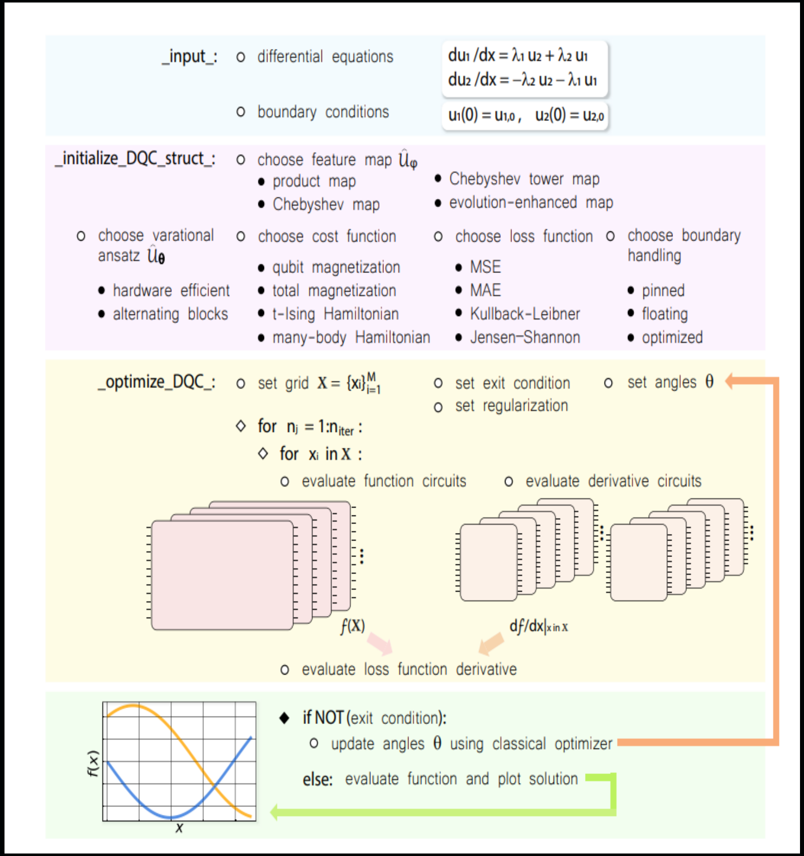
Función de pérdida

1

Queremos cuantificar qué tan buena es nuestra solución de prueba para "resolver" las ED.



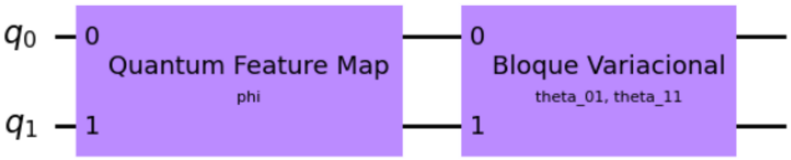
¡Pueden tener tantas versiones como quieran!



¿EJEMPLO IMPLEMENTACIÓN?

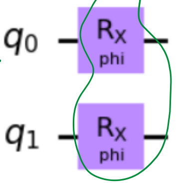
$$\frac{dT}{dt} = -k(T - T_a)$$

$$\frac{d\,\langle\;|\;|\;\rangle}{d}$$

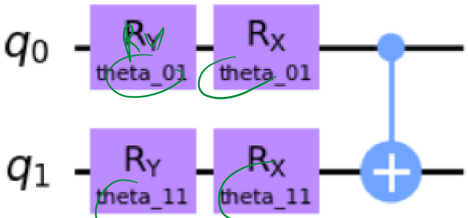


Ansatz:

Quantum Feature Map:



encoding



¡Gracias!

$$\frac{dp(t)}{d\tau} = Mp(t) + b$$

$$p' = e^{-cM\tau} p$$

$$p = e^{cM\tau} \cdot p'$$

$$\dot{p} = cMe^{cM\tau} p' + e^{cM\tau}\dot{p}' = M e^{cM\tau} \cdot p + b$$

$c=1$

$$\rightarrow e^{cM\tau}\dot{p}' = b$$

$$\dot{p}' = be^{-cM\tau}$$

$$= -cMb\frac{1}{c}M^{-1}e^{-cM\tau}$$

$$p' = be^{-cM\tau}$$

$$= \frac{b}{-cM} \cdot (e^{-cM\tau})$$

$$p' = -\frac{b}{c}M^{-1}e^{-cM\tau} + c\text{tte}$$

$$\left(-\frac{b}{c}M^{-1}e^{-cM\tau} + c\text{tte} = e^{-cM\tau}p\right)e^{cM\tau}$$