Dr.
Carolina
Perdomo

Del Qubit al Cálculo: La computación cuántica y la resolución de ecuaciones diferenciables con circuitos cuánticos.

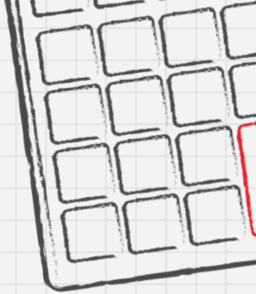
Seminario de Cómputo Cuántico - UNAM

Presentación breve

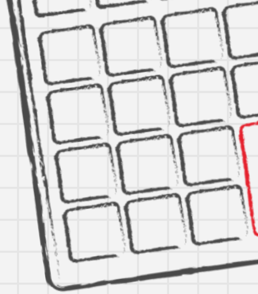
Algunas cositas sobre mí antes de empezar:

1	**Dafne Carolina. 31 años. Venezolana**. Actualmente en Francia.
2	**Doctora UFABC, Brasil-2022** (Intercambio con la Universidad de Coimbra, Portugal-2021); Master Université Paris Cité, Francia-2016; Licenciatura UCV, Venezuela-2014.
3	**Quantum Fellow en QuantumQuipu** (Perú/Latinoamérica). Cientifico de datos en GreenCarLane (Dinamarca).
4	Fundadora de **<Quantum|Chamitas>**, proyecto educativo en computación cuántica como parte del Global Quantum Project de Womanium.

Contenidos



Contenidos



01

Una nueva tecnología

Entendiendo la Computación Cuántica.

02

Ecuaciones diferenciables

¿Por qué son importantes? Puntos principales.

03

Algoritmos

Ecuaciones lineales. Ecuaciones diferenciables lineales. Flujo de trabajo e implementación.

04

Circuitos cuánticos variacionales (VQA)

Ecuaciones no-lineales. Otros algoritmos (Differential Quantum Circuits).

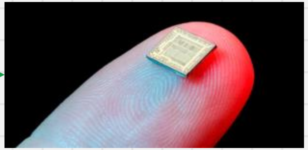
01

Una nueva tecnología

¿Qué es lo que hace especial a la computación cuántica?

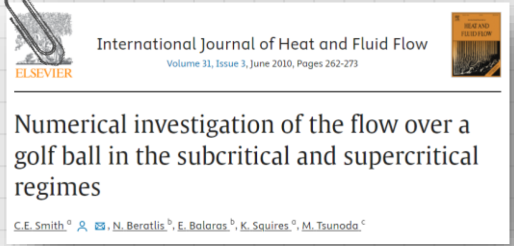
La computación “clásica”

Modelo tradicional de procesamiento de la información que se ha utilizado durante décadas en la industria de la tecnología.



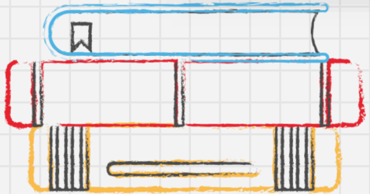
Algunas limitaciones

Procesamiento de grandes cantitades de datos

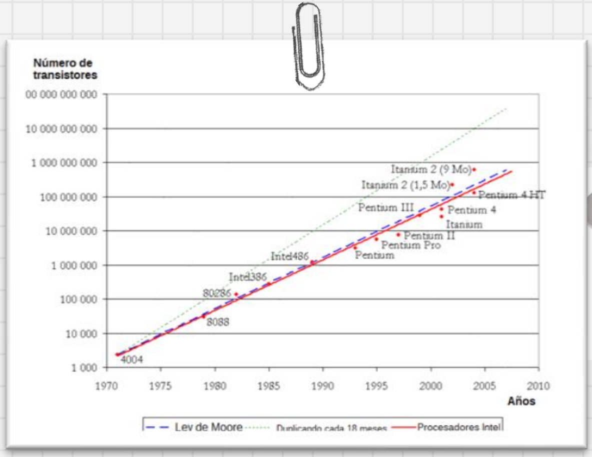

International Journal of Heat and Fluid Flow
Volume 31, Issue 3, June 2010, Pages 262-273

Numerical investigation of the flow over a golf ball in the subcritical and supercritical regimes

C.E. Smith [a], N. Beratlis [b], E. Balaras [b], K. Squires [a], M. Tsunoda [c]



Límite de la miniaturización

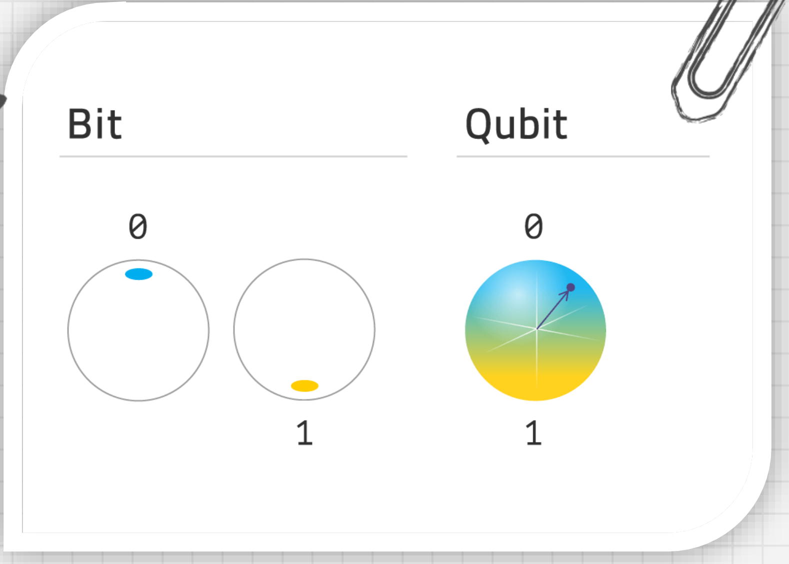


Bit vs Qubits

¿Qué sucede en el mundo cuántico?

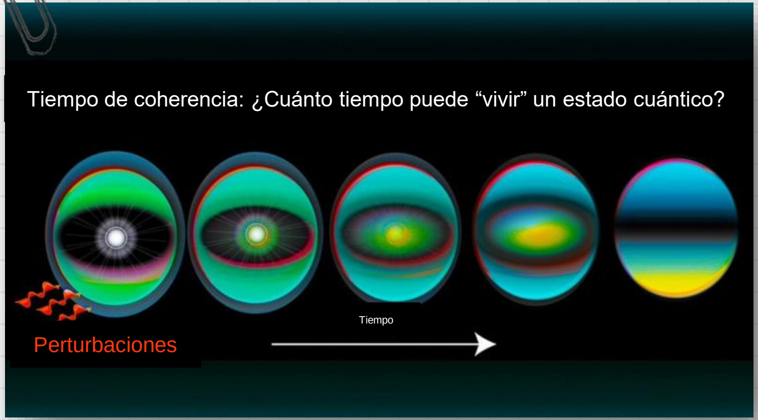
- Superposición.
- Entrelazamiento.

Existe una necesidad imperante de superar las limitaciones de la computación clásica. Los qubits son una de las soluciones viables.

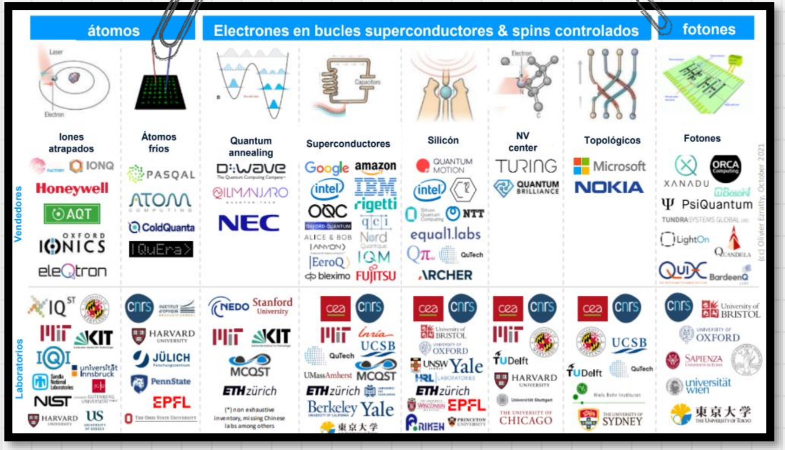


Nueva era

- Era de los primeros ordenadores cuánticos disponibles.
- **Imperfectos**: Noisy Intermediate-Scale Quantum (NISQ) computer.
- **Computación híbrida cuántica/clásica.**

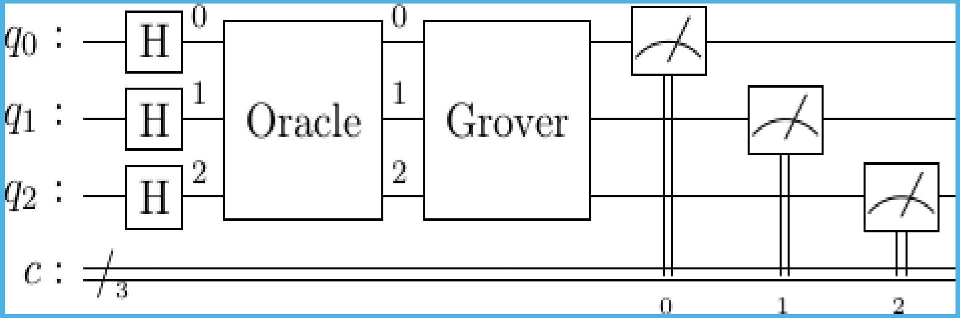


©Edwards/JQI (NIST)



Circuitos cuánticos

Un cálculo es una secuencia de puertas cuánticas, mediciones, inicializaciones de qubits, y otros.



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.

02

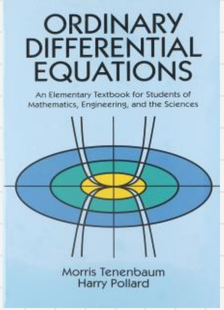
Ecuaciones diferenciables

Desafío de las ED.

¿Por qué queremos estudiar ED?

Básicos

Muchos modelos industriales pueden modelarse con ED. Las ED describen la física y las reglas del sistema.



¿Cuáles?

Tasas de cambio, leyes de conservación, etc.

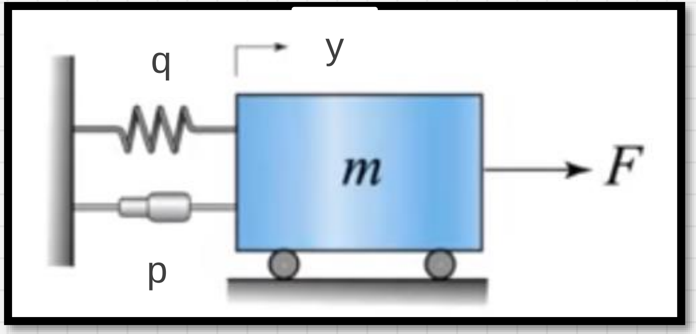
Consideremos la ED lineal de segundo orden:

$$y'' + p(x)y' + q(x)y = f(x)$$

Donde p(x), q(x), y f(x) son funciones continuas conocidas en un intervalo. Se busca encontrar y(x) que satisface esta ecuación en el intervalo.

Consideremos la ED lineal de segundo orden:

$$m\cdot y'' + \underbrace{p(x)}_{\text{amortiguación}}\, y' + \underbrace{q(x)}_{\text{rigidez}}\, \underbrace{y}_{\text{Desplazamiento del sistema.}} = F$$



Sistema masa–resorte–amortiguador.

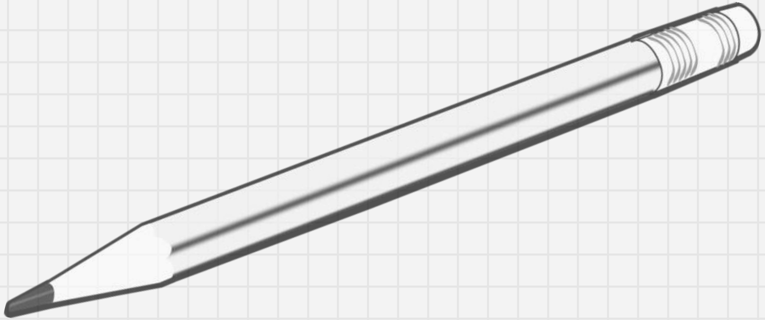
- Física.

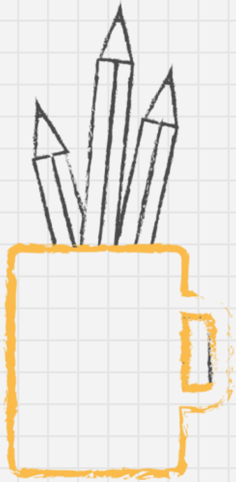
Consideremos la ED lineal de segundo orden:

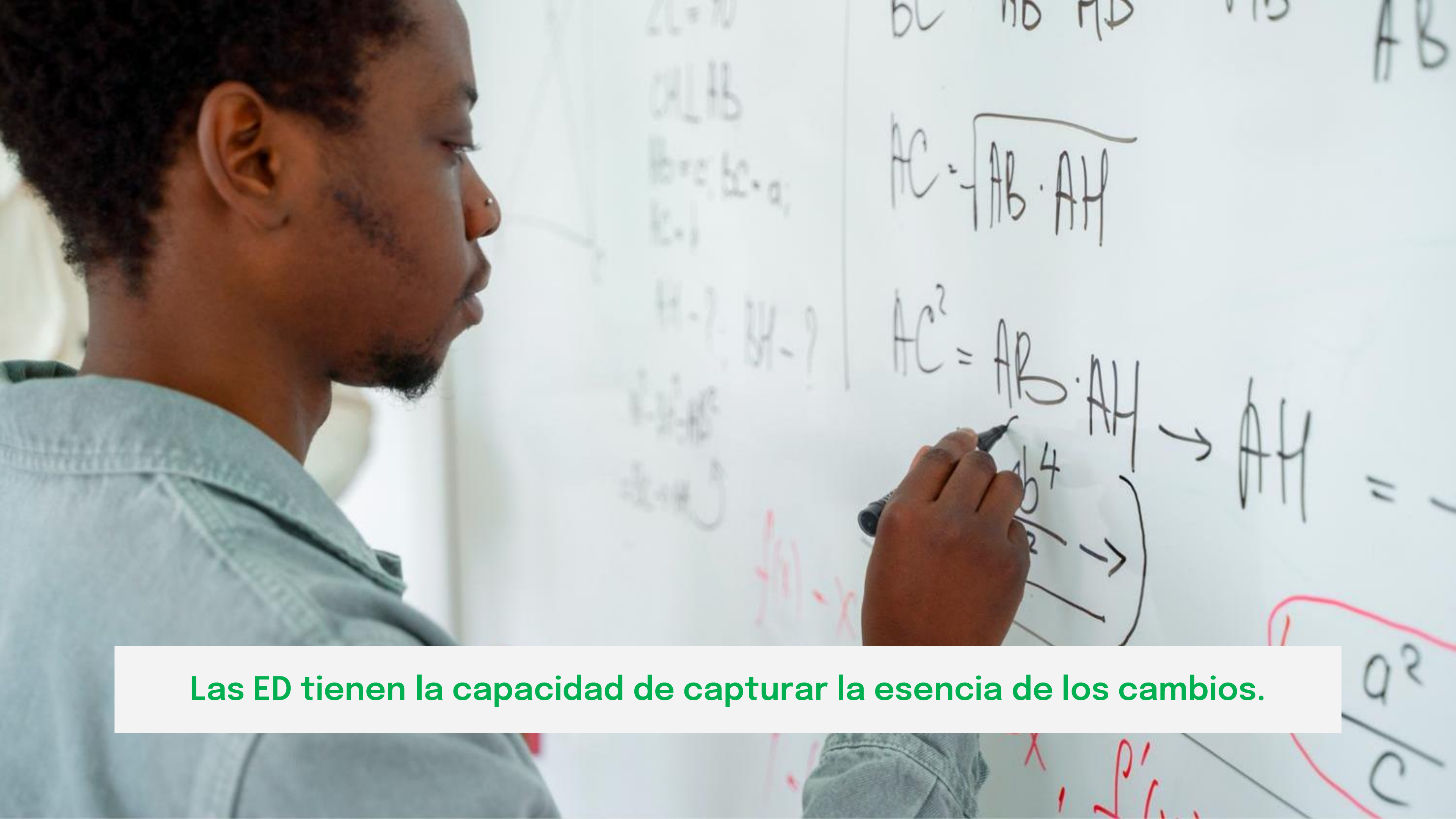
$$y'' + \underbrace{p}_{\substack{\text{constante} \\ 1>p>0}} y' + \overset{\substack{\text{constante} \\ q>0}}{q}\underbrace{y}_{\substack{\text{Renta} \\ \text{nacional}}} = \underbrace{c}_{\substack{\text{constante} \\ c>0}}\overset{\substack{\text{Oferta de} \\ \text{dinero}}}{M}$$

Modelo Hicks-Hansen como ecuación de 2do orden.

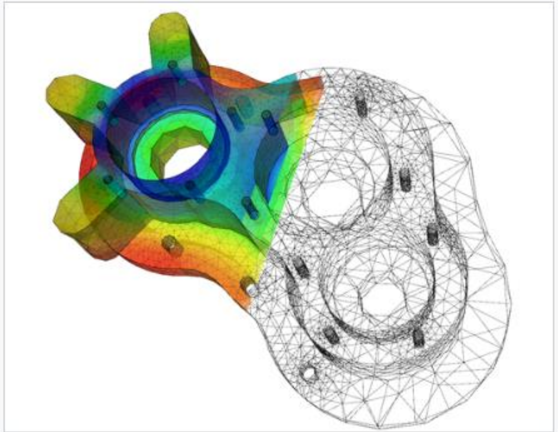
- Economía.



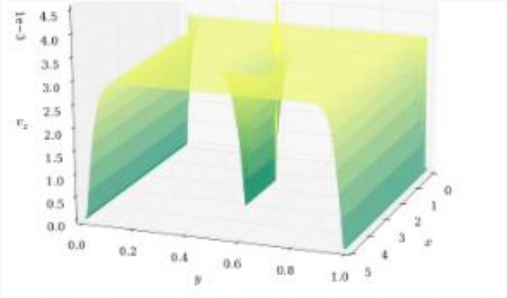




Las ED tienen la capacidad de capturar la esencia de los cambios.



Visualización de la transferencia de calor en la carcasa de una bomba, creada mediante la **resolución de la ecuación del calor**. El calor se genera internamente.



Ecuaciones diferenciales de Navier-Stokes utilizadas para simular el flujo de aire alrededor de una obstrucción.

Simulaciones físicas: procesos gobernados por ED

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

Computación cuántica (QC) y ED

1

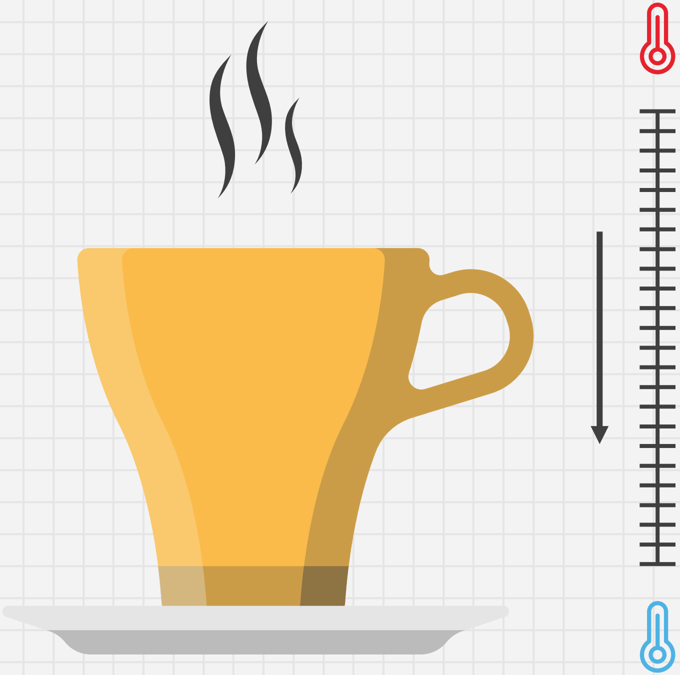
Encontrar soluciones a sistemas no lineales de ED es todo un reto.

2

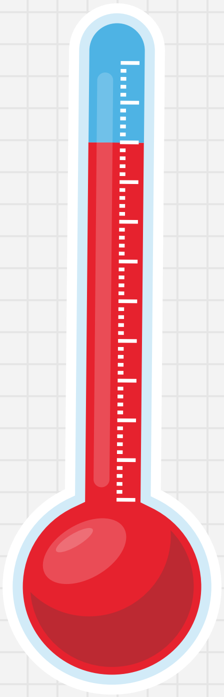
Los ordenes cuánticos ofrecen un enfoque fundamentalmente diferente para realizar.

3

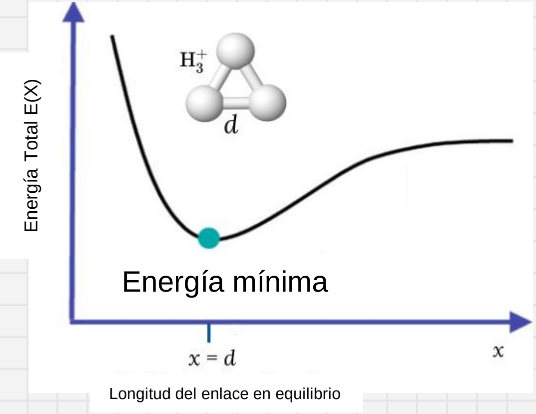
Encontrar protocolos NISQ que puedan ofrecer ventajas para tareas de relevancia industrial representa un reto abierto.



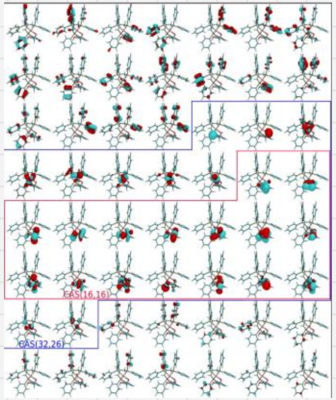
Usos potenciales de QC para ED



Ingienería Química

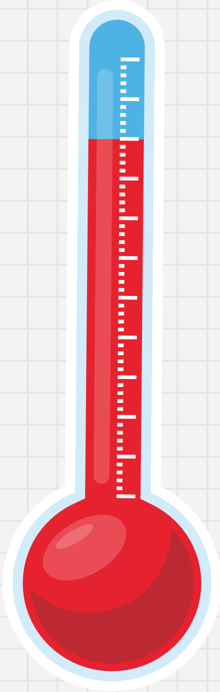


Clima

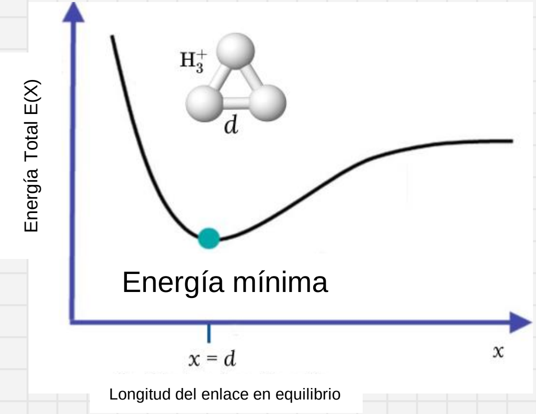


Quantum computing enhanced computational catalysis arXiv:2007.14460

Usos potenciales de QC para ED

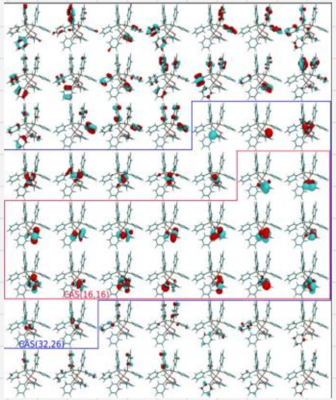


Ingieniería Química



Qubit pharmaceuticals

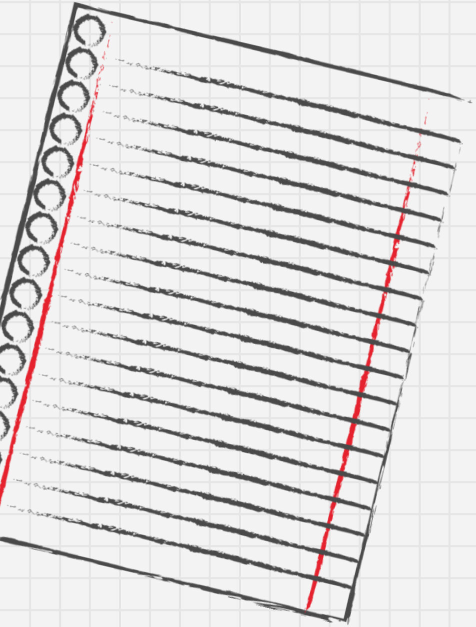
Clima



TotalEnergies

Resumen parte A:

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



03

Algoritmos

Idea general

$$A\vec{x} = \vec{b}$$

Idea general

$$\underbrace{A}_{\text{la conoces}}\vec{x} = \underbrace{\vec{b}}_{\text{la conoces}}$$

no la conoces

N variables

Idea general

$$\underbrace{A}_{\text{la conoces}}\,\underbrace{\vec{x}}_{\text{no la conoces}} = \underbrace{\vec{b}}_{\text{la conoces}}$$

N variables

Encontrar: $\vec{x} = A^{-1}\vec{b}$

Eliminación Gaussiana

El mejor algoritmo clásico.

Versión mecánico cuántica

$$\underbrace{A}_{\text{la conoces}}\underbrace{\vec{x}}_{\text{no la conoces}} = \underbrace{\vec{b}}_{\text{la conoces}}$$

N variables

Encontrar: $\vec{x} = A^{-1}\vec{b}$

Eliminación Gaussiana

El mejor algoritmo clásico.

Versión mecánico cuántica

$$\underbrace{A}_{\text{la conoces}}\vec{x} = \underbrace{\vec{b}}_{\text{la conoces}}$$

no la conoces

N variables

Encontrar: $\vec{x} = A^{-1}\vec{b}$

Eliminación Gaussiana

El mejor algoritmo clásico.

$$A|x >= |b >$$

Versión mecánico cuántica

$$\underbrace{A}_{\text{la conoces}}\vec{x} = \underbrace{\vec{b}}_{\text{la conoces}}$$

no la conoces

N variables

Encontrar: $\vec{x} = A^{-1}\vec{b}$

El mejor algoritmo clásico.

→ Eliminación Gaussiana

$$A|x >= |b >$$

CONSTRUIR:

$$|x >= A^{-1}|b >$$

Versión mecánico cuántica

$$\underbrace{A}_{\text{la conoces}}\vec{x} = \underbrace{\vec{b}}_{\text{la conoces}}$$

$\vec{x}$: no la conoces

N variables

Encontrar: $\vec{x} = A^{-1}\vec{b}$

El mejor → Eliminación Gaussiana algoritmo clásico.

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

CONSTRUIR:

$$|x\rangle = A^{-1}|b\rangle$$

Si $\underbrace{A = A^{\dagger}}_{\text{hermítica}}$ podemos implementar:

$$e^{-iA\tau}|b\rangle$$

$\rightarrow \langle x|M|x\rangle$

Versión mecánico cuántica

$$\underbrace{A}_{\text{la conoces}}\vec{x} = \underbrace{\vec{b}}_{\text{la conoces}}$$

no la conoces

N variables

Encontrar: $\vec{x} = A^{-1}\vec{b}$

Eliminación Gaussiana

El mejor algoritmo clásico.

$$A|x> = |b>$$

"encode"

CONSTRUIR:

$$|x> = A^{-1}|b>$$

Objetivo

Si $\underbrace{A = A^{\dagger}}_{\text{hermítica}}$ podemos implementar:

$$e^{-iA\tau}|b> \longrightarrow \langle x|M|x\rangle$$

Procesamiento

Evaluar respuesta

“Resolver” una ecuación lineal

Resumen.

1 Clásica:

Dada A, matriz $N \times N$, y $b \in \mathbb{R}^N$, podemos encon-

trar X, tal que: $A\vec{x} = \vec{b}$

2 “Cuántica”:

(una forma)

Dada la capacidad de producir el estado $|b\rangle = \sum_i b_i |i\rangle$, y tener la matriz A, podemos construir $|x\rangle = \sum_i x_i |i\rangle$

“Resolver” una ecuación lineal

1 Clásica:

Dada A, matriz $N \times N$, y $b \in \mathbb{R}^N$, podemos encontrar X, tal que: $A\vec{x} = \vec{b}$

2 “Cuántica”:

(una forma)

Dada la capacidad de producir el estado $|b\rangle = \sum_i b_i |i\rangle$, y tener la matriz A, podemos construir $|x\rangle = \sum_i x_i |i\rangle$

Interesante:

Teorema: Para un número de condición de A, |x> puede ser aproximadamente construido en tiempo polinomial(log N) [Harrow et al 0811.3171] [y otros].

Algoritmo para ED lineales

12 Jul 2018

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)

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.

Publicado

arXiv:1807.04553v1 [quant-ph]

PACS numbers:

¿Qué hicieron? Solucionar ED lineales a pequeña escala.
¿Cómo? 4-qubit en un sistema de resonancia magnética.

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

lo conoces

No lo conoces

Lo conoces

¿Cuál es la intuición “clásica” detrás de estos problemas?

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

factor de integración

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

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

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

Solución por expansión de Taylor:

$$\overrightarrow{\boldsymbol{p}(\boldsymbol{\tau})} \approx \sum_{m=0}^{k} \frac{(M\tau)^m}{m!}\vec{\boldsymbol{p}}(\mathbf{0}) + \sum_{n=1}^{k} \frac{M^{n-1}\tau^n}{n!}\vec{\boldsymbol{b}}$$

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

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

Solución por expansión de Taylor:

$$\vec{\boldsymbol{p}}(\boldsymbol{\tau}) \approx \sum_{m=0}^{k} \frac{(M\tau)^m}{m!}\vec{\boldsymbol{p}}(\mathbf{0}) + \sum_{n=1}^{k} \frac{M^{n-1}\tau^n}{n!}\vec{\boldsymbol{b}}$$

Operador

Describir como un estado cuántico.

"Encode"

$|p(0)\rangle$

$|b\rangle$

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

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

Operador

$|p(0)\rangle$

Solución por expansión de Taylor:

"Encode"

$|b\rangle$

$$|\vec{p}(\tau)\rangle \approx \sum_{m=0}^{k} \frac{(M\tau)^m}{m!} |\vec{p}(0)\rangle + \sum_{n=1}^{k} \frac{M^{n-1}\tau^n}{n!} |\vec{b}\rangle$$

Objetivo:

$$|p(t)\rangle \approx \sum_{m=0}^{k} \frac{\|x(0)\|(\|M\| \Delta t)^m}{m!} |p(0)\rangle$$

$$+ \sum_{n=1}^{k} \frac{\|b\|(\|M\| \Delta)^{n-1} t^n}{n!} |b\rangle$$

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

Operador

$|p(0)\rangle$

“Encode”

$|b\rangle$

Solución por expansión de Taylor:

$$|\vec{p}(\tau)\rangle \approx \sum_{m=0}^{k} \frac{(M\tau)^m}{m!} |\vec{p}(0)\rangle + \sum_{n=1}^{k} \frac{M^{n-1}\tau^n}{n!} |\vec{b}\rangle$$

Objetivo:

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

A unitario

Redefiniciones: $U_m = A^m$; $U_n = A^n$

$C_m = \|x(0)\| (\|M\|\tau)^m / m!$

$D_n = \|b\| (\|M\|\tau)^{n-1} \tau / n!$

$$|p(\tau)\rangle \approx \frac{1}{\mathbb{N}^2}\left(\sum_{m=0}^{k} C_m U_m |p(0)\rangle + \sum_{n=1}^{k} D_n U_{n-1} |b\rangle\right)$$

Ejemplo de implementación

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

$$\frac{\mathrm{d}\boldsymbol{p}}{d\tau} = M\boldsymbol{p}(\boldsymbol{0}) + \boldsymbol{b}$$

$M = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$ $b = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$

x y

Ejemplo tomado de:
https://github.com/andrijapau/QIC710-Final-Presentation/tree/main
(Andrija Paurevic, 2022)

Ejemplo de implementación

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

$$\frac{dp}{d\tau} = M\boldsymbol{p}(\mathbf{0}) + \boldsymbol{b}$$

$M = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$ $b = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$

2 Tech Stack

```python
import numpy as np
import scipy as sp

# Espacio t:
t_d = np.linspace(0.01, 3.01, 25)

# Definimos parametros
M = 1/np.sqrt(2) * np.array([[1, 1], [1, -1]])
p0 = (1 / np.sqrt(2)) * np.array([1, 1j])
b = (1 / np.sqrt(2)) * np.array([1, -1])

```

Ejemplo tomado de:
https://github.com/andrijapau/QIC710-Final-Presentation/tree/main
(Andrija Paurevic, 2022)

Ejemplo de implementación

1 Caso:

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

$$\frac{d\boldsymbol{p}}{d\tau} = M\boldsymbol{p}(\boldsymbol{0}) + \boldsymbol{b}$$

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

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

2 Tech Stack

```python
import numpy as np
import scipy as sp

# Espacio t:
t_d = np.linspace(0.01, 3.01, 25)

# Definimos parametros
M = 1/np.sqrt(2) * np.array([[1, 1], [1, -1]])
p0 = (1 / np.sqrt(2)) * np.array([1, 1j])
b = (1 / np.sqrt(2)) * np.array([1, -1])
```

3 Flujo de trabajo.

Creamos un circuito cuántico compuesto por diferentes compuertas.

Objetivo:

$$\langle p(\tau)\rangle \approx \frac{1}{\mathbb{N}}\left(\sum_{m=0}^{k} C_m U_m |p(0)\rangle + \sum_{n=1}^{k} D_n U_{n-1} |b\rangle\right)$$

Ejemplo tomado de: https://github.com/andrijapau/QIC710-Final-Presentation/tree/main (Andrija Paurevic, 2022)

¿Cómo construir la solución?

(de forma más óptima)

1 → 2 → 3 → 4

“Encoding”

Necesitamos qubits y **qubits auxiliares** que nos ayuden a “codificar” |p(o)> y |b>. Para ello, realizamos **operaciones unitarias.**

Operador “evolución”

Operaciones controladas para construir la solución.

“Decoding”

Las operaciones en los qubits auxiliares se revierten. Sólo el subespacio de la solución es el que queda.

Medida

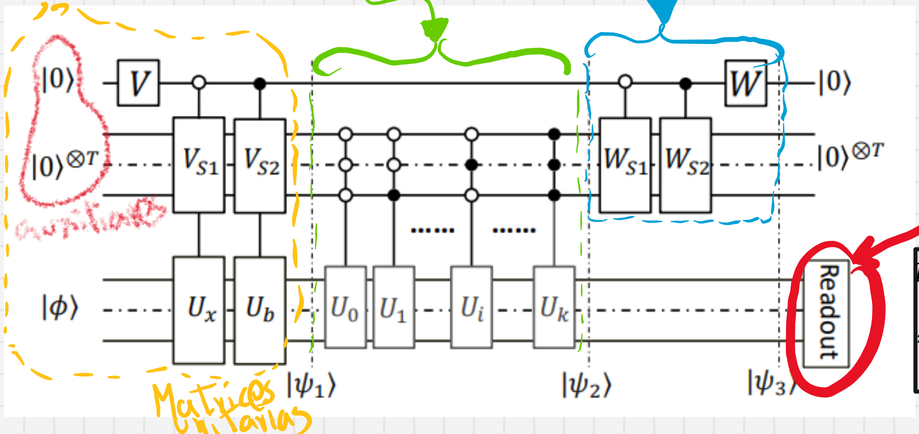
Medimos el estado final de los qubits en el subespacio donde los qubits auxiliares son |o>.

¿Cómo construir la solución?

(de forma más óptima)

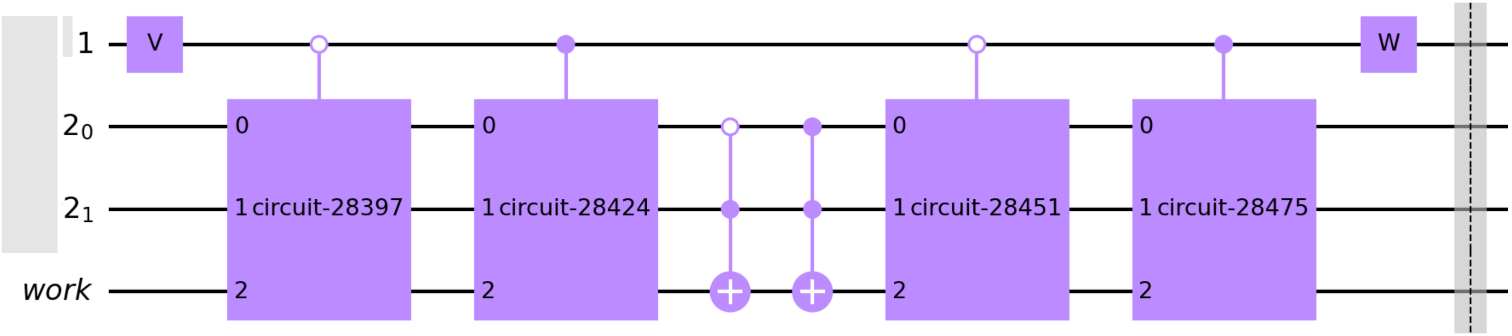
1 → 2 → 3 → 4

“Encoding” | Operador “evolución” | “Decoding” | Medida



$$|p(\tau)\rangle \approx \frac{1}{\mathbb{N}} \cdot \text{factor} \left(\sum_{m=0}^{k} C_m U_m |p(0)\rangle + \sum_{n=1}^{k} D_n U_{n-1} |b\rangle \right)$$

$$U_m = A^m, \; U_n = A^n$$



evolución

"decoding"

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

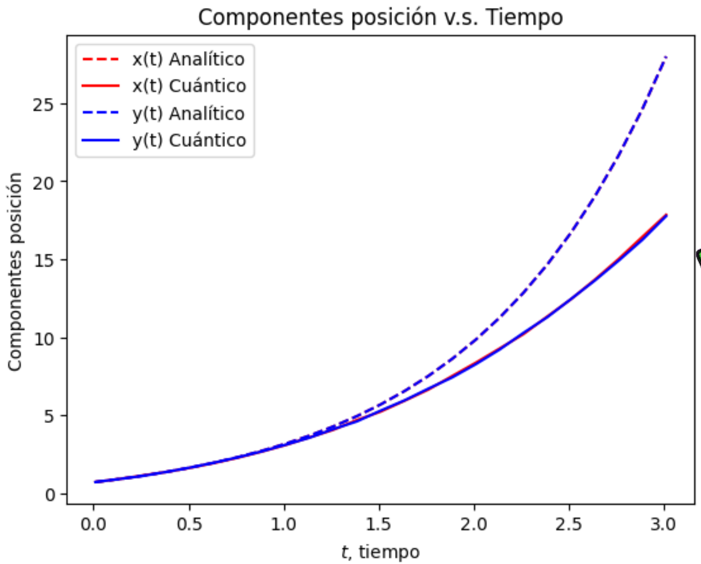
$$\frac{d\boldsymbol{p}}{d\tau} = M\boldsymbol{p}(\boldsymbol{0}) + \boldsymbol{b}$$

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

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

ejemplo:

```python
def encode(self):

    # Circuito para transformar a la computerta VS1.
    S1_circuit = QuantumCircuit(3)

    # Ux (para evolucionar el work qubit a |p(0)>):
    S1_circuit.h(2)

    # Vs1 (equation 12 artículo):
    S1_circuit.unitary(self.v_S1_U, [0, 1], label='VS1')
    S1_gate = S1_circuit.to_gate().control(1, ctrl_state='0')

    # Circuito para transformar a la computerta VS2.
    S2_circuit = QuantumCircuit(3)

    # Ub (para evolucionar el work qubit a |b>):
    S2_circuit.h(2)

    # Vs2 (equation 13):
    S2_circuit.unitary(self.v_S2_U, [0, 1], label='VS2')
    S2_gate = S2_circuit.to_gate().control(1, ctrl_state='1')

    # Colocamos todo en el circuito:
    self.circuit.unitary(self.V_U, 0, label='V')
    self.circuit.append(S1_gate, [0, 1, 2, 3])
    self.circuit.append(S2_gate, [0, 1, 2, 3])
```



Ejemplo tomado de:
https://github.com/andrijapau/QIC710-Final-Presentation/tree/main
(Andrija Paurevic, 2022)

¿Qué es importante recordar?

1

Construimos una solución con la computadora cuántica. Es un acercamiento diferente a la computación clásica.

2

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

3

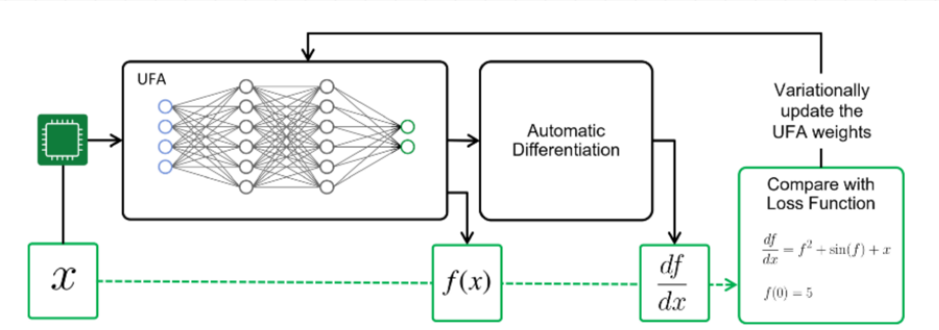
En este caso particular: El uso de las compuertas para mapear la expansión de Taylor.

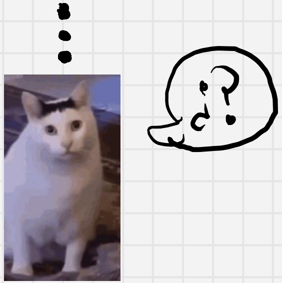
4 Ecuaciones NO lineales

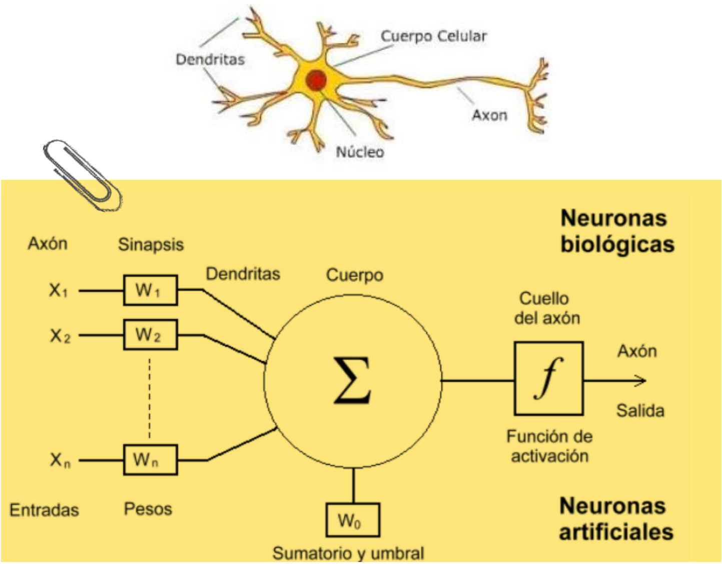
4 Ecuaciones NO lineales

Redes neuronales

Clásico.

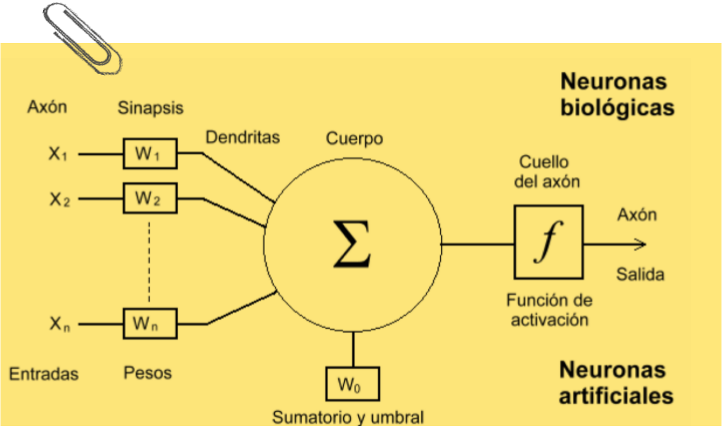


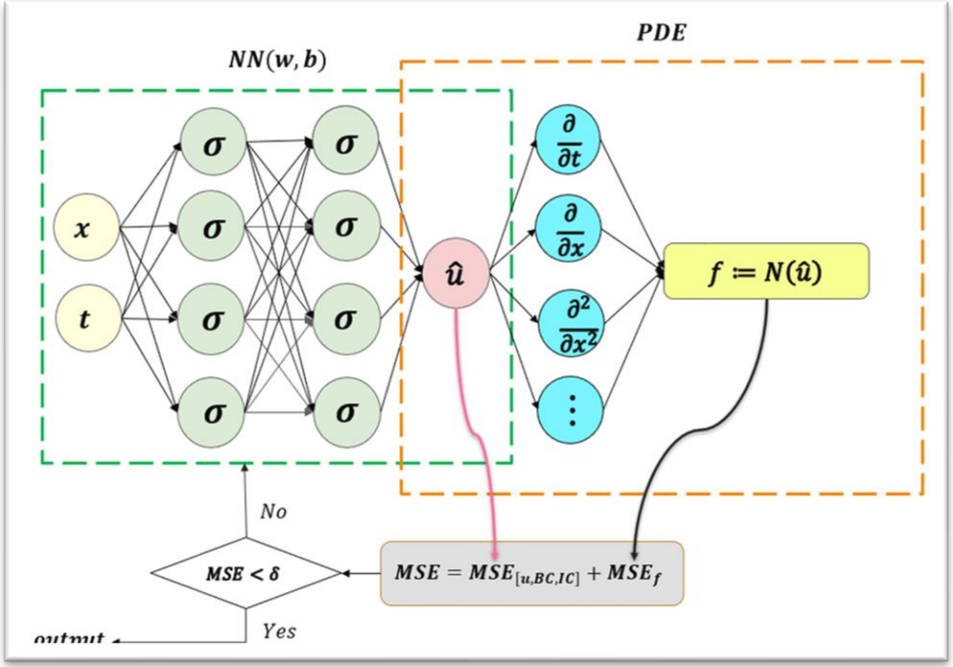




Una serie de algoritmos que intentan reconocer las relaciones subyacentes en un conjunto de datos mediante un proceso que "imita" el funcionamiento del cerebro humano.

López Pacheco, M. A. (2017) *Identificación de sistemas no lineales con redes neuronales convolucionales.* Instituto Politécnico Nacional. Tesis de maestría.

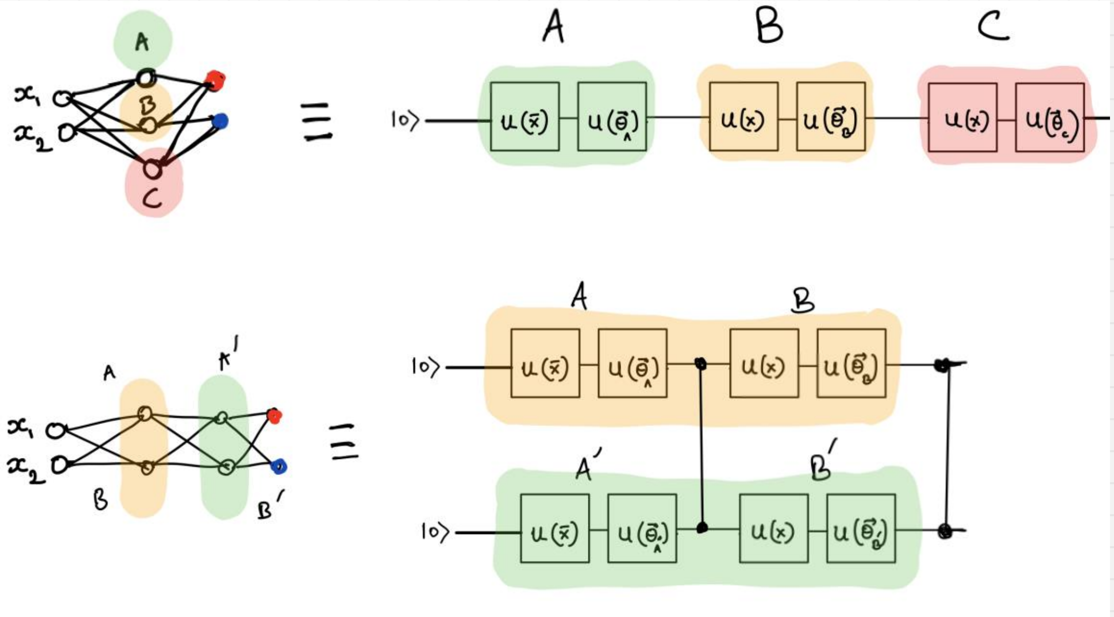




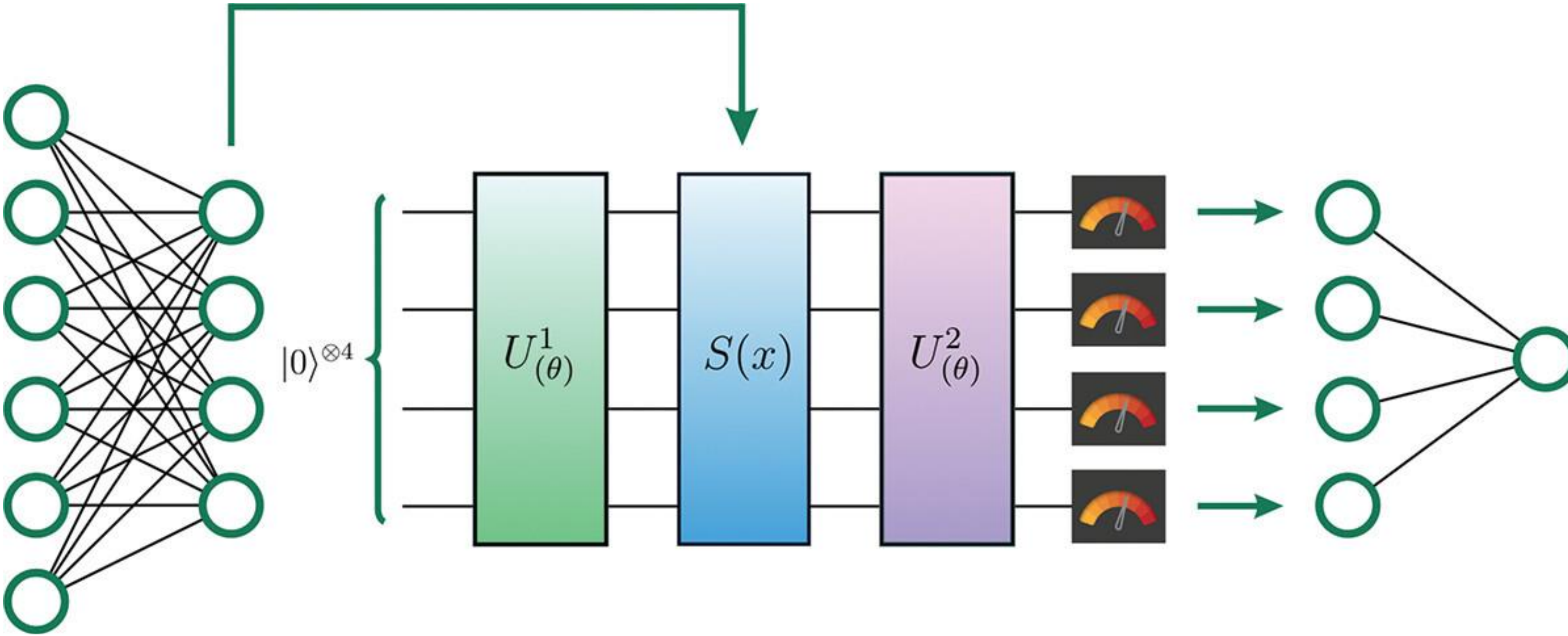
Para el caso de redes neuronales informadas con física (PINNs), se utiliza *diferenciación automática* (descomponer el cáculo en una secuencia de operaciones elementales) como parte del proceso de entrenamiento. Cada **nodo representa una operación** y las **aristas representan las dependencias entre las operaciones**.

Guo Y, Cao X, Liu B, Gao M. Solving Partial Differential Equations Using Deep Learning and Physical Constraints. *Applied Sciences*. 2020; 10(17):5917. https://doi.org/10.3390/app10175917

¿Qué podemos aprender de las redes neuronales?



Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E., & Latorre, J. I. (2020). Data re-uploading for a universal quantum classifier. *Quantum*, *4*, 226.

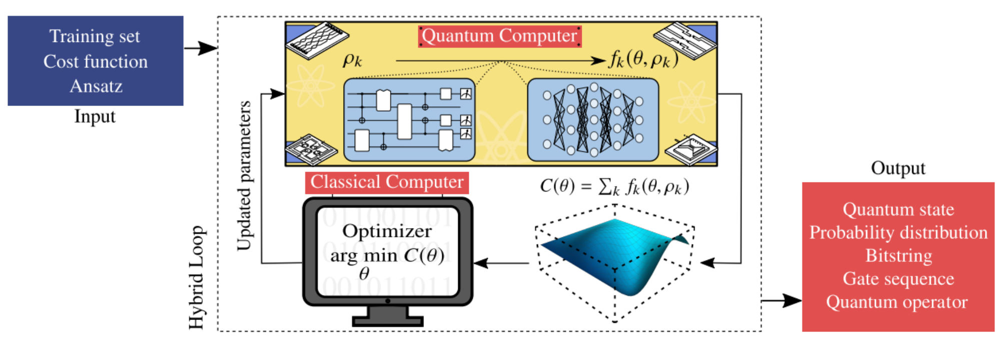


Alexey Melnikov, Mohammad Kordzanganeh, Alexander Alodjants & Ray-Kuang Lee (2023) Quantum machine learning: from physics to software engineering, Advances in Physics: X, 8:1, DOI: 10.1080/23746149.2023.2165452

4 Ecuaciones NO lineales

Variational Quantum Algorithm

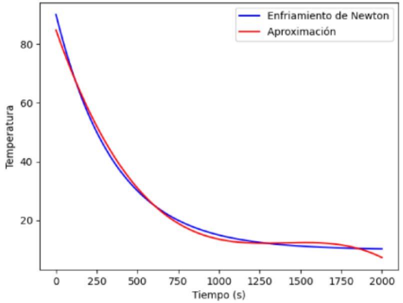
Cuántico.

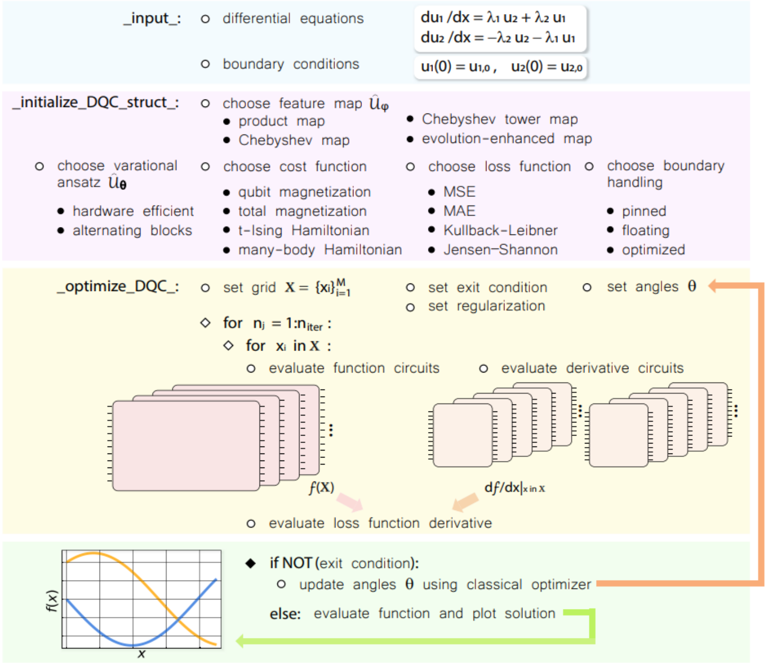


Circuitos cuánticos diferenciables (DQC)

¿Cómo construyes el circuito?

¡Puedes tener tantas versiones del circuito como quieras!





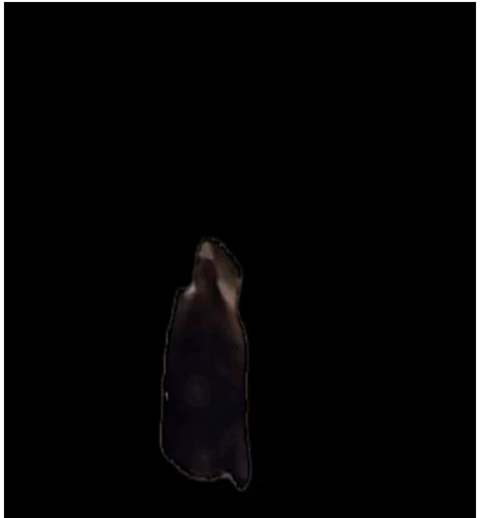
Referencia principal: Kyriienko, O., Paine, A. E., & Elfving, V. E. (2021). Solving nonlinear differential equations with differentiable quantum circuits. *Physical Review A*, *103*(5), 052416.

Conclusiones

1. Muchos modelos industriales pueden ser resueltos con ED. ¿Por qué? **Las ED describen la física o las reglas de un sistema en términos de sus ratas de cambio**.
2. La computación cuántica ofrece un acercamiento **diferente** para realizar cálculos. Para algunas situaciones, ofrece ventajas algorítmicas con respecto a la computación clásica.
3. Diferentes "tipos" de "solucionadores" para ED han sido propuestos en el dominio cuántico. Básicamente, se **construye** y mide el estado de energía del sistema.
4. **Para ecuaciones no lineales y soluciones numéricas, se utilizan algoritmos cuánticos basados en las redes neuronales modernas**. Estos algoritmos se ejecutan en sistemas "híbridos".

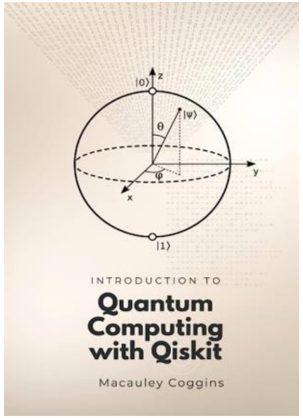
"La buena suerte se produce cuando la preparación se encuentra con la oportunidad."

… Y si, hay que aprender algebra lineal.



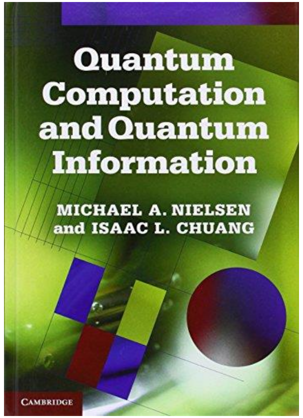
… Y si, hay que aprender algebra lineal.

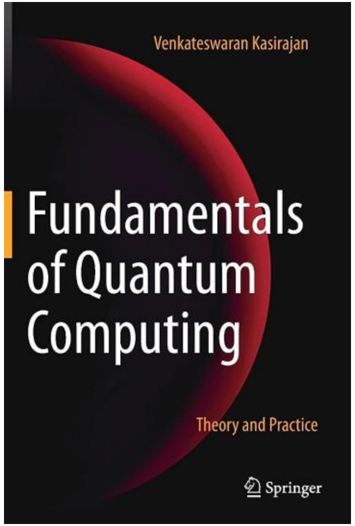
¿Qué puedo hacer para aprender más desde México?



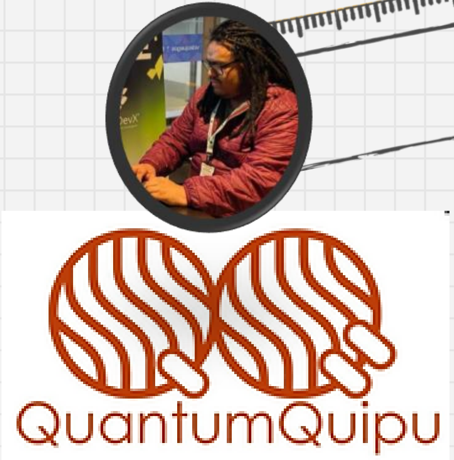
Qiskit

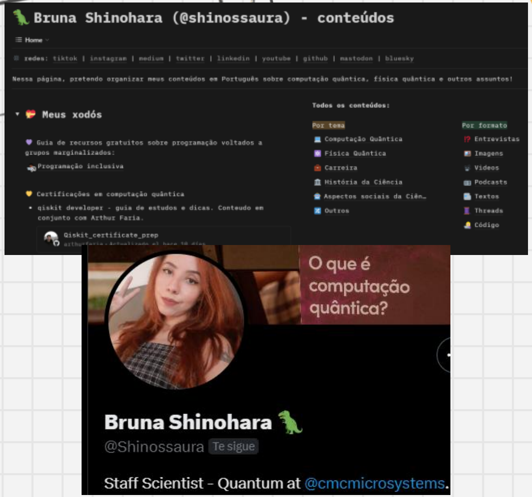
PENNYLANE





Para aprender más:





Womanium y QWorld

Excelentes programas.
Contactar: Dr. Claudia Zendejas-Morales.

QuantumQuipu

¡En español!
Contactar: Estudiante de doctorado Ricardo Angelo Quispe Mendizábal.

Divulgación

Seguir: Dr. Bruna Shinohara.

¡Gracias!

drcarolinaperdomo.com

< Quantum | Chamitas >

QuantumQuipu

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, infographics & images by Freepik

Please keep this slide for attribution

Comentarios/Preguntas/Etc…

$$\begin{cases} 2x & +y & -z & = & 8 \\ -3x & -y & +2z & = & -11 \\ -2x & +y & +2z & = & -3 \end{cases}$$

$$\begin{cases} 2x & +y & -z & = & 8 \\ & \frac{1}{2}y & +\frac{1}{2}z & = & 1 \\ & 2y & +z & = & 5 \end{cases}$$

$$\begin{cases} 2x & & -2z & = & 6 \\ & \frac{1}{2}y & +\frac{1}{2}z & = & 1 \\ & & -z & = & 1 \end{cases}$$

$$\begin{cases} 2x & & & = & 4 \\ & \frac{1}{2}y & & = & \frac{3}{2} \\ & & -z & = & 1 \end{cases}$$

$$\begin{cases} x & & & = & 2 \\ & y & & = & 3 \\ & & z & = & -1 \end{cases}$$

$$\begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

$$O(\kappa^2 \log N/\varepsilon)$$

$$\sum_{i=1}^{N} \beta_i \lambda_j^{-1} |u_j\rangle = A^{-1}|b\rangle = |x\rangle$$

$$H = \sum_i \alpha_i P_i \qquad P_i \in \{I, X, Y, Z\}^{\otimes n}$$

$$\alpha_i = \frac{1}{2^n} Tr(P_i H)$$

Hamiltonian simulation

From Wikipedia, the free encyclopedia

For broader coverage of this topic, see Quantum simulator.

Hamiltonian simulation (also referred to as **quantum simulation**) is a problem in quantum information science that attempts to find the computational complexity and quantum algorithms needed for simulating quantum systems. Hamiltonian simulation is a problem that demands algorithms which implement the evolution of a quantum state efficiently. The Hamiltonian simulation problem was proposed by Richard Feynman in 1982, where he proposed a quantum computer as a possible solution since the simulation of general Hamiltonians seem to grow exponentially with respect to the system size.[1]

Problem statement [edit]

In the Hamiltonian simulation problem, given a Hamiltonian H ($2^n \times 2^n$ hermitian matrix acting on n qubits), a time t and maximum simulation error ϵ, the goal is to find an algorithm that approximates U such that $||U - e^{-iHt}|| \leq \epsilon$, where e^{-iHt} is the ideal evolution and $||\cdot||$ is the spectral norm. A special case of the Hamiltonian simulation problem is the local Hamiltonian simulation problem. This is when H is a k-local Hamiltonian on n qubits where $H = \sum_{j=1}^{m} H_j$ and H_j acts non-trivially on at most k qubits instead of n qubits.[2] The local Hamiltonian simulation problem is important because most Hamiltonians that occur in nature are k-local.[2]

$$\frac{d^2Y}{dt^2} + s\frac{dY}{dt} + akY = aM,$$

$$\begin{pmatrix} \frac{dX_1}{dt} \\ \frac{dX_2}{dt} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -ak & -s \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} + \begin{pmatrix} 0 \\ aM \end{pmatrix}$$