UNIVERSITY
OF COLOGNE

ADVANCED SEMINAR ON QUANTUM COMPUTING [SOSE26]

HAMILTONIAN SIMULATION: QUBITIZATION & QUANTUM SIGNAL PROCESSING

Based on a paper by Guang Hao Low and Isaac L. Chuang (2019)

Presented by Vincent Borlisch

Motivation

- Quantum simulation implements physical systems directly in a quantum circuit instead of performing numerical calculations

  → Promises an advantage over classical simulation

- Quantum systems are governed by a Hermitian Hamiltonian H and obey the Schrödinger equation:

$$i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = H |\psi(t)\rangle$$

- General solution for a time-independent Hamiltonian is the time-evolution operator:

$$|\psi(t)\rangle = U(t) |\psi(0)\rangle = e^{-\frac{i}{\hbar} H t} |\psi(0)\rangle$$

Motivation

- If H is unitary, then

$$H^\dagger H = HH = H^2 = I$$

- And the time-evolution operator trivializes to:

$$U(t) = e^{-\frac{i}{\hbar}Ht} = \cos\left(\frac{t}{\hbar}\right) I - i \sin\left(\frac{t}{\hbar}\right) H$$

- But usually, the Hamiltonian is not unitary

→ Other methods necessary

Motivation

- Ideally, we would just diagonalize H using its eigenstates:

$$H = \sum_{\lambda} \lambda \, |\lambda\rangle\langle\lambda|$$

$$\Rightarrow U(t) = e^{-\mathrm{i}Ht} = \sum_{\lambda} e^{-\mathrm{i}\lambda t} \, |\lambda\rangle\langle\lambda|$$

- But for large Hamiltonians, calculating the eigenvalues and eigenstates is numerically intensive

  → Low and Chuang present an optimal quantum computation solution

Main theorem

- Given:
  - System Hamiltonian $\hat{H} \in \mathbb{C}^{N \times N}$ ➔ $log_2(N)$ qubits
  - H is related to a unitary $\hat{U} \in \mathbb{C}^{Nd \times Nd}$ via $(\langle G|_a \otimes \hat{\mathcal{I}}_s)\hat{U}(|G\rangle_a \otimes \hat{\mathcal{I}}_s) = \hat{H}$
  - $log_2(d)$ ancillary qubits for state preparation

$$\hat{G}|0\rangle_a = |G\rangle_a \in \mathbb{C}^d$$

We can simulate the time-evolution for a target error ε, simulation time t, failure probability using:

- The system, ancillary and max. 2 $e^{-i\hat{H}t}$ qubits $\mathcal{O}(\epsilon)$
- uses of controlled G, U and their inverses
- 2-qubit gates

$\Theta(Q)$ $Q = \mathcal{O}(t + \log{(1/\epsilon)})$

$\mathcal{O}(Q \log{(d)})$

Hamiltonian Simulation
– Overview –

-

1. Encode H in a unitary U: $\left(\langle G|_a \otimes I_s\right) U \left(|G\rangle_a \otimes I_s\right) = H$

2. Transform U into the iterate W, another H-encoding acting on each eigenspace separately

3. Produce polynomials of H using the iterate: $\langle G|W^k|G\rangle = T_k(H)$

4. Approximate $e^{-iHt} = \sum_{n=0}^{\infty} \frac{(-iHt)^n}{n!}$ using these polynomials of H

Standard form

- Since H is not unitary, we encode it as a block of a larger unitary U: $\hat{U} = \begin{pmatrix} \hat{H} & \cdot \\ \cdot & \cdot \end{pmatrix}$

- The **signal oracle** U lives in a bigger Hilbert space, combining ancilla space and system space:

$$\hat{U} : \mathcal{H}_a \otimes \mathcal{H}_s \;\rightarrow\; \mathcal{H}_a \otimes \mathcal{H}_s$$

- To retrieve H, we use an ancilla **signal state** $|G\rangle_a \in \mathcal{H}_a$ in the **standard-form encoding**:

$$\hat{H} = (\langle G|_a \otimes I_s)\; \hat{U}\; (|G\rangle_a \otimes I_s)$$

Standard form

- Turning H into a unitary U:

Example:

$$H = \frac{1}{2}(X+Z) = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$U = \begin{pmatrix} X & 0 \\ 0 & Z \end{pmatrix} = |0\rangle\langle 0| \otimes X + |1\rangle\langle 1| \otimes Z \qquad |G\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$\rightarrow \; (\langle G| \otimes I)\, U\, (|G\rangle \otimes I) = \frac{1}{2}(\langle 0| + \langle 1|) \otimes I \; (|0\rangle\langle 0| \otimes X + |1\rangle\langle 1| \otimes Z) \; (|0\rangle + |1\rangle) \otimes I$$

$$= \frac{1}{2}(X+Z) \; = H$$

Standard form

Example: Linear combination of 2-qubit Pauli strings: $H = \frac{1}{2}(X \otimes Y) + \frac{1}{3}(Y \otimes Z) + \frac{1}{6}(Z \otimes X)$

$$U = \begin{pmatrix} X \otimes Y & 0 & 0 \\ 0 & Y \otimes Z & 0 \\ 0 & 0 & Z \otimes X \end{pmatrix}$$

$$|G\rangle = \sqrt{\frac{1}{2}}\,|0\rangle + \sqrt{\frac{1}{3}}\,|1\rangle + \sqrt{\frac{1}{6}}\,|2\rangle$$

Standard form

-
  - Example: Generalize to Linear combinations of unitaries (LCUs)

$$\hat{H} = \sum_{j=1}^{d} \alpha_j \hat{U}_j, \quad \|\hat{H}\| \leq \|\vec{\alpha}\|_1 = \sum_{j=1}^{d} |\alpha_j|,$$

$$\hat{G} = \sum_{j=1}^{d} \sqrt{\frac{\alpha_j}{\|\vec{\alpha}\|_1}} |j\rangle\langle 0|_a + \cdots , \quad \hat{U} = \sum_{j=1}^{d} |j\rangle\langle j|_a \otimes \hat{U}_j = \begin{pmatrix} U_1 & 0 & \cdots & 0 \\ 0 & U_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & U_d \end{pmatrix}$$

→ Put U_j on j-th block on main diagonal and let $|G\rangle$ be a superposition weighted by α_j

Standard form

- In general, the action of U on a signal state is:

$$\hat{U}|G\rangle_a|\psi\rangle_s = \underbrace{|G\rangle_a\hat{H}|\psi\rangle_s}_{\mathcal{H}_G \,=\, |G\rangle \otimes \mathcal{H}_s} + \underbrace{\sqrt{1-\|\hat{H}|\psi\rangle\|^2}|G_\psi^\perp\rangle_{as}}_{\mathcal{H}_{G^\perp}}$$

with $(\langle G|_a \otimes \hat{\mathcal{I}}_s)|G_\psi^\perp\rangle_{as} = 0$

- For eigenvectors $\hat{H}|\lambda\rangle = \lambda|\lambda\rangle$:

$$\hat{U}|G\rangle|\lambda\rangle = \hat{U}|G_\lambda\rangle = \lambda|G_\lambda\rangle + \sqrt{1-|\lambda|^2}|G_\lambda^\perp\rangle$$

- This defines a subspace for each eigenvalue: $\mathcal{H}_\lambda = \mathrm{span}\{|G_\lambda\rangle, \hat{U}|G_\lambda\rangle\}$

Quantum Signal Processor

- We require powers of H to get $$e^{-iHt} = \sum_{n=0}^{\infty} \frac{(-iHt)^n}{n!} = I - iHt - \frac{1}{2!}H^2t^2 + \frac{i}{3!}H^3t^3 - \cdots$$

→ Why not just use $U, U^2, U^3, \ldots$?

Quantum Signal Processor

- Example: $H = \frac{1}{2}(X+Z) = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

$$U = \begin{pmatrix} X & 0 \\ 0 & Z \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$|G\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$\rightarrow$ $U^2 = \begin{pmatrix} X^2 & 0 \\ 0 & Z^2 \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}$

$\rightarrow$ $(\langle G| \otimes I)\, U^2\, (|G\rangle \otimes I) = I$

$\rightarrow$ $H^2 = \left(\frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\right)^2 = \frac{1}{4}\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = \frac{1}{2} I$

$\rightarrow$ $H^2 \neq \langle G|U^2|G\rangle$

Quantum Signal Processor

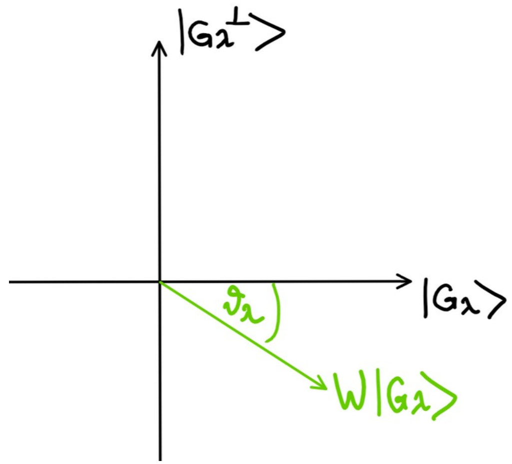
→ Solution: The **iterate** W

- W is another H-encoding, i.e. $\hat{H} = \langle G|\hat{W}|G\rangle$, but is constructed in a way that also satisifes something like:

$$\langle G|W^k|G\rangle = H^k$$

- Idea: For each eigenvalue, W performs a controlled rotation in the subspace $\mathcal{H}_\lambda = \mathrm{span}\{|G_\lambda\rangle, |G_\lambda^\perp\rangle\}$

- Rotation by exactly: $\theta_\lambda = \cos^{-1}(\lambda)$



Quantum Signal Processor

$$\hat{W}_\lambda = \begin{pmatrix} \lambda & -\sqrt{1-|\lambda|^2} \\ \sqrt{1-|\lambda|^2} & \lambda \end{pmatrix}_\lambda = \begin{array}{rr} \lambda|G_\lambda\rangle\langle G_\lambda| & -\sqrt{1-|\lambda|^2}|G_\lambda\rangle\langle G_\lambda^\perp| \\ +\sqrt{1-|\lambda|^2}|G_\lambda^\perp\rangle\langle G_\lambda| & +\lambda|G_\lambda^\perp\rangle\langle G_\lambda^\perp| \end{array}$$

$$\rightarrow \quad \hat{W} = \bigoplus_\lambda \begin{pmatrix} \lambda & -\sqrt{1-|\lambda|^2}. \\ \sqrt{1-|\lambda|^2}. & \lambda \end{pmatrix}_\lambda = \bigoplus_\lambda e^{-i\hat{Y}_\lambda \theta_\lambda}$$

- Predictable, independent evolution of each eigenspace
- The evolution depends only on the corresponding eigenvalue

Quantum Signal Processor

- Iterating W yields:

$$W = \begin{pmatrix} \lambda & -\sqrt{1-\lambda^2} \\ \sqrt{1-\lambda^2} & \lambda \end{pmatrix}$$

$$\langle G|W|G\rangle = H$$

$$W^2 = \begin{pmatrix} 2\lambda^2-1 & -2\lambda\sqrt{1-\lambda^2} \\ 2\lambda\sqrt{1-\lambda^2} & 2\lambda^2-1 \end{pmatrix}$$

$$\langle G|W^2|G\rangle = 2H^2 - I$$

$$W^3 = \begin{pmatrix} 4\lambda^3-3\lambda & -(4\lambda^2-1)\sqrt{1-\lambda^2} \\ (4\lambda^2-1)\sqrt{1-\lambda^2} & 4\lambda^3-3\lambda \end{pmatrix}$$

$$\langle G|W^3|G\rangle = 4H^3 - 3H$$

→ We can produce polynomials of H using W!

→ $\langle G|W^k|G\rangle = T_k(H)$

Qubitization

- But we don‘t have access to the eigenvalues/eigenspaces

  → We don‘t need to!

Theorem 2 (Qubitization). *Given a Hermitian matrix $\hat{H} = \langle G|_a \hat{U} |G\rangle_a$ encoded in standard-form as described in Definition 1, the iterate $\hat{W}$ of Eq. (6) can be constructed using at most one query each to $\hat{G}$, controlled-U, their inverses, at most one additional qubit, and $\mathcal{O}(\log(\dim(\mathcal{H}_a)))$ quantum gates.*

Qubitization

- Start with the ansatz $W = RU$, where $R = (2|G\rangle\langle G| - I_a) \otimes I_s$ is a reflection about $|G\rangle$
- For now, assume that $U^2 = I$, i.e. that U is also a reflection
- Remember the action of U on an eigenstate: $U|G_\lambda\rangle = \lambda|G_\lambda\rangle + \sqrt{1-\lambda^2}\,|G_\lambda^\perp\rangle$
- And using $U^2|G_\lambda\rangle = |G_\lambda\rangle$, we find the action of U on the orthogonal state:

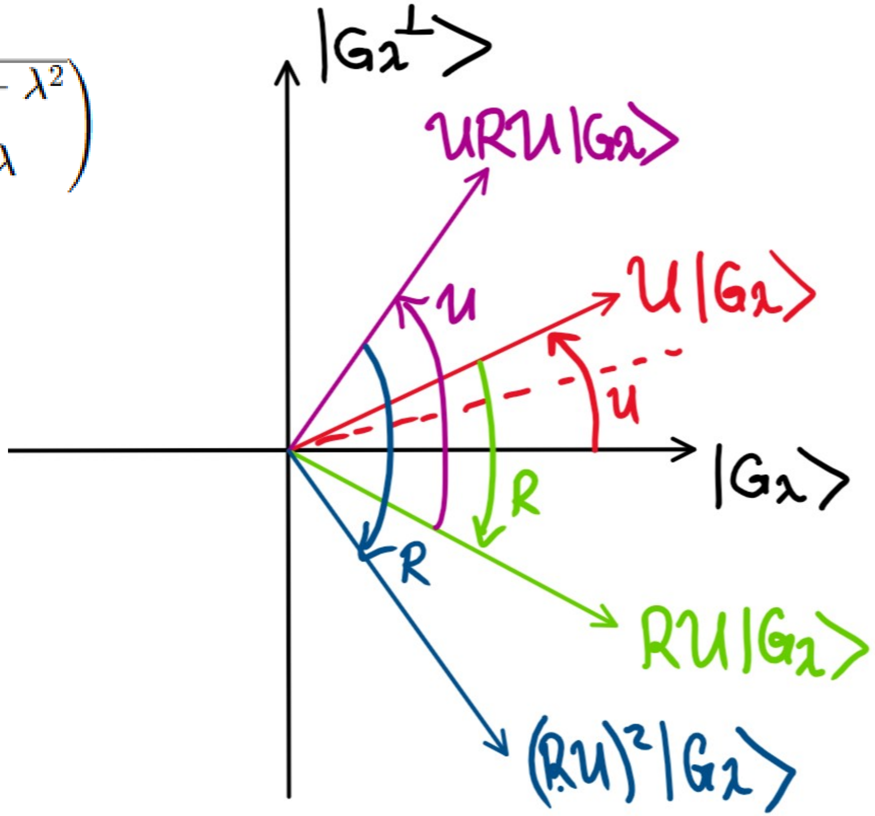
$$U|G_\lambda^\perp\rangle = \sqrt{1-\lambda^2}\,|G_\lambda\rangle - \lambda|G_\lambda^\perp\rangle$$

- Combine these to see that $U = \begin{pmatrix} \lambda & \sqrt{1-\lambda^2} \\ \sqrt{1-\lambda^2} & -\lambda \end{pmatrix}$

Qubitization

- Start with the ansatz $W = RU$, where $R = (2|G\rangle\langle G| - I_a) \otimes I_s$ is a reflection about $|G\rangle$
- For now, assume that $U^2 = I$, i.e. that U is also a reflection

$$R = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad U = \begin{pmatrix} \lambda & \sqrt{1-\lambda^2} \\ \sqrt{1-\lambda^2} & -\lambda \end{pmatrix}$$



$$W = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \lambda & \sqrt{1-\lambda^2} \\ \sqrt{1-\lambda^2} & -\lambda \end{pmatrix}$$

$$= \begin{pmatrix} \lambda & \sqrt{1-\lambda^2} \\ -\sqrt{1-\lambda^2} & \lambda \end{pmatrix}$$

→ We obtain the rotation W again!

Qubitization

- If $U^2 \neq I$, U and $|G\rangle$ can be **qubitized** into U' and $|G'\rangle$ such that $U'^2 = I$ and $\langle G'|W'|G'\rangle = \langle G'|RU'|G'\rangle = H$

- This costs an additional ancilla qubit, one controlled-U and one controlled-$U^\dagger$

→ We can always construct the iterate W!

Quantum Signal Processor

Definition 2 (Quantum Signal Processor). *A quantum signal processor solves the following.*

Inputs:
- *A Hermitian matrix* $\hat{H}$ *with bounded spectral norm* $\|\hat{H}\| \leq 1$.
- *A function* $f : [-1, 1] \to \mathbb{D}$, *where* $\mathbb{D}$ *is the complex unit disc, i.e.* $\forall x \in [-1, 1]$, $|f(x)| \leq 1$.

Resources:
- *An encoding of* $\hat{H} = \langle G|\hat{U}|G\rangle$ *by the oracles of Definition 1.*
- *A constant number of additional qubits.*
- *Arbitrary single and two-qubit gates.*

Outputs:
- *A standard-form encoding of* $f[\hat{H}]$, *i.e. a unitary* $\hat{Q}$ *where* $f[\hat{H}] = \langle 0|\hat{Q}|0\rangle$.

Operator function design

- How to get operator functions out of W?
- Our building block will be the **phased iterate** $\hat{W}_\phi = \hat{Z}_{\phi+\pi/2}\hat{W}\hat{Z}_{-\phi-\pi/2}$

  with $\hat{Z}_\phi = \bigoplus_\lambda \begin{pmatrix} e^{-i\phi} & 0 \\ 0 & -1 \end{pmatrix}_\lambda$

- Instead of adding powers of W, phased iterates are multiplied to form a valid quantum circuit:

$$\hat{W}_{\vec{\phi}} = \hat{W}_{\phi_Q} \cdots \hat{W}_{\phi_2}\hat{W}_{\phi_1} = \bigoplus_\lambda e^{-i\hat{\phi}_Q^\lambda \theta_\lambda} \cdots e^{-i\hat{\phi}_2^\lambda \theta_\lambda} e^{-i\hat{\phi}_1^\lambda \theta_\lambda}, \quad \hat{\phi}_j^\lambda = \cos(\phi_j)\hat{X}_\lambda + \sin(\phi_j)\hat{Y}_\lambda.$$

$$= \bigoplus_\lambda \underbrace{\mathcal{A}(2\theta_\lambda)\hat{\mathcal{I}}_\lambda + i\mathcal{B}(2\theta_\lambda)\hat{Z}_\lambda + i\mathcal{C}(2\theta_\lambda)\hat{X}_\lambda + i\mathcal{D}(2\theta_\lambda)\hat{Y}_\lambda.}_{e^{-i\lambda t}}$$

- Note: To achieve more general functions, a more complicated version of this is needed

Hamiltonian simulation

- The functions $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ that can be obtained are real-valued polynomials of degree Q

- There are some restrictions on which polynomials exactly

- We can approximate the time-evolution by truncating after $Q = \mathcal{O}(t + \log(1/\epsilon))$ terms

- Actual decomposition of the time-evolution:

$$e^{-i\lambda t} = A(\lambda) + iC(\lambda)$$

$$e^{-i\lambda t} = \underbrace{J_0(t) + 2 \sum_{k \text{ even}>0}^{\infty} (-1)^{k/2} J_k(t) T_k(\lambda)}_{\tilde{A}(\lambda)} + \underbrace{i2 \sum_{k \text{ odd}>0}^{\infty} (-1)^{(k-1)/2} J_k(t) T_k(\lambda)}_{\tilde{C}(\lambda)}$$

Simplifications

- We do not need to know |G>, only its oracle function
- H could be only normal instead of Hermitian
- QSP requires H to be bounded with ||H|| <= 1. But if it is not, we can rescale time $\tilde{t} = \alpha t$ to get

$$\tilde{H} = \frac{H}{||H||} = \frac{H}{\alpha}$$

Comparison with other methods

Algorithm	Model	Ancilla qubits	Query Complexity $\mathcal{O}(\cdot)$	Gates per Query $\mathcal{O}(\cdot)$ Oracle	Primitive
LC [7]	Sparse	$n+\mathcal{O}(m)$	$dt+\frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}$	Varies	$n+\mathrm{poly}(m)$
BCCKS [22]	$\sum_j \alpha_j \hat{U}_j$	$\mathcal{O}(\frac{\log(d)\log(\Vert\vec{\alpha}\Vert_1 t/\epsilon)}{\log\log(\Vert\vec{\alpha}\Vert_1 t/\epsilon)})$	$\frac{\Vert\vec{\alpha}\Vert_1 t\log(\Vert\vec{\alpha}\Vert_1 t/\epsilon)}{\log\log(\Vert\vec{\alpha}\Vert_1 t/\epsilon)}$	dC	$\log(d)$
LMR [5]	Mixed ρ	$n+1$	t^2/ϵ	Varies	$\log(n)$
Theorem 1	$\langle G\vert\hat{U}\vert G\rangle$	$\lceil\log_2(d)\rceil+2$	$t+\frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}$	Varies	$\log(d)$
Corollary 16	$\sum_j \alpha_j \hat{U}_j$	$\lceil\log_2(d)\rceil+2$	$\Vert\vec{\alpha}\Vert_1 t+\frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}$	dC	$\log(d)$
Corollary 17	Purified ρ	$n+\lceil\log_2(d)\rceil+2$	$t+\frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}$	Varies	$\log(n)$

Conclusion

- Using the unitary standard-form encoding of a Hamiltonian H, the iterate W was constructed
- W performs controlled operations on each eigenspace of H separately
- By combining „tuned“ versions of W, the phased iterates, polynomials of H were constructed
- With a careful choice of phases, the time-evolution operator appeared as a product of phased iterates
- The construction uses optimal oracle queries to G and U for given error and simulation time

Sources

- https://quantum-journal.org/papers/q-2019-07-12-163/pdf/
- https://pennylane.ai/qml/demos/tutorial_lcu_blockencoding/
- Nielsen, M.A. and Chuang, I.L. (2010) Quantum Computation and Quantum Information
- https://www.wikipedia.org/

– Thank you for your attention –

And special thanks for their advice to:

- René Wolters
- Markus Heinrich
- Felix Motzoi

UNIVERSITY
OF COLOGNE

ADVANCED SEMINAR ON QUANTUM COMPUTING [SOSE26]

HAMILTONIAN SIMULATION: QUBITIZATION & QUANTUM SIGNAL PROCESSING

based on a paper by Guang Hao Low and Isaac L. Chuang (2019)

Presented by Vincent Borlisch

UNIVERSITY
OF COLOGNE

Foto: Gregor Hübl

Institut / Lehrstuhl / Dezernat

17.07.2026