Quantum Algorithms for Modal Analysis in Structural Mechanics

Tobias Stollenwerk

Quantum Algorithms Group Lead
Institute for Quantum Computing Analytics (PGI-12)
Forschungszentrum Jülich

December 13, 2024, RWTH Aachen, Guest Lecture Quantum Computing for Engineering

JÜLICH
Forschungszentrum

Content

Credits

Sven Danz

Alessandro Ciani

Mario Berta

Stefan Schröder

Pascal Kienast

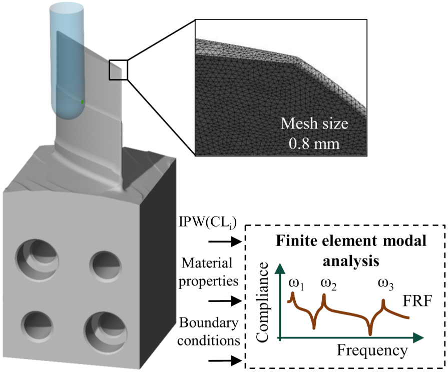
Frank Wilhelm-Mauch

Danz, Berta, Schröder, Kienast, Wilhelm, Ciani, arXiv:2405.08694
Schröder et. al. Procedia CIRP, Vol. 121, p. 13-18, 2024

Modal Analysis with Finite-Element-Method

Motivation – Vibrations in Blisk Milling

- Vibrations during milling process can lead to low quality surfaces
- Solution: Avoid eigenfrequencies during milling
- Eigenfrequencies change constantly due to material removal
- Knowledge of eigenfrequencies crucial for predictiong of response function



Schröder et. al. Procedia CIRP, Vol. 121, p. 13-18, 2024

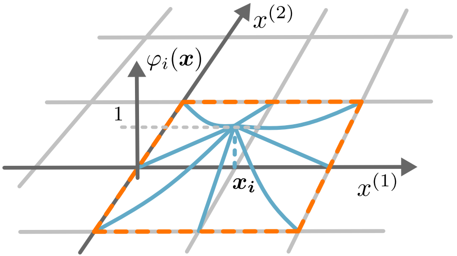
Finite Element Method (FEM) for Structural Mechanics

- Lagrangian for continuous elastic system

$$\mathcal{L} = \frac{1}{2}\int_\Omega \mathrm{d}^d x\, \rho \left(\frac{\partial u^{(\alpha)}}{\partial t}\right)^2 - \frac{1}{2}\sum_{\alpha,\beta,\gamma,\delta=1}^{d}\int_\Omega \mathrm{d}^d x\, Y^{(\alpha\beta\gamma\delta)} \frac{\partial u^{(\alpha)}}{\partial x^{(\beta)}}\frac{\partial u^{(\gamma)}}{\partial x^{(\delta)}}$$

- Discretized by use of local test functions

$$u^{(\alpha)}(\mathbf{x}) \approx \sum_i^N u_i^{(\alpha)} \varphi_i(\mathbf{x}) \qquad v^{(\alpha)}(\mathbf{x}) =: \frac{\partial u^{(\alpha)}(\mathbf{x})}{\partial t} \approx \sum_i^N v_i^{(\alpha)} \phi_i(\mathbf{x})$$



local 2d-test function for FEM

Danz, Ciani, Stollenwerk, Quantum oracle implementation for the finite element method (in preparation)

Finite Element Method (FEM) for Structural Mechanics

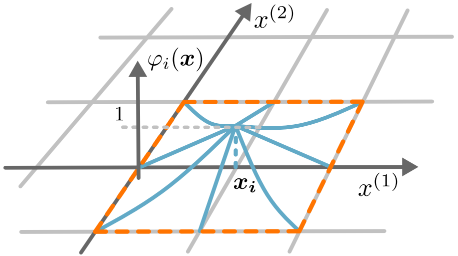
- Approximate discretized Lagrangian

$$\mathcal{L} \approx \frac{1}{2} \sum_{\alpha,\gamma=1}^{d} \sum_{i,j=1}^{N} \left[v_i^{(\alpha)} M_{ij}^{(\alpha\gamma)} v_j^{(\gamma)} - u_i^{(\alpha)} K_{ij}^{(\alpha\gamma)} u_j^{(\gamma)} \right] .$$

with mass and stiffness matrices

$$M_{ij}^{(\alpha\gamma)} = \int_\Omega d^d x \, \rho \phi_i \phi_j \leftarrow \text{ diagonal}$$

$$K_{ij}^{(\alpha\gamma)} = \sum_{\beta,\delta=1}^{d} \int_\Omega d^d x \, Y^{(\alpha\beta\gamma\delta)} \frac{\partial \varphi_i}{\partial x^{(\beta)}} \frac{\partial \varphi_j}{\partial x^{(\delta)}} = \text{ sparse}$$



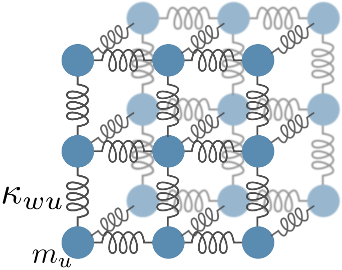
local 2d-test function for FEM

Danz, Ciani, Stollenwerk, Quantum oracle implementation for the finite element method (in preparation)

Modal Analysis using Finite Element Method (FEM)

- Euler-Lagrange Equations (Eq. of Motion)

$$
\begin{aligned}
0 &= -\frac{\partial \mathcal{L}}{\partial u_i^{(\alpha)}} + \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial v_i^{(\alpha)}} \\
&= \sum_{j=1}^{N}\sum_{\gamma=1}^{d} K_{ij}^{(\alpha\gamma)} u_j^{(\gamma)} + \sum_{j=1}^{N} M_{ij}\ddot{u}_j^{(\alpha)}, \\
&= K\vec{\mathbf{u}} + M\ddot{\vec{\mathbf{u}}}
\end{aligned}
$$



- Frequency domain ($\vec{\mathbf{u}}'(t) = \int_{-\infty}^{\infty} e^{-i\omega t} u'(\omega)\,\mathrm{d}\omega$)

$$
0 = K\vec{\mathbf{u}}' - \omega^2 M\vec{\mathbf{u}}'
$$

$$
\overset{\vec{\mathbf{y}}:=W^T M^{\frac{1}{2}}\vec{\mathbf{u}}'}{\Rightarrow} \quad 0 = (W^T \underbrace{M^{-\frac{1}{2}} K M^{-\frac{1}{2}}}_{=:H} W - \omega^2 \mathbb{1})\vec{\mathbf{y}} = \underbrace{W^T H W}_{=:\Lambda}\vec{\mathbf{y}} = \omega^2 \vec{\mathbf{y}}
$$

Danz, Ciani, Stollenwerk, Quantum oracle implementation for the finite element method (in preparation)

Response Function

- Apply external force $\vec{\mathbf{f}}$
- Response function G in time domain

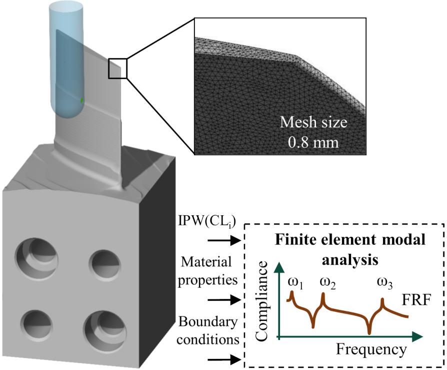
$$\vec{\mathbf{u}}(t) = \int_{-\infty}^{\infty} d\tau G(t-\tau)\vec{\mathbf{f}}(\tau)$$

- Response function G in frequency domain

$$\vec{\mathbf{u}}'(w) = G(w)\vec{\mathbf{f}}(w)$$

- Consider force on single point j and direction β:

$$G_{kj}^{\delta\beta}(\omega) = \frac{1}{\sqrt{m_k m_j}} \sum_{i\alpha} \frac{W_{ki}^{\delta\alpha} W_{ji}^{\beta\alpha}}{\lambda_i^{\alpha} - \omega^2}$$

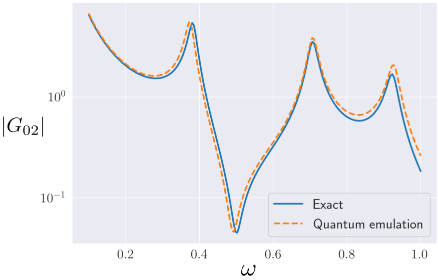


Danz, Berta, Schröder, Kienast, Wilhelm, Ciani, arXiv:2405.08694

Task for Quantum Computer

- Assume isotropic material $K_{ij}^{\alpha\beta} \propto \delta_{\alpha\beta}$
- Calculate local response function

$$G_{kk}(\omega) = \frac{1}{m_k} \sum_i \frac{W_{ki}^2}{\lambda_i - \omega^2}$$



Tasks for Quantum Computer

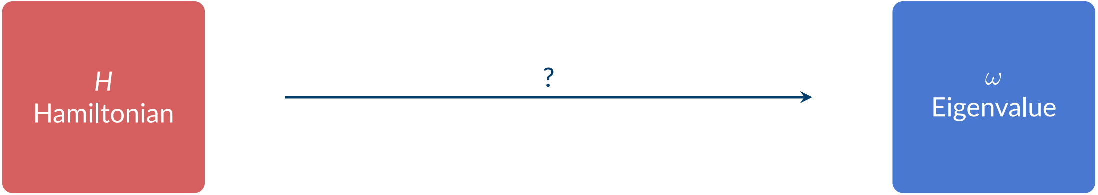
Given the matrix $H = M^{-\frac{1}{2}} K M^{-\frac{1}{2}}$, estimate

- Eigenvector coefficients W_{ki}
- Eigenvalues λ_i

Danz, Berta, Schröder, Kienast, Wilhelm, Ciani, arXiv:2405.08694

Eigenvalue Estimation via Phase Estimation

Eigenvalue Estimation via Phase Estimation – Overview



Hamiltonian rescaled to $\omega \in [0, 1)$

Eigenvalue Estimation via Phase Estimation – Overview

H
Hamiltonian

$U = e^{2\pi i H t}$
Unitary

ω
Eigenvalue

Hamiltonian rescaled to $\omega \in [0, 1)$

Eigenvalue Estimation via Phase Estimation – Overview

H
Hamiltonian

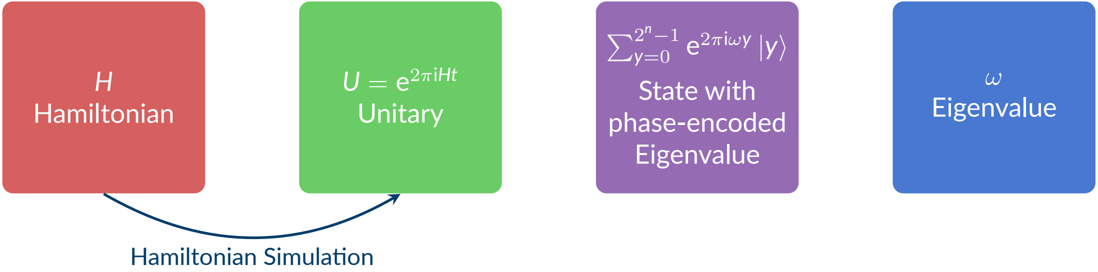
$U = e^{2\pi i H t}$
Unitary

$\sum_{y=0}^{2^n-1} e^{2\pi i \omega y} |y\rangle$
State with phase-encoded Eigenvalue

ω
Eigenvalue

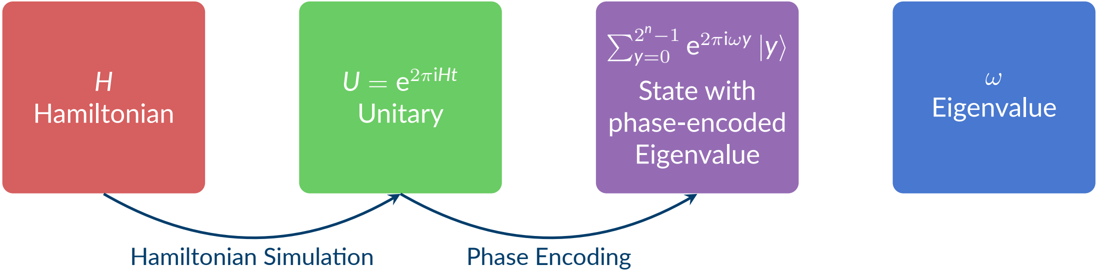
Hamiltonian rescaled to $\omega \in [0, 1)$

Eigenvalue Estimation via Phase Estimation – Overview



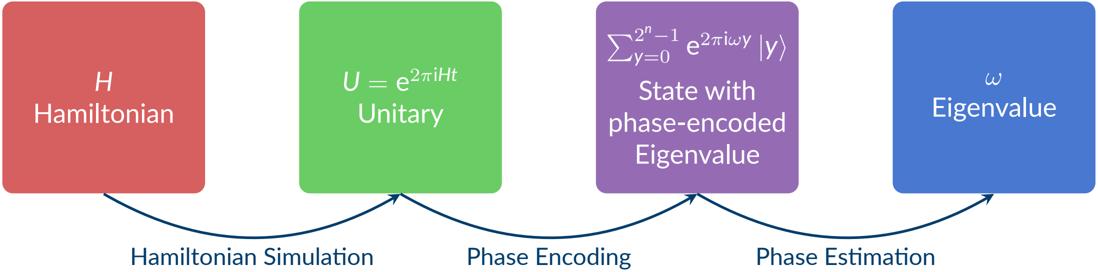
Hamiltonian rescaled to $\omega \in [0, 1)$

Eigenvalue Estimation via Phase Estimation – Overview



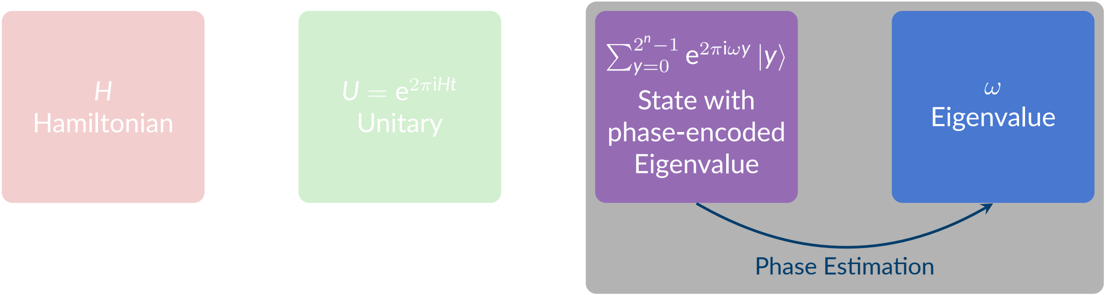
Hamiltonian rescaled to $\omega \in [0, 1)$
$|y\rangle$: binary encoding of integer y in $n-$ qubit state

Eigenvalue Estimation via Phase Estimation – Overview



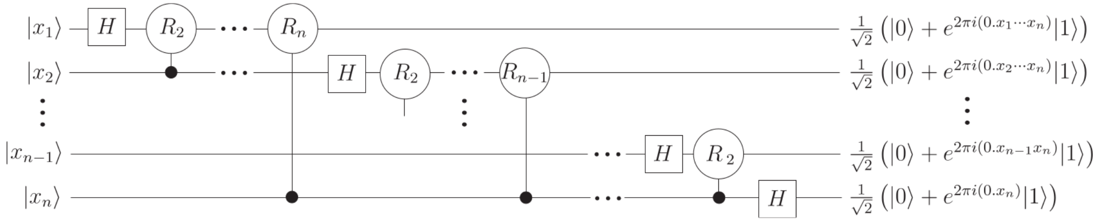
Hamiltonian rescaled to $\omega \in [0, 1)$
$|y\rangle$: binary encoding of integer y in $n-$ qubit state

Quantum Phase Estimation



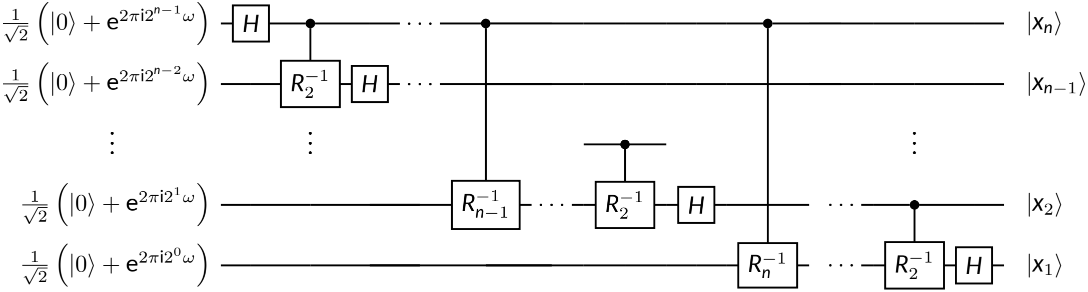
Hamiltonian rescaled to $\omega \in [0, 1)$
$|y\rangle$: binary encoding of integer y in $n-$ qubit state

Quantum Fourier Transformation (QFT)



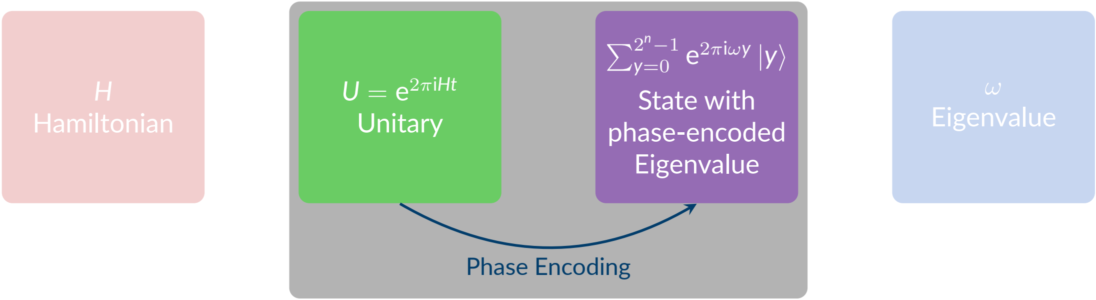
where $R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{pmatrix}$

Quantum Phase Estimation (QFT^{-1})



JÜLICH Forschungszentrum

Overview Quantum Eigenvalue Estimation



Hamiltonian rescaled to $\omega \in [0, 1)$

$|y\rangle$: binary encoding of integer y in $n-$ qubit state

How to prepare $\sum_{y=0}^{2^n-1} \mathrm{e}^{2\pi \mathrm{i}\omega \cdot y} |y\rangle$?

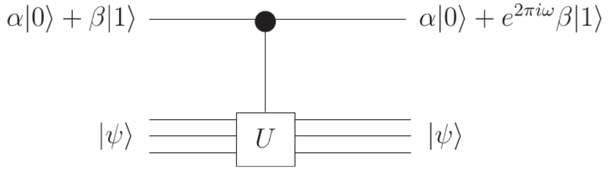
Consider a unitary U with eigenvector $|\psi\rangle$ and eigenvalue $\mathrm{e}^{2\pi \mathrm{i}\omega}$.

$$U |\psi\rangle = \mathrm{e}^{2\pi \mathrm{i}\omega} |\psi\rangle$$

Apply controlled unitary c$-U$ to $|1\rangle |\psi\rangle$

$$\mathrm{c}-U |1\rangle |\psi\rangle = |1\rangle U |\psi\rangle = \mathrm{e}^{2\pi \mathrm{i}\omega} |1\rangle |\psi\rangle$$

Apply to superposition

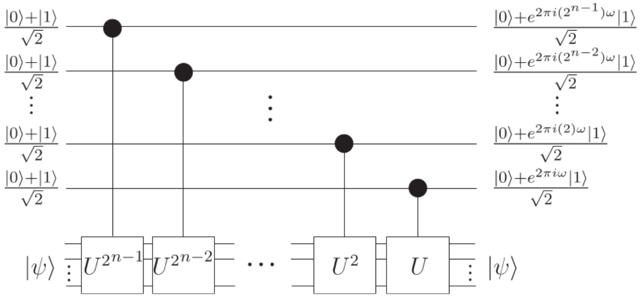


How to prepare $\sum_{y=0}^{2^n-1} \mathrm{e}^{2\pi \mathrm{i}\omega \cdot y} |y\rangle$?

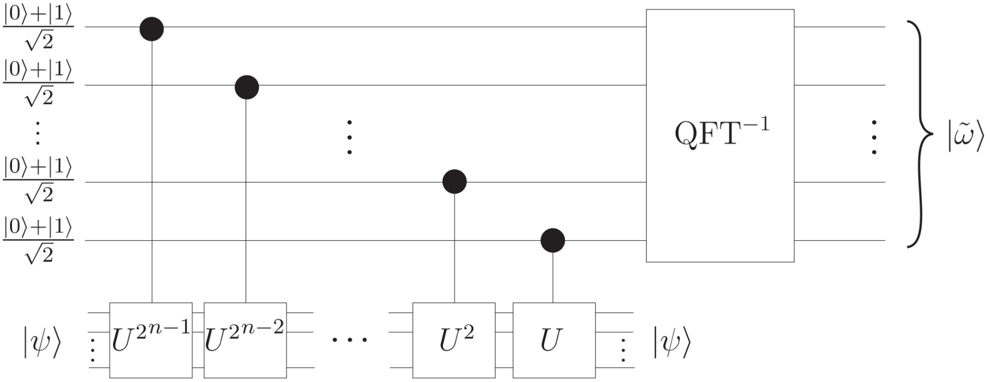
Since $U^v |\psi\rangle = \mathrm{e}^{(2\pi \mathrm{i}\omega)^v} |\psi\rangle = \mathrm{e}^{2\pi \mathrm{i}\omega \cdot v} |\psi\rangle$, apply controlled unitary $\mathrm{c}-U^2$

$$\mathrm{c}-U^{2^j} \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) |\psi\rangle = \left(\frac{|0\rangle + \mathrm{e}^{2\pi \mathrm{i}(2^j \omega)} |1\rangle}{\sqrt{2}} \right) |\psi\rangle$$

State preparation circuit



Eigenvalue Estimation Circuit



Caveat: We need to prepare the exact eigenvector of *U*: $|\psi\rangle$.

Eigenvalue Estimation given approximate Eigenvector

Suppose we have an approximation eigenvector $|\psi\rangle$, written in terms of exact eigenvectors $|\psi_i\rangle$, with $U|\psi_j\rangle = e^{2\pi i \omega_j}|\psi_j\rangle$

$$|\psi\rangle = \sum_j \alpha_j |\psi_j\rangle$$

Since QPE: $|0\rangle^{\otimes n}|\psi_j\rangle \mapsto |\omega_j\rangle|\psi_j\rangle$, By linearity, we have

$$\sum_j \alpha_j |0\rangle^{\otimes n}|\psi_j\rangle \mapsto \sum_j \alpha_j |\omega_j\rangle|\psi_j\rangle$$

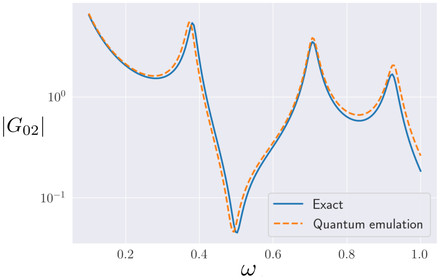
$\rightarrow |\alpha_j|^2$ is probability to get Eigenvalue ω_j

Quantum Algorithm for Modal Analysis

Task for Quantum Computer

- Assume isotropic material $K_{ij}^{\alpha\beta} \propto \delta_{\alpha\beta}$
- Calculate local response function

$$G_{kk}(\omega) = \frac{1}{m_k} \sum_i \frac{W_{ki}^2}{\lambda_i - \omega^2}$$



Tasks for Quantum Computer

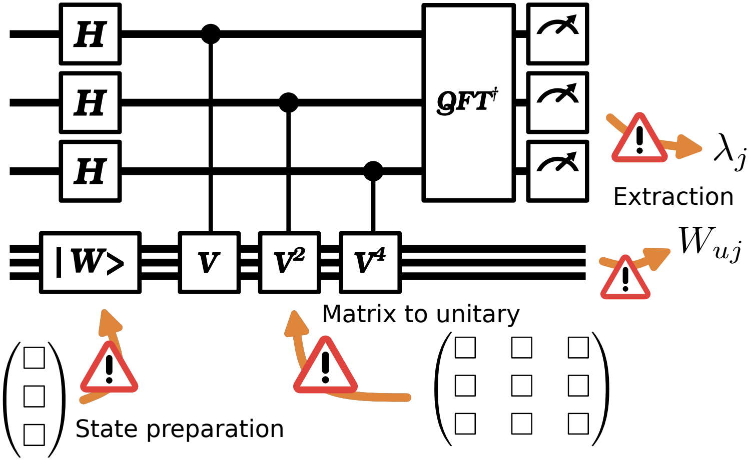
Given the matrix $H = M^{-\frac{1}{2}} K M^{-\frac{1}{2}}$, estimate

- Eigenvector coefficients W_{ki}
- Eigenvalues λ_i

Danz, Berta, Schröder, Kienast, Wilhelm, Ciani, arXiv:2405.08694

Problems with QPE for our use case: Input and Output

- Eigenstate needed in advance
- Matrix implementation cost
- Extraction cost



Danz, Berta, Schröder, Kienast, Wilhelm, Ciani, arXiv:2405.08694

Main Idea

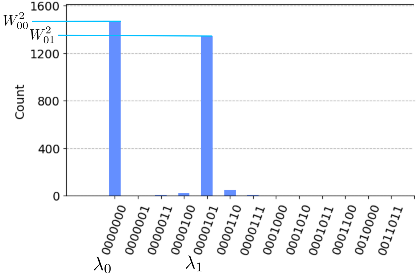
- Standard QPE uses estimate for Eigenstate

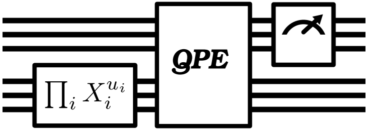
$$|\lambda_j\rangle = \sum_u W_{ju} |u\rangle$$

- Instead we prepare computational basis state (easy)

$$|u\rangle = \sum_j W_{uj} |\lambda_j\rangle$$

- Then we measure in the Eigenbasis to get Eigenvalues λ_j with probability W_{uj}^2.





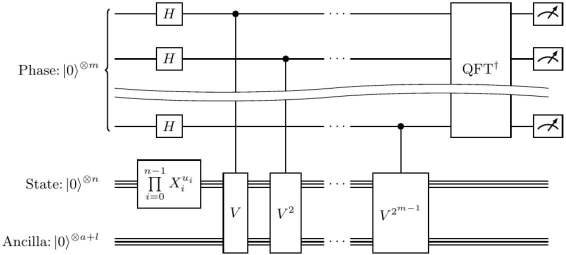
Danz, Berta, Schröder, Kienast, Wilhelm, Ciani, arXiv:2405.08694

Algorithm Overview

- Use different unitary to perform QPE

$$e^{iH} \to V$$

  where eigenvalues of V are $e^{\pm i \arccos(\alpha\lambda)}$ with α a normalization factor.
- V encodes e^{iH} and acts on a larger register



Danz, Berta, Schröder, Kienast, Wilhelm, Ciani, arXiv:2405.08694

Matrix Encoding (I)

- Goal: Unitary U_H that implements block-encoding for H

$$\langle 0|^{\otimes a} U_H |0\rangle^{\otimes a} = \alpha H,$$

- Action of U_H on the state $|0\rangle^{\otimes a} |\lambda\rangle$

$$U_H |0\rangle^{\otimes a} |\lambda\rangle = \alpha\lambda |0\rangle^{\otimes a} |\lambda\rangle + \sqrt{1-\alpha^2\lambda^2} \left|0\lambda^\perp\right\rangle,$$

where $\left|0\lambda^\perp\right\rangle$ is the perpendicular state such that $(\langle 0|^{\otimes a} \langle\psi|) \left|0\lambda^\perp\right\rangle = 0 \; \forall$ n-qubit state $|\psi\rangle$ and $\left\langle 0\lambda'^\perp \middle| 0\lambda^\perp\right\rangle = \delta_{\lambda\lambda'}$.

- Action of U_H on the perpendicular state

$$U_H \left|0\lambda^\perp\right\rangle = \sqrt{1-\alpha^2\lambda^2} |0\rangle^{\otimes a} |\lambda\rangle - \alpha\lambda \left|0\lambda^\perp\right\rangle.$$

Danz, Berta, Schröder, Kienast, Wilhelm, Ciani, arXiv:2405.08694

Matrix Encoding (II)

- This implies that $\forall\lambda$ a qubit-like subspace $\mathcal{H}^{(\lambda)} = \text{span}\{|0\rangle^{\otimes a}|\lambda\rangle, |0\lambda^{\perp}\rangle\}$, hence

$$U_H^{(\lambda)} = \begin{pmatrix} \alpha\lambda & \sqrt{1-\alpha^2\lambda^2} \\ \sqrt{1-\alpha^2\lambda^2} & -\alpha\lambda \end{pmatrix}$$

- However, the eigenvalues of this matrix are ± 1, since U_H. Rescaling to get information about λ

$$\begin{aligned} V^{(\lambda)} &= U_H^{(\lambda)}(2\Pi - I_{a+n})^{(\lambda)} \\ &= \begin{pmatrix} \alpha\lambda & -\sqrt{1-\alpha^2\lambda^2} \\ \sqrt{1-\alpha^2\lambda^2} & \alpha\lambda \end{pmatrix} \end{aligned}$$

where $\Pi = |0\rangle\langle 0|^{\otimes a} \otimes I_n$

Danz, Berta, Schröder, Kienast, Wilhelm, Ciani, arXiv:2405.08694

Matrix Encoding (III)

- The matrix $V^{(\lambda)}$ has eigenvalues

$$\mu_{\pm} = \alpha\lambda \pm i\sqrt{1-\alpha^2\lambda^2} = e^{\pm i \arccos(\alpha\lambda)},$$

and eigenvectors

$$|\mu_{\pm}\rangle = \frac{1}{\sqrt{2}}(|0\rangle^{\otimes a}|\lambda\rangle \mp i\left|0\lambda^{\perp}\right\rangle).$$

Danz, Berta, Schröder, Kienast, Wilhelm, Ciani, arXiv:2405.08694

Algorithm Overview – Benefits of Matrix Encoding

- Start with a superposition of all eigenstates

$$|\psi\rangle = \sum_{\lambda} c_\lambda |0\rangle^{\otimes a} |\lambda\rangle$$

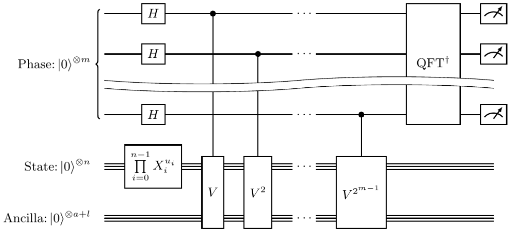
- Perform QPE with $V \to$ system will collapse in either

$$|\mu_+\rangle \text{ or } |\mu_-\rangle$$

- Values in the phase register:

$$+\arccos(\alpha\lambda) \text{ or } -\arccos(\alpha\lambda)$$

- Measurement of the ancilla register in the computational basis would give all zeros with probability $1/2$ and thus, prepare the eigenstate $|0\rangle^{\otimes a} |\lambda\rangle$.



Danz, Berta, Schröder, Kienast, Wilhelm, Ciani, arXiv:2405.08694

Block Encoding Implementation (I)

- For the implementation of V, a block encoding

$$\langle 0|^{\otimes a} U_H |0\rangle^{\otimes a} = \alpha H,$$

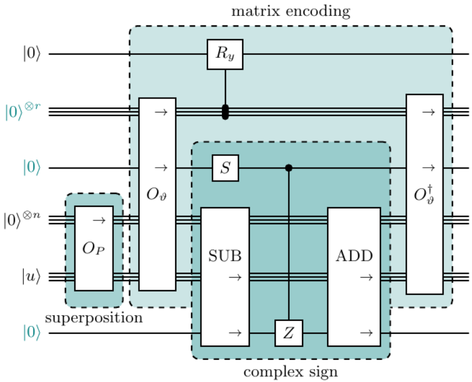
is needed.

- Use an $2(n+1)$ register including $a = n+2$ ancillas

$$U_H = U_T^\dagger \text{SWAP}_{n+1} U_T,$$

- Ansatz

$$U_T |0\rangle^{\otimes n+2} = \sum_{u=1}^{N} |\psi_u, 0, u\rangle \langle u| ,$$



Encoding Circuit U_T

Danz, Berta, Schröder, Kienast, Wilhelm, Ciani, arXiv:2405.08694

Block Encoding Implementation (II)

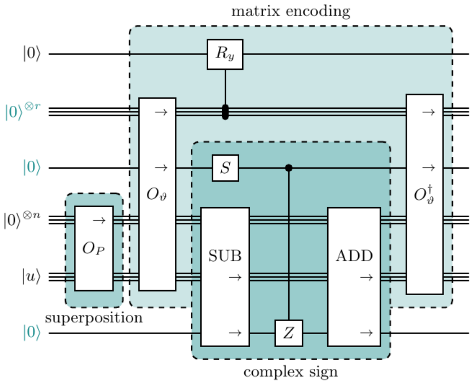
- We require

$$\alpha H_{uv} = \langle \psi_u, 0, u | 0, v, \psi_v \rangle \,.$$

- Solution is to set $\alpha = \frac{1}{s\|H\|_{\max}}$ and get

$$|\psi_u\rangle = \frac{1}{\sqrt{s}} \sum_{v \in \tilde{N}_G[u]} (i\mathrm{sgn}(u-v))^{\Theta(-H_{uv})}$$

$$\times \left(\sqrt{\frac{|H_{uv}|}{\|H\|_{\max}}} |0\rangle + \sqrt{1 - \frac{|H_{uv}|}{\|H\|_{\max}}} |1\rangle \right) |v\rangle \,,$$



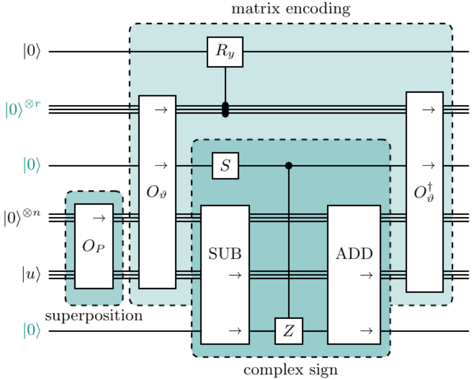
Encoding Circuit U_T

Danz, Berta, Schröder, Kienast, Wilhelm, Ciani, arXiv:2405.08694

Oracles – Neighbor Nodes

- Oracle for writing the neighbor nodes into additional register

$$O_P \left|u\right\rangle \left|0\right\rangle^{\otimes n} = \frac{1}{\sqrt{s}} \sum_{v \in N_G(u)} \left|u\right\rangle \left|v\right\rangle$$



Encoding Circuit U_T

Danz, Berta, Schröder, Kienast, Wilhelm, Ciani, arXiv:2405.08694

Oracles – Matrix Encoding

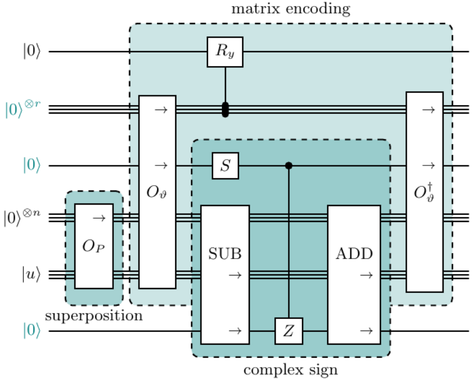
- Using additional register of $r+1$
- Encode information about the matrix entries

$$\vartheta_{uv} = \arccos \sqrt{\frac{|H_{uv}|}{||H||_{\max}}}$$

into an oracle O_θ

$$O_\vartheta |u,v\rangle |0\rangle^{\otimes n} = |u,v\rangle |\vartheta_{uv}, \Theta(-H_{uv})\rangle$$

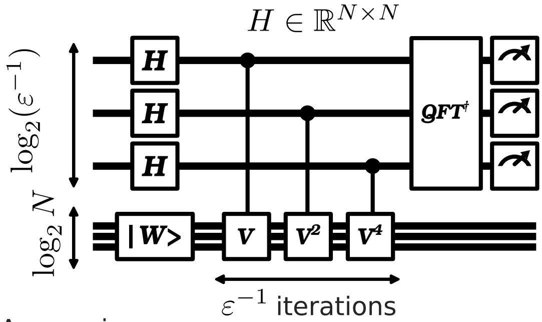
with $\Theta(\cdot)$ the Heaviside step function.



Encoding Circuit U_T

Danz, Berta, Schröder, Kienast, Wilhelm, Ciani, arXiv:2405.08694

Computational Complexity



Accuracies

$$\varepsilon \geq H_{\max}^{-1}|\lambda' - \lambda|$$

$$\delta \geq \left|\tilde{W}_{uj} - W_{uj}\right|$$

Memory $\mathcal{O}(\log(N\varepsilon^{-1}))$

Runtime $\mathcal{O}(\varepsilon^{-1} \times \delta^{-2} \log N)$

Danz, Berta, Schröder, Kienast, Wilhelm, Ciani, arXiv:2405.08694

Summary

- QPE can be used to simulate oscillations occurring in manufacturing.
- We solved the bottleneck issues from QPE for our use case
- The advantage in complexity is exponential in memory and problem dependent in runtime

Thanks