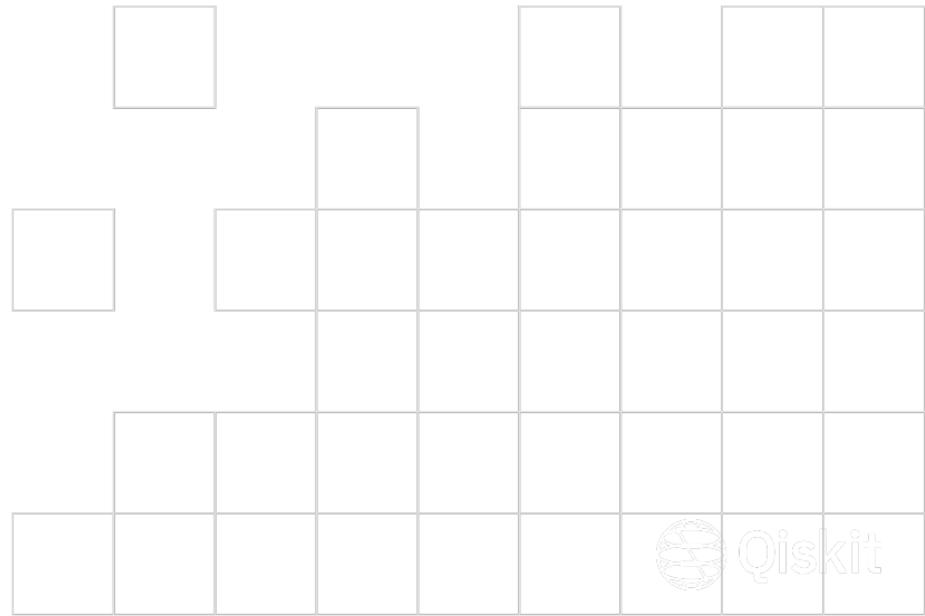


# Quantum algorithms for quantum dynamics

---

Alexander Miessen

PhD student @ IBM Quantum, IBM Research – Zürich



## Recap: Quantum dynamics in a nutshell

- Why quantum dynamics?
- Open vs. closed systems
- Real vs. imaginary time evolution

## Recap: Quantum dynamics in a nutshell

- Why quantum dynamics?
- Open vs. closed systems
- Real vs. imaginary time evolution

## Algorithms

- Terminology
- Decomposition methods
  - Product formulas
  - Linear combination of unitaries
- Qubitization
- Variational methods
  - EOM-based
  - Optimization-based

## Recap: Quantum dynamics in a nutshell

- Why quantum dynamics?
- Open vs. closed systems
- Real vs. imaginary time evolution

## Algorithms

- Terminology
- Decomposition methods
  - Product formulas
  - Linear combination of unitaries
- Qubitization
- Variational methods
- EOM-based
- Optimization-based

## Applications

- What's our goal?
- Quantum Chemistry
  - Molecular quantum dynamics
- Qiskit code example

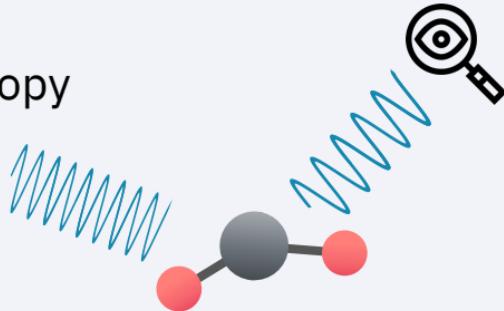
# Literature

- **Motta, Rice, arXiv:2109.02873 (2021)** – technical review on quantum algorithms for quantum dynamics focusing on decomposition methods and applications in chemistry
- **Cao et al., arXiv:1812.09976 (2018)** – review of quantum algorithms, including decomposition and variational methods, applications in chemistry
- **Miessen, Ollitrault, Tacchino, Tavernelli, submitted (2022)** – high-level perspective paper on quantum dynamics simulations on quantum computers, including broad overview of algorithms and applications
- Griffiths, Introduction to Quantum Mechanics
- Sakurai, Modern Quantum Mechanics
- Nielsen & Chuang, Quantum Computation and Quantum Information
- Breuer & Petruccione, The Theory of Open Quantum Systems

# Recap: Quantum dynamics in a nutshell

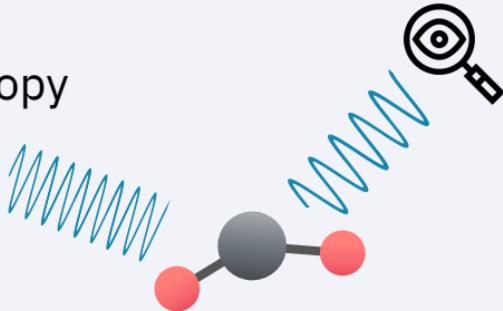
# Why quantum dynamics?

Example:  
spectroscopy



# Why quantum dynamics?

Example:  
spectroscopy

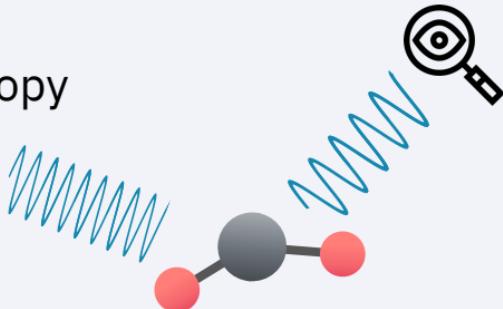


1. Prepare initial state


$$|\Psi(0)\rangle$$

# Why quantum dynamics?

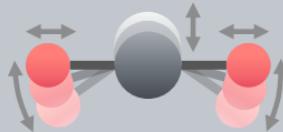
Example:  
spectroscopy



1. Prepare initial state

 $|\Psi(0)\rangle$ 

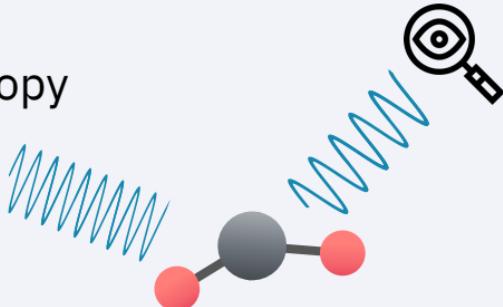
2. (Perturb and) evolve



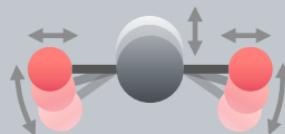
$$e^{-iHt} |\Psi(0)\rangle$$

# Why quantum dynamics?

Example:  
spectroscopy



2. (Perturb and) evolve



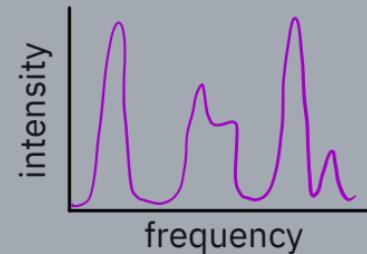
$$e^{-iHt} |\Psi(0)\rangle$$

1. Prepare initial state



$$|\Psi(0)\rangle$$

3. Measure



$$\langle \Psi(t) | \hat{O} | \Psi(t) \rangle$$

# Recap: open vs. closed systems

## *Closed systems*

- system completely isolated
- “pure” states  $|\Psi\rangle = |\Psi(t)\rangle$
- dynamics governed by time-dependent Schrödinger equation

$$i \frac{d|\Psi\rangle}{dt} = H |\Psi\rangle$$

# Recap: open vs. closed systems

## Closed systems

- system completely isolated
- “pure” states  $|\Psi\rangle = |\Psi(t)\rangle$
- dynamics governed by time-dependent Schrödinger equation

$$i \frac{d|\Psi\rangle}{dt} = H |\Psi\rangle$$

## Open systems

- system embedded in & interacting with larger environment
- “mixed” state
$$\rho(t) = \sum_i p_i |\Psi_i(t)\rangle \langle \Psi_i(t)|$$
- dynamics governed by more general Gorini–Kossakowski–Sudarshan–Lindblad equation

$$\frac{\partial \rho}{\partial t} = -i[H, \rho] + \mathcal{L}[\rho]$$

# Recap: real vs. imaginary time evolution

## *Real time evolution*

- “standard” – system evolves for time t
- **time-independent** Hamiltonian

$$|\Psi(t)\rangle = U_t |\Psi(0)\rangle, \quad U_t = e^{-iHt}$$

# Recap: real vs. imaginary time evolution

## Real time evolution

- “standard” – system evolves for time t
- **time-independent** Hamiltonian

$$|\Psi(t)\rangle = U_t |\Psi(0)\rangle, \quad U_t = e^{-iHt}$$

- **time-dependent** Hamiltonian

$$U_t = \mathcal{T} \exp\left(i \int_0^t dt' H(t')\right)$$

- example: short laser pulse exciting molecule

$$[H(t_1), H(t_2)] \neq 0$$

# Recap: real vs. imaginary time evolution

## Real time evolution

- “standard” – system evolves for time t
- **time-independent** Hamiltonian

$$|\Psi(t)\rangle = U_t |\Psi(0)\rangle, \quad U_t = e^{-iHt}$$

- **time-dependent** Hamiltonian

$$U_t = \mathcal{T} \exp\left(i \int_0^t dt' H(t')\right)$$

- example: short laser pulse exciting molecule

## Imaginary time evolution

- mathematical trick to compute stationary & thermodynamic properties
- Wick rotation  $it \rightarrow \tau$
- new propagator  $e^{-H\tau}$

# Recap: real vs. imaginary time evolution

## *Real time evolution*

- “standard” – system evolves for time t
- **time-independent** Hamiltonian

$$|\Psi(t)\rangle = U_t |\Psi(0)\rangle, \quad U_t = e^{-iHt}$$

- **time-dependent** Hamiltonian

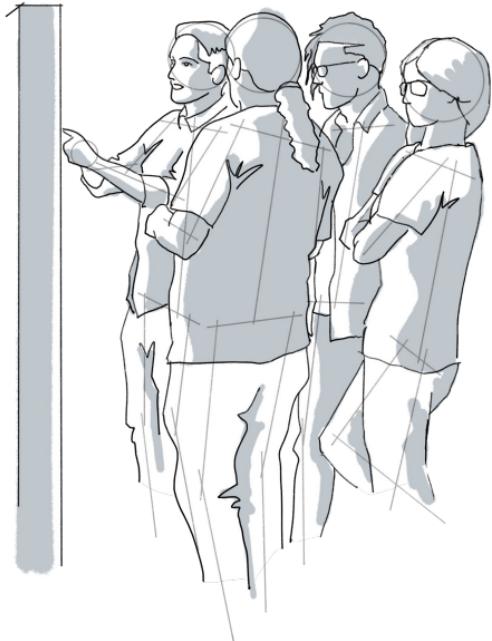
$$U_t = \mathcal{T} \exp\left(i \int_0^t dt' H(t')\right)$$

- example: short laser pulse exciting molecule

## *Imaginary time evolution*

- mathematical trick to compute stationary & thermodynamic properties
- Wick rotation  $it \rightarrow \tau$
- new propagator  $e^{-H\tau}$
- higher energies are exponentially suppressed
- Note: only combined action is unitary!

$$|\Phi(\tau)\rangle \rightarrow e^{-H\tau'} |\Phi(\tau)\rangle / \|e^{-H\tau'} |\Phi(\tau)\rangle\|$$



# Algorithms

# Why do we need algorithms?

- cannot directly implement evolution operator

$$U_t = e^{-iHt}$$

to obtain  $|\Psi(t)\rangle = U_t |\Psi(0)\rangle$

- **Why?**  $U_t$  is typically a complicated operator exponential

$$U_t = \exp(\alpha_1 XYII + \alpha_2 IZIX + \alpha_3 YXZY + \dots)$$

- But  $e^{A+B} \neq e^A e^B$  if  $[A, B] \neq 0$

→ cannot be easily implemented using one- and two-qubit gates

*! digital computing: every operation decomposed using a universal set of quantum gates !*

# Terminology: long- vs. near-term methods?



- common distinction in literature
- **long**-term methods
  - long circuits, many ancillas → feasible only with fault-tolerant QC
- **near**-term methods
  - hybrid quantum-classical → circuit depths small enough for noisy near-term hardware

# Terminology: long- vs. near-term methods?



- common distinction in literature
- **long**-term methods
  - long circuits, many ancillas → feasible only with fault-tolerant QC
- **near**-term methods
  - hybrid quantum-classical → circuit depths small enough for noisy near-term hardware

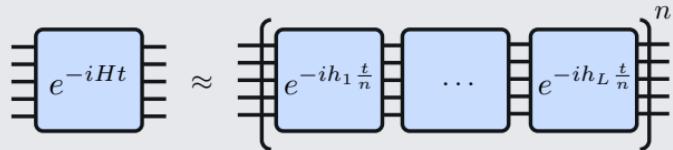
Does every algorithm always fit into one of the two categories?

# Terminology: long vs. near term methods? decomposition vs. variational methods

→ contains information about construction of algorithm

## *Decomposition*

- approximately decompose time-evolution operator  $\|e^{-iHt} - U\| \leq \epsilon$
  - yields rigorous scaling laws for resources
- estimate simulation cost and accuracy

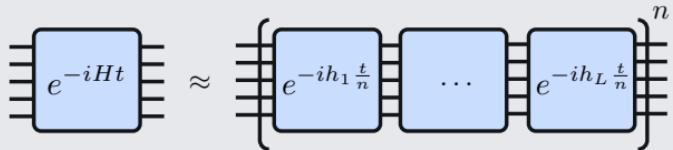


# Terminology: long vs. near term methods? decomposition vs. variational methods

→ contains information about construction of algorithm

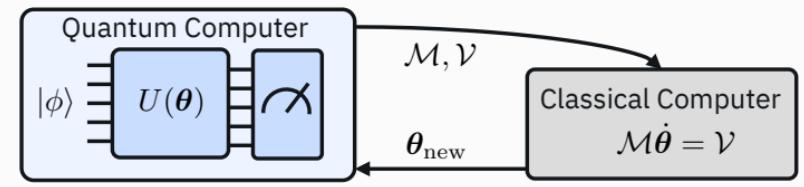
## Decomposition

- approximately decompose time-evolution operator  $\|e^{-iHt} - U\| \leq \epsilon$
  - yields rigorous scaling laws for resources
- estimate simulation cost and accuracy



## Variational

- approximate time evolution directly (wavefunction)
  - heuristic
- cannot estimate cost and accuracy



# Decomposition methods

# Product formulas

Hamiltonian as sum of unitaries

$$H = \sum_{l=1}^L \alpha_l V_l$$

$$|\Psi(t)\rangle = e^{-iHt} |\Psi(0)\rangle$$

Approximately decompose exponential

$$U(\Delta t) = e^{-i\Delta t \sum_l \alpha_l V_l}$$

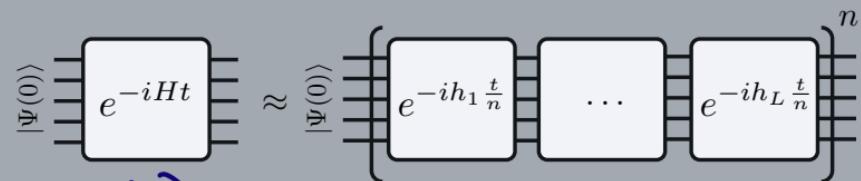
$$\approx \prod_l e^{-i\alpha_l V_l \Delta t}$$

$$\epsilon = O\left(\frac{t^2}{n} \sum_{i < j} \|V_i, V_j\|_1\right)$$

Split evolution into  $n$  steps

$$U(t) = (U(\Delta t))^n = (e^{-iH\frac{t}{n}})^n$$

$$|\Psi(t)\rangle \approx \left( \prod_l e^{-i\alpha_l V_l \frac{t}{n}} \right)^n |\Psi(0)\rangle$$



# Product formulas

- straight-forward: decompose into product of “simple” exponentials

Recall:  $e^{A+B} \neq e^A e^B$   $[A, B] \neq 0$

BCH → Zassenhaus formula

$$e^{t(A+B)} = e^{tA} e^{tB} e^{-\frac{t^2}{2}[A, B]} e^{\frac{t^3}{3!}([B, [A, B]] + \dots)} \times \dots$$

$$\downarrow A = \sum_i A_i$$

$$e^{t \sum_i A_i} = \left( \prod_i e^{t A_i} \right) \underbrace{\left( \prod_{i < j} e^{-\frac{t^2}{2}[A_i, A_j]} \right)}_{1 - \frac{t^2}{2} \sum_{i < j} [A_i, A_j]} \times \dots$$

(to 2nd order)

$$\Rightarrow \underline{e^{t \sum_i A_i}} = \prod_i e^{t A_i} + t^2 \sum_{i < j} [A_i, A_j] + \dots$$

$$\approx \prod_i e^{t A_i} + \mathcal{O}(t^2 \sum_{i < j} \| [A_i, A_j] \|)$$

# Product formulas

- straight-forward: decompose into product of “simple” exponentials
- for  $H = \sum_{l=1}^L \alpha_l V_l$  devide  $U(t) = e^{-iHt}$  into  $n$  time-steps, with  $\frac{t}{n} = \Delta t$

$$U(t) = (U(\Delta t))^n = (e^{-iH\frac{t}{n}})^n$$

- decompose

$$U(\Delta t) = e^{-i\Delta t \sum_l \alpha_l V_l} = \prod_l e^{-i\alpha_l V_l \Delta t} + c\Delta t^2 \sum_{i < j} [V_i, V_j] + \dots$$



$$(U(\Delta t))^n = \left( \prod_l e^{-i\alpha_l V_l \frac{t}{n}} \right)^n + c\frac{t^2}{n} \sum_{i < j} [V_i, V_j] + \dots$$

- Error depends on commutators! Most generally,  $\epsilon = \mathcal{O}(L^2 t^2 / n)$

# Product formulas

- **higher order** PFs (order  $2k, k \geq 1$ ) with error  $\mathcal{O}((Lt)^{2k+1}/n^{2k})$

$$U_2 = \left( \prod_{l=L}^1 e^{-i\alpha_l V_l \frac{\Delta t}{2}} \right) \left( \prod_{l=1}^L e^{-i\alpha_l V_l \frac{\Delta t}{2}} \right)$$

$$U_{2k+2}(\Delta t) = U_{2k}^2(s_{2k}\Delta t)U_{2k}((1 - 4s_{2k})\Delta t)U_{2k}^2(s_{2k}\Delta t), \quad s_{2k} = \frac{1}{4 - 4^{\frac{1}{2k+1}}}$$

$$e^{\sum_i \alpha_i v_i}$$

$$\sum_i \alpha_i v_i = \alpha_1 v_1 + \alpha_2 v_2 + \dots$$

$$= \alpha_{13} v_{13} + \alpha_{48} v_{48} + \alpha_1 v_1 + \dots$$

$$e^{\sum_i \alpha_i v_i} \approx \prod_i e^{\alpha_i v_i} = e^{\alpha_1 v_1} e^{\alpha_2 v_2} \dots$$
$$\neq e^{\alpha_2 v_2} e^{\alpha_{38} v_{38}} \dots$$

# Product formulas

- **higher order** PFs (order  $2k, k \geq 1$ ) with error  $\mathcal{O}((Lt)^{2k+1}/n^{2k})$

$$U_2 = \left( \prod_{l=L}^1 e^{-i\alpha_l V_l \frac{\Delta t}{2}} \right) \left( \prod_{l=1}^L e^{-i\alpha_l V_l \frac{\Delta t}{2}} \right)$$

$$U_{2k+2}(\Delta t) = U_{2k}^2(s_{2k}\Delta t)U_{2k}((1 - 4s_{2k})\Delta t)U_{2k}^2(s_{2k}\Delta t), \quad s_{2k} = \frac{1}{4 - 4^{\frac{1}{2k+1}}}$$

- **randomized** PFs – order of terms in  $H$  is arbitrary!

# Product formulas

- **higher order** PFs (order  $2k, k \geq 1$ ) with error  $\mathcal{O}((Lt)^{2k+1}/n^{2k})$

$$U_2 = \left( \prod_{l=L}^1 e^{-i\alpha_l V_l \frac{\Delta t}{2}} \right) \left( \prod_{l=1}^L e^{-i\alpha_l V_l \frac{\Delta t}{2}} \right)$$

$$U_{2k+2}(\Delta t) = U_{2k}^2(s_{2k}\Delta t)U_{2k}((1 - 4s_{2k})\Delta t)U_{2k}^2(s_{2k}\Delta t), \quad s_{2k} = \frac{1}{4 - 4^{\frac{1}{2k+1}}}$$

- **randomized** PFs – order of terms in  $H$  is arbitrary!
- example: QDRIFT (quantum stochastic drift)
  - randomly sample exponentials  $\{e^{-i\alpha_l V_l \frac{t}{n}}\}_{l \in \{1, \dots, L\}}$
  - error independent of  $L$

# Product formulas

- simple and versatile
  - no ancilla qubits
  - in practice, much reduced scaling for some problems (commutator structure)
  - can be used for time-dependent Hamiltonians
- 
- asymptotically bad scaling  
→ large gate-counts
  - impractical for first quantization  
(Hamiltonians with algebraic expressions of coordinates encoded in qubit register)

# Linear combination of unitaries

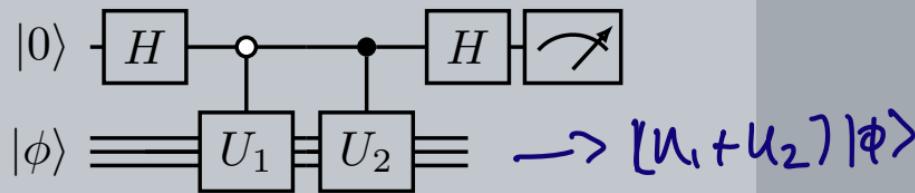
Hamiltonian as sum of unitaries

$$H = \sum_{l=0}^{L-1} \alpha_l U_l$$

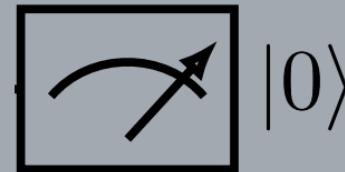
Taylor-expand evolution operator

$$U(\Delta t) = e^{-iH\frac{t}{n}} \approx \sum_{k=0}^K \frac{1}{k!} (-iHt/n)^k$$

Circuit implementation of non-unitary sum



Success only if all ancillas measured in  $|0\rangle$



# Linear combination of unitaries

- again,  $H = \sum_{l=0}^{L-1} \alpha_l U_l$  with each  $U_l$  **unitary**
- Taylor series of evolution operator  $U(\Delta t) = e^{-iH\frac{t}{n}} \approx \sum_{k=0}^K \frac{1}{k!} (-iHt/n)^k$

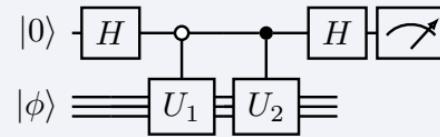
# Linear combination of unitaries

- again,  $H = \sum_{l=0}^{L-1} \alpha_l U_l$  with each  $U_l$  **unitary**
  - Taylor series of evolution operator  $U(\Delta t) = e^{-iH\frac{t}{n}} \approx \sum_{k=0}^K \frac{1}{k!} (-iHt/n)^k$
- not unitary!  
but lin. comb. of  
unitaries

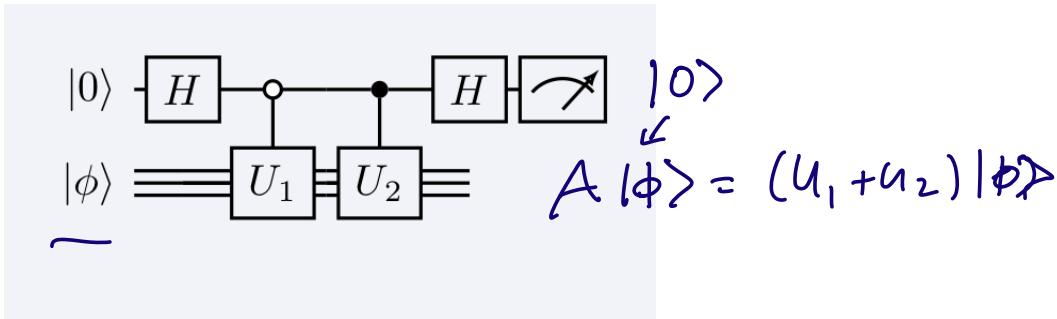

# Linear combination of unitaries

- again,  $H = \sum_{l=0}^{L-1} \alpha_l U_l$  with each  $U_l$  **unitary**
- Taylor series of evolution operator  $U(\Delta t) = e^{-iH\frac{t}{n}} \approx \sum_{k=0}^K \frac{1}{k!} (-iHt/n)^k$

**not unitary!**  
but lin. comb. of  
unitaries



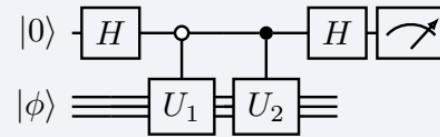
$$A = U_1 + U_2 \quad , \quad A \text{ non-unitary}$$



$$\begin{aligned}
 & H(|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U_2)(|1\rangle\langle 1| \otimes I + |0\rangle\langle 0| \otimes U_1)H|0\rangle|\phi\rangle \\
 &= \frac{1}{\sqrt{2}}(|0\rangle U_1 |\phi\rangle + |1\rangle |\phi\rangle) \\
 &= \frac{1}{\sqrt{2}}(|0\rangle U_1 |\phi\rangle + |1\rangle U_2 |\phi\rangle) \\
 &= \frac{1}{2}[(|0\rangle + |1\rangle)U_1 |\phi\rangle + (|0\rangle - |1\rangle)U_2 |\phi\rangle] \\
 &= \frac{1}{2}[|0\rangle(U_1 + U_2)|\phi\rangle + |1\rangle(U_1 - U_2)|\phi\rangle] =: |\tilde{\phi}\rangle \\
 \leadsto \langle 0|\tilde{\phi}\rangle &= \frac{1}{2}(U_1 + U_2)|\phi\rangle
 \end{aligned}$$

# Linear combination of unitaries

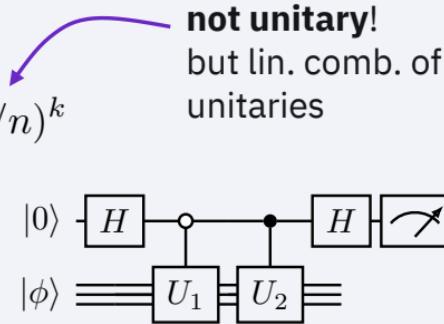
- again,  $H = \sum_{l=0}^{L-1} \alpha_l U_l$  with each  $U_l$  **unitary**
  - Taylor series of evolution operator  $U(\Delta t) = e^{-iH\frac{t}{n}} \approx \sum_{k=0}^K \frac{1}{k!} (-iHt/n)^k$
  - generally, for LCU  $A = \sum_{l=0}^{L-1} \alpha_l U_l$  ,  $|\alpha| = \sum_l \alpha_l$
- not unitary!  
but lin. comb. of  
unitaries



# Linear combination of unitaries

- again,  $H = \sum_{l=0}^{L-1} \alpha_l U_l$  with each  $U_l$  **unitary**
- Taylor series of evolution operator  $U(\Delta t) = e^{-iH\frac{t}{n}} \approx \sum_{k=0}^K \frac{1}{k!} (-iHt/n)^k$
- generally, for LCU  $A = \sum_{l=0}^{L-1} \alpha_l U_l$  ,  $|\alpha| = \sum_l \alpha_l$

$$W_{\text{prep}} |0\rangle = \sum_{l=0}^{L-1} \sqrt{\frac{\alpha_l}{|\alpha|}} |l\rangle , \quad W_{\text{sel}} = \sum_{l=0}^{L-1} |l\rangle \langle l| \otimes U_l \quad \text{s.th.} \quad W_{\text{sel}} |l\rangle |\phi\rangle = |l\rangle U_l |\phi\rangle$$



# Linear combination of unitaries

- again,  $H = \sum_{l=0}^{L-1} \alpha_l U_l$  with each  $U_l$  **unitary**

- Taylor series of evolution operator  $U(\Delta t) = e^{-iH\frac{t}{n}} \approx \sum_{k=0}^K \frac{1}{k!} (-iHt/n)^k$

- generally, for LCU  $A = \sum_{l=0}^{L-1} \alpha_l U_l$ ,  $|\alpha| = \sum_l \alpha_l$

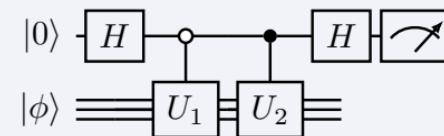
$$W_{\text{prep}} |0\rangle = \sum_{l=0}^{L-1} \sqrt{\frac{\alpha_l}{|\alpha|}} |l\rangle, \quad W_{\text{sel}} = \sum_{l=0}^{L-1} |l\rangle \langle l| \otimes U_l \quad \text{s.th.} \quad W_{\text{sel}} |l\rangle |\phi\rangle = |l\rangle U_l |\phi\rangle$$

$$W = (W_{\text{prep}}^\dagger \otimes I) W_{\text{sel}} (W_{\text{prep}} \otimes I) \quad \text{s.th.} \quad W |0\rangle |\phi\rangle = \frac{1}{\sqrt{|\alpha|}} |0\rangle A |\phi\rangle + \sqrt{1 - \frac{1}{|\alpha|^2}} |\perp\rangle$$

*#ancillas =  $O(K \log_2 L)$*

**project onto**  $|0\rangle$

**not unitary!**  
but lin. comb. of  
unitaries



# Linear combination of unitaries

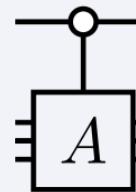
- asymptotically better scaling than PFs
- can be used for time-dependent Hamiltonians

generally, method of implementing LCU<sup>s</sup> extends reach of quantum computing to non-unitary operations

- lots of ancilla qubits  
 $\mathcal{O}(K \log L)$
- exponentially decreasing success probability (all ancillas have to be measured in  $|0\rangle$ )

# Qubitization + quantum signal processing

block-encoding



$$= \begin{pmatrix} A & \cdot \\ \cdot & \cdot \end{pmatrix}$$

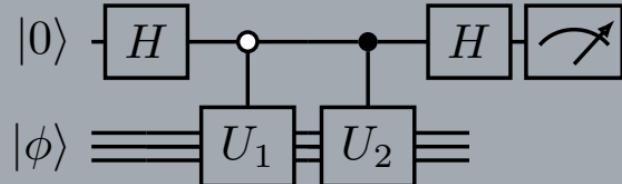
successively transform block-encoding

$$\begin{pmatrix} H & \cdot \\ \cdot & \cdot \end{pmatrix} \rightarrow \begin{pmatrix} f(H) & \cdot \\ \cdot & \cdot \end{pmatrix}$$

approximate time-evolution operator

$$\exp(-ixt) \approx f_1(x) + if_2(X)$$

implement via LCU



# Qubitization + quantum signal processing

- transform block-encoding of operator  $H$   $\begin{pmatrix} H & \cdot \\ \cdot & \cdot \end{pmatrix} \rightarrow \begin{pmatrix} f(H) & \cdot \\ \cdot & \cdot \end{pmatrix}$  into polynomial  $f(H)$
- achieved with series of rotations applied to block-encoding
- optimal scaling of gate-cost  $\mathcal{O}(t + \log \epsilon^{-1})$  and ancillas  $\mathcal{O}(\log_2 L)$

# Qubitization + quantum signal processing

- transform block-encoding of operator  $H$   $\begin{pmatrix} H & \cdot \\ \cdot & \cdot \end{pmatrix} \rightarrow \begin{pmatrix} f(H) & \cdot \\ \cdot & \cdot \end{pmatrix}$

- achieved with series of rotations applied to block-encoding

- optimal scaling of gate-cost  $\mathcal{O}(t + \log \epsilon^{-1})$  and ancillas  $\mathcal{O}(\log_2 L)$

- for Hamiltonian dynamics,

$$\exp(-ixt) \approx J_0(t) + 2 \sum_{k \text{ even}} (-1)^{\frac{k}{2}} J_k(t) T_k(x) + 2i \sum_{k \text{ odd}} (-1)^{\frac{k-1}{2}} J_k(t) T_k(x)$$

implement via qubitization
implement via qubitization

*Chebyshev poly.*

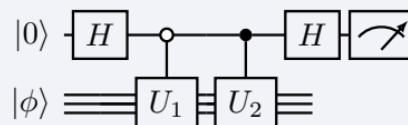
# Qubitization + quantum signal processing

- transform block-encoding of operator  $H$   $\begin{pmatrix} H & \cdot \\ \cdot & \cdot \end{pmatrix} \rightarrow \begin{pmatrix} f(H) & \cdot \\ \cdot & \cdot \end{pmatrix}$  into polynomial  $f(H)$
- achieved with series of rotations applied to block-encoding
- optimal scaling of gate-cost  $\mathcal{O}(t + \log \epsilon^{-1})$  and ancillas  $\mathcal{O}(\log_2 L)$
- for Hamiltonian dynamics,

$$\exp(-ixt) \approx J_0(t) + 2 \sum_{k \text{ even}} (-1)^{\frac{k}{2}} J_k(t) T_k(x) + 2i \sum_{k \text{ odd}} (-1)^{\frac{k-1}{2}} J_k(t) T_k(x)$$

implement via qubitization      implement via qubitization

combine both via  
LCU technique



$$\omega = \begin{pmatrix} H & -\sqrt{1-H^2} \\ \sqrt{1-H^2} & H \end{pmatrix}, \quad H = \sum_i \lambda_i |\lambda_i\rangle\langle\lambda_i|$$

$$\sqrt{1-H^2} := \sum_i \sqrt{1-\lambda_i^2} |\lambda_i\rangle\langle\lambda_i|$$

$$\omega = I \otimes H - i Y \otimes \sqrt{1-H^2}$$

for each  $|\lambda_j\rangle$  of  $H$ :

$$\begin{aligned}\omega |0\rangle |\lambda_j\rangle &= H |0\rangle |\lambda_j\rangle + \sqrt{1-H^2} |1\rangle |\lambda_j\rangle \\ &= \lambda_j |0\rangle |\lambda_j\rangle + \sqrt{1-\lambda_j^2} |1\rangle |\lambda_j\rangle\end{aligned}$$

$$\omega |1\rangle |\lambda_j\rangle = -\sqrt{1-\lambda_j^2} |0\rangle |\lambda_j\rangle + \lambda_j |1\rangle |\lambda_j\rangle$$

→  $\omega$  decomposes into direct sum of  $SU(2)$  matrices

→ separate Bloch-sphere for EACH  $\lambda_j \in \mathbb{R}$

$$\omega = \bigoplus_j \begin{pmatrix} \lambda_j & -\sqrt{1-\lambda_j^2} \\ \sqrt{1-\lambda_j^2} & \lambda_j \end{pmatrix} |\lambda_j\rangle\langle\lambda_j| \stackrel{?}{=} \bigoplus_j \omega_j |\lambda_j\rangle\langle\lambda_j|$$

$n$ -qubit Hamiltonian →  $2^n$  eigenstates  $|\lambda_j\rangle$

→  $N = 2^n$  separate Bloch-spheres

Jordan's Lemma:

if (for any block-encoding)  $W$  is composed of

2 reflections, one can **ALWAYS** find a basis in which  $W$  decomposes like above

$$W = \bigoplus_j W_j |z_j\rangle\langle z_j|$$

eigenvalues of  $W_j$ :  $w_j = z_j \pm \sqrt{1 - \lambda_j^2} = e^{\pm i \arccos \lambda_j}$

### Quantum Signal Processing (QSP)

Theorem (QSP): • matrix  $\tilde{W}(x) = \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix}$

↳ find phases

$\phi = (\phi_0, \dots, \phi_d) \in [-\pi, \pi)^{d+1}$  such that

$$\tilde{U}_\phi(x) = e^{i\phi_0 \sigma_2} \left( \prod_{j=1}^d \tilde{W}(x) e^{i\phi_j \sigma_2} \right) = \begin{pmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{pmatrix}$$

• iff: for  $P, Q \in \mathbb{C}[x]$ ,  $d \in \mathbb{N}$

(1)  $\deg P \leq d$ ,  $\deg Q \leq d-1$

$d=4$   
 $\deg P=4 \rightarrow P \text{ even}$   
 $x^2 + x^4$

(2)  $P$  has parity  $d \bmod 2$   
 $Q$  has parity  $d-1 \bmod 2$

$\deg Q=3 \rightarrow Q \text{ odd}$   
 $x + x^3$

(3)  $|P(x)|^2 + (1-x^2) |Q(x)|^2 = 1 \quad \forall x \in [-1, 1]$

$$\tilde{\omega}(I_j) = e^{-i\frac{\pi}{4}\sigma_2} \omega_j e^{i\frac{\pi}{4}\sigma_2} \quad \omega_j = \begin{pmatrix} \lambda_j & \sqrt{1-\lambda_j^2} \\ \sqrt{1-\lambda_j^2} & \bar{\lambda}_j \end{pmatrix}$$

Summarize:

$\omega = \begin{pmatrix} H & \cdot \\ \cdot & \cdot \end{pmatrix} \xrightarrow{\text{Compound of 2 reflections}} \omega = \bigoplus_j \omega_j |I_j\rangle\langle I_j|$

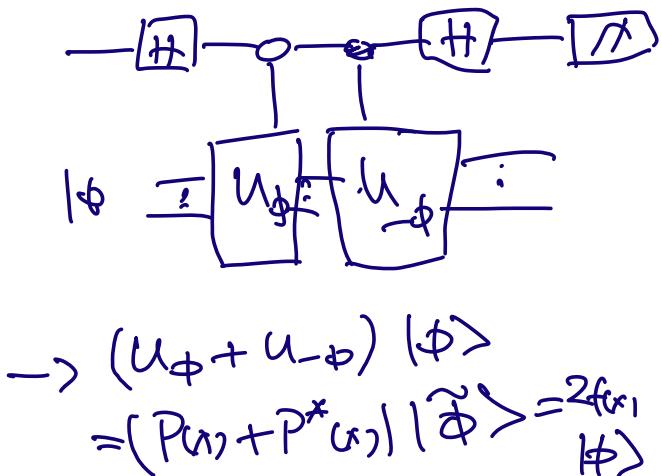
then, can use quantum signal processing to obtain  $U_\phi = \begin{pmatrix} P(H) & \cdot \\ \cdot & \cdot \end{pmatrix}$  with  $P(x)$  satisfying QSP theorem

with  $e^{i\phi_j \sigma_2} \rightarrow e^{i\phi_j \Pi}$   $\Pi$  generalize  $\sigma_2$  to multiple ancillas

$x \rightarrow f(x)$  real polynomial

$$f(x) = \operatorname{Re}(P(x)) = \frac{1}{2} (P(x) + P^*(x))$$

$$U_\phi = \begin{pmatrix} P(x) & \cdot \\ \cdot & \cdot \end{pmatrix} \quad U_{-\phi} = \begin{pmatrix} P^*(x) & \cdot \\ \cdot & \cdot \end{pmatrix}$$



$$L(\phi) = \text{dist} (\langle 0 | \tilde{U}_\phi(x_1) | 0 \rangle, f(x_1)) \\ = \frac{1}{d} \sum_{j=1}^d | \text{Re}(\langle 0 | \tilde{U}_\phi(x_j) | 0 \rangle) - f(x_j) |^2$$

[only  $2 \times 2$  matrices  $\rightarrow$  optimize  
 $\hookrightarrow$  gives  $H \rightarrow f(H)$

# Qubitization

- block-encoding of  $H = \sum_{l=0}^{L-1} \alpha_l U_l$  to obtain  $W_H = \begin{pmatrix} H & \cdot \\ \cdot & \cdot \end{pmatrix}$
- e.g., prepare / select

$$(\langle 0^m | \otimes I_n) W_H (|0^m\rangle \otimes I_n) = H$$

$$W_H = (W_{\text{prep}}^\dagger \otimes I_n) W_{\text{sel}} (W_{\text{prep}} \otimes I_n)$$

$$W_{\text{prep}} |0^m\rangle = \sum_l \sqrt{\frac{\alpha_l}{|\alpha|}} |l\rangle , \quad W_{\text{sel}} = \sum_l |l\rangle \langle l| \otimes U_l$$



# Qubitization

- block-encoding of  $H = \sum_{l=0}^{L-1} \alpha_l U_l$  to obtain  $W_H = \begin{pmatrix} H & \cdot \\ \cdot & \cdot \end{pmatrix}$
- e.g., prepare / select

$$(\langle 0^m | \otimes I_n) W_H (|0^m\rangle \otimes I_n) = H$$

$$W_H = (W_{\text{prep}}^\dagger \otimes I_n) W_{\text{sel}} (W_{\text{prep}} \otimes I_n)$$

$$W_{\text{prep}} |0^m\rangle = \sum_l \sqrt{\frac{\alpha_l}{|\alpha|}} |l\rangle , \quad W_{\text{sel}} = \sum_l |l\rangle \langle l| \otimes U_l$$

- if combined with a second reflection (with  $H |\lambda_j\rangle = \lambda_j |\lambda_j\rangle$ )

$$U_\Pi = 2 |0^m\rangle \langle 0^m| \otimes I_n - I_m \otimes I_n$$

# Qubitization

- block-encoding of  $H = \sum_{l=0}^{L-1} \alpha_l U_l$  to obtain  $W_H = \begin{pmatrix} H & \cdot \\ \cdot & \cdot \end{pmatrix}$
- e.g., prepare / select

$$(\langle 0^m | \otimes I_n) W_H (|0^m\rangle \otimes I_n) = H$$

$$W_H = (W_{\text{prep}}^\dagger \otimes I_n) W_{\text{sel}} (W_{\text{prep}} \otimes I_n)$$

$$W_{\text{prep}} |0^m\rangle = \sum_l \sqrt{\frac{\alpha_l}{|\alpha|}} |l\rangle, \quad W_{\text{sel}} = \sum_l |l\rangle \langle l| \otimes U_l$$

- if combined with a second reflection (with  $H |\lambda_j\rangle = \lambda_j |\lambda_j\rangle$ )

$$U_\Pi = 2 |0^m\rangle \langle 0^m| \otimes I_n - I_m \otimes I_n$$

- Jordan's lemma  $\rightarrow$  new block-encoding decomposes into direct sum

$$W := U_\Pi W_H = \bigoplus_j \begin{pmatrix} \lambda_j & -\sqrt{1-\lambda_j^2} \\ \sqrt{1-\lambda_j^2} & \lambda_j \end{pmatrix} =: \bigoplus_j (U_\Pi W_H)(\lambda_j)$$

# Quantum signal processing (Theorem)

- for a matrix  $\tilde{W}(x) = \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix} = e^{i \arccos(x)\sigma_x}$
- we can find phase factors  $\phi = (\phi_0, \dots, \phi_d) \in [-\pi, \pi]^{d+1}$  such that

$$\tilde{U}_\phi(x) = e^{i\phi_0\sigma_z} \prod_{j=1}^d \tilde{W}(x) e^{i\phi_j\sigma_z} = \begin{pmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{pmatrix}$$

# Quantum signal processing (Theorem)

- for a matrix  $\tilde{W}(x) = \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix} = e^{i \arccos(x)\sigma_x}$
- we can find phase factors  $\phi = (\phi_0, \dots, \phi_d) \in [-\pi, \pi]^{d+1}$  such that

$$\tilde{U}_\phi(x) = e^{i\phi_0\sigma_z} \prod_{j=1}^d \tilde{W}(x) e^{i\phi_j\sigma_z} = \begin{pmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{pmatrix}$$

- **Iff:** for  $P(x)Q(x) \in \mathbb{C}[x], d \in \mathbb{N}$ 
  1.  $\deg P \leq d$  and  $\deg Q \leq d - 1$
  2.  $P$  has parity  $d \pmod 2$ ,  $Q$  has parity  $d - 1 \pmod 2$
  3.  $|P(x)|^2 + (1 - x^2)|Q(x)|^2 = 1 \quad \forall x \in [-1, 1]$

# Quantum signal processing (Theorem)

- for a matrix  $\tilde{W}(x) = \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix} = e^{i \arccos(x) \sigma_x}$
- we can find phase factors  $\phi = (\phi_0, \dots, \phi_d) \in [-\pi, \pi]^{d+1}$  such that

$$\tilde{U}_\phi(x) = e^{i\phi_0 \sigma_z} \prod_{j=1}^d \tilde{W}(x) e^{i\phi_j \sigma_z} = \begin{pmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{pmatrix}$$

- Connection with  $W = U_\Pi W_H$

$$\tilde{W}(\lambda_j) = e^{i \arccos(\lambda_j) \sigma_x} = e^{-i \frac{\pi}{4} \sigma_z} \begin{pmatrix} \lambda_j & -\sqrt{1-\lambda_j^2} \\ \sqrt{1-\lambda_j^2} & \lambda_j \end{pmatrix} e^{i \frac{\pi}{4} \sigma_z}$$

# Quantum signal processing (Theorem)

- for a matrix  $\tilde{W}(x) = \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix} = e^{i \arccos(x)\sigma_x}$
- we can find phase factors  $\phi = (\phi_0, \dots, \phi_d) \in [-\pi, \pi]^{d+1}$  such that

$$\tilde{U}_\phi(x) = e^{i\phi_0\sigma_z} \prod_{j=1}^d \tilde{W}(x) e^{i\phi_j\sigma_z} = \begin{pmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{pmatrix}$$

- Connection with  $W = U_\Pi W_H$

$$\tilde{W}(\lambda_j) = e^{i \arccos(\lambda_j)\sigma_x} = e^{-i\frac{\pi}{4}\sigma_z} \begin{pmatrix} \lambda_j & -\sqrt{1-\lambda_j^2} \\ \sqrt{1-\lambda_j^2} & \lambda_j \end{pmatrix} e^{i\frac{\pi}{4}\sigma_z}$$

**Quantum signal processing = protocol to block-encode polynomials using qubitization!**

# Qubitization + QSP + LCU

- know how to implement block-encoded polynomials  $\begin{pmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{pmatrix}$
- BUT: QSP puts tight restrictions on  $P(x)$

# Qubitization + QSP + LCU

- know how to implement block-encoded polynomials  $\begin{pmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{pmatrix}$
  - BUT: QSP puts tight restrictions on  $P(x)$
1. approximate target function (that we want to implement, e.g.  $e^{ixt}$ ) with polynomials
  2. decompose polynomials into polynomials satisfying QSP theorem

$$\exp(-ixt) \approx J_0(t) + 2 \sum_{k \text{ even}} (-1)^{\frac{k}{2}} J_k(t) T_k(x) + 2i \sum_{k \text{ odd}} (-1)^{\frac{k-1}{2}} J_k(t) T_k(x)$$

# Qubitization – how to get phase factors?

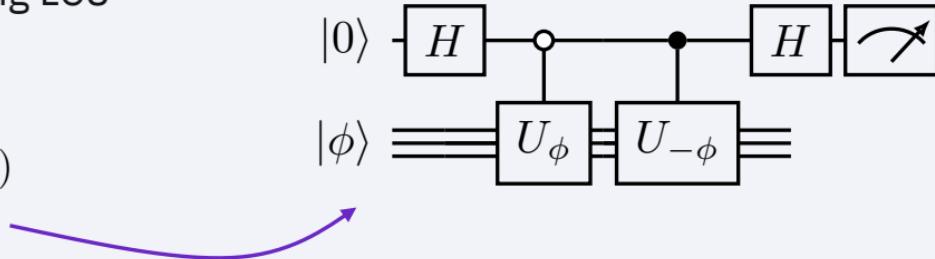
- know how to implement block-encoded polynomials  $\begin{pmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{pmatrix}$
  - BUT: QSP puts tight restrictions on  $P(x)$
1. approximate target function (that we want to implement, e.g.  $e^{ixt}$ ) with polynomials
  2. decompose polynomials into polynomials satisfying QSP theorem
  3. compute phases
  - optimization problem!

$$L(\Phi) = \text{dist}\left(\text{Re}[\langle 0|U_\Phi(x)|0\rangle], f(x)\right) = \frac{1}{d} \sum_{j=1}^d |\text{Re}[\langle 0|U_\Phi(x_j)|0\rangle] - f(x_j)|^2$$

# Qubitization + QSP + LCU

- know how to implement block-encoded polynomials  $\begin{pmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{pmatrix}$
- BUT: QSP puts tight restrictions on  $P(x)$
- 1. approximate target function (that we want to implement, e.g.  $e^{ixt}$ ) with polynomials
- 2. decompose polynomials into polynomials satisfying QSP theorem
- 3. compute phases
- 4. implement sum of polynomials using LCU
- for instance, real polynomial

$$f(x) = \text{Re}[P(x)] = \frac{1}{2}(P(x) + P^*(x))$$



Low, Chuang, Quantum **3**, 163 (2019)  
 Dong et al., arXiv:2002.11649 (2020)  
 Rajput et al., arXiv:2109.03308 (2021)

# Qubitization

- asymptotically optimal scaling gate counts
- constant ancilla overhead

generally, offers rich possibilities  
to implement arbitrary  
polynomials

[c.f. Quantum Singular Value  
Transformation]

- cannot be used for time-dependent Hamiltonians (in current form)

# Variational methods

# Equation of motion-based variational algorithms

- time-dependent variational state (time-dependent parameterization)

$$|\Psi(t)\rangle \approx |\Phi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta}) |\phi\rangle , \quad \boldsymbol{\theta} = \boldsymbol{\theta}(t)$$

- classical variational principle (not unique to quantum algorithm)

$$\delta \| i |\dot{\Phi}\rangle - H |\Phi\rangle \| = 0$$

McLachlan

# Equation of motion-based variational algorithms

- time-dependent variational state (time-dependent parameterization)

$$|\Psi(t)\rangle \approx |\Phi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta}) |\phi\rangle , \quad \boldsymbol{\theta} = \boldsymbol{\theta}(t)$$

- classical variational principle (not unique to quantum algorithm)

$$\delta \| i |\dot{\Phi}\rangle - H |\Phi\rangle \| = 0$$

- parameters  $\theta$  must be **real** in quantum computer!

$$-\boxed{R_z(\theta_i)}- = -\boxed{e^{-i\frac{\theta_i}{2}\sigma^z}}-$$

# Equation of motion-based variational algorithms

- time-dependent variational state (time-dependent parameterization)

$$|\Psi(t)\rangle \approx |\Phi(\theta)\rangle = U(\theta) |\phi\rangle , \quad \theta = \theta(t)$$

- classical variational principle (not unique to quantum algorithm)

$$\delta \| i |\dot{\Phi}\rangle - H |\Phi\rangle \| = 0$$

- parameters  $\theta$  must be **real** in quantum computer!

$$R_z(\theta_i) = e^{-i \frac{\theta_i}{2} \sigma^z}$$

- resulting equations of motion (EOMs) for parameters:

$$\mathcal{M}\dot{\theta} = \nu \quad \text{with}$$

$$\theta(t+\Delta t) = g(\theta(t), \dot{\theta}(t), \nu)$$

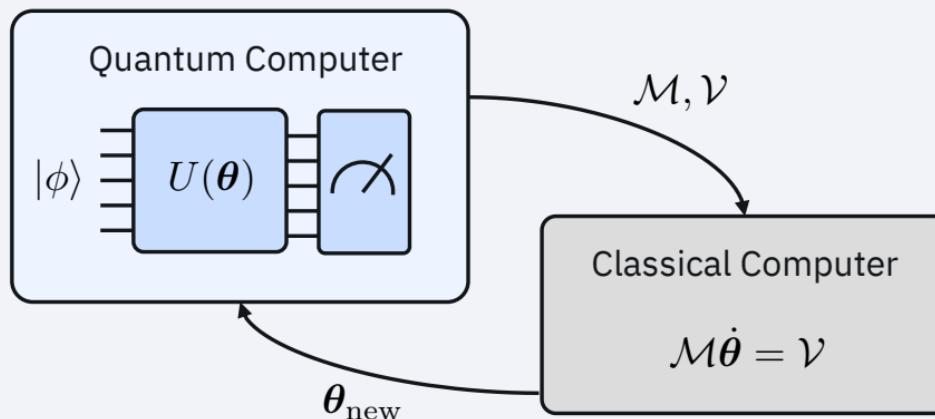
$$\mathcal{M}_{ij} = \Re \left( \frac{\partial \langle \Phi |}{\partial \theta_i} \frac{\partial | \Phi \rangle}{\partial \theta_j} - \frac{\partial \langle \Phi |}{\partial \theta_i} | \Phi \rangle \langle \Phi | \frac{\partial | \Phi \rangle}{\partial \theta_j} \right) \quad \nu_i = \Im \left( \frac{\partial \langle \Phi |}{\partial \theta_i} H | \Phi \rangle - \frac{\partial \langle \Phi |}{\partial \theta_i} | \Phi \rangle \langle \Phi | H | \Phi \rangle \right)$$

Li, Benjamin, Phys Rev X **7**, 021050 (2017)  
 Yuan et al., Quantum **3**, 191 (2019)  
 Hackl et al., SciPost Physics **9**, 048 (2020)

# Equation of motion-based variational algorithms

- time-dependent variational state  $|\Phi(\theta)\rangle = U(\theta) |\phi\rangle$  with EOM  $\mathcal{M}\dot{\theta} = \mathcal{V}$
- in quantum computer,  $\theta$  must be **real** (gate angles)
- solve iteratively

$$R_z(\theta_i) = e^{-i\frac{\theta_i}{2}\sigma^z}$$



# Optimization-based variational algorithms

- time-dependent variational state (time-dependent parameterization)

$$|\Psi(t)\rangle \approx |\Phi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta}) |\phi\rangle , \quad \boldsymbol{\theta} = \boldsymbol{\theta}(t)$$

1. at each step: minimize cost function with "approximate time-step"

$$\boldsymbol{\theta}' = \arg \min_{\boldsymbol{\theta}} \frac{1 - |\langle \Phi(\boldsymbol{\theta}') | e^{-iH\Delta t} | \Phi(\boldsymbol{\theta}) \rangle|}{\Delta t^2}$$

2. variationally approximate evolution operator

$$(e^{-iH\frac{t}{n}})^n \approx W(\boldsymbol{\theta}) D(n\gamma, \Delta t) W(\boldsymbol{\theta})$$

# Variational algorithms

- constant circuit depth for entire evolution  
→ near-term friendly

## EOM-based

---

- much shorter circuit depth **if** good ansatz is found

## Optimization-based

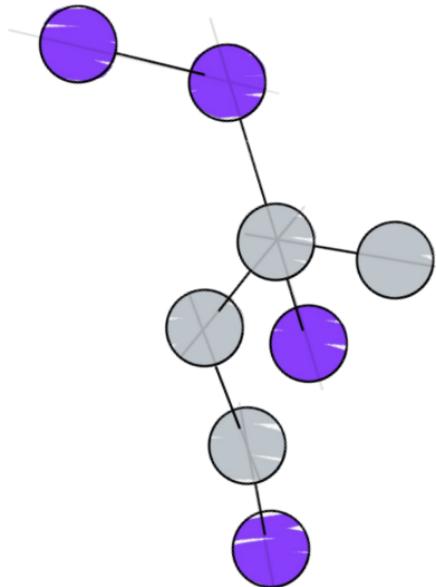
---

- less measurements compared to EOM-based

- generally heuristic, no estimates of cost and accuracy
- highly dependent on choice of ansatz and system

- 
- large number of measurements, prohibitive for relevant system sizes
  - can be unstable due to matrix-inversion

- 
- possibly larger circuit / ancilla overhead compared to EOM-based



# Applications

# Ultimate goal: solve hard problems!

## 1. Identify hard problems

- limits of classical algorithms
- system size, entanglement, ...

## 2. Can we do better on a quantum computer?

- representation of problem
- extract desired information

## 3. In which regime could quantum advantage occur?

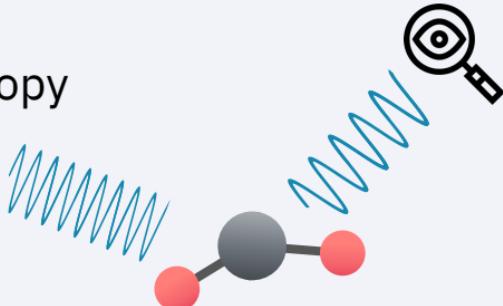
- resource requirements (qubits, gates)
- desired accuracy

## 4. Which application will likely lead to quantum advantage first?

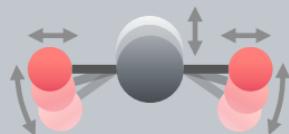
# Quantum Chemistry

# Why quantum dynamics?

Example:  
spectroscopy



2. (Perturb and) evolve



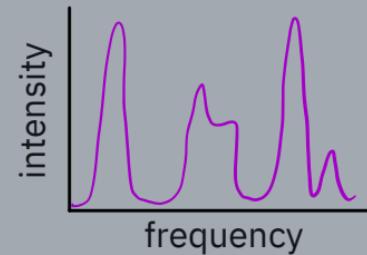
$$e^{-iHt} |\Psi(0)\rangle$$

1. Prepare initial state



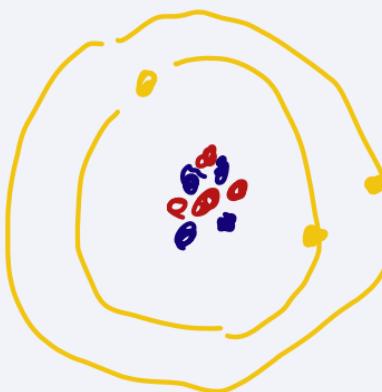
$$|\Psi(0)\rangle$$

3. Measure



$$\langle \Psi(t) | \hat{O} | \Psi(t) \rangle$$

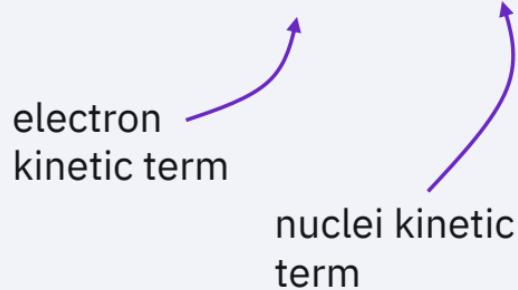
# Molecular quantum mechanics



$$H = - \sum_i \frac{\nabla_{e,i}^2}{2} - \sum_i \frac{\nabla_{N,i}^2}{2M_i} + \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i < j} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} - \sum_{i,j} \frac{Z_j}{|\mathbf{r}_i - \mathbf{R}_j|}$$

# Molecular quantum mechanics

$$H = - \sum_i \frac{\nabla_{e,i}^2}{2} - \sum_i \frac{\nabla_{N,i}^2}{2M_i} + \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i < j} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} - \sum_{i,j} \frac{Z_j}{|\mathbf{r}_i - \mathbf{R}_j|}$$



The diagram illustrates the components of the molecular Hamiltonian. It shows the full equation above, with two specific terms highlighted by purple arrows pointing upwards from below. The first term, labeled "electron kinetic term", corresponds to the sum of the first two terms in the equation. The second term, labeled "nuclei kinetic term", corresponds to the third term.

electron  
kinetic term

nuclei  
kinetic  
term

# Molecular quantum mechanics

$$H = - \sum_i \frac{\nabla_{e,i}^2}{2} - \sum_i \frac{\nabla_{N,i}^2}{2M_i} + \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i < j} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} - \sum_{i,j} \frac{Z_j}{|\mathbf{r}_i - \mathbf{R}_j|}$$

electron  
kinetic term

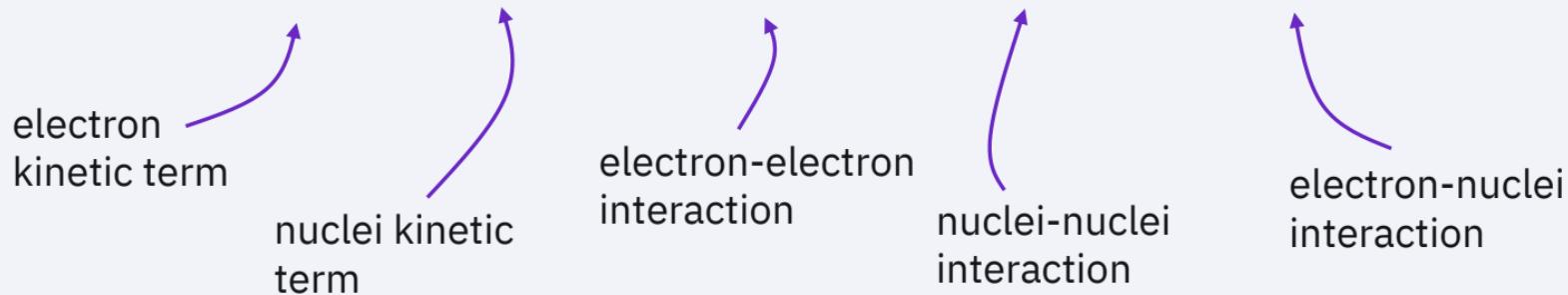
nuclei kinetic  
term

electron-electron  
interaction

nuclei-nuclei  
interaction

# Molecular quantum mechanics

$$H = - \sum_i \frac{\nabla_{e,i}^2}{2} - \sum_i \frac{\nabla_{N,i}^2}{2M_i} + \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i < j} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} - \sum_{i,j} \frac{Z_j}{|\mathbf{r}_i - \mathbf{R}_j|}$$



# Molecular quantum mechanics

$$H = - \sum_i \frac{\nabla_{e,i}^2}{2} - \sum_i \frac{\nabla_{N,i}^2}{2M_i} + \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i < j} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} - \sum_{i,j} \frac{Z_j}{|\mathbf{r}_i - \mathbf{R}_j|}$$

kinetic terms

potential terms

electron kinetic term

nuclei kinetic term

electron-electron interaction

nuclei-nuclei interaction

electron-nuclei interaction

# Molecular quantum mechanics

$$H = - \sum_i \frac{\nabla_{\text{e},i}^2}{2} - \sum_i \frac{\nabla_{\text{N},i}^2}{2M_i} + \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i < j} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} - \sum_{i,j} \frac{Z_j}{|\mathbf{r}_i - \mathbf{R}_j|}$$

- full description extremely challenging!  $|\Psi(\mathbf{r}, \mathbf{R})\rangle$  many DOFs

# Molecular quantum mechanics

$$H = - \sum_i \frac{\nabla_{e,i}^2}{2} - \sum_i \frac{\nabla_{N,i}^2}{2M_i} + \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i < j} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} - \sum_{i,j} \frac{Z_j}{|\mathbf{r}_i - \mathbf{R}_j|}$$

- full description extremely challenging!  $|\Psi(\mathbf{r}, \mathbf{R})\rangle$  many DOFs
- usually in **Born-Oppenheimer (BO) approximation**
- **Idea:**
  - electrons much lighter than nuclei  $m_e \ll M_N$
  - move more rapidly
  - electrons see nuclei as “fixed”
  - nuclei only feel “mean field of electrons”

# Molecular quantum mechanics

$$H = - \sum_i \frac{\nabla_{e,i}^2}{2} - \sum_i \frac{\nabla_{N,i}^2}{2M_i} + \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i < j} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} - \sum_{i,j} \frac{Z_j}{|\mathbf{r}_i - \mathbf{R}_j|}$$

- electrons see nuclei as “fixed”  $|\Psi(\mathbf{r}, \mathbf{R})\rangle \approx \phi(\mathbf{r}; \mathbf{R})\chi(\mathbf{R})$

# Molecular quantum mechanics

$$H = - \sum_i \frac{\nabla_{\mathbf{e},i}^2}{2} - \sum_i \frac{\nabla_{\mathbf{N},i}^2}{2M_i} + \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i < j} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} - \sum_{i,j} \frac{Z_j}{|\mathbf{r}_i - \mathbf{R}_j|}$$

- electrons see nuclei as “fixed”  $|\Psi(\mathbf{r}, \mathbf{R})\rangle \approx \phi(\mathbf{r}; \mathbf{R})\chi(\mathbf{R})$

$$H_{\text{el}}(\mathbf{R})\phi(\mathbf{r}; \mathbf{R}) = V(\mathbf{R})\phi(\mathbf{r}; \mathbf{R})$$

# Molecular quantum mechanics

$$H = - \sum_i \frac{\nabla_{\text{e},i}^2}{2} - \sum_i \frac{\nabla_{\text{N},i}^2}{2M_i} + \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i < j} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} - \sum_{i,j} \frac{Z_j}{|\mathbf{r}_i - \mathbf{R}_j|}$$

- electrons see nuclei as “fixed”  $|\Psi(\mathbf{r}, \mathbf{R})\rangle \approx \phi(\mathbf{r}; \mathbf{R})\chi(\mathbf{R})$

$$H_{\text{el}}(\mathbf{R})\phi(\mathbf{r}; \mathbf{R}) = V(\mathbf{R})\phi(\mathbf{r}; \mathbf{R})$$

$$\begin{aligned} H |\Psi(\mathbf{r}, \mathbf{R})\rangle &= [T_N + H_{\text{el}}(\mathbf{R})]\phi(\mathbf{r}; \mathbf{R})\chi(\mathbf{R}) \\ &= V(\mathbf{R})\phi\chi - \sum_i \frac{1}{2M_i} \left[ \phi \nabla_{\text{N},i}^2 \chi + (\nabla_{\text{N},i}\phi)(\nabla_{\text{N},i}\chi) + (\nabla_{\text{N},i}^2\phi)\chi \right] \end{aligned}$$

# Molecular quantum mechanics

$$H = - \sum_i \frac{\nabla_{\text{e},i}^2}{2} - \sum_i \frac{\nabla_{\text{N},i}^2}{2M_i} + \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i < j} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} - \sum_{i,j} \frac{Z_j}{|\mathbf{r}_i - \mathbf{R}_j|}$$

- electrons see nuclei as “fixed”  $|\Psi(\mathbf{r}, \mathbf{R})\rangle \approx \phi(\mathbf{r}; \mathbf{R})\chi(\mathbf{R})$

$$H_{\text{el}}(\mathbf{R})\phi(\mathbf{r}; \mathbf{R}) = V(\mathbf{R})\phi(\mathbf{r}; \mathbf{R})$$

$$\begin{aligned} H |\Psi(\mathbf{r}, \mathbf{R})\rangle &= [T_N + H_{\text{el}}(\mathbf{R})]\phi(\mathbf{r}; \mathbf{R})\chi(\mathbf{R}) \\ &= V(\mathbf{R})\phi\chi - \sum_i \frac{1}{2M_i} \left[ \phi \nabla_{\text{N},i}^2 \chi + (\nabla_{\text{N},i}\phi)(\nabla_{\text{N},i}\chi) + (\nabla_{\text{N},i}^2\phi)\chi \right] \end{aligned}$$

- nuclei only feel “mean field of electrons”

$$H_{\text{BO}}\chi(\mathbf{R}) = [T_N + V(\mathbf{R})]\chi(\mathbf{R})$$

# Molecular quantum mechanics

- electrons see nuclei as “fixed”
- nuclei only feel “mean field of electrons”

# Molecular quantum mechanics

- electrons see nuclei as “fixed”
- nuclei only feel “mean field of electrons”
- one step further:

$$|\Psi(\mathbf{r}, \mathbf{R})\rangle = \sum_{n=0}^K \phi_n(\mathbf{r}; \mathbf{R}) \chi_n(\mathbf{R})$$

- same procedure gives electronic solutions



$$H_{\text{el}}(\mathbf{R})\phi_n(\mathbf{r}; \mathbf{R}) = V_n(\mathbf{R})\phi_n(\mathbf{r}; \mathbf{R})$$
  

$$H_{\text{BO}}\chi(\mathbf{R}) = [T_{\text{N}} + V_n(\mathbf{R})]\chi(\mathbf{R})$$

# Molecular quantum mechanics

- electrons see nuclei as “fixed”
- nuclei only feel “mean field of electrons”
- one step further:

$$|\Psi(\mathbf{r}, \mathbf{R})\rangle = \sum_{n=0}^K \phi_n(\mathbf{r}; \mathbf{R}) \chi_n(\mathbf{R})$$

- same procedure gives electronic solutions

$$H_{\text{el}}(\mathbf{R}) \phi_n(\mathbf{r}; \mathbf{R}) = V_n(\mathbf{R}) \phi_n(\mathbf{r}; \mathbf{R})$$

$$H_{\text{BO}} \chi(\mathbf{R}) = [T_{\text{N}} + V_n(\mathbf{R})] \chi(\mathbf{R})$$

**potential energy surfaces**

# Molecular quantum mechanics

- electrons see nuclei as “fixed”
- nuclei only feel “mean field of electrons”
- one step further:

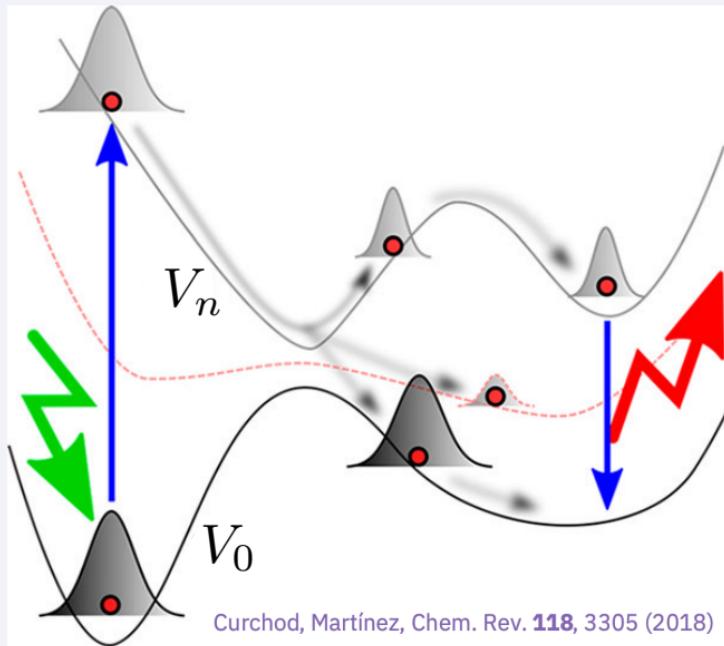
$$|\Psi(\mathbf{r}, \mathbf{R})\rangle = \sum_{n=0}^K \phi_n(\mathbf{r}; \mathbf{R}) \chi_n(\mathbf{R})$$

- same procedure gives electronic solutions

$H_{\text{el}}(\mathbf{R})\phi_n(\mathbf{r}; \mathbf{R}) = V_n(\mathbf{R})\phi_n(\mathbf{r}; \mathbf{R})$

$H_{\text{BO}}\chi(\mathbf{R}) = [T_N + V_n(\mathbf{R})]\chi(\mathbf{R})$

**potential energy surfaces**



# Molecular quantum mechanics – exact!

$$H = - \sum_i \frac{\nabla_{\mathbf{e},i}^2}{2} - \sum_i \frac{\nabla_{\mathbf{N},i}^2}{2M_i} + \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i < j} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} - \sum_{i,j} \frac{Z_j}{|\mathbf{r}_i - \mathbf{R}_j|}$$

- exact expansion (complete basis)

$$|\Psi(\mathbf{r}, \mathbf{R})\rangle = \sum_{n=0}^{\infty} \phi_n(\mathbf{r}; \mathbf{R}) \chi_n(\mathbf{R})$$

$$H_{\text{el}}(\mathbf{R}) \phi_n(\mathbf{r}; \mathbf{R}) = V_n(\mathbf{R}) \phi_n(\mathbf{r}; \mathbf{R})$$

**potential energy surfaces**

$$H |\Psi(\mathbf{r}, \mathbf{R})\rangle = \sum_n [T_N + H_{\text{el}}(\mathbf{R})] \phi_n \chi_n$$

$$= \sum_n \left( V_n(\mathbf{R}) \phi_n \chi_n - \sum_i \frac{1}{2M_i} \left[ \phi_n \nabla_{\mathbf{N},i}^2 \chi_n + (\nabla_{\mathbf{N},i} \phi_n)(\nabla_{\mathbf{N},i} \chi_n) + (\nabla_{\mathbf{N},i}^2 \phi_n) \chi_n \right] \right)$$

# Molecular quantum mechanics – exact!

$$H = - \sum_i \frac{\nabla_{\text{e},i}^2}{2} - \sum_i \frac{\nabla_{\text{N},i}^2}{2M_i} + \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i < j} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} - \sum_{i,j} \frac{Z_j}{|\mathbf{r}_i - \mathbf{R}_j|}$$

- potential for **quantum computation**

$$|\Psi(\mathbf{r}, \mathbf{R})\rangle = \sum_{n=0}^{\infty} \phi_n(\mathbf{r}; \mathbf{R}) \chi_n(\mathbf{R})$$

$$\begin{aligned} H |\Psi(\mathbf{r}, \mathbf{R})\rangle &= \sum_n [T_N + H_{\text{el}}(\mathbf{R})] \phi_n \chi_n \\ &= \sum_n \left( V_n(\mathbf{R}) \phi_n \chi_n - \sum_i \frac{1}{2M_i} \left[ \phi_n \nabla_{\text{N},i}^2 \chi_n + (\nabla_{\text{N},i} \phi_n)(\nabla_{\text{N},i} \chi_n) + (\nabla_{\text{N},i}^2 \phi_n) \chi_n \right] \right) \end{aligned}$$

# Molecular quantum mechanics – exact!

$$H = - \sum_i \frac{\nabla_{\mathbf{e},i}^2}{2} - \sum_i \frac{\nabla_{\mathbf{N},i}^2}{2M_i} + \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i < j} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} - \sum_{i,j} \frac{Z_j}{|\mathbf{r}_i - \mathbf{R}_j|}$$

- potential for **quantum computation**

**directly encode  
and evolve this!**

$$|\Psi(\mathbf{r}, \mathbf{R})\rangle = \sum_{n=0}^{\infty} \phi_n(\mathbf{r}; \mathbf{R}) \chi_n(\mathbf{R})$$

$$H |\Psi(\mathbf{r}, \mathbf{R})\rangle = \sum_n [T_N + H_{\text{el}}(\mathbf{R})] \phi_n \chi_n$$

$$= \sum_n \left( V_n(\mathbf{R}) \phi_n \chi_n - \sum_i \frac{1}{2M_i} \left[ \phi_n \nabla_{\mathbf{N},i}^2 \chi_n + (\nabla_{\mathbf{N},i} \phi_n) (\nabla_{\mathbf{N},i} \chi_n) + (\nabla_{\mathbf{N},i}^2 \phi_n) \chi_n \right] \right)$$

# Molecular quantum mechanics – exact!

$$H = - \sum_i \frac{\nabla_{\mathbf{e},i}^2}{2} - \sum_i \frac{\nabla_{\mathbf{N},i}^2}{2M_i} + \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i < j} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} - \sum_{i,j} \frac{Z_j}{|\mathbf{r}_i - \mathbf{R}_j|}$$

- potential for **quantum computation**

**directly encode  
and evolve this!**

$$|\Psi(\mathbf{r}, \mathbf{R})\rangle = \sum_{n=0}^{\infty} \phi_n(\mathbf{r}; \mathbf{R}) \chi_n(\mathbf{R})$$

$$H |\Psi(\mathbf{r}, \mathbf{R})\rangle = \sum_n [T_N + H_{\text{el}}(\mathbf{R})] \phi_n \chi_n$$

$$= \sum_n \left( V_n(\mathbf{R}) \phi_n \chi_n - \sum_i \frac{1}{2M_i} \left[ \phi_n \nabla_{\mathbf{N},i}^2 \chi_n + (\nabla_{\mathbf{N},i} \phi_n) (\nabla_{\mathbf{N},i} \chi_n) + (\nabla_{\mathbf{N},i}^2 \phi_n) \chi_n \right] \right)$$

# Qiskit code example