- Quantum algorithms
  - Trotter simulations of dynamics
- Variational quantum algorithms
  - Introduction & motivation

Hamiltonian simulation:

Goal: compute time-evolution of a quantum system described by Hamiltonian (= energy functional) H.

Many applications:

- Real-time dynamics of quantum systems
  - investigate nonequilibrium behavior
    * chemical reactions
    * scattering experiments
    * phase transformations, synthesis, modeling experimental measurements (optics, quenches, (non) linear transport)
    * fundamental interest in understanding nonequilibrium matter (criticality, ETH - thermalization, MIPT)
  - adiabatic state preparation: $H(t) = H_0\left(1-\frac{t}{T}\right) + H_1 \frac{t}{T}$.
    * slowly preparing ground states of desired Hamiltonians
    * general optimization problems, e.g., find GS of MFIM $H = \sum_{j=1}^{n} \left(h_{x,j} X_j + h_{z,j} Z_j + J_{ij} Z_i Z_j \right)$

- Imaginary-time evolution

  * prepare GS of Hamiltonians : $|\psi_0\rangle$.

  * prepare thermal states of Hamiltonians [e.g. Gibbs state $\frac{e^{-\beta H}}{Z}$)

We will focus on NISQ-implementable approaches:

- Lie-Suzuki-Trotter product formula (PF) approach
- Randomized compilation
- Multi-product formulas
- Variational quantum algorithms (hybrid quantum classical algorithms)

Lie-Suzuki-Trotter product formulas (PF):

Follow lecture notes by Ronald de Wolf lec.

Want to implement $U(t) = e^{-iHt}$

for $2^n \times 2^n$ Hermitian matrix H describing n qubit system:

$$\boxed{H = \sum_{j=1}^{m} H_j}$$

, where H_j acts only on a few of the qubits. Then, m is polynomial in system size N. For simplicity, let's assume that H acts nontrivially only on two qubits (this is called 2-local).

The unitary operator then reads

$$U(t) = e^{-it\sum_{j=1}^{m} H_j}$$

For those terms that commute, we can transform this into a product of k-local unitaries, i.e.,

If $[H_i, H_j] = 0 \ \forall j, k$, then

$$e^{-it\sum_j H_j} = \prod_j e^{-itH_j}$$

implementable unitaries $\hat{=}$ one & two qubit gates for 2-local H.

Generally terms in H do not all commute. So we group terms into different parts F_a, where terms in each group all commute. Different such groupings are possible.

This yields a decomposition

$$H = \sum_{a=1}^{d} F_a$$

where F_a are hermitian operators such that each unitary e^{-itF_a} admits an efficient implementation by a quantum circuit for any evolution time t.

For example, this is the case if F_a is a sum of few-particle interactions that pairwise commute.

Then, we divide the time evolution into r steps $\Delta t = \frac{t}{r}$ with final time t and use that

$$U(t) = e^{-iHt} = \left(e^{-iH\frac{t}{r}}\right)^r =$$

$$= \left(e^{-i\frac{t}{r}\sum_{j=1}^{m} H_j}\right)^r = \left(\prod_{j=1}^{m} e^{-i\frac{t}{r}H_j} + E\right)^r$$

with error matrix whose operator norm obeys

$$\|E\| = O\left(\frac{\max_j \|H_j\|\, m^2 t^2}{r^2}\right).$$

Note that we can group commuting terms together into F_a and then obtain $U(t) = \left(\prod_{a=1}^{d} e^{-i\frac{t}{r}F_a} + E\right)^r$.

Example. $H = \sum_j Z_j Z_{j+1} + \sum_j X_j$

$\Rightarrow$ $F_1 = \sum_{j \text{ even}} Z_j Z_{j+1}$, $F_2 = \sum_{j \text{ odd}} Z_j Z_{j+1}$, $F_3 = \sum_j X_j$

$$\Rightarrow U(t) = \prod_{m=1}^{r} e^{-i\frac{mt}{r}F_1} e^{-i\frac{mt}{r}F_2} e^{-i\frac{mt}{r}F_3} + O(t^2)$$

More accurate derivation of the error term:

Example.

$H = H_1 + H_2$, where H_j are 2-local and $[H_1, H_2] \neq 0$.

For example: $H_1 = X_1$, $H_2 = Z_1 Z_2$

$$\text{Then,}\quad U(t) = e^{-i(H_1+H_2)t} =$$

$$= \left(e^{-i\frac{t}{r}H_1} e^{-i\frac{t}{r}H_2} + E\right)^r$$

Using the Baker-Hausdorff formula

$$e^{(A+B)\Delta t} = e^{A\Delta t} e^{B\Delta t} e^{-\frac{1}{2}[A,B](\Delta t)^2} + O(\Delta t)^3,$$

we find

$$E = -\frac{1}{2}[A,B]\,(\Delta t)^2 + O(\Delta t)^3 .$$

$$\Rightarrow \|E\| = \frac{\|\overbrace{[A,B]}^{=AB-BA}\|}{2}\overbrace{(\Delta t)^2}^{=t^2/r^2} + O(\Delta t)^3 .$$

$$\leq \|A\|\,\|B\|\,(\Delta t)^2 + O(\Delta t)^3$$

Errors of a product of unitaries add up at most linearly (HW), show in HW

and thus

$$\Big\| \mathcal{U}(t) - \underbrace{\left(e^{-iH_1\frac{t}{r}}\, e^{-iH_2\frac{t}{r}}\right)^r}_{=\tilde{\mathcal{U}}(t)\ (\text{1st order Trotter approx. of } \mathcal{U})} \Big\| \leq r E$$

with total Trotter error $E_{Trot} = rE = O\left(\frac{t^2}{r}\right)$.

If we demand $E_{Trot} \leq \varepsilon \Rightarrow$ we need to choose

$$\boxed{r \geq \frac{t^2}{\varepsilon} \text{ Trotter steps.}}$$

The total number of 2 qubit gates is then

$$N_{2q} = 2r = O\left(\frac{t^2}{\varepsilon}\right).$$

This is easily generalized to m terms in the Hamiltonian, which yields

$$U(t) = \left(e^{-i\left(H_1 \frac{t}{r} + \ldots + H_m \frac{t}{r}\right)} + E \right)^r$$

with $E = m^2 \max_j \|H_j\| \frac{t^2}{r^2}$. Note that the error can be much smaller in practice if many terms in H commute with each other. This will be addressed in more detail below.

Again, errors in products add up linearly, so we find the total Trotter error to be

$$E_{\text{Trot}} = r \|E\| = O\left(\frac{m^2 \Lambda t^2}{r} \right)$$

with $\Lambda = \max_j \|H_j\|$ (can be chosen to be 1)
$\|$ largest singular value of H.

The number of 2-qubit gates for error ε is then

$$N_{2q} = m r = O\left(\frac{m^3 \Lambda t^2}{\varepsilon} \right).$$

One can use higher-order PFs to bring the dependence on t close to linear. The 2nd order PF for $H = H_1 + H_2$ reads

$$\tilde{U}(t) = \left(e^{-iH_1 \frac{\Delta t}{2}} e^{-iH_2 \Delta t} e^{-iH_1 \frac{\Delta t}{2}} \right)^r$$

The error of one individual term is then only

$$E = O\left(\frac{\Lambda^2 t^3}{r^3} \right)$$

and the total Trotter error reads

$$E_{Trot} = rE = O\left(\frac{\Lambda^2 t^3}{r^2} \right)$$

Demanding error bounded by ε requires $r = O\left(\frac{\Lambda t^{3/2}}{\varepsilon} \right)$.

The number of 2-qubit gates scales as ($m = 2$ here)

$$N_{2q} = O\left(\frac{\Lambda t^{3/2}}{\varepsilon} \right).$$

For m terms in H this becomes $r = O\left(\frac{m^2 \Lambda t^{3/2}}{\varepsilon} \right)$

and thus $N_{2q} = mr = O\left(\frac{m^3 \Lambda t^{3/2}}{\varepsilon} \right)$.

Derivations of first and second-order Trotter product formula

Let's show

$$e^{(A+B)\Delta t} = e^{A\Delta t}\, e^{B\Delta t}\, e^{-\frac{1}{2}[A,B]\Delta t^2} + O(\Delta t^3).$$

Explicit calculation:

$$LHS = I + (A+B)\Delta t + \frac{1}{2}(A^2 + AB + BA + B^2)\Delta t^2 + O(\Delta t^3)$$

$$RHS = \left(I + A\Delta t + \frac{1}{2}A^2\Delta t^2 + O(\Delta t^3)\right)\left(I + B\Delta t + \frac{1}{2}B^2\Delta t^2 + O(\Delta t^3)\right)$$

$$\cdot\left(I - \frac{1}{2}(AB - BA)\Delta t^2 + O(\Delta t^4)\right) + O(\Delta t^3) =$$

$$= I + \Delta t(A+B) + \frac{\Delta t^2}{2}\Big(A^2 + B^2 + 2AB + \underbrace{BA - AB}_{\text{from } -\frac{1}{2}[A,B]}\Big) + O(\Delta t^3)$$

$$= I + (A+B)\Delta t + (A^2 + AB + BA + B^2)\frac{\Delta t^2}{2} + O(\Delta t^3)$$

$\Rightarrow$ RHS = LHS up to terms of $O(\Delta t^3)$.

Next, it follows the <u>first-order Trotter product formula</u>

$$e^{i(A+B)\Delta t} = e^{iA\Delta t} e^{iB\Delta t} + O(\Delta t^2).$$

Directly follows as $e^{+\frac{1}{2}[A,B]\Delta t^2}$ is $I + O(\Delta t^2)$.

We can also use it to prove the <u>second-order Trotter product formula</u>:

$$e^{i(A+B)\Delta t} = e^{iA\frac{\Delta t}{2}} e^{iB\Delta t} e^{iA\frac{\Delta t}{2}} + O(\Delta t^3)$$

$$\text{LHS} = e^{iA\Delta t} e^{iB\Delta t} e^{+\frac{1}{2}[A,B]\Delta t^2} + O(\Delta t^3) =$$

$$= \left(I + iA\Delta t - \frac{1}{2}A^2\Delta t^2\right)\left(I + iB\Delta t - \frac{1}{2}B^2\Delta t^2\right)\left(I + \frac{1}{2}(AB - BA)\Delta t^2\right) + O(\Delta t^3) =$$

$$= I + \Delta t(iA + iB) - \frac{\Delta t^2}{2}\left(A^2 + B^2 + 2AB + BA - AB\right) + O(\Delta t^3) =$$

$$= I + \Delta t(iA + iB) - \frac{\Delta t^2}{2}\left(A^2 + B^2 + AB + BA\right) + O(\Delta t^3)$$

The RHS reads

$$\text{RHS} = \left(I + iA\frac{\Delta t}{2} - \frac{1}{8}A^2\Delta t^2\right)\left(I + iB\Delta t - \frac{1}{2}B^2\Delta t^2\right)$$

$$\left(I + iA\frac{\Delta t}{2} - \frac{1}{8}A^2\Delta t^2\right) + \mathcal{O}(\Delta t^3) =$$

$$= I + \Delta t\underbrace{\left(\frac{i}{2}A + iB + \frac{i}{2}A\right)}_{= iA + iB} - \frac{\Delta t^2}{2}\left(\frac{A^2}{4} + B^2 + \frac{A^2}{4}\right.$$

$$\left. + AB + \frac{A^2}{2} + BA\right) =$$

$$= A^2\underbrace{\left(\frac{1}{4} + \frac{1}{4} + \frac{1}{2}\right)}_{=1} + B^2 + AB + BA$$

$$= I + \Delta t(iA + iB) - \frac{\Delta t^2}{2}\left(A^2 + B^2 + AB + BA\right)$$

$$+ \mathcal{O}(\Delta t^3).$$

Thus, RHS = LHS and thus

$$e^{i(A+B)\Delta t} = e^{iA\frac{\Delta t}{2}}\, e^{iB\Delta t}\, e^{iA\frac{\Delta t}{2}} + \mathcal{O}(\Delta t^3).$$

Generally, there exist order-p Trotter product formulas for which

$$S(t) = \underbrace{e^{-iHt}}_{=U(t)} + O(t^{p+1})$$

Variational quantum algorithms (VQAs)

What is a variational quantum algorithm?

Key idea: represent a quantum state of interest via a parametrized quantum circuit (= parametrized unitary operator):

reference state, e.g., $|0\rangle$

$$|\psi(\vec{\theta})\rangle = U(\vec{\theta})\,|\psi_0\rangle$$

vector of real parameters $\vec{\theta} = (\theta_1, \dots, \theta_{N_\theta}) \in \mathbb{R}^{N_\theta}$

$\vec{\theta}$ is a classical representation of the quantum state $|\psi(\vec{\theta})\rangle$.

Then, determine parameters $\vec{\theta}$ by classically optimizing an objective cost function $C(\vec{\theta})$, which can be computed by preparing $|\psi(\vec{\theta})\rangle$ on a QC and measuring expectation values. Typically choose $U(\vec{\theta})$ to consist of single & two-qubit gates.

VQAs are ideally tailored to NISQ conditions as we restrict $U(\vec{\theta})$ to circuits that can be efficiently implemented on hardware.

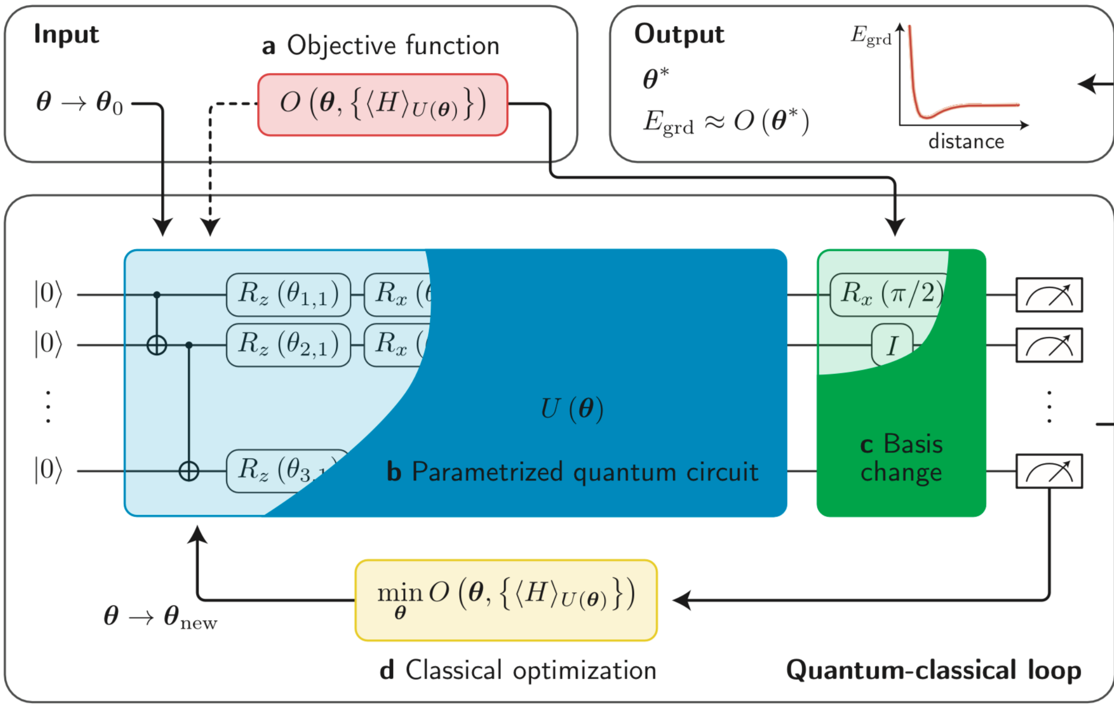

Figure 2 Diagrammatic representation of a Variational Quantum Algorithm (VQA). A VQA workflow can be divided into four main components: *a)* the objective function O that encodes the problem to be solved; *b)* the parameterized quantum circuit (PQC) U, which variables $\boldsymbol{\theta}$ are tuned to minimize the objective; *c)* the measurement scheme, which performs the basis changes and measurements needed to compute expectation values that are used to evaluate the objective; and *d)* the classical optimizer that minimizes the objective. The PQC can be defined heuristically, following hardware-inspired ansätze, or designed from the knowledge about the problem Hamiltonian H. Inputs of a VQA are the circuit ansatz $U(\boldsymbol{\theta})$ and the initial parameter values $\boldsymbol{\theta}_0$. Outputs include optimized parameter values $\boldsymbol{\theta}^*$and the minimum of the objective.

From: Bharti et al., RMP (2022)

Caveat 1: only certain states can be reached with shallow circuits,

but circuit complexity ≠ entanglement (unlike in classical TN approaches)

Caveat 2: trade circuit depth for number of measurements (often this is a bottleneck in practice).