École polytechnique fédérale de Lausanne

Master Semester Project

Laboratory of Quantum Information and Computation

Riemannian gradients, DBF, ITE and all the rest

Author:
Juan Felipe Celis Rojas

Supervisors:
Zoe Holmes
Yudai Suzuki

Introduction

Variational quantum algorithms are promising for NISQ devices. A big class of VQA consists of variational quantum eigen-solvers, i.e. VQA designed to diagonalize Hamiltonians, or just find their ground state. Some VQEs seem very distinct from one an other but many of them rely on the same principle: Riemannian geometry. We want to properly establish the relation between double-bracket, Riemannian gradient and geodesic algorithms among others. We also study the algorithm presented in [6] and we do a new error analysis on it, that proves that up to first order it has the same error than quantum time evolution.

1 Double-bracket flow

Let $\hat{H}_0$ be a Hamiltonian of dimension $n \times n$. Considering the double-bracket equation

$$\dot{H}(t) = [H(t), [H(t), N]], \quad H(0) = H_0 \tag{1}$$

where N is a symmetric real matrix. This equation is the central piece of the double-bracket quantum algorithm (DBQA), which diagonalizes H_0 and finds its eigen-vectors, in particular the ground-state.

Before going into details of DBQA, it is important to look at the big picture. Why does solving this ODE also finds the ground state of H_0? And how is it related to other methods?

1.1 Double-bracket iteration

There are many ways of approximating eq. (1). We follow the iterative method from [5], Appendix A. Define

$$\hat{H}_{k+1} = e^{s_k[\hat{D}_k, \hat{H}_k]} \hat{H}_k e^{-s_k[\hat{D}_k, \hat{H}_k]} \tag{2}$$

where $\hat{D}_1, \hat{D}_2, \dots$ is a sequence of diagonal operators. The function

$$\hat{H}_k(s) = e^{s[\hat{D}_k, \hat{H}_k]} \hat{H}_k e^{-s[\hat{D}_k, \hat{H}_k]}.$$

is a double bracket rotation because it satisfies a Heisenberg type equation

$$\partial_s \hat{H}_k(s) = [\hat{H}_k(s), [\hat{H}_k(0), \hat{D}_k]] \tag{3}$$

which is very similar to eq. (1). Then apply an optimization scheme to a cost function $c : \mathbb{C}^{n \times n} \to \mathbb{R}$ to get

$$s_k = \arg\min_{s \in \mathbb{R}} E^{(k)}(s), \tag{4}$$

where

$$E^{(k)}(s) = c(\hat{H}_k(s)).$$

For example we can define the cost function using matrix norms. Once we have our s_k we can do an iteration step eq. (2)

$$\hat{H}_{k+1} = \hat{H}_k(s_k).$$

This yields the following algorithm.

Algorithm 1 DBQA

Input: $K \geq 0, \Delta, A_0$
  $|\psi\rangle = |0\rangle$
  $U_0 \leftarrow I$
  for $k \in \{0, \ldots, K-1\}$ **do**
    $D_k \leftarrow \Delta(A_k)$
    $W_k \leftarrow [D_k, A_k]$
    $s_k \leftarrow \arg\min_s(\langle\psi_k|A_0|\psi_k\rangle))$
    $A_{k+1} \leftarrow e^{s_k W_k} A_k e^{-s_k W_k}$
    $U_{k+1} \leftarrow U_k e^{-s_k W_k}$
    $|\psi_{k+1}\rangle \leftarrow U_{k+1}|0\rangle$
  end for
return $|\psi_K\rangle$

1.1.1 Group commutator iteration

A careful reader surely noticed that in the double-bracket iteration algorithm 1 we missed a big computation. Calculating the double-bracket unitary

$$\hat{U}_k = e^{-s_k[\hat{D}_k, \hat{H}_k]}$$

requires a lot of computational resources. We use the approximations explored in the appendix B from [5]. Usually it is not an easy task to do, so we are forced to approximate. Define

$$\hat{V}^{(GC)}(\hat{A}, \hat{B}) = e^{i\hat{A}} e^{i\hat{B}} e^{-i\hat{A}} e^{-i\hat{B}}$$

and let us approximate $\hat{U}_k$ by

$$\hat{V}_k = \hat{V}^{(GC)}(\sqrt{s_k}\hat{H}_k, -\sqrt{s_k}\hat{D}_k) = \hat{U}_k + O(s_k^{3/2}). \tag{5}$$

Since we use this approximation in the rotation eq. (2) then we can simplify the group commutator to reduce the depth of the circuit. Let

$$\hat{V}^{(RGC)}(\hat{A}, \hat{B}) = e^{i\hat{B}} e^{-i\hat{A}} e^{-i\hat{B}}$$

and

$$\hat{V}_k = \hat{V}^{(RGC)}(\sqrt{s_k}\hat{H}_k, -\sqrt{s_k}\hat{D}_k).$$

Now $\hat{V}_k$ does not approximates $\hat{U}_k$ but when we replace it in eq. (2) we get the same result.

1.1.2 Third order group commutation

One can improve the error magnitude from eq. (5) by modifying $\hat{V}^{(GC)}$. For example, if we use the approximation

$$\hat{V}^{(HOPF)}(\hat{A},\hat{B}) = e^{i\phi\hat{A}}e^{i\phi\hat{B}}e^{-i\hat{A}}e^{-i(\phi+1)\hat{B}}e^{i(1-\phi)\hat{A}}e^{i\hat{B}} \tag{6}$$

with $\phi = \frac{1}{2}(\sqrt{5}-1)$ we get

$$\hat{V}_k = \hat{V}^{(HOPF)}(\sqrt{s_k}\hat{H}_k, -\sqrt{s_k}\hat{D}_k) = \hat{U}_k + O(s_k^2). \tag{7}$$

Comparing eq. (5) and eq. (7) and their rotated version we can see that the 3rd order group commutation reduces the error of approximation but it increases the depth of the circuit.

1.2 Relation with Riemannian geometry

Let Q be the diagonal matrix defined by

$$Q = Diag(eigs(H_0))$$

where the eigenvalues are ordered by decreasing order. Then, the space

$$M(Q) = \{\Theta' Q \Theta : \Theta \in O(n)\}$$

is a connected compact smooth manifold, see 2.1 Proposition 1.1 in [2]. We can see that this is the space of all real symmetric matrices with the same eigenvalues than H_0. For any $H \in M(Q)$ we can describe the tangent space $T_H M(Q)$ using the adjoint representation of $O(n)$. There is a surjection of smooth manifolds

$$Ad_\bullet(H) : O(n) \to M(Q); \Theta \mapsto Ad_\Theta(H) = \Theta H \Theta'$$

with differential

$$d(Ad_\bullet(H)) : T_{I_n}O(n) \to T_H M(Q); \Omega \mapsto d(Ad_\Omega(H)) = ad_\Omega(H) = [\Omega, H].$$

Since the differential is also surjective we can see that

$$T_H M(Q) = \{[\Omega, H] : \Omega \in o(n)\}.$$

Now that we have a described the manifold structure on $M(Q)$ and its tangent space we can define a metric. If we choose the appropriate metric, eq. (1) becomes the gradient flow of the function

$$f_N : M(Q) \to \mathbb{R}; H \to \frac{1}{2}\|N - H\|^2. \tag{8}$$

In other terms,

$$\dot{H} = \nabla f_N(H). \tag{9}$$

Moreover, solutions of eq. (1) always exist and converges to a unique equilibrium point H_∞ that commutes with N. The proof of this statements can be found in Chapter 2.1 Theorem 1.5 from [2].

Since the solution H_∞ commutes with N, we can see that

$$\dot{H}_\infty = 0. \tag{10}$$

Combining eq. (10) with eq. (9) it follows that

$$\nabla f_N(H_\infty) = 0. \tag{11}$$

Therefore, the solution obtained through DBF corresponds to a stationary point of f_N. By construction of f_N it is a critical point. Notice that at each DBI we reduce the cost function, so finding a solution of the double-bracket equation eq. (1) through DBI eq. (2) is equivalent to finding the optimum value of the cost function eq. (8).

1.3 Double-Bracket for ground state preparation

The original double-bracket algorithm finds the whole eigen-basis of the initial Hamiltonian. But if we only want the ground state how can we modify the algorithm to make it more efficient? To find the whole eigen-basis only for the ground state is an overkill.

First we can choose the cost function in eq. (4) so that each step brings us closer to the ground state. For example,

$$E^{(k)}(s) = \langle 0|\hat{H}_k(s)|0\rangle. \tag{12}$$

Together with warm-start VQE algorithms we obtain a double-bracket flow algorithm to find ground state. These have been efficient so far and details can be found in [5, 1].

Algorithm 2 VQE×DBQA

Input: $K \geq 0, \Delta, H$

$\theta^* \leftarrow VQE(H)$ ▷ so that $\theta^* = \arg\min\{\langle 0|U_\theta^\dagger H U_\theta|0\rangle\}$

$Q \leftarrow U_{\theta^*}$

$A_0 \leftarrow Q^\dagger H Q$ ▷ This gives us a warm-start for double-bracket

return $DBQA(K, \Delta, A_0)$

1.4 Riemannian Gradient Flow and Double-Bracket Flow

Now we would like to understand why Double-Bracket Flow and Riemannian Gradient Flow are essentially the same algorithm. Recall that the cost function in RGF is

$$f_{RGF}(U) = Tr[U\rho_0 U^\dagger H_0] \tag{13}$$

where H_0 is is the Hamiltonian whose ground state we want to approximate. The gradient of this cost function is

$$\nabla f_{RGF}(U) = [U\rho_0 U^\dagger, H_0]U.$$

On the other hand, the cost function in ground state preparation DBF is

$$f_{DBF}(H) = \frac{-1}{2}\|N - H\|^2. \tag{14}$$

Note that we can rewrite this cost function as

$$f_{DBF}(H) = \frac{-1}{2}\|N\|^2 + \frac{-1}{2}\|H\|^2 + Tr[HN]$$

The first observation we make is that in DBF we have to choose a matrix N and in RGF we don't have that freedom. The key to relate this two algorithms is to determine which is the right pick of N so that DBF becomes equivalent to RGF.

In the DBF setup, let us pick $N = H_0, H = U\rho_0 U^\dagger$ and replace it in the DBF cost function eq. (14):

$$f_{DBF}(H = U\rho_0 U^\dagger) = \frac{-1}{2}(\|H_0\|^2 + \|\rho_0\|^2) + Tr[U\rho_0 U^\dagger H_0]. \tag{15}$$

Notice that the term

$$\frac{-1}{2}(\|\rho_0\|^2 + \|H\|^2)$$

is constant in eq. (15). Therefore optimizing f_{RGF} is equivalent to optimize f_{DBF}.

2 Geodesic Algorithm

We can further exploit the manifold structure of $SU(N)$ after doing the $\mathbb{S}U(N)$ parametrization from [6]. Given a Riemannian metric on $SU(N)$, we can build a target unitary V from an initial unitary U following the geodesic in $SU(N)$ from U to V. We follow the steps presented in [3].

The cost function is

$$C(\phi) = 1 - \frac{1}{N}Tr[U^\dagger(\phi)V]. \tag{16}$$

Remark 1. *This is cost function is equivalent to the one in Riemannian Gradient Flow. Indeed, optimizing eq.* (16) *is the same than optimizing*

$$\frac{-1}{2}\|U(\phi) - V\|^2.$$

We claim that the Geodesic algorithm is equivalent to Riemannian Gradient Flow. Geodesics in $SU(N)$ are straight lines in the Lie algebra $\mathfrak{su}(N)$ mapped in $SU(N)$ with the exponential map. This corresponds to what RGF does.

Let us use Double-Bracket Flow techniques to rewrite the equations in the Geodesic algorithm. We see that it corresponds to the equation

$$\dot{U}(\phi) = [U(\phi), [U(\phi), V]].$$

Notice that this is not a ground state preparation. It is the general version of DBF, so the task is to find all eigenvectors of V. As mentioned in Section 1.4 we can use this algorithm for ground state preparation. A particular case of this algorithm is equivalent to RGF.

3 Here Comes The SU(N)

The key point of Here comes the SU(N) paper [6] is the $\mathbb{S}U(N)$ gate. It uses results in Lie Group theory to give clever parametrization of unitaries.

3.1 Parametrization of unitaries

To parametrize any arbitrary unitary we need a good understanding of the Lie group $SU(N)$. We already know that $SU(N)$ is connected, but what else can we say?

Proposition 1. *The Lie group $SU(N)$ is compact.*

Proof. To show compactness we show it is closed and bounded. Recall that

$$SU(N) = \{U \in M_N(\mathbb{C}) : U^\dagger U = I, det(U) = 1\}.$$

Both are closed conditions, so $SU(N)$ is closed. We also know that the operator norm of any unitary in $SU(N)$ is 1, hence it is bounded. □

Corollary 0.1. *The exponential map*

$$\exp : \mathfrak{su}(N) \to SU(N)$$

is surjective and a local diffeomorphism at the identity.

Proof. The exponential map of any Lie group is local diffeomorphism at the identity. It remains to show it is surjective. This is a consequence of the *Maximal Torus Theorem* on Lie groups. □

This gives us a nice way to describe all unitaries. For any $U \in SU(N)$ we know there is some $H \in \mathfrak{su}(N)$ such that $\exp(H) = U$.

The Lie algebra $\mathfrak{su}(N)$ is a vector space which admits as basis the set of Pauli operators on $N = 2^n$ quibits:

$$\mathcal{P}^{(N)} = \{i(\sigma_1 \otimes \cdots \otimes \sigma_n)\} \backslash \{iI\}$$

where $\sigma_j \in \{I, X, Y, Z\}$. Therefore we get the following parametrization of $SU(N)$

$$U(\theta) = \exp\left(\sum_m \theta_m G_m\right) \tag{17}$$

with $G_m \in \mathcal{P}^{(N)}$ and $\theta \in \mathbb{R}^{N^2-1}$. Let us write $A(\theta) = \sum_m \theta_m G_m$ to simplify our natation.

Remark 2. *By taking a subspace of $\mathcal{P}^{(N)}$ we can adapt our parametrization to constraints given by the hardware of the quantum computer. For example, if tour quantum computer can only perform one- and two-quibit gates then restrict to 2-local hamiltonians, i.e.*

$$\{i(\sigma_1 \otimes \cdots \otimes \sigma_n) : |\{j : \sigma_j \neq I\}| \leq 2\} \setminus \{iI\}$$

This parametrizations allow us to view unitaries U as functions

$$U : \mathbb{R}^{N^2-1} \to SU(N).$$

This yields optimization over $\mathbb{R}^n$ instead of $SU(N)$.

3.2 Gradient

To use our preferred optimization techniques on our parametrization we need to compute its gradient. The partial derivatives of $U(\theta)$ are

$$\frac{\partial}{\partial\theta_l} U(\theta) = U(\theta) \sum_{p=0}^{\infty} \frac{(-1)^p}{(p+1)!} (ad_{A(\theta)})^p G_l.$$

Here $(ad_{A(\theta)})G_l = [A(\theta), G_l]$ and for $p > 1$ we define it recursively by

$$(ad_{A(\theta)})^{p+1} G_l = [A(\theta), (ad_{A(\theta)})^p G_l].$$

The define the *effective generator* by

$$\Omega_l(\theta) = \sum_{p=0}^{\infty} \frac{(-1)^p}{(p+1)!} (ad_{A(\theta)})^p G_l \in \mathfrak{su}(N). \tag{18}$$

Observe that

$$U^\dagger(\theta)\left(\partial_l U(x)|_\theta\right) = \Omega_l(\theta).$$

Standard manipulation of the exponential function allows us to rewrite $\Omega_l(\theta)$ as

$$\Omega_l(\theta) = \left.\frac{d}{dt}\right|_{t=0} \exp(t\Omega_l(\theta))$$

and $\partial_l(U\theta)$ as

$$\frac{\partial}{\partial\theta_l} U(\theta) = U(\theta) \left.\frac{d}{dt}\right|_{t=0} \exp(t\Omega_l(\theta)).$$

For a typical cost function

$$C(\theta) = Tr[U(\theta)\rho U^{\dagger}(\theta)H]$$

where ρ is the initial state and H the Hamiltonian, we obtain

$$\frac{\partial}{\partial\theta_l}C(\theta) = \left.\frac{d}{dt}\right|_{t=0} Tr[U(\theta)e^{t\Omega_l(\theta)}\rho e^{-t\Omega_l(\theta)}U^{\dagger}(\theta)H]. \tag{19}$$

Our parametrization allowed us to replace the left circuit in by the right circuit in fig. 1.

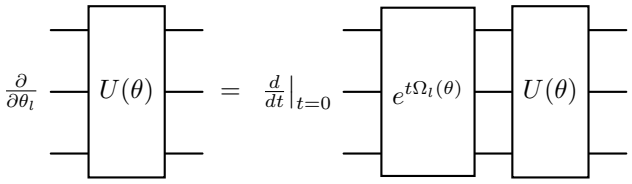

Figure 1: Improved quantum circuit

We can compute gradient of the right hand side circuit in fig. 1 with a generalized parameter shift-rule. All the above yields the *SU(N) gradients* algorithm 3.

Algorithm 3 $SU(N)$ gradients

Input: $U(x), \rho, H, \theta$

Obtain the Jacobian function:
for $l \in \{1, \ldots, N^2 - 1\}$ **do**
  $\partial_l U_l(x) \leftarrow \partial_l Re(U(x)) + i\partial_l Im(U(x))$
end for
For each gradient step:
for $l \in \{1, \ldots, N^2 - 1\}$ **do**
  $\Omega_l(\theta) \leftarrow U^{\dagger}(\theta)dU_l(x)|_{\theta}$
  $C(t) \leftarrow Tr\left[U(\theta)e^{t\Omega_l(\theta)}\rho e^{-t\Omega_l(\theta)}U^{\dagger}(\theta)H\right]$
  $\frac{\partial}{\partial\theta_l}C(\theta) \leftarrow gpsr(\Omega_l(\theta))$
end for

3.3 Connection to RGF

Both Here Comes the SU(N) and in RGF start with the same cost function

$$C(U) = Tr[U\rho U^{\dagger}H]$$

but the parametrization from eq. (17) yields a modified cost function

$$C(\theta) = Tr[U(\theta)\rho U^{\dagger}(\theta)H].$$

If we rewrite it as a cost function on the Lie algebra we get

$$C(X) = Tr[e^X \rho e^{-X} H]$$

for $U = e^X$ with $X = \sum_m \theta_m G_m \in \mathfrak{su}(N)$ and $U \in SU(N)$. The minimum of both cost functions is the same, however the gradient, see eq. (19), is different.

In Riemannian gradient descent, the gradient is on the Lie Group $SU(N)$ and it is defined by

$$\nabla C(U) = -U[\rho, U^\dagger H U] \tag{20}$$

and the corresponding gradient flow is

$$\dot{U} = -U[\rho, U^\dagger H U]. \tag{21}$$

In *Here comes the SU(N)* the gradient is on the Lie algebra $\mathfrak{su}(N)$ and is defined by

$$\nabla C(X) = \Phi_X \left([\rho, e^{-X} H e^X] \right), \tag{22}$$

where

$$\Phi_X = \sum_{n=0}^{\infty} \frac{(-1)^n}{(n+1)!} (ad_X)^n.$$

Observe that the gradients from eqs. (20) and (22) coincide only if $\Phi_X = Id$. This happens if X, H are in a Cartan subalgebra of $su(N)$, that means $[X, H] = 0$. We know that any Cartan subalgebra of $su(N)$ is of the form $\mathfrak{t} = V \mathfrak{t}_N^{(0)} V^\dagger$ where $V \in SU(N)$ and

$$\mathfrak{t}_N^{(0)} = \left\{ diag(i\theta_1, \ldots, i\theta_N) : \sum_{k=0}^{N} \theta_k = 1, \theta_k \in \mathbb{R} \forall 1 \leq k \leq N \right\}.$$

From here we recover the well known result stating that $[X, H] = 0$ if and only if X, H have a common eigen-basis. However, if this is true then e^X and H also have a common eigen-basis so we have actually found the eigen-basis of our Hamiltonian. We see that if $[X, H] = 0$ then we can terminate the algorithm.

In out initial step, there is no reason for us to have $[X, H] = 0$, in some sense this is the target of our algorithm. So at each step $\Phi_X \neq Id$ yielding a different gradient flow from RGF.

3.4 Error Analysis

In the previous section we explained why this algorithm does not follow the path on $SU(N)$ given by RGF. Now we may ask, how far is this algorithm from the correct path? To answer this question we must do some error analysis. We get inspiration from [7]. Denote by $|\psi_t^*\rangle$ the *true* state evolving under DBF/ITE, i.e.

$$|\dot{\psi}_t^*\rangle = (H - E_t^* \mathbb{I})|\psi_t^*\rangle, \tag{23}$$

where $E_t^* = \langle \psi_t^* | H | \psi_t^* \rangle$. And let $|\psi_t^\theta\rangle$ be the quantum state found by the variational algorithm.

We want to estimate (give an upper bound for) the Bures metric

$$B(|\psi^*_{t+\delta t}\rangle, |\psi^\theta_{t+\delta t}\rangle) = \sqrt{2 - 2\big|\langle \psi^\theta_{t+\delta t}|\psi^*_{t+\delta t}\rangle\big|}. \tag{24}$$

Note that using a triangle inequality we obtain that

$$\begin{aligned} B(|\psi^*_{t+\delta t}\rangle, |\psi^\theta_{t+\delta t}\rangle) \leq\ & B(|\psi^*_{t+\delta t}\rangle, (\mathbb{I} + \delta t(E^\theta_t - H))|\psi^\theta_t\rangle) \\ & + B((\mathbb{I} + \delta t(E^\theta_t - H))|\psi^\theta_t\rangle, |\psi^\theta_{t+\delta t}\rangle) \end{aligned} \tag{25}$$

First of all we can write

$$|\psi^*_{t+\delta t}\rangle = |\psi^*_t\rangle + \delta t|\dot{\psi}^*_t\rangle \tag{26}$$

and

$$|\psi^\theta_{t+\delta t}\rangle = |\psi^\theta_t\rangle + \delta t(|\dot{\psi}^\theta_t\rangle - i\dot{\nu}|\psi^\theta_t\rangle) \tag{27}$$

where ν is a phase coming from variational approximation, that is

$$|\psi^\nu_t\rangle = e^{i\nu}|\psi^\theta_t\rangle. \tag{28}$$

McLachlan's variational principle [4] yields that

$$\dot{\nu} = -\mathfrak{Im}(\langle\dot{\psi}^\theta_t|\psi^\theta_t\rangle). \tag{29}$$

Returning to the Bures distance we can see that

$$\begin{aligned} B(|\psi^*_{t+\delta t}\rangle, (\mathbb{I} + \delta t(E^\theta_t - H))|\psi^\theta_t\rangle) &= B((\mathbb{I} + \delta t(E^*_t - H))|\psi^*_t\rangle, (\mathbb{I} + \delta t(E^\theta_t - H))|\psi^\theta_t\rangle) \\ &\leq B(|\psi^*_t\rangle, |\psi^\theta_t\rangle) \end{aligned} \tag{30}$$

where the inequality holds because all physical processes are non-trace-increasing. It remains to approximate $B((\mathbb{I} + \delta t(E^\theta_t - H))|\psi^\theta_t\rangle, |\psi^\theta_{t+\delta t}\rangle)$. We have

$$\begin{aligned} B((\mathbb{I} + \delta t(E^\theta_t - H))|\psi^\theta_t\rangle, |\psi^\theta_{t+\delta t}\rangle) &= \sqrt{2 - 2\big|\langle \psi^\theta_{t+\delta t}|(\mathbb{I} + \delta t(E^\theta_t - H))|\psi^\theta_t\rangle\big|} \\ &\leq \big\||\psi^\theta_{t+\delta t}\rangle - (\mathbb{I} + \delta t(E^\theta_t - H))|\psi^\theta_t\rangle\big\|_2 \\ &= \delta t\big\||\dot{\psi}^\theta_t\rangle - i\dot{\nu}|\psi^\theta_t\rangle + (H - E^\theta_t)|\psi^\theta_t\rangle\big\|_2 \end{aligned} \tag{31}$$

Now denote

$$|e_t\rangle = |\dot{\psi}^\theta_t\rangle + (H - E^\theta_t - i\dot{\nu})|\psi^\theta_t\rangle. \tag{32}$$

Hence

$$B(|\psi^*_{t+\delta t}\rangle, |\psi^\theta_{t+\delta t}\rangle) \leq B(|\psi^*_t\rangle, |\psi^\theta_t\rangle) + \delta t\||e_t\rangle\|_2. \tag{33}$$

Our goal is to give an approximation of the error $|e_t\rangle$. The explicit computation yields

$$\begin{aligned} \||e_t\rangle\|_2^2 &= \langle\dot{\psi}^\theta_t|\dot{\psi}^\theta_t\rangle + 2\mathfrak{Re}(\langle\dot{\psi}^\theta_t|H - (E^\theta_t + i\dot{\nu})|\psi^\theta_t\rangle) + Var_\theta(H) + \dot{\nu}^2 \\ &= \langle\dot{\psi}^\theta_t|\dot{\psi}^\theta_t\rangle + 2\mathfrak{Re}(\langle\dot{\psi}^\theta_t|H|\psi^\theta_t\rangle) - 2\mathfrak{Re}((E^\theta_t + i\dot{\nu})\langle\dot{\psi}^\theta_t|\psi^\theta_t\rangle) + Var_\theta(H) + \dot{\nu}^2 \end{aligned} \tag{34}$$

because

$$\langle \psi_t^\theta | (H - E_t^\theta - i\dot{\nu})(H - E_t\theta + i\dot{\nu}) | \psi_t^\theta \rangle = Var_\theta(H) + \dot{\nu}^2. \tag{35}$$

Let us estimate the remaining terms. First we observe that

$$2\mathfrak{Re}(\langle \dot{\psi_t^\theta} | \psi_t^\theta \rangle) = \frac{\partial}{\partial t}(\langle \psi_t^\theta | \psi_t^\theta \rangle) = 0. \tag{36}$$

Then

$$\begin{aligned} \| |e_t\rangle \|_2^2 &= \langle \dot{\psi_t^\theta} | \dot{\psi_t^\theta} \rangle + 2\mathfrak{Re}(\langle \dot{\psi_t^\theta} | H | \psi_t^\theta \rangle) + 2\dot{\nu}\mathfrak{Im}(\langle \dot{\psi_t^\theta} | \psi_t^\theta \rangle) + Var_\theta(H) + \dot{\nu}^2 \\ &= \langle \dot{\psi_t^\theta} | \dot{\psi_t^\theta} \rangle + 2\mathfrak{Re}(\langle \dot{\psi_t^\theta} | H | \psi_t^\theta \rangle) + Var_\theta(H) - \dot{\nu}^2 \end{aligned} \tag{37}$$

Now we find an expression for $|\dot{\psi_t^\theta}\rangle$. Using the the $\mathbb{S}U(N)$ gradient of $U(\theta)$ from Algorithm 3 we can write

$$\begin{aligned} |\dot{\psi_t^\theta}\rangle &= (\mathbb{S}U(N))(U, |\psi_t^\theta\rangle, H, \theta)|\psi_0^\theta\rangle \\ &= \left(\sum_l \dot{\theta}_l U(\theta)\Omega_l(\theta) \right) |\psi_0^\theta\rangle \end{aligned} \tag{38}$$

Then

$$\begin{aligned} \langle \dot{\psi_t^\theta} | \dot{\psi_t^\theta} \rangle &= \sum_{k,l} \dot{\theta}_k \dot{\theta}_l \langle \psi_0 | \Omega_k^\dagger(\theta)\Omega_l(\theta) | \psi_0 \rangle \\ \langle \psi_t^\theta | \dot{\psi_t^\theta} \rangle &= \sum_l \dot{\theta}_l \langle \psi_0 | \Omega_l(\theta) | \psi_0 \rangle \\ \langle \psi_t^\theta | H | \dot{\psi_t^\theta} \rangle &= \sum_l \dot{\theta}_l \langle \psi_0 | U^\dagger(\theta) H U(\theta) \Omega_l(\theta) | \psi_0 \rangle \end{aligned} \tag{39}$$

Putting all this together we obtain

$$\begin{aligned} \| |e_2\rangle \|^2 &= Var_\theta(H) - \dot{\nu}^2 \\ &\quad + \sum_{k,l} \dot{\theta}_k \dot{\theta}_l \langle \psi_0 | \Omega_k^\dagger(\theta)\Omega_l(\theta) | \psi_0 \rangle \\ &\quad + 2 \sum_l \dot{\theta}_l \mathfrak{Re} \left[\langle \psi_0 | U^\dagger(\theta) H U(\theta) \Omega_l(\theta) | \psi_0 \rangle \right]. \end{aligned} \tag{40}$$

Recall from Equation (19) that

$$\begin{aligned} \partial_l C(\theta) &= \langle \psi_0 | U^\dagger(\theta) H U(\theta) \Omega_l(\theta) | \psi_0 \rangle - \langle \psi_0 | U(\theta) H \Omega_l(\theta) U^\dagger(\theta) | \psi_0 \rangle \\ &= 2\mathfrak{Re}[\langle \psi_0 | U^\dagger(\theta) H U(\theta) \Omega_l(\theta) | \psi_0 \rangle]. \end{aligned} \tag{41}$$

Now use the fact that $\Omega_l^\dagger = -\Omega_l$ to get

$$\overline{\langle \psi_0 | \Omega_l(\theta) | \psi_0 \rangle} = \langle \psi_0 | \Omega_l^\dagger(\theta) | \psi_0 \rangle = -\langle \psi_0 | \Omega_l(\theta) | \psi_0 \rangle. \tag{42}$$

Therefore

$$\mathfrak{Re}\left[\langle\psi_0|\Omega_l(\theta)|\psi_0\rangle\right] = 0. \tag{43}$$

Now let us compute the Fisher information matrix

$$\begin{aligned}
\mathcal{F}^Q_{i,j} &= \mathfrak{Re}\left(\frac{\partial\langle\psi_t^\theta|}{\partial\theta_i}\frac{\partial|\psi_t^\theta\rangle}{\partial\theta_j} - \frac{\partial\langle\psi_t^\theta|}{\partial\theta_i}|\psi_0\rangle\langle\psi_0|\frac{\partial|\psi_t^\theta\rangle}{\partial\theta_j}\right) \\
&= \mathfrak{Re}\left(\langle\psi_0|\Omega_i^\dagger(\theta)\Omega_j(\theta)|\psi_0\rangle - \langle\psi_0|\Omega_i^\dagger(\theta)|\psi_0\rangle\langle\psi_0|\Omega_j(\theta)|\psi_0\rangle\right) \\
&= \mathfrak{Re}\left(\langle\psi_0|\Omega_i^\dagger(\theta)\Omega_j(\theta)|\psi_0\rangle\right) - \langle\psi_0|\Omega_i^\dagger(\theta)|\psi_0\rangle\langle\psi_0|\Omega_j(\theta)|\psi_0\rangle
\end{aligned} \tag{44}$$

Plug Equations (41), (43) and (44) into Equation (40) to obtain

$$\begin{aligned}
\||e_2\rangle\|^2 = Var_\theta(H) - \dot{\nu}^2 + \sum_l \dot{\theta}_l\partial_l C(\theta) + \sum_{k,l}\dot{\theta}_k\dot{\theta}_l\mathcal{F}^Q_{k,l} \\
+ \sum_{k,l}\dot{\theta}_k\dot{\theta}_l\langle\psi_0|\Omega_k^\dagger(\theta)|\psi_0\rangle\langle\psi_0|\Omega_l(\theta)|\psi_0\rangle.
\end{aligned} \tag{45}$$

To finish this computation notice that

$$\dot{\nu}^2 = \sum_{k,l}\dot{\theta}_k\dot{\theta}_l\langle\psi_0|\Omega_k^\dagger(\theta)|\psi_0\rangle\langle\psi_0|\Omega_l(\theta)|\psi_0\rangle \tag{46}$$

then we get

$$\||e_2\rangle\|^2 = Var_\theta(H) + \sum_l \dot{\theta}_l\partial_l C(\theta) + \sum_{k,l}\dot{\theta}_k\dot{\theta}_l\mathcal{F}^Q_{k,l}. \tag{47}$$

Since $|\psi_0^\theta\rangle = |\psi_0\rangle = |\psi_0^*\rangle$ we can do the following computation:

$$B(|\psi_T^\theta\rangle, |\psi_T^*\rangle) \leq \int_0^T \||e_t\rangle\|_2 dt. \tag{48}$$

Remark 3. *Remember that in our algorithm we don't really use any $\Omega_l(\theta)$ gate. Instead of this we use a parameter shift rule to compute*

$$\left.\frac{d}{dt}\right|_{t=0} e^{t\Omega_l(\theta)}. \tag{49}$$

References

[1] Marek Gluza. "Double-bracket quantum algorithms for diagonalization". In: *Quantum* 8 (Apr. 2024), p. 1316. ISSN: 2521-327X. DOI: 10.22331/q-2024-04-09-1316. URL: http://dx.doi.org/10.22331/q-2024-04-09-1316.

[2] Uwe Helmke and John B. Moore. *Optimization and dynamical systems*. Communications and Control Engineering Series. With a foreword by R. Brockett. Springer-Verlag London, Ltd., London, 1994, pp. xiv+391. ISBN: 0-387-19857-1. DOI: 10.1007/978-1-4471-3467-1. URL: https://doi.org/10.1007/978-1-4471-3467-1.

[3] Dylan Lewis et al. *Geodesic Algorithm for Unitary Gate Design with Time-Independent Hamiltonians*. 2024. arXiv: 2401.05973 [quant-ph]. URL: https://arxiv.org/abs/2401.05973.

[4] Andrew D McLachlan. "A variational solution of the time-dependent Schrodinger equation". In: *Molecular Physics* 8.1 (1964), pp. 39–44.

[5] Matteo Robbiati et al. *Double-bracket quantum algorithms for high-fidelity ground state preparation*. 2024. arXiv: 2408.03987 [quant-ph]. URL: https://arxiv.org/abs/2408.03987.

[6] Roeland Wiersema et al. "Here comes the SU(N): multivariate quantum gates and gradients". In: *Quantum* 8 (Mar. 2024), p. 1275. ISSN: 2521-327X. DOI: 10.22331/q-2024-03-07-1275. URL: http://dx.doi.org/10.22331/q-2024-03-07-1275.

[7] Christa Zoufal, David Sutter, and Stefan Woerner. "Error bounds for variational quantum time evolution". In: *Physical Review Applied* 20.4 (Oct. 2023). ISSN: 2331-7019. DOI: 10.1103/physrevapplied.20.044059. URL: http://dx.doi.org/10.1103/PhysRevApplied.20.044059.