

Kagome VQE – Notes

1 Hardware layout

1.1 Mapping Kagome to Guadalupe layout

We can map the Kagome vertices and edges to the qubits of the Guadalupe device (cf. figure 1) as shown in figure 2.

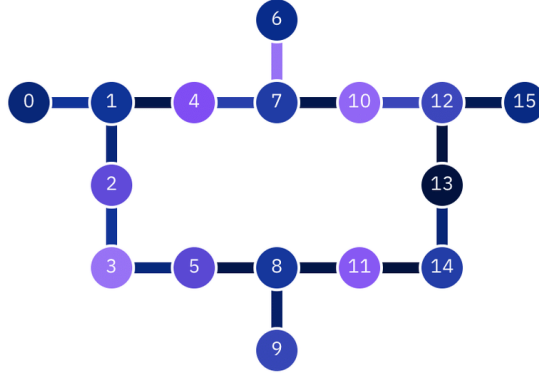


Figure 1: Layout of the Guadalupe device

1.2 IBM hardware gates

Note that IBM hardware implements R_Z , \sqrt{X} , X and CNOT gates only. We can decompose:

$$\begin{aligned} H &= e^{i\pi/4} R_Z(\pi/2) \sqrt{X} R_Z(\pi/2) \\ S &= e^{i\pi/4} R_Z(\pi/2) \\ S^\dagger &= e^{-i\pi/4} R_Z(-\pi/2) \\ SH &= e^{i3\pi/2} R_Z(-\pi) \sqrt{X} R_Z(\pi/2) \\ HS &= e^{i3\pi/2} R_Z(\pi/2) \sqrt{X} R_Z(-\pi) \\ HS^\dagger &= R_Z(\pi/2) \sqrt{X} \\ S^\dagger H &= \sqrt{X} R_Z(\pi/2) \end{aligned} \tag{1}$$

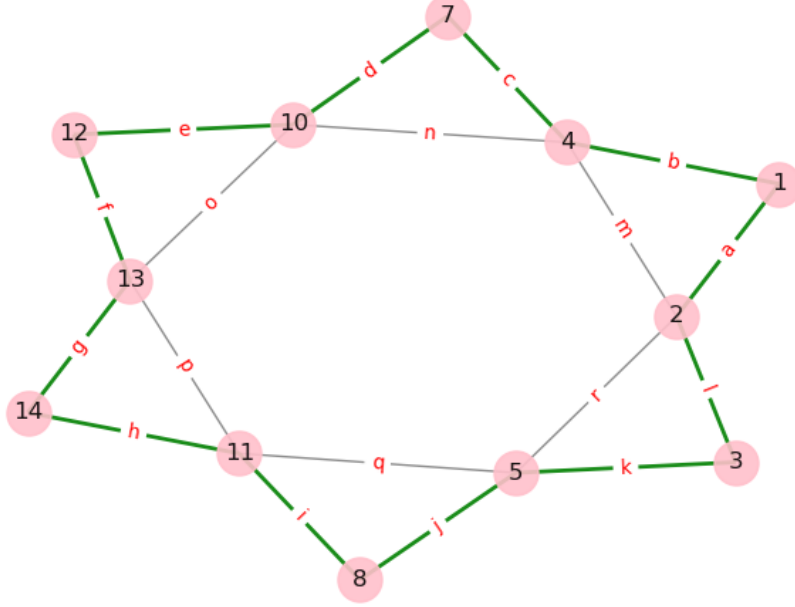


Figure 2: Kagome unit cell. Numbers correspond to the qubits of the Guadalupe layout. Green edges correspond to physically connected qubits, whereas grey edges have no physical link and need to be achieved with swap gates.

2 Unitaries

Labeling edges with letters $a..r$ as shown in figure 2, we have the edge operators:

$$\begin{aligned}
 U_X^a(t) &= \exp(-itX_1X_2) = \begin{array}{c} 1 \text{ --- } \boxed{H} \text{ --- } \bullet \text{ --- } \bullet \text{ --- } \boxed{H} \text{ ---} \\ 2 \text{ --- } \boxed{H} \oplus \boxed{R_Z(2t)} \oplus \boxed{H} \text{ ---} \end{array} \\
 U_Y^a(t) &= \exp(-itY_1Y_2) = \begin{array}{c} 1 \text{ --- } \boxed{S} \boxed{H} \text{ --- } \bullet \text{ --- } \bullet \text{ --- } \boxed{H} \boxed{S^\dagger} \text{ ---} \\ 2 \text{ --- } \boxed{S} \boxed{H} \oplus \boxed{R_Z(2t)} \oplus \boxed{H} \boxed{S^\dagger} \text{ ---} \end{array} \quad (2) \\
 U_Z^a(t) &= \exp(-itZ_1Z_2) = \begin{array}{c} 1 \text{ --- } \bullet \text{ --- } \bullet \text{ ---} \\ 2 \text{ --- } \oplus \boxed{R_Z(2t)} \oplus \text{ ---} \end{array}
 \end{aligned}$$

and correspondingly $U_{X,Y,Z}^b(t)$ through $U_{X,Y,Z}^r(t)$ acting on the two qubit vertices given by the corresponding edge $b..r$.

Note that every single $U_\mu^\alpha(t)$ commutes with the total spins $S_x = \sum_{i=0}^{11} X_i$, $S_y = \sum_{i=0}^{11} Y_i$, $S_z = \sum_{i=0}^{11} Z_i$, and therefore with $S^2 = S_x^2 + S_y^2 + S_z^2$.

We can combine these unitaries to a Heisenberg gate [1]:

$$U_H^a(\chi, \psi, \omega) = U_Z^a(\omega)U_Y^a(\psi)U_X^a(\chi) \quad (3)$$

$U_{X,Y,Z}^a(t)$ through $U_{X,Y,Z}^l(t)$ can be directly implemented because the edges $a..l$ correspond to physical links. The operators on the edges $m..r$ must be implemented using SWAP gates:

$$U_X^m(t) = \begin{array}{c} 4 \\ 1 \\ 2 \end{array} \begin{array}{c} \text{---} \\ \bullet \oplus \bullet \\ \oplus \bullet \oplus \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ \bullet \oplus \bullet \\ \oplus \bullet \oplus \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \quad (4)$$

and correspondingly for all $U_{X,Y,Z}^m(t)$ through $U_{X,Y,Z}^r(t)$.

We can group these 54 operators into:

$$\begin{aligned} U_X^1 &= U_X^a(\chi_a)U_X^c(\chi_c)U_X^e(\chi_e)U_X^g(\chi_g)U_X^i(\chi_i)U_X^k(\chi_k) \\ U_Y^1 &= U_Y^a(\psi_a)U_Y^c(\psi_c)U_Y^e(\psi_e)U_Y^g(\psi_g)U_Y^i(\psi_i)U_Y^k(\psi_k) \\ U_Z^1 &= U_Z^a(\omega_a)U_Z^c(\omega_c)U_Z^e(\omega_e)U_Z^g(\omega_g)U_Z^i(\omega_i)U_Z^k(\omega_k) \\ U_X^2 &= U_X^b(\chi_b)U_X^d(\chi_d)U_X^f(\chi_f)U_X^h(\chi_h)U_X^j(\chi_j)U_X^l(\chi_l) \\ U_Y^2 &= U_Y^b(\psi_b)U_Y^d(\psi_d)U_Y^f(\psi_f)U_Y^h(\psi_h)U_Y^j(\psi_j)U_Y^l(\psi_l) \\ U_Z^2 &= U_Z^b(\omega_b)U_Z^d(\omega_d)U_Z^f(\omega_f)U_Z^h(\omega_h)U_Z^j(\omega_j)U_Z^l(\omega_l) \\ U_X^3 &= U_X^m(\chi_m)U_X^o(\chi_o)U_X^q(\chi_q) \\ U_Y^3 &= U_Y^m(\psi_m)U_Y^o(\psi_o)U_Y^q(\psi_q) \\ U_Z^3 &= U_Z^m(\omega_m)U_Z^o(\omega_o)U_Z^q(\omega_q) \\ U_X^4 &= U_X^n(\chi_n)U_X^p(\chi_p)U_X^r(\chi_r) \\ U_Y^4 &= U_Y^n(\psi_n)U_Y^p(\psi_p)U_Y^r(\psi_r) \\ U_Z^4 &= U_Z^n(\omega_n)U_Z^p(\omega_p)U_Z^r(\omega_r) \end{aligned} \quad (5)$$

where each $U_{X,Y,Z}^i$ contains only mutually commuting terms.

We can group these further to

$$\begin{aligned} U_X(\vec{\chi}) &= U_X^1 U_X^2 U_X^3 U_X^4 \\ U_Y(\vec{\psi}) &= U_Y^1 U_Y^2 U_Y^3 U_Y^4 \\ U_Z(\vec{\omega}) &= U_Z^1 U_Z^2 U_Z^3 U_Z^4 \end{aligned} \quad (6)$$

each of which contains mutually commuting terms. $\vec{\chi}$, $\vec{\psi}$, and $\vec{\omega}$ denote vectors of the 3×18 parameters $\chi_{a..r}$, $\psi_{a..r}$, and $\omega_{a..r}$, respectively. With the

representation in the hardware native gates (1), these circuits have depths

$$\begin{aligned}\text{depth}(U_X) &= 4 \times 9 + 12 = 48 \\ \text{depth}(U_Y) &= 4 \times 8 + 12 = 44 \\ \text{depth}(U_Z) &= 4 \times 3 + 12 = 24,\end{aligned}\tag{7}$$

amounting to a total depth of 116 per layer.

Using, instead, the Heisenberg gate (3) for a grouping into four groups $U_H^i = U_Z^i U_Y^i U_X^i$ as given in equation (5) results in flatter circuits

$$U_H^1(\vec{\theta}_1) = \begin{array}{c} \begin{array}{c} 2 \\ 1 \end{array} \begin{array}{|c|} \hline U_H(\chi_a, \psi_a, \omega_a) \\ \hline \end{array} \begin{array}{c} \\ \end{array} \\ \begin{array}{c} 4 \\ 7 \end{array} \begin{array}{|c|} \hline U_H(\chi_c, \psi_c, \omega_c) \\ \hline \end{array} \begin{array}{c} \\ \end{array} \\ \begin{array}{c} 10 \\ 12 \end{array} \begin{array}{|c|} \hline U_H(\chi_e, \psi_e, \omega_e) \\ \hline \end{array} \begin{array}{c} \\ \end{array} \\ \begin{array}{c} 13 \\ 14 \end{array} \begin{array}{|c|} \hline U_H(\chi_g, \psi_g, \omega_g) \\ \hline \end{array} \begin{array}{c} \\ \end{array} \\ \begin{array}{c} 11 \\ 8 \end{array} \begin{array}{|c|} \hline U_H(\chi_i, \psi_i, \omega_i) \\ \hline \end{array} \begin{array}{c} \\ \end{array} \\ \begin{array}{c} 5 \\ 3 \end{array} \begin{array}{|c|} \hline U_H(\chi_k, \psi_k, \omega_k) \\ \hline \end{array} \begin{array}{c} \\ \end{array} \end{array}, \tag{8}$$

where $\vec{\theta}_1$ is a vector of the 18 parameters entering in the definition of U_H^1 , and correspondingly for $U_H^2(\vec{\theta}_2)$ with all 2-qubit gates shifted by one position. The inner edges result in

$$U_H^3(\vec{\theta}_3) = \begin{array}{c} \begin{array}{c} 2 \\ 1 \end{array} \begin{array}{|c|} \hline \begin{array}{c} \bullet \oplus \bullet \\ \oplus \bullet \oplus \end{array} \\ \hline \end{array} \begin{array}{|c|} \hline U_H(\chi_m, \psi_m, \omega_m) \\ \hline \end{array} \begin{array}{|c|} \hline \begin{array}{c} \bullet \oplus \bullet \\ \oplus \bullet \oplus \end{array} \\ \hline \end{array} \\ \begin{array}{c} 4 \\ 7 \end{array} \begin{array}{|c|} \hline \\ \hline \end{array} \\ \begin{array}{c} 10 \\ 12 \end{array} \begin{array}{|c|} \hline \begin{array}{c} \bullet \oplus \bullet \\ \oplus \bullet \oplus \end{array} \\ \hline \end{array} \begin{array}{|c|} \hline U_H(\chi_o, \psi_o, \omega_o) \\ \hline \end{array} \begin{array}{|c|} \hline \begin{array}{c} \bullet \oplus \bullet \\ \oplus \bullet \oplus \end{array} \\ \hline \end{array} \\ \begin{array}{c} 13 \\ 14 \end{array} \begin{array}{|c|} \hline \\ \hline \end{array} \\ \begin{array}{c} 11 \\ 8 \end{array} \begin{array}{|c|} \hline \begin{array}{c} \bullet \oplus \bullet \\ \oplus \bullet \oplus \end{array} \\ \hline \end{array} \begin{array}{|c|} \hline U_H(\chi_q, \psi_q, \omega_q) \\ \hline \end{array} \begin{array}{|c|} \hline \begin{array}{c} \bullet \oplus \bullet \\ \oplus \bullet \oplus \end{array} \\ \hline \end{array} \\ \begin{array}{c} 5 \\ 3 \end{array} \begin{array}{|c|} \hline \\ \hline \end{array} \end{array} \tag{9}$$

and correspondingly for $U_H^4(\vec{\theta}_4)$ with everything shifted by two qubits.

U_H and thus U_H^1 and U_H^2 have a depth of 15, U_H^3 and U_H^4 consequentially have depths of 21, resulting in a total depth of 72 per layer.

3 Hamiltonian

The Hamiltonian can be split correspondingly in $54 = 3 \times (6 + 6 + 3 + 3)$ terms:

$$H = \sum_i H_i^X + H_i^Y + H_i^Z \quad \text{with} \quad H_i^\sigma = \sum_{\langle j,k \rangle \in \Gamma_i} \sigma^{(j)} \sigma^{(k)} \quad (10)$$

with $\sigma^{(j)} \in \{X_j, Y_j, Z_j\}$ the respective Pauli operator on the j -th qubit and with edge sets

$$\begin{aligned} \Gamma_1 &= \{a, c, e, g, i, k\} \\ \Gamma_2 &= \{b, d, f, h, j, l\} \\ \Gamma_3 &= \{m, o, q\} \\ \Gamma_4 &= \{n, p, r\}, \end{aligned} \quad (11)$$

where we denote with $a = \langle 1, 2 \rangle$ the vertices at the edge a etc. We can group it into either the three groups

$$H^X = \sum_i H_i^X \quad H^Y = \sum_i H_i^Y \quad H^Z = \sum_i H_i^Z \quad (12)$$

or the four groups

$$H_i = H_i^X + H_i^Y + H_i^Z \quad (i = 1, \dots, 4) \quad (13)$$

of mutually commuting terms. A trotterization of H is then given by

$$\begin{aligned} e^{-ip\varepsilon H} &= \prod_{j=1}^p e^{-i\varepsilon H^X} e^{-i\varepsilon H^Y} e^{-i\varepsilon H^Z} \\ &= \left(U_X(\varepsilon \vec{1}) U_Y(\varepsilon \vec{1}) U_Z(\varepsilon \vec{1}) \right)^p \end{aligned} \quad (14)$$

or

$$\begin{aligned} e^{-ip\varepsilon H} &= \prod_{j=1}^p e^{-i\varepsilon H_1} e^{-i\varepsilon H_2} e^{-i\varepsilon H_3} e^{-i\varepsilon H_4} \\ &= \left(U_H^1(\varepsilon \vec{1}) U_H^2(\varepsilon \vec{1}) U_H^3(\varepsilon \vec{1}) U_H^4(\varepsilon \vec{1}) \right)^p \end{aligned} \quad (15)$$

where $\vec{1}$ is a vector of the matching dimension with all entries 1, i.e. all parameters take the same value ε .

3.1 Dimer ground state

The Hamiltonian H_1 is a separable sum of 2-qubit Hamiltonians:

$$H_1 = \sum_{\langle i,j \rangle \in \{a,c,e,g,i,k\}} (X_i X_j + Y_i Y_j + Z_i Z_j) \quad (16)$$

Each such 2-qubit term

$$XX + YY + ZZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (17)$$

(where the matrix is in the computational basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$) has the ground state with non-degenerate eigenvalue $h_1 = -3$ and eigenvector

$$|\psi_D^{(i,j)}\rangle = \frac{1}{\sqrt{2}} (|01\rangle - |10\rangle) = \frac{1}{\sqrt{2}} (|2^i\rangle - |2^j\rangle) \quad (18)$$

where i, j are the qubit labels. Hence, the ground state of H_1 is the product state (with $|0\rangle$ on all remaining labels):

$$|\Psi_D\rangle = \bigotimes_{\langle i,j \rangle \in \{a,c,e,g,i,k\}} |\psi_D^{(i,j)}\rangle \quad (19)$$

This state can be prepared [2] via the circuit

$$U_D = \prod_{\langle i,j \rangle \in \{a,c,e,g,i,k\}} U_D^{(i,j)} = \prod_{\langle i,j \rangle \in \{a,c,e,g,i,k\}} \begin{array}{c} i \text{ --- } \oplus \text{ --- } \boxed{Z} \\ j \text{ --- } \boxed{H} \text{ --- } \bullet \text{ --- } \boxed{X} \end{array} \quad (20)$$

3.2 Trimer ground state

We can split the Hamiltonian in

$$H = H_A + H_B = \sum_{i=1}^3 H_A^i + \sum_{i=1}^3 H_B^i \quad (21a)$$

with

$$H_A^1 = \sum_{\langle j,k \rangle \in \{a,b,m\}} (X_j X_k + Y_j Y_k + Z_j Z_k) \quad (21b)$$

$$H_A^2 = \sum_{\langle j,k \rangle \in \{e,f,o\}} (X_j X_k + Y_j Y_k + Z_j Z_k) \quad (21c)$$

$$H_A^3 = \sum_{\langle j,k \rangle \in \{i,j,q\}} (X_j X_k + Y_j Y_k + Z_j Z_k) \quad (21d)$$

$$H_B^1 = \sum_{\langle j,k \rangle \in \{c,d,n\}} (X_j X_k + Y_j Y_k + Z_j Z_k) \quad (21e)$$

$$H_B^2 = \sum_{\langle j,k \rangle \in \{g,h,p\}} (X_j X_k + Y_j Y_k + Z_j Z_k) \quad (21f)$$

$$H_B^3 = \sum_{\langle j,k \rangle \in \{k,l,r\}} (X_j X_k + Y_j Y_k + Z_j Z_k) \quad (21g)$$

These are six trimer Hamiltonians

$$\begin{aligned} & XXI + XIX + IXX + YYI + YIY + IYY + ZZI + ZIZ + IZZ \\ &= \begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 2 & 0 & 0 & 0 \\ 0 & 2 & -1 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & 2 & 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix} \quad (22) \end{aligned}$$

which have the minimal eigenvalue -3 with the four linearly independent eigenvectors

$$|\psi_{T1}\rangle = \frac{1}{\sqrt{2}} (|011\rangle - |101\rangle) \quad (23a)$$

$$|\psi_{T2}\rangle = \frac{1}{\sqrt{2}} (|001\rangle - |010\rangle) \quad (23b)$$

$$|\psi_{T3}\rangle = \frac{1}{\sqrt{2}} (|011\rangle - |110\rangle) \quad (23c)$$

$$|\psi_{T4}\rangle = \frac{1}{\sqrt{2}} (|001\rangle - |100\rangle) \quad (23d)$$

which can be prepared with the circuits

$$U_{T1}^{(i,j,k)} = \begin{array}{c} i \text{---} \boxed{H} \text{---} \bullet \text{---} \boxed{Z} \text{---} \\ j \text{---} \oplus \text{---} \boxed{X} \text{---} \\ k \text{---} \boxed{X} \text{---} \end{array} \quad (24a)$$

$$U_{T2}^{(i,j,k)} = \begin{array}{c} i \text{---} \\ j \text{---} \boxed{H} \text{---} \bullet \text{---} \boxed{Z} \text{---} \\ k \text{---} \oplus \text{---} \boxed{X} \text{---} \end{array} \quad (24b)$$

$$U_{T3}^{(i,j,k)} = \begin{array}{c} i \text{---} \boxed{H} \text{---} \bullet \text{---} \boxed{Z} \text{---} \\ j \text{---} \text{---} \oplus \text{---} \boxed{X} \text{---} \\ k \text{---} \oplus \text{---} \boxed{X} \text{---} \end{array} \quad (24c)$$

$$U_{T4}^{(i,j,k)} = \begin{array}{c} i \text{---} \boxed{H} \text{---} \bullet \text{---} \boxed{Z} \text{---} \\ j \text{---} \text{---} \oplus \text{---} \boxed{X} \text{---} \\ k \text{---} \oplus \text{---} \boxed{X} \text{---} \end{array} \quad (24d)$$

As before, the full (12+4)-qubit ground state of H_A is then obtained as the product

$$|\Psi_{T\alpha}\rangle = U_{T\alpha} |0\rangle^{\otimes 16} = U_{T\alpha}^{(2,1,4)} U_{T\alpha}^{(10,12,13)} U_{T\alpha}^{(11,8,5)} |0\rangle^{\otimes 16} \quad (25)$$

for $\alpha \in \{1, 2, 3, 4\}$. Note that, due to the gate-connectivity of the Guadalupe device, U_{T1} and U_{T2} can be implemented straightforwardly, whereas U_{T3} and U_{T4} require an additional six CNOTs for swapping.

The evolution with the Hamiltonians H_A and H_B (encoded by the purple labels in (26)) is achieved with the unitaries

$$U_{A,B}(\vec{\theta}_{A,B}) = \begin{array}{c} \begin{array}{l} 2, 4 \\ 1, 7 \\ 4, 10 \\ 7, 12 \end{array} \begin{array}{c} \boxed{U_H} \\ \boxed{U_H} \\ \boxed{U_H} \end{array} \begin{array}{c} \boxed{U_H} \\ \boxed{U_H} \\ \boxed{U_H} \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \\ \begin{array}{l} 10, 13 \\ 12, 14 \\ 13, 11 \end{array} \begin{array}{c} \boxed{U_H} \\ \boxed{U_H} \\ \boxed{U_H} \end{array} \begin{array}{c} \boxed{U_H} \\ \boxed{U_H} \\ \boxed{U_H} \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \\ \begin{array}{l} 14, 8 \\ 11, 5 \\ 8, 3 \\ 5, 2 \\ 3, 1 \end{array} \begin{array}{c} \boxed{U_H} \\ \boxed{U_H} \\ \boxed{U_H} \end{array} \begin{array}{c} \boxed{U_H} \\ \boxed{U_H} \\ \boxed{U_H} \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \end{array} \quad (26)$$

which have depth $3 \times 15 + 2 \times 3 = 51$ and contain 72 CNOTs each.

3.3 Double-dimer ground state

Alternatively, we can split the Hamiltonian in three “double-dimer” terms, as depicted in figure 3:

$$H = H_E + H_F + H_G = \sum_{i=1}^3 H_E^i + \sum_{i=1}^3 H_F^i + \sum_{i=1}^3 H_G^i \quad (27a)$$

with

$$H_E^1 = \sum_{\langle j,k \rangle \in \{b,c\}} (X_j X_k + Y_j Y_k + Z_j Z_k) \quad (27b)$$

$$H_E^2 = \sum_{\langle j,k \rangle \in \{f,g\}} (X_j X_k + Y_j Y_k + Z_j Z_k) \quad (27c)$$

$$H_E^3 = \sum_{\langle j,k \rangle \in \{j,k\}} (X_j X_k + Y_j Y_k + Z_j Z_k) \quad (27d)$$

$$H_F^1 = \sum_{\langle j,k \rangle \in \{a,m\}} (X_j X_k + Y_j Y_k + Z_j Z_k) \quad (27e)$$

$$H_F^2 = \sum_{\langle j,k \rangle \in \{e,o\}} (X_j X_k + Y_j Y_k + Z_j Z_k) \quad (27f)$$

$$H_F^3 = \sum_{\langle j,k \rangle \in \{i,q\}} (X_j X_k + Y_j Y_k + Z_j Z_k) \quad (27g)$$

$$H_G^1 = \sum_{\langle j,k \rangle \in \{d,n\}} (X_j X_k + Y_j Y_k + Z_j Z_k) \quad (27h)$$

$$H_G^2 = \sum_{\langle j,k \rangle \in \{h,p\}} (X_j X_k + Y_j Y_k + Z_j Z_k) \quad (27i)$$

$$H_G^3 = \sum_{\langle j,k \rangle \in \{l,r\}} (X_j X_k + Y_j Y_k + Z_j Z_k) \quad (27j)$$

These are nine double-dimer Hamiltonians

$$XXI + IXX + YYI + IYY + ZZI + IZZ$$

$$= \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & -2 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & -2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix} \quad (28)$$

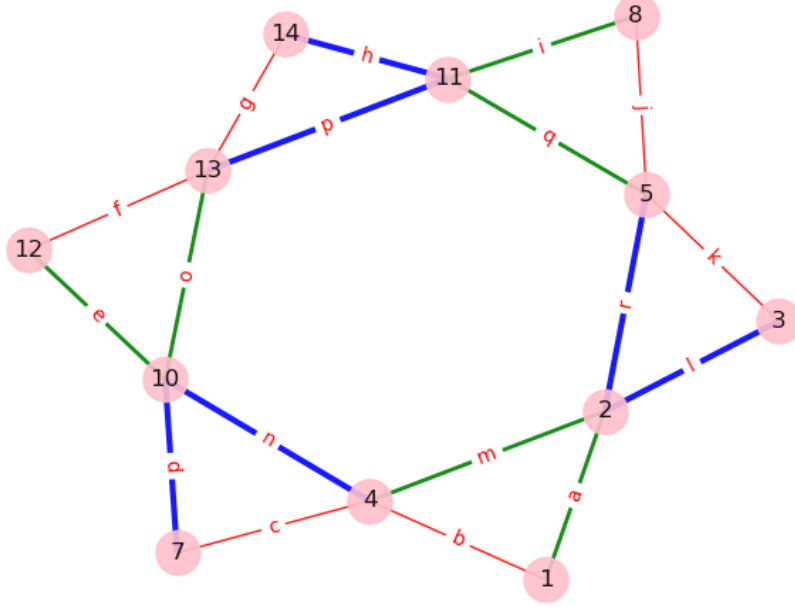


Figure 3: Kagome unit cell with edge colors showing the splitting into three double-dimer configurations.

which have the minimal eigenvalue -4 with the two linearly independent eigenvectors

$$|\psi_{DD1}\rangle = \frac{1}{\sqrt{6}} (2|101\rangle - |011\rangle - |110\rangle) \quad (29a)$$

$$|\psi_{DD2}\rangle = \frac{1}{\sqrt{6}} (2|010\rangle - |100\rangle - |001\rangle) . \quad (29b)$$

The former eigenvector can be prepared with the circuit

$$U_{DD1}^{(i,j,k)} = \begin{array}{c} i \text{ --- } \boxed{\text{H}} \text{ --- } \boxed{\text{X}} \text{ --- } \bullet \\ j \text{ --- } \boxed{\text{R}_Y(\arccos \frac{1}{3})} \text{ --- } \boxed{\text{Z}} \text{ --- } \bullet \\ k \text{ --- } \boxed{\text{X}} \text{ --- } \bullet \end{array} \quad (30)$$

a transpiled version of which, in terms of hardware gates, is given in appendix B. The state $|\psi_{DD2}\rangle$ is obtained by attaching an X to each qubit:

$$U_{DD2}^{(i,j,k)} = \begin{array}{c} i \\ j \\ k \end{array} \text{ --- } \boxed{U_{DD1}^{(i,j,k)}} \text{ --- } \begin{array}{c} \boxed{\text{X}} \\ \boxed{\text{X}} \\ \boxed{\text{X}} \end{array} \quad (31)$$

Moreover, we can create a superposition by simply attaching R_Y gates:

$$\cos\left(\frac{\alpha}{2}\right) |\psi_{DD1}\rangle + \sin\left(\frac{\alpha}{2}\right) |\psi_{DD2}\rangle \simeq U_{DD3}^{(i,j,k)} = \begin{array}{c} i \\ j \\ k \end{array} \begin{array}{|c|} \hline U_{DD1}^{(i,j,k)} \\ \hline \end{array} \begin{array}{l} R_Y(\alpha) \\ R_Y(\alpha) \\ R_Y(\alpha) \end{array} \quad (32)$$

As for the trimer state above, the full (12+4)-qubit ground state of H_E is then obtained as the product ($\alpha \in \{1, 2, 3\}$)

$$|\Psi_{DD\alpha}\rangle = U_{DD\alpha} |0\rangle^{\otimes 16} = U_{DD\alpha}^{(1,4,7)} U_{DD\alpha}^{(3,5,8)} U_{DD\alpha}^{(12,13,14)} |0\rangle^{\otimes 16} \quad (33)$$

with the unitaries for the evolution with $H_{E,F,G}$ given by

$$U_E(\vec{\theta}_E) = \begin{array}{c} 2 \\ 1 \\ 4 \\ 7 \\ 10 \\ 12 \\ 13 \\ 14 \\ 11 \\ 8 \\ 5 \\ 3 \end{array} \begin{array}{|c|} \hline U_H \\ \hline \end{array} \begin{array}{|c|} \hline U_H \\ \hline \end{array} \quad (34a)$$

$$U_F(\vec{\theta}_F) = \begin{array}{c} 2 \\ 1 \\ 4 \\ 7 \\ 10 \\ 12 \\ 13 \\ 14 \\ 11 \\ 8 \\ 5 \\ 3 \end{array} \begin{array}{|c|} \hline U_H \\ \hline \end{array} \begin{array}{|c|} \hline U_H \\ \hline \end{array} \quad (34b)$$

$$U_G(\vec{\theta}_G) = \begin{array}{c} \begin{array}{c} 5 \\ 3 \\ 2 \end{array} \begin{array}{c} \text{---} \times \text{---} \\ \text{---} \times \text{---} \\ \text{---} \times \text{---} \end{array} \\ \begin{array}{c} 1 \\ 4 \\ 7 \\ 10 \end{array} \begin{array}{c} \text{---} \times \text{---} \\ \text{---} \times \text{---} \\ \text{---} \times \text{---} \\ \text{---} \times \text{---} \end{array} \\ \begin{array}{c} 12 \\ 13 \\ 14 \\ 11 \end{array} \begin{array}{c} \text{---} \times \text{---} \\ \text{---} \times \text{---} \\ \text{---} \times \text{---} \\ \text{---} \times \text{---} \end{array} \\ 8 \end{array} \quad (34c)$$

4 Hamiltonian Variational Ansatz (HVA)

Knowing the ground state $|\Psi_1\rangle$ of H_1 we introduce

$$H(t) = H_1 + t(H_2 + H_3 + H_4) = (1 - t)H_1 + tH \quad (35)$$

The adiabatic theorem states that if we evolve H from $H(0) = H_1$ to $H(1) = H$ slowly, the system state remains in the ground state. The unitary evolution with $H(t)$ can be approximated by \tilde{p} steps of times t_i such that $\sum_i t_i = 1$ with the Hamiltonian $H(T_i)$ at the middle time point

$$T_i = \sum_{j=1}^{i-1} t_j + \frac{t_i}{2} \quad (36)$$

Hence, starting from the dimer ground state in subsection 3.1, the circuit

$$\begin{aligned} U_{\text{HVA}} &= \left(\prod_{j=1}^{\tilde{p}} e^{-it_j H(T_j)} \right) U_D \\ &\approx \left(\prod_{j=1}^{\tilde{p}} \prod_{k=1}^q e^{-i\frac{t_j}{q} H_1} e^{-i\frac{t_j}{q} T_j H_2} e^{-i\frac{t_j}{q} T_j H_3} e^{-i\frac{t_j}{q} T_j H_4} \right) U_D \\ &= \left(\prod_{j=1}^p U_H^1(\vec{\theta}_1) U_H^2(\vec{\theta}_2) U_H^3(\vec{\theta}_3) U_H^4(\vec{\theta}_4) \right) U_D \end{aligned} \quad (37)$$

should prepare the ground state of H for sufficiently large $p = \tilde{p}q$, where we trotterized from the first to the second line. The equality of the last line holds if $\vec{\theta}_1 = (t_j/q)\vec{1}$ and $\vec{\theta}_{2,3,4} = (t_j/q)T_j\vec{1}$ for the corresponding j , which, therefore, is a good choice for the initial VQE parameters.

However, for the purpose of the VQE we leave the $\vec{\theta}_{1,2,3,4}$ as free variational parameters. Furthermore, instead of keeping the same parameters for each Trotter step, we choose different parameters $\vec{\theta}_{1,2,3,4}^j$ (with entries in $[0, 1]$) for every step, resulting in a total number of $54p$ parameters and the ansatz

$$U_{\text{HVA}} = \left(\prod_{j=1}^p U_H^1(\vec{\theta}_1^j) U_H^2(\vec{\theta}_2^j) U_H^3(\vec{\theta}_3^j) U_H^4(\vec{\theta}_4^j) \right) U_D \quad (38)$$

Assuming equal time steps $t_i \equiv 1/\tilde{p}$, and $\tilde{p} = q = \sqrt{p}$, the initial values according to equation (37) are

$$\vec{\theta}_1^j = \frac{\vec{1}}{p} \quad \text{and} \quad \vec{\theta}_{2,3,4}^j = \frac{\vec{1}}{p\sqrt{p}} \left(\left\lfloor \frac{j}{\sqrt{p}} \right\rfloor - \frac{1}{2} \right). \quad (39)$$

In complete analogy, the HVA ansatz starting from the trimer ground state found in subsection 3.2 is given by

$$U_{\text{HVA}}^\alpha = \left(\prod_{j=1}^p U_A(\vec{\theta}_A^j) U_B(\vec{\theta}_B^j) \right) U_{T\alpha} \quad (40)$$

where α labels the selected initial state and each $\vec{\theta}_{A,B}^j$ is a vector of 27 parameters amounting, again, to a total of $54p$ parameters, initialized with

$$\vec{\theta}_A^j = \frac{\vec{1}}{p} \quad \text{and} \quad \vec{\theta}_B^j = \frac{\vec{1}}{p\sqrt{p}} \left(\left\lfloor \frac{j}{\sqrt{p}} \right\rfloor - \frac{1}{2} \right). \quad (41)$$

5 Degenerate subspace optimization

The basic idea is that, since the Hamiltonian H has a degenerate ground state, we search for a 2-dimensional subspace rather than a single ground state, and then continue by searching the state with minimal energy on this subspace.

In terms of real vector spaces, this can be motivated as follows: Assume the eigenspace to the lowest eigenvalue is spanned by the normalized orthogonal vectors $\vec{v}_{1,2} \in \mathbb{R}^n$ with $\vec{v}_1 \cdot \vec{v}_2 = 0$, such that for any given $a \in [0, 1]$

$$\vec{v} = a\vec{v}_1 + \sqrt{1-a^2}\vec{v}_2 \quad (42)$$

is a normalized eigenvector to the lowest eigenvalue. Now say our algorithm finds orthogonal vectors $\vec{w}_{1,2} \in \mathbb{R}^n$ close to $\vec{v}_{1,2}$, respectively, i.e. with

$$\vec{v}_1 \cdot \vec{w}_1 = \vec{v}_2 \cdot \vec{w}_2 = 1 - \varepsilon \quad (43a)$$

$$\vec{v}_1 \cdot \vec{w}_2 = \vec{v}_2 \cdot \vec{w}_1 = \varepsilon \quad (43b)$$

Then we have for

$$\vec{w} = a\vec{w}_1 + \sqrt{1-a^2}\vec{w}_2 \quad (44)$$

that the overlap with the exact eigenstate \vec{v} is

$$\vec{v} \cdot \vec{w} = 1 - \varepsilon \left(1 - 2a\sqrt{1-a^2}\right) \quad (45)$$

which reaches a maximum value of 1 for $a = 1/\sqrt{2}$.

Hence, whereas \vec{w}_1 and \vec{w}_2 are both only approximately in the eigenspace with an error of ε , there exists a vector \vec{w} in the span of both vectors that overlaps exactly with an eigenvector \vec{v} . Although in practical situations the two vectors $\vec{w}_{1,2}$ will not be related to the pair $\vec{v}_{1,2}$ by a simple rotation as in this example, intuition suggests that there are still vectors in the subspace spanned by \vec{w}_1 and \vec{w}_2 which are closer to a ground state than either of these vectors.

In a complex vector space, the superposition will depend on three real parameters (global and relative phase, as well as one magnitude, with the other determined by the normalization constraint). Finding the optimal state in the two dimensional subspace is then an optimization problem with three real parameters and should show much better convergence with much less severe barren plateau issues. As we will see below, in practice we can only search a one-parameter subspace of the full three-parameter space.

5.1 Cost function

In order to apply this to our VQE problem, we start with the orthogonal initial trimer eigenstates $|\Psi_{T1}\rangle$ and $|\Psi_{T2}\rangle$ for our HVA ansatz. Since unitary evolution conserves orthogonality, the resulting ansatz states

$$|\Psi_{1,2}(\vec{\theta})\rangle = U_{\text{HVA}}^{1,2}(\vec{\theta}) |0\rangle^{\otimes n} \quad (46)$$

will be orthogonal, as long as we evolve them with the *same* parameters $\vec{\theta}$.

In order to find the degenerate subspace, we therefore optimize the cost function

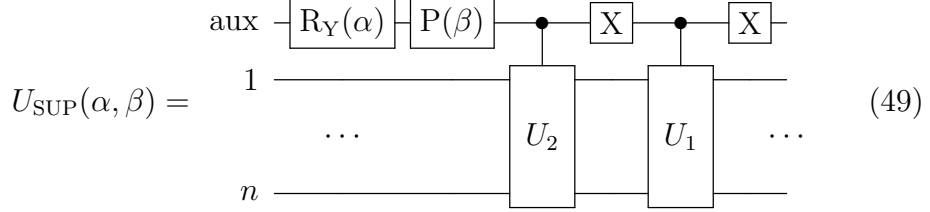
$$C(\vec{\theta}) = \langle \Psi_1(\vec{\theta}) | H | \Psi_1(\vec{\theta}) \rangle + \langle \Psi_2(\vec{\theta}) | H | \Psi_2(\vec{\theta}) \rangle \quad (47)$$

5.2 Superposition states via circuit

In principle, we can create a single qubit superposition state as

$$\cos \frac{\alpha}{2} |0\rangle + e^{i\beta} \sin \frac{\alpha}{2} |1\rangle = P(\beta) R_Y(\alpha) |0\rangle \quad (48)$$

Thus, for any two n -qubit states $|\Psi_{1,2}\rangle = U_{1,2}|0\rangle^{\otimes n}$ we can prepare a superposition $\cos \frac{\alpha}{2} |\Psi_1\rangle + e^{i\beta} \sin \frac{\alpha}{2} |\Psi_2\rangle$ via an auxiliary qubit:



Precisely speaking, this circuit prepares the entangled $n + 1$ qubit state $\cos \frac{\alpha}{2} |0\rangle_0 \otimes |\Psi_1\rangle_{1,\dots,n} + e^{i\beta} \sin \frac{\alpha}{2} |1\rangle_0 \otimes |\Psi_2\rangle_{1,\dots,n}$, which amounts to a superposition for operators acting only on the qubits $1, \dots, n$. However, creating the controlled gates $U_{1,2}$ will generally require a circuit of large depth.

For the circuits U_{T1} and U_{T2} as defined in equations (24a) and (24b), we can construct a circuit of depth 117 matched to the Guadalupe layout, as shown in appendix C. The full 16-qubit circuit representing the superposition is given by three parallel copies of this circuit for the qubits $(\text{aux}, i, j, k) \in \{(0, 2, 1, 4), (15, 10, 12, 13), (9, 11, 8, 5)\}$, respectively. That is, we use the adjacent qubits 0, 15, and 9 for each of the three trimer terms, with the same parameters α and β . We must then post-select for measurements where these three qubits have the same value. We also omit the final X on the auxiliary qubit and swap the circuits setting $U_1 = U_{T2}$ and $U_2 = U_{T1}$ in equation (49) which results in a shorter circuit. The prepared state is then

$$|\Psi_{\text{SUP}}\rangle = \left(e^{i\beta} \sin \frac{\alpha}{2} |0\rangle_{\text{aux}} \otimes |\psi_{T1}\rangle_{i,j,k} + \cos \frac{\alpha}{2} |1\rangle_{\text{aux}} \otimes |\psi_{T2}\rangle_{i,j,k} \right)^{\otimes 3}. \quad (50)$$

Hence, measuring $q_0 = q_{15} = q_9 = 0$ corresponds to finding the system in the initial state $|\Psi_{T1}\rangle$ and measuring $q_0 = q_{15} = q_9 = 1$ to $|\Psi_{T2}\rangle$.

Using this superposition state as the initial state in our HVA ansatz (40) results in the ansatz

$$U_{\text{HVA}}^{\text{SUP}} = \left(\prod_{j=1}^p U_A(\vec{\theta}_A^j) U_B(\vec{\theta}_B^j) \right) U_{\text{SUP}} \quad (51)$$

which uses 15 active qubits, has two more parameters, α and β , totalling to $54p + 2$ parameters, and a circuit depth of $102p + 117$.

We can either optimize the ansatz (51) independently, or use the parameters $\vec{\theta}_{\text{min}}$ obtained by minimizing the cost function (47) as fixed parameters and only optimize (51) over the (α, β) parameter space.

5.3 Superpositions in classical computation

Given m approximated ground states $|\Psi_i\rangle = U_i |0\rangle^{\otimes n}$ —i.e. given optimized parameter vectors $\vec{\theta}_i$ for the circuits U_i (which can be different from each other or the same)—we can calculate the matrix elements

$$H_{ij} = \langle \Psi_i | H | \Psi_j \rangle = e^{i\varphi_{ij}} h_{ij} \quad (52)$$

with $h_{ij}, \varphi_{ij} \in \mathbb{R}$, $h_{ij} = h_{ji}$, and $\varphi_{ij} + \varphi_{ji} = 0$, specifically $\varphi_{ii} = 0$.

For a generic superposition state

$$|\Psi\rangle = \sum_{i=1}^m \alpha_i e^{i\beta_i} |\Psi_i\rangle \quad (53)$$

with $\vec{\alpha}, \vec{\beta} \in \mathbb{R}^m$ and normalization constraint $\sum_{i=1}^m \alpha_i^2 = 1$, we find

$$\langle \Psi | H | \Psi \rangle = \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j e^{i(\beta_j - \beta_i)} H_{ij} = \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j e^{i(\beta_j - \beta_i + \varphi_{ij})} h_{ij} \quad (54)$$

Generally, we cannot calculate this expression, as we only have access to h_{ij} via measurements, and not the phases φ_{ij} . For $m \leq 3$, however, we can absorb the $m(m-1)/2$ nontrivial phases by the choice of the m parameters β_i (restricting the subspace spanned by the superposition states).

Specifically, for $m = 3$ we have

$$\begin{aligned} \langle \Psi | H | \Psi \rangle &= \alpha_1^2 (h_{11} - h_{33}) + \alpha_2^2 (h_{22} - h_{33}) + h_{33} \\ &\quad + 2\alpha_1 \alpha_2 h_{12} + 2\sqrt{1 - \alpha_1^2 - \alpha_2^2} (\alpha_1 h_{13} + \alpha_2 h_{23}) \end{aligned} \quad (55)$$

with free parameters $\alpha_{1,2}$, and for $m = 2$ we have

$$\langle \Psi | H | \Psi \rangle = \alpha^2 (h_{11} - h_{22}) + h_{22} + 2\alpha \sqrt{1 - \alpha^2} h_{12} \quad (56)$$

which depends only on the free parameter $\alpha = \alpha_1$ and is independent of the remaining relative global phase (the relative phase is fixed by $\varphi_{12} = \beta_2 - \beta_1$). Equation (56) has four local extrema (for $\sigma, \tau \in \{\pm 1\}$):

$$\alpha^{\min} = \pm \frac{1}{\sqrt{2}} \left(1 \pm \left(1 + \left(\frac{2h_{12}}{h_{11} - h_{22}} \right)^2 \right)^{-1/2} \right)^{1/2}, \quad (57)$$

resulting in the minimal eigenvalue

$$\langle H \rangle_{\min} = \frac{1}{2} \left(h_{11} + h_{22} - \sqrt{(h_{11} - h_{22})^2 + 4h_{12}^2} \right). \quad (58)$$

If $h_{12} = 0$, this is simply the smaller of h_{11} and h_{22} , becoming the smaller, the larger $|h_{12}|$.

Whereas the expectation values are easily measured, the off-diagonal matrix element h_{12} is not. Considering that $|\Psi_{1,2}\rangle$ are orthogonal, we can however write

$$\begin{aligned}
h_{12} &= |\langle \Psi_1 | H | \Psi_2 \rangle| \\
&= \left| \langle \Psi_1 | i e^{iHt} \frac{\partial}{\partial t} e^{-iHt} | \Psi_2 \rangle \right| \\
&= \left| \langle \Psi_1 | \frac{i}{\varepsilon} \left(e^{-iH\varepsilon} - 1 - \frac{\varepsilon^2}{2} \frac{\partial^2}{\partial \delta^2} e^{-iH\delta} \right) | \Psi_2 \rangle \right| \\
&= \left| \frac{1}{\varepsilon} \langle \Psi_1 | e^{-iH\varepsilon} | \Psi_2 \rangle + \frac{\varepsilon}{2} \langle \Psi_1 | H^2 e^{-iH\delta} | \Psi_2 \rangle \right| \\
&\geq \frac{1}{\varepsilon} |\langle \Psi_1 | e^{-iH\varepsilon} | \Psi_2 \rangle| - \frac{\varepsilon}{2} |\langle \Psi_1 | H^2 e^{-iH\delta} | \Psi_2 \rangle| \\
&\geq \frac{1}{\varepsilon} |\langle \Psi_1 | e^{-iH\varepsilon} | \Psi_2 \rangle| - \frac{\varepsilon}{2} \|H^2 e^{-iH\delta}\| \\
&\geq \frac{1}{\varepsilon} |\langle \Psi_1 | e^{-iH\varepsilon} | \Psi_2 \rangle| - \frac{\varepsilon}{2} \|H\|^2 \\
&= \frac{1}{\varepsilon} |\langle \Psi_1 | e^{-iH\varepsilon} | \Psi_2 \rangle| - \frac{\varepsilon}{2} 18^2, \tag{59}
\end{aligned}$$

where in the third line we evaluate the (time independent) expression at $t = 0$ and use first order Taylor expansion with remainder in Lagrange form for some $\delta \in [0, \varepsilon]$. In the fourth line we make use of the orthogonality $\langle \Psi_1 | \Psi_2 \rangle = 0$.

We can then trotterize the exponential to second order

$$U_T(\varepsilon) = U_Z(\varepsilon \vec{1}) U_Y(\varepsilon \vec{1}) U_X(\varepsilon \vec{1}) U_X(\varepsilon \vec{1}) U_Y(\varepsilon \vec{1}) U_Z(\varepsilon \vec{1}) \tag{60}$$

where we use the term grouping with the smallest number of mutually commuting clusters into $U_{X,Y,Z}$, where there ordering is chosen to minimize the circuit depth. This circuit has the error bound [3]

$$\|U_T(\varepsilon) - e^{-iH\varepsilon}\| \leq \frac{\varepsilon^3}{24} (2L_X + 2L_Y + M_X + M_Y) \tag{61}$$

where

$$\begin{aligned}
L_X &= \|[[H_X, H_Y + H_Z], H_Y + H_Z]\| \\
&\leq \| [H_Y, [H_X, H_Y]] \| + \| [H_Y, [H_X, H_Z]] \| \\
&\quad + \| [H_Z, [H_X, H_Y]] \| + \| [H_Z, [H_X, H_Z]] \| \\
L_Y &= \| [H_Z, [H_Y, H_Z]] \| \\
M_X &= \| [H_X, [H_X, H_Y + H_Z]] \| \\
&\leq \| [H_X, [H_X, H_Y]] \| + \| [H_X, [H_X, H_Z]] \| \\
M_Y &= \| [H_Y, [H_Y, H_Z]] \|
\end{aligned} \tag{62}$$

and $\|A\|$ denotes the spectral norm of A (i.e. $\sqrt{\lambda}$ where λ is the largest eigenvalue of $A^\dagger A$). Hence, writing

$$N_{\rho\sigma\tau} = \| [H_\rho, [H_\sigma, H_\tau]] \| \tag{63}$$

we have

$$\begin{aligned}
\|U_T(\varepsilon) - e^{-iH\varepsilon}\| &\leq \frac{\varepsilon^3}{24} (2N_{YXY} + 2N_{YXZ} + 2N_{ZXY} + 2N_{ZXZ} \\
&\quad + 2N_{ZYZ} + N_{XXY} + N_{XXZ} + N_{YYZ}) \\
&\approx 283\varepsilon^3
\end{aligned} \tag{64}$$

With the trotterized circuit

$$U_{\text{off}}(\varepsilon, \vec{\theta}) = U_{\text{HVA}}^{1\dagger}(\vec{\theta}) U_T(\varepsilon) U_{\text{HVA}}^2(\vec{\theta}) \tag{65}$$

evaluated at the parameters $\vec{\theta}_{\min}$ minimizing the cost function (47), we can obtain

$$P(\varepsilon) = \left| \langle 0 |^{\otimes n} U_{\text{off}}(\varepsilon, \vec{\theta}_{\min}) | 0 \rangle^{\otimes n} \right|^2 \tag{66}$$

by preparing the state $U_{\text{off}}(\varepsilon, \vec{\theta}_{\min}) | 0 \rangle^{\otimes n}$ and measuring the probability of finding $| 0 \rangle^{\otimes n}$. We then use the lower bound

$$h_{12} = \max_{\varepsilon} \left(\sqrt{\frac{P(\varepsilon)}{\varepsilon^2}} - 283\varepsilon - 162\varepsilon \right) \tag{67}$$

and, assuming this expression is positive, can obtain an upper bound on the ground state energy via equation (58).

A note on scalability Calculating the norms $N_{\rho\sigma\tau}$ numerically is quite time consuming. A better scalable approach would be to estimate the norms by counting the number of noncommuting Pauli strings involved in the double commutators. These can be easily counted to be 456 for N_{YXZ} and N_{ZXY} , and 408 for the remaining six double commutators, where every double commutator adds 4 to the norm (note that the nonzero commutators of two Pauli strings are $2i^k$ times another Pauli string, and all Pauli strings have norm 1), amounting to an upper bound of $916\varepsilon^3$ instead in equation (64). Similarly, we could have used the total number of Pauli strings as a bound on $\|H\| \leq 54$ in equation (59) for better scalability.

Accuracy of the limit All inequalities given in this section are, in principle, precise. Thus, in theory, $\langle H \rangle_{\min}$ obtained according to equation (58) is guaranteed not to become smaller than the exact value. A measurement error resulting in a too large value $P(\varepsilon)$ could nonetheless result in a value of $\langle H \rangle_{\min}$ below the exact one. This is, however, not different from the standard VQE algorithm, which possesses the exact ground state energy as a theoretical bound but could fall below it by coincidence due to measurement errors on single operators adding up. **A crucial difference is that noise is typically expected to drive the states $|\Psi_{1,2}\rangle$ away from a ground state, which for the variational principle implies that it increases the expectation value. For the off-diagonal h_{12} here, the same noise would result in a larger $P(\varepsilon)$ for small ε and, thus, increase h_{12} , i.e. decrease the ground state energy, potentially below its exact value.**

6 Measuring the matrix elements

In order to obtain the expectation value

$$H_{\alpha\alpha} = \langle \Psi_\alpha | H | \Psi_\alpha \rangle = \langle 0 | U_{\text{HVA}}^\alpha \dagger H U_{\text{HVA}}^\alpha | 0 \rangle \quad (68)$$

we can use Qiskit Runtime estimator. First we load Qiskit Runtime with options. Note that we choose to skip transpilation, assuming that the ansatz circuits are already optimized using hardware gates.

```
from qiskit_ibm_runtime import QiskitRuntimeService,
    Session, Estimator, Sampler, Options
from qiskit.providers.fake_provider import FakeGuadalupeV2
from qiskit_aer.noise import NoiseModel

# Load Qiskit Runtime Service
service = QiskitRuntimeService(channel='ibm_quantum')
```

```

# For simulator use FakeGuadalupe as model:
fake_backend = FakeGuadalupeV2()
noise_model = NoiseModel.from_backend(fake_backend)
coupling_map = fake_backend.coupling_map
basis_gates = fake_backend.operation_names
options = Options(simulator={'noise_model': noise_model,
                             'coupling_map': coupling_map,
                             'basis_gates': basis_gates},
                  transpilation={'skip_transpilation': True})

# Select the backend
backend = service.backends(simulator=True)[0]

```

The Hamiltonian can be defined from a list of edges of the Kagome lattice, taking care of the correct assignment of vertices to physical qubits:

```

from qiskit.quantum_info import SparsePauliOp
from qiskit.opflow.primitive_ops import PauliSumOp

num_qubits = 16

# every edge in the Kagome lattice contributes three Pauli
# strings with X/Y/Z on the corresponding vertices:
kagome_edges = [(2,1),(1,4),(4,7),(7,10),(10,12),(12,13),
                (13,14),(14,11),(11,8),(8,5),(5,3),(3,2),
                (2,4),(4,10),(10,13),(13,11),(11,5),(5,2)]

# Construct the hamiltonians for the estimator
pauli_strings = {'X': [], 'Y': [], 'Z': []}
for e in kagome_edges:
    for c in 'XYZ':
        s = ''
        for i in range(num_qubits):
            if i in e:
                s = c + s
            else:
                s = 'I' + s
        pauli_strings[c].append(s)
hamiltonian = PauliSumOp(SparsePauliOp(pauli_strings['X']
                                         +pauli_strings['Y']
                                         +pauli_strings['Z']))

```

The routine to estimate the expectation value is then given by:

```

def expectation_value(hamiltonian, ansatz, parameters):
    with Session(service=service, backend=backend):
        estimator = Estimator(options=options)
        job = estimator.run(circuits=[ansatz],
                           parameter_values=[parameters],
                           observables=[hamiltonian])

```

```

    res = job.result()
    ev = res.values[0]
    return ev

```

where `ansatz` is the ansatz circuit U_{HVA} as defined in equations (38) for the dimer, (40) for the trimer ground state, and (51) for the superposition, respectively, and `ansatz_parameters` the vector $\vec{\theta}$ parameterizing this circuit.

6.1 Measuring a Pauli string

Note that we can represent any hermitian operator H in terms of its orthonormal eigenbasis:

$$H = \sum_i h_i |h_i\rangle \langle h_i| \quad (69)$$

Let B be the unitary basis transformation to the computational basis, i.e. $|i\rangle = B |h_i\rangle$. Then we have

$$\tilde{H} = B H B^\dagger = \sum_i h_i B |h_i\rangle \langle h_i| B^\dagger = \sum_i h_i |i\rangle \langle i| \quad (70)$$

diagonal in the computational basis. For an ansatz state $|\psi\rangle = U |0\rangle$ with some unitary U , we then have

$$\begin{aligned} \langle \psi | H | \psi \rangle &= \langle \psi | B^\dagger \tilde{H} B | \psi \rangle \\ &= \sum_i h_i \langle 0 | U^\dagger B^\dagger | i \rangle \langle i | B U | 0 \rangle \\ &= \sum_i h_i |\langle i | B U | 0 \rangle|^2 \\ &= \sum_i h_i P_i(B | \psi \rangle) \end{aligned} \quad (71)$$

where $P_i(B | \psi \rangle)$ is the probability to measure $|i\rangle$ in the prepared state $B | \psi \rangle$.

Writing the single-qubit Pauli operators as $\sigma_0 = \text{I}$, $\sigma_1 = \text{X}$, $\sigma_2 = \text{Y}$, $\sigma_3 = \text{Z}$, we can define n -qubit Pauli strings by vectors $\vec{\mu} \in \{0, 1, 2, 3\}^n$:

$$H_{\vec{\mu}} = \prod_{i=0}^{n-1} \sigma_{\mu_i}^{(i)} \quad \text{with} \quad \mu_i \in \{0, 1, 2, 3\}. \quad (72)$$

$H_{\vec{\mu}}$ is diagonal in the basis

$$|\tilde{s}\rangle = \bigotimes_{i=0}^{n-1} |(s_i)_{\mu_i}\rangle \quad (73)$$

where $s \in \{0, 1, \dots, 2^n - 1\}$, $s_i \in \{0, 1\}$ denotes the value of the i -th bit of s in binary form, and we have

$$\begin{aligned} |0_1\rangle &= |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), & |1_1\rangle &= |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\ |0_2\rangle &= |y_+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle), & |1_2\rangle &= |y_-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \\ |0_3\rangle &= |0_0\rangle = |0\rangle, & |1_3\rangle &= |1_0\rangle = |1\rangle \end{aligned} \quad (74)$$

The corresponding eigenvalues $H_{\vec{\mu}} |\tilde{s}\rangle = h_{\vec{\mu}} |\tilde{s}\rangle$ are

$$h_{\vec{\mu}} = \prod_{i=0}^{n-1} (-1)^{r_i s_i} = \pm 1, \quad (75)$$

where

$$r_i = 1 - 0^{\mu_i} = \begin{cases} 0 & \text{for } \sigma^{(i)} = I_i \\ 1 & \text{for } \sigma^{(i)} \in \{X_i, Y_i, Z_i\} \end{cases} \quad (76)$$

i.e. each X_i , Y_i , or Z_i contributes a factor of -1 if the corresponding bit $s_i = 1$, whereas all qubits with either an I_i (and therefore $\mu_i = 0$) in the Pauli string or a bit $s_i = 0$ in the state contribute a factor of 1.

For single qubits $s_i \in \{0, 1\}$ we have the transformations from the x and y bases to the computational basis

$$H |(s_i)_1\rangle = |(s_i)_0\rangle \quad (77)$$

$$HS^\dagger |(s_i)_2\rangle = |(s_i)_0\rangle \quad (78)$$

(Note the difference in notation: H is the Hadamard gate, whereas the Hamiltonian is denoted H in italics.) The basis transformation B that renders $H_{\vec{\mu}}$ diagonal in the n -qubit computational basis, is then the tensor product

$$B_{\vec{\mu}} = \bigotimes_{i=0}^{n-1} B_{\mu_i} \quad \text{with} \quad B_1 = H, B_2 = HS^\dagger, B_3 = B_0 = I. \quad (79)$$

As a circuit, $B_{\vec{\mu}}$ appends \boxed{H} to all qubits where the Pauli string $H_{\vec{\mu}}$ has an X and $\boxed{S^\dagger} \boxed{H}$ to all qubits where it has a Y .

Generally, commuting Pauli strings can be measured with the same basis transformation B . In our case, where $H = H^X + H^Y + H^Z$ can be split in parts containing only I as well as X , Y , or Z , respectively, we can measure all

Pauli strings in $H^{X,Y,Z}$ simultaneously, requiring only three measurements in total with the circuits:

$$U_{\text{meas}}^X = \begin{array}{c} 0 \\ 1 \\ \dots \\ 15 \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} \boxed{U_{\text{HVA}}^{\text{SUP}}} \\ \boxed{U_{\text{HVA}}^{\text{SUP}}} \\ \boxed{U_{\text{HVA}}^{\text{SUP}}} \\ \boxed{U_{\text{HVA}}^{\text{SUP}}} \end{array} \begin{array}{c} \boxed{H} \\ \boxed{H} \\ \dots \\ \boxed{H} \end{array} \begin{array}{c} \boxed{\text{meter}} \\ \boxed{\text{meter}} \\ \dots \\ \boxed{\text{meter}} \end{array} \quad (80)$$

$$U_{\text{meas}}^Y = \begin{array}{c} 0 \\ 1 \\ \dots \\ 15 \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} \boxed{U_{\text{HVA}}^{\text{SUP}}} \\ \boxed{U_{\text{HVA}}^{\text{SUP}}} \\ \boxed{U_{\text{HVA}}^{\text{SUP}}} \\ \boxed{U_{\text{HVA}}^{\text{SUP}}} \end{array} \begin{array}{c} \boxed{S^\dagger} \\ \boxed{S^\dagger} \\ \dots \\ \boxed{S^\dagger} \end{array} \begin{array}{c} \boxed{H} \\ \boxed{H} \\ \dots \\ \boxed{H} \end{array} \begin{array}{c} \boxed{\text{meter}} \\ \boxed{\text{meter}} \\ \dots \\ \boxed{\text{meter}} \end{array} \quad (81)$$

$$U_{\text{meas}}^Z = \begin{array}{c} 0 \\ 1 \\ \dots \\ 15 \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} \boxed{U_{\text{HVA}}^{\text{SUP}}} \\ \boxed{U_{\text{HVA}}^{\text{SUP}}} \\ \boxed{U_{\text{HVA}}^{\text{SUP}}} \\ \boxed{U_{\text{HVA}}^{\text{SUP}}} \end{array} \begin{array}{c} \boxed{\text{meter}} \\ \boxed{\text{meter}} \\ \dots \\ \boxed{\text{meter}} \end{array} \quad (82)$$

The measurement results are the probabilities $P^{X,Y,Z}(s)$ for all possible bit-strings $s \in \{0, 1, \dots, 2^{16} - 1\}$.

From the eigenvalues (75) we can calculate the weights as the sum over all Kagome edges

$$w_K(s) = \sum_{\langle i,j \rangle \in \Gamma} (-1)^{s_i + s_j}. \quad (83)$$

Note that, for the Kagome lattice, all these weights are nonzero. In order to correctly account for the superposition in our ansatz (51), we must post-select those results with identical values on the auxiliary qubits 0, 9, 15. Therefore, we introduce the modified weight function

$$w_S(s) = \begin{cases} w_K(s) & \text{if } s_0 = s_9 = s_{15} \\ 0 & \text{else} \end{cases} \quad (84)$$

and the cumulative probability of all the disregarded results

$$\bar{P}^{X,Y,Z} = \sum_{\substack{s=0 \\ w_S(s)=0}}^{2^{16}-1} P^{X,Y,Z}(s). \quad (85)$$

We then have the expectation value

$$\langle \psi | H | \psi \rangle = \sum_{\sigma \in \{X,Y,Z\}} \frac{\sum_{s=0}^{2^{16}-1} w_S(s) P^\sigma(s)}{1 - \overline{P}^\sigma} \quad (86)$$

The weights w_S are calculated using the following code:

```
# set number of qubits and the format string used to turn
# numbers into bitstrings
num_qubits = 16
format_string = '016b' # must match num_qubits

kagome_edges = [(2,1),(1,4),(4,7),(7,10),(10,12),(12,13),
                (13,14),(14,11),(11,8),(8,5),(5,3),(3,2),
                (2,4),(4,10),(10,13),(13,11),(11,5),(5,2)]
aux_qubits = (0,9,15) # these must be all 0 or all 1

# convert the Pauli strings into a bitstring, where the
# position of X/Y/Z is marked by a 1 and I by a 0
ops = list(map(lambda x: 2**x[0]+2**x[1], kagome_edges))

# calculate the weights
weights = []
for i in range(2**num_qubits):
    w = 0
    aux_vals = [i//2**aux%2 for aux in aux_qubits]
    if all(aux_vals) or not any(aux_vals):
        for o in ops:
            s = format(i & o, format_string)
            w += (-1)**(s.count('1'))
    weights.append(w)
```

6.2 Superposition states

In order to use the superposition ansatz (51) to optimize the ground state energy over the degenerate subspace, we must condition measurement results on the auxiliary qubits. Hence, we use `Sampler` rather than `Estimator`.

```
from math import pi

def sampler_expectation_value(weights, ansatz, parameters):
    # Construct circuits for X, Y, Z measurements
    num_qubits = ansatz.num_qubits
    qcX = QuantumCircuit(num_qubits)
    qcY = QuantumCircuit(num_qubits)
    qcZ = QuantumCircuit(num_qubits)
    for qc in [qcX, qcY, qcZ]:
```



```

        qc.append(ansatz, range(num_qubits))
    for i in range(num_qubits):
        # Apply Hadamard for rotation to X-basis
        qcX.rz(pi/2, i)
        qcX.sx(i)
        qcX.rz(pi/2, i)
        # Apply H Sdg for rotation to Y-basis
        qcY.sx(i)
        qcY.rz(pi/2, i)
    for qc in [qcX, qcY, qcZ]:
        qc.measure_all()
    # Sample quantum circuits with ansatz parameters:
    with Session(service=service, backend=backend):
        sampler = Sampler(options=options)
        job = sampler.run(circuits=[qcX, qcY, qcZ],
                          parameter_values=[parameters]*3)
        res = job.result()
    ev = 0
    # sum over the probabilities for X, Y, Z circuit
    for quasi_dist in res.quasi_dists:
        pd = quasi_dist.nearest_probability_distribution()
        ev_part = 0
        p_bar = 0
        for i in pd.keys():
            if weights[i]:
                ev_part += weights[i] * pd[i]
            else:
                p_bar += pd[i]
        ev += ev_part / (1 - p_bar)
    return ev

```

6.3 Off-diagonal elements

In order to use equations (58) and (67) to optimize the ground state energy over the degenerate subspace classically, we need to measure $P(\varepsilon)$, i.e. sample the circuit U_{off} as defined in equation (65) and read the probability for obtaining a zero-bitstring:

```

def off_diagonal_p(epsilon, circuit, parameters):
    with Session(service=service, backend=backend):
        sampler = Sampler(options=options)
        job = sampler.run(circuits=[circuit],
                          parameter_values=[parameters])
        res = job.result()
    prob_dist = res.quasi_dists[0]
    .nearest_probability_distribution()
    if 0 in prob_dist.keys():

```

```

    return prob_dist[0]
return 0

```

where `circuit` is a transpiled version of the circuit U_{off} and `parameters` the optimal set of ansatz parameters $\vec{\theta}_{\text{min}}$.

7 Optimizer

We use Rotosolve [4, 5] to optimize the parameter vector $\vec{\theta} \in \mathbb{R}^l$. We decompose the HVA circuit

$$U(\vec{\theta}) = V_0 \prod_{i=1}^l R_Z^{(q_i)}(\theta_i) V_i \quad (87)$$

where V_i ($i = 0, \dots, l$) are arbitrary unparameterized unitaries and R_Z rotations acting on qubits q_i . For a given set of parameters $\vec{\theta}^0$ and a given index i , we then have the single parameter circuit

$$U_i(\theta) = U(\theta_1^0, \theta_2^0, \dots, \theta_{i-1}^0, \theta, \theta_{i+1}^0, \dots, \theta_l^0) \quad (88)$$

obtained by keeping all parameters except for the i -th fixed.

If the cost function $C(\vec{\theta})$ is a weighted sum of expectation values of hermitian operators, it can be shown that the cost function with respect to only the i -th parameter

$$C_i(\theta) = C(\theta_1^0, \theta_2^0, \dots, \theta_{i-1}^0, \theta, \theta_{i+1}^0, \dots, \theta_l^0) \quad (89)$$

takes the form

$$C_i(\theta) = \alpha \cos(\theta - \beta) + \gamma. \quad (90)$$

with $\alpha \in [0, \infty)$, $\beta \in [0, 2\pi)$, $\gamma \in \mathbb{R}$. Measuring only the three values

$$c_0 = C_i(\theta_i^0), \quad c_- = C_i(\theta_i^0 - \frac{\pi}{2}), \quad c_+ = C_i(\theta_i^0 + \frac{\pi}{2}), \quad (91)$$

we can find analytically

$$\alpha = \sqrt{\frac{(c_0 - c_+)^2 + (c_0 - c_-)^2}{2}} \quad (92)$$

$$\beta = \theta_i^0 + \text{atan2}(c_+ - c_-, 2c_0 - c_+ - c_-) \mod 2\pi \quad (93)$$

$$\gamma = \frac{c_+ + c_-}{2} \quad (94)$$

where atan2 is the 2-argument arctangent function, i.e. $\text{atan2}(y, x)$ is the angle between the ray from the origin to the point (x, y) and the x -axis. In

fact, we only need to use that $C_i(\theta)$ takes its minimum for $\theta = \beta + \pi$. We have the updated parameter, as well as the next c_0 value, given by

$$\theta_i^{\text{new}} = \theta_i^0 + \text{atan2}(c_+ - c_-, 2c_0 - c_+ - c_-) + \pi \mod 2\pi \quad (95)$$

$$C(\vec{\theta}^{\text{new}}) = \gamma - \alpha = \frac{c_+ + c_-}{2} - \sqrt{\frac{(c_0 - c_+)^2 + (c_0 - c_-)^2}{2}} \quad (96)$$

We draw the next index to consider at random from the first 50 parameters of a list of indices, giving quadratically decreasing likelihood, i.e. the longer a parameter has not been optimized the more likely it is to be chosen:

```
import numpy as np

def next_parameter_index(indices):
    # choose from first 50 parameter indices
    n_params = min(len(indices), 50)
    # first choose a random number between 1 and 2^n
    random_i = np.random.randint(1, 2**n_params)
    # index is chosen with quadratically decreasing
    # likelihood,
    # i.e. 0 is twice as likely as 1 is twice as likely as 2,
    # ...
    i_in_indices = n_params - 1 - int(np.floor(np.log2(
        random_i)))
    # pop next index from indices and append it at the end
    next_index = indices.pop(i_in_indices)
    indices.append(next_index)
    return next_index
```

We can initialize `indices = list(range(len(theta)))`.

We define the cost function according to equation (47)

```
def cost_function(theta):
    ev1 = expectation_value(hamiltonian, uhva1, theta)
    ev2 = expectation_value(hamiltonian, uhva2, theta)
    return ev1 + ev2
```

where `uhva1` and `uhva2` are the circuits $U_{\text{HVA}}^{1,2}$ according to equation (40)

A single optimization step takes the parameter vector $\vec{\theta}$ and an index to be optimized. It updates the parameter vector to an optimized parameter vector and returns a tuple with the optimized value for the cost function and the difference to the previous value:

```
import numpy as np

def optimization_step(theta, index, c0 = None):
    # cost function at old theta values if not provided
    if c0 is None:
```

```

        c0 = cost_function(theta)
        # new theta vectors with theta_i +/- pi/2
        tm = theta[:index]+[theta[index]-pi/2]+theta[index+1:]
        tp = theta[:index]+[theta[index]+pi/2]+theta[index+1:]
        # cost function at +/- pi/2
        cm = cost_function(tm)
        cp = cost_function(tp)
        dp = c0 - cp
        dm = c0 - cm
        # update theta
        theta[index] += np.arctan2(dp+dm, cp-cm)
        # return new cost function value (next c0)
        new_cost = (cp+cm)/2 - np.sqrt((dp**2+dm**2)/2)
        return (new_cost, c0-new_cost)

```

We also add a simple function to save all optimization steps to a file:

```

def save_step(filename, step, cost, change, theta):
    with open(filename, 'a') as f:
        f.write('STEP {}'.format(step))
        f.write('  COST:  {}'.format(cost))
        f.write('  CHANGE: {}'.format(change))
        f.write('  PARAMS: {}'.format(theta))
    f.close()

```

Hence, a minimal optimization procedure, starting with the initial parameters according to equation (41), could look like

```

import numpy as np

# path to file that takes the results
outfile = 'test.txt'

num_layers = 1
ha_params = 27
hb_params = 27
num_params = (ha_params+hb_params)*num_layers

# stopping criterion:
min_steps = num_params
max_steps = 10000
# stop if cost function does not change by more than
# stopping_delta for stopping_after consecutive steps
stopping_delta = .0001
stopping_after = 10

# initialize parameters and indices
indices = list(range(num_params))
theta = []
for j in range(num_layers):

```

```

    pa = 1/num_layers
    sqp = np.sqrt(num_layers)
    pb = (np.floor((1+j)/sqp) - .5) / sqp / num_layers
    for _ in range(ha_params):
        theta.append(pa)
    for _ in range(hb_params):
        theta.append(pb)
# cost function evaluated in first optimization step
    cost = None

    last_change = 0 # last step with minimum required change
    # optimization loop
    for step in range(max_steps):
        index = next_parameter_index(indices)
        cost, change = optimization_step(theta, index, cost)
        save_step(outfile, step, cost, change, theta)
        if change < stopping_delta:
            if step >= max(min_steps - 1,
                           last_change + stopping_after):
                break
        else:
            last_change = step

```

A Dimer ground state

Written out for 12 qubits, the dimer ground state (19) for H_1 is

$$\begin{aligned}
|\Psi_D^1\rangle = \frac{1}{8} & \left(|010101010101\rangle - |010101010110\rangle - |010101011001\rangle + |010101011010\rangle \right. \\
& - |010101100101\rangle + |010101100110\rangle + |010101101001\rangle - |010101101010\rangle \\
& - |010110010101\rangle + |010110010110\rangle + |010110011001\rangle - |010110011010\rangle \\
& + |010110100101\rangle - |010110100110\rangle - |010110101001\rangle + |010110101010\rangle \\
& - |011001010101\rangle + |011001010110\rangle + |011001011001\rangle - |011001011010\rangle \\
& + |010101100101\rangle - |010101100110\rangle - |010101101001\rangle + |010101101010\rangle \\
& + |011010010101\rangle - |011010010110\rangle - |011010011001\rangle + |011010011010\rangle \\
& - |011010100101\rangle + |011010100110\rangle + |011010101001\rangle - |011010101010\rangle \\
& - |100101010101\rangle + |100101010110\rangle + |100101011001\rangle - |100101011010\rangle \\
& + |100101100101\rangle - |100101100110\rangle - |100101101001\rangle + |100101101010\rangle \\
& + |100110010101\rangle - |100110010110\rangle - |100110011001\rangle + |100110011010\rangle \\
& - |100110100101\rangle + |100110100110\rangle + |100110101001\rangle - |100110101010\rangle \\
& + |101001010101\rangle - |101001010110\rangle - |101001011001\rangle + |101001011010\rangle \\
& - |100101100101\rangle + |100101100110\rangle + |100101101001\rangle - |100101101010\rangle \\
& - |101010010101\rangle + |101010010110\rangle + |101010011001\rangle - |101010011010\rangle \\
& \left. + |101010100101\rangle - |101010100110\rangle - |101010101001\rangle + |101010101010\rangle \right) \quad (97)
\end{aligned}$$

whereas the ground state for H_2 is the same state rotated by one qubit:

$$\begin{aligned}
|\Psi_D^2\rangle = \frac{1}{8} & \left(|101010101010\rangle - |101010101100\rangle - |101010110010\rangle + |101010110100\rangle \right. \\
& - |101011001010\rangle + |101011001100\rangle + |101011010010\rangle - |101011010100\rangle \\
& - |101100101010\rangle + |101100101100\rangle + |101100110010\rangle - |101100110100\rangle \\
& + |101101001010\rangle - |101101001100\rangle - |101101010010\rangle + |101101010100\rangle \\
& - |110010101010\rangle + |110010101100\rangle + |110010110010\rangle - |110010110100\rangle \\
& + |101011001010\rangle - |101011001100\rangle - |101011010010\rangle + |101011010100\rangle \\
& + |110100101010\rangle - |110100101100\rangle - |110100110010\rangle + |110100110100\rangle \\
& - |110101001010\rangle + |110101001100\rangle + |110101010010\rangle - |110101010100\rangle \\
& - |001010101011\rangle + |001010101101\rangle + |001010110011\rangle - |001010110101\rangle \\
& + |001011001011\rangle - |001011001101\rangle - |001011010011\rangle + |001011010101\rangle \\
& + |001100101011\rangle - |001100101101\rangle - |001100110011\rangle + |001100110101\rangle \\
& - |001101001011\rangle + |001101001101\rangle + |001101010011\rangle - |001101010101\rangle \\
& + |010010101011\rangle - |010010101101\rangle - |010010110011\rangle + |010010110101\rangle \\
& - |001011001011\rangle + |001011001101\rangle + |001011010011\rangle - |001011010101\rangle \\
& - |010100101011\rangle + |010100101101\rangle + |010100110011\rangle - |010100110101\rangle \\
& \left. + |010101001011\rangle - |010101001101\rangle - |010101010011\rangle + |010101010101\rangle \right) \quad (98)
\end{aligned}$$

Hence, we have $\langle \Psi_D^1 | \Psi_D^2 \rangle = \frac{1}{32}$.

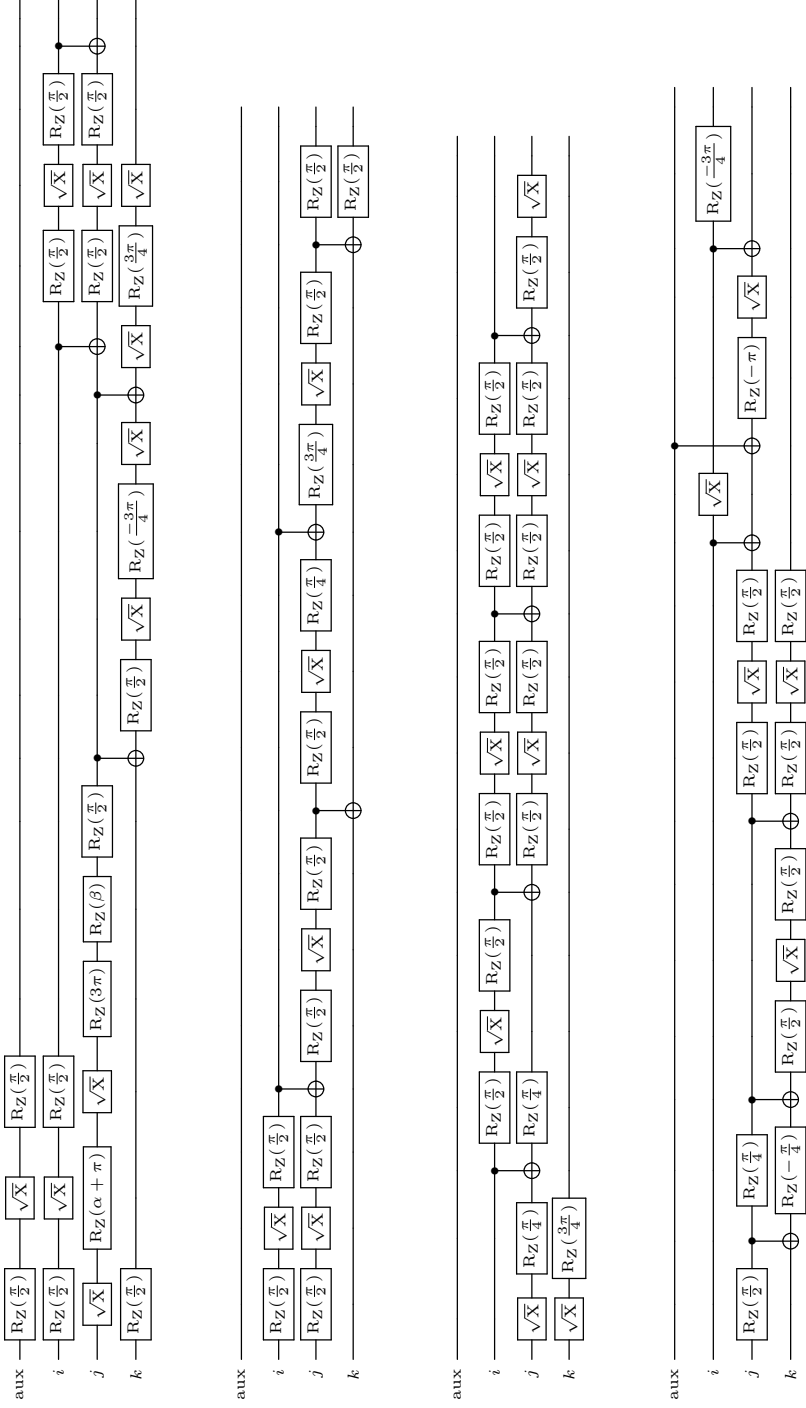
B Double-dimer ground state circuit

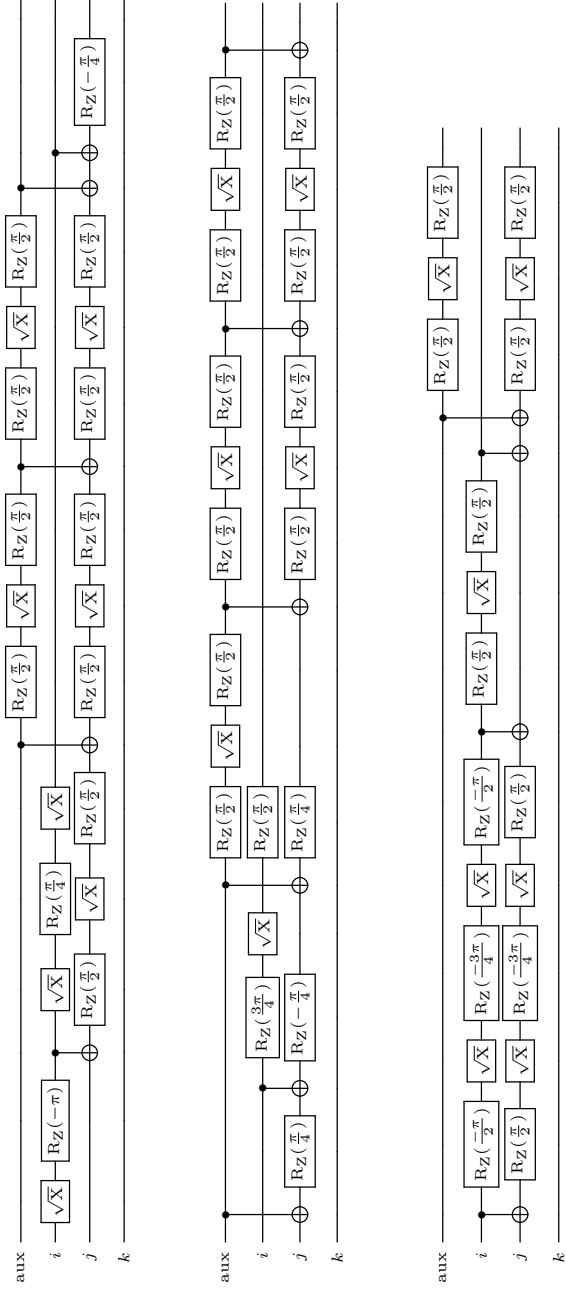
We introduce $\Omega = \arccos(1/3) - \pi \approx -1.91$ and write the transpiled circuit for the double-dimer ground state

$$U_{DD1}^{(i,j,k)} = \begin{array}{c} \begin{array}{c} i \\ j \\ k \end{array} \begin{array}{c} \boxed{R_Z(-\pi)} \quad \boxed{\sqrt{X}} \quad \boxed{R_Z(\frac{3\pi}{4})} \quad \oplus \quad \boxed{R_Z(-\frac{3\pi}{4})} \quad \boxed{\sqrt{X}} \quad \boxed{R_Z(-\pi)} \\ \boxed{\sqrt{X}} \quad \boxed{R_Z(\Omega)} \quad \boxed{\sqrt{X}} \quad \oplus \quad \boxed{R_Z(-\frac{\pi}{4})} \quad \boxed{\sqrt{X}} \quad \boxed{R_Z(-\frac{3\pi}{4})} \quad \oplus \quad \boxed{R_Z(\frac{\pi}{4})} \quad \boxed{\sqrt{X}} \quad \boxed{R_Z(-\frac{\pi}{4})} \quad \oplus \quad \boxed{R_Z(\frac{3\pi}{4})} \quad \boxed{\sqrt{X}} \quad \boxed{R_Z(\frac{\pi}{2})} \end{array} \end{array} \quad (99a)$$

C Superposition quantum circuit

The following circuit represents a superposition of the trimer circuits U_{T1} and U_{T2} :





References

- [1] J. Kattemölle and J. van Wezel, Physical Review B **106**, 214429 (2022), arxiv:2108.02175 [cond-mat, physics:quant-ph] .
- [2] J. L. Bosse and A. Montanaro, “Probing ground state properties of the kagome antiferromagnetic Heisenberg model using the Variational Quantum Eigensolver,” (2021), arxiv:2108.08086 [quant-ph] .
- [3] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, Physical Review X **11**, 011020 (2021).
- [4] K. M. Nakanishi, K. Fujii, and S. Todo, Physical Review Research **2**, 043158 (2020).
- [5] M. Ostaszewski, E. Grant, and M. Benedetti, Quantum **5**, 391 (2021).