Quantum Circuit for Estimating the Ground State Energy of Molecules

Robert R. Tucci
P.O. Box 226
Bedford, MA 01730
tucci@ar-tiste.com

May 28, 2016

Abstract

The purpose of this paper is to present some technical details on a possible quantum circuit that represents a Trotter approximation of the evolution operator $U = e^{-itH}$ over time t for a Hamiltonian H that applies to molecules. Such a quantum circuit is useful in order to estimate the ground state energy of molecules via Kitaev's Phase Estimation Algorithm.

1 Introduction

The purpose of this paper is to present some technical details on a possible quantum circuit that represents a Trotter approximation of the evolution operator $U = e^{-itH}$ over time t for a Hamiltonian H given by

$$H = \sum_{k_1,k_2} w_{k_1,k_2} a^\dagger(k_1)a(k_2) + \sum_{k_1,k_2,k_3,k_4} w_{k_1,k_2,k_3,k_4} a^\dagger(k_1)a^\dagger(k_2)a(k_3)a(k_4) \ , \tag{1}$$

where $k_j \in \{0, 1, 2, \ldots, n-2\}$ for $j = 1, 2, 3, 4$. In this Hamiltonian, $a()$ and $a^\dagger()$ represent fermionic degrees of freedom that satisfy the anticommutation relations ($[x, y]_+ = xy + yx$)

$$[a(k_1), a(k_2)]_+ = 0 \ , \quad [a(k_1), a^\dagger(k_2)]_+ = \delta(k_1, k_2) \tag{2}$$

for all $k_1, k_2 \in \{0, 1, 2, \ldots, n-2\}$.

In this paper, n will denote the total number of qubits in our circuit[1]. The letter k with or without a subscript will stand for integers ranging over $\{0, 1, 2, \ldots, n-2\}$ for a total of $n-1$ orbitals. The qubit labelled $n-1$ will be called the anchor ancilla or *ank* for short, and will be unrelated to orbitals. It will be used to enforce fermionic statistics via the Jordan Wigner transformation.

So why is it of interest to find a quantum circuit that represents a Trotter approximation for the evolution operator of this Hamiltonian?

Such a quantum circuit is useful in order to estimate the ground state energy of molecules via Kitaev's Phase Estimation Algorithm (PEA). This paper describes the particular circuit used by a Python software package called "my-chemistry", written by R.R.Tucci, and available at GitHub, Ref.[1]. The software can be used in conjunction with "Qubiter", another Python software package available at GitHub, Ref.[2].

A quantum circuit that is very similar to the one presented in this paper has previously been implemented FIRST in Ref.[4] and more recently and exhaustively in the closed source software package called Liqui$|\rangle$ produced by Microsoft, with Dave Wecker as main author.

Here is a super brief, by no means exhaustive review of some of the highlights in the history of this quantum computing approach to chemistry.

The person deserving the lion share of the credit for this method is A. Kitaev, who in 1995, Ref.[3], was the first to propose the PEA. Also very deserving are Trotter for his expansion, and Jordan/Wigner for their transformation.

The first paper to present an actual computer program for calculating the ground state energy of an H_2 molecule using PEA appears to be Ref.[4], by Whitfield, Biamonte and Aspuru-Guzik.

[1] We will also use n to denote the number operator $|1\rangle\langle 1|$. Which of these two n's we are referring to should be clear from context and should not lead to any confusion.

Researcher working for Microsoft applied the method to more complicated molecules and found some very clever optimization methods, such as using the identity $(CNOT)^2 = 1$. Here is their epiphany paper Ref.[5], and here is their most recent paper Ref.[6]. The latter is recommended for what appears to be a very fair and exhaustive list of references of this approach.

Finally, one should mention that Microsoft has several patents on this method, so it is possible that Microsoft will claim in the future that the software described in this paper infringes on one of their patents. Going to the USPTO website and using the query `IN/wecker AND AN/Microsoft`, I located 4 patents Refs. [7][8][9][10] on Liqui$|\rangle$. There might be more pending.

2 Notation and Preliminaries

In this section, we will review briefly some of the more unconventional notation used in this paper. For a more detailed discussion of Tucci's notation, especially its more idiosyncratic aspects, see, for example, Ref.[11].

We will denote the projectors onto the states 0 and 1 by

$$P_1 = |1\rangle\langle 1| = n \tag{3}$$

(n is often referred to as the number operator because it equals 1 if the state has 1 particle and 0 if 0) and

$$P_0 = |0\rangle\langle 0| = 1 - n = \overline{n}\ . \tag{4}$$

For this paper, it is also convenient to define raising and lowering operators given by

$$P_{0|1} = |0\rangle\langle 1| = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} = \sigma_+ \tag{5}$$

and

$$P_{1|0} = |1\rangle\langle 0| = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} = \sigma_-\ . \tag{6}$$

For a vector x, we will denote its reverse by x^R

$$x = (x_0, x_1, \ldots, x_{t-1})\ ,\ \ x^R = (x_{t-1}, \ldots, x_1, x_0)\ . \tag{7}$$

The raising and lowering matrices are 2 dimensional special cases of diagonal and anti-diagonal matrices. So let us define n dimensional diagonal (denoted by ϵ, which stands for "even") and anti-diagonal (denoted by ω, which stands for "odd") by:

$$\epsilon(x) = \begin{bmatrix} x_0 & 0 & \vdots & 0 & 0 \\ 0 & x_1 & \vdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \vdots & x_{t-2} & 0 \\ 0 & 0 & \vdots & 0 & x_{t-1} \end{bmatrix} \tag{8}$$

and

$$\omega(x) = \begin{bmatrix} 0 & 0 & \vdots & 0 & x_{t-1} \\ 0 & 0 & \vdots & x_{t-2} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & x_1 & \vdots & 0 & 0 \\ x_0 & 0 & \vdots & 0 & 0 \end{bmatrix} . \tag{9}$$

We will denote the componentwise product of two vectors of the same length simply by juxtaposition,

$$(xy)_k = x_k y_k \ . \tag{10}$$

Clearly,

$$xy = yx \ , \quad (xy)^R = x^R y^R \ . \tag{11}$$

It's very simple to verify (for example, just use 2 dimensional matrices to check it) that

$$\epsilon(x)\epsilon(y) = \epsilon(xy) \ , \tag{12a}$$

$$\omega(x)\epsilon(y) = \omega(xy) \ , \tag{12b}$$

$$\epsilon(x)\omega(y) = \omega(x^R y) \ , \tag{12c}$$

and

$$\omega(x)\omega(y) = \epsilon(x^R y) \ . \tag{12d}$$

So if the left hand side has $\omega()$ as the second term, the right hand side has a reversal of the first term of the product argument.

Using the linearity of $\epsilon()$ and $\omega()$, we get

$$\begin{aligned} [\epsilon(x) + \omega(y)][\epsilon(a) + \omega(b)] &= \epsilon(xa) + \omega(x^R b) + \omega(ya) + \epsilon(y^R b) && (13) \\ &= \epsilon(xa + y^R b) + \omega(ya + x^R b) \ . && (14) \end{aligned}$$

Corresponding to the usual diagonal identity matrix, one can define an anti-diagonal analogue:

$$I_m = \epsilon(1^m)\ ,\ \ \hat{I}_m = \omega(1^m)\ . \tag{15}$$

When dealing with qubits, one often encounters matrices whose dimension is a power of 2. Note that in such cases, $\hat{I}_{2^m}$ can be expressed as an m-fold tensor product of the Pauli matrix σ_X

$$\hat{I}_{2^m} = \sigma_X^{\otimes m}\ . \tag{16}$$

The effect of pre or post multiplying a matrix by $\hat{I}$ is given by

$$\hat{I}\epsilon(x) = \omega(x)\ ,\ \ \hat{I}\omega(x) = \epsilon(x)\ , \tag{17}$$

$$\epsilon(x)\hat{I} = \omega(x^R)\ ,\ \ \omega(x)\hat{I} = \epsilon(x^R)\ . \tag{18}$$

3 Jordan Wigner Tails

As previous workers in this field, we will use the Jordan Wigner (JW) Transformation to enforce the fermionic anticommutators Eqs.(2). This transformation adds a "tail" to a local raising or lower operator in order to make it non-local. To deal with the book-keeping of JW tails, we will use dumbbell gates.

We will often use strings of σ_Z gates located at qubits $k_1, k_2, \ldots, k_t$. The following shorthand notation will be helpful in those cases:

$$\sigma_Z(k_1, k_2, \ldots, k_t) = \sigma_Z(k_1)\sigma_Z(k_2)\ldots\sigma_Z(k_t)\ . \tag{19}$$

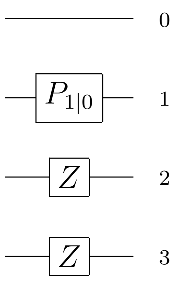
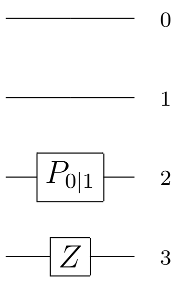
The Jordan Wigner (JW) transformation is defined by

$$a(k) = P_{0|1}(k)\sigma_Z(k+1, k+2, \ldots, n-1)\ , \tag{20a}$$

and

$$a^\dagger(k) = P_{1|0}(k)\sigma_Z(k+1, k+2, \ldots, n-1)\ . \tag{20b}$$

For example,

$a^\dagger(1) =$  , $a(2) =$  . (21)

We will refer to the strings of σ_Z gates in Eqs.(20) as JW tails.

It is easy to check that if $a(k)$ and $a^\dagger(k)$ are defined by Eqs.(20) for all qubits k except the last one $ank = n - 1$, which we will call the anchor ancilla, then the anti-commutator relations Eqs.(2) are satisfied for all k except the anchor ancilla.

To deal with the book-keeping of JW tails, we will use dumbbell gates. Let us define those and explore their properties next.

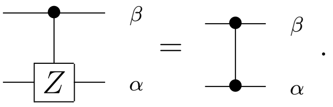
Recall that

$$\sigma_Z = (-1)^n \ . \tag{22}$$

A dumbbell with endpoints α, β is defined as the following gate

$$\sigma_Z(\alpha)^{n(\beta)} = (-1)^{n(\alpha)n(\beta)} \tag{23}$$

for two distinct qubits α, β. We will express Eq.(23) graphically by



(24)

Thus, a dumbbell is just a controlled σ_Z. Dumbbells are symmetric under exchange of their two endpoint α and β.

Recall that the 2-dim Hadamard matrix defined by

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{25}$$

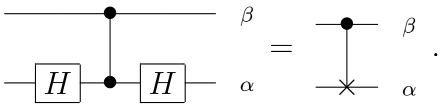
satisfies

$$H^2 = H \ , \quad H\sigma_X H = \sigma_Z \ . \tag{26}$$

Using the properties of H, we get

$$H(\alpha)\sigma_Z(\alpha)^{n(\beta)}H(\alpha) = \sigma_X(\alpha)^{n(\beta)} = CNOT(\beta \to \alpha) \ . \tag{27}$$

Hence, a dumbbell is closely related to a CNOT, which itself is just a controlled σ_X. One can apply a Hadamard similarity transformation to the target of a CNOT gate or a dumbbell gate to get the other gate. Graphically, Eq.(27) is expressed by

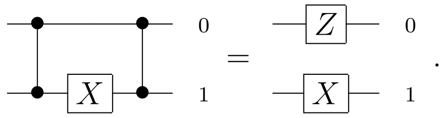


(28)

Note that the effect of a similarity transformation by a matching pair of dumbbells on an in-between σ_X at one endpoint is

$$\begin{aligned} \sigma_Z(1)^{n(0)}\sigma_X(1)\sigma_Z(1)^{n(0)} &= \sigma_X(1)(-1)^{n(0)} \qquad (29) \\ &= \sigma_X(1)\sigma_Z(0) \ , \qquad (30) \end{aligned}$$

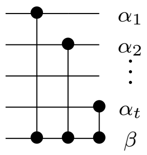
which expressed graphically is



. (31)

Later on in this paper, we will apply, one after the other, several of these similarity transformations of matching dumbbells. The following notation will then be useful:

$$db\begin{pmatrix} \alpha_1,\alpha_2,\ldots,\alpha_t \\ \beta \end{pmatrix} =$$



, (32)

where $\alpha_1, \alpha_2, \ldots, \alpha_t, \beta$ are distinct qubits. We will refer to the circuit Eq.(32) as a stairs or cascade of dumbbells.

4 Trotter Approximation

In this section, we discuss the Trotter approximation of an evolution operator for the Hamiltonian of Eq.(1). In the process, we introduce the concept of spokes, and Qubiter's classes CombisInterlacer and DumbbellNeutralizer.

The Trotter approximation of an evolution operator e^{-itH} is given by

$$e^{-iHt} = \left\{e^{-iH_1 dt} e^{-iH_2 dt} \ldots e^{-iH_m dt}\right\}^{t/dt} . \tag{33}$$

We will refer to t/dt as the number of Trotter cycles. We will refer to each H_r as a **spoke** (of the wheel that is being cycled).

Combis is short for combinations. A combi is a set of integers from $\{0, 1, \ldots, n-2\}$ (Not $n-1$. Recall that we reserve $ank = n-1$ for the ancilla anchor that we use to enforce fermionic statistics). A bunch of combis will be said to be **interlaced** if they are non-overlapping.

A spoke has a core and a sheath.

The **core** is a bunch of interlaced combis.

The **sheath** of a spoke is a unitary transformation V (a.k.a. as the left endcap) on left side of the core and the Hermitian conjugate of V (a.k.a. the right endcap) on the right side of the core.

We choose the combis in the bunch of each spoke to be interlaced, because then all the combis of the bunch can be performed in parallel. Unlike the combis in the core bunch, the operations in the sheath of the spoke cannot be parallelized in any obvious way. However, some gates in the sheaths of adjacent spokes will cancel out. Qubiter's class `DumbBellNeutralizer` tries do such cancellations. This class reads an English file to identify clusters of adjacent dumbbells. Then it looks at each cluster and tries to cancel (neutralize) pairs of dumbbells with the same endpoints. Then it writes a new English file which is identical to the original one, except that all neutralized dumbbells are changed from a SIGZ to a NOTA line.

The purpose of Qubiter's class `CombisInterlacer` is to, given an input list, call it L, of combis of arbitrary length, to redistribute the elements of L into interlaced bunches of combis. The class goes through each combi c in L and checks to see if c fits into any of the bunches that already exist. If c fits inside an already existing bunch, the class inserts c into that bunch. If c doesn't fit inside any of the existing bunches, the class creates a new bunch and places c inside that one.

Each combi corresponds to one of the basic subcircuits that we introduce in the next section. Each subcircuit is labelled by a set of integers which constitutes its combi signature.

5 Basic Subcircuits

The Hamiltonian given by Eq.(20) can be decomposed into four subterms

$$H = H_{diag} + H_{1bit} + H_{2bit} + H_{4bit} \ . \tag{34}$$

In this section we define each of these four subterms. Each H subterm leads to a distinctive quantum circuit which we will also discuss in this section.

5.1 Diagonal Subcircuits

We define H_{diag} by

$$H_{diag} = \sum_k h(k)n(k) + \sum_{k_1,k_2} h(k_1,k_2)n(k_1)n(k_2) \ . \tag{35}$$

Let

$$J_{diag,1}(k) = e^{i\theta_1 n(k)} \tag{36}$$

and

$$J_{diag,2}(k_1,k_2) = e^{i\theta_2 n(k_1)n(k_2)} \tag{37}$$

for some real numbers θ_1 and θ_2. $J_{diag,1}$ and $J_{diag,2}$ can be considered elementary gates already, or else they can be decomposed into a global phase factor, some Z rotations and some controlled Z rotations as follows. Recall that $n = \frac{1-\sigma_Z}{2}$ so

$$e^{i\theta n(k)} = e^{i\frac{\theta}{2}} e^{-i\frac{\theta}{2}\sigma_Z(k)} \tag{38}$$

and

$$\begin{aligned} e^{i\theta n(k_1)n(k_2)} &= e^{i\frac{\theta}{2}n(k_2)}e^{-i\frac{\theta}{2}\sigma_Z(k_1)n(k_2)} && (39) \\ &= e^{i\frac{\theta}{4}}e^{-i\frac{\theta}{4}\sigma_Z(k_2)}e^{-i\frac{\theta}{2}\sigma_Z(k_1)n(k_2)} \; . && (40) \end{aligned}$$

5.2 2 Qubit Subcircuits

We define H_{2bit} by

$$H_{2bit} = \sum_{k_1<k_2} h(k_1,k_2)[a^\dagger(k_1)a(k_2) + h.c.] \; . \tag{41}$$

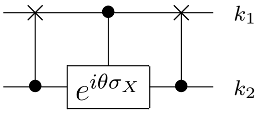
Let

$$J_2(k_1,k_2) = e^{i\theta[a^\dagger(k_1)a(k_2)+h.c.]} \tag{42}$$

for some real number θ. Then we claim

Claim 1

$$J_2(k_1,k_2) = (db)\sigma_X(ank)$$



$$(db)^\dagger \; , \tag{43}$$

where

$$db = db\begin{pmatrix} k_1+1,k_1+2,\ldots,k_2-1 \\ ank \end{pmatrix} \; . \tag{44}$$

proof:

Using the JW transformation, one gets

$$\begin{aligned} J_2(k_1,k_2) &= e^{i\theta[P_{1|0}(k_1)\sigma_Z(k_1+1,k_1+2,\ldots,k_2-1)P_{0|1}(k_2)+h.c.]} && (45) \\ &= (db)\sigma_X(ank)\underbrace{e^{i\theta[P_{1|0}(k_1)P_{0|1}(k_2)+h.c.]}}_{J_2'(k_1,k_2)}(db)^\dagger \; . && (46) \end{aligned}$$

Note that both $J_2(k_1,k_2)$ and $J_2'(k_1,k_2)$ are symmetric under exchange of k_1 and k_2.

Now define

$$\Gamma = P_{1|0}(1)P_{0|1}(0) \tag{47}$$

so that

$$J_2'(0,1) = e^{i\theta(\Gamma+h.c.)} \; . \tag{48}$$

Γ can be expressed as an explicit 4 dim matrix as follows:

$$\Gamma = \omega(1,0) \otimes \omega(0,1) = \omega(0,1,0^2) \ . \tag{49}$$

Thus,

$$\Gamma^\dagger = \omega(0^2,1,0) \ . \tag{50}$$

If we define x by

$$x = (0,1^2,0) \ , \tag{51}$$

then

$$J_2'(0,1) = e^{i\theta\omega(x)} \ . \tag{52}$$

Since

$$[\omega(x)]^2 = \epsilon(x) \ , \tag{53}$$

Eq.(52) can be expanded into a Taylor series and then re-summed to obtain

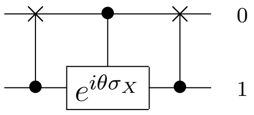
$$J_2'(0,1) = 1 + (-1+C)\epsilon(x) + iS\omega(x) \ , \tag{54}$$

where $C = \cos(\theta), S = \sin(\theta)$. Eq.(54) can now be expressed as an explicit 4 dim matrix, and that matrix as a quantum circuit, as follows:

$$J_2'(0,1) \quad = \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C & iS & 0 \\ 0 & iS & C & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{55}$$

$$= \quad (I_2 \oplus \sigma_X) \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & C & 0 & iS \\ \hline 0 & 0 & 1 & 0 \\ 0 & iS & 0 & C \end{array} \right] (I_2 \oplus \sigma_X) \tag{56}$$

$=$



. (57)

QED

5.3 3 Qubit Subcircuits

In this section, let $k = (k_1, k_2, k_3)$. We define H_{3bit} by

$$H_{3bit} = \sum_{k_1<k_2} \sum_{k_3 \neq k_1,k_2} h(k)[a^\dagger(k_1)a(k_2) + h.c.]n(k_3) \ . \tag{58}$$

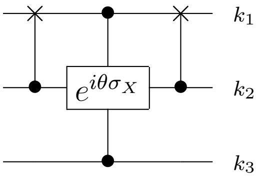
Let

$$J_3(k) = e^{i\theta[a^\dagger(k_1)a(k_2)+h.c.]n(k_3)} \tag{59}$$

for some real number θ. Then we claim

Claim 2

$$J_3(k) = (db)\sigma_X(ank)$$



$$(db)^\dagger \; , \tag{60}$$

where

$$db = db\begin{pmatrix} k_1+1,k_1+2,\ldots,k_2-1 \\ ank \end{pmatrix} \; . \tag{61}$$

proof:
Obvious from Section 5.2 on 2-qubit subcircuits.
QED

5.4 4 Qubit Subcircuits

In this section, let $k = (k_1, k_2, k_3, k_4)$. We define H_{4bit} by

$$H_{4bit} = \sum_{\substack{k \\ \text{all } k_j \text{ different}}} h(k)[a^\dagger(k_1)a^\dagger(k_2)a(k_3)a(k_4) + h.c.] \; . \tag{62}$$

If we want the k_j to be in increasing order, then the daggers in $a_3^\dagger a_2^\dagger a_1 a_0 + a_0^\dagger a_1^\dagger a_2 a_3$ can no longer be attached to only the first two a's. A moment's reflection reveals that there can be exactly 3 ways of attaching those daggers. Hence, it becomes clear that H_{4bit} defined by Eq.(62) can also be expressed as

$$\begin{aligned} H_{4bit} \;&=\; \sum_{k_1<k_2<k_3<k_4} h_a(k)[a^\dagger(k_1)a^\dagger(k_2)a(k_3)a(k_4) + h.c.] \\ &\quad + \sum_{k_1<k_2<k_3<k_4} h_b(k)[a^\dagger(k_1)a(k_2)a^\dagger(k_3)a(k_4) + h.c.] \\ &\quad + \sum_{k_1<k_2<k_3<k_4} h_c(k)[a^\dagger(k_1)a(k_2)a(k_3)a^\dagger(k_4) + h.c.] \; . \end{aligned} \tag{63}$$

This second expression Eq.(63) for H_{4bit} is more convenient for our future analysis because it deals only with k_j that are in increasing order, whereas the k_j in the first expression Eq.(62) are all distinct but not necessarily in increasing order.

Let

$$J_{4a}(k) = e^{i\theta_a[a^\dagger(k_1)a^\dagger(k_2)a(k_3)a(k_4)+h.c.]} \ , \tag{64}$$

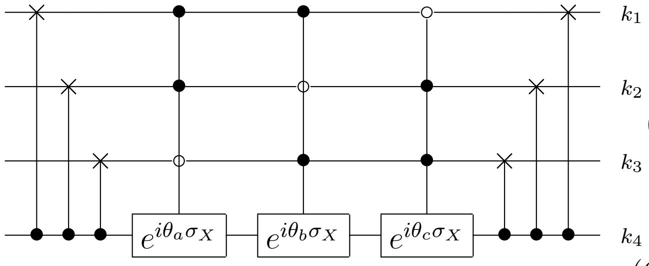
$$\begin{aligned} J_{4b}(k) &= e^{i\theta_b[a^\dagger(k_1)a(k_2)a^\dagger(k_3)a(k_4)+h.c.]} && (65)\\ &= Swap(k_2,k_3)J_{4a}(k)|_{\theta_a\to\theta_b}Swap(k_2,k_3)\ , && (66)\end{aligned}$$

$$\begin{aligned} J_{4c}(k) &= e^{i\theta_c[a^\dagger(k_1)a(k_2)a(k_3)a^\dagger(k_4)+h.c.]} && (67)\\ &= Swap(k_2,k_4)J_{4a}(k)|_{\theta_a\to\theta_c}Swap(k_2,k_4) && (68)\end{aligned}$$

for some real numbers $\theta_a, \theta_b, \theta_c$. Then we claim that

Claim 3

$J_{4a}(k)J_{4b}(k)J_{4c}(k) = (db)\sigma_X(ank)$



$(db)^\dagger$

(69)

where

$$db = db\begin{pmatrix} k_1{+}1,k_1{+}2,\ldots,k_2{-}1 \\ ank\end{pmatrix} db\begin{pmatrix} k_3{+}1,k_3{+}2,\ldots,k_4{-}1 \\ ank\end{pmatrix} \ . \tag{70}$$

proof:

Using the JW transformation, one gets

$$\begin{aligned} J_{4a}(k) &= e^{i\theta[P_{1|0}(k_1)\sigma_Z(k_1+1,k_1+2,\ldots,k_2-1)P_{1|0}(k_2)P_{0|1}(k_3)\sigma_Z(k_3+1,k_3+2,\ldots,k_4-1)P_{0|1}(k_4)+h.c.]} && (71)\\ &= (db)\sigma_X(ank)\underbrace{e^{i\theta[P_{1|0}(k_1)P_{1|0}(k_2)P_{0|1}(k_3)P_{0|1}(k_4)+h.c.]}}_{J'_{4a}(k)}(db)^\dagger\ . && (72)\end{aligned}$$

Note that both $J_{4a}(k)$ and $J'_{4a}(k)$ are symmetric under exchanges of k and k^R. Now define

$$\Gamma = P_{1|0}(3)P_{1|0}(2)P_{0|1}(1)P_{0|1}(0) \tag{73}$$

so that

$$J'_{4a}(3,2,1,0) = \exp\left[i\theta(\Gamma + h.c.)\right] \ . \tag{74}$$

Γ can be expressed as an explicit 16 dim matrix as follows:

$$\begin{aligned}\Gamma &= \omega(1,0)\otimes\omega(1,0)\otimes\omega(0,1)\otimes\omega(0,1) \qquad (75)\\ &= \omega(0^3,1,0^{12})\ . \qquad (76)\end{aligned}$$

Thus,

$$\Gamma^\dagger = \omega(0^{12},1,0^3)\ . \qquad (77)$$

If we define x by

$$x = (0^3,1,0^8,1,0^3)\ , \qquad (78)$$

then

$$J'_{4a}(3,2,1,0) = e^{i\theta\omega(x)}\ . \qquad (79)$$

Since

$$[\omega(x)]^2 = \epsilon(x)\ , \qquad (80)$$

Eq.(79) can be expanded into a Taylor series and then re-summed to obtain

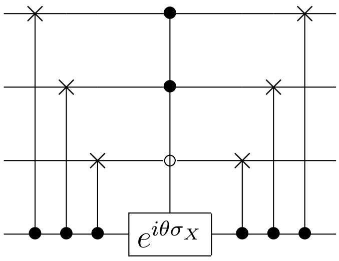
$$J'_{4a}(3,2,1,0) = 1 + (-1+C)\epsilon(x) + iS\omega(x)\ , \qquad (81)$$

where $C=\cos(\theta), S=\sin(\theta)$. Eq.(81) can now be expressed as an explicit 16 dim matrix, and that matrix as a quantum circuit, as follows:

$$J'_{4a}(3,2,1,0) = \left[\begin{array}{c|c|c|c} \epsilon(1^3,C) & & & \omega(iS,0^3) \\ \hline & I_4 & 0_4 & \\ \hline & 0_4 & I_4 & \\ \hline \omega(0^3,iS) & & & \epsilon(C,1^3) \end{array}\right] \qquad (82)$$

$$= (I_8 \oplus \hat{I}_8) \left[\begin{array}{c|c|c|c} \epsilon(1^3,C) & & \epsilon(0^3,iS) & \\ \hline & I_4 & & 0_4 \\ \hline \epsilon(0^3,iS) & & I_4 & \\ \hline & 0_4 & & \epsilon(1^3,C) \end{array}\right] (I_8 \oplus \hat{I}_8) \qquad (83)$$

$=$



. (84)

QED

References

[1] https://github.com/rrtucci/my-chemistry

[2] https://github.com/artiste-qb-net/qubiter

[3] "Quantum measurements and the Abelian Stabilizer Problem", by A.Yu.Kitaev, http://arxiv.org/abs/quant-ph/9511026

[4] "Simulation of Electronic Structure Hamiltonians Using Quantum Computers", J. D. Whitfield, J. Biamonte, A. Aspuru-Guzik, https://arxiv.org/abs/1001.3855

[5] "Improving Quantum Algorithms for Quantum Chemistry", by M. B. Hastings, D. Wecker, B. Bauer, M. Troyer https://arxiv.org/abs/1403.1539

[6] "Elucidating Reaction Mechanisms on Quantum Computers", by M. Reiher, N. Wiebe, K. M Svore, D. Wecker, M. Troyer, https://arxiv.org/abs/1605.03590

[7] "Quantum gate optimizations ", Dave Wecker (assignee Microsoft) US Patent 9,064,067

[8] "Optimizing quantum simulations by intelligent permutation ", Dave Wecker, (assignee Microsoft) US Patent 8,972,237

[9] "Language integration via function redirection ", Dave Wecker, (assignee Microsoft) US Patent 9,292,304

[10] "Quantum annealing simulator ", M. Troyer, D. Wecker, B. Bauer (assignee Microsoft) US Patent 9,152,746

[11] R.R. Tucci, "QC Paulinesia", http://arxiv.org/abs/quant-ph/0407215