Jordan-Wigner Transform: Simulation of Fermionic Creation and Annihilation Operators in a Quantum Computer

by José Luis Gómez-Muñoz
http://homepage.cem.itesm.mx/lgomez/quantum/
jose.luis.gomez@itesm.mx

Introduction

The Jordan-Wigner transform allows us to take a system of interacting Fermions (second quantization, occupation notation), and map it into an equivalent model of interacting spins, which can then, in principle, be simulated using standard techniques in a quantum computer. This enables to use quantum computers to simulate systems of interacting Fermions, as explained by Michael A. Nielsen, "The Fermionic canonical relations and the Jordan-Wigner transform", 2005, published in his personal blog. Here is a copy of that document:


NOTE:

This could be an efficient implementation of fermionic operators in a quantum computer. However, in this "normal" computer, these calculations are very slow. If you need faster calculations with these operators, and you do not care about their possible implementation in a quantum computer, these documents might be useful for you:


Load the Package

First load the Quantum`Computing` package. Write:
Needs["Quantum`Computing`"];

then press at the same time the keys \[\text{shift}-\text{enter}\] to evaluate. Mathematica will load the package.
In order to use the keyboard to enter quantum objects write:
SetComputingAliases[];
then press at the same time the keys $\texttt{[SET]-\textbar}$ to evaluate. The semicolon prevents Mathematica from printing the help message. Remember that SetComputingAliases[ ] must be evaluated again in each new notebook:

\begin{verbatim}
SetComputingAliases[];
\end{verbatim}

### Annihilation Operator in terms of Pauli Operators

This is the definition of a fermionic annihilation operator in terms of Pauli operators and the operator $| \ 0 \rangle \langle 1 \ |$. Pauli operators $\sigma_{x,k}$ produce the correct sign when acting on a ket (or, equivalent, produce the correct commutation relations) and the operator $| \ 0 \rangle \langle 1 \ |$ annihilates the corresponding fermion (qubit).

\[
n\text{fermions} = 3;
a[j_\_] := \left(\bigotimes_{k=1}^{j-1} \sigma_{z,k}\right) \otimes \left(-| 0 \rangle \langle j_\_ | \right) \otimes \left(\bigotimes_{m=j+1}^{\text{fermions}} \sigma_{\sigma_{y,m}} \right)
\]

Each qubit represents a fermionic state. A qubit value of "zero" means that the state is not occupied; "one" means the fermionic state is occupied. Here the the annihilation operator $a[2]$ annihilates the second fermionic state, $1_2 \rightarrow 0_2$.

\begin{verbatim}
QuantumEvaluate[a[2] \cdot | 1_1, 1_2, 0_3 | ]
\end{verbatim}

\[
| 1_1, 0_2, 0_3 |
\]

If the annihilation operator is applied to a ket with the corresponding fermionic state equal to "zero" (nonoccupied), then the ket is "destroyed":

\begin{verbatim}
QuantumEvaluate[a[2] \cdot | 1_1, 0_2, 0_3 | ]
\end{verbatim}

\[
0
\]

Here we can see the action of the annihilation operator $a[2]$ in any possible basis ket for a three fermionic-states system. Notice the negative signs, which depend on the occupation of the previous fermionic states:

\begin{verbatim}
QuantumTableForm[a[2]]
\end{verbatim}

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0_1, 0_2, 0_3</td>
</tr>
<tr>
<td>1</td>
<td>0_1, 0_2, 1_3</td>
</tr>
<tr>
<td>2</td>
<td>0_1, 1_2, 0_3</td>
</tr>
<tr>
<td>3</td>
<td>0_1, 1_2, 1_3</td>
</tr>
<tr>
<td>4</td>
<td>1_1, 0_2, 0_3</td>
</tr>
<tr>
<td>5</td>
<td>1_1, 0_2, 1_3</td>
</tr>
<tr>
<td>6</td>
<td>1_1, 1_2, 0_3</td>
</tr>
<tr>
<td>7</td>
<td>1_1, 1_2, 1_3</td>
</tr>
</tbody>
</table>
Here we can see the action of the annihilation operator $a[3]$ in any possible basis ket for a three fermionic-states system. Notice the negative signs, which depend on the occupation of the previous fermionic states:

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>$</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>$</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>$</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>$</td>
</tr>
</tbody>
</table>

**Creation Operator in terms of Pauli Operators**

This is the fermionic annihilation operator that was defined above:

$$a[j] = -\bigotimes_{k=1}^{j} \sigma_{z,k} \cdot |0\rangle \cdot \bigotimes_{m=j+1}^{3} \sigma_{z,m}$$

The fermionic creation operator is the hermitian conjugate of the annihilation operator. Press [ESC]her[ESC] for the hermitian-conjugate template, or use the "Quantum Notation" Palette, in the menu Palettes of Mathematica:

$$a[j]^\dagger = -\bigotimes_{m=j+1}^{3} \sigma_{z,m}^\dagger \cdot |0\rangle \cdot \bigotimes_{k=1}^{j} \sigma_{z,k}^\dagger$$

Each qubit represents a fermionic state. A qubit value of "zero" means that the state is not occupied; "one" means the fermionic state is occupied. Here the the creation operator $a[2]^\dagger$ "creates" a fermion in the second fermionic state, $0_2 \rightarrow 1_2$.

Notice the negative sign. This sign depends on the occupation of the previous fermionic states:

$$\text{QuantumEvaluate}[a[2]^\dagger \cdot |0, 0, 0\rangle]$$

$$- |0, 1, 0\rangle$$

If the creation operator is applied to a ket with the corresponding fermionic state equal to "one" (occupied), then the ket is "destroyed":

$$\text{QuantumEvaluate}[a[3]^\dagger \cdot |0, 0, 1\rangle]$$

$$- |0, 1, 0\rangle$$
QuantumEvaluate\[a[2]^\dagger \cdot |0_1, 1_2, 0_3]\]

Here we can see the action of the creation operator $a[2]^\dagger$ in any possible basis ket for a three fermionic-states system. Notice the negative signs, which depend on the occupation of the previous fermionic states:

QuantumTableForm\[a[2]^\dagger\]

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>0_1, 0_2, 0_3\rangle$</td>
</tr>
<tr>
<td>$</td>
<td>0_1, 0_2, 1_3\rangle$</td>
</tr>
<tr>
<td>$</td>
<td>0_1, 1_2, 0_3\rangle$</td>
</tr>
<tr>
<td>$</td>
<td>0_1, 1_2, 1_3\rangle$</td>
</tr>
<tr>
<td>$</td>
<td>1_1, 0_2, 0_3\rangle$</td>
</tr>
<tr>
<td>$</td>
<td>1_1, 0_2, 1_3\rangle$</td>
</tr>
<tr>
<td>$</td>
<td>1_1, 1_2, 0_3\rangle$</td>
</tr>
<tr>
<td>$</td>
<td>1_1, 1_2, 1_3\rangle$</td>
</tr>
</tbody>
</table>

Here we can see the action of the creation operator $a[3]^\dagger$ in any possible basis ket for a three fermionic-states system. Notice the negative signs, which depend on the occupation of the previous fermionic states:

QuantumTableForm\[a[3]^\dagger\]

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>0_1, 0_2, 0_3\rangle$</td>
</tr>
<tr>
<td>$</td>
<td>0_1, 0_2, 1_3\rangle$</td>
</tr>
<tr>
<td>$</td>
<td>0_1, 1_2, 0_3\rangle$</td>
</tr>
<tr>
<td>$</td>
<td>0_1, 1_2, 1_3\rangle$</td>
</tr>
<tr>
<td>$</td>
<td>1_1, 0_2, 0_3\rangle$</td>
</tr>
<tr>
<td>$</td>
<td>1_1, 0_2, 1_3\rangle$</td>
</tr>
<tr>
<td>$</td>
<td>1_1, 1_2, 0_3\rangle$</td>
</tr>
<tr>
<td>$</td>
<td>1_1, 1_2, 1_3\rangle$</td>
</tr>
</tbody>
</table>

Commutation Relations

This is the fermionic annihilation operator that was defined above. TraditionalForm gives a format closer to the format used in papers and textbooks:

TraditionalForm\[a[j]\]

$$\left\langle j^{-1} \otimes (\sigma_z^j) \right|0\left\langle 1 \otimes (\sigma_z^0) \right|_{\text{av}/j=1}$$
The anticommutator of annihilation and creation operators of the same state give the identity. Press \[ESC\]anti[ESC] for the anticommutator template and \[ESC\]her[ESC] for the hermitian conjugate, or use the "Quantum Notation" Palette, in the menu Palettes of Mathematica. This calculation is too slow, see the Note at the beginning of this document:

\[
PauliExpand[[a[2], a[2]']]
\]

\[\sigma_{o,1} \cdot \sigma_{o,2} \cdot \sigma_{o,3}\]

The PauliIdentities->False option transforms the identity operators to the number one. This calculation is too slow, see the Note at the beginning of this document:

\[
PauliExpand[[a[2], a[2]'], PauliIdentities \to False]
\]

\[
1
\]

Anticommutator of annihilation and creation in different fermionic states:

\[
PauliExpand[[a[2], a[3]']]
\]

\[
0
\]

Annihilate two times destroys any ket, therefore this anticommutator is zero:

\[
PauliExpand[[a[2], a[2]]]
\]

\[
0
\]

Annihilate in different states does not destroy any ket, however the anticommutator is zero, see below the True-tables to understand this result:

\[
PauliExpand[[a[2], a[3]]]
\]

\[
0
\]

Here we see why the previous anticommutator \[[a[2], a[3]']\] is zero: \(a[3] a[2]\) gives kets with opposite sign to those given by \(a[2] a[3]\):
Special Products of Annihilation and Creation Operators

This is the fermionic annihilation operator that was defined above. TraditionalForm gives a format closer to the format used in papers and textbooks:

$$\text{TraditionalForm}[a[j]]$$

\[ \otimes_{i=0}^{j-1} (\sigma^z_i) |0\rangle \langle 1| \otimes_{m=j+1}^{l} (\sigma^d_m) \]

Next expression evaluates to $\sigma_{z,0} \cdot \sigma_{z,1} \cdot \sigma_{z,2}$ (times the identities in the other qubits/fermionic-states). This can be useful to write Hamiltonians, see the last two pages of http://homepage.cem.itesm.mx/lgomez/quantum/NielsenJordanWigner.pdf

$$\text{PauliExpand}[a[2]\cdot a[2]^\dagger - a[2]^\dagger \cdot a[2]]$$

\[ \sigma_{0,\hat{1}} \cdot \sigma_{z,2} \cdot \sigma_{0,\hat{3}} \]

The PauliIdentites->False option transforms the identity operators to the number one. **This calculation is too slow, see the Note at the beginning of this document:**

$$\text{PauliExpand}[a[2]\cdot a[2]^\dagger - a[2]^\dagger \cdot a[2], \text{PauliIdentites} \to \text{False}]$$

\[ \sigma_{z,0} \]

Next expression evaluates to $\sigma_{x,\hat{1}} \cdot \sigma_{x,2}$ (times the identities in the other qubits/fermionic-states). This can be useful to write Hamiltonians, see the last two pages of http://homepage.cem.itesm.mx/lgomez/quantum/NielsenJordanWigner.pdf

$$\text{PauliExpand}[(a[1]^\dagger - a[1]) \cdot (a[2]^\dagger + a[2])]$$

\[ \sigma_{x,\hat{1}} \cdot \sigma_{x,2} \cdot \sigma_{0,\hat{3}} \]
The PauliIdentities->False option transforms the identity operators to the number one. This calculation is too slow, see the Note at the beginning of this document:

```
PauliExpand[(a[1]^† - a[1]) · (a[2]^† + a[2]), PauliIdentities -> False]
```

Next expression can be useful to write Hamiltonians, see the last two pages of http://homepage.cem.itesm.mx/lgomez/quantum/NielsenJordanWigner.pdf

```
PauliExpand[(a[1]^† + a[1]) · (a[2]^† - a[2])]
```

The PauliIdentities->False option transforms the identity operators to the number one. This calculation is too slow, see the Note at the beginning of this document:

```
PauliExpand[(a[1]^† - a[1]) · (a[2]^† - a[2]), PauliIdentities -> False]
```

Next expression can be useful to write Hamiltonians, see the last two pages of http://homepage.cem.itesm.mx/lgomez/quantum/NielsenJordanWigner.pdf

```
PauliExpand[(a[1]^† - a[1]) · (a[2]^† + a[2])]
```

The PauliIdentities->False option transforms the identity operators to the number one. This calculation is too slow, see the Note at the beginning of this document:

```
PauliExpand[(a[1]^† + a[1]) · (a[2]^† - a[2])]
```

Next expression can be useful to write Hamiltonians, see the last two pages of http://homepage.cem.itesm.mx/lgomez/quantum/NielsenJordanWigner.pdf

```
PauliExpand[(a[1]^† + a[1]) · (a[2]^† + a[2])]
```

TraditionalForm[] gives a format closer to the format of papers and textbooks:
\begin{verbatim}
TraditionalForm[PauliExpand[(a[1]^d + a[1]) \cdot (a[2]^d + a[2])]]

-i \sigma_1^\mathcal{Y} \sigma_2^\mathcal{X} \sigma_3^0
\end{verbatim}

TeXForm can be used to generate expression for papers and books

\begin{verbatim}
TeXForm[TraditionalForm[PauliExpand[(a[1]^d + a[1]) \cdot (a[2]^d + a[2])]]]

-i \texttt{\sigma}_1^{\mathcal{Y}} \texttt{\sigma}_2^{\mathcal{X}} \texttt{\sigma}_3^0
\end{verbatim}

**NOTE:**

This **could** be an efficient implementation of fermionic operators in a **quantum computer**. However, in this "normal" computer, these calculations are very slow. If you need faster calculations with these operators, and you do not care about their possible implementation in a quantum computer, these documents might be useful for you:


by José Luis Gómez-Muñoz
http://homepage.cem.itesm.mx/lgomez/quantum/
jose.luis.gomez@itesm.mx