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:

http://homepage.cem.itesm.mx/lgomez/quantum/NielsenJordanWigner.pdf

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:

Algebra of operators and commutators. Mathematica file: http://homepage.cem.itesm.mx/lgomez/quantum/v7algebra.nb

Algebra of operators and commutators. PDF file: http://homepage.cem.itesm.mx/lgomez/quantum/v7algebra.pdf

Load the Package

First load the Quantum`Computing` package. Write:
Needs["Quantum`Computing`"];
then press at the same time the keys SHIFT-ENTER to evaluate. *Mathematica* will load the package.

```
Needs["Quantum`Computing`"]
```

```
Quantum`Computing` Version 2.2.0. (June 2010)
A Mathematica package for Quantum Computing
  in Dirac bra-ket notation and plotting of quantum circuits
by José Luis Gómez-Muñoz

Execute SetComputingAliases[] in order to use
  the keyboard to enter quantum objects in Dirac's notation
SetComputingAliases[] must be executed again in each new notebook that is created
```

In order to use the keyboard to enter quantum objects write:
SetComputingAliases[ ];
then press at the same time the keys SHIFT-ENTER to evaluate. The semicolon prevents *Mathematica* from printing the help message. Remember that SetComputingAliases[ ] must be evaluated again in each new notebook:

```
SetComputingAliases[];
```

Annihilation Operator in terms of Pauli Operators

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

```
nfermions = 3;
```

$$a[j\_] := \left(\bigotimes_{k=1}^{j-1} \sigma_{\mathcal{Z},\hat{k}}\right) \otimes \left(-\left|0_{\hat{j}}\right\rangle \cdot \left\langle 1_{\hat{j}}\right|\right) \otimes \left(\bigotimes_{m=j+1}^{\text{nfermions}} \sigma_{0,\hat{m}}\right)$$

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, $1_{\hat{2}}\to 0_{\hat{2}}$.

$$\text{QuantumEvaluate}\left[a[2]\cdot\left|1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\right\rangle\right]$$

$$\left|1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\right\rangle$$

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

$$\text{QuantumEvaluate}\left[a[2]\cdot\left|1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\right\rangle\right]$$

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:

```
QuantumTableForm[a[2]]
```

	Input	Output
0	$\left\|0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\right\rangle$	0
1	$\left\|0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\right\rangle$	0
2	$\left\|0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\right\rangle$	$-\left\|0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\right\rangle$
3	$\left\|0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\right\rangle$	$-\left\|0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\right\rangle$
4	$\left\|1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\right\rangle$	0
5	$\left\|1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\right\rangle$	0
6	$\left\|1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\right\rangle$	$\left\|1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\right\rangle$
7	$\left\|1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\right\rangle$	$\left\|1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\right\rangle$

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:

```
QuantumTableForm[a[3]]
```

	Input	Output
0	$\mid 0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$	0
1	$\mid 0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle$	$-\mid 0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$
2	$\mid 0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle$	0
3	$\mid 0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle$	$\mid 0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle$
4	$\mid 1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$	0
5	$\mid 1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle$	$\mid 1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$
6	$\mid 1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle$	0
7	$\mid 1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle$	$-\mid 1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle$

Creation Operator in terms of Pauli Operators

This is the fermionic annihilation operator that was defined above:

```
a[j]
```

$$-\bigotimes_{k=1}^{-1+j} \sigma_{Z,\hat{k}} \cdot \mid 0_{\hat{j}}\rangle \cdot \langle 1_{\hat{j}} \mid \cdot \bigotimes_{m=j+1}^{3} \sigma_{0,\hat{m}}$$

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

```
a[j]†
```

$$-\left(\bigotimes_{m=j+1}^{3} \sigma_{0,\hat{m}}\right)^{\dagger} \cdot \mid 1_{\hat{j}}\rangle \cdot \langle 0_{\hat{j}} \mid \cdot \left(\bigotimes_{k=1}^{-1+j} \sigma_{Z,\hat{k}}\right)^{\dagger}$$

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, $0_{\hat{2}} \to 1_{\hat{2}}$. Notice the negative sign. This sign depends on the occupation of the previous fermionic states:

```
QuantumEvaluate[a[2]† · | 0₁̂, 0₂̂, 0₃̂〉]
```

$$-\mid 0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle$$

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

```
QuantumEvaluate[a[2]† · |0_1̂, 1_2̂, 0_3̂⟩]
```

```
0
```

Here we can see the action of the creation 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:

```
QuantumTableForm[a[2]†]
```

	Input	Output
0	$\vert 0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$	$-\vert 0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle$
1	$\vert 0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle$	$-\vert 0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle$
2	$\vert 0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle$	0
3	$\vert 0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle$	0
4	$\vert 1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$	$\vert 1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle$
5	$\vert 1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle$	$\vert 1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle$
6	$\vert 1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle$	0
7	$\vert 1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle$	0

Here we can see the action of the creation 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:

```
QuantumTableForm[a[3]†]
```

	Input	Output
0	$\vert 0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$	$-\vert 0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle$
1	$\vert 0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle$	0
2	$\vert 0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle$	$\vert 0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle$
3	$\vert 0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle$	0
4	$\vert 1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$	$\vert 1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle$
5	$\vert 1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle$	0
6	$\vert 1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle$	$-\vert 1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle$
7	$\vert 1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle$	0

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]]
```

$$-\bigotimes_{k=1}^{j-1}\left(\sigma_{\hat{k}}^{Z}\right) \,|\, 0\rangle\langle 1 \,|\, \bigotimes_{m=j+1}^{3}\left(\sigma_{\hat{m}}^{0}\right)$$

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]†⟧+]
```

$\sigma_{0,\hat{1}} \cdot \sigma_{0,\hat{2}} \cdot \sigma_{0,\hat{3}}$

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[2], a[2]†⟧+, PauliIdentities → 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 anitcommutator ⟦a[2], a[3]⟧+=a[2]·a[3]+a[3]·a[2] is zero: a[3]·a[2] gives kets with opposite sign to those given by a[2]·a[3]:

```
Grid[{{QuantumTableForm[a[2] · a[3]], QuantumTableForm[a[3] · a[2]]}}, Dividers → All]
```

	Input	Output		Input	Output
0	$\|0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$	0	0	$\|0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$	0
1	$\|0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle$	0	1	$\|0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle$	0
2	$\|0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle$	0	2	$\|0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle$	0
3	$\|0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle$	$-\|0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$	3	$\|0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle$	$\|0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$
4	$\|1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$	0	4	$\|1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$	0
5	$\|1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle$	0	5	$\|1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle$	0
6	$\|1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle$	0	6	$\|1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle$	0
7	$\|1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle$	$-\|1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$	7	$\|1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle$	$\|1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle$

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:

```
TraditionalForm[a[j]]
```

$$-\bigotimes_{k=1}^{j-1}\left(\sigma_{\hat{k}}^{\mathcal{Z}}\right) |\,0\rangle\langle 1\,| \bigotimes_{m=j+1}^{3}\left(\sigma_{\hat{m}}^{0}\right)$$

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

```
PauliExpand[a[2] · a[2]† - a[2]† · a[2]]
```

$\sigma_{0,\hat{1}} \cdot \sigma_{\mathcal{Z},\hat{2}} \cdot \sigma_{0,\hat{3}}$

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[2] · a[2]† - a[2]† · a[2], PauliIdentities → False]
```

$\sigma_{\mathcal{Z},\hat{2}}$

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

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

$\sigma_{\mathcal{X},\hat{1}} \cdot \sigma_{\mathcal{X},\hat{2}} \cdot \sigma_{0,\hat{3}}$

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]
```

$\sigma_{x,\hat{1}} \cdot \sigma_{x,\hat{2}}$

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

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

$-\sigma_{y,\hat{1}} \cdot \sigma_{y,\hat{2}} \cdot \sigma_{0,\hat{3}}$

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]
```

$-\sigma_{y,\hat{1}} \cdot \sigma_{y,\hat{2}}$

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

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

$-i\, \sigma_{x,\hat{1}} \cdot \sigma_{y,\hat{2}} \cdot \sigma_{0,\hat{3}}$

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]
```

$-i\, \sigma_{x,\hat{1}} \cdot \sigma_{y,\hat{2}}$

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

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

$-i\, \sigma_{y,\hat{1}} \cdot \sigma_{x,\hat{2}} \cdot \sigma_{0,\hat{3}}$

TraditionalForm[] gives a format closer to the format of papers and textbooks:

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

$-i\,\sigma_1^{\mathcal{Y}}\sigma_2^{\mathcal{X}}\sigma_3^{\mathit{0}}$

TeXForm can be used to generate expression for papers and books

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

```
-i \sigma _1^{\mathcal{Y}}\sigma _2^{\mathcal{X}}\sigma
   _3^{\mathit{0}}
```

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:

Algebra of operators and commutators. Mathematica file: http://homepage.cem.itesm.mx/lgomez/quantum/v7algebra.nb

Algebra of operators and commutators. PDF file: http://homepage.cem.itesm.mx/lgomez/quantum/v7algebra.pdf

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