Q3: Quick Start

Q3 is a Mathematica application to help study quantum information processing, quantum many-body systems, and quantum spin systems. It provides various tools and utilities for symbolic and numerical calculations in these areas of quantum physics.

Load the application Q3.

```
In[1]:= << Q3`
```

Quantum Computation and Information

Q3 provides many functions for studying quantum information processing.

If your tasks are relatively simple, Pauli will be enough. With Pauli, you are dealing with *unlabelled* qubits. Different qubits are distinguished by the position in Ket , Pauli , and other related functions.

For example, the following expression assumes three qubits -- the first in the state 0, the second 1, and the last 0.

```
In[9]:= ket = Ket[0, 1, 0]
Out[9]= |0, 1, 0⟩
```

The number of qubits must be consistent.

```
In[13]:= op = Pauli[1, 2, 1] + I Pauli[3, 1, 1]
```

Out[13]= $\sigma^x \otimes \sigma^y \otimes \sigma^x + i\, \sigma^z \otimes \sigma^x \otimes \sigma^x$

```
In[14]:= op ** ket
Out[14]= i |0, 0, 1⟩ - i |1, 0, 1⟩
```

For more complicated tasks, Quisso would be better. With Quisso, you can deal with *labelled* qubits.

First, choose a symbol to use to label the qubits. For example, choose **S**, and declare it to be a qubit using Let .

```
In[15]:= Let[Qubit, S]
```

The following expression involves two qubits `S[1,None]` and `S[2,None]`. The final index denotes different Pauli operators acting on the qubit. For example, `S[1,3]` means the Pauli Z acting on `S[1,None]`.

In[59]:=

```
op = S[1, 3] ** S[2, 1]
```

Out[59]=

$S_1^z S_2^x$

The logical state vector can be denoted by Ket [<|…|>]. As in many functions in Quisso, one can skip the final None for each qubit -- it is added automatically.

In[54]:=

```
Ket[S[1, None] → 1, S[2, None] → 1]
Ket[S[1] → 1, S[2] → 1]
```

Out[54]=

$\left|1_{S_1} 1_{S_2}\right\rangle$

Out[55]=

$\left|1_{S_1} 1_{S_2}\right\rangle$

In[60]:=

```
ket = Ket[] + 3 Ket[S[1] → 1, S[2] → 1, S[3] → 1]
```

Out[60]=

$\left|\_\right\rangle + 3 \left|1_{S_1} 1_{S_2} 1_{S_3}\right\rangle$

In this case, you do no have to worry about the number of spins in the operator and the state vector. Different qubits are distinguished by the symbol and the flavor indices (excluding the final index).

In[61]:=

```
new = op ** ket
new // LogicalForm
```

Out[61]=

$-3 \left|1_{S_1} 1_{S_3}\right\rangle + \left|1_{S_2}\right\rangle$

Out[62]=

$\left|0_{S_1} 1_{S_2} 0_{S_3}\right\rangle - 3 \left|1_{S_1} 0_{S_2} 1_{S_3}\right\rangle$

Quantum Many-Body Systems

If you want to calculate some properties of Fermionic systems, Fock is the choice.

Choose a symbol to denote various species of Fermions.

In[40]:=

```
Let[Fermion, c]
```

In[41]:=

```
c[1] ** Dagger[c[1]] ** c[2] ** Dagger[c[2]]
```

Out[41]=

$1 - c_1^\dagger c_1 - c_2^\dagger c_2 + c_1^\dagger c_2^\dagger c_2 c_1$

In[42]:=

```
% // InputForm
```

Out[42]//InputForm=

```
1 - Multiply[Dagger[c[1]], c[1]] - Multiply[Dagger[c[2]], c[2]] +
 Multiply[Dagger[c[1]], Dagger[c[2]], c[2], c[1]]
```

Quantum Spin Systems

You can study quantum spin systems with the Wigner package.

Choose a symbol to use to denote different spins.

In[43]:=

```
Let[Spin, J]
```

Here are two spin angular momentum operators acting on two spins `S[1,None]` and `S[2,None]`, respectively.

In[48]:=

```
op = J[1, 3] ** J[2, 2]
```

Out[48]=

$J_1^z J_2^y$

Related Guides

- Quantum Computation and Information
- Quantum Many–Body Systems
- Quantum Spin Systems
- Q3

Related Tech Notes

- Quantum Computation and Information: Overview
- Quantum Many–Body Systems with Q3
- Quantum Spin Systems with Q3
- Staying Up To Date with Q3

Related Wolfram Education Group Courses

An Elementary Introduction to the Wolfram Language

The Wolfram Language: Fast Introduction for Programmers