Power Series of Operators

by José Luis Gómez-Muñoz
http://homepage.cem.itesm.mx/lgomez/quantum/
jose.luis.gomez@itesm.mx
With contributions by Rubén Darío Santiago-Acosta

Introduction

This is a tutorial on the use of Quantum *Mathematica* add-on to work with power series of operators. The work with power series is compared with the work with matrices.

Load the Package

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

```
Needs["Quantum`Notation`"]
```

```
Quantum`Notation` Version 2.2.0. (July 2010)
A Mathematica package for Quantum calculations in Dirac bra-ket notation
by José Luis Gómez-Muñoz

Execute SetQuantumAliases[] in order to use
  the keyboard to enter quantum objects in Dirac's notation
SetQuantumAliases[] must be executed again in each new
  notebook that is created, only one time per notebook.
```

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

```
SetQuantumAliases[];
```

Powers of Operators

The *Mathematica* command Expand can be used to expand powers of variables that are assumed to be complex numbers:

```
Expand[(a + b)^3]
```

$a^3 + 3 a^2 b + 3 a b^2 + b^3$

After loading Quantum, Expand can be used to expand powers of variables that are assumed to be quantum objects (operators), taking into account that operator application is noncommutative:

```
SetQuantumObject[qa];
SetQuantumObject[qb];
Expand[(qa + qb)^3]
```

```
qa^3 + qb^3 + qb^2 · qa + qb · qa^2 + qa^2 · qb + qa · qb^2 + qa · qb · qa + qb · qa · qb
```

So far qa and qb are treated as operators, but they do Not perform any action in a ket:

```
qa · | 2ĥ⟩
```

```
qa · | 2ĥ⟩
```

We can create another operator that does perform a specific action on eigenkets:

```
DefineOperatorOnKets[qc, { | x_ĥ⟩ :→ | (x + 1)ĥ⟩}]
```

```
| x_ĥ⟩ :→ | (x + 1)ĥ⟩
```

Here we can see our new operator in action:

```
qc · | 2ĥ⟩
```

```
| 3ĥ⟩
```

Here we can see a power of our new operator in action:

```
qc^2 · | 2ĥ⟩
```

```
| 4ĥ⟩
```

We can add powers of this operator and apply them to a ket

```
(a + b * qc + c * qc^2) · | 2ĥ⟩
```

```
a | 2ĥ⟩ + b | 3ĥ⟩ + c | 4ĥ⟩
```

Using power series to approximate the exponential of an operator

This is the definition of the operator we are working with.

```
DefineOperatorOnKets[qc, { | x_ĥ⟩ :→ | (x + 1)ĥ⟩}]
```

```
| x_ĥ⟩ :→ | (x + 1)ĥ⟩
```

Here we use the standard *Mathematica* commands Exp, Series and Normal, in order to define an operator that is a (fifth power) approximation of the exponential of operator qc:

```
operatorFromSeries = Normal[Series[Exp[qc], {qc, 0, 5}]]
```

$$1 + qc + \frac{qc^2}{2} + \frac{qc^3}{6} + \frac{qc^4}{24} + \frac{qc^5}{120}$$

And here we have the approximated exponential operator in action:

$$\text{operatorFromSeries} \cdot \left| 2_{\hat{h}} \right\rangle$$

$$\left| 2_{\hat{h}} \right\rangle + \left| 3_{\hat{h}} \right\rangle + \frac{1}{2} \left| 4_{\hat{h}} \right\rangle + \frac{1}{6} \left| 5_{\hat{h}} \right\rangle + \frac{1}{24} \left| 6_{\hat{h}} \right\rangle + \frac{1}{120} \left| 7_{\hat{h}} \right\rangle$$

We store our result in the ket aproxSeries, because later we will compare it with another way to calculate the exponential:

$$\left| \text{approxSeries} \right\rangle = \text{operatorFromSeries} \cdot \left| 2_{\hat{h}} \right\rangle$$

$$\left| 2_{\hat{h}} \right\rangle + \left| 3_{\hat{h}} \right\rangle + \frac{1}{2} \left| 4_{\hat{h}} \right\rangle + \frac{1}{6} \left| 5_{\hat{h}} \right\rangle + \frac{1}{24} \left| 6_{\hat{h}} \right\rangle + \frac{1}{120} \left| 7_{\hat{h}} \right\rangle$$

Using matrices to approximate the exponential of an operator

This is the definition of the operator we are working with.

$$\text{DefineOperatorOnKets}\left[qc, \left\{ \left| x\_{\hat{h}} \right\rangle :\rightarrow \left| (x+1)_{\hat{h}} \right\rangle \right\}\right]$$

$$\left| x\_{\hat{h}} \right\rangle :\rightarrow \left| (x+1)_{\hat{h}} \right\rangle$$

There is another way to calculate approximations to exponentials of operators. First we assume a finite-dimensional space for the base operator h, then we can get the matrix representation of the operator qc that was defined above. Assuming a six-dimensional space:

$$\text{qcmatrix} = \text{DiracToMatrix}\left[qc, \left\{\left\{0_{\hat{h}}, 1_{\hat{h}}, 2_{\hat{h}}, 3_{\hat{h}}, 4_{\hat{h}}, 5_{\hat{h}}\right\}\right\}\right]$$

```
{{0, 0, 0, 0, 0, 0}, {1, 0, 0, 0, 0, 0}, {0, 1, 0, 0, 0, 0},
 {0, 0, 1, 0, 0, 0}, {0, 0, 0, 1, 0, 0}, {0, 0, 0, 0, 1, 0}}
```

Here we can visualize the matrix

```
MatrixForm[qcmatrix]
```

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Now we can easily get the exponential of this matrix using the standard *Mathematica* command MatrixExp:

```
expmatrix = MatrixExp[qcmatrix]
```

$$\left\{\{1, 0, 0, 0, 0, 0\}, \{1, 1, 0, 0, 0, 0\}, \left\{\frac{1}{2}, 1, 1, 0, 0, 0\right\}, \left\{\frac{1}{6}, \frac{1}{2}, 1, 1, 0, 0\right\}, \left\{\frac{1}{24}, \frac{1}{6}, \frac{1}{2}, 1, 1, 0\right\}, \left\{\frac{1}{120}, \frac{1}{24}, \frac{1}{6}, \frac{1}{2}, 1, 1\right\}\right\}$$

Here we can visualize the exponential matrix

```
MatrixForm[expmatrix]
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 1 & 0 & 0 & 0 \\ \frac{1}{6} & \frac{1}{2} & 1 & 1 & 0 & 0 \\ \frac{1}{24} & \frac{1}{6} & \frac{1}{2} & 1 & 1 & 0 \\ \frac{1}{120} & \frac{1}{24} & \frac{1}{6} & \frac{1}{2} & 1 & 1 \end{pmatrix}$$

Now we go back to Dirac notation, and store the operator under the name aproxFromMatrix

$$\textbf{operatorFromMatrix = MatrixToDirac}\left[\textbf{expmatrix, \{6\},} \left\{\mathbf{x}_{-\hat{1}} \mapsto \mathbf{x}_{\hat{h}}\right\}\right]$$

$$|0_{\hat{h}}\rangle\cdot\langle 0_{\hat{h}}| + |1_{\hat{h}}\rangle\cdot\langle 0_{\hat{h}}| + \frac{1}{2}\,|2_{\hat{h}}\rangle\cdot\langle 0_{\hat{h}}| + \frac{1}{6}\,|3_{\hat{h}}\rangle\cdot\langle 0_{\hat{h}}| + \frac{1}{24}\,|4_{\hat{h}}\rangle\cdot\langle 0_{\hat{h}}| + \frac{1}{120}\,|5_{\hat{h}}\rangle\cdot\langle 0_{\hat{h}}| + |1_{\hat{h}}\rangle\cdot\langle 1_{\hat{h}}| + |2_{\hat{h}}\rangle\cdot\langle 1_{\hat{h}}| + \frac{1}{2}\,|3_{\hat{h}}\rangle\cdot\langle 1_{\hat{h}}| + \frac{1}{6}\,|4_{\hat{h}}\rangle\cdot\langle 1_{\hat{h}}| + \frac{1}{24}\,|5_{\hat{h}}\rangle\cdot\langle 1_{\hat{h}}| + |2_{\hat{h}}\rangle\cdot\langle 2_{\hat{h}}| + |3_{\hat{h}}\rangle\cdot\langle 2_{\hat{h}}| + \frac{1}{2}\,|4_{\hat{h}}\rangle\cdot\langle 2_{\hat{h}}| + \frac{1}{6}\,|5_{\hat{h}}\rangle\cdot\langle 2_{\hat{h}}| + |3_{\hat{h}}\rangle\cdot\langle 3_{\hat{h}}| + |4_{\hat{h}}\rangle\cdot\langle 3_{\hat{h}}| + \frac{1}{2}\,|5_{\hat{h}}\rangle\cdot\langle 3_{\hat{h}}| + |4_{\hat{h}}\rangle\cdot\langle 4_{\hat{h}}| + |5_{\hat{h}}\rangle\cdot\langle 4_{\hat{h}}| + |5_{\hat{h}}\rangle\cdot\langle 5_{\hat{h}}|$$

We can use our new operator that was obtained from the Matrix representation:

```
operatorFromMatrix · |2ĥ⟩
```

$$|2_{\hat{h}}\rangle + |3_{\hat{h}}\rangle + \frac{1}{2}|4_{\hat{h}}\rangle + \frac{1}{6}|5_{\hat{h}}\rangle$$

And we can compare with the result that was obtained from the power series

```
|approxSeries⟩
```

$$|2_{\hat{h}}\rangle + |3_{\hat{h}}\rangle + \frac{1}{2}|4_{\hat{h}}\rangle + \frac{1}{6}|5_{\hat{h}}\rangle + \frac{1}{24}|6_{\hat{h}}\rangle + \frac{1}{120}|7_{\hat{h}}\rangle$$

Notice that the two calculations (series and matrices) can be forced to give exactly the same answer by changing either the size of the matrix or the degree of the truncated series.

---

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