

QSystem: simulador quântico para Python

Evandro Chagas Ribeiro da Rosa

GCQ-UFSC

Setembro de 2019

Conteúdo programático

Computação Quântica

- Bit Quântico

- Circuitos quântico

Simulador QSystem

- Evolução

- Medida

- Representação do estado quântico

- Canais de erro quântico

Exemplo de código

Benchmarks

Conclusão

Computação Quântica

Bit Quântico

Circuitos quântico

Simulador QSystem

Evolução

Medida

Representação do estado quântico

Canais de erro quântico

Exemplo de código

Benchmarks

Conclusão

Computação Quântica

- ▶ Computar problemas NP em tempo polinomial.
- ▶ Algoritmo de Shor
Fatoração em tempo polinomial.
- ▶ Algoritmo de Grover
Busca em banco de dados desordenado em $O(\sqrt{n})$.

Bit Quântico - Qubit

Superposição, Medida e Entrelaçamento

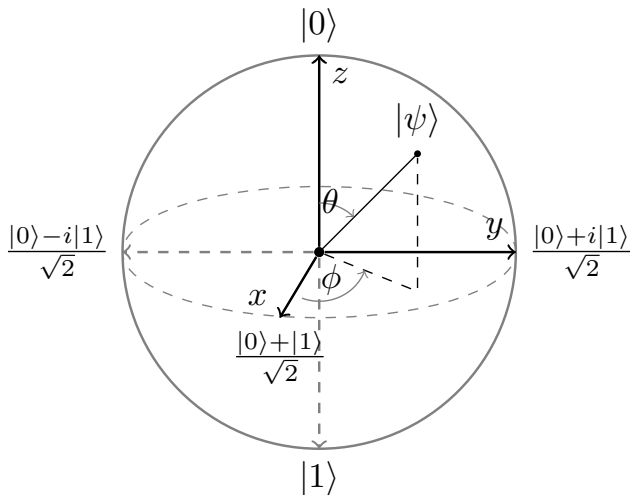


Figura: Esfera de Bloch.

Circuitos quântico

Algoritmo quântico

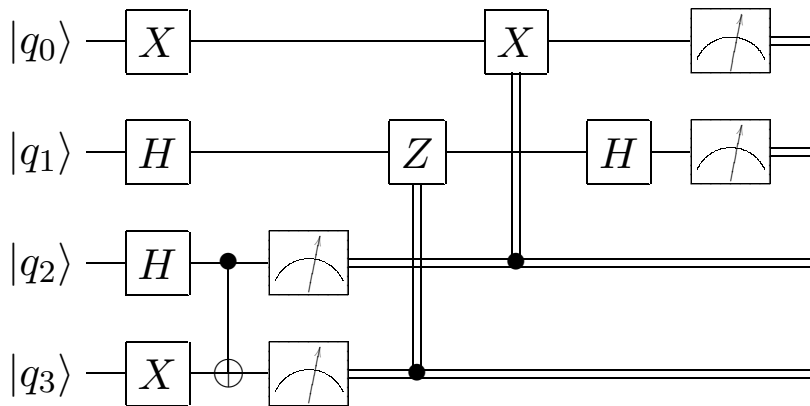


Figura: Exemplo de circuito quântico.

Computação Quântica

Bit Quântico

Circuitos quântico

Simulador QSystem

Evolução

Medida

Representação do estado quântico

Canais de erro quântico

Exemplo de código

Benchmarks

Conclusão

Simulador QSystem

Motivação

- ▶ Ainda é necessário muita pesquisa.
- ▶ Ferramenta de auxilio
 - ▶ Algoritmos quântico
 - ▶ Protocolos quântico
 - ▶ Códigos quântico
- ▶ Simples e intuitivo.
- ▶ Abstrair a matemática por traz da simulação.

Modulo Python







versão 1.2.0b2

```
QSystem(num_qbits,  
        seed=42,  
        representation="bitwise",  
        init=0)
```

Evolução

Portas de um qubit

```
def evol (self, gate, qbit, count=1, invert=False):
```

'X'		$ 0\rangle \rightarrow 1\rangle$ $ 1\rangle \rightarrow 0\rangle$
'Z'		$ 0\rangle \rightarrow 0\rangle$ $ 1\rangle \rightarrow - 1\rangle$
'Y'		$ 0\rangle \rightarrow i 1\rangle$ $ 1\rangle \rightarrow -i 1\rangle$
'H'		$ 0\rangle \rightarrow (0\rangle + 1\rangle)/\sqrt{2}$ $ 1\rangle \rightarrow (0\rangle - 1\rangle)/\sqrt{2}$
'S'		$ 0\rangle \rightarrow 0\rangle$ $ 1\rangle \rightarrow i 1\rangle$
'T'		$ 0\rangle \rightarrow 0\rangle$ $ 1\rangle \rightarrow e^{i\pi/4} 1\rangle$

Evolução

Portas de um qubit

```
def rot (self, axis, angle, qbit, count=1):
```

$$'X' = \begin{bmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}$$

$$'Y' = \begin{bmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}$$

$$'Z' = \begin{bmatrix} -e^{i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix}$$

Evolução

Portas de um qubit

```
def u3(self, theta, phi, lambda, qbit, count=1):  
def u2(self, phi, lambda, qbit, count=1):  
def u1(self, lambda, qbit, count=1):
```

$$u3(\theta, \phi, \lambda) = \begin{bmatrix} \cos \frac{\theta}{2} & -e^{i\lambda} \sin \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} & e^{i(\lambda+\phi)} \cos \frac{\theta}{2} \end{bmatrix}$$

$$u3(\pi/2, \phi, \lambda) = u2(\phi, \lambda) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -e^{i\lambda} \\ e^{i\phi} & e^{i(\lambda+\phi)} \end{bmatrix}$$

$$u3(0, 0, \lambda) = u1(\lambda) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{bmatrix}$$

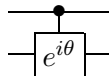
Evolução

Portas de múltiplos qubits

```
def cnot (self, target, control):
```



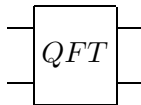
```
def cphase (self, phase, target, control):
```



```
def swap (self, qbit_a, qbit_b):
```



```
def qft (self, begin, end, invert=False):
```



Criação de portas lógicas quântica

Classe Gate

`Gate.from_matrix(matrix):`

▶ `matrix = [U00, U01, U10, U11]`

`Gate.from_sp_matrix(size, row, col, value):`

▶ `m(row[i], col[i]) = value[i]`

`Gate.from_func(func, size, iterator=None):`

`Gate.cxz_gate(gates, control):`

Criação de portas lógicas quântica

Aplicar porta criada

```
def apply(self, gate, qbit, count=1, invert=False):
```

Medida

Medida na base computacional

```
def measure(self, qbit, count=1):
```



```
def measure_all(self):
```



```
def bits(self):
```

```
[0, 1, None]
```


Representação do estado quântico

Vetor de estado

```
QSystem(num_qbits, representation='vector')
```

$+1/\sqrt{8}$	$ 000\rangle$
$+1/\sqrt{8}$	$ 001\rangle$
$+1/\sqrt{8}$	$ 010\rangle$
$+1/\sqrt{8}$	$ 011\rangle$
$+1/\sqrt{8}$	$ 100\rangle$
$+1/\sqrt{8}$	$ 101\rangle$
$+1/\sqrt{8}$	$ 110\rangle$
$+1/\sqrt{8}$	$ 111\rangle$

Representação do estado quântico

Bitwise

```
QSystem(num_qbits, representation='bitwise')
```

$+1/\sqrt{8}$	$ 101\rangle$
$+1/\sqrt{8}$	$ 100\rangle$
$+1/\sqrt{8}$	$ 111\rangle$
$+1/\sqrt{8}$	$ 110\rangle$
$+1/\sqrt{8}$	$ 001\rangle$
$+1/\sqrt{8}$	$ 000\rangle$
$+1/\sqrt{8}$	$ 011\rangle$
$+1/\sqrt{8}$	$ 010\rangle$

Representação do estado quântico

Matriz densidade

```
QSystem(num_qbits, representation='matrix')
```

(0, 0)	+0.5000000000
(1, 0)	+0.5000000000
(0, 1)	+0.5000000000
(1, 1)	+0.5000000000

Canais de erro quântico

Para matriz densidade

```
def flip(self, gate, qbit, p):
```

- ▶ gate='X' *Bit flip*
- ▶ gate='Z' *Phase flip*
- ▶ gate='Y' *Bit-phase flip*

```
def dpl_channel(self, qbit, p):
```

```
def amp_damping(self, qbit, p):
```

```
def sum(self, qbit, kraus, p):
```

Computação Quântica

Bit Quântico

Circuitos quântico

Simulador QSystem

Evolução

Medida

Representação do estado quântico

Canais de erro quântico

Exemplo de código

Benchmarks

Conclusão

Algoritmo de Shor

Exemplo de código

```
from qsystem import QSystem, Gate
from random import randint, seed
from math import log2, ceil, gcd
to_int = lambda c : sum(
    [m*2**i for m, i in zip(
        c, reversed(range(len(c)))])])
seed(47)

n = 15
```

Algoritmo de Shor

Exemplo de código

```
a = 0
while gcd(n, a) != 1:
    a = randint(2, n)
```

1. Selecione aleatoriamente um número a coprimo a n .
 2. Ache o período r da função $f(x) = a^x \pmod n$.
- ▶ $a = 7$

Algoritmo de Shor

Exemplo de código

```
s = ceil(log2(n+1))
```

```
def pown(x):
```

```
    x = x >> s
```

```
    fx = pow(a, x, n)
```

```
    return (x << s) | fx
```

```
def it():
```

```
    for x in range(2**s):
```

```
        yield x << s
```

```
pown_gate = Gates.from_func(pown, 2*s, it())
```

► Cria porta lógica

► $|x\rangle |0\rangle \xrightarrow{\text{POWN}} |x\rangle |a^x \bmod n\rangle$

Algoritmo de Shor

Exemplo de código

```
q = QSystem(s, 13, 'vector')
q.evol('H', 0, s)
q.add_ancillas(s)
q.apply(pown_gate, 0)
q.measure(s, s)
q.rm_ancillas()
q.qft(0, s)

q.measure_all()
c = to_int(q.bits())
```

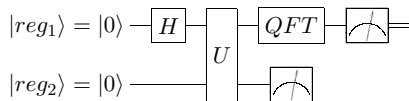


Figura: Circuito para achar o período da função $a^x \pmod n$.

Algoritmo de Shor

Exemplo de código

```
q = QSystem(s, 13, 'vector') <-  
q.evol('H', 0, s)  
q.add_ancillas(s)  
q.apply(pown_gate, 0)  
q.measure(s, s)                +1.000                |0000>  
q.rm_ancillas()  
q.qft(0, s)  
  
q.measure_all()  
c = to_int(q.bits())
```

Algoritmo de Shor

Exemplo de código

```
q = QSystem(s, 13, 'vector')
q.evol('H', 0, s) <-
q.add_ancillas(s)
q.apply(pown_gate, 0)
q.measure(s, s)
q.rm_ancillas()
q.qft(0, s)

q.measure_all()
c = to_int(q.bits())
```

+1/sqrt(16)	0000>
+1/sqrt(16)	0001>
+1/sqrt(16)	0010>
+1/sqrt(16)	0011>
+1/sqrt(16)	0100>
+1/sqrt(16)	0101>
+1/sqrt(16)	0110>
+1/sqrt(16)	0111>
+1/sqrt(16)	1000>
+1/sqrt(16)	1001>
+1/sqrt(16)	1010>
+1/sqrt(16)	1011>
+1/sqrt(16)	1100>
+1/sqrt(16)	1101>
+1/sqrt(16)	1110>
+1/sqrt(16)	1111>

Algoritmo de Shor

Exemplo de código

```
q = QSystem(s, 13, 'vector')
q.evol('H', 0, s)
q.add_ancillas(s) <-
q.apply(pown_gate, 0)
q.measure(s, s)
q.rm_ancillas()
q.qft(0, s)

q.measure_all()
c = to_int(q.bits())
```

+1/sqrt(16)	0000> 0000>
+1/sqrt(16)	0001> 0000>
+1/sqrt(16)	0010> 0000>
+1/sqrt(16)	0011> 0000>
+1/sqrt(16)	0100> 0000>
+1/sqrt(16)	0101> 0000>
+1/sqrt(16)	0110> 0000>
+1/sqrt(16)	0111> 0000>
+1/sqrt(16)	1000> 0000>
+1/sqrt(16)	1001> 0000>
+1/sqrt(16)	1010> 0000>
+1/sqrt(16)	1011> 0000>
+1/sqrt(16)	1100> 0000>
+1/sqrt(16)	1101> 0000>
+1/sqrt(16)	1110> 0000>
+1/sqrt(16)	1111> 0000>

Algoritmo de Shor

Exemplo de código

```
q = QSystem(s, 13, 'vector')
q.evol('H', 0, s)
q.add_ancillas(s)
q.apply(pown_gate, 0) <-
q.measure(s, s)
q.rm_ancillas()
q.qft(0, s)

q.measure_all()
c = to_int(q.bits())
```

+1/sqrt(16)	0000> 0001>
+1/sqrt(16)	0001> 0111>
+1/sqrt(16)	0010> 0100>
+1/sqrt(16)	0011> 1101>
+1/sqrt(16)	0100> 0001>
+1/sqrt(16)	0101> 0111>
+1/sqrt(16)	0110> 0100>
+1/sqrt(16)	0111> 1101>
+1/sqrt(16)	1000> 0001>
+1/sqrt(16)	1001> 0111>
+1/sqrt(16)	1010> 0100>
+1/sqrt(16)	1011> 1101>
+1/sqrt(16)	1100> 0001>
+1/sqrt(16)	1101> 0111>
+1/sqrt(16)	1110> 0100>
+1/sqrt(16)	1111> 1101>

Algoritmo de Shor

Exemplo de código

```
q = QSystem(s, 13, 'vector')
q.evol('H', 0, s)
q.add_ancillas(s)
q.apply(pown_gate, 0)
q.measure(s, s) <-
q.rm_ancillas()
q.qft(0, s)

q.measure_all()
c = to_int(q.bits())
```

```
+1/sqrt(4)      |0001>|0111>
+1/sqrt(4)      |0101>|0111>
+1/sqrt(4)      |1001>|0111>
+1/sqrt(4)      |1101>|0111>
```

$$\sqrt{\frac{r}{2^s}} \sum_{k=0}^{\frac{2^s}{r}-1} |k r + x_0\rangle |a^{x_0} \pmod n\rangle$$

Algoritmo de Shor

Exemplo de código

```
q = QSystem(s, 13, 'vector')
q.evol('H', 0, s)
q.add_ancillas(s)
q.apply(pown_gate, 0)          +1/sqrt(4)      |0001>
q.measure(s, s)               +1/sqrt(4)      |0101>
q.rm_ancillas() <-           +1/sqrt(4)      |1001>
q.qft(0, s)                   +1/sqrt(4)      |1101>

q.measure_all()
c = to_int(q.bits())
```

Algoritmo de Shor

Exemplo de código

```
q = QSystem(s, 13, 'vector')
q.evol('H', 0, s)
q.add_ancillas(s)
q.apply(pown_gate, 0)
q.measure(s, s)
q.rm_ancillas()
q.qft(0, s) <-
+1/sqrt(4) |0001>
+1/sqrt(4) |0101>
+1/sqrt(4) |1001>
+1/sqrt(4) |1101>
q.measure_all()
c = to_int(q.bits())
+1/sqrt(4) |0000>
+1/sqrt(4)i |0100>
-1/sqrt(4) |1000>
-1/sqrt(4)i |1100>
```

$$\sqrt{\frac{r}{2^s}} \sum_{k=0}^{\frac{2^s}{r}-1} |k r + x_0\rangle \xrightarrow{\text{qft}} \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \left| k \frac{2^s}{r} \right\rangle e^{i\phi_k}$$

Algoritmo de Shor

Exemplo de código

```
q = QSystem(s, 13, 'vector')
q.evol('H', 0, s)
q.add_ancillas(s)
q.apply(pown_gate, 0)
q.measure(s, s)
q.rm_ancillas()
q.qft(0, s)
```

► $c = 4$

```
q.measure_all() <-
c = to_int(q.bits())
```

Algoritmo de Shor

Exemplo de código

```
mea = [c]
for _ in range(s-1):
    se = randint(210,760)
    q = QSystem(s, se, 'vector')
    q.evol('H', 0, s)
    q.add_ancillas(s)
    q.apply(pown_gate, 0)
    q.rm_ancillas()
    q.qft(0, s)
    q.measure_all()
    c = to_int(q.bits())
    mea.append(c)
```

► mea = [4, 12, 4, 12]

Algoritmo de Shor

Exemplo de código

```
c = mea[0]
for m in mea:
    c = gcd(c, m)
if c == 0:
    print('Erro')
else:
    r = 2**s/c
    if r % 2 == 1:
        print('Erro')
    else:
        p = gcd(int(a**(r/2)+1), n)
        q = gcd(int(a**(r/2)-1), n)
```

► $r = 2^s/c = 4$

3. Calcular p e $q \mid p \times q = n$

$$p = \gcd(a^{r/2} + 1, n)$$

$$q = \gcd(a^{r/2} - 1, n)$$

► $5 \times 3 = 15$

Computação Quântica

Bit Quântico

Circuitos quântico

Simulador QSystem

Evolução

Medida

Representação do estado quântico

Canais de erro quântico

Exemplo de código

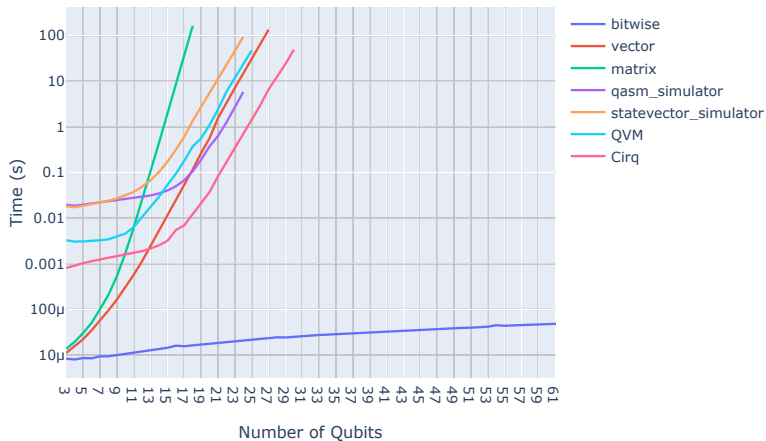
Benchmarks

Conclusão

Estado GHZ

$$\frac{1}{\sqrt{2}} |0 \cdots 0\rangle + \frac{1}{\sqrt{2}} |1 \cdots 1\rangle$$

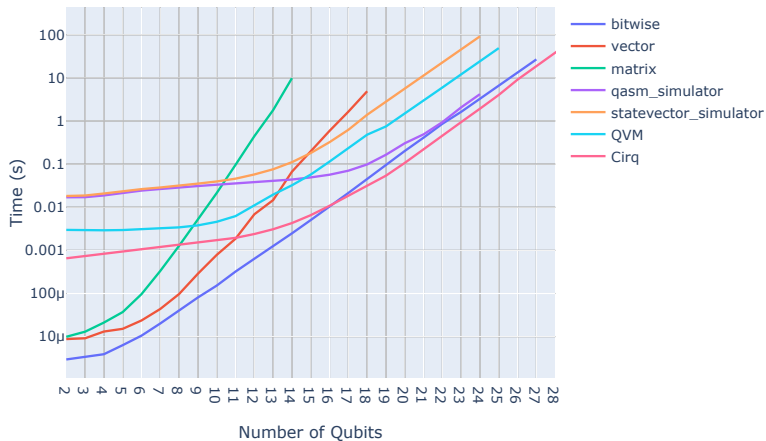
GHZ



Superposição

$$\frac{1}{\sqrt{2^n}} |0\rangle + \frac{1}{\sqrt{2^n}} |1\rangle + \dots + \frac{1}{\sqrt{2^n}} |2^n - 1\rangle$$

Hadamard



Computação Quântica

Bit Quântico

Circuitos quântico

Simulador QSystem

Evolução

Medida

Representação do estado quântico

Canais de erro quântico

Exemplo de código

Benchmarks

Conclusão

Conclusão

- ▶ Instalação: `pip install QSystem==1.2.0b2`
- ▶ Documentação: `evandro-crr.gitlab.io/qsystem`
- ▶ Performance
 - ▶ Código C++
 - ▶ Complexidade exponencial
- ▶ Trabalhos futuros
 - ▶ *Multithreading*
 - ▶ Matrizes esparsas
- ▶ Open source: `gitlab.com/evandro-crr/qsystem`

Obrigado

Perguntas?