decomposition

April 7, 2022

1 Decomposition

1.1 In this notebook we are going to explain step-by-step our optimal decomposition

2 1) Trotterization

We decide to split the XXX Hamiltonian into two pieces $H = H_1 + H_2$ (instead of 6):

$$H_1 = I^{(0)} \otimes \sigma_x^{(1)} \otimes \sigma_x^{(2)} + I^{(0)} \otimes \sigma_y^{(1)} \otimes \sigma_y^{(2)} + I^{(0)} \otimes \sigma_z^{(0)} \otimes \sigma_z^{(1)}$$

$$H_2 = \sigma_x^{(0)} \otimes \sigma_x^{(1)} \otimes I^{(2)} + \sigma_y^{(0)} \otimes \sigma_y^{(1)} \otimes I^{(2)} + \sigma_z^{(0)} \otimes \sigma_z^{(1)} \otimes I^{(2)}$$

so using Trotter's formula:

$$e^{-iHt} = e^{-i(H_1+H_2)t} \simeq \left(e^{-iH_1\frac{t}{N}} e^{-iH_2\frac{t}{N}}\right)^N$$

Now we need to compute a sigle Trotter Step in function of $\frac{t}{N}$

[ ]:
```
import numpy as np
from sympy import *
from sympy.physics.quantum import TensorProduct as Tp
import warnings
warnings.filterwarnings('ignore')

X = Matrix([[0,1],[1,0]])  #defining the pauli matrices
Y = Matrix([[0,-I],[I,0]])
Z = Matrix([[1,0],[0,-1]])
Id = eye(2)

H1 = Tp(X,X,Id) + Tp(Y,Y,Id) + Tp(Z,Z,Id)
H1
```

[ ]:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

[ ]:
```
a = symbols("a")

m = Matrix([
    [exp(-I*a),0,0,0],
    [0,cos(a)*exp(I*a),-I*sin(a)*exp(I*a),0],
    [0,-I*sin(a)*exp(I*a),cos(a)*exp(I*a),0],
    [0,0,0,exp(-I*a)]
])
m
```

[ ]:
$$\begin{bmatrix} e^{-ia} & 0 & 0 & 0 \\ 0 & e^{ia}\cos(a) & -ie^{ia}\sin(a) & 0 \\ 0 & -ie^{ia}\sin(a) & e^{ia}\cos(a) & 0 \\ 0 & 0 & 0 & e^{-ia} \end{bmatrix}$$

[ ]:
```
exp_H1p = Tp(m, Id)
exp_H1p
```

[ ]:
$$\begin{bmatrix} e^{-ia} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{-ia} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e^{ia}\cos(a) & 0 & -ie^{ia}\sin(a) & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{ia}\cos(a) & 0 & -ie^{ia}\sin(a) & 0 & 0 \\ 0 & 0 & -ie^{ia}\sin(a) & 0 & e^{ia}\cos(a) & 0 & 0 & 0 \\ 0 & 0 & 0 & -ie^{ia}\sin(a) & 0 & e^{ia}\cos(a) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{-ia} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-ia} \end{bmatrix}$$

where $a = \frac{2t}{N}$

We can also compute H_2

[ ]:
```
exp_H2p = Tp(Id,m)
exp_H2p
```

[ ]:
$$\begin{bmatrix} e^{-ia} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{ia}\cos(a) & -ie^{ia}\sin(a) & 0 & 0 & 0 & 0 & 0 \\ 0 & -ie^{ia}\sin(a) & e^{ia}\cos(a) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{-ia} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{-ia} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{ia}\cos(a) & -ie^{ia}\sin(a) & 0 \\ 0 & 0 & 0 & 0 & 0 & -ie^{ia}\sin(a) & e^{ia}\cos(a) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-ia} \end{bmatrix}$$

So, the final form of a Trotter step is :

$$e^{-iHt} = e^{-i(H_1+H_2)t} \simeq \left(e^{-iH_1\frac{t}{N}}e^{-iH_2\frac{t}{N}}\right)^N$$

$$T_{step} = e^{-iH_1\frac{t}{N}}e^{-iH_2\frac{t}{N}} =$$

```
[ ]: Trotter_Step =  exp_H1p * exp_H2p
     Trotter_Step
```

[ ]:
$$\begin{bmatrix} e^{-2ia} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cos(a) & -i\sin(a) & 0 & 0 & 0 & 0 \\ 0 & -ie^{2ia}\sin(a)\cos(a) & e^{2ia}\cos^2(a) & 0 & -i\sin(a) & 0 & 0 \\ 0 & 0 & 0 & \cos(a) & 0 & -ie^{2ia}\sin(a)\cos(a) & -e^{2ia}\sin^2(a) \\ 0 & -e^{2ia}\sin^2(a) & -ie^{2ia}\sin(a)\cos(a) & 0 & \cos(a) & 0 & 0 \\ 0 & 0 & 0 & -i\sin(a) & 0 & e^{2ia}\cos^2(a) & -ie^{2ia}\sin(a)\cos(\\ 0 & 0 & 0 & 0 & 0 & -i\sin(a) & \cos(a) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

3 2) Single Column Decomposition

This Decomposition works only in a symmetry preserving subspace of the Hamiltonian. In this case we are in the magnetization m=2 subspace.

We need to find a U_{best} gate for which $T^n_{step}|110>= U_{best}|110>$ and reduces the depth of the circuit. If the initial state is a vector of the computational basis this decomposition coincides with the preparation of a specific column of the matrix, in this case the 7th column.

The idea is to find a generic circuit that preserves the symmetry and dependes on 2d real parameters,where d is the dimension of the subspace.
Instead of repeating the Trotter Step we numerically calculate T^n_{step} and find the parameters of the optimizated circuit. In this case we need at least a six parameters circuit.

3.1 2.1) Parametric Circuit

To find the most generic magnetization preserving circuit we start by defining a general 2-qubit operator that do this and depends on 3 parameters:

```
[ ]: a1, r1, f1 = symbols("a1 r1 f1")

     H = Matrix([
         [1/sqrt(2),1/sqrt(2)],
         [1/sqrt(2),-1/sqrt(2)]
     ])

     cx_01= Matrix([
         [1,0,0,0],
         [0,0,0,1],
         [0,0,1,0],
         [0,1,0,0]
     ])

     def ry(alpha):   # generic ry gate
         return Matrix([
             [cos(alpha/2),-sin(alpha/2)],
             [sin(alpha/2),cos(alpha/2)]
```

```
    ])

def rz(alpha):   # generic rz gate
    return Matrix([
        [exp(-1j*(alpha/2)),0],
        [0,exp(1j*(alpha/2))]
    ])

M1 =␣
 ↪Matrix(simplify(Tp(rz(2*f1),Id)*Tp(Id,H)*cx_01*Tp(ry(a1),ry(a1))*cx_01*Tp(Id,H)*Tp(rz(2*r1)
M1
```

[ ]:

$$\begin{bmatrix} e^{1.0i(-f_1-r_1)} & 0 & 0 & 0 \\ 0 & e^{-1.0i(f_1+r_1)}\cos(a_1) & -e^{-1.0i(f_1-r_1)}\sin(a_1) & 0 \\ 0 & e^{1.0i(f_1-r_1)}\sin(a_1) & e^{1.0i(f_1+r_1)}\cos(a_1) & 0 \\ 0 & 0 & 0 & e^{1.0i(f_1+r_1)} \end{bmatrix}$$

We need at least six parameters so we decide to apply M_1 to qubits $[0,1]$ and M_2 to $[1,2]$.

$$U_{best} = (I \otimes M_2(f_2, r_2, a_2))(M_1(f_1, r_1, a_1) \otimes I)$$

[ ]:

```
a2, r2, f2 = symbols("a2 r2 f2")
M2 =␣
 ↪Matrix(simplify(Tp(rz(2*f2),Id)*Tp(Id,H)*cx_01*Tp(ry(a2),ry(a2))*cx_01*Tp(Id,H)*Tp(rz(2*r2)

Ubest = Tp(M2, Id) * Tp(Id, M1)
Ubest
```

[ ]:

$$\begin{bmatrix} e^{1.0i(-f_1-r_1)}e^{1.0i(-f_2-r_2)} & 0 & 0 & \\ 0 & e^{-1.0i(f_1+r_1)}e^{1.0i(-f_2-r_2)}\cos(a_1) & -e^{-1.0i(f_1-r_1)}e^{1.0i(-f_2-r_2)}\sin(a_1) & \\ 0 & e^{1.0i(f_1-r_1)}e^{-1.0i(f_2+r_2)}\sin(a_1)\cos(a_2) & e^{1.0i(f_1+r_1)}e^{-1.0i(f_2+r_2)}\cos(a_1)\cos(a_2) & \\ 0 & 0 & 0 & e^{1.0i(f_1+} \\ 0 & e^{1.0i(f_1-r_1)}e^{1.0i(f_2-r_2)}\sin(a_1)\sin(a_2) & e^{1.0i(f_1+r_1)}e^{1.0i(f_2-r_2)}\sin(a_2)\cos(a_1) & \\ 0 & 0 & 0 & e^{1.0i(f_1-} \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \end{bmatrix}$$

Perfect, now imposing the equality between the elements of the 7^{th} column (relative to $|110>$) of the parametric matrix U_{best} and of the numerical evaluation $(T_{step})^n$ matrix we can define our Gate.

So, the $(T_{step})^n$ matrix is easly computed here:

[ ]:

```
steps = 42
time = np.pi
U = eye(8)

for _ in range(steps):
    U=U*Trotter_Step
    U=U.subs(a,2*time/steps)
```

```
    U=U.evalf(40)

U.evalf(4)
```

[ ]:

$$\begin{bmatrix} 1.0 - 2.873 \cdot 10^{-16} i & 0 & 0 & 0 & 0 & \\ 0 & 0.4367 + 0.3027i & 0.07228 + 0.526i & 0 & -0.5554 + 0.3569i & \\ 0 & -0.05856 + 0.6128i & -0.1395 - 0.5654i & 0 & 0.07228 + 0.526i & \\ 0 & 0 & 0 & 0.4367 + 0.3027i & 0 & -0.0585 \\ 0 & -0.5507 + 0.1882i & -0.05856 + 0.6128i & 0 & 0.4367 + 0.3027i & \\ 0 & 0 & 0 & 0.07228 + 0.526i & 0 & -0.139 \\ 0 & 0 & 0 & -0.5554 + 0.3569i & 0 & 0.0722 \\ 0 & 0 & 0 & 0 & 0 & \end{bmatrix}$$

We impose the following equation in order to have the same values on the 7^{th} column of both the matrices:

$$\begin{cases} b_0 = im(U_{4,7}) \\ b_1 = im(U_{6,7}) \\ b_2 = im(U_{7,7}) \\ \alpha_0 = re(U_{4,7}) \\ \alpha_1 = re(U_{6,7}) \\ \alpha_2 = re(U_{7,7}) \end{cases}$$

$$\begin{cases} -f_1 + r_1 - f_2 + r_2 = atan(\frac{b_0}{\alpha_0}) = x_0 \\ -\pi - f_1 + r_1 + f_2 + r_2 = atan(\frac{b_1}{\alpha_1}) = x_1 \\ f_1 + r_1 + f_2 + r_2 = atan(\frac{b_2}{\alpha_2}) = x_2 \\ sin(a_1)sin(a_2) = |U_{4,7}| = \sqrt{\alpha_0^2 + b_0^2} \\ sin(a_1)cos(a_2) = |U_{6,7}| = \sqrt{\alpha_1^2 + b_1^2} \\ cos(a_1) = |U_{7,7}| = \sqrt{\alpha_2^2 + b_2^2} \end{cases}$$

solving them one can find:

$$\begin{cases} r_1 = \frac{x_0 + x_2}{2} \\ r_2 = 0 \\ f_2 = \frac{x_2 - x_1 - \pi}{2} \\ f_1 = \frac{x_0 - x_1 - \pi}{2} \\ a_1 = acos(\sqrt{\alpha_2^2 + b_2^2}) \\ a_2 = acos(\frac{\sqrt{\alpha_1^2 + b_1^2}}{sin(a_1)}) \end{cases}$$

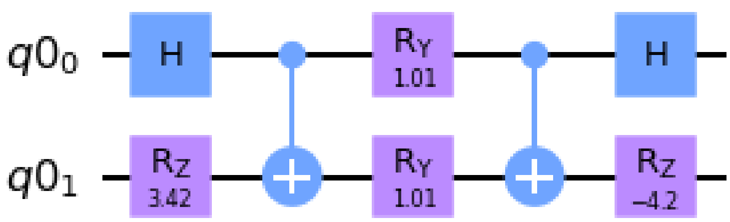
3.2 3) Decomposition of M_1 and M_2 operators

We need to find the minimal decomposition of the M gate. The M gate acts as generic 1 qubit gate on the subspace $|10>, |01>$ and as a phase for $|00>$ and $|11>$ So we used the Givens Rotation gate as defined in: "https://arxiv.org/pdf/2104.05695.pdf" that perform a RY in the subspace $|10>, |01>$ with just 2 cnots. We can think to do a ZYZ decomposition to finally obtain :

$$M = (I \otimes RZ(f))G(r)(I \otimes RZ(a))$$

[ ]:
```
from qiskit import Aer, QuantumCircuit, QuantumRegister, execute

r1=float(atan2(im(U[3*8+6]),re(U[3*8+6]))+atan2(im(U[6*8+6]),re(U[6*8+6])))/2
r2=0
f1=float(atan2(im(U[6*8+6]),re(U[6*8+6]))-atan2(im(U[5*8+6]),re(U[5*8+6]))-np.
↪pi)/2
f2=float((atan2(im(U[6*8+6]),re(U[6*8+6]))-atan2(im(U[3*8+6]),re(U[3*8+6])))/
↪2-f1)
a1=float(acos(abs(U[6*8+6])))
a2=float(acos(abs(U[5*8+6])/sin(a1)))

qr1=QuantumRegister(2)
M1_qc=QuantumCircuit(qr1, name="M1")

M1_qc.rz(2*r1,qr1[1])
M1_qc.h(qr1[0])
M1_qc.cx(qr1[0],qr1[1])
M1_qc.ry(a1,qr1)
M1_qc.cx(qr1[0],qr1[1])
M1_qc.h(qr1[0])
M1_qc.rz(2*f1,qr1[1])

M1_qc.draw(output="mpl")
```

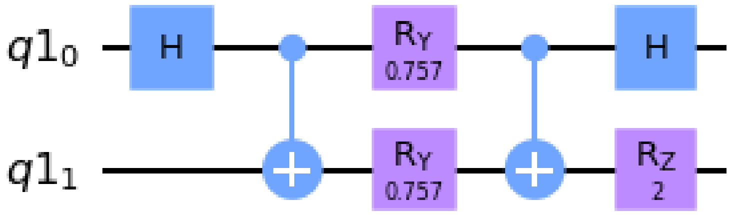
[ ]:



[ ]:
```
qr2=QuantumRegister(2)
M2_qc=QuantumCircuit(qr2, name="M2")

#M2_qc.rz(2*r2,qr2[1])
M2_qc.h(qr2[0])
M2_qc.cx(qr2[0],qr2[1])
M2_qc.ry(a2,qr2)
M2_qc.cx(qr2[0],qr2[1])
M2_qc.h(qr2[0])
M2_qc.rz(2*f2,qr2[1])
```

```
M2_qc.draw(output="mpl")
```

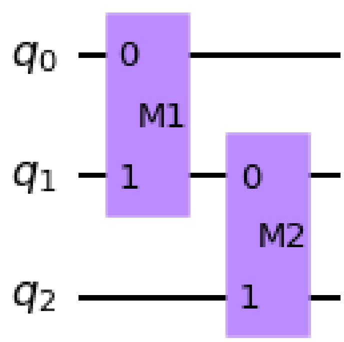
[ ]:



3.3 Buiding the final evolution cirquit

[ ]:
```
qr = QuantumRegister(3 ,name="q")
qc = QuantumCircuit(qr, name="U")

qc.append(M1_qc, [qr[0],qr[1]])
qc.append(M2_qc, [qr[1],qr[2]])

qc.draw(output="mpl")
```

[ ]:



[ ]:
```
def matrix_from_cirquit(qc, phase=0):
    backend = Aer.get_backend('unitary_simulator')
    job = execute(qc, backend, shots=32000)
    result = job.result()
    A=result.get_unitary(qc, decimals=10)*np.exp(1j*phase)
    return Matrix(A)
```

You can see that the 7^{th} columns are equivalent.

[ ]:
```
## 7^th column of the matrix rapresentation of the quantumcircuit.
Matrix([matrix_from_cirquit(qc)[j*8+6] for j in range(8)])
```

[ ]:
$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ -0.5506938802 + 0.1881775479i \\ 0 \\ -0.0585630369 + 0.6128121094i \\ 0.4367423281 + 0.3026768721i \\ 0 \end{bmatrix}$$

[ ]:
```
## 7^th column of the matrix obtained multipling n times trotter_steps_matrix␣
↪with itself.
Matrix([U.evalf(10)[j*8+6] for j in range(8)])
```

[ ]:
$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ -0.5506938803 + 0.1881775479i \\ 0 \\ -0.0585630369 + 0.6128121094i \\ 0.4367423281 + 0.3026768721i \\ 0 \end{bmatrix}$$

In conclusion we have a circuit with a fixed depth (4 cnots and 11 sigle qubit gates) for every choices of time and number of Trotter Steps