

Qibotn TEBD Feature

First steps:

- Understand the input (Qibo level) and output (Quimb level) of this feature
- Understand TEBD algorithm theory

Second step:

- Quantum gates => TEBD-related TN graphs (e.g. MPS/MPO as per this link: <https://tensornetwork.org/mps/algorithms/timeevo/tebd.html>)
- More points and doubts highlighted in II. for discussion

I. Self-experiment 1: The input layer

Step 1) Casting Hamiltonian as Qibo Circuit using Trotter Decomposition

Step 2) Extracting Local Unitaries for next steps in process – *this is the start of the translational layer*

Experimental observations (1) and (2) highlighted as comments in code

```
self_experiment1.py > ...
1 # A self experiment on how to cast H as qibo circuit and extract unitaries for next steps in process
2
3 import qibo
4 from qibo import hamiltonians, gates
5
6 # Casting of a hamiltonian as a circuit using TD
7 ham = hamiltonians.XXZ(nqubits=5, dense=False)
8 circuit = ham.circuit(dt=1e-2)
9
10 # Method 1: OpenQASM
11 # Challenge 1: Unitary is not supported by OpenQASM
12
13 print(circuit.to_qasm())
14
15 # Method 2: Extracting Unitaries manually
16 # Challenge 2: We can extract the unitaries but we need as matrices
17
18 def extract_local_unitaries(circuit):
19     """Extract local unitaries from a Qibo circuit."""
20     local_unitaries = []
21     for gate in circuit.gates_of_type(qibo.gates.Unitary):
22         local_unitaries = gate.matrix
23     return local_unitaries
24
25 print([extract_local_unitaries(circuit)])
26
27
28
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Traceback (most recent call last):
  File "C:\Users\nithy\qibotn-optimize-nith\self_experiment1.py", line 13, in <module>
    print(circuit.to_qasm())
  File "C:\Users\nithy\.conda\envs\nithyasri\lib\site-packages\qibo\models\circuit.py", line 1177, in to_qasm
    name = f"{gate.qasm_label}({', '.join(params)})"
  File "C:\Users\nithy\.conda\envs\nithyasri\lib\site-packages\qibo\gates\abstract.py", line 144, in qasm_label
    raise_error(
  File "C:\Users\nithy\.conda\envs\nithyasri\lib\site-packages\qibo\config.py", line 46, in raise_error
    raise exception(message)
NotImplementedError: Unitary is not supported by OpenQASM
```

Brief notes: Trotter decomposition in qibo lets users write a circuit with a hamiltonian where the hamiltonian gets stored as `qibo.gates.Unitary` (Lines 7-8). It works only with symbolic or built-in hamiltonians with `dense=False`. This means that qibotn's current structure that is able to only deal with circuits and not directly hamiltonians, can further be extended to do TEBD by using this method for input.

User should be able to use this feature using a sample code like: (We might need to introduce TEBD enabled as a runcard setting accordingly)

```

computation_settings = {
    "MPI_enabled": False,
    "MPS_enabled": False,
    "NCCL_enabled": False,
    "expectation_enabled": False
}

qibo.set_backend(backend="qibotn", platform="qutensornet", runcard=computation_settings)

#ham = hamiltonians.TFIM(nqubits=5, dense=False)
ham = hamiltonians.XXZ(nqubits=5, dense=False)
circuit = ham.circuit(dt=1e-2)

result = circuit()

print(result.state())

```

II. TEBD algorithm and Translational layer

My current understanding is that the local unitaries we extract in the previous step will be required to start the algorithm. Sharing some of my understanding and my **doubts** below:

Compatibility:

- > Of what form are the local unitaries?
- > What do we require? (Summands of internally commuting parts which need to be individually exponentiated – is this correct, what exactly does this mean?)
- > The above two – are they the same type of object?
- > Does doing the above step result in a MPO as output?
- > If yes, how to make this MPO compatible with a quimb MPO type?

Computation steps:

- > Decompose the above MPO (First decomposition)
- > Decompose MPO again (Second decomposition)
- > What decomposition – is there a specific type that's employed? SVD
- > To the result of the 2 decompositions, say U, keep applying U
- > Is this correct?
- > What does it mean to "apply" U?
- > Will there be any inputs needed for the functions that do this, for eg. Bond dimension etc?
- > ***Request for a briefing on this to understand the algorithm and steps involved***

III. The output layer – my understanding

We are currently dealing with quimb backend. Meaning, our current output is a dense vector. Thus, output of using TEBD with quimb should be the final dense vector after the MPS goes through evolution.

So, this means, some bridging is to be done in the previous step to ensure compatibility with quimb?