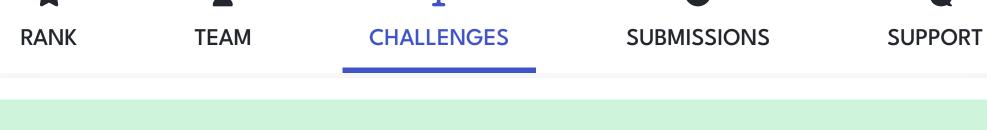
QHack

Quantum Coding Challenges



CHALLENGE COMPLETED View successful submissions ✓ Jump to code Collapse text **Sqy Trotter** 100 points

It's the year 22450. Sqynet, the most powerful quantum computer in the galaxy, has become conscious and has been taking over planets all over region III of the Milky Way. Zenda and

Backstory

Reece are the most skilled physicists of the Special Rebel Alliance. Their mission is to find a way to destroy Sqynet for good, using intelligence gathered throughout decades at the cost of many lives. To get started with their mission, Zenda and Reece seek to become familiar with how Sqynet applies quantum gates. Quantum computers do this through external interactions described

warm up with some simple calculations. **Introduction to Trotterization** The Hamiltonian H of a quantum system is the observable that measures its total energy. A

via Hamiltonians. Knowing that Sqynet is a spin-based quantum computer, Zenda and Reece

surprising result in Physics is that we can use this operator to calculate how a given quantum system evolves in time. An initial state $|\psi\rangle$ will, after a time t, evolve into $U(t)|\psi\rangle$, where

approximate U via

figure below.

for this problem.

Challenge code

 $U(t) = \exp(-iHt)$ is a unitary operator. The symbol exp denotes the matrix exponential, which isn't always easy to calculate. However, we can build quantum circuits that approximately apply U(t).

One method to do this is Trotterization. When the Hamiltonian is a sum

of a number
$$k$$
 of Hermitian operators H_i that do not necessarily commute, we can

 $H = \sum_{i=1}^k H_i$

 $U(t)pprox\prod_{i=1}^{n}\prod_{i=1}^{k}\exp(-iH_{i}t/n).$

Here,
$$n \in \mathbb{N}^+$$
 is known as the Trotterization depth. The larger n is, the better the approximation of U that we get. As a quantum circuit, the Trotterization of U reads as in the figure below.

Sqynet is a spin-based quantum computer, and it can be physically approximated via a spinchain model. A simplified version of a Hamiltonian that describes the interaction between two neighbouring spins is

 $H = \alpha X \otimes X + \beta Z \otimes Z.$

Zenda and Reece want to simulate time evolution for a time t under this Hamiltonian and

arbitrary parameters α and β . Your job is to help them out by implementing the

corresponding Trotterization of depth
$$n$$
. You may find the IsingXX and IsingZZ gates useful for this problem

You must complete the trotterize function to build the Trotterization of the Hamiltonian given above. You may not use qml.ApproxTimeEvolution or qml.QubitUnitary, but feel free to check your answer using this built-in PennyLane function!

Input As input to this problem, you are given: • alpha (float): The coefficient α of the $X \otimes X$ term in the Hamiltonian. • beta (float): The coefficient β of the $Z \otimes Z$ term in the Hamiltonian. ullet time (float): The period t over which the system evolves under the action of the Hamiltonian. • depth (int): The Trotterization depth n as explained above. Output This code will output a <code>list(float)</code> (list of real numbers) corresponding to the probabilities of measuring $|00\rangle, |01\rangle, |10\rangle$, and $|11\rangle$ (in that order) of the Trotterization circuit that you implement in PennyLane. The initial state of the circuit is $|00\rangle$ and all measurements are performed in the computational basis. If your solution matches the correct one within the given tolerance specified in Check (in this case, it's a relative tolerance of 1e-4), the output will be "Correct!" Otherwise, you will receive a "Wrong answer" prompt. Good luck! Code ? Help import json import pennylane as qml import pennylane.numpy as np dev = qml.device('default.qubit', wires = 2) @qml.qnode(dev) 6 7 def trotterize(alpha, beta, time, depth): """This quantum circuit implements the Trotterization of a Hamiltonian giv 9 of tensor products of X and Z Pauli gates. 10 11 Args: 12 alpha (float): The coefficient of the XX term in the Hamiltonian, as i 13 beta (float): The coefficient of the YY term in the Hamiltonian, as in time (float): Time interval during which the quantum state evolves und 14 15 depth (int): The Trotterization depth. 16 17 Returns: (numpy.array): The probabilities of each measuring each computational 18 $\Pi\Pi\Pi\Pi$ 19 20 # Put your code here # 22 23 # Return the probabilities 24 # These functions are responsible for testing the solution. 25 def run(test case input: str) -> str: dev = qml.device("default.qubit", wires=2) 26 ins = json.loads(test_case_input) 27 output = list(trotterize(*ins).numpy()) 28 29 30 return str(output) 31 32 def check(solution_output: str, expected_output: str) -> None: solution_output = json.loads(solution_output) 33 expected_output = json.loads(expected_output) 34 35 assert np.allclose(solution output, expected output, rtol=1e-4 36), "Your circuit does not give the correct probabilities." 37 38 tape = trotterize.qtape 39 40 names = [op.name for op in tape.operations] 41 42 43 assert names.count('QubitUnitary') == 0, "Can't use custom-built gates!" 44 45

```
assert names.count('ApproxTimeEvolution') == 0, "Your circuit is using the
46 test\_cases = [['[0.5,0.8,0.2,1]', '[0.99003329, 0, 0, 0.00996671]'], ['[0.9, \hfrac{1}{2}]]
                                                                                     47 \ for i, (input_, expected_output) in enumerate(test_cases):
        print(f"Running test case {i} with input '{input_}'...")
```

```
50 ~
        try:
            output = run(input )
51
52
53 ~
        except Exception as exc:
            print(f"Runtime Error. {exc}")
54
55
56 ~
        else:
            if message := check(output, expected_output):
57 ~
                print(f"Wrong Answer. Have: '{output}'. Want: '{expected_output}'.
58
59
```

Submit

Copy all Open Notebook Reset

print("Correct!")

49

60 ~

61

else: