# Basics of IBM Quantum Experience and QisKit

## Introduction to IBM Quantum Experience

IBM Quantum Experience, now part of the IBM Quantum Platform, provides access to IBM's quantum computing resources, including real quantum processors and simulators. Users can interact with these resources through graphical tools like the Quantum Composer or programmatically using Qiskit. The platform supports learning and experimentation with quantum algorithms, offering features such as:

- Access to superconducting qubit-based quantum processors.
- Tools for constructing and visualizing quantum circuits.
- Integration with Jupyter notebooks for advanced programming.
- A community forum and extensive documentation for support 13.

IBM's quantum processors use superconducting transmon qubits housed in dilution refrigerators. Users can execute quantum circuits either on simulators or on real hardware, enabling both theoretical exploration and practical experimentation 1.

## Programming with Qiskit

Qiskit is an open-source SDK for creating and running quantum programs. It provides tools for building circuits, simulating their behavior, and executing them on IBM's quantum hardware. Below are key aspects of programming with Qiskit:

### Circuit Construction & Visualization

Quantum circuits in Qiskit are created using the `QuantumCircuit` class. This involves defining qubits, applying gates, and visualizing the circuit. Here's an example:

python
```python
from qiskit import QuantumCircuit

# Create a circuit with 2 qubits
qc = QuantumCircuit(2)

# Apply a Hadamard gate on qubit 0
```

```python
qc.h(0)

# Apply a CNOT gate between qubit 0 (control) and qubit 1 (target)
qc.cx(0, 1)

# Visualize the circuit
print(qc.draw())
```

This code constructs a Bell state, demonstrating entanglement between two qubits. Qiskit also allows visualization in various formats, such as text-based diagrams or matplotlib plots 23.

## Running on Simulators vs. Real Hardware

Qiskit supports running circuits on both simulators and real quantum devices:

1. **Simulators**:
    - Simulators like `StatevectorSimulator` or `QASM Simulator` emulate quantum behavior using classical computation.
    - They provide noiseless results ideal for debugging and testing algorithms.
    - Example usage:
      python

```python
from qiskit import Aer, execute
backend = Aer.get_backend('statevector_simulator')
result = execute(qc, backend).result()
print(result.get_statevector())
```

    - 
    - Simulators are faster but lack the noise characteristics of real hardware 45.
2. **Real Hardware**:
    - Circuits are transpiled to match the topology and gate set of the target device.
    - Execution involves queuing, calibration processes, and noise mitigation techniques.
    - Example usage:
      python

```python
from qiskit import IBMQ
provider = IBMQ.load_account()
backend = provider.get_backend('ibmq_quito')  # Example backend
job = execute(qc, backend)
```

```python
result = job.result()
print(result.get_counts())
```

- Results reflect real-world imperfections like decoherence and gate errors 6.

By combining Qiskit's powerful programming tools with IBM Quantum's hardware capabilities, students can experiment with quantum algorithms in both idealized and realistic settings. This hands-on approach builds foundational skills for future advancements in quantum software development.

## Citations:


1. https://en.wikipedia.org/wiki/IBM_Quantum_Platform
2. https://docs.quantum.ibm.com/guides/construct-circuits
3. https://docs.quantum.ibm.com/guides
4. https://quantumcomputing.stackexchange.com/questions/22289/simulators-vs-real-backends
5. https://quantumpositioned.com/lesson-7-using-qiskits-simulators/
6. https://docs.quantum.ibm.com/guides/execute-on-hardware
7. https://docs.quantum.ibm.com/migration-guides/local-simulators
8. https://learning.quantum.ibm.com
9. https://learning.quantum.ibm.com/course/quantum-computing-in-practice/introduction
10. https://www.ibm.com/quantum/
11. https://www.youtube.com/watch?v=pwwuF-DaPk0
12. https://mediacenter.ibm.com/media/Start+quantum+programming+with+IBM+Quantum+Experience/1_qsxcangp
13. https://www.reddit.com/r/QuantumComputing/comments/1cjddqm/ibm_quantum_computing/
14. https://quantum.ibm.com
15. https://quantumcomputing.stackexchange.com/questions/9074/find-the-q-experiment-tutorial-for-beginner
16. https://www.reddit.com/r/QuantumComputing/comments/6nwvpr/those_who_have_used_ibms_quantum_experience_what/
17. https://www.reddit.com/r/QuantumComputing/comments/15q8mpa/is_ibm_quantum_experiences_freeopen_plan_enough/
18. https://learning.quantum.ibm.com/learning-path/getting-started-with-qiskit
19. https://quantumcomputing.stackexchange.com/questions/29782/quantum-computing-coding-setup-for-a-beginner
20. https://www.reddit.com/r/QuantumComputing/comments/1fn52it/how_to_use_an_ibm_quantum_computer_with_qiskit/
21. https://www.youtube.com/watch?v=dZWz4Gs_BuI
22. https://quantumcomputing.stackexchange.com/questions/38590/simple-basic-qiskit-example-program

23. https://github.com/Qiskit/qiskit-tutorials/blob/master/tutorials/circuits_advanced/03_advanced_circuit_visualization.ipynb
24. https://www.youtube.com/watch?v=Tk9LOL9--Y4
25. https://www.youtube.com/watch?v=NTplT4WnNbk
26. https://github.com/Qiskit/qiskit-tutorials/blob/master/tutorials/circuits/01_circuit_basics.ipynb
27. https://quantumcomputing.stackexchange.com/questions/15748/visualizing-nice-circuits-in-qiskit
28. https://www.sharetechnote.com/html/QC/QuantumComputing_BasicCircuit_Qiskit.html
29. https://www.reddit.com/r/QuantumComputing/comments/18apxws/upsides_of_real_quantum_computing_vs_simulated/
30. https://quantumcomputing.stackexchange.com/questions/24517/how-do-qiskit-simulators-generally-work
31. https://www.linkedin.com/pulse/quantum-simulators-vs-computers-when-use-vmbff
32. https://quantumzeitgeist.com/how-to-use-a-quantum-computer-with-qiskit/
33. https://docs.quantum.ibm.com/guides/simulate-with-qiskit-sdk-primitives
34. https://docs.quantum.ibm.com/guides/execution-modes
35. https://www.jonvet.com/blog/using-a-real-quantum-computer
36. https://discuss.pennylane.ai/t/could-quantum-advantage-be-achieved-using-quantum-simulator/1831
37. https://www.youtube.com/watch?v=KmMS0S6BMao
38. https://ionq.com/resources/the-value-of-classical-quantum-simulators
39. https://quantumcomputing.stackexchange.com/questions/5073/introspecting-quantum-circuit-execution-on-qiskit-aer-simulators
40. https://www.youtube.com/watch?v=pYD6bvKLI_c
41. https://quantumcomputingreport.com/ibm-quantum-experience-first-looks/
42. https://research.ibm.com/quantum-computing
43. https://www.ibm.com/think/topics/quantum-computing
44. https://www.ibm.com/quantum/pricing
45. https://learning.quantum.ibm.com/course/quantum-business-foundations/introduction-to-quantum-computing
46. https://blog.mlq.ai/quantum-programming-with-qiskit/
47. https://www.youtube.com/watch?v=93-zLTppFZw
48. https://www.youtube.com/watch?v=XzNP-WJyIqQ
49. https://docs.quantum.ibm.com/guides/visualize-circuits
50. https://quantumcomputing.stackexchange.com/questions/38267/what-is-an-introductory-book-or-resource-for-qiskit
51. https://qmunity.thequantuminsider.com/2024/06/28/qiskit-version-1-0-crash-course-how-to-build-a-quantum-circuit/
52. https://quantumcomputing.stackexchange.com/questions/28130/inference-on-real-hardware-using-a-pre-trained-quantum-model-on-a-simulator
53. https://in.mathworks.com/help/matlab/math/run-quantum-circuit-on-hardware-using-IBM.html
54. https://www.nersc.gov/assets/Uploads/10-Qiskit-Aer-GPU-Doi.pdf

55. https://www.reddit.com/r/QuantumComputing/comments/sezau8/whats_the_point_of_using_the_backends_in_qiskit/