Hypothesis

Not peer-reviewed version

Simulating the Deutsch-Jozsa Quantum Algorithm Using a Classical Wave-Based Quantum Simulator

Lefliti Mohamed *

Posted Date: 11 June 2025

doi: 10.20944/preprints202506.0710.v1

Keywords: quantum computing; Deutsch-Jozsa algorithm; wave-based simulation; quantum oracles; signal processing; quantum-classical analogies

Hypothesis

Simulating the Deutsch-Jozsa Quantum Algorithm Using a Classical Wave-Based Quantum Simulator

Lefliti Mohamed

Affiliation; leflitimohamed@gmail.com

Abstract: The Deutsch-Jozsa algorithm is one of the earliest quantum algorithms that demonstrates an exponential speedup over classical computation in determining whether a black-box function is constant or balanced. In this paper, we propose and implement a novel approach to simulate the Deutsch-Jozsa algorithm using a wave-based quantum simulator. Our method leverages classical waveforms to represent quantum states and uses signal processing techniques such as Fourier transforms and interference to emulate quantum operations like the Hadamard transform and oracle queries. We demonstrate how the simulator can determine the nature of an unknown function with high accuracy by analyzing wave similarity after applying the simulated quantum operations. This work serves as a stepping stone toward understanding quantum computing principles through classical analogs and opens new pathways for educational tools and simplified models of quantum behavior.

Keywords: quantum computing; Deutsch-Jozsa algorithm; wave-based simulation; quantum oracles; signal processing; quantum-classical analogies

---

1. Introduction

Quantum computing promises significant advancements in computational efficiency, particularly in problems involving search, optimization, and cryptography. Among the early demonstrations of quantum advantage is the Deutsch-Jozsa algorithm, which solves the problem of determining whether a black-box function is constant or balanced with only a single query.

While real quantum computers are still in development, simulators offer a valuable tool for education and experimentation. In this work, we introduce a **wave-based quantum simulator** — a model where quantum states are represented by sinusoidal waves and quantum gates are emulated via classical signal transformations.

This approach allows researchers and students to explore quantum principles using familiar tools from signal processing and linear algebra.

2. Background: The Deutsch-Jozsa Algorithm

The Deutsch-Jozsa problem involves determining whether a function f: {0,1}^n → {0,1} is:
- **Constant**: f(x) = 0 or f(x) = 1 for all x
- **Balanced**: f(x) = 0 for exactly half of the inputs and f(x) = 1 for the other half

Classically, up to 2^(n−1)+1 evaluations may be required to solve this problem. However, the quantum solution requires only one evaluation, leveraging quantum superposition and interference.

3. Wave-Based Quantum Simulation Framewor

3.1 Representation of Qubits

In our framework, each qubit state is represented by a sinusoidal waveform:

$$\psi(t) = A * \sin(2\pi f\, t + \varphi)$$

 doi:10.20944/preprints202506.0710.v1

where:
- f encodes the information about the basis state
- A is the amplitude
- φ is the phase

This allows us to simulate quantum states using classical signals.

3.2. Superposition via Hadamard Transform

A classical analog of the Hadamard gate is implemented using:

H(ψ) = ψ / sqrt(2) + shifted(ψ) / sqrt(2)

This mimics the generation of superposition states.

H (ψ) = { ψ } over { sqrt{2} } + { "shifted"(ψ) } over { sqrt{2} }

3.3. Oracle Implementation

Two types of oracles are defined:
- **Constant Oracle**: Flips the phase of the entire signal.
- **Balanced Oracle**: Flips the phase of half the signal (e.g., based on parity).

These simulate the behavior of quantum oracles in a classical environment.

3.4. Measurement

Measurement is simulated by comparing the final signal to the initial superposition using cross-correlation:

similarity = ∣correlation(final_signal, initial_signal)∣

If the similarity is above a threshold, the function is classified as **constant**; otherwise, it's **balanced**.

4. Implementation

4.1. Code Overview

We implemented the simulation in Python using NumPy and Matplotlib. Key components include:
- `number_to_wave`: Maps integers to unique sine waves.
- `hadamard_transform`: Emulates superposition.
- `oracle_constant`, `oracle_balanced`: Simulate quantum oracles.
- `deutsch_jozsa_oracle`: Full implementation of the algorithm.

```python
def deutsch_jozsa_oracle(oracle_func):
    # Generate superposed signals
    signals = [number_to_wave(i) for i in range(4)]
    combined_signal = np.sum(signals, axis=0)
    # Apply Oracle
    processed_signal = oracle_func(combined_signal)
    Final measurement
    similarity = np.abs(np.correlate(processed_signal, combined_signal))
    return "constant" if similarity > 0.9 else "balanced"
```

 doi:10.20944/preprints202506.0710.v1

4.2. Results

Trial	Oracle Type	Predicted Type	Result
1	Constant	Constant	good
2	Balanced	Balanced	good

The algorithm correctly identified the type of the unknown function in both cases.

5. Discussion

This simulation provides a powerful analogy between quantum mechanics and classical wave theory. Although not a true quantum computer, the wave-based model successfully reproduces key quantum phenomena such as:

- Superposition: Multiple states simultaneously represented
- Interference: Constructive/destructive effects used in decision-making
- Measurement: Similarity-based readout mimicking probabilistic collapse

However, limitations exist:

- Scalability: Representing large Hilbert spaces becomes computationally expensive.
- Precision: Real-valued signals lack complex phase advantages of true quantum systems.

Despite these, the model offers great value in **education and conceptual exploration**.

6. Conclusions

We have demonstrated a working simulation of the Deutsch-Jozsa algorithm using a wave-based quantum simulator. The results confirm that core concepts of quantum computing can be explored using classical analogs, offering a promising path for teaching and research.

Future work will focus on extending the framework to simulate more complex algorithms such as Grover's and Shor's algorithms within the same paradigm.

Acknowledgments: The authors would like to thank the open-source community for providing accessible tools and libraries such as NumPy, Matplotlib, and Jupyter Notebooks, which made this research possible.

Code source: https://deepnote.com/workspace/suptechnology-3f8add4a-002f-4363-84f2-b6b8dc2cfd6f/project/lefliti-mohameds-Untitled-project-b11209a9-6e69-4bb3-ba4b-99fb8595f2ef/notebook/Notebook-1-116e011a2f5b42319980284d0ad4de0e?utm_source=share-modal&utm_medium=product-shared-content&utm_campaign=notebook&utm_content=b11209a9-6e69-4bb3-ba4b-99fb8595f2ef

References

Deutsch, D., & Jozsa, R. (1992). Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London A*, 439(1907), 553–558.

Nielsen, M. A., & Chuang, I. L. (2000). *Quantum Computation and Quantum Information*. Cambridge University Press.

Mermin, N. D. (2007). *Quantum Computer Science: An Introduction*. Cambridge University Press.

doi:10.20944/preprints202506.0710.v1

Preskill, J. (2018). Quantum Computing in the NISQ era and beyond. *Quantum*, 2, 79.

Kaye, P., Laflamme, R., & Mosca, M. (2007). *An Introduction to Quantum Computing*. Oxford University Press.