QuantumSkynet: A High-Dimensional Quantum Computing Simulator

Andres Giraldo-Carvajal¹, Daniel A. Duque-Ramirez¹,
Jose A. Jaramillo-Villegas¹,²
¹ Universidad Tecnológica de Pereira, Pereira, Risaralda, Colombia
² Laboratory for Research in Complex Systems, Menlo Park, California, USA
E-mail: jjv@utp.edu.co

Septiembre 2020

Abstract. The use of classical computers to simulate quantum computing has been successful in aiding the study of quantum algorithms and circuits that are too complex to examine analytically. Current implementations of quantum computing simulators are limited to two-level quantum systems. Recent advances in high-dimensional quantum computing systems have demonstrated the viability of working with multi-level superposition and entanglement. These advances allow an agile increase in the number of dimensions of the system while maintaining quantum entanglement, achieving higher encoding of information and making quantum algorithms less vulnerable to decoherence and computational errors. In this paper, we introduce QuantumSkynet, a novel high-dimensional cloud-based quantum computing simulator. This platform allows simulations of qudit-based quantum algorithms. We also propose a unified generalization of high-dimensional quantum gates, which are available for simulations in QuantumSkynet. Finally, we report simulations and their results for qudit-based versions of the Deutsch–Jozsa and quantum phase estimation algorithms using QuantumSkynet.

1. Introduction

Quantum computing has made significant progress in recent years. The race to build a computer capable of solving problems that classical computers cannot is reaching an intense level [1, 2].

Cryptography [3, 4], drug design and discovery [5, 7], the study of complex molecular structures [8, 9], and optimization and financial forecasting [10, 11] are some areas that hope to benefit from the construction of a quantum computer. However, today’s quantum computers are highly susceptible to noise, which can lead to errors in the calculations they perform [12].

An emerging alternative that allows increased resistance to noise is high-dimensional quantum computing [13]. Working with quantum states that handle multiple dimensions
not only enables systems that are more robust to noise but also more secure systems with greater information capacity [14].

Different physical platforms can be used to implement high-dimensional quantum systems [15], such as photons [13, 16–20], trapped ions [21, 22], superconducting systems [23, 24], and molecules [25–27]. The practical application of these qudit-based systems is a very active research field with great potential.

One tool that is instrumental in the progress in this area is quantum computing simulators. With these simulators, it is possible to study quantum circuits and measure their performance in the presence of noise, which helps make decisions that can improve the design of quantum hardware [28].

Currently, different quantum computing simulators and frameworks, such as qHiPSTER [28], Quantum Composer [29], Strawberry Fields [30], Rigetti Computing [31], Libquantum [32], and Quirk [33], among others, allow the execution of quantum algorithms and circuits to be efficiently simulated based on qubits; however, there is currently no quantum computing tool that allows simulation for high-dimensional quantum circuits.

In this paper, we propose QuantumSkynet, the first high-dimensional quantum computing simulator, which will enable scientists to evaluate, analyze, and tune high-dimensional quantum algorithms in a cloud-based environment. Additionally, we propose a generalization for high-dimensional quantum gates, where we show a relationship between Weyl’s adjoint operator and high-dimensional Pauli gates. Furthermore, we propose a more generalized equation to describe high-dimensional Fourier quantum transform.

2. High-dimensional quantum gates

During the research and development of QuantumSkynet, a compendium and generalization of quantum gates were employed to build arbitrary high-dimensional quantum algorithms, as described in the following.

2.1. Weyl operators

Weyl operators are a type of gate that apply to a single qudit. They can be used to create states of maximum entanglement [34, 35]. These unitary operators form an orthonormal basis of a $d$-dimensional Hilbert–Schmidt space and are represented by $d^2$ matrices generated from the following expression:

$$W_{p,q} = \sum_{k=0}^{d-1} \omega^{kp} |k\rangle \langle k\oplus q|$$  \hspace{1cm} (1)

where $k\oplus q$ is equal to $(k+q) \mod d$, $\omega$ and is defined by $e^{(2\pi i)/d}$, and $p, q$ are subscripts that take values from the standard base of the Hilbert space of dimension $d$. 

2.2. Weyl’s adjoint operator

One of the analyses conducted in this study involved establishing a correspondence between Weyl’s adjoint operator,

$$W_{q,p}^\dagger = \sum_{k=0}^{d-1} \omega^{kq} |k \oplus p\rangle \langle k|$$

and the $I, X, Y, \text{ and } Z$ gates in high dimensions:

$$I^m = W_{0,0}^\dagger$$

$$X^m = W_{0,m}^\dagger$$

$$Z^m = W_{m,0}^\dagger$$

$$Y^m = i^{(m \% d)} W_{m,m}^\dagger$$

In this manner, it is possible to find the $I, X, Y, \text{ and } Z$ gates for any dimension raised to any exponent. The identity operator for high dimensions can be found using equation (2) by replacing $(q,p)$ with $(0,0)$:

$$I^m = W_{0,0}^\dagger$$

$$= \sum_{k=0}^{d-1} \omega^0 |k \oplus 0\rangle \langle k|$$

$$= \sum_{k=0}^{d-1} |k\rangle \langle k|$$

$$= \sum_{k=0}^{d-1} P_k$$

where $P_k$ is known as the projection operator, which is equal to $|k\rangle \langle k|$. By replacing $(q,p)$ with $(0,m)$ in equation (2), it is possible to find the expression that generalizes the $X$ Pauli matrix for high dimensions and raising to any exponent:
To find the generalized form of the $Z$ Pauli matrix, simply replace $(q,p)$ with $(m,0)$ in equation (2):

$$Z^m = W^\dagger_{m,0} = \sum_{k=0}^{d-1} \omega^k \langle k \oplus 0 | k \rangle$$

$$= \sum_{k=0}^{d-1} \omega^k \langle k | k \rangle$$

(6)

The $Y$ matrix can be viewed as a combination of the $X$ and $Z$ matrices. It is possible to find its generalized equation for high dimensions and raising to any exponent by using equation (2) again, replacing $(q,p)$ with $(m,m)$ and multiplying by a cyclical global phase of $i^{(m\%d)}$:

$$Y^m = i^{(m\%d)} W^\dagger_{m,m}$$

$$Y^m = i^{(m\%d)} \sum_{k=0}^{d-1} \omega^k \langle k \oplus m | k \rangle$$

(7)

In this manner, we find the $X^m$, $Y^m$, and $Z^m$ gates generalized to $d$ dimensions of the qubit-based $X$, $Y$, and $Z$ gates, respectively, raised to the $m$ power. Below is a summary of the gate generalizations described thus far:
2.3. **Controlled gate**

The \textit{CNOT} gate, which acts on qubits, can be generalized to high dimensions by the \textit{SUM} gate, which is defined by the following expression:

\[
\text{SUM} = \sum_{k=0}^{d-1} P_k \otimes X^k
\]  

This gate can also be seen as a particular case of the arbitrary generalized controlled gate:

\[
\text{CU} = \sum_{k=0}^{d-1} P_k \otimes U^k
\]  

where \( U \) can be \( X, Y, Z \), or any high-dimensional operator.

In this manner, the high-dimensional controlled gates available in QuantumSkynet are as follows:

\[
\text{Figure 2.} \text{ Multi-qudit generalized controlled gate and SUM gate. U is an arbitrary unit matrix, and } P_k \text{ is the projection operator equal to } |k\rangle \langle k|.
\]

2.4. **Generalized quantum Fourier transform**

The classic discrete Fourier transform maps a series of \( N \) complex numbers into another series of complex numbers as follows \cite{36}:
where $x = [x_0, ..., x_N - 1]^T \in \mathbb{C}^N$, $y = [y_0, ..., y_N - 1]^T \in \mathbb{C}^N$, $\omega_N = e^{\frac{2\pi i}{N}}$.

Similarly, the quantum Fourier transform acts on a qudit with quantum state of dimension $d$ (where $N = d$) and maps it to another $d$-dimensional quantum state according to the following expression:

$$|j\rangle \xrightarrow{F} \frac{1}{\sqrt{d}} \sum_{k=0}^{d-1} \omega^{kj} |k\rangle$$

where $j$ is a quantum state belonging to the standard base of a $d$-dimensional Hilbert space with $\omega = e^{\frac{2\pi i}{d}}$.

When $d$ is equal to 2, the quantum Fourier transform is the Hadamard gate acting on 1 or $N$ qubits. $d$ greater than 2 gives a generalized Hadamard gate that acts on 1 or $N$ qudits, which is known as $DFT$ in some papers [18, 19].

We grouped all different combinations in which the quantum Fourier transform can be applied for any number of qudits and any number of dimensions, which we call a Generalized QFT ($GQFT$). For this, it is necessary to cover all elements of the quantum state resulting from the interaction between these qudits. That is, the size $N$ of the vector that will be affected by the $GQFT$ will be $d^n$, which is the same size as the tensor product between the qudits affected by the gate:

$$|j\rangle \xrightarrow{GQFT} \frac{1}{\sqrt{d^n}} \sum_{k=0}^{d^n-1} \omega^{kj} |k\rangle$$

where $n$ refers to the number of qudits of $d$ dimensions that will be affected by the gate.

The following figure shows the proposed $GQFT$. 
As can be seen in the previous figure, the Hadamard gate for the case of qubits is a special case of the $QFT$ gate, while for the case of qudits, the $GH$ or Generalized Hadamard gate (commonly misnamed $DFT$) is a special case of the $GQFT$ gate.

Therefore, this is another gate available for use with QuantumSkynet:

$$|j\rangle \xrightarrow{GQFT} \frac{1}{\sqrt{d^n}} \sum_{k=0}^{d^n-1} \omega^{kj} |k\rangle$$

Figure 4. GQFT gate

3. Methods

QuantumSkynet was written using the C++ programming language. It is divided into two core classes, Qudit and QRegister, as described below.

3.1. Qudit class

Consider the following definition of a qudit:

$$|\Psi\rangle = \sum_{k=0}^{d-1} \alpha_k |k\rangle$$

where $d$ is the number of dimensions and $\alpha_k$ are the complex amplitudes that accompany the base states of the qudit. A class for the construction of qudits was implemented in the simulator as follows:
# include <vector>
# include <complex>

class Qudit {
private:
    vector<complex<double>> amplitudes;

public:
    Qudit(vector<complex<double>> coefficients) {
        amplitudes = coefficients;
    }
    vector<complex<double>> getAmplitudes() {
        return amplitudes;
    }
};

This class receives a vector of complex values that correspond to the amplitudes \( \alpha_k \) that make up the quantum state of the qudit. This vector is assigned to the private attribute amplitudes, which can be accessed through the getAmplitudes() function.

3.2. QRegister class

To store and process the input qudits and allow their interaction between themselves and other functions in the quantum circuit, the QRegister class was created to define the structure of the quantum register:

class QRegister {
private:
    int quditsDimensions;
    int quditsNumber;
    vector<Qudit> initialQudits;
    vector<complex<double>> statesVector;

public:
    QRegister(
        int dimensions,
        vector<vector<complex<double>>> initialQuditsState
    ) {
        quditsDimensions = dimensions;
        quditsNumber = initialQuditsState.size();
        if (quditsNumber == 1) {
            Qudit qudit(initialQuditsState[0]);
            initialQudits.push_back(qudit);
            statesVector = initialQuditsState[0];
```
The `statesVector` variable defined within the QRegister class stores the vector of amplitudes with the global quantum state of the circuit. This variable is modified by the application of the high-dimensional quantum gates supported by the simulator.

3.3. Cloud-based architecture

To allow access to the different functions of the simulator, cloud computing services provided by Amazon Web Services (AWS) were used. The following diagram shows the set of AWS-specific technologies that allowed the simulator to be deployed on the web:
As shown in the previous diagram, this project was deployed in the form of an application programming interface (API) to be used as a gateway to the data and functionalities of the simulator.

A Cloud9 instance was used to debug the code and configure the simulator deployment on the cloud. Cloud9 automatically creates an EC2 instance and manages the lifecycle of this instance as required.

From the EC2 instance, a Lambda function was created in C++ with the logic to process the requests received by the client and return a response from the simulator. An API was then created using the Amazon API Gateway service and integrated with the pre-built Lambda function.

To request information from the API, an HTTP request is sent from the client with the quantum circuit to be simulated in JSON format in the body of the request, and a response that also arrives in JSON format is expected.

4. Results and discussion

4.1. Quantum phase estimation algorithm

Quantum phase estimation is one of the most important subroutines in quantum computation [37]. This algorithm is a central part of many other quantum algorithms, including the Shor factoring algorithm.

The high-dimensional version of this algorithm, which was presented in [18], was studied and implemented in QuantumSkynet. The quantum circuit associated with the phase estimation algorithm (PEA) is shown below.
The $DFT$ gates are qudit Hadamard gates, which can be represented by the $GQFT$ gate for a qudit in QuantumSkynet, and the $U$ gate used to test this algorithm was the same as that was used as a proof of concept in [18], the generalized 3-dimensional $Z$ gate.

The eigen-phases that correspond to each of the eigen-states of the generalized gate $Z$ are as follows:

$$
\begin{align*}
\text{EigenState} & \quad |0\rangle \quad |1\rangle \quad |2\rangle \\
\text{True Phase, } \phi & \quad 0 \quad \frac{2\pi}{3} \quad \frac{4\pi}{3}
\end{align*}
$$

The simulation of the above circuit in QuantumSkynet was conducted as follows:

where $|\Psi\rangle$ represents each of the eigen-states $|0\rangle, |1\rangle y |2\rangle$ required to find the respective eigen-phases.

Below are the respective resulting states based on the vector of amplitudes and value $\tau$ equal to the measured qudit value with index 0 based on the vector of probabilities, calculated by QuantumSkynet for each eigen-state (the qubit at the bottom of the circuit is the least significant):
QuantumSkynet: A High-Dimensional Quantum Computing Simulator

| EigenState | 0 | 1 | 2 |
|------------|---|---|---|
| Result     | 00 | 11 | 22 |
| $\tau$     | 0 | 1 | 2 |

The phase $\phi$ is equal to $2\pi\tau/3$. Therefore, the values of the phases found by QuantumSkynet for the respective eigen-states of the 3-dimensional $Z$ matrix were as follows:

| EigenState | 0 | 1 | 2 |
|------------|---|---|---|
| True Phase, $\phi$ | 0 | $\frac{2\pi}{3}$ | $\frac{4\pi}{3}$ |

which are effectively the values of the phases presented in [18].

4.2. Deutsch–Jozsa algorithm

The Deutsch–Jozsa algorithm is a remarkable example that demonstrates quantum speed-up. This algorithm can determine whether a function $f(x)$ is constant or balanced using only one function evaluation.

![Figure 8. Deutsch–Jozsa algorithm](image)

In the case of qudits, the Hadamard gate is replaced by the GQFT gate, but the analysis to identify a function as constant or balanced is similar to that for the two-dimensional case; if each $x$-register qudit returns 0, it is a constant function; otherwise, the function is balanced [15].

As an example, the following high-dimensional quantum circuits were tested on QuantumSkynet to evaluate whether the functions $f(x)$ are constant or balanced:
For the previous circuit, the following probability vector was obtained for each qudit:

| Chance result | | 0 | 1 | 2 |
|---------------|---|---|---|
| q[0]          | 1 | 0 | 0 |
| q[1]          | 1 | 0 | 0 |
| q[2]          | 0.33 | 0.33 | 0.33 |

The probability of obtaining $|0\rangle$ for the qudits at the top of the circuit was 100%, which means that the function is constant. A similar occurrence is seen with the following circuit, where all the probability values for the two least significant qudits of the circuit are equal to 1:
Now we show two examples of quantum circuits in which the function $f(x)$ is balanced:

**Figure 11.** Deutsch–Jozsa algorithm on QuantumSkynet

Because the three least significant qudits have zero probability of being at $|0\rangle$, the function $f(x)$ is balanced.
In this last example, the function \( f(x) \) is also balanced because the least significant qudit has no chance of being \( |0\rangle \).

5. Conclusion

Due to the advantages of qudit-based circuits over qubit-based ones, high-dimensional quantum computing simulation will continue to be an area of high interest.

The calculations derived from the analysis of quantum circuits based on qudits can become complex and grow as the number of dimensions increases. In this study, we explored a very useful tool to easily and intuitively simulate high-dimensional quantum circuits.

QuantumSkynet opens to the possibility of helping extend qubit-based quantum circuits to high-dimensional circuits as well as exploring and analyzing new qudit-based quantum circuits.

As future work, we intend to add support for the distinction between pure and mixed states and allow the simulation of different levels of quantum decoherence.

References

[1] Mario Piattini, Guido Peterssen, and Ricardo Pérez-Castillo. “Quantum Computing: A New Software Engineering Golden Age”. In: ACM SIGSOFT Software Engineering Notes 45.3 (2020), pp. 12–14.

[2] Gideon Lichfield. Inside the race to build the best quantum computer on Earth. url: https://www.technologyreview.com/2020/02/26/916744/quantum-computer-race-ibm-google/ (accessed: 26.02.2020).
REFERENCES

[3] Dan Boneh and Mark Zhandry. “Secure signatures and chosen ciphertext security in a quantum computing world”. In: Annual Cryptology Conference. Springer. 2013, pp. 361–379.

[4] Kevin Marshall et al. “Continuous-variable quantum computing on encrypted data”. In: Nature communications 7.1 (2016), pp. 1–7.

[5] Yudong Cao, Jhonathan Romero, and Alán Aspuru-Guzik. “Potential of quantum computing for drug discovery”. In: IBM Journal of Research and Development 62.6 (2018), pp. 6–1.

[6] Saad M Darwish, Tamer A Shendi, and Ahmed Younes. “Chemometrics approach for the prediction of chemical compounds’ toxicity degree based on quantum inspired optimization with applications in drug discovery”. In: Chemometrics and Intelligent Laboratory Systems 193 (2019), p. 103826.

[7] Ting Zhou, Danzhi Huang, and Amedeo Caflisch. “Quantum mechanical methods for drug design”. In: Current topics in medicinal chemistry 10.1 (2010), pp. 33–45.

[8] Yudong Cao et al. “Quantum chemistry in the age of quantum computing”. In: Chemical reviews 119.19 (2019), pp. 10856–10915.

[9] Rongxin Xia and Sabre Kais. “Qubit coupled cluster singles and doubles variational quantum eigensolver ansatz for electronic structure calculations.” In: Quantum Science and Technology (2020).

[10] Roman Orus, Samuel Mugel, and Enrique Lizaso. “Quantum computing for finance: overview and prospects”. In: Reviews in Physics 4 (2019), p. 100028.

[11] Román Orús, Samuel Mugel, and Enrique Lizaso. “Forecasting financial crashes with quantum computing”. In: Physical Review A 99.6 (2019), p. 060301.

[12] Benjamin Nachman et al. “Unfolding quantum computer readout noise”. In: npj Quantum Information 6.1 (2020), pp. 1–7.

[13] Manuel Erhard et al. “Twisted photons: new quantum perspectives in high dimensions”. In: Light: Science & Applications 7.3 (2018), pp. 17146–17146.

[14] Daniele Cozzolino et al. “High-Dimensional Quantum Communication: Benefits, Progress, and Future Challenges”. In: Advanced Quantum Technologies 2.12 (2019), p. 1900038.

[15] Yuchen Wang et al. “Qudits and high-dimensional quantum computing”. In: arXiv preprint arXiv:2008.00959 (2020).

[16] Nurul T Islam et al. “Provably secure and high-rate quantum key distribution with time-bin qudits”. In: Science advances 3.11 (2017), e1701491.

[17] Poolad Imany et al. “50-GHz-spaced comb of high-dimensional frequency-bin entangled photons from an on-chip silicon nitride microresonator”. In: Optics express 26.2 (2018), pp. 1825–1840.

[18] Hsuan-Hao Lu et al. “Quantum Phase Estimation with Time-Frequency Qudits in a Single Photon”. In: Advanced Quantum Technologies 3.2 (2020), p. 1900074.
REFERENCES

[19] Poolad Imany et al. “High-dimensional optical quantum logic in large operational spaces”. In: *npj Quantum Information* 5.1 (2019), pp. 1–10.

[20] Joseph M Lukens. “Quantum information processing with frequency-bin qubits: progress, status, and challenges”. In: *CLEO: Applications and Technology.* Optical Society of America. 2019, JTu4A–3.

[21] Pei Jiang Low et al. “Practical trapped-ion protocols for universal qudit-based quantum computing”. In: *Physical Review Research* 2.3 (2020), p. 033128.

[22] Brendan Bramman. “Measuring Trapped Ion Qudits”. MA thesis. University of Waterloo, 2019.

[23] EO Kiktenko et al. “Single qudit realization of the Deutsch algorithm using superconducting many-level quantum circuits”. In: *Physics Letters A* 379.22-23 (2015), pp. 1409–1413.

[24] Xinsheng Tan et al. “Experimental observation of tensor monopoles with a superconducting qudit”. In: *Physical Review Letters* 126.1 (2021), p. 017702.

[25] Rahul Sawant et al. “Ultracold polar molecules as qudits”. In: *New Journal of Physics* 22.1 (2020), p. 013027.

[26] Wolfgang Wernsdorfer and Mario Ruben. “Synthetic Hilbert space engineering of molecular qudits: Isotopologue chemistry”. In: *Advanced Materials* 31.26 (2019), p. 1806687.

[27] Eufemio Moreno-Pineda et al. “Molecular spin qudits for quantum algorithms”. In: *Chemical Society Reviews* 47.2 (2018), pp. 501–513.

[28] Gian Giacomo Guerreschi et al. “Intel Quantum Simulator: A cloud-ready high-performance simulator of quantum circuits”. In: *Quantum Science and Technology* 5.3 (2020), p. 034007.

[29] Shaeema Zaman Ahmed et al. “Quantum composer: A programmable quantum visualization and simulation tool for education and research”. In: *American Journal of Physics* 89.3 (2021), pp. 307–316.

[30] Nathan Killoran et al. “Strawberry fields: A software platform for photonic quantum computing”. In: *Quantum* 3 (2019), p. 129.

[31] Eyob A Sete, William J Zeng, and Chad T Rigetti. “A functional architecture for scalable quantum computing”. In: 2016 IEEE International Conference on Rebooting Computing (ICRC). IEEE. 2016, pp. 1–6.

[32] B Butscher and H Weimer. *The libquantum library.*

[33] *Quirk Quantum Simulator.* [https://algassert.com/quirk](https://algassert.com/quirk) Accessed: 2021-04-27.

[34] Reinhold A Berthmann and Philipp Krammer. “Bloch vectors for qudits”. In: *Journal of Physics A: Mathematical and Theoretical* 41.23 (2008), p. 235303.

[35] Bernhard Baumgartner, Beatrix Hiesmayr, and Heide Narnhofer. “A special simplex in the state space for entangled qudits”. In: *Journal of Physics A: Mathematical and Theoretical* 40.28 (2007), p. 7919.
[36] Daan Camps, Roel Van Beeumen, and Chao Yang. “Quantum Fourier Transform Revisited”. In: arXiv preprint arXiv:2003.03011 (2020).

[37] Stefano Paesani et al. “Experimental Bayesian quantum phase estimation on a silicon photonic chip”. In: Physical review letters 118.10 (2017), p. 100503.