

# Implementing the Klein Gordon Equation on a Quantum Computer

Desiree Vogt-Lee, 44354471

10 June, 2019

All components of this project including code, Jupyter Notebooks, LaTeX files and images can be found in my [Github repository](#), please check this out for reference.

## 1 The Klein Gordon Equation

The Klein Gordon equation set out to unify quantum mechanics and special relativity and ultimately led to the creation of the Dirac equation from which quantum field theory was born.

The Klein Gordon (KG) equation is unable to describe a particle as the Schrodinger equation does due to there being no conserved probability.

$$\frac{1}{c^2} \frac{\partial^2}{\partial t^2} \psi - \nabla^2 \psi + \frac{m^2 c^2}{\hbar^2} \psi = 0 \quad (1)$$

The Schrodinger equation is a first order equation in the time derivative where its' wavefunction is able to predict its future evolution. The Klein Gordon equation on the other hand is second order in the time derivative and the boundary condition for the field must be given to work out the field evolution. The KG equation also allows energy to be a negative value which can't be eliminated by limiting to only positive initial conditions due to run off effects that result in particles with  $E < 0$ . The Dirac equation was Dirac's attempt to create an equation that had a first order time derivative and was relativistic [1].

The Klein paradox arises from using the Dirac equation to solve the problem of electron scattering from a barrier potential. Under non-relativistic circumstances, the electron would tunnel into the barrier. However if the barrier potential is greater than  $mc^2$ , the particle is able to penetrate the barrier as if it were transparent [2].

This report will be based off a paper titled *Quantum Simulation of Klein Gordon Equation and Observation of Klein Paradox in IBM Quantum Computer* by Manik, K et al. (2018) [3] which aimed to simulate the time dependent Klein Gordon equation in a barrier potential on a quantum computer to see the tunneling of both the particle and anti-particle with the Klein Paradox.

The present report aims to replicate results, justify decisions made in the original publication and offer criticism and suggestions.

## 1.1 Klein Gordon Equation in Unitary Form

To implement the KG equation on a quantum computer, it must be written in terms of unitary operators [3]. To do this we start with the equation for a Hamiltonian in terms of mass of a particle ( $m$ ), momentum operator ( $\hat{p}$ ), potential ( $V$ ), speed of light ( $c$ ), identity matrix ( $I$ ) and Pauli matrices ( $\sigma_{1,2,3}$ ):

$$\hat{H} = \frac{\sigma_3 + i\sigma_2}{2m} \hat{p}^2 + \sigma_3 mc^2 + I\hat{V} \quad (2)$$

By using the vector:

$$\varphi = \begin{bmatrix} \phi \\ \chi \end{bmatrix}$$

We can solve two simultaneous equations, resulting in:

$$\begin{aligned} \hat{H}_1 \phi &= \frac{\hat{p}^2}{2m} (\phi + \chi) + (mc^2 + \hat{V}) \phi \\ \hat{H}_2 \chi &= -\frac{\hat{p}^2}{2m} (\phi + \chi) + (-mc^2 + \hat{V}) \chi \end{aligned}$$

These equations then need to be represented as a time evolution equation which can be done by taking the second order Trotter-Suzuki (TS) decomposition [4] of the Hamiltonian:

$$|\psi(x, t + \delta)\rangle = e^{-i\hat{H}_1 t} |\psi(x, t)\rangle$$

The TS decomposition equation simplifies calculations of matrix exponentials that are required to solve the time dependent Schrodinger equation by proposing a way to rewrite exponentials to unitary operators allowing for less intensive computation:

$$\begin{aligned} e^{-i\hat{H}_1 t} &= e^{-i\left(\frac{\hat{p}^2}{2m}(\square + \square) + \hat{V}_1\right)t} \\ e^{-i\hat{H}_2 t} &= e^{i\left(\frac{\hat{p}^2}{2m}(\square + \square) - \hat{V}_2\right)t} \end{aligned}$$

Where  $\square$  is the component of the wavefunction. Each momentum operator can be expressed using the quantum Fourier transform and its inverse in momentum space, which is explained in the IBM Research Report (1994) [5] to result in the final unitary operators that can be implemented on the quantum computer:

$$e^{-i\hat{H}_1 t} = F^{-1} e^{-i\hat{p}_p^2 \square} . F . F^{-1} . e^{-i\hat{p}_p^2 \square} F . e^{-iV_1 \sigma_z t} \quad (3)$$

$$e^{-i\hat{H}_2 t} = F^{-1} e^{i\hat{p}_p^2 \square} . F . F^{-1} . e^{i\hat{p}_p^2 \square} F . e^{-iV_2 \sigma_z t} \quad (4)$$

## 2 Quantum Computing

Quantum computing is the melding together of quantum mechanics and classical computation. The computer chips used in quantum computers use the same fundamental components as that of their classical counterparts, however are pieced together in a way which allow quantum-bits, or qubits, to be used instead of bits.

A fundamental flaw in the current design of quantum computers is the large amount of error that is created due to noise and decoherence, which increases with the number of qubits used. This severely limits the computation that can be performed with a negligible amount of error, as at some point, the current error correcting codes will have little effect.

Many believe that quantum computation is the future. Due to the superposition and entanglement properties these qubits can possess, all sorts of optimisation and new algorithms can be created, which would have significant impact on many fields of technology. For example, modern cryptography is centered around the assumption that creating incredibly long numbers take current computers an impossibly long time to factorise. Using Shor's algorithm, a quantum computer would be able to factorise a number in polynomial time, exponentially faster than the best classical algorithm, and thus being able to break modern cryptography. [6]

At present, this is all merely a premonition; as although there exists many quantum algorithms, the error is too great and the number of qubits too little to be able to use them. I believe that applying the Klein Gordon equation on a quantum computer is useful in showing the present capabilities of this new technology as well as just being a really interesting application.

### 2.1 Gates

To modify the state that a qubit is in, a gate or procession of gates can be applied to it. The quantum gates are similar to the logic gates present in classical computing, but work with the quantum nature of the qubit. To easily conceptualise the effect of a quantum gate, a geometric representation called the Bloch sphere is used, where the poles of the sphere represent the standard basis vectors  $|0\rangle$  and  $|1\rangle$ .

In classical computation, the NOT gate simply flips a given bit state: a 1 will be transformed to a 0 and vice versa. In quantum computing there is an equivalent X gate which when applied to a qubit in the  $|0\rangle$  state, performs a rotation about the x axis on a Bloch sphere by  $\pi$  radians resulting in the  $|1\rangle$  state when measured (for more detailed explanation on the fundamental gates and terminology, please see my [blog post](#)).

Here I will briefly explain the gates that will be used to create the circuit that implements the Klein Gordon equation (minus those explained in my blog post):

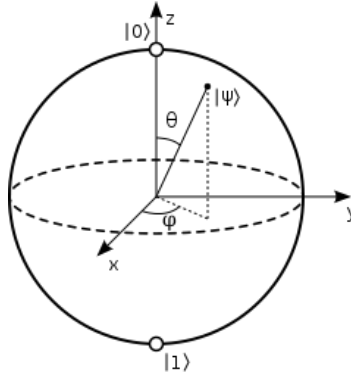


Figure 1: Bloch Sphere [7]

- **T** T: performs a  $\frac{\pi}{4}$  rotation about the z axis, which is equivalent to a quarter of one Z gate operation. This gate acts on the phase of the qubit and is represented by the matrix:

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$$

- **T<sup>†</sup>**  $T^\dagger$ : simply the opposite of the T gate and applies a  $-\frac{\pi}{4}$  rotation about the z axis [8].
- **U1**  $U_1$ : performs a unitary rotation on the state vector where the continuous phase can be specified. It is represented by the following matrix and as can be seen, is of a similar to that of the T gate, suggesting that a  $U_1(\frac{\pi}{4})$  gate is equivalent to a T gate.



$$U_1(\lambda) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{pmatrix}$$


- **U3**  $U_3$ : used to perform arbitrary unitary operation, where the  $\theta$ ,  $\psi$  and  $\lambda$  position of the vector on the Bloch sphere can be modified to any value. This is the most general form of a single qubit unitary operation and is represented in matrix form as [9]:

$$U = \begin{pmatrix} \cos(\theta/2) & -e^{i\lambda} \sin(\theta/2) \\ e^{i\phi} \sin(\theta/2) & e^{i\lambda+i\phi} \cos(\theta/2) \end{pmatrix}$$

- R

$$R = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix}$$

-  SWAP: operates on a pair of qubits and swaps their state.
-  Measurement: used at the end of the circuit to measure the outcome of the states in the z basis.

-  Barrier: directives which inform the compiler not to optimise across barriers. They serve the purpose of keeping gates, especially measurement gates, in their place and are as well used as a visualisation tool to separate parts of the circuit.

### 3 The Circuits

Equations xxxxxx and xxxxxxxx can be broken up and implemented into a circuit. As we are attempting to recreate results from Manik, K. (2018) [3], we will take  $V_1 = 12$  and  $V_2 = 10$ . No justification is given as to why these potentials are used. The circuits described in Figures xxxxxx and xxxxxxxxxx have their components separated by barriers and we will refer to each section in the order they appear starting at section 1 where the U1 gate appears.

The first and last U1 gates (section 1 and 8) implement the potential term in the Klein Gordon equation. Then as per Equations xxxxx and xxxxxxxxx, inverse QFTs appear at sections 2 and 5 of the circuit and regular QFTs appear at sections 4 and 7 (explanation of these circuit implementations of the QFT can be seen shortly).

In the code there is an option to choose to run the circuit on a real IBM quantum computer or the simulator.

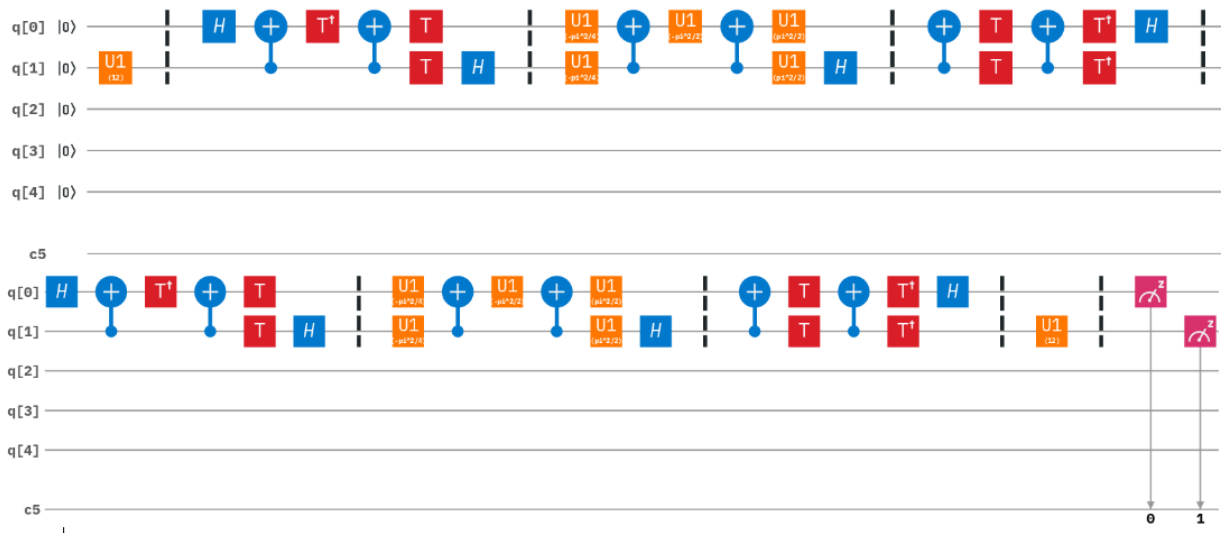
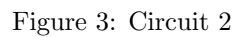


Figure 2: Circuit 1

### 3.1 Fast Fourier Transform

The Fourier transform is able to decompose a function into its constituent functions, which has many practical applications including signal processing, analysis of differential equations and spectroscopy [10]. One particular variation of Fourier transform of interest is the discrete Fourier transform (DFT) as


$$y_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi kn/N} \quad (5)$$

$$\begin{aligned}
&= \sum_{m=0}^{N/2-1} x_{2m} \cdot e^{-i2\pi k(2m)/N} + \sum_{m=0}^{N/2-1} x_{2m+1} \cdot e^{-i2\pi k(2m+1)/N} \\
&= \sum_{m=0}^{N/2-1} x_{2m} \cdot e^{-i2\pi km/(N/2)} + e^{-i2\pi k/N} \sum_{m=0}^{N/2-1} x_{2m+1} \cdot e^{-i2\pi km/(N/2)}
\end{aligned}$$

### 3.2 Quantum Fourier Transform

6

$$\alpha_j |j\rangle \rightarrow \alpha_j \left( \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i \frac{kj}{N}} |k\rangle \right) \quad (6)$$

As in with standard DFT, a delta function will be transformed to a sine function using a QFT. For a two qubit system, with one qubit in the  $|1\rangle$  state and the other in the  $|0\rangle$  state,  $|10\rangle$  will transform into the following superposition:

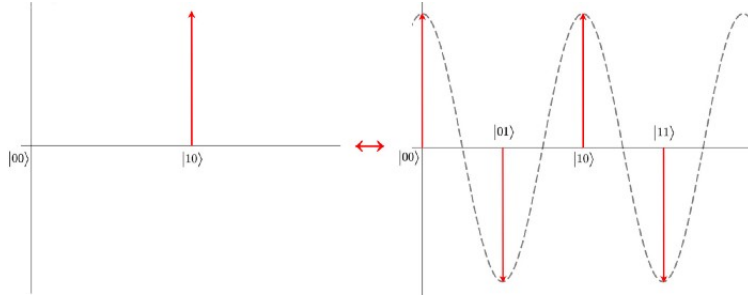


Figure 4: Delta function transformed to a sine function of two qubits:  $|10\rangle \rightarrow |00\rangle - |01\rangle + |10\rangle - |11\rangle$  [13]

Just as the FFT is performed iteratively on subproblems, so is the QFT: Equation xxxxxxxxxxxxxxxx can be broken down into the following expression which allows for easier understanding of how the equation can be implemented as gates on a quantum circuit.

$$|j_1 \dots j_n\rangle \rightarrow \frac{(|0\rangle + e^{2\pi i 0 \cdot j_n} |1\rangle) (|0\rangle + e^{2\pi i 0 \cdot j_{n-1} j_n} |1\rangle) \dots (|0\rangle + e^{2\pi i 0 \cdot j_1 j_2 \dots j_n} |1\rangle)}{2^{n/2}} \quad (7)$$

The first Hadamard in the following circuit representation of Equation xxxxxxxxxxxx puts the system into the superposition state required (the  $1/\sqrt{2}$ ) and the following multiplications are implemented using the R controlled phase gates. The pattern apparent in the circuit indicates the recursive nature of the algorithm. The swap gate can be applied either at the beginning or end of the system as it is just required that the states of the two qubits are swapped prior to measurement and its position has no effect on the outcome.

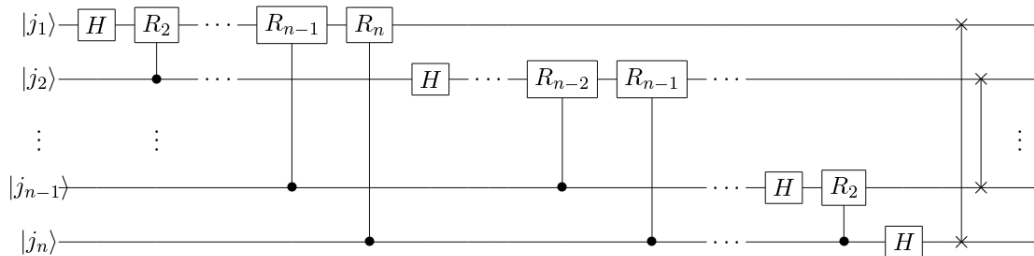


Figure 5: QFT circuit implemented on an n qubit system [14].

As our implementation of the Klein Gordon equation only uses two qubits, the quantum Fourier transform on two qubits is represented by the following circuit.

Like the classical Fourier transform, the quantum Fourier transform is reversible as well as unitary.

## 4 Exploration

The time step interval ( $t$ ) in Equation xxxxxx and xxxxxxxx was not provided in the original paper. I reached out to the author who was unable to provide the original values for these time steps however noted that the time started at  $t=0$  and increased at a very small rate linearly. After trial and error of various time steps, the time step that yielded the most similar heatmap was  $t=0.02$ . Despite the long time spent trying to work out the time step, the resulting heatmaps were significantly different to those gained in the reference paper (see Figure xxxxxxxx).

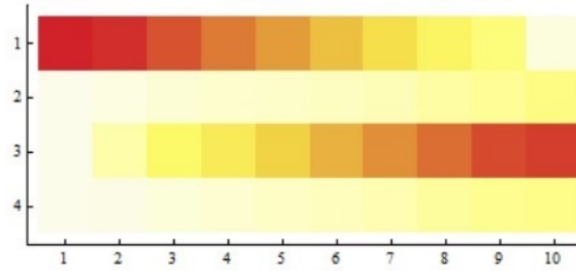


Figure 6: Circuit 1 heatmap results from original paper [3].

As the timestep used by the author was most likely chosen to best show the evolution of the states over time, it is not significantly important to get it exactly correct. However even with smaller timesteps/greater evolution time (these can be easily modified in the code [?]), the pattern found in the original heatmap was unable to be replicated which might suggest an error in the implementation of the KG equation.

The original paper used the IBM Q simulator exclusively, speculatively due to the ideal, consistent results and fast output as opposed to the real device. In the code I have written for these circuits [?], there is a toggle to run the circuit on either the IBM Q simulator or the IBMQx2 two qubit quantum computer. Below are the results of running the exact circuit from the reference paper on both devices.

As seen from Figure xxxxxxxxxxxx, there is significant difference between the results from the simulated and real devices. Although some difference is to be expected due to an increase of error and uncertainty in real qubits, such a large discrepancy was unforeseen.

The QFT and IQFT implementations in the circuit by Manik, K (2018) [3] is different that mentioned in Section



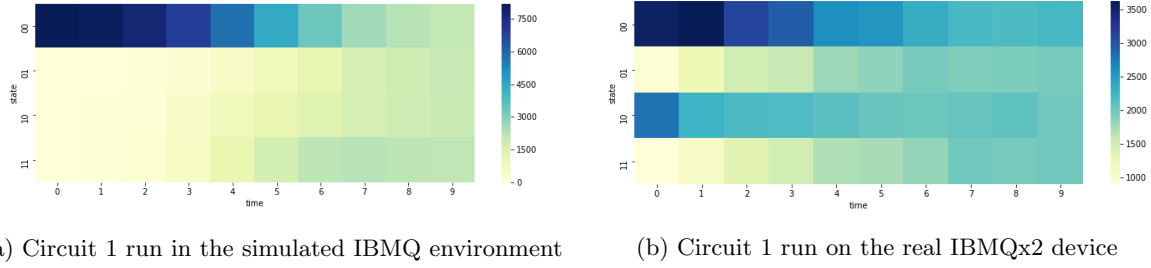


Figure 7: Circuit 1 run on different devices.

## 5 Results

The expected results of the implementation of the Klein Gordon equation is that for a pair of qubits at  $t=0$ , both qubits start in a  $|00\rangle$  state and cross the barrier potential as expected by the Klein paradox. In the reference paper results this phenomenon is clearly observed as the particle and anti-particle tunnel at the high and low potentials, leading to the final state of the particle being  $|10\rangle$ . The author of the reference paper claims that the Klein paradox and tunneling effects are due to the chosen strong potential values and setting  $mc^2 = 1$ .

In this report, I was unable to reproduce the exact results gathered in the Manik, K. (2018) [3] paper. Using the IBMQ simulator, the qubit state started at  $|00\rangle$  and ended spread out across the range of possible states however with slightly stronger appearance in the  $|11\rangle$  state.

maybe you can't get tunneling with ideal simulator

## References

- [1] I. McCulloch, "Applications of lagrangian field theory." [https://learn.uq.edu.au/bbcswebdav/pid-4474235-dt-content-rid-18539051\\_1/xid-18539051\\_1](https://learn.uq.edu.au/bbcswebdav/pid-4474235-dt-content-rid-18539051_1/xid-18539051_1).
- [2] O. Klein, "Die Reflexion von Elektronen an einem Potentialsprung nach der relativistischen Dynamik von Dirac," *Zeitschrift fur Physik*, vol. 53, pp. 157–165, Mar. 1929.
- [3] M. Kapil, B. K. Behera, and P. K. Panigrahi, "Quantum Simulation of Klein Gordon Equation and Observation of Klein Paradox in IBM Quantum Computer," *arXiv e-prints*, p. arXiv:1807.00521, Jul 2018.
- [4] H. F. Trotter, "On the product of semi-groups of operators," *Proceedings of the American Mathematical Society*, vol. 10, no. 4, pp. 545–551, 1959.
- [5] D. Coppersmith, "An approximate Fourier transform useful in quantum factoring," *arXiv e-prints*, pp. quant-ph/0201067, Jan 2002.

- [6] H. Reich, “How quantum computers break encryption — shor’s algorithm explained.” <https://www.youtube.com/watch?v=lvTqbM5Dq4Q>, 2019.
- [7] S. Meister, “Bloch sphere.” [https://en.wikipedia.org/wiki/File:Bloch\\_sphere.svg](https://en.wikipedia.org/wiki/File:Bloch_sphere.svg), 2009.
- [8] IBM, “Introducing qubit phase.” [https://quantumexperience.ng.bluemix.net/proxy/tutorial/full-user-guide/002-The\\_Weird\\_and\\_Wonderful\\_World\\_of\\_the\\_Qubit/003-Introducing\\_qubit\\_phase.html](https://quantumexperience.ng.bluemix.net/proxy/tutorial/full-user-guide/002-The_Weird_and_Wonderful_World_of_the_Qubit/003-Introducing_qubit_phase.html), nd.
- [9] IBM, “Advanced single-qubit gates.” [https://quantumexperience.ng.bluemix.net/proxy/tutorial/full-user-guide/002-The\\_Weird\\_and\\_Wonderful\\_World\\_of\\_the\\_Qubit/004-advanced\\_qubit\\_gates.html](https://quantumexperience.ng.bluemix.net/proxy/tutorial/full-user-guide/002-The_Weird_and_Wonderful_World_of_the_Qubit/004-advanced_qubit_gates.html), nd.
- [10] P. Bevelacqua, “Fourier transforms, fourier transform, tutorial, fourier series, fourier.” <http://www.thefouriertransform.com/applications/fourier.php>, 2010.
- [11] J. Cooley and J. Tukey, “An algorithm for the machine calculation of complex fourier series,” *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.
- [12] J. VanderPlas, “Understanding the fft algorithm.” <https://jakevdp.github.io/blog/2013/08/28/understanding-the-fft/>, 2013.
- [13] J. Hui, “Qc — quantum fourier transform.” [https://medium.com/@jonathan\\_hui/qc-quantum-fourier-transform-45436f90a43](https://medium.com/@jonathan_hui/qc-quantum-fourier-transform-45436f90a43), 2018.
- [14] J. V. Gael, “The role of interference and entanglement in quantum computing.” <http://pages.cs.wisc.edu/~dieter/Papers/vangael-thesis.pdf>, ”May” 2005.