# The Quantum Abacus: A Break-Even Point

Dan-Adrian German
Luddy School of Informatics, Computing and Engineering (SICE)
Department of Computer Science
Indiana University, Bloomington, IN, USA

"It always bothers me that, according to the laws as we understand them today, it takes a computing machine an infinite number of logical operations to figure out what goes on in no matter how tiny a region of space, and no matter how tiny a region of time. How can all that be going on in that tiny space? Why should it take an infinite amount of logic to figure out what one tiny piece of space/time is going to do? So I have often made the hypothesis that ultimately physics will not require a mathematical statement, that in the end the machinery will be revealed, and the laws will turn out to be simple, like the chequer board with all its apparent complexities."

— Richard Feynman, The Character of Physical Law

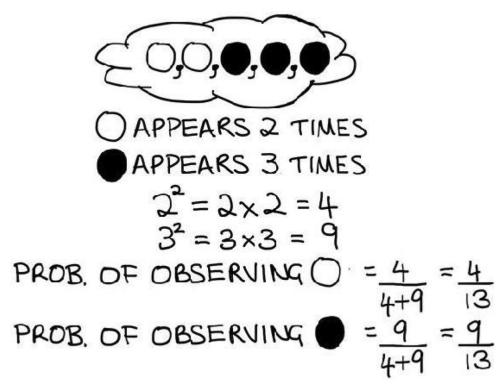
***Abstract*—In 2017 Terry Rudolph proposed a method of teaching quantum mechanics and quantum computing using only the simple rules of arithmetic to students as early as sixth grade. The method is incredibly effective and in a series of papers we showed how we use it to introduce superposition, phase, interference and entanglement with virtually no mathematical overhead. Furthermore we showed that a complete eight week introductory course (for computer science sophomores) has been built around this approach with the following milestones: quantum gates and circuits, phase kickback, the Deutsch-Josza algorithm, Bernstein-Vazirani and the extended Church-Turing thesis, the GHZ game and quantum teleportation. There is general consensus that the actual mathematics behind quantum computation is an inevitable and desirable destination for our students. But for those students that lack an adequate mathematical background (HS and younger students) one can reliably use Terry's method (i.e., computing with misty states, also referred to here as The Quantum Abacus) to communicate a visual and entirely operational understanding of key quantum computing concepts without resorting to complex numbers or matrix multiplication. Here we present concrete evidence that the approach can also create a genuine bridge to the actual mathematics behind quantum computation. We start with superdense coding and Grover's algorithm (to illustrate how effective the original system is) then we identify an elementary break-even point when creating a W-entangled state. Terry's abacus is based on a paper by Shih that Toffoli plus Hadamard gates are universal. To create the W-entangled state we need to accommodate rotations and we must use controlled-Hadamard gates. And this is what allows for an elementary break-even point in the formalism: A Hadamard gate controlled by the output of another Hadamard gate breaks the ubiquitous symmetry in Terry's system, and from then on one has to carry around (i.e., specify) the actual probability amplitudes in misty states. This means that students can proceed to developing, in parallel, with (extended) misty states and Dirac notation. And after crossing that bridge we have an entirely conventional Quantum Computation course, but the intuition we acquired while computing with misty states remains with us.**

## I. The Quantum Abacus

A classical bit is much like the side of a coin that sits on the table: it is either head or tails (i.e., 0 or 1) no matter how many times you look at it. A quantum bit is more like a coin that's spinning on the table: the only thing you can hope to know is the set of outcomes and their associated probabilities.

A qubit is written as $\alpha|0\rangle+\beta|1\rangle$ with $\alpha, \beta \in \mathbb{C}$ and such that $|\alpha|^2+|\beta|^2 = 1$. This linear superposition specifies the expected outcomes (i.e., $|0\rangle$ and $|1\rangle$) the probability amplitudes ($\alpha$ and $\beta$) and states that the two probabilities should add up to 1.

In "Q is for Quantum" ([7], [8]) Terry Rudolph "teaches [quantum mechanics] to an audience presumed only to know basic arithmetic." The book uses a simple (but very effective) string-rewriting technique that starts $|0\rangle = \bigcirc$ and $|1\rangle = \bullet$ and represents[1] superposition as "misty states":




Here $\{\bigcirc, \bigcirc, \bullet, \bullet, \bullet\} = \frac{2}{\sqrt{13}}|0\rangle + \frac{3}{\sqrt{13}}|1\rangle$, clearly.

A misty state is just an un-normalized quantum state. Note that a state like $\{\bigcirc, \bigcirc, \bullet, \bullet, \bullet\}$ is not written in the book as $\{2 \times \bigcirc, 3 \times \bullet\}$ or $\{2 \cdot \bigcirc, 3 \cdot \bullet\}$ or even $\{2\bigcirc, 3\bullet\}$ by choice. We refer to the misty state formalism from [7] as the "Quantum Abacus". The quantum abacus is simply a different way of representing a subset of the quantum formalism which is actually universal for quantum computing, a fact first proven by Shih [9]. Syntactically, within the abacus, misty states are shown always only in white and black and no numbers are used (as coefficients) anywhere in the formalism. Since misty states always contain a whole number of black and white balls, it follows that $\frac{1}{\sqrt{3}}|0\rangle+\frac{\sqrt{2}}{\sqrt{3}}|1\rangle$ does not admit a representation as a misty state, something that will prove very important shortly.

[1] This diagram appears on p. 83 in the book.

The remaining of the paper is structured as follows: we first introduce one- and two-qubit gates and prove some basic properties using the misty state formalism to better describe its syntax and to give the reader a feel for how it works. We then present superdense coding and the Grover search algorithm using this formalism[2] to show how incredibly convenient and efficient it is. In the process we point out earlier papers where we demonstrated in detail how to use the misty state formalism to prove (trace, understand) phase kickback, entanglement (both Bell and GHZ states) and the Deutsch-Josza algorithm. Through $\sqrt{\mathtt{NOT}} = \mathtt{HSH}$ and $\mathtt{S} = \sqrt{\mathtt{Z}}$ we propose an extension to the misty formalism and show that in certain circumstances (i.e., the break-even point) the extension becomes a necessity. We conclude the paper by using the extended formalism to trace the formation of a three qubit W-entangled state and introduce arbitrary rotations. We give complete references to papers where we have used the extended formalism to present in detail quantum phenomena like teleportation and the GHZ game. All our examples involve a small number of qubits so our calculations can be readily tested via Qiskit simulations. The overall arching goal of this paper is to show how we can lead students with limiting backgrounds in math to the actual math used in Quantum Information Science (QIS) and Quantum Computation (QC). We argue that using this approach, students can develop an intuition as they start thinking in terms of the math they don't know yet and proceed to actually learn it.

## II. The Misty State(s) Formalism (MSF)

Let's use this terminology to describe the system presented by Terry Rudolph in his book. It represents qubits as sets of possible outcomes where the probability of each outcome can be calculated as shown earlier. The system in the book does remind one of an abacus because all the calculations are done using exclusively sets and combinations of ◯ and ●.

We start by describing these building blocks and the rules by which they evolve. The ◯ and ● are essentially classical values. They represent the two sides of a coin that is sitting on a table. It's what you get if you measure a qubit in the computational basis. We write $|0\rangle =$ ◯ and $|1\rangle =$ ● and it's a remarkable fact that we can reduce the understanding of virtually all introductory concepts in QC to manipulation of these symbols as if they were beads on an abacus.

In this formalism embedded superpositions of states can be resolved by taking their set union. We will see examples shortly. Quantum gates are linear maps that keep the total probability equal to 1. Classical reversible gates are valid quantum gates. Some common one-qubit quantum gates are $\mathtt{I}$, $\mathtt{X}$, $\mathtt{Y}$, $\mathtt{Z}$, $\mathtt{S}$, $\mathtt{T}$ and $\mathtt{H}$. The MSF calls them 'boxes' and in a note posted on the book's website [22] Terry says: "[u]nitary evolution in the quantum formalism just gives the 'boxes'. In the misty formalism we can only incorporate boxes whose representation in the quantum formalism is via a unitary matrix which is proportional to a matrix of integer entries. [...] It is a remarkable and nontrivial feature of quantum computation that such unitaries can be universal, i.e. used to simulate all unitaries, even ones with irrational complex entries."

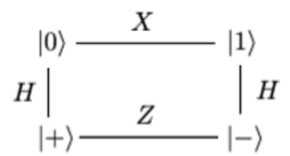
[2]Since we haven't presented them elsewhere.

Let's introduce some of the gates. The $\mathtt{NOT}$ gate is defined as follows: $\mathtt{X}(◯) = ●$ and $\mathtt{X}(●) = ◯$. The Hadamard gate creates a superposition of equally likely outcomes:

$$\mathtt{H}(◯) = \{◯, ●\} \quad \text{and} \quad \mathtt{H}(●) = \{◯, \overline{●}\}$$

These are essentially all the one-qubit gates used in the book. The line on top of the qubit represents phase and acts like a unary minus: $\mathtt{Z}(◯) = ◯$ and $\mathtt{Z}(●) = \overline{●}$.

To capture the linear evolution of these gates we say that for any quantum gate $f$ acting on a superposition of $n$ states we have $f(\{s_1, \ldots, s_n\}) = \{f(s_1), \ldots, f(s_n)\}$. Now the reader has all the tools to prove, for example, that the following commutative diagram holds:




The top part involving the $\mathtt{NOT}$ gate already checks out, from definition. We remind ourselves that $\{◯, ●\} = |+\rangle$ and likewise $\{◯, \overline{●}\} = |-\rangle$ and we have:

$$\mathtt{Z}(|+\rangle) = \mathtt{Z}(\{◯, ●\}) = \{\mathtt{Z}(◯), \mathtt{Z}(●)\} = \{◯, \overline{●}\} = |-\rangle$$

Keeping in mind that the phase (acting as a unary minus) is also linear we write: $\mathtt{Z}(\overline{●}) = \overline{\mathtt{Z}(●)} = \overline{\overline{●}} = ●$ which gives us the reverse relationship $\mathtt{Z}(|-\rangle) = |+\rangle$ as follows:

$$\mathtt{Z}(\{◯, \overline{●}\}) = \{◯, \mathtt{Z}(\overline{●})\} = \{◯, \overline{\mathtt{Z}(●)}\} = \{◯, ●\}$$

We now note that $\mathtt{H}(|0\rangle) = |+\rangle$ and $\mathtt{H}(|1\rangle) = |-\rangle$ follow from the definition of the Hadamard gate (see above on this page), so to complete the diagram we need only to prove the transitions $\mathtt{H}(\mathtt{H}(|0\rangle)) = |0\rangle$ and $\mathtt{H}(\mathtt{H}(|1\rangle)) = |1\rangle$.

We are now in a position to illustrate something we said earlier (that embedded superpositions of states can be resolved by taking their set union). Here's an example:

$$\{\{◯, ●\}, \{◯, \overline{●}\}\} = \{◯, ●, ◯, \overline{●}\}$$

This could be further used to show the effects of both destructive and constructive interference:

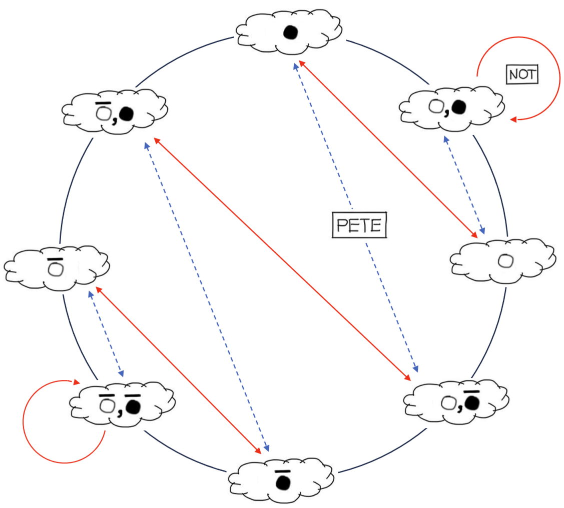
$$\{◯, ●, ◯, \overline{●}\} = \{◯, ◯\} = \{◯\} = ◯$$

The unary minus (i.e., change of phase) applied to a superposition of states changes the phase on each outcome:

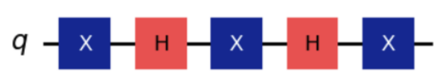
$$\overline{\{◯, ●\}} = \{\overline{◯}, \overline{●}\}$$

The rest follows by linearity of $\mathtt{H}(\{◯, ●\}) = \{\mathtt{H}(◯), \mathtt{H}(●)\}$ and $\mathtt{H}(\{◯, \overline{●}\}) = \{\mathtt{H}(◯), \mathtt{H}(\overline{●})\}$ and everything that was discussed above. We hope this offers a first glimpse into why the system devised by Terry Rudolph in "Q is for Quantum" is so accessible and suitable for even middle school students.

As Andrew Helwer showed us in his Feb. 14, 2016 talk [23] (and the accompanying, updated slides [24]) we can organize some of the most important transitions performed by the `H` and `X` gates in a very convenient unit circle state machine:




He uses this state diagram to calculate:




```
from qiskit.quantum_info import Statevector
sv = Statevector.from_instruction(qc)
sv.draw('latex')
```

$-|0\rangle$

Starting from the $◯$ state (east on the unit circle) we can trace the effect of the gates as alternative transitions leading into state $\overline{◯}$ (situated on the west side of the unit circle).

We now need to introduce two-qubit gates.

## III. Two-Qubit Gates

In the context of the previous diagram a good exercise is to prove the transitions between the states on the unit circle. You will then be reminded that the superposition operator is in fact a set operator, so the order of outcomes in the set does not matter[3]. When considering systems of more than one qubit that is no longer the case. Furthermore, phase acts differently, since the mult-qubit system is considered as a whole.

$$\mathtt{X}(\{◯, \overline{●}\}) = \{\mathtt{X}(◯), \overline{\mathtt{X}(●)}\} = \{●, \overline{◯}\} = \{\overline{◯}, ●\}$$

The state of multiple qubits is written as a tensor product. For us that simplifies even further, and closely matches Dirac notation, e.g. $◯● = |01\rangle$, $●●◯ = |110\rangle$, etc.

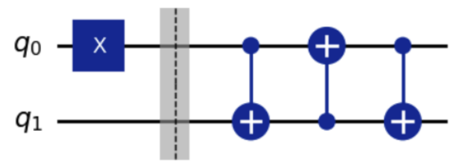
The order of qubits is important, i.e. $◯● \neq ●◯$ and the following will communicate reliably how phase acts on such an ensemble:

$$-◯●◯ = \overline{◯●◯} = \overline{◯}●◯ = ◯\overline{●}◯ = ◯●\overline{◯}$$

[3]Although, in general, an order is preferred.

Now there's one more aspect regarding the order (or numbering) of qubits in such a system that we need to clarify before going further. It has to do with the way we draw a circuit and reason about it on paper (or on a board) and the conventions used by Qiskit for same.

Consider the following Qiskit circuit: We know that this




```
sv = Statevector.from_instruction(qc)
sv.draw('latex')
```

$|10\rangle$

circuit is equivalent to a `SWAP` so if $q_0 = q_1 = |0\rangle$ then after the `NOT` gate the state is $|10\rangle$ which means that at the end, after the three `C-NOT` gates will have performed their intended functions the state should be $|01\rangle$. Yet Qiskit reports the final state as $|10\rangle$ Is there a mistake somewhere?

Everything is correct but Qiskit[4] reports the qubits left to right if you rotate the circuit clockwise 90 degrees. That means the exact inverse ordering we use when we draw the circuit on paper. So the result checks out but we need to keep this convention in mind from now on when comparing our calculations with Qiskit, so as to not get confused.

A two-qubit gate has two parameters. In the case of `C-NOT` one of them is the control qubit. As we can see from the circuit above—no matter how we look at it—sometimes the control qubit is the first qubit sometimes it is the second. We introduce a little arrow on top of the `X` to indicate where the controlled qubit is. Then, we have:

$$\overrightarrow{\mathtt{X}}(●◯) = ●● \quad \text{but} \quad \overleftarrow{\mathtt{X}}(●◯) = ●◯$$

And the entire circuit could be described as follows:

$$\overrightarrow{\mathtt{X}}(\overleftarrow{\mathtt{X}}(\overrightarrow{\mathtt{X}}(\mathtt{X}(◯)◯))) = ◯●$$

We now need to show the misty state formalism in action and before we start the next section we need to state two theorems[5]. **Theorem 1**: The amount of information extractable from one qubit is 1 bit. **Theorem 2**: An EPR pair cannot carry any information. We are now ready for superdense coding.

## IV. Superdense Coding

Superdense coding is a procedure that allows someone to send two classical bits to another party using just a single qubit of communication. It is a form of quantum advantage.

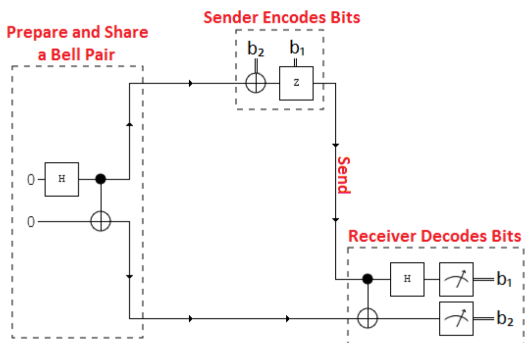
In superdense coding [20] the process starts with a third party, that we'll refer to as Charlie. Two qubits are prepared by Charlie in an entangled state. Charlie sends the first qubit

[4]As explained by John Watrous in [25]

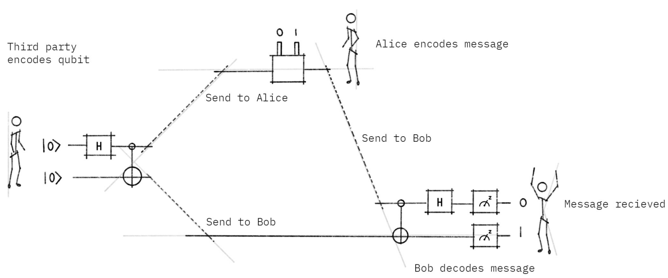
[5]See page 2 of the Peter Shor lecture notes at [26].

to Alice and the second qubit to Bob. The goal of the protocol is for Alice to send two classical bits of information to Bob using her qubit. Based on the theorems stated before this section, that should not be possible. First, we said that a maximally entangled pair of qubits carries no information [27]. Furthermore the amount of information extractable from one qubit is 1 (one) bit. So how are we going to be able to send two bits of information if we put together these two resources? They just don't seem to add up.

Here's the plan: Alice needs to apply a set of quantum gates to her qubit depending on the two bits of information that she wants to send. In [21] circuit diagram looks as follows:




A similar diagram can be found in the Qiskit textbook [20]:




Let's calculate the four cases and then compare one of them with Qiskit. Charlie starts with $\bigcirc\bigcirc$ always. After the Hadamard gate the state is: $\texttt{H}(\bigcirc)\bigcirc = \{\bigcirc, \bullet\}\bigcirc$ which boils down to $\{\bigcirc\bigcirc, \bullet\bigcirc\}$ which is the input to the C-NOT gate.

We then have $\overrightarrow{\texttt{X}}(\{\bigcirc\bigcirc, \bullet\bigcirc\}) = \{\overrightarrow{\texttt{X}}(\bigcirc\bigcirc), \overrightarrow{\texttt{X}}(\bullet\bigcirc)\}$ which is $\{\bigcirc\bigcirc, \bullet\bullet\}$ namely the Bell state that we were expecting. Now Alice needs to take one of four courses of action based on the intended message she wants to send.

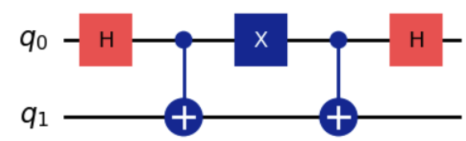
If the intended message is 00 Alice needs to apply the identity gate I, that is, she needs to leave the qubit alone. You can see this in the first picture on this page where the message Alice wants to send (two bits) represents the controls of the X and Z gates. When the control is 0 such a gate does nothing. In that case the Bell state reaches Bob and after the C-NOT we have: $\overrightarrow{\texttt{X}}(\{\bigcirc\bigcirc, \bullet\bullet\}) = \{\overrightarrow{\texttt{X}}(\bigcirc\bigcirc), \overrightarrow{\texttt{X}}(\bullet\bullet)\}$ which becomes $\{\bigcirc\bigcirc, \bullet\bigcirc\} = \{\bigcirc, \bullet\}\bigcirc$. Now the Hadamard gate acts as follows: $\texttt{H}(\{\bigcirc, \bullet\})\bigcirc = \bigcirc\bigcirc$.

If the intended message is 01 Alice needs to apply an X gate. It's easy to check that the pictures are in fact consistent; but while the second picture shows a black box that Alice controls, the first picture clearly numbers the control bits and matches them with the outputs. The first picture also shows the contents of the black box. After the action of the X gate the quantum state is $\{\texttt{X}(\bigcirc)\bigcirc, \texttt{X}(\bullet)\bullet\} = \{\bullet\bigcirc, \bigcirc\bullet\}$ and that's what Bob receives. After the C-NOT this becomes:

$$\overrightarrow{\texttt{X}}(\{\bullet\bigcirc, \bigcirc\bullet\}) = \{\bullet\bullet, \bigcirc\bullet\} = \{\bullet, \bigcirc\}\bullet$$

After the Hadamard we have: $\texttt{H}(\{\bullet, \bigcirc\})\bullet = \bigcirc\bullet$.

```
qc = QuantumCircuit(2)
qc.h(0)
qc.cx(0, 1)
qc.x(0)
qc.cx(0, 1)
qc.h(0)
qc.draw(output='mpl')
```




```
sv = Statevector.from_instruction(qc)
sv.draw('latex')
```

$|10\rangle$

The picture above is a reminder of how Qiskit produces the answer: backwards (as $q_1 q_0$, and we have explained why).

If the intended message is 10 (i.e., $q_0 = 1$ and $q_1 = 0$, which in Qiskit convention would be reported as 01) Alice needs to apply a Z gate: $\{\texttt{Z}(\bigcirc)\bigcirc, \texttt{Z}(\bullet)\bullet\} = \{\bigcirc\bigcirc, \overline{\bullet\bullet}\}$. Bob receives this. The effect of the C-NOT gate is:

$$\overrightarrow{\texttt{X}}(\{\bigcirc\bigcirc, \overline{\bullet\bullet}\}) = \{\bigcirc\bigcirc, \overline{\bullet}\bigcirc\} = \{\bigcirc, \overline{\bullet}\}\bigcirc$$

The Hadamard gate makes this: $\texttt{H}(\{\bigcirc, \overline{\bullet}\})\bigcirc = \bullet\bigcirc$.

If the intended message is 11 Alice needs to apply an X gate and then a Z gate. After the X gate we already calculated the state to be: $\{\bullet\bigcirc, \bigcirc\bullet\}$. The effect of the Z gate on this state is: $\{\texttt{Z}(\bullet)\bigcirc, \texttt{Z}(\bigcirc)\bullet\} = \{\overline{\bullet}\bigcirc, \bigcirc\bullet\}$ and that's what Bob receives. After the C-NOT the state becomes:

$$\overrightarrow{\texttt{X}}(\{\overline{\bullet}\bigcirc, \bigcirc\bullet\}) = \{\overrightarrow{\texttt{X}}(\overline{\bullet}\bigcirc), \overrightarrow{\texttt{X}}(\bigcirc\bullet)\} = \{\overline{\bullet\bullet}, \bigcirc\bullet\}$$

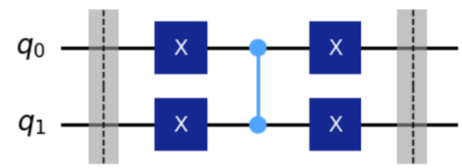
Now this further becomes: $\{\overline{\bullet}, \bigcirc\}\bullet = \{\bigcirc, \overline{\bullet}\}\bullet$ and after the Hadamard gate we have $\texttt{H}(\{\bigcirc, \overline{\bullet}\})\bullet = \bullet\bullet$ so all checks out as originally announced. This concludes our description of superdense coding.

Superdense coding and teleportation are dual phenomena. Teleportation can be described as entanglement-assisted quantum information transfer over a classical channel; superdense coding can be described as entanglement-assisted classical information transfer over a quantum channel. In both cases entanglement plays a crucial role. We have addressed the topic in general [17] and with respect to very specific phenomena (e.g., the GHZ game, [18]) in the context of the misty states formalism and the quantum abacus in other papers [19]. Quantum particles seem to influence each other with superluminal speed over arbitrarily long distances [16]. Quantum algorithms make use of this property. Entanglement swapping, another important protocol, allows particles that never interacted in the past to become entangled. In that sense entanglement swapping is a sort of teleportation of entanglement.

## V. Grover Search Algorithm

Grover's algorithm [28], [29] can speed up an unstructured search problem quadratically, but its uses extend beyond that; it can serve as a general trick or subroutine to obtain quadratic run time improvements for a variety of other algorithms. This is called the amplitude amplification trick. Suppose you are given a large list of $N$ items. Among these items there is one item with a unique property that we wish to locate; we will call this one the winner $w$. Think of each item in the list as a box of a particular color. Say all items in the list are gray except the winner $w$, which is purple. To find the purple box—the marked item—using classical computation, one would have to check on average $\frac{N}{2}$ of these boxes, and in the worst case, all of them. On a quantum computer, however, we can find the marked item in roughly $\sqrt{N}$ steps[6] with Grover's amplitude amplification trick. Grover's algorithm consists of three main algorithmic steps: state preparation, the oracle, and the diffusion operator. The state preparation is where we create the search space, which is all possible cases the answer could take. In the list example we mentioned above, the search space would be all the items of that list. The oracle is what marks the correct answer, or answers we are looking for, and the diffusion operator magnifies these answers so they can stand out and be measured at the end of the algorithm.

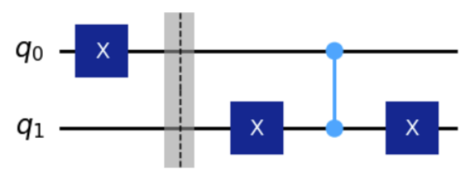
The first step of Grover's algorithm is the initial state preparation. As we just mentioned, the search space is all possible values we need to search through to find the answer we want. Here our 'database' is comprised of all the possible computational basis states our qubits can be in. For example, if we have 2 qubits, our list is the states $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$ (i.e the states $|1\rangle$ to $|4\rangle$). So, in this case the size of our search space will be $N = 2^2 = 4$. For two qubits we can imagine there are four cards face-down on the table only one of which is an ace. To find the ace we would have to turn face up on average more than two cards. Grover's algorithm finds the card in just one iteration, always[7]. The second and most important step of Grover's algorithm is the oracle. Oracles add a negative phase to the solution states so they can standout from the rest and be measured[8]. For our two-qubit example we have four oracles. We show them below and start with the one for $|00\rangle$:




```
sv = Statevector.from_instruction(oracle_00)
sv.draw('latex')
```

$-|00\rangle$

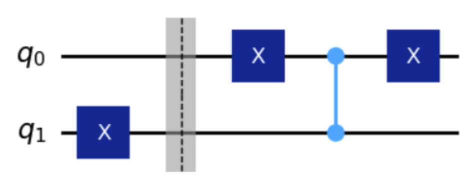
We know that the controlled-Z (`C-Z`) gate adds a phase only when both qubits are ●. That should be enough to realize why this is indeed the oracle for ◯◯: any other state would fail to produce the needed prerequisite for an added phase. It is now easy to determine the other three oracles:




```
sv = Statevector.from_instruction(oracle_10)
sv.draw('latex')
```

$-|01\rangle$

First off we note again, here, that the name of the quantum oracle (circuit) is chosen with the traditional numbering convention of qubits in mind (i.e., $q_0$ first, then $q_1$) whereas the reporting is done using Qiskit numbering convention, that is, $q_1$ is listed first, then $q_0$. Then the state that is identified by the oracle, in Qiskit, as the solution, is consistent with ●◯.




```
sv = Statevector.from_instruction(oracle_01)
sv.draw('latex')
```

$-|10\rangle$

Meanwhile the oracle for ◯● $= |01\rangle$ or, in Qiskit notation $|10\rangle$, is exactly symmetric. With input ◯● to the oracle the state after the controlled-Z gate is `C-Z(X(`◯`)`●`))` = ●●;

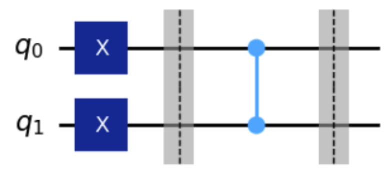
[6] A quadratic speedup is indeed a substantial time-saver for finding marked items in long lists. Additionally, the algorithm does not use the list's internal structure, which makes it generic; this is why it immediately provides a quadratic quantum speed-up for many other classical problems.

[7] Another game associated with the two qubit Grover's algorithm is 'money or tigers' by Ed Barnes from Virginia Tech: imagine you have four doors, and behind one of them there's a large sum of money, while behind each one of the other three there is a hungry tiger. Grover's algorithm shows you which door to open to get to the money in one iteration, thus eliminating the risk of running, in the process, into any of the tigers.

[8] What makes Grover's algorithm so powerful is how easy it is to convert a problem to an oracle of this form. There are many computational problems in which it is difficult to find a solution, but relatively easy to verify a solution.

all other inputs do not acquire a phase. Then what comes out of the oracle in this case is $\overline{\mathtt{X}(\CIRCLE)\CIRCLE} = \overline{\bigcirc\CIRCLE}$ and Qiskit is reporting that as $-|10\rangle$. All other inputs are reconstructed unchanged, and are not marked as solution(s).

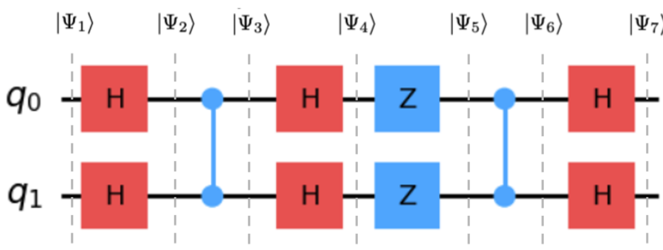
Finally, the simplest oracle is the one for $|11\rangle$:




```
sv = Statevector.from_instruction(oracle_11)
sv.draw('latex')
```

$-|11\rangle$

We will now present the circuit for Grover's search algorithm when the input consists of two qubits. We put the




oracle for $|11\rangle$ in and remind the reader that the oracle simply recognizes (or validates) the right answer—it does not attempt to construct it in any way. It is through the procedure called amplitude amplification that this quantum algorithm significantly enhances the probability of guessing the right answer $w$. This procedure stretches out (amplifies) the amplitude of the marked item, which shrinks the other items' amplitude, so that measuring the final state will return the right item with near-certainty. We will trace the algorithm step by step and naturally we start with $|\Psi_1\rangle = \bigcirc\bigcirc = |00\rangle$.

Then,

$$
\begin{aligned}
|\Psi_2\rangle &= H(\bigcirc)H(\bigcirc) \\
&= \{\bigcirc, \CIRCLE\}\{\bigcirc, \CIRCLE\} \\
&= \{\bigcirc\bigcirc, \CIRCLE\bigcirc, \bigcirc\CIRCLE, \CIRCLE\CIRCLE\} \\
&= \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle
\end{aligned}
$$

Next stage is after the controlled-Z (C-Z) gate:

$$
\begin{aligned}
|\Psi_3\rangle &= \{\mathtt{C{-}Z}(\bigcirc\bigcirc), \mathtt{C{-}Z}(\CIRCLE\bigcirc), \mathtt{C{-}Z}(\bigcirc\CIRCLE), \mathtt{C{-}Z}(\CIRCLE\CIRCLE)\} \\
&= \{\bigcirc\bigcirc, \CIRCLE\bigcirc, \bigcirc\CIRCLE, \overline{\CIRCLE\CIRCLE}\} \\
&= \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle
\end{aligned}
$$

And here comes a long (but instructive) calculation:

$$
\begin{aligned}
|\Psi_4\rangle &= \{\mathtt{H}(\bigcirc)\mathtt{H}(\bigcirc), \mathtt{H}(\CIRCLE)\mathtt{H}(\bigcirc), \mathtt{H}(\bigcirc)\mathtt{H}(\CIRCLE), \overline{\mathtt{H}(\CIRCLE)\mathtt{H}(\CIRCLE)}\} \\
&= \{\{\bigcirc, \CIRCLE\}\{\bigcirc, \CIRCLE\}, \\
&\quad \{\bigcirc, \overline{\CIRCLE}\}\{\bigcirc, \CIRCLE\}, \\
&\quad \{\bigcirc, \CIRCLE\}\{\bigcirc, \overline{\CIRCLE}\}, \\
&\quad \overline{\{\bigcirc, \overline{\CIRCLE}\}}\{\bigcirc, \overline{\CIRCLE}\}\} \\
&= \{\ \bigcirc\bigcirc, \bigcirc\CIRCLE, \CIRCLE\bigcirc, \CIRCLE\CIRCLE, \\
&\quad \bigcirc\bigcirc, \bigcirc\CIRCLE, \overline{\CIRCLE}\bigcirc, \overline{\CIRCLE}\CIRCLE, \\
&\quad \bigcirc\bigcirc, \bigcirc\overline{\CIRCLE}, \CIRCLE\bigcirc, \CIRCLE\overline{\CIRCLE}, \\
&\quad \{\overline{\bigcirc}, \CIRCLE\}\{\bigcirc, \overline{\CIRCLE}\}\} \\
&= \{\ \bigcirc\bigcirc, \bigcirc\CIRCLE, \\
&\quad \bigcirc\bigcirc, \\
&\quad \bigcirc\bigcirc, \qquad \CIRCLE\bigcirc, \CIRCLE\overline{\CIRCLE}, \\
&\quad \overline{\bigcirc}\bigcirc, \overline{\bigcirc\CIRCLE}, \CIRCLE\bigcirc, \CIRCLE\overline{\CIRCLE}\} \\
&= \{\ \bigcirc\bigcirc, \bigcirc\CIRCLE, \CIRCLE\bigcirc, \CIRCLE\overline{\CIRCLE}, \\
&\quad \bigcirc\bigcirc, \bigcirc\CIRCLE, \CIRCLE\bigcirc, \CIRCLE\overline{\CIRCLE}\} \\
&= \{\ \bigcirc\bigcirc, \bigcirc\CIRCLE, \CIRCLE\bigcirc, \overline{\CIRCLE\CIRCLE}\} \\
&= \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle
\end{aligned}
$$

As one can see some pairs of terms cancel each other. I have removed them (please note that the canceling pairs are conveniently placed one term above the other) but also kept their place in the original equation for easier tracking.

For the last simplification we recall the calculation of probability from the first page: if $\bigcirc$ appears $n$ times in the misty state and $\CIRCLE$ appears $m$ times in the misty state then their probabilities are $\frac{n^2}{n^2+m^2}$ and $\frac{m^2}{n^2+m^2}$ respectively. It follows that if states occur an equal multiple of times (e.g., $nk$ and $mk$ with $k \in \mathbb{N}$) then the probabilities are unchanged (because $k^2$ appears everywhere and consequently it simplifies).

Now $|\Psi_5\rangle$ reflects the action of the Z gates on $|\Psi_4\rangle$:

$$
\begin{aligned}
|\Psi_5\rangle &= \{\mathtt{Z}(\bigcirc)\mathtt{Z}(\bigcirc), \mathtt{Z}(\CIRCLE)\mathtt{Z}(\bigcirc), \mathtt{Z}(\bigcirc)\mathtt{Z}(\CIRCLE), \overline{\mathtt{Z}(\CIRCLE)\mathtt{Z}(\CIRCLE)}\} \\
&= \{\bigcirc\bigcirc, \overline{\CIRCLE\bigcirc}, \overline{\bigcirc\CIRCLE}, \overline{\overline{\CIRCLE\ \CIRCLE}}\} \\
&= \{\bigcirc\bigcirc, \overline{\CIRCLE\bigcirc}, \overline{\bigcirc\CIRCLE}, \CIRCLE\CIRCLE\} \\
&= \frac{1}{2}|00\rangle - \frac{1}{2}|01\rangle - \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle
\end{aligned}
$$

The calculation of $|\Psi_6\rangle$ is similar to the one for $|\Psi_3\rangle$ :

$$
\begin{aligned}
|\Psi_6\rangle &= \{\mathtt{C{-}Z}(\bigcirc\bigcirc), \overline{\mathtt{C{-}Z}(\CIRCLE\bigcirc)}, \overline{\mathtt{C{-}Z}(\bigcirc\CIRCLE)}, \overline{\mathtt{C{-}Z}(\CIRCLE\CIRCLE)}\} \\
&= \{\bigcirc\bigcirc, \overline{\CIRCLE\bigcirc}, \overline{\bigcirc\CIRCLE}, \overline{\overline{\CIRCLE\CIRCLE}}\} \\
&= \{\bigcirc\bigcirc, \overline{\CIRCLE\bigcirc}, \overline{\bigcirc\CIRCLE}, \CIRCLE\CIRCLE\} \\
&= \frac{1}{2}|00\rangle - \frac{1}{2}|01\rangle - \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle
\end{aligned}
$$

The calculation for $|\Psi_7\rangle$ also matches the type of steps we have seen at $\Psi_4$ just that the result is, convincingly, different:

$$
\begin{aligned}
|\Psi_7\rangle &= \{\texttt{H}(\bigcirc)\texttt{H}(\bigcirc), \overline{\texttt{H}(\CIRCLE)\texttt{H}(\bigcirc)}, \overline{\texttt{H}(\bigcirc)\texttt{H}(\CIRCLE)}, \texttt{H}(\CIRCLE)\texttt{H}(\CIRCLE)\} \\
&= \{\{\bigcirc, \CIRCLE\}\{\bigcirc, \CIRCLE\}, \\
&\quad \{\bigcirc, \overline{\CIRCLE}\}\overline{\{\bigcirc, \CIRCLE\}}, \\
&\quad \overline{\{\bigcirc, \CIRCLE\}}\{\bigcirc, \overline{\CIRCLE}\}, \\
&\quad \{\bigcirc, \overline{\CIRCLE}\}\{\bigcirc, \overline{\CIRCLE}\}\} \\
&= \{\ \bigcirc\bigcirc, \bigcirc\CIRCLE, \CIRCLE\bigcirc, \CIRCLE\CIRCLE, \\
&\quad \{\bigcirc, \overline{\CIRCLE}\}\{\overline{\bigcirc}, \overline{\CIRCLE}\}, \\
&\quad \{\overline{\bigcirc}, \overline{\CIRCLE}\}\{\bigcirc, \overline{\CIRCLE}\}, \\
&\quad \bigcirc\bigcirc, \bigcirc\overline{\CIRCLE}, \overline{\CIRCLE}\bigcirc, \overline{\CIRCLE}\,\overline{\CIRCLE}\} \\
&= \{\ \bigcirc\bigcirc, \qquad\qquad \CIRCLE\CIRCLE, \\
&\quad \bigcirc\overline{\bigcirc}, \bigcirc\overline{\CIRCLE}, \overline{\CIRCLE}\,\overline{\bigcirc}, \overline{\CIRCLE}\,\overline{\CIRCLE}, \\
&\quad \overline{\bigcirc}\bigcirc, \overline{\bigcirc}\CIRCLE, \overline{\CIRCLE}\bigcirc, \overline{\CIRCLE}\,\overline{\CIRCLE}, \\
&\quad \bigcirc\bigcirc, \qquad\qquad \CIRCLE\CIRCLE\} \\
&= \{ \qquad\qquad \CIRCLE\CIRCLE, \\
&\quad \qquad\qquad \CIRCLE\CIRCLE, \\
&\quad \qquad\qquad \CIRCLE\CIRCLE, \\
&\quad \qquad\qquad \CIRCLE\CIRCLE\} \\
&= \{\CIRCLE\CIRCLE\} = |11\rangle
\end{aligned}
$$

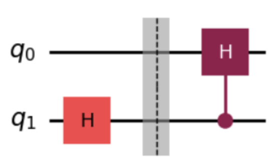
This time the pairs of items that cancel each other are not next to each other but they're still in the same vertical column.

Grover's algorithm produces the answer in one step[9]. Using a quantum oracle that is able to identify (not construct) the correct answer we know which of the four cards face down on the table is the $w$ card[10].

## VI. The Break-Even Point

Here's an example constructed with just Hadamard gates:

```
be = QuantumCircuit(2)
be.h(1)
be.barrier()
be.ch(1, 0)
be.draw(output='mpl')
```




```
sv = Statevector.from_instruction(be)
sv.draw('latex')
```

$$\frac{\sqrt{2}}{2}|00\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle$$

Let's see if we can calculate the same with the abacus.

[9]Calculations for the other three cases proceed in a similar manner.

[10]Or, in the Eddie Barnes game, which door to open to get to the money, in just one step, avoiding altogether the other three doors that lead to a hungry tiger.

We have to appreciate the fact that so far there have been no numbers (coefficients, for probability amplitudes) in the misty states formalism. That's what makes it accessible to students as early as middle school. The initial state is $q_0q_1 = \bigcirc\bigcirc$. After the first Hadamard gate it becomes $|\Psi_1\rangle = \bigcirc\{\bigcirc, \CIRCLE\}$. That's the state at the barrier. We then have to calculate the effect of $\overleftarrow{\texttt{H}}(\bigcirc\{\bigcirc, \CIRCLE\}) = \{\overleftarrow{\texttt{H}}(\bigcirc\bigcirc), \overleftarrow{\texttt{H}}(\bigcirc\CIRCLE)\}$ so we write:

$$
\begin{aligned}
|\Psi_2\rangle &= \{\bigcirc\bigcirc, \{\bigcirc, \CIRCLE\}\CIRCLE\} \\
&= \{\bigcirc\bigcirc, \{\bigcirc\CIRCLE, \CIRCLE\CIRCLE\}\} \\
&= \{\bigcirc\bigcirc, \bigcirc\CIRCLE, \CIRCLE\CIRCLE\} \\
&= \frac{1}{\sqrt{3}}|00\rangle + \frac{1}{\sqrt{3}}|10\rangle + \frac{1}{\sqrt{3}}|11\rangle
\end{aligned}
$$

We switched to Qiskit ordering at the end but it's clear that the calculation is not accurate: the probability amplitudes don't match. Let's do a little research:

$$
\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{pmatrix} = \{\bigcirc\bigcirc, \bigcirc\CIRCLE\}
$$

This is the effect of the first Hadamard gate acting on $q_1$ when on the other wire we have the identity gate (the combined matrix is their tensor product). Now we have to use the matrix representation of the controlled Hadamard gate:

$$
\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{2} \\ 0 \\ \frac{1}{2} \end{pmatrix}
$$

Now the resulting vector is equivalent to:

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|11\rangle$$

In Qiskit ordering this is:

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle$$

So now we have the answer: as Terry warned us, in the MSF "we can only [use] boxes whose representation in the quantum formalism is via a unitary matrix which is proportional to a matrix of integer entries." Clearly this is not true of the controlled Hadamard matrix and that's the reason for which our calculations fail. What can we do?

This is the point where the MSF and the conventional formalism need to break even. We propose we extend the MSF by allowing coefficients representing the probability amplitudes. This will bring us closer to the Dirac algebraic notation but at this point we have so much that we have been able to understand with just the pure MSF. The upgrade does not feel gratuitous, in fact it seems to be earned. Here's how the calculation proceeds now: $|\Psi_1\rangle = \bigcirc\{\frac{1}{\sqrt{2}}\bigcirc, \frac{1}{\sqrt{2}}\CIRCLE\}$ and then calculate $\overleftarrow{\texttt{H}}(\bigcirc\{\frac{1}{\sqrt{2}}\bigcirc, \frac{1}{\sqrt{2}}\CIRCLE\})$ like we did before.

The difference is that now we have the probability amplitudes with us. So we have the following sequence of steps:

$$\begin{aligned}
\overleftarrow{\texttt{H}}(\bigcirc\texttt{H}(\bigcirc)) &= \overleftarrow{\texttt{H}}(\bigcirc\{\frac{1}{\sqrt{2}}\bigcirc, \frac{1}{\sqrt{2}}\bullet\}) \\
&= \{\frac{1}{\sqrt{2}}\overleftarrow{\texttt{H}}(\bigcirc\bigcirc), \frac{1}{\sqrt{2}}\overleftarrow{\texttt{H}}(\bigcirc\bullet)\} \\
&= \{\frac{1}{\sqrt{2}}\bigcirc\bigcirc, \frac{1}{\sqrt{2}}\{\frac{1}{\sqrt{2}}\bigcirc, \frac{1}{\sqrt{2}}\bullet\}\bullet\} \\
&= \{\frac{1}{\sqrt{2}}\bigcirc\bigcirc, \frac{1}{2}\bigcirc\bullet, \frac{1}{2}\bullet\bullet\} \\
&= \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle
\end{aligned}$$

On the last line above we switched to Qiskit ordering.

## VII. $\sqrt{\texttt{NOT}}$ AND $\sqrt{\texttt{Z}}$

Now that we have extended the MSF with coefficients we can introduce (and prove) $\texttt{S} = \sqrt{\texttt{Z}}$ and $\sqrt{\texttt{X}} = \texttt{HSH}$. First we have $\texttt{S}(\bigcirc) = \bigcirc$ and $\texttt{S}(\bullet) = i\bullet$ where $i = \sqrt{-1}$. From this it's clear that $\texttt{S}^2 = \texttt{Z}$ so $\texttt{S} = \sqrt{\texttt{Z}}$ because $i^2 = -1$.

Likewise $\texttt{HSH} \cdot \texttt{HSH} = \texttt{HS}^2\texttt{H} = \texttt{HZH} = \texttt{X}$ which we proved early in this paper so $\texttt{HSH} = \sqrt{\texttt{X}} = \sqrt{\texttt{NOT}}$ checks out.

These are also great opportunities to introduce students to matrices and properties of matrix multiplication as well as the notion of inverse and/or unitary matrix. When we can derive a result in more than one way we feel more confident about its correctness.

## VIII. $R_y(\theta_3)$

It's time to introduce another gate that does not have a representation in the MSF (but readily has one in the extended MSF). At QSEEC 2023 in Seattle we were asked how we define arbitrary rotations in the MSF. The answer is: we define them as primitives in the extended MSF. We were also asked how we define arbitrary qubits, but by now we have already answered that question[11]. So let's consider a specific rotation gate that will be useful a bit later. The first axiom is:

$$R_y(\theta_3)(\bigcirc) = \{\frac{1}{\sqrt{3}}\bigcirc, \sqrt{\frac{2}{3}}\bullet\}$$

This is precisely the quantum state that we said, in the beginning of the paper, that it did not have a representation in the MSF. The other axiom is:

$$R_y(\theta_3)(\bullet) = \{-\frac{2}{\sqrt{3}}\bigcirc, \sqrt{\frac{1}{3}}\bullet\}$$

From this we can already calculate in general how this gate acts on a generic superposition of $|0\rangle = \bigcirc$ and $|1\rangle = \bullet$. The reason this gate does not exist in the MSF will become clear below. First off $\theta_3 = 2\arccos\frac{1}{\sqrt{3}}$ and so the matrix is:

$$\begin{pmatrix} \cos\frac{\theta_3}{2} & -\sin\frac{\theta_3}{2} \\ \sin\frac{\theta_3}{2} & \cos\frac{\theta_3}{2} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{3}} & -\sqrt{\frac{2}{3}} \\ \sqrt{\frac{2}{3}} & \frac{1}{\sqrt{3}} \end{pmatrix}$$

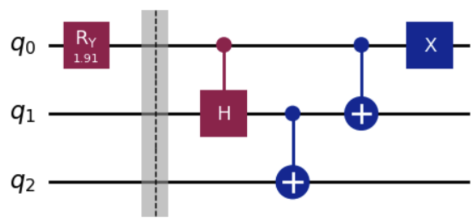
[11] e.g., $\texttt{X}(\{\alpha\bigcirc, \beta\bullet\}) = \{\beta\bigcirc, \alpha\bullet\}\ \forall\alpha, \beta \in \mathbb{C}$

## IX. THREE-QUBIT W-STATES

We can now create W-entangled states:

```
import numpy as np
import math
theta = 2 * np.arccos(1/math.sqrt(3))
```

```
w = QuantumCircuit(3)
w.ry(theta, 0)
w.draw(output='mpl')
w.barrier()
w.ch(0, 1)
w.cx(1, 2)
w.cx(0, 1)
w.x(0)
w.draw('mpl')
```




```
sv = Statevector.from_instruction(w)
sv.draw('latex')
```

$$\frac{\sqrt{3}}{3}|001\rangle + \frac{\sqrt{3}}{3}|010\rangle + \frac{\sqrt{3}}{3}|100\rangle$$

Let's calculate: the initial state is still $\bigcirc\bigcirc\bigcirc$. After the rotation we have $\{\frac{1}{\sqrt{3}}\bigcirc, \sqrt{\frac{2}{3}}\bullet\} = \{\frac{1}{\sqrt{3}}\bigcirc\bigcirc\bigcirc, \frac{1}{\sqrt{3}}\bullet\bullet\bullet\}$.

When the controlled Hadamard kicks in we have:

$$\begin{aligned}
&\{\frac{1}{\sqrt{3}}\overrightarrow{\texttt{H}}(\bigcirc\bigcirc)\bigcirc, \sqrt{\frac{2}{3}}\overrightarrow{\texttt{H}}(\bullet\bigcirc)\bigcirc\} = \\
&\{\frac{1}{\sqrt{3}}\bigcirc\bigcirc\bigcirc, \sqrt{\frac{2}{3}}\bullet\{\frac{1}{\sqrt{2}}\bigcirc, \frac{1}{\sqrt{2}}\bullet\}\bigcirc\} = \\
&\{\frac{1}{\sqrt{3}}\bigcirc\bigcirc\bigcirc, \frac{1}{\sqrt{3}}\bullet\bigcirc\bigcirc, \frac{1}{\sqrt{3}}\bullet\bullet\bigcirc\} = \\
&\frac{1}{\sqrt{3}}|000\rangle + \frac{1}{\sqrt{3}}|001\rangle + \frac{1}{\sqrt{3}}|011\rangle
\end{aligned}$$

In the last line we switched to Qiskit ordering of qubits.

After the first C-NOT we have:

$$\begin{aligned}
&\{\frac{1}{\sqrt{3}}\bigcirc\overrightarrow{\texttt{X}}(\bigcirc\bigcirc), \frac{1}{\sqrt{3}}\bullet\overrightarrow{\texttt{X}}(\bigcirc\bigcirc), \frac{1}{\sqrt{3}}\bullet\overrightarrow{\texttt{X}}(\bullet\bigcirc)\} = \\
&\{\frac{1}{\sqrt{3}}\bigcirc\bigcirc\bigcirc, \frac{1}{\sqrt{3}}\bullet\bigcirc\bigcirc, \frac{1}{\sqrt{3}}\bullet\bullet\bullet\} = \\
&\frac{1}{\sqrt{3}}|000\rangle + \frac{1}{\sqrt{3}}|001\rangle + \frac{1}{\sqrt{3}}|111\rangle
\end{aligned}$$

Again, we switched to Qiskit ordering at the very end.

The calculation after the second C-NOT proceeds similarly:

$$\begin{aligned}
&\{\frac{1}{\sqrt{3}}\overrightarrow{\texttt{X}}(\bigcirc\bigcirc)\bigcirc, \frac{1}{\sqrt{3}}\overrightarrow{\texttt{X}}(\bullet\bigcirc)\bigcirc, \frac{1}{\sqrt{3}}\overrightarrow{\texttt{X}}(\bullet\bullet)\bullet\} = \\
&\{\frac{1}{\sqrt{3}}\bigcirc\bigcirc\bigcirc, \frac{1}{\sqrt{3}}\bullet\bullet\bigcirc, \frac{1}{\sqrt{3}}\bullet\bigcirc\bullet\} = \\
&\frac{1}{\sqrt{3}}|000\rangle + \frac{1}{\sqrt{3}}|011\rangle + \frac{1}{\sqrt{3}}|101\rangle
\end{aligned}$$

Finally after the X gate we have:

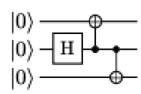
$$\{\frac{1}{\sqrt{3}}\bullet\bigcirc\bigcirc, \frac{1}{\sqrt{3}}\bigcirc\bullet\bigcirc, \frac{1}{\sqrt{3}}\bigcirc\bigcirc\bullet\} =$$
$$\frac{1}{\sqrt{3}}|001\rangle + \frac{1}{\sqrt{3}}|010\rangle + \frac{1}{\sqrt{3}}|100\rangle$$

As before the Dirac notation is with Qiskit ordering.

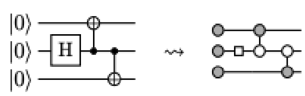
Everything checks out.

## X. Commencement

The system presented uses string rewriting rules to show what happens with the quantum state as it travels through a circuit. By contrast the ZX-calculus is a diagrammatic language that rewrites entire portions of the circuit (so it's a graph-rewriting technique) while preserving the equivalence of the circuit. One is a global technique; the other helps trace a quantum state through a circuit in "slow-motion". As an example consider the following circuit:

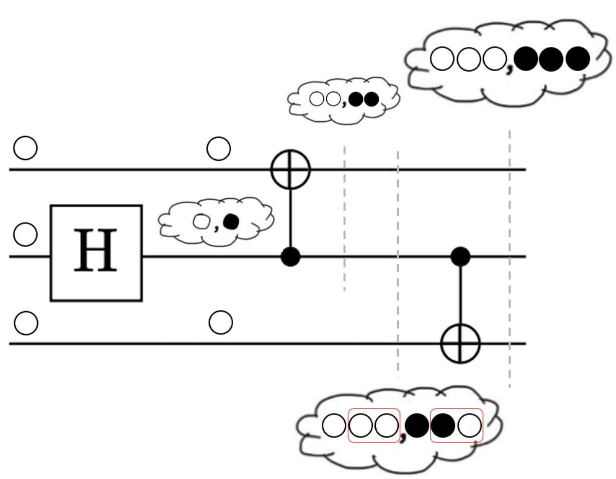



It can be written as a ZX-diagram (see [5]):




Which can then be simplified as follows:

This proves (diagrammatically) that the circuit implements a GHZ state[12]. By comparison the same proof with the "Quantum Abacus" proceeds as follows:



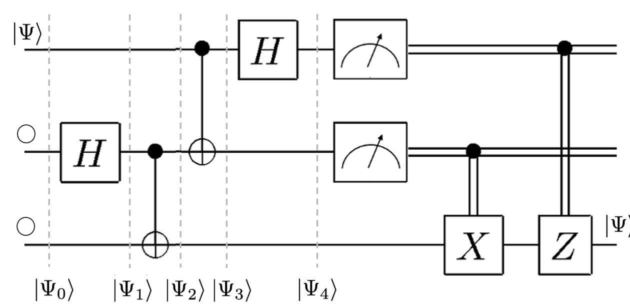

[12]The Greenberger–Horne–Zeilinger (GHZ) state is an entangled quantum state for 3 qubits with this expression: $\frac{|000\rangle+|111\rangle}{\sqrt{2}}$ (so, $\{\bigcirc\bigcirc\bigcirc, \bullet\bullet\bullet\}$).

Here superpositions are represented as "misty states" which is just a graphically somewhat richer representation of our superposition (set) operator. We now have all the tools to address teleportation (inaccessible in the regular MSF). Initially introduced in Bennett et al. (1993), quantum teleportation describes a protocol allowing to reconstruct an unknown quantum state $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ at a new location by using a classical information channel and a pair of entangled states. So the first step is going to be to find a way to represent an arbitrary $|\Psi\rangle$ state in our "abacus" system. But following our argument thus far this is no longer a challenge (since we are now using the extended MSF). That will allow us to morph gradually into the traditional, mathematical representation.

In that case $\forall\alpha, \beta \in \mathbb{C}$ we may also have[13]:

$$\alpha|0\rangle + \beta|1\rangle = \{\alpha\bigcirc, \beta\bullet\}$$

Quantum teleportation ([10], [12]) requires three qubits, where the first one holds the state to be teleported and the remaining ones are initialised to $|0\rangle$. The protocol consists of performing the following quantum circuit:




The word teleportation does fit well here as this phenomenon occurs instantaneously[14] and is not affected by distance or separating barriers. Let's prove the protocol by calculating intermediary stages $|\Psi_0\rangle, \ldots, |\Psi_4\rangle$. We start with:

$$|\Psi_0\rangle = \left\{\alpha\bigcirc\bigcirc\bigcirc, \beta\bullet\bigcirc\bigcirc\right\}$$

Traditional calculation confirms this:

$$|\Psi_0\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes |00\rangle = \alpha|000\rangle + \beta|100\rangle$$

In the classroom this would be a good moment to talk about tensor products and relate the following:

$$\bullet\bigcirc\bullet \equiv |1\rangle \otimes |0\rangle \otimes |1\rangle = \begin{pmatrix}0\\0\\0\\0\\1\\0\\0\\0\end{pmatrix}$$

[13]This is the extension to the misty state(s) formalism (MSF).

[14]The instantaneously teleported state cannot be used to achieve faster than light communication, as in order to be properly reconstructed requires classical information about measurement performed at the sender location, making it sensitive to limitations imposed by the speed of light.

Please don't forget that $|0\rangle$ and $|1\rangle$ are in fact vectors. We will revisit this topic briefly at the end of this section. Furthermore we can continue to calculate and relate the results obtained via the "abacus" to those obtained via standard mathematical operations. As an example we can calculate:

$$\begin{aligned}\mathtt{H}(\{\alpha \bigcirc, \beta \bullet\}) =& \alpha \{\bigcirc, \bullet\} + \beta \{\bigcirc, \overline{\bullet}\} = \\ =& \{(\alpha+\beta)\bigcirc, (\alpha-\beta)\bullet\}\end{aligned}$$

This is clearly confirmed by the standard calculation:

$$\mathtt{H}(\alpha|0\rangle + \beta|1\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}\alpha+\beta\\ \alpha-\beta\end{pmatrix}$$

So now we can calculate:

$$|\Psi_1\rangle = \Big\{\alpha \bigcirc\bigcirc\bigcirc, \alpha \bigcirc\bullet\bigcirc, \beta \bullet\bigcirc\bigcirc, \beta \bullet\bullet\bigcirc\Big\}$$

Traditional calculation, again, confirms our result:

$$|\Psi_1\rangle = \frac{1}{\sqrt{2}}\Big(\alpha|000\rangle + \alpha|010\rangle + \beta|100\rangle + \beta|110\rangle\Big)$$

After the first `C-NOT` gate:

$$|\Psi_2\rangle = \Big\{\alpha \bigcirc\bigcirc\bigcirc, \alpha \bigcirc\bullet\bullet, \beta \bullet\bigcirc\bigcirc, \beta \bullet\bullet\bullet\Big\}$$

Traditional calculation yields:

$$|\Psi_2\rangle = \frac{1}{\sqrt{2}}\Big(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle\Big)$$

The second `C-NOT` acts on the first two qubits:

$$|\Psi_3\rangle = \Big\{\alpha \bigcirc\bigcirc\bigcirc, \alpha \bigcirc\bullet\bullet, \beta \bullet\bullet\bigcirc, \beta \bullet\bigcirc\bullet\Big\}$$

Using standard calculation techniques:

$$|\Psi_3\rangle = \frac{1}{\sqrt{2}}\Big(\alpha|000\rangle + \alpha|011\rangle + \beta|110\rangle + \beta|101\rangle\Big)$$

We now have only one stage left but it should be relatively clear that developments are now in lockstep. So, after the second Hadamard gate (acting on just the first qubit):

$$\begin{aligned}|\Psi_4\rangle = \Big\{&\alpha\{\bigcirc, \bullet\}\bigcirc\bigcirc, \alpha\{\bigcirc, \bullet\}\bullet\bullet, \\ &\beta\{\bigcirc, \overline{\bullet}\}\bullet\bigcirc, \beta\{\bigcirc, \overline{\bullet}\}\bigcirc\bullet\Big\} = \\ = \Big\{&\alpha \bigcirc\bigcirc\bigcirc, \alpha \bullet\bigcirc\bigcirc, \alpha \bigcirc\bullet\bullet, \alpha \bullet\bullet\bullet, \\ &\beta \bigcirc\bullet\bigcirc, \beta \overline{\bullet}\bullet\bigcirc, \beta \bigcirc\bigcirc\bullet, \beta \overline{\bullet}\bigcirc\bullet\Big\}\end{aligned}$$

Traditional calculation meanwhile yields (same thing):

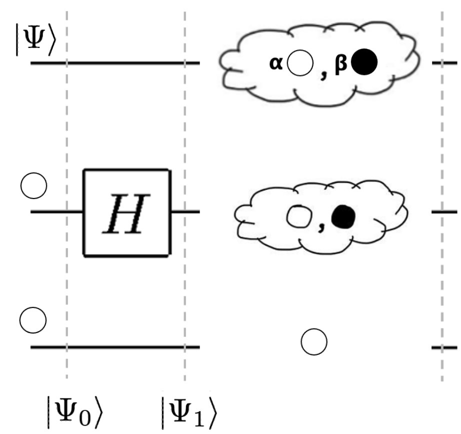
$$\begin{aligned}|\Psi_4\rangle =& \frac{1}{2}|00\rangle\Big(\alpha|0\rangle + \beta|1\rangle\Big) + \quad \texttt{; nothing} \\ &+ \frac{1}{2}|01\rangle\Big(\beta|0\rangle + \alpha|1\rangle\Big) + \quad \texttt{; apply X} \\ &+ \frac{1}{2}|10\rangle\Big(\alpha|0\rangle - \beta|1\rangle\Big) + \quad \texttt{; apply Z} \\ &+ \frac{1}{2}|11\rangle\Big(-\beta|0\rangle + \alpha|1\rangle\Big) \quad \texttt{; X, then Z}\end{aligned}$$

In this form it is visible what gates have to be applied[15] to the last qubit to make it the input teleported state $\alpha|0\rangle + \beta|1\rangle$.

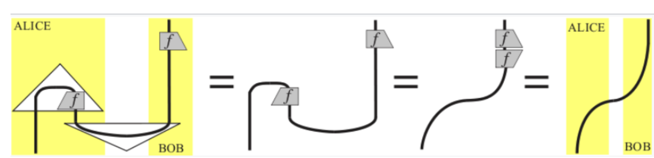
Let's now revisit, as we promised, the topic of tensorial product in the context of our derivation:

$$\begin{pmatrix}x_0\\ x_1\end{pmatrix}\otimes\begin{pmatrix}y_0\\ y_1\end{pmatrix}\otimes\begin{pmatrix}z_0\\ z_1\end{pmatrix} = \begin{pmatrix}x_0\\ x_1\end{pmatrix}\otimes\begin{pmatrix}y_0\begin{pmatrix}z_0\\ z_1\end{pmatrix}\\ y_1\begin{pmatrix}z_0\\ z_1\end{pmatrix}\end{pmatrix} = \begin{pmatrix}x_0\\ x_1\end{pmatrix}\otimes\begin{pmatrix}y_0 z_0\\ y_0 z_1\\ y_1 z_0\\ y_1 z_1\end{pmatrix}$$

This is exactly what is happening in our "abacus" calculations, for example in the first stage, as we determine $|\Psi_1\rangle$:




Two final comments in this section. First, that a(nother) diagrammatical proof of teleportation would look like this:




This is Penrose notation [6] and the approach is similar to what we saw when we mentioned the ZX-calculus. Note also that there is no transfer of matter or energy involved. No particle has been physically moved (from Alice to Bob); only its state has been transferred. The term "teleportation", coined by Bennett, Brassard, Crépeau, Jozsa, Peres and Wootters, reflects the indistinguishability of quantum mechanical particles.

## XI. Conclusion

So, in this paper we have argued that starting from the original misty state formalism one can successfully introduce the following topics to an audience that only knows basic arithmetic and simple operations on sets: superposition, interference (both constructive and destructive), entanglement (Bell states and GHZ states) and, as shown in this and other papers

[15] The gates to be applied depend on the measurement of the first two qubits, as teleported state is still entangled with them. That is the motivation behind the idea of classical correction, which is the last stage in this protocol (and indicated via annotations in this last equation).

(see references [17], [18], and [18]) the following cases of quantum advantage: Deutsch-Josza, Bernstein-Vazirani (based on the phase kickback phenomenon), superdense coding, the GHZ game (including the variant where we only use just the pure misty state formalism as demonstrated in Terry's book) and Grover's algorithm. For teleportaton we need the extended misty state formalism (MSF) and we showed that the extension is necessary if we want to deal with controlled Hadamard gates and arbitrary rotations. We called that example the break-even point of the quantum abacus[16].

## Appendix

This is probably a good place to review the basic rules of the "abacus" as presented in this paper:

- the superposition operator describes a qubit as a set of possible outcomes, with their associated probabilities
- the phase operator acts like the unary minus sign in a multiplication (or product)
- when applied to a set of possible outcomes the phase operator changes the sign (phase) on each of the elements in that set (of possible outcomes)
- the tensor product between two qubits is the cartesian product between the superposition sets representing those qubits (and the order matters)
- if we allow (as we did when we explained teleportation) complex coefficients in the superposition operator's representation introduced in the first few lines of the paper then the full generality of representation for qubits is achieved.

Compared to the ZX-calculus the system presented in our paper resembles the "slow-motion" replays in televised sports. A slow-motion replay is an exponential process and nobody would ever argue that it would be useful, advisable or otherwise meaningful to watch an entire game in slow motion. Graph-rewriting (mentioned in passing twice, with examples) has significant advantages over the string rewriting techniques that we presented. However graph rewriting is in effect an orthogonal process to what we advocated here; it can't provide any of the insight this slow-motion "abacus" technique provides[17]. Furthermore, a lot of the teaching (and learning) that happens when a student is first introduced to a complicated topic is, of necessity, of the slow-motion type.

Finally, we emphasized the implementation aspect and thus the connection with the online Qiskit textbook

## Acknowledgment

The authors would like to thank their colleagues in the QED-C[18] (especially the Workforce Development TAC) for numerous talks around how best these topics could be introduced to a noob (neophyte) audience. A summary of this paper was presented at the 2024 APS March Meeting [30].

[16]We still think about the extended MSF as an abacus of sorts.

[17]You can simplify a circuit with the ZX-calculus but then you may still want to trace it with the abacus.

[18]QED-C is a broad international group of stakeholders from industry, academia, national labs and professional organizations that aims to enable and grow the quantum industry and its associated supply chain. QED-C was established with support from NIST as part of the federal strategy for advancing QIST as per the National Quantum Initiative Act in 2018.

## References

[1] Bob Coecke and Aleks Kissinger, *Picturing Quantum Processes—A First Course in Quantum Theory and Diagrammatic Reasoning*, CUP, 2017.
[2] Sophia Economou, Terry Rudolph and Edwin Barnes, *Teaching quantum information science to high-school and early undergraduate students*, available at https://arxiv.org/abs/2005.07874, 2020
[3] Edwin Barnes, *Teaching QIS at the QISE Summer School organized by Virginia Tech* presentation at Spring 2022 CSAAPT Semi-Virtual Meeting available at https://indico.phys.vt.edu/event/48/contributions/996/
[4] Martin LaForêt, *Mathematics of Quantum Mechanics* at https://uwaterloo.ca/institute-for-quantum-computing/download-qcsys-math-primer
[5] John van de Wetering, *ZX-calculus for the working quantum computer scientist*, https://arxiv.org/abs/2012.13966v1, 2020
[6] Bob Coecke, "Quantum picturalism", *Contemporary Physics*, 2010.
[7] Terry Rudolph, *Q is for Quantum*, Terrence Rudolph 2017.
[8] Terry Rudolph, https://www.qisforquantum.org/ (book website)
[9] Yaoyun Shi, *Both Toffoli and Controlled-NOT need little help to do universal quantum computation* https://arxiv.org/abs/quant-ph/0205115
[10] Marek Narozniak, Simulating Quantum Teleportation, March 2020 available at https://mareknarozniak.com/
[11] Adrian German, Marcelo Pias, Qiao Xiang *On the Design and Implementation of a Quantum Architectures Knowledge Unit for a CS Curriculum* ACM, SIGCSE 2023, https://doi.org/10.1145/3545945.3569845
[12] Wikipedia article, https://en.wikipedia.org/wiki/Quantum_teleportation
[13] Diana Franklin, *Quantum Computing for Everyone (Part 1)*, EdX course available at https://www.edx.org/course/quantum-computing, 2021.
[14] Diana Franklin, *Quantum Computing for Everyone (Part 2)*, 2021. www.edx.org/course/introduction-to-quantum-computing-for-everyone-2
[15] Abe Asfaw, Programming in IBM Qiskit (Season 1, Ep. 6: Bernstein-Vazirani) https://www.youtube.com/watch?v=sqJIpHYl7oo&t=6s
[16] Bettina Just, *Quantum Computing Compact: Spooky Action at a Distance and Teleportation Easy to Understand*, Springer Nature 2022.
[17] Adrian German, Marcelo Pias, Qiao Xiang and Sreesha Kuruvadi, *A Quantum Abacus for Teaching Quantum Algorithms*, available at https://legacy.cs.indiana.edu/~dgerman/2023/fie/07/30/camera-ready.pdf. Paper presented at FIE 2023, October 2023, College Station, TX.
[18] Adrian German and Alex Alani, *The GHZ Game with Misty States*, August 2023, preprint available at https://legacy.cs.indiana.edu/~dgerman/2023/qseec/ghz-game-with-misty-states.pdf
[19] Alex Alani, Adrian German and Terry Rudolph, *The GHZ Game in Black and White*, August 2023, preprint available at https://legacy.cs.indiana.edu/~dgerman/2023/qseec/ghz-game-in-black-and-white.pdf
[20] Old Qiskit Textbook, Superdense Coding Python Notebook, available at https://github.com/Qiskit/textbook/blob/main/notebooks/ch-algorithms/superdense-coding.ipynb
[21] Macauley Coggins, *Introduction to Quantum Computing with Qiskit*, available at https://quantumcomputinguk.org/tutorials/superdense 2019
[22] Terry Rudolph, https://www.qisforquantum.org/supplementary-material
[23] Andrew Helwer, *Quantum Computing for Computer Scientists* (video, 02/14/2016) https://www.youtube.com/watch?v=F_Riqjdh2oM
[24] Andrew Helwer, slides https://ahelwer.ca/files/qc-for-cs.pdf Slides
[25] John Watrous, Lesson 03: Quantum Circuits (on the YouTube Qiskit channel) available at https://www.youtube.com/watch?v=30U2DTfIrOU&t=650s The link takes you directly to that moment.
[26] Peter Shor, MIT Open Courseware, Lecture 12: Superdense Coding, lecture by Peter Shor, scribed by Sebastian Bauer. 10/14/2003. Available at https://ocw.mit.edu/courses/18-435j-quantum-computation-fall-2003/4f97e39d2509d075121823e6b999455c_qc_lec12.pdf
[27] John Preskill, Online Notes for PHYS 219 and CSCI 219 Quantum Computation. This is chapter 4: Quantum Entanglement available at http://theory.caltech.edu/~preskill/ph229/notes/chap4.pdf
[28] Old Qiskit Textbook, Grover's Algorithm Python Notebook, available at https://github.com/Qiskit/textbook/blob/main/notebooks/ch-algorithms/grover.ipynb (GitHub repository)
[29] Dayton Ellwanger, *A Quantum Computing Pamphlet*, custom published, Monee, IL, 02/15/2020. Concise presentation of Grover (linear algebra).
[30] Adrian German, APS March Meeting 2024, abstract at https://meetings.aps.org/Meeting/MAR24/Session/K61.4