# Quantum Game Theory and Ideal Quantum Strategies

Noah Dane Hebdon

A senior thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Bachelor of Science

Jean-Francois Van Huele, Advisor

Department of Physics and Astronomy

Brigham Young University

ABSTRACT

Quantum Game Theory and Ideal Quantum Strategies

Noah Dane Hebdon
Department of Physics and Astronomy, BYU
Bachelor of Science

Game theory is the study of strategic interactions between rational actors. Classical games within this field, such as the penny flip and the prisoner's dilemma, may be modeled with quantum systems, where select players leverage principles like superposition and entanglement to maximize their payoff. Such games offer a platform to investigate notions of "quantum advantage" over classical strategies. Presenting a striking example of quantum advantage, Meyer [1] claims that quantum strategies guarantee victory against a classical player within the PQ penny flip game, which he devised. Using IBM quantum simulators [2], I first investigate whether this quantum advantage is solely attributable to the use of quantum strategies. I then generalize Meyer's quantum strategy to games with an arbitrary number of plays and to games featuring abstract multi-sided objects. I also demonstrate that in a variation of Meyer's three-play game biased in favor of the classical player, no quantum strategy may ensure victory. Throughout this work, I show that quantum advantage within the PQ penny flip game is genuine, although not absolute.

Keywords: quantum game theory, penny flip game, quantum simulators, IBM quantum computing, quantum computing applications

## ACKNOWLEDGMENTS

I acknowledge the support of the College of Computational, Physical and Mathematical Sciences of Brigham Young University. I also express gratitude to my advisor, Dr. Jean-Francois Van Huele, and to the members of the Quantum Information Dynamics research group within the Department of Physics and Astronomy for their suggestions, guidance, and assistance. IBM Quantum services were utilized in the course of this research, although the findings and thoughts presented by the author are his alone. As such, they reflect no position or policy of IBM or the IBM Quantum team.

# Contents


**Table of Contents** **v**

**List of Figures** **vii**

**1 Introduction** **1**
1.1 Game Theory . . . . . . . . . . 1
1.2 Quantum Computing . . . . . . . . . . 2
1.2.1 Overview . . . . . . . . . . 2
1.2.2 IBM Quantum Platform and Qiskit . . . . . . . . . . 2
1.3 Quantum Game Theory . . . . . . . . . . 3

**2 Methods** **5**
2.1 Penny Flip Game . . . . . . . . . . 5
2.1.1 Classical Penny Flip . . . . . . . . . . 5
2.1.2 PQ Penny Flip . . . . . . . . . . 6
2.2 Verification of Quantum Advantage . . . . . . . . . . 8
2.2.1 Initial Tails State . . . . . . . . . . 8
2.2.2 Two Plays to Classical Player . . . . . . . . . . 9
2.3 Generalized Findings . . . . . . . . . . 15
2.3.1 Multi-Play Games . . . . . . . . . . 15
2.3.2 Multi-Sided Objects . . . . . . . . . . 16

**3 Discussion** **19**
3.1 Results . . . . . . . . . . 19
3.2 Conclusion . . . . . . . . . . 20

**Appendix A Computational Methods and Proofs** **21**
A.0.1 Operator Verification: Initial Tails State . . . . . . . . . . 21
A.0.2 Operator Identification: Biased Game, Draft 1 . . . . . . . . . . 23
A.0.3 Operator Identification: Biased Game, Draft 2: Uncorrelated Plays . . . . . 26
A.0.4 Operator Identification: Biased Game, Draft 2: Correlated Plays . . . . . . 27
A.0.5 Operator Verification: Biased Game . . . . . . . . . . 29

A.0.6 Density Matrix Proof . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

**Bibliography** **39**

# List of Figures


1.1 Example Circuit from IBM Quantum Platform . . . . . . . . . . . . . . . . . . . . . 3

2.1 Quantum Circuit for Uncorrelated Classical Plays . . . . . . . . . . . . . . . . . . . 10

2.2 Results for Uncorrelated Classical Plays . . . . . . . . . . . . . . . . . . . . . . . 11

2.3 Quantum Circuit for Correlated Classical Plays - Identical Plays . . . . . . . . . . 12

2.4 Results for Correlated Classical Plays - Identical Plays . . . . . . . . . . . . . . . 13

2.5 Quantum Circuit for Correlated Classical Plays - Opposite Plays . . . . . . . . . . 13

2.6 Results for Correlated Classical Plays - Opposite Plays . . . . . . . . . . . . . . . 14

2.7 Games of Multiple Plays . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

# Chapter 1

# Introduction

## 1.1 Game Theory

Game theory, first formulated in the mid-twentieth century, concerns the study of strategic interaction between rational actors [3]. These actors, with access to a defined set of strategies for a given interaction, attempt to maximize their payoff by encouraging an outcome with a pre-established reward. These interactions include games such as the penny flip game, the prisoner's dilemma, or even chess.

By analyzing various strategies within these small confrontations, economists, political scientists, and military strategists can maximize their payoff, leading to victory in the stock market [4], on the campaign trail [5], and even on the battlefield [6]. As such, the scope of game theory's application is wide.

## 1.2 Quantum Computing

### 1.2.1 Overview

In recent years, the application of quantum mechanics to computational challenges has aroused great excitement [7]. A quantum system refers to any microscopic system whose states take on only discrete or quantized values. For example, the spin of an electron is described by quantum mechanics because it may only take on discrete values of ±1/2. In contrast, macroscopic objects can "spin" or self-rotate at any value of angular momentum.

In many instances, the development of quantum computing has been validated. Shor's quantum algorithm, for instance, solves the problem of prime factorization exponentially faster than its classical counterpart [8]. Grover's algorithm, a quantum search algorithm for unsorted databases, outperforms its classical equivalent by a quadratic amount [9]. Quantum supremacy, at least in some cases, seems unquestionable, motivating the development of other methods of quantum computation and hardware as well as the widening of its application.

### 1.2.2 IBM Quantum Platform and Qiskit

Through their "Quantum Platform," IBM is working to make quantum computing accessible to anyone with a laptop. With a Python package known as Qiskit, any enthusiast is able to utilize IBM's simulators and superconducting quantum computers. To send computational tasks to some piece of hardware, a user simply constructs a quantum circuit designed around their computational issue. In general, the user specifies the number of qubits - the quantum-equivalent of a bit - needed for their computation. Each qubit is represented as a line at the top of a quantum circuit, as shown in Figure 1.1. The user also specifies the number of classical bits needed to store the results of any measurements conducted on the qubits in question. These are labeled as $A$ and $B$ in Figure 1.1. Throughout a computation, time moves from left to right as a user applies unitary operators to the qubits. These operators, referred to as "gates," are used to manipulate the qubits. The measurement gates determine the state of a qubit at a certain point in time and store its value is one of the bits mentioned before.

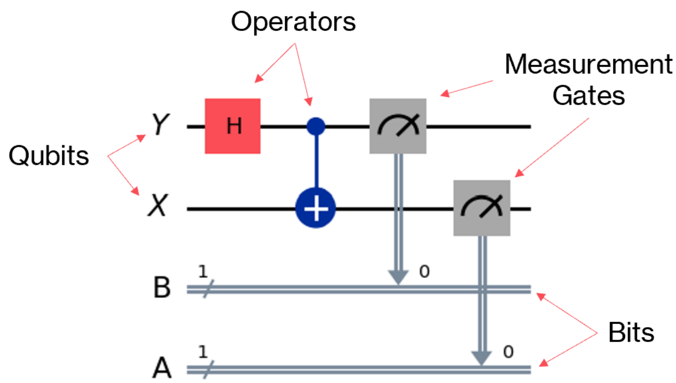


**Figure 1.1** Example Circuit from IBM Quantum Platform

These circuits can be designed to handle any number of computational tasks. This is due to the existence of universal quantum gates, which can be used to approximate any unitary operation on a quantum system to arbitrary accuracy. Quite simply, the existence of a universal set of quantum gates allows any quantum circuit or quantum algorithm to be constructed. As we will show, these circuits are particularly useful when modeling games and other competitive exchanges of information.

## 1.3 Quantum Game Theory

One development within quantum computing was instigated in 1998 by D. A. Meyer. Ultimately, he built a bridge between game theory and quantum computing, asserting that one may not only successfully represent interactions between rational actors with quantum systems, but one may find promising new strategies in these systems as well [1]. These new strategies incorporate quantum principles of superposition, entanglement, and uncertainty, ultimately providing a quantum player with access to strategies that a classical player cannot claim, conferring upon them a certain "quantum advantage."

Meyer first introduced quantum game theory with a simple penny flip game; a game lucid enough to clearly illustrate the advantages of employing quantum strategies. Other games have also been “quantized,” such as Prisoner’s Dilemma [10] and even Rock, Paper, Scissors [11]. In such games, however, the plays do not build off one another, and the players do not manipulate a central object, making the application of superposition and other quantum principles more difficult. The current work is centered around the penny flip game, its quantum representation - the PQ penny flip game - first devised by Meyer, and the advantage a quantum player enjoys. Some of the results found in the following sections have been presented at conferences of the American Physical Society and the Utah Academy of Sciences, Arts, and Letters and have also been published separately [12].

# Chapter 2

# Methods

We begin with the classical description of the penny flip game in Section 2.1.1. Then, an introduction to quantum principles will be presented, followed by details regarding the quantum representation of the penny flip game in Section 2.1.2. Then, in Section 2.2, Meyer's findings will be critically discussed and followed by an analysis of the limitations of quantum advantage.

## 2.1 Penny Flip Game

### 2.1.1 Classical Penny Flip

The classical penny flip game involves two players which we will call $C$ and $Q$ for later convenience. There is a penny between the two players, whose initial state is heads, which is known to both participants. The penny, however, is then placed in a box and hidden from both players. Both players have access to the following strategies: Flip the penny so that the other side is facing up ($F$); Don't flip the penny ($N$). After the penny is placed in the box, one game includes three moves, including an opening move from $Q$, a move from $C$, and a closing move from $Q$. After these three moves, the penny is revealed. If the final state of the penny is heads-up, then $Q$ wins. Otherwise, $C$

wins. In this case, each player has a 50% chance of victory. With the details of the game set forth, we now turn to an explanation of its quantum formulation.

### 2.1.2 PQ Penny Flip

In classical descriptions, any state may be represented as a collection of bits. For our penny, the states Heads ($|H\rangle$) and Tails ($|T\rangle$) can be defined as follows, utilizing Dirac notation and matrix representations [13]:

$$|H\rangle = |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |T\rangle = |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2.1}$$

In quantum mechanics, however, each state is represented as a linear combination of basis states. Generally, a complex two-dimensional vector space with orthonormal basis vectors $|0\rangle$ and $|1\rangle$ is used within quantum computing. Our case is no different. We can then represent our penny $|\Psi\rangle$ in this complex vector space as follows, where $\alpha$ and $\beta$ are complex numbers:

$$|\Psi\rangle = \alpha |H\rangle + \beta |T\rangle. \tag{2.2}$$

With this description, use is made of superposition, where a probability ($|\alpha|^2$ or $|\beta|^2$) is associated with each basis state. For this reason, it must be the case that

$$|\alpha|^2 + |\beta|^2 = 1. \tag{2.3}$$

The system, however, exists between these two basis states until measured. Imagine our penny standing on its edge rather than being purely characterized as heads or tails. In a classical setting, any state is manipulated with matrix multiplication. For example, the classical play $F$ is applied to $|H\rangle$ as a matrix operation, producing $|T\rangle$:

$$F|H\rangle = |T\rangle \quad \text{or} \quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2.4}$$

The other available classical play $N$ is given by the identity matrix and does not alter the state of the penny:

$$N|H\rangle = |H\rangle \quad \text{or} \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{2.5}$$

In a quantum setting, similarly, operators must take a physical state as an input and then produce a physical state as an output. In other words, the magnitude of a vector in the complex vector space cannot be compromised by any application of an operator. As such, any operator must be unitary, meaning that its inverse must be equal to its conjugate transpose. A quantum play could then be given by a Hadamard matrix, an intrinsically quantum matrix that produces superposition in a system. Such matrices have rows that are mutually orthogonal, have entries limited to $\pm 1$, and may be devised with the Paley construction [14]:

$$U_H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{2.6}$$

It is important to note that players limited to classical strategies may select only one of two plays ($F$ or $N$), both of which are inherently unitary, while a quantum player has access to unlimited plays, given that their matrix equivalent is unitary.

Meyer showed that the application of the above quantum strategy ($U_H$) renders victory for our quantum player regardless of the play that is made by our classical player [1]. This is because the Hadamard operator, when applied as the opening and closing moves of the penny flip game, ultimately eliminates the action of the classical player. This is shown as follows:

$$U_H F U_H |H\rangle = |H\rangle, \tag{2.7}$$

$$U_H N U_H |H\rangle = |H\rangle. \tag{2.8}$$

Such a result from Meyer, it seems, is a mighty display of the advantages of quantum information exchange over its classical counterpart. However, could one more appropriately attribute the advantage not to quantum strategies, but to the arbitrary construction of the penny flip game? After all, the quantum player is given two turns, including the opening and closing plays, while the classical player is given only one. Such is the central question of this work.

## 2.2 Verification of Quantum Advantage

To verify that the advantage of our quantum player is actually derived from quantum strategies, the construction of the penny flip game was altered, and ideal quantum strategies identified. The probability of quantum victory was then assessed to determine whether quantum advantage was preserved despite added biases in favor of our classical player. These biases included the alteration of the initial state of the penny as well as the number of moves granted to each player.

### 2.2.1 Initial Tails State

Firstly, the initial state of the penny was altered to begin as $|T\rangle$ rather than $|H\rangle$, seemingly giving the classical player a subtle advantage that the quantum player once enjoyed. An ideal quantum strategy in response to this was identified with ease:

$$U_{Q1} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}. \tag{2.9}$$

When applied, this strategy guarantees quantum victory as before, strengthening the assumption that the advantage of our quantum player is derived primarily from the use of quantum strategies:

$$U_{Q1} F U_{Q1} |T\rangle = |H\rangle, \tag{2.10}$$

$$U_{Q1} N U_{Q1} |T\rangle = |H\rangle. \tag{2.11}$$

To model this situation and verify our analytic solution, Qiskit was utilized to create a quantum circuit. The reader may refer to Section A.0.1 for the associated code. After running on an IBM quantum simulator, the circuit produced a result of 100% chance of quantum victory [2].

### 2.2.2 Two Plays to Classical Player

The game was then further biased against our quantum player. Specifically, the classical player was granted the opening and closing plays, leaving our quantum player with only one move. In this case, however, the beginning state of the penny was $|H\rangle$ once again. Within this game construction, two scenarios were investigated: one in which there was a 50% chance of the classical player flipping the penny during each of their turns and one in which the classical plays were correlated. We begin by considering the former.

#### Uncorrelated Plays

For this portion, Qiskit was employed in the identification of an ideal strategy. Further details are given in Section A.0.3. As shown in Figure 2.1, the quantum circuit for this case included one qubit to function as the penny. Three operators or gates were also included to function as each play in the game. The gate below labeled as "Unitary" on the line associated with the "Penny" qubit signifies the quantum play. The two blue "+" operators, also known as Controlled-Not or CNOT operators, function as the classical moves. At the very end of the circuit, the penny is measured, indicated by the black measurement icon, while its final state is stored in a classical bit on the bottom line, indicating the game's victor. Also, notice the other two qubits $C1$ and $C2$ included above the "Penny" qubit. These qubits function as random number generators to approximate the behavior of a classical player that chooses randomly between the $F$ and $N$ plays. Using Hadamard operators, indicated by the blue "H" icons, both $C1$ and $C2$ are placed in a superposition of both $|0\rangle$

and $|1\rangle$. They are then individually measured using the CNOT gates, possessing a 50% of being $|0\rangle$. Based on the outcome of each measurement, the penny is either flipped or not affected.

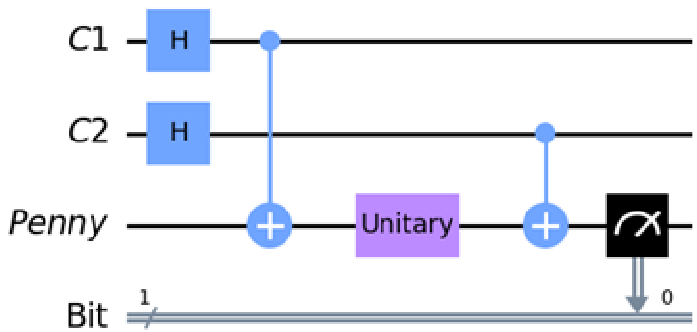


**Figure 2.1** Quantum Circuit for Uncorrelated Classical Plays

For this scenario, the circuit was run one hundred times with a unique and randomly produced quantum play. With the intent of identifying the quantum operator that produced the greatest likelihood of quantum success, results for each quantum operator were compared. For each of the one hundred operators, however, the simulator ran the circuit one hundred times and produced the probabilistic results illustrated in Figure 2.2:

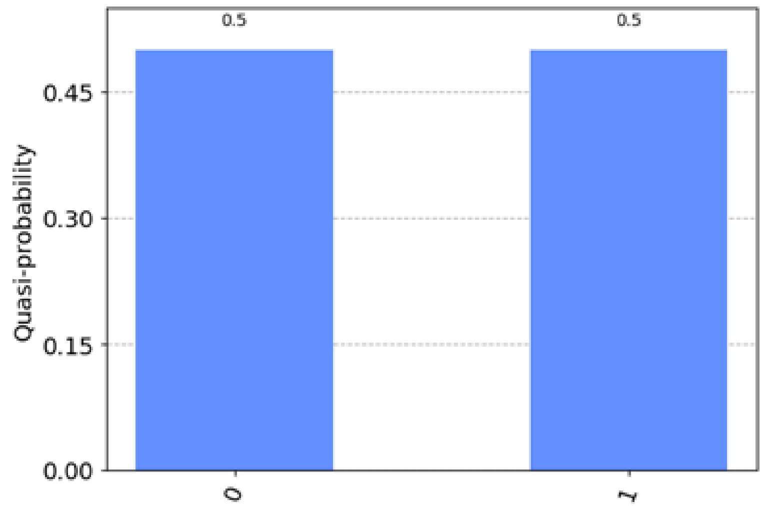


**Figure 2.2** Results for Uncorrelated Classical Plays

The leftmost column relates to the $|0\rangle$ or $|H\rangle$ state, and the other to the $|1\rangle$ or $|T\rangle$ state. The magnitude of each column reflects the probability that each state is found to be the final state of the penny. Clearly, the likelihood of quantum victory is 50%, regardless of the quantum operator utilized. This makes sense. The best the quantum player can do is put the penny into a superposition of $|H\rangle$ and $|T\rangle$, making the probability of each state 50%. Had they been given the last play, they could also pull the penny out of the superposition into the $|H\rangle$ state as before. Hence, in this case, any quantum operator produces the maximum chance of quantum victory, which is 50%. With this result, it appears that the advantage of two plays outweighs the advantage provided by quantum strategies.

### Correlated Plays

We now consider the case where the classical plays are correlated. It should be noted that correlated plays may be identical to one another or the opposite of each other. Of these two cases, we first consider the case where the classical plays are the same. Hence, the classical plays are either both $F$ or both $N$. Removing the $C2$ qubit and having $C1$ act as the control qubit for both CNOT gates allows us to represent the scenario where the plays of the classical player are identical. The resulting quantum circuit was constructed with the code found in Section A.0.4 and is given as follows:

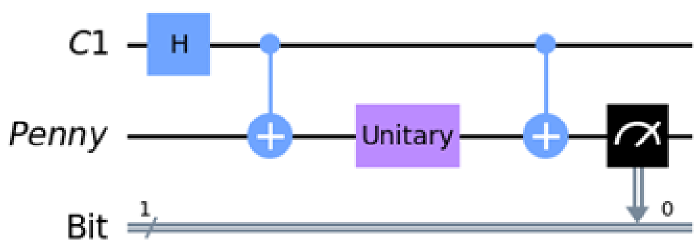


**Figure 2.3** Quantum Circuit for Correlated Classical Plays - Identical Plays

Once again, the circuit was run one hundred times on an IBM quantum simulator, with a randomly produced unitary operator acting as the quantum play. Then, the operator producing the greatest chance of quantum victory was recorded. It was found that a quantum strategy does exist that guarantees quantum victory in the case where the plays made by the classical player are identical. The ideal strategy is given as follows:

$$U_{Q2} = \begin{bmatrix} -0.59021396+0.80351867i & -0.02145772-0.07446331i \\ 0.05446345-0.05512667i & -0.13393704-0.98795529i \end{bmatrix} \tag{2.12}$$

For verification, this operator was inserted as the "Unitary" gate on the quantum circuit of Figure 2.3 and run another hundred times. The code developed for this step is given in Section A.0.5. For each iteration, a 99% chance of quantum victory was discovered.

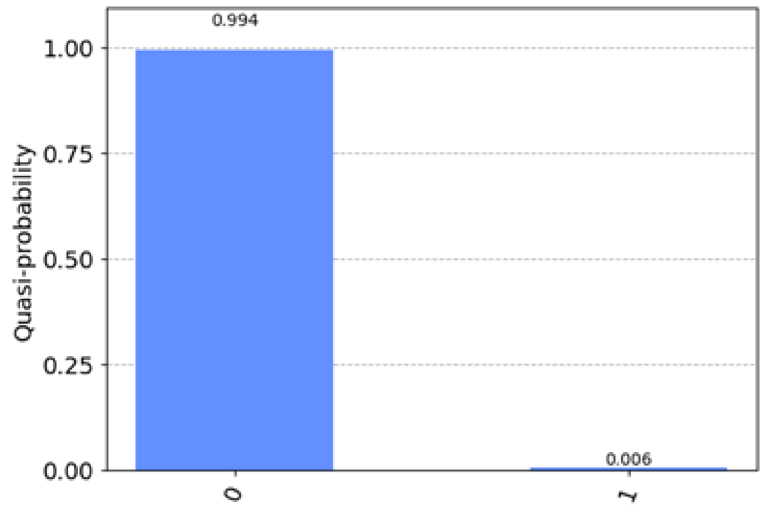


**Figure 2.4** Results for Correlated Classical Plays - Identical Plays

In the other case of correlated classical plays, the opening and closing moves are dissimilar. The quantum circuit was only slightly modified to discover an ideal quantum operator. Specifically, an X-gate was added to the $C1$ qubit path between the two CNOT gates. This guarantees that the opening and closing classical plays are opposites.

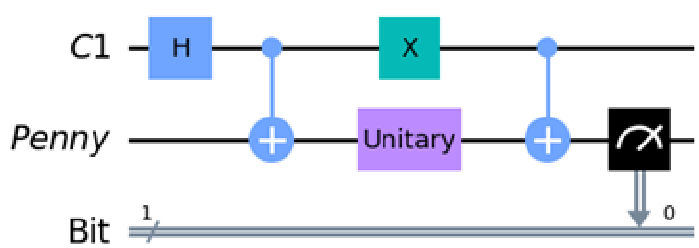


**Figure 2.5** Quantum Circuit for Correlated Classical Plays - Opposite Plays

Again, the circuit was run one hundred times with a unique and random quantum operator. From the one hundred histograms produced, an ideal quantum operator was identified:

$$U_{Q3} = \begin{bmatrix} -0.18287237 - 0.00699522349i & -0.38743878 + 0.903548534i \\ 0.35143341 + 0.918152122i & -0.18300595 - 0.000236252265i \end{bmatrix}$$

This ideal operator was then plugged directly into the circuit of Figure 2.5 and run another hundred times (A.0.5). The verified histogram, showing a 96.7% chance of quantum victory, was produced as shown in Figure 2.6. We may assume that if more than one hundred unique quantum plays were developed, a strategy guaranteeing greater than a 96.7% chance of quantum victory could be discovered.

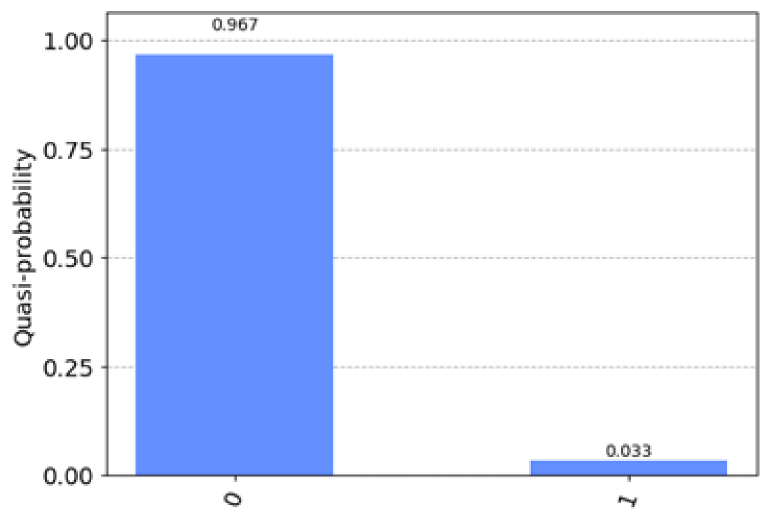


**Figure 2.6** Results for Correlated Classical Plays - Opposite Plays

So, in the case where the classical player has both the opening and closing moves, the quantum player has a probability of victory of 50%-100%, ultimately depending on whether the classical moves are correlated. In this case, it does not matter whether $F$ or $N$ is played first by the classical player. It should also be noted, however, that in the case of correlated classical plays, a quantum player may only increase their chance of victory to nearly 100% if they know which of the two ideal operators to employ. That is, they must have some valid prediction as to the behavior or strategy of the classical player. Guessing randomly between the two strategies found above in the case of correlated classical plays, the quantum player has only a 50% chance of victory.

Density matrices were also employed in the course of this work to show that when the classical player is granted the opening and closing moves, the quantum player is unable to guarantee victory for herself with just one move. In Section A.0.6, a formal proof is given to justify this finding.

## 2.3 Generalized Findings

It is possible to generalize the above findings to other variations of the penny flip game. To that end, we first expand the penny flip game to an arbitrary number of moves. This is found in Section 2.3.1. Then, in Section 2.3.2, we generalize our results to games involving abstract objects with more than two basis states.

### 2.3.1 Multi-Play Games

We divide all possible games into those of an odd number of plays and those with an even number of plays. The plays alternate between players $Q$ and $C$ as before.

| Odd-Play Games | Chance of QV | Even-Play Games | Chance of QV |
|---|---|---|---|
| $U_QU_CU_Q\|H\rangle = \|H\rangle$ | 100% | $U_QU_C\|H\rangle = \|H\rangle$ | 50% |
| $U_CU_QU_C\|H\rangle = \|H\rangle$ | 50% | $U_CU_Q\|H\rangle = \|H\rangle$ | 50% |
| $U_QU_CU_QU_CU_Q\|H\rangle = \|H\rangle$ | 100% | $U_QU_CU_QU_C\|H\rangle = \|H\rangle$ | 50% |
| $U_CU_QU_CU_QU_C\|H\rangle = \|H\rangle$ | 50% | $U_CU_QU_CU_Q\|H\rangle = \|H\rangle$ | 50% |
| ⋮ | | ⋮ | |

**Figure 2.7** Games of Multiple Plays

As seen in Figure 2.7, the likelihood of quantum victory is highly dependent on the number of moves within the game. It should be noted that when one unitary matrix is applied to another, the resulting matrix is also unitary. Hence, in an odd-play game, all moves between the opening and closing quantum moves can be collapsed into a single unitary operator. This yields the three-play game, in which an ideal strategy has already been found. Within games of an even number of plays, the quantum player may only guarantee a 50% chance of victory, regardless of whether they possess the opening or closing move. Once again, the quantum player cannot place the penny in a superposition of $|H\rangle$ and $|T\rangle$ and still pull it back into the heads state without interference from the classical player. Therefore, within the penny flip game of an arbitrary number of moves, the quantum player has a 50%-100% chance of victory.

### 2.3.2 Multi-Sided Objects

We now consider games of three plays involving an abstract object of an arbitrary number of sides. This may be visualized with a die. Initially, we discuss a four-sided object but later show how the results may be generalized. Our object then becomes a quantum state involving four basis states that we will associate with color for simplicity: Red ($|R\rangle$), Yellow ($|Y\rangle$), Green ($|G\rangle$), and Blue ($|B\rangle$). Our penny is then shown as follows:

$$|\Psi\rangle = \alpha\,|R\rangle + \beta\,|Y\rangle + \delta\,|G\rangle + \gamma|B\rangle\,, \tag{2.13}$$

where

$$|\alpha|^2 + |\beta|^2 + |\delta|^2 + |\gamma|^2 = 1. \tag{2.14}$$

In a game such as this, the classical play of flipping may be equated with altering the color of the penny. The classical player may convert the color from red to yellow, for example. These classical moves are easily defined in a matrix where the transpose of the target state is placed in the same row as the "1" that appears in the state that is being manipulated. Similarly, the transpose of the state that is being manipulated is placed in the same row as the "1" that appears in the target state. The remaining rows are filled out such that "1" occurs only once in each row and column. Operators defined this way, although not unique, satisfy the requirements of unitary matrices. They allow us to "rotate" the multi-sided object in our Hilbert space, mimicking the act of flipping a penny as before. For example, define ($|R\rangle$) and ($|Y\rangle$) as follows:

$$|R\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad |Y\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}. \tag{2.15}$$

The classical operator to convert $|R\rangle$ to $|Y\rangle$ is easily produced, following the guidelines above.

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

If we are to follow the strategies already discussed above, then supplying a Hadamard matrix as our quantum strategy will guarantee victory in a three-play game. We find this to be exactly the case. In four dimensions, the Hadamard is constructed and applied as follows, placing our object into a superposition of $|R\rangle$, $|Y\rangle$, $|G\rangle$, $|B\rangle$:

$$\frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

With this strategy, given the opening and closing moves of a three-play game, our quantum player is able to nullify any move made by the classical player, making them victorious 100% of the time. It follows that our quantum player may guarantee victory when there exists a Hadamard operator in the dimension that matches the number of basis states that describe our object. It is a known fact that there exists an $N \times N$ Hadamard matrix, where $N = 1, 2$, or $4k$, for $k \in \mathbb{Z}$ [14]. Hence, victory is guaranteed for our quantum player in any game involving an $N$-sided penny.

For the cases in which $N \neq 1, 2$, or $4k$, for $k \in \mathbb{Z}$, it is possible to place the object into a superposition of all states. In particular, one may employ a discrete Fourier transform matrix as a quantum strategy, as shown below:

$$F_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix},$$

where

$$\omega = e^{-2\pi i/N}.$$

However, in this case, when the operator is applied again as a closing move, the quantum player cannot "pull" the object back into a desired state to guarantee victory. Despite the current issues, there is great promise in the discovery of quantum strategies that may guarantee victory in a game including an object where $N$ may take any integer-value.

# Chapter 3

# Discussion

## 3.1 Results

When the penny begins in state $|T\rangle$, the desired final state for our classical player, there exists a quantum operator that guarantees quantum victory. This was shown in Section 2.2.1. When the classical player is given two plays, there exist operators that provide a 50%-100% chance of victory for our quantum player, depending on the assumption made by the quantum player as to the correlation of the classical plays. If the classical plays are uncorrelated, as was shown in Figure 2.2, there exists a 50% chance of quantum victory. However, when the classical plays are correlated, Figure 2.4 demonstrates that we may guarantee up to a 99% chance of quantum victory. In either case, this is far greater than the 0%-50% chance of victory that the classical player possesses when the quantum player is given the same two-play advantage.

It was also shown in Section 2.3.1 that quantum victory is guaranteed in modified games of an odd number of plays where the quantum player is given both the opening and closing plays. As was shown in Section 2.3.2, the same guarantee of quantum victory persists in games involving an abstract object with N basis states, where $N = 1, 2$, or $4k$, for $k \in \mathbb{Z}$.

## 3.2 Conclusion

The advantage enjoyed by the quantum player in Meyer's penny flip game is generally authentic, being derived from quantum principles rather than the arbitrary construction of the penny flip game. As has been shown, quantum representations of competitive exchanges of information provide strategies that are generally superior to their classical counterparts, not only for Meyer's penny flip, but also for its more general variations. Shor's algorithm [8], Grover's algorithm [9], and others have already provided examples of the superiority of quantum computation over classical information exchange. Yes, quantum mechanics serves well to find the prime factorization of large integers and search through unsorted databases. However, we may now also say that quantum mechanics provides superior strategies for specific scenarios in game theory, bringing us one step closer to the application of these strategies to competitive environments wherein game theory is already applied, including the stock market, the campaign trail, and even the battlefield.

# Appendix A

# Computational Methods and Proofs

Sections A.0.1 - A.0.5 contain Qiskit code used in the production of the quantum circuits described in earlier sections. It should be noted that Qiskit packages have been updated since these files were written. As such, it will be necessary to download the Qiskit "primitives" package before executing any code found below.

Finally, Section A.0.6 includes a proof showing that the quantum player is unable to guarantee victory for herself with just one move when the classical player is granted the opening and closing moves within the PQ penny flip game. Within this proof, density matrices are used, facilitating the inclusion of the parameters $p$ and $q$, which corresponds to the probability that the classical player makes a certain move.

### A.0.1 Operator Verification: Initial Tails State

```

from qiskit import QuantumRegister, QuantumCircuit, execute, IBMQ
from qiskit_ibm_runtime import QiskitRuntimeService
from qiskit.visualization import plot_histogram
from qiskit import *
```

```python
from math import sqrt

service = QiskitRuntimeService() # Initialize Qiskit Runtime service

quant = Operator([[1/sqrt(2), -1/sqrt(2)], [1/sqrt(2), 1/sqrt(2)]])  # Ideal quantum strategy
flip = Operator([[0, 1], [1, 0]])  # Classical plays
noFlip = Operator([[1, 0], [0, 1]])

q = QuantumRegister(1, "q") # Create quantum register and circuit
qc = QuantumCircuit(q)

qc.append(quant, q) # Append operations to quantum circuit
qc.append(flip, q)
qc.append(quant, q)
qc.measure_all()

provider = IBMQ.get_provider(hub="ibm-q", group="open", project="main") # Define backend provider and simulator
backend = provider.get_backend("ibmq_qasm_simulator")

job = execute(qc, backend, shots=1024) # Execute circuit on simulator backend
print('Executing Job...\n')
job_monitor(job)

result = job.result() # Get results and plot histogram
counts = result.get_counts()
print("Counts: ", counts)

```

```
plot_histogram(counts)
```

### A.0.2 Operator Identification: Biased Game, Draft 1

```
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit,
    execute
from qiskit_ibm_runtime import QiskitRuntimeService
from qiskit.quantum_info.operators import Operator
from qiskit.quantum_info import random_unitary
from qiskit.tools.monitor import job_monitor
from qiskit_ibm_provider import IBMProvider
from qiskit.extensions import UnitaryGate
from qiskit import *

service = QiskitRuntimeService() # Initialize the IBM Quantum service
circuit_list = []  # List to store quantum circuits
operators = []  # List to store randomly generated unitary operators

def randNum():
    global zero, one

    q = QuantumRegister(1, 'q')
    c = ClassicalRegister(1, 'c')

    circuit = QuantumCircuit(q, c)  # Create quantum circuit with one
        qubit and one bit
    circuit.h(q)  # Apply Hadamard gate to place qubit in superposition
    circuit.measure(q, c)  # Measure qubit

```

```
    provider = IBMProvider(instance="ibm-q/open/main") # Execute circuit on IBM Quantum backend
    backend = provider.get_backend("ibm_perth")
    job = execute(circuit, backend, shots=1)

    print('Executing Job...\n')
    job_monitor(job)
    counts = job.result().get_counts()
    one = counts.get("1")  # Store count of outcome "1" or tails
    zero = counts.get("0")  # Store count of outcome "0" or heads
    print(counts)

for i in range(100): # Generate 100 quantum circuits

    randNum()

    penny = QuantumRegister(1, "Penny") # Create new quantum register for each circuit
    qc = QuantumCircuit(penny)

    rand = UnitaryGate(random_unitary(2)) # Generate random unitary gate

    # Define Pauli-X (flip) and Identity (no flip) operators
    flip = Operator([[0, 1], [1, 0]])
    noFlip = Operator([[1, 0], [0, 1]])

    if one == 1: # Apply classical play based on quantum-generated bit
        print("Classical Play: Flip")
        qc.append(flip, penny)
    else:
```

```
        print("Classical Play: No Flip")
        qc.append(noFlip, penny)

    qc.append(rand, penny) # Apply random unitary gate
    operators.append(rand.to_matrix())

    randNum()  # Generate another random bit

    if one == 1: # Apply second classical play based on quantum-generated bit
        print("Classical Play: Flip")
        qc.append(flip, penny)
    else:
        print("Classical Play: No Flip")
        qc.append(noFlip, penny)

    qc.measure_all() # Measure the qubit

    # Store the circuit in the list
    circuit_list.append(qc)

provider = IBMProvider(instance="ibm-q/open/main") # Execute all circuits on IBM Quantum backend
backend = provider.get_backend("ibm_perth")
job = execute(circuit_list, backend, shots=100)

for i in operators: # Save the unitary operators to a .txt file
    file1 = open("operators.txt", "a")
    file1.write(str(operators[i]) + "\n")
    file1.close()
```

### A.0.3 Operator Identification: Biased Game, Draft 2: Uncorrelated Plays

```
from qiskit import QuantumRegister, QuantumCircuit
from qiskit_ibm_runtime import QiskitRuntimeService
from qiskit.quantum_info import random_unitary
from qiskit.extensions import UnitaryGate
from qiskit.primitives import Sampler
from qiskit.visualization import plot_histogram

service = QiskitRuntimeService() # Initialize IBM Quantum service

circuit_list = []

for i in range(100): # Create and store 100 quantum circuits

    q = QuantumRegister(3, "q")
    c1 = ClassicalRegister(1, 'c1')

    qc = QuantumCircuit(q, c1) # Define a quantum circuit with three qubits and 1 bit

    qc.h(0) # Apply Hadamard gate to first two qubits to create superposition
    qc.h(1)

    qc.cx(q[0], q[2]) # Apply a CNOT gate from qubit 0 (control) to qubit 2 (target)

    rand = UnitaryGate(random_unitary(2))
    qc.append(rand, [2])  # Apply random 2x2 unitary matrix

```

```
    qc.cx(q[1], q[2]) # Apply another CNOT gate from qubit 1 (control) to qubit 2 (target)

    qc.measure(2, 0)

    circuit_list.append(qc)

sampler = Sampler() # Initialize Sampler primitive to run quantum circuits
results = sampler.run(circuit_list).result()

statistics = results.quasi_dists[0].binary_probabilities() # Extract and process measurement results

display(plot_histogram(statistics)) # Display measurement results as histogram

qc.draw(output="mpl", initial_state=True) # Draw final quantum circuit
```

### A.0.4 Operator Identification: Biased Game, Draft 2: Correlated Plays

```
from qiskit import QuantumRegister, QuantumCircuit
from qiskit_ibm_runtime import QiskitRuntimeService
from qiskit.quantum_info import random_unitary
from qiskit.extensions import UnitaryGate
from qiskit.primitives import Sampler
from qiskit.visualization import plot_histogram

service = QiskitRuntimeService()

circuit_list = []
```

```

for i in range(100): # Create and store 100 quantum circuits

    q = QuantumRegister(2, "q") # Construct a quantum circuit of three
        qubits and three bits
    c1 = ClassicalRegister(1,'c1')
    qc = QuantumCircuit(q, c1)

    qc.h(0) # Apply hadamard gate to first qubit

    qc.cx(q[0], q[1])

    rand = UnitaryGate(random_unitary(2))
    qc.append(rand, [2])  # Apply random 2x2 unitary matrix

    # qc.x([0]) Uncomment to make classical plays opposites of one another
    qc.append(rand, [1])
    qc.cx(q[0],q[1])

    qc.measure(1,0)

    circuit_list.append(qc)

sampler = Sampler() # Initialize Sampler primitive to run quantum circuits
results = sampler.run(circuit_list).result()

statistics = results.quasi_dists[0].binary_probabilities() # Extract and
    process measurement results

```

```
display(plot_histogram(statistics)) # Display measurement results as histogram

qc.draw(output="mpl", initial_state=True) # Draw final quantum circuit
```

### A.0.5 Operator Verification: Biased Game

```
from qiskit import QuantumRegister, QuantumCircuit, ClassicalRegister, display
from qiskit_ibm_runtime import QiskitRuntimeService
from qiskit.quantum_info.operators import Operator
from qiskit.primitives import Sampler
from qiskit.visualization import plot_histogram

service = QiskitRuntimeService() # Initialize IBM Quantum service

q = QuantumRegister(2, "q")
c1 = ClassicalRegister(1, 'c1')

qc = QuantumCircuit(q, c1) # Define a quantum circuit with two qubits and 1 bit

qc.h(0) # Apply Hadamard gate to first qubit to create superposition

qc.cx(q[0], q[1]) # Apply CNOT gate with qubit 0 as control and qubit 1 as target

op = Operator([ # Define 2*2 unitary operator solution
    [-0.18287237 - 6.99522349e-03j, -0.38743878 + 9.03548534e-01j],
    [ 0.35143341 + 9.18152122e-01j, -0.18300595 - 2.36252265e-04j]
```

```
])

# qc.x([0]) Uncomment to make classical plays opposites of one another

qc.append(op, [1]) # Apply solution operator to qubit 1

qc.cx(q[0], q[1]) # Apply another CNOT gate with qubit 0 as control and
    qubit 1 as target

qc.measure(1, 0)

sampler = Sampler() # Initialize Sampler primitive to execute quantum
    circuit

results = sampler.run(qc).result()
statistics = results.quasi_dists[0].binary_probabilities()

display(plot_histogram(statistics)) # Display measurement results as
    histogram

qc.draw(output="mpl") # Draw final quantum circuit
```

### A.0.6 Density Matrix Proof

Within quantum mechanics, density matrices may describe quantum states in scenarios involving mixed states and statistical ensembles [15]. In practice, a density matrix $\rho$ is defined as follows:

$$\rho = \sum_i p_i \left|\psi_i\right\rangle \left\langle\psi_i\right|,$$

where $|\psi_i\rangle$ are pure state vectors and $p_i$ are probabilities satisfying $\sum_i p_i = 1$.

Embracing this formalism, we define the current state of the penny on the *nth* stage of the PQ Penny Flip game as $\rho_n$. A single game is then composed of the initial state of the penny ($\rho_0$) and three subsequent plays ($\rho_1, \rho_2, \rho_3$). The beginning state of the penny is heads and is defined as $\rho_0 = |0\rangle\langle 0|$. The classical player is then limited to the following moves: flip ($F$), and do not flip ($N$). The quantum player has only one strategy, which we denote as $U$. The parameter $p$ is associated with the probability that the classical player flips the penny on their first move. Similarly, the parameter $q$ indicates the probability that the classical player flips the penny on their second turn. A single game is then defined as follows:

$$
\begin{aligned}
\rho_0 &= |0\rangle\langle 0| \\
\rho_1 &= pF\rho_0F^\dagger + (1-p)N\rho_0N^\dagger \\
\rho_2 &= U\rho_1U^\dagger \\
&= U\big(pF\rho_0F^\dagger + (1-p)N\rho_0N^\dagger\big)U^\dagger \\
\rho_3 &= qF\rho_2F^\dagger + (1-q)N\rho_2N^\dagger \\
&= qF\big(U\big(pF\rho_0F^\dagger + (1-p)N\rho_0N^\dagger\big)U^\dagger\big)F^\dagger \\
&\quad + (1-q)N\big(U\big(pF\rho_0F^\dagger + (1-p)N\rho_0N^\dagger\big)U^\dagger\big)N^\dagger
\end{aligned}
$$

We then let $F = \sigma_1$ and $N = \mathbb{I}$. The action of $\sigma_1$, or the Pauli-X matrix, corresponds to flipping the penny while $\mathbb{I}$, or the identity matrix, corresponds to the action of not flipping. We make use of the fact that $\sigma_1 = \sigma_1^\dagger$ and that $\mathbb{I} = \mathbb{I}^\dagger$ to simplify the final state of our penny in the following way:

$$
\begin{aligned}
\rho_3 &= qp\sigma_1U\sigma_1\rho_0\sigma_1U^\dagger\sigma_1 \\
&\quad + (1-p)q\sigma_1U\rho_0U^\dagger\sigma_1 \\
&\quad + (1-q)pU\sigma_1\rho_0\sigma_1U^\dagger \\
&\quad + (1-q)(1-p)U\rho_0U^\dagger
\end{aligned}
$$

### Quantum Victory

For a quantum victory, the final state of the penny must be heads, or, to put it explicitly, it must be the case that $\rho_3 = \rho_0$. To find a general solution $U$ that produces quantum victory, we must set each term in the equation for $\rho_3$ proportional to $\rho_0$. We then have the following four conditions:

$$1)\ \sigma_1 U|1\rangle\langle 1|U^\dagger \sigma_1 \propto |0\rangle\langle 0|$$

$$2)\ \sigma_1 U|0\rangle\langle 0|U^\dagger \sigma_1 \propto |0\rangle\langle 0|$$

$$3)\ U|1\rangle\langle 1|U^\dagger \propto |0\rangle\langle 0|$$

$$4)\ U|0\rangle\langle 0|U^\dagger \propto |0\rangle\langle 0|$$

To satisfy the conditions above, $U$ must preserve $\rho_0$ when acted upon by combinations of $U$ and $\sigma_1$. It must do this regardless of the values that $p$ and $q$ take. It is clear, however, that $\rho_0$ is not invariant under the following transformations: $U\rho_0 U^\dagger$, and $\sigma_1\rho_0\sigma_1$. Each of these alters the eigenbasis of $\rho_0$, which prevents a general solution $U$ from satisfying all four of the equations at once.

However, for certain combinations of $p$ and $q$, we are able to simplify $\rho_3$ and produce conditions where a solution $U$ is possible. There are four cases of interest.

### Case One

Let $p = q = 1$. As a result, $\rho_3$ reduces to the following:

$$\rho_3 = \sigma_1 U|1\rangle\langle 1|U^\dagger \sigma_1$$

As a result, we only need to satisfy condition 1). This is done by setting $U = \mathbb{I}$.

**Case Two**

Now, let $p = 0$ and $q = 1$, which reduces $\rho_3$ to the following:

$$\rho_3 = \sigma_1 U|0\rangle\langle 0|U^\dagger \sigma_1$$

Then, we set $U = \sigma_1$ to satisfy condition $2)$.

**Case Three**

Let $p = 1$ and $q = 0$. From this, $\rho_3$ reduces to the following:

$$\rho_3 = U|1\rangle\langle 1|U^\dagger$$

To satisfy condition $3)$, we set $U = \sigma_1$.

**Case Four**

Finally, let $p = q = 0$. As a result, $\rho_3$ reduces to the following:

$$\rho_3 = U|0\rangle\langle 0|U^\dagger$$

Now, we only need to satisfy condition $4)$. This is done easily by setting $U = \mathbb{I}$.

Notice that the solution $U$ is identical for cases one and three, where $U = \sigma_1$. Similarly, $U$ is the same for cases two and four, where $U = \mathbb{I}$. With this in mind, we see that if the quantum player knows that either or both $p$ and $q$ are 0 or 1, there exists a strategy $U$ that guarantees victory. Of course, the quantum player must have knowledge of their classical opponent's moves to apply any of these strategies. Consider, however, a similar scenario where the classical player is granted knowledge of their quantum opponent's strategy. This case is set in Meyer's original PQ Penny Flip game, where the classical player is given only one move. Applying the ideal quantum strategy devised by Meyer, the quantum player places the penny into a superposition of heads and tails. As a result, the penny is neither in the heads or tails state completely, which deprives the classical player of any advantage even when they know of the quantum player's strategy.

Although the quantum player can't always win the classically biased PQ penny flip game, they certainly possess a greater advantage with knowledge of their opponent's strategy. The classical player shares no such advantage in the game where they possess the same knowledge. As a result, we conclude that quantum strategies are superior to their classical counterpart.

### Generalized Results

These results shown above may be generalized to a game of an arbitrary number of odd moves. With this in mind, we justify the use of $p$ and $q$ as our probability parameters rather than $p1$, $p2$, $p3$,.... To make this point, we briefly consider a game of five plays. The final two plays of the game would be given as follows:

$$\begin{aligned}
\rho_4 &= U_2 \rho_3 U_2^\dagger \\
\rho_5 &= p_3 F \rho_4 F^\dagger + (1 - p_3) N \rho_4 N^\dagger \\
&= p_3 U_2 F \rho_4 F^\dagger U_2^\dagger + (1 - p_3) U_2 N \rho_4 N^\dagger U_2^\dagger
\end{aligned}$$

Applying our definition of $\rho_3$ above, we then have the following:

$$\begin{aligned}
\rho_5 = p_1 p_2 p_3 (F U_4 \sigma_1 U_2 \sigma_1 \rho_0 \sigma_1 U_2^\dagger \sigma_1 U_4^\dagger F^\dagger) \\
+ (1-p_1) p_2 p_3 (F U_4 \sigma_1 U_2 \rho_0 U_2^\dagger \sigma_1 U_4^\dagger F^\dagger) \\
+ (1-p_2) p_1 p_2 (F U_4 U_2 \sigma_1 \rho_0 \sigma_1 U_2^\dagger U_4^\dagger F^\dagger) \\
+ (1-p_2)(1-p_1) p_3 (F U_4 U_2 \rho_0 U_2^\dagger U_4^\dagger F^\dagger) \\
+ (1-p_3) p_1 p_2 (N U_4 \sigma_1 U_2 \sigma_1 \rho_0 \sigma_1 U_2^\dagger \sigma_1 U_4^\dagger N^\dagger) \\
+ (1-p_3)(1-p_1) p_2 (N U_4 \sigma_1 U_2 \rho_0 U_2^\dagger \sigma_1 U_4^\dagger N^\dagger) \\
+ (1-p_3)(1-p_2) p_1 (N U_4 U_2 \sigma_1 \rho_0 \sigma_1 U_2^\dagger U_4^\dagger N^\dagger) \\
+ (1-p_3)(1-p_2)(1-p_1)(N U_4 U_2 \rho_0 U_2^\dagger U_4^\dagger N^\dagger)
\end{aligned}$$

Once again, we then let $F = \sigma_1$ and $N = \mathbb{I}$. This gives the following:

$$\begin{aligned}
\rho_5 = p_1 p_2 p_3 (\sigma_1 U_4 \sigma_1 U_2 \sigma_1 \rho_0 \sigma_1 U_2^\dagger \sigma_1 U_4^\dagger \sigma_1) \\
+ (1-p_1) p_2 p_3 (\sigma_1 U_4 \sigma_1 U_2 \rho_0 U_2^\dagger \sigma_1 U_4^\dagger \sigma_1) \\
+ (1-p_2) p_1 p_2 (\sigma_1 U_4 U_2 \sigma_1 \rho_0 \sigma_1 U_2^\dagger U_4^\dagger \sigma_1) \\
+ (1-p_2)(1-p_1) p_3 (\sigma_1 U_4 U_2 \rho_0 U_2^\dagger U_4^\dagger \sigma_1) \\
+ (1-p_3) p_1 p_2 (U_4 \sigma_1 U_2 \sigma_1 \rho_0 \sigma_1 U_2^\dagger \sigma_1 U_4^\dagger) \\
+ (1-p_3)(1-p_1) p_2 (U_4 \sigma_1 U_2 \rho_0 U_2^\dagger \sigma_1 U_4^\dagger) \\
+ (1-p_3)(1-p_2) p_1 (U_4 U_2 \sigma_1 \rho_0 \sigma_1 U_2^\dagger U_4^\dagger) \\
+ (1-p_3)(1-p_2)(1-p_1)(U_4 U_2 \rho_0 U_2^\dagger U_4^\dagger)
\end{aligned}$$

For our case, we then let $U_4 = \mathbb{I}$. From this, we will find that $\rho_5 = \rho_3$. One may apply this to any multi-play game and set $U_{n>2} = \mathbb{I}$. This will reduce any game of an odd number of plays down to a game of three plays. We proceed by seeing how this applies in the five-play game. We have the following:

$$\begin{aligned}
\rho_5 = {} & p_1 p_2 p_3 (\sigma_1 \sigma_1 U_2 \sigma_1 \rho_0 \sigma_1 U_2^\dagger \sigma_1 \sigma_1) \\
& + (1-p_1) p_2 p_3 (\sigma_1 \sigma_1 U_2 \rho_0 U_2^\dagger \sigma_1 \sigma_1) \\
& + (1-p_2) p_1 p_2 (\sigma_1 U_2 \sigma_1 \rho_0 \sigma_1 U_2^\dagger \sigma_1) \\
& + (1-p_2)(1-p_1) p_3 (\sigma_1 U_2 \rho_0 U_2^\dagger \sigma_1) \\
& + (1-p_3) p_1 p_2 (\sigma_1 U_2 \sigma_1 \rho_0 \sigma_1 U_2^\dagger \sigma_1) \\
& + (1-p_3)(1-p_1) p_2 (\sigma_1 U_2 \rho_0 U_2^\dagger \sigma_1) \\
& + (1-p_3)(1-p_2) p_1 (U_2 \sigma_1 \rho_0 \sigma_1 U_2^\dagger) \\
& + (1-p_3)(1-p_2)(1-p_1)(U_2 \rho_0 U_2^\dagger)
\end{aligned}$$

Considering that $\sigma_1 \sigma_1 = \mathbb{I}$, we can simplify further.

$$\begin{aligned}
\rho_5 = {} & p_1 p_2 p_3 (U_2 \sigma_1 \rho_0 \sigma_1 U_2^\dagger) \\
& + (1-p_1) p_2 p_3 (U_2 \rho_0 U_2^\dagger) \\
& + (1-p_2) p_1 p_2 (\sigma_1 U_2 \sigma_1 \rho_0 \sigma_1 U_2^\dagger \sigma_1) \\
& + (1-p_2)(1-p_1) p_3 (\sigma_1 U_2 \rho_0 U_2^\dagger \sigma_1) \\
& + (1-p_3) p_1 p_2 (\sigma_1 U_2 \sigma_1 \rho_0 \sigma_1 U_2^\dagger \sigma_1) \\
& + (1-p_3)(1-p_1) p_2 (\sigma_1 U_2 \rho_0 U_2^\dagger \sigma_1) \\
& + (1-p_3)(1-p_2) p_1 (U_2 \sigma_1 \rho_0 \sigma_1 U_2^\dagger) \\
& + (1-p_3)(1-p_2)(1-p_1)(U_2 \rho_0 U_2^\dagger)
\end{aligned}$$

We are then able to collect terms.

$$\begin{aligned}\rho_5 &= (p_1p_2p_3 + ((1-p_3)(1-p_2)p_1)(U_2\sigma_1\rho_0\sigma_1U_2^\dagger) \\ &\quad + ((1-p_1)p_2p_3 + (1-p_3)(1-p_2)(1-p_1))(U_2\rho_0U_2^\dagger) \\ &\quad + ((1-p_2)p_1p_2 + (1-p_3)p_1p_2)(\sigma_1U_2\sigma_1\rho_0\sigma_1U_2^\dagger\sigma_1) \\ &\quad + (1-p_3)(1-p_1)p_2(\sigma_1U_2\rho_0U_2^\dagger\sigma_1)\end{aligned}$$

We then reorder the equation.

$$\begin{aligned}\rho_5 &= ((1-p_2)p_1p_2 + (1-p_3)p_1p_2)(\sigma_1U_2\sigma_1\rho_0\sigma_1U_2^\dagger\sigma_1) \\ &\quad + (1-p_2)(1-p_1)p_3(\sigma_1U_2\rho_0U_2^\dagger\sigma_1) \\ &\quad + (p_1p_2p_3 + ((1-p_3)(1-p_2)p_1)(U_2\sigma_1\rho_0\sigma_1U_2^\dagger) \\ &\quad + ((1-p_1)p_2p_3 + (1-p_3)(1-p_2)(1-p_1))(U_2\rho_0U_2^\dagger)\end{aligned}$$

Finally, let

$$\begin{aligned}p_ap_b &= ((1-p_2)p_1p_2 + (1-p_3)p_1p_2), \\ (1-p_a)p_b &= (1-p_2)(1-p_1)p_3, \\ (1-p_b)p_a &= (p_1p_2p_3 + ((1-p_3)(1-p_2)p_1), \\ (1-p_b)(1-p_a) &= ((1-p_1)p_2p_3 + (1-p_3)(1-p_2)(1-p_1)).\end{aligned}$$

With these definitions, we are able to simplify $p_5$ to the following:

$$\begin{aligned}\rho_5 = & \, p_a p_b \sigma_1 U \sigma_1 \rho_0 \sigma_1 U^\dagger \sigma_1 \\ & + (1-p_a) p_b \sigma_1 U \rho_0 U^\dagger \sigma_1 \\ & + (1-p_b) p_a U \sigma_1 \rho_0 \sigma_1 U^\dagger \\ & + (1-p_b)(1-p_a) U \rho_0 U^\dagger\end{aligned}$$

Assigning $p_a = p$ and $p_b = q$, we get the following:

$$\begin{aligned}\rho_5 = & \, qp \sigma_1 U \sigma_1 \rho_0 \sigma_1 U^\dagger \sigma_1 \\ & + (1-p) q \sigma_1 U \rho_0 U^\dagger \sigma_1 \\ & + (1-q) p U \sigma_1 \rho_0 \sigma_1 U^\dagger \\ & + (1-q)(1-p) U \rho_0 U^\dagger \\ & = \rho_3\end{aligned}$$

This technique may be applied to a game of an arbitrary number of odd plays. In short, $\rho_{(2n+1>3)} = \rho_3$ for $n \in \mathbb{N}$. For this reason, the proofs above only include the three-play game with probability parameters defined as $p$ and $q$. From this, it also follows that if the quantum player possesses knowledge that both or either $p$ or $q$ is equal to 0 or 1, for any game of an odd number of plays, they may always employ a strategy that guarantees victory.

It can just as easily be shown that games involving an even number of plays all reduce to games of two plays. In this case, however, the quantum player is unable to "pull" the penny out of its superposition after applying the Hadamaard matrix as their strategy. As a result, they are left with a 50% chance of victory. Still, it is clear that, when considering games of even and odd plays, the quantum player possesses an undeniable advantage over their classical opponent.

# Bibliography


[1] D. Meyer, “Quantum strategies,” Phys. Rev. Lett. **82,** 1052–1055 (1999).

[2] IBM, “IBM Quantum Documentation,” https://docs.quantum.ibm.com/ (Accessed April 2025).

[3] J. von Neumann, “Zur Theorie der Gesellschaftsspiele,” Math. Ann. **100,** 295–320 (1928), English translation: "On the Theory of Games of Strategy", in Selected Works of John von Neumann, edited by N. A. Taub, MIT Press, 1959, pp. 13-42.

[4] L. Samuelson, “Game theory in economics and beyond,” Journal of Economic Perspectives **30,** 107–30 (2016).

[5] N. McCarty and A. Meirowitz, *Political game theory: An introduction* (Cambridge University Press, 2014).

[6] E. Ho, A. Rajagopalan, A. Skvortsov, S. Arulampalam, and M. Piraveenan, “Game theory in defence applications: a review,” Sensors **22,** 1032 (2022).

[7] S. Gill and R. Buyya, “Transforming Research with Quantum Computing,” Journal of Economy and Technology (2024).

[8] P. Shor, “Polynomial-Time algorithms for prime factorization and discrete logarithms on a quantum computer,” SIAM Journal on Computing **26,** 1484–1509 (1997).

[9] L. Grover, "A fast quantum mechanical algorithm for database search," In *Proceedings, 28th Annual ACM Symposium on the Theory of Computing (STOC)*, pp. 212–219 (1996).

[10] J. Eisert, M. Wilkens, and M. Lewenstein, "Quantum games and quantum strategies," Phys. Rev. Lett. **83,** 3077–3080 (1999).

[11] K. Becker, "Quantum rock paper scissors," https://gist.github.com/primaryobjects/f58e5528e3b183b6bfcb8d7a9dc34934 (Accessed April 2025).

[12] N. Hebdon and J. Van Huele, "Quantum Game Theory and Genuine Quantum Advantage," Journal of the Utah Academy of Sciences, Arts, and Letters **101,** 318 (2024).

[13] P. Dirac, "A new notation for quantum mechanics," Mathematical Proceedings of the Cambridge Philosophical Society **35,** 416–418 (1939).

[14] R. Paley, "On orthogonal matrices," J. Math. Phys. **12,** 311–320 (1933).

[15] A. Patel, "The Quantum Density Matrix and Its Many Uses: From Quantum Structure to Quantum Chaos and Noisy Simulators," Journal of the Indian Institute of Science **103,** 401–417 (2023).