Quantum Game Theory – I*

A Comprehensive Study

Indranil Ghosh

Quantum computation has grown into a successful field of research during the last few decades. Parallelly the field of game theory has also evolved, resulting in the pursuit of quantum game theory. Works on this interdisciplinary field from early researchers like David A. Meyer, J. Eisert, M. Wilkens, A. Iqbal, E. Piotrowski, J. Orlin Grabbe, Adrian P. Flitney, and Derek Abbott are highly recommended. This article presents an introductory review of studies on understanding the workflow of quantum game-theoretic models along with their computer simulations. It starts with an introduction to game theory and quantum computation, followed by theoretical analyses of the classical and quantum versions of three game theory models—the penny flip game, prisoner's dilemma, and the two-person duel, supported by their simulation results. The simulations are carried out by writing Python codes that help us analyze the models. We will be able to understand the differences in the behaviors of both versions of the game models from the analyses.

Indranil Ghosh is a graduate student doing post graduation in physics specialising in condensed matter physics, from the Department of Physics, Jadavpur University, Kolkata. His research interests include computational physics, numerical computing, quantum mechanics and quantum computing.

1. Introduction

Quantum game theory is the extension of the classical game theory to the quantum domain, where the participating players make use of quantum manipulations, and apply quantum strategies instead of classical moves. One of the earliest studies on considering game theory models from the viewpoint of quantum algorithms was commenced by David A. Meyer [1] in 1999 from the University of California, San Diego. His work was based on a

Keywords
Quantum game theory, quantum algorithms, quantum computing, strategy dominance, Pareto efficiency, sequential games, Nash equilibrium.

*Vol.26, No.5, DOI: https://doi.org/10.1007/s12045-021-1168-2

quantum version of the penny flip game, where he showed that a player applying quantum strategies always beats a player applying classical ones. Next, Eisert and Wilkens [2] made a detailed investigation on implementing quantum operations on two-player binary choice games like prisoner's dilemma. Also, J. Orlin Grabbe's essay [3], generally oriented towards economists with a little or no background on quantum mechanics, made a comprehensive survey of these famous game models with applications of quantum operators.

Quantum game theory has found intriguing applications in several fields like population biology and market economics.

Quantum game theory has found intriguing applications in several fields like population biology and market economics. A study on quantum evolutionary stable strategies (QESS) was carried out by A. Iqbal and A. H. Toor [4], where they applied quantum game theory concepts to the original work on ESS by J. Maynard Smith and G. R. Price [5]. Games of survival that are played at the molecular level can be modeled using QESS. Another application of the quantum game theory was on market games. A series of articles was published by E. W. Piotrowski and J. Sladkowski beginning with *quantum market games* [6]. They worked on quantum bargaining games, quantum English auctions, etc. These revolutionized their idea of *quantum anthropic principle* [7] for the evolution of markets being governed by quantum laws instead of classical laws.

Keeping the above information in mind, this article thus presents a systematic study on the quantum game-theoretic models starting with introducing the classical game theory and quantum computing, followed by the development of three quantum games. A simulator is designed with Python version 3.6.8 [8] [9], and the structures and issues of these games are simulated with the same to understand their basic frameworks.

2. Classical Game Theory

The rigorous mathematical treatment of game theory began in 1944 with the seminal work *Theory of Games and Economic Behavior* by John von Neumann and Oskar Morgenstern [10]. Later, John Nash, who received a Nobel Prize in Economics, revolutionised the field with his influential works [11] [12].

Game theory is the study of mechanism to represent and analyse strategic interactions among rational agents called *players* whose joint decisions ultimately results in a final outcome. These *players* change their roles according to the problem being handled. Game theory can be bifurcated into two main branches:

Game theory is the study of mechanism to represent and analyse strategic interactions among rational agents called *players* whose joint decisions ultimately results in a final outcome.

- *Cooperative game theory*, where players are allowed to communicate with each other and exchange knowledge.
- *Non-cooperative game theory*, where players are unable to communicate with each other or cannot share information or sign contracts.

2.1 *Game-frames for Modeling Games*

Giacomo Bonanno [13], professor of economics from the University of California describes a *game-frame* as, a list of five items i.e, $G = \{P, \{S_i\}, O, F, \{\succsim_i\}\}$, where,

1. $P = \{1, 2, ..., n\}$ is a set of players and $n \geq 2$.
2. $\{S_i\}$ is a set of sets, for each player $i \in P$. Player i has access to the set of strategies S_i. Now, the set of *strategy profiles* is defined as, $S = S_1 \times S_2 \times ... \times S_n$, where S is the Cartesian product of these sets. As a result, an element of S is a list, $s = \{s_1, s_2, ..., s_n\}$ which represents the strategy of each player.
3. O is a set of outcomes.
4. $F : S \rightarrow O$ is function that maps with every strategy profile s an outcome $F(s) \in O$.
5. $\{\succsim_i\}$ is a set of complete and transitive ranking (similar to 'ordering' in set theory) of the set of outcomes O, each for every player $i \in P$. Here 'complete' means the complete information of the players' strategies are denoted by the ranking. Now, an ordinal utility function $U : O \rightarrow \mathbb{R}$ is a mechanism for quantizing the ranking (ordering) $\geq$ between outcomes. Here, 'quantizing' means replacing the qualitative ranking with numeric ranking,

Table 1. Reduced form of a game-frame.

$P1/P2$	S_1	S_2
S_1	(a, a)	(b, c)
S_2	(c, b)	(d, d)

i.e, if for every two outcomes, $o_1 \in O$ and $o_2 \in O$, $o_1 \succ o_2$, then $U(o_1) > U(o_2)$ and if $o_1 \sim o_2$, then $U(o_1) = U(o_2)$. It means, if the number assigned to o_1 is greater than the number assigned to o_2, then o_1 is preferred to o_2. Now, a payoff function of the player i, $\pi_i : S \to \mathbb{R}$ is defined by $\pi_i(s) = U_i(f(s))$.

These fetch us a reduced-form of the game, defined by a list of three elements, i.e, $G = \{P, \{S_i\}, \{\pi_i\}\}$. We will be considering the reduced-form of the game models in this article given by the following tabular structure:

Here, $P1$ and $P2$ are the two players involved and the strategies available are $\{S_1, S_2\}$. The corresponding payoff values are {a, b, c, d}. The first element in the parenthesis corresponds to the payoff of $P1$ and the second element of $P2$.

2.2 *Strategy Dominance*

Now, as strategies are concerned, for a player, there are two relations on the set of its strategies.

1. *Strictly Dominant Strategy*: For a player p in a game, let us assume, there are two strategies, s_{1p} and s_{2p}. Now, s_{1p} strictly dominates s_{2p} if and only if for every choice of strategies of the other players, $p's$ payoff from choosing s_{1p} is strictly greater that $p's$ payoff from choosing s_{2p}. So, s_{1p} is a dominant strategy. For all other players q who has access to strategies s_q, the notation s_{-p} can be used to mean the same, i.e,

$$s_{-p} = \{s_1, ..., s_{p-1}, s_{p+1}, ..., s_n\}. \quad (1)$$

So, the strict dominance of s_{1p} over s_{2p} is maintained if and only if,

$$\pi_p(s_{1p}, s_{-p}) > \pi_p(s_{2p}, s_{-p}). \quad (2)$$

2. *Weakly Dominant Strategy* s_{1p} is weakly dominant over s_{2p} if, for every choice of strategies of the other players, $p's$ payoff from choosing s_{1p} is at least as great as $p's$ payoff from choosing s_{2p}. So, the weak dominance of s_{1p} over s_{2p} is maintained if and only if,

$$\pi_p(s_{1p}, s_{-p}) \geq \pi_p(s_{2p}, s_{-p}). \quad (3)$$

2.3 *Pareto Efficiency*

For a reduced-form of a game, if s_1 and s_2 are two strategy profiles, then

- s_1 is strictly Pareto superior to s_2 if for every player p, $\pi_p(s_1) > \pi_p(s_2)$.
- s_1 is weakly Pareto superior to s_2 if for every player p, $\pi_p(s_1) \geq \pi_p(s_2)$ together with a player q such that $\pi_q(s_1) > \pi_q(s_2)$.
- A strategy s_q of a player q is Pareto optimal if it cannot be improved by hurting all the other players.

2.4 *Iterated Deletion of Strictly Dominated Strategies Algorithm*

A rational player in a game will always play a strictly dominated strategy, if available. Rationality of a player refers to the fact that, the player will always tend to choose a strategy that will maximize his/her utility. It also informs us which strategies are never played. The *iterated deletion of strictly dominated strategies* (IDSDS) algorithm is built on the common knowledge of rationality. Steven Tadelis [14] gives a detailed description of the algorithm (see *Algorithm* 1)

Rationality of a player refers to the fact that, the player will always tend to choose a strategy that will maximize his/her utility. It also informs us which strategies are never played.

Now, If G^∞ (the final game form) consists of a single strategy profile, that profile is called the *iterated strict dominant strategy equilibrium*. But G^∞ may consist of two or more strategy profiles. We consider an example, where G^0 (the initial game form) is defined as the one given in *Table* 2.

Applying the IDSDS algorithm on G^0, the final form of the game G^∞ that is generated is given by *Table* 3.

Algorithm 1: IDSDS algorithm

Input: Initial reduced-game form G^0

Output: Final game form G^∞

1 The original strategy of player p in the game is set by defining $S_p^0 = S_p$ and $i = 0$

2 Check whether there are players whose strategies $s_p \in S_p^i$ are strictly dominated. If the condition is `True` go to step 3, if not go to step 4.

3 $\forall i \in n$, remove those strategies $s_p \in S_p^i$ that are strictly dominated. Set $i = i + 1$, and define a new game with strategy set S_p^i without the removed strictly dominated strategies. Go back to step 2.

4 The remaining strategies in S_p^i are the solutions of the game and the final form is G^∞.

Table 2. G^0 for IDSDS algorithm.

$P1/P2$	S_1	S_2	S_3
S_1	(8, 6)	(0, 9)	(3, 8)
S_2	(3, 2)	(2, 1)	(4, 3)
S_3	(2, 8)	(1, 5)	(3, 1)

Table 3. G^∞ for IDSDS algorithm.

$P1/P2$	S_3
S_2	(4, 3)

(S_2, S_3) is the *iterated strict dominant strategy equilibrium* that survives the algorithm and is compatible with common belief of rationality.

According to game theorists, an 'equilibriated' game means a game that has reached its stable state, where all the causal strategies internal to the game balance each and every outcome out and leave the game state to be in 'rest'.

2.5 *Nash Equilibrium*

If the IDSDS algorithm is unable to solve a game, Nash equilibrium, named after John Nash, offers an alternative. Here, 'solving a game' means finding the equilibria of a game. According to game theorists, an 'equilibriated' game means a game that has reached its stable state, where all the causal strategies internal to the game balance each and every outcome out and leave the

$P1/P2$	S_1	S_2	S_3	S_4
S_1	(4, 0)	(3, 2)	(2, 3)	(4, 8)
S_2	(4, 2)	(2, 1)	(1, 2)	(0, 2)
S_3	(3, 6)	(5, 5)	(3, 1)	(5, 0)
S_4	(2, 3)	(3, 2)	(1, 2)	(3, 3)

Table 4. A reduced game frame.

game state to be in 'rest'. For two players, a strategy profile $s^* = (s_1^*, s_2^*) \in S_1 \times S_2$ is a Nash equilibrium, if the two following conditions are satisfied,

- For every $s_1 \in S_1$, $\pi_1(s_1^*, s_2^*) \geq \pi_1(s_1, s_2^*)$.
- For every $s_2 \in S_2$, $\pi_2(s_1^*, s_2^*) \geq \pi_1(s_1^*, s_2)$.

For example, the game represented by *Table* 4 has a unique Nash equilibrium $(S_2, S_1) \equiv (4, 2)$.

2.6 *Sequential Games*

Besides being simultaneous, interactions between players can be sequential too. These games are also called *dynamic games* or *games in extensive form*. Dynamic games have a sequence of moves by the players, one after another. The knowledge of the full history of these moves to all the players corresponds to perfect information, and partial or no knowledge of the full history of the moves to the players correspond to imperfect information. Chess is a sequential game with perfect information, whereas the penny flip game and the two-person duel are sequential games with imperfect information.

Dynamic games have a sequence of moves by the players, one after another. The knowledge of the full history of these moves to all the players corresponds to perfect information, and partial or no knowledge of the full history of the moves to the players correspond to imperfect information. Chess is a sequential game with perfect information, whereas the penny flip game and the two-person duel are sequential games with imperfect information.

These games can be modeled by *rooted directed trees* consisting of a set of nodes and directed edges joining them. A dynamic game is defined by a list of 7 elements, i.e, $G = \{T, P, f_P, S, f_S, O, f_O\}$, where,

1. T is a finite rooted directed tree.
2. $P = \{1, ..., n\}$ is a set of players and $n \geq 2$.

3. A function f_P that assigns a single player to every decision tree.

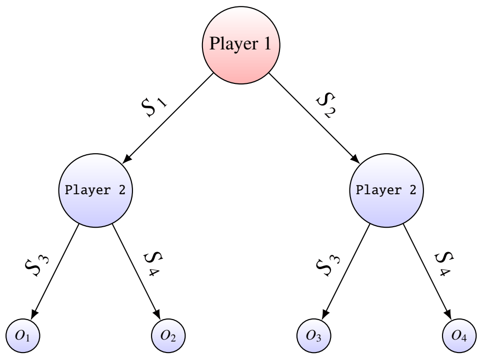
4. A set of strategies S.

5. A function f_S that assigns a strategy to every directed edge of the tree, with the constraint that no two edges out of the same node are assigned the same strategy application.

6. A set of outcomes O.

7. A function that assigns an outcome to every terminal node.

A simple structure of this kind of game is:



Here, The game starts with Player 1, who has strategies S_1 or S_2. Player 1 plays one of its strategies followed by the move of Player 2, where it plays one of its strategies from $S3$ and S_4. O_i are different outcomes of the game that are analysed by computing the corresponding payoffs for each player and placing them on reduced game-frames. In the next section, we give a brief introduction to quantum computation.

3. Quantum Computation (QC)

The process of manipulating quantum systems, like superconducting qubits, in order to process information is referred to as *quantum computation* [15] [16]. During the 1980s, renowned physicists like Paul Benioff [17] and Richard Feynman [18] made foundational contributions to this new field. A quantum mechanical model of the Turing machine was proposed by Paul Benioff, and the potential of a quantum computing[1] device to simulate things that a classical computer could not was pointed out by Richard Feynman. In 1992, David Deutsch and Richard Jozsa [19] proposed a pioneering deterministic quantum algorithm, which is exponentially faster than any possible deterministic classical algorithm. Later, drawing inspiration from this work, in 1994, Peter Shor [20] developed a quantum algorithm for factoring integers, which was one of the beginnings of practical implementations of quantum computing. Besides, the development of Grover's search algorithm [21] was another notable breakthrough in quantum computation. In recent times, Seth Lloyd's works [22] [23] have been influential in this field. Although there are several models in quantum computing, like quantum circuit model, quantum Turing machine, adiabatic quantum computer [24], and quantum cellular automata [25], we will be dealing with the quantum circuit model in this article to study different quantum game-theoretic models.

[1] Apoorva Patel, Quantum Computation: Particle and Wave Aspects of Algorithms, *Resonance*, Vol.16, No.9, pp:821–835, 2011.

In this computational model, the state space is a two dimensional Hilbert space $\mathbf{H}^2$. Unlike classical computation, where the basic unit of computation is a bit, in QC, the basic unit is a qubit, represented by the basis states, $\{|0\rangle, |1\rangle\}$ and a quantum state is represented as,

$$|\psi\rangle = a|0\rangle + b|1\rangle \tag{4}$$

where $a, b \in \mathbb{C}$ and $|0\rangle, |1\rangle \in \mathbf{H^2}$. Now, a and b are the probability amplitudes of the basis states, where, $|a|^2$ and $|b|^2$ represent the probabilities of finding the final state in either of the basis states after measurement. This results in the fact that, $|a|^2 + |b|^2 = 1$. The qubits in its pure states, i.e, $|0\rangle$ and $|1\rangle$ are represented geometrically by the Bloch sphere.

This notation of *bra-ket* was introduced by Paul Dirac [26] and the basis states are generally represented as vectors in the complex Hilbert space. In a computer we can simulate the basis states as, $|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}$ and $|1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$. And a quantum state, $|\psi\rangle = \begin{bmatrix}a\\b\end{bmatrix} = a\begin{bmatrix}1\\0\end{bmatrix} + b\begin{bmatrix}0\\1\end{bmatrix}$. This helps us writing simple simulators through a

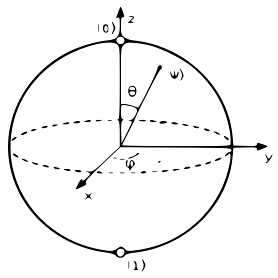


Figure 1. Bloch sphere. (Source: *Wikipedia*)

programming language for carrying out simple quantum computing simulations in our classical computers.

In the quantum circuit model, a quantum algorithm can be written down as,

Algorithm 2: A quantum algorithm

Input: An initial quantum state $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$ is prepared

Output: The final quantum state $|\phi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ obtained

1 Apply a series of combinations of quantum logic gates (mathematically represented by local unitary operators), U on the initial state, $|\psi\rangle$.

2 Measure the circuits in the computational eigenbasis to obtain the final state $|\phi\rangle$, where α^2 and β^2 are computed to analyse further. The whole operation is represented by, $|\phi\rangle = U\,|\psi\rangle$.

Both $|\psi\rangle$ and $|\phi\rangle$ are linear superpositions of the basis eigenstates $\{|0\rangle, |1\rangle\}$. That the squares of the probabilities add to 1 means that the quantum gates are always unitaries. One of the major characteristics of quantum computation is entanglement [27] [28], where measurements on some qubits will affect other qubits. It is a physical phenomenon between a pair of particles where the quantum state of each particle, even though are separated by an infinitely large distance, can not be described independently of that of the others. Entanglement produces a quantum association between the initial state and the final measured state. It has no classical analogue and is a complete quantum effect. Also, superposition, i.e, the ability of a quantum observable to be in a linear combination of one or more basis eigenstates, is another property that is used by classical computers for storing exponentially more data than their classical counterparts. A classical computer with n bits stores one of the 2^n possible values, whereas, a quantum computer stores all the 2^n values.

Schrodinger's equation [29] governs the evolution of a quantum computer over time,

$$i\hbar\frac{\partial|\psi(t)\rangle}{\partial t} = H(t)\,|\psi(t)\rangle \tag{5}$$

Here, $i = \sqrt{-1}$, $\hbar = \frac{h}{2\pi}$, $|\psi(t)\rangle$ is the state vector that illustrates a quantum state at a time t and $H(t)$ is the Hamiltonian operator. Assuming that the solution of the above equation is $U(t)$, i.e, the sequence of combination of complex unitaries, the time evolution of the given state is given by,

$$|\psi(t)\rangle = U(t)\,|\psi(0)\rangle\,. \tag{6}$$

As $U(t)$ is reversible, the number of input qubits equals the number of output qubits. Some of the quantum gates that we would require in our analyses are,

- Hadamard: $H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$
- Identity: $I = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$
- Pauli X: $\sigma_x = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$
- Pauli Y: $\sigma_y = \begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}$
- Pauli Z: $\sigma_z = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$
- T($\frac{\pi}{8}$): $T = \begin{bmatrix}1 & 0\\ 0 & e^{i\frac{\pi}{4}}\end{bmatrix}$

- CNOT: $C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

- SWAP: $SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

- Other single qubit or multi-qubit gates like S-gate and Toffoli gate.

The Pauli X gate is the quantum analogue of the classical NOT gate. The controlled-NOT (CNOT) gate flips the target bit if and only if the input bit is 1. The output of this gate is analogous to the output of the classical XOR gate. The {NAND, NOR} forms the universal gate set for classical computations, and the {H, T, CNOT} forms the universal gate set for quantum computations. With these information in mind, we will be able to start analysing the quantum game theory models in the upcoming sections.

Suggested Reading

[1] David A. Meyer, Quantum strategies, *Phys. Rev. Lett.*, Vol.82, pp.1052–1055, Feb 1999.

[2] Jens Eisert and Martin Wilkens, Quantum games, *Journal of Modern Optics*, Vol.47, 14-15, pp.2543–2556, 2000.

[3] J Orlin Grabbe, An introduction to quantum game theory, *arXiv preprint quant-ph/0506219*, 2005.

[4] A Iqbal and A H Toor, Evolutionarily stable strategies in quantum games, *Physics Letters A*, Vol.280, Nos.5-6, pp.249–256, 2001.

[5] J Maynard Smith and George R Price, The logic of animal conflict, *Nature*, Vol.246, No.5427, pp.15–18, 1973.

[6] Edward W Piotrowski and J Sładkowski, Quantum market games, *Physica A: Statistical Mechanics and its Applications*, Vol.312, Nos.1–2, pp.208–216, 2002.

[7] Edward W Piotrowski and Jan Sładkowski, Quantum-like approach to financial risk: Quantum anthropic principle, *arXiv preprint quant-ph/0110046*, 2001.

[8] Indranil Ghosh, indrag49/quantum-game-theory: Python program written to run simulations on quantum game theory models, https://github.com/indrag49/Quantum-Game-Theory, 2020. (Accessed on 05/02/2020).

[9] Indranil Ghosh, *indrag49/quantum-game-theory: Python program written to run simulations on quantum game theory models*, https://github.com/indrag49/Quantum-Game-Theory, 2020. (Accessed on 05/02/2020).

[10] John Von Neumann and Oscar Morgenstern, *Theory of Games and Economic Behavior*, Princeton University Press, 1944.

[11] John F Nash *et al.*, Equilibrium points in n-person games, *Proceedings of the National Academy of Sciences*, Vol.36, No.1, pp.48–49, 1950.

[12] John Nash, Non-cooperative games, *Annals of Mathematics*, pp.286–295, 1951.

[13] Giacomo Bonanno, *Game Theory* (open access textbook with 165 solved exercises), 2015.

[14] Steven Tadelis, *Game Theory: An Introduction*, Number 10001 in Economics Books, Princeton University Press, 2012.

[15] Michael A Nielsen and Isaac L Chuang, Quantum computation and quantum information, *Phys. Today*, Vol.54, 60–2, 2001.

[16] N David Mermin, *Quantum Computer Science: An Introduction*, Cambridge University Press, 2007.

[17] Paul Benioff, The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by Turing machines, *Journal of Statistical Physics*, Vol.22, No.5, pp.563–591, 1980.

[18] Richard P Feynman, Simulating physics with computers, *Int. J. Theor. Phys*., Vol.21, (6/7), 1999.

[19] D Deutsch and R Jozsa, Rapid solutions of problems by quantum parallelism, *Proc.r. In Soc. Lond. A*., Vol.439, pp.553–558, 1992.

[20] Peter W Shor, Algorithms for quantum computation: Discrete logarithms and factoring, *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pp.124–134, Ieee, 1994.

[21] Lov K Grover, A fast quantum mechanical algorithm for database search, *In Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, pp.212–219, 1996.

[22] Aram W Harrow, Avinatan Hassidim and Seth Lloyd, Quantum algorithm for linear systems of equations, *Physical Review Letters*, Vol.103, No.15, p.150502, 2009.

[23] Seth Lloyd and Samuel L Braunstein, Quantum computation over continuous variables, *Quantum Information with Continuous Variables*, pp.9–17, Springer, 1999.

[24] Edward Farhi, Jeffrey Goldstone, Sam Gutmann and Michael Sipser, Quantum computation by adiabatic evolution, *arXiv preprint quant-ph/0001106*, 2000.

[25] John Watrous, On one-dimensional quantum cellular automata, *In Proceedings of IEEE 36th Annual Foundations of Computer Science*, pp.528–537, IEEE, 1995.

[26] Paul Adrien Maurice Dirac, A new notation for quantum mechanics, *In Mathematical Proceedings of the Cambridge Philosophical Society*, Vol.35, pp.416–418, Cambridge University Press, 1939.

[27] Albert Einstein, Boris Podolsky and Nathan Rosen, Can quantum-mechanical description of physical reality be considered complete? *Physical Review*, Vol.47,

Address for Correspondence
Indranil Ghosh
Sib Bari, Ward No.-18
Jalpaiguri
West Bengal 735 101, India.
Email:
indranilg49@gmail.com

No.10, 777, 1935.

[28] Erwin Schrödinger, Discussion of probability relations between separated systems, *In Mathematical Proceedings of the Cambridge Philosophical Society*, Vol.31, pp.555–563, Cambridge University Press, 1935.

[29] Jun John Sakurai and Eugene D Commins, *Modern Quantum Mechanics*, revised edition, 1995.