LECTURE: 7

QUANTUM CIRCUITS

PART ONE

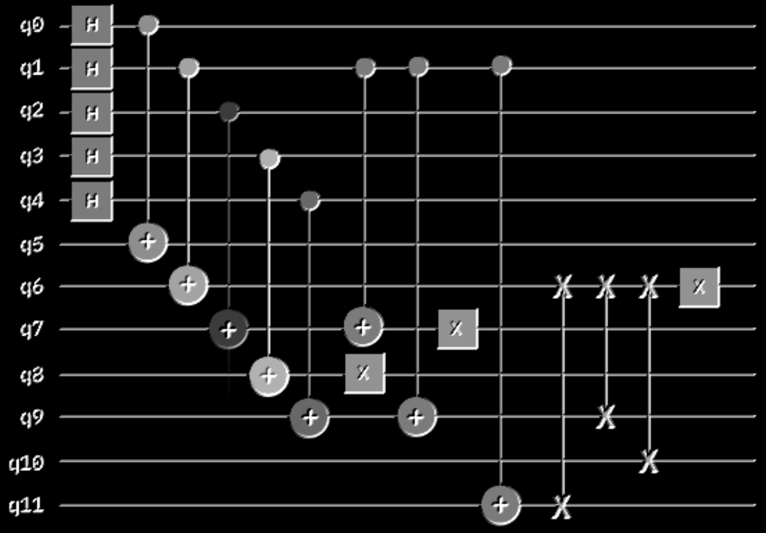
quantum circuits: outline

- today part one:
  - intro to computational QM
  - Quantum Prisoner's Dilemma

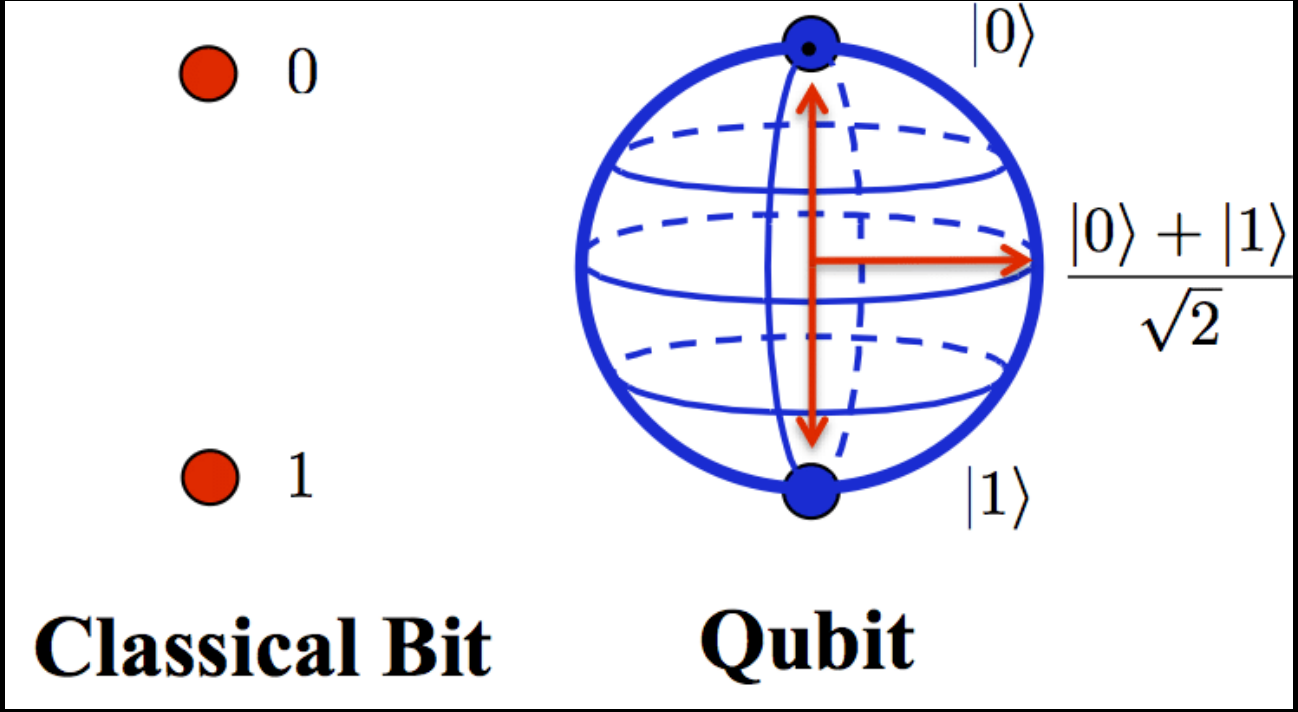
based on
Quantum Games and Quantum Strategies
Jens Eisert, Martin Wilkens, and Maciej Lewenstein
Phys. Rev. Lett. **83**, 3077

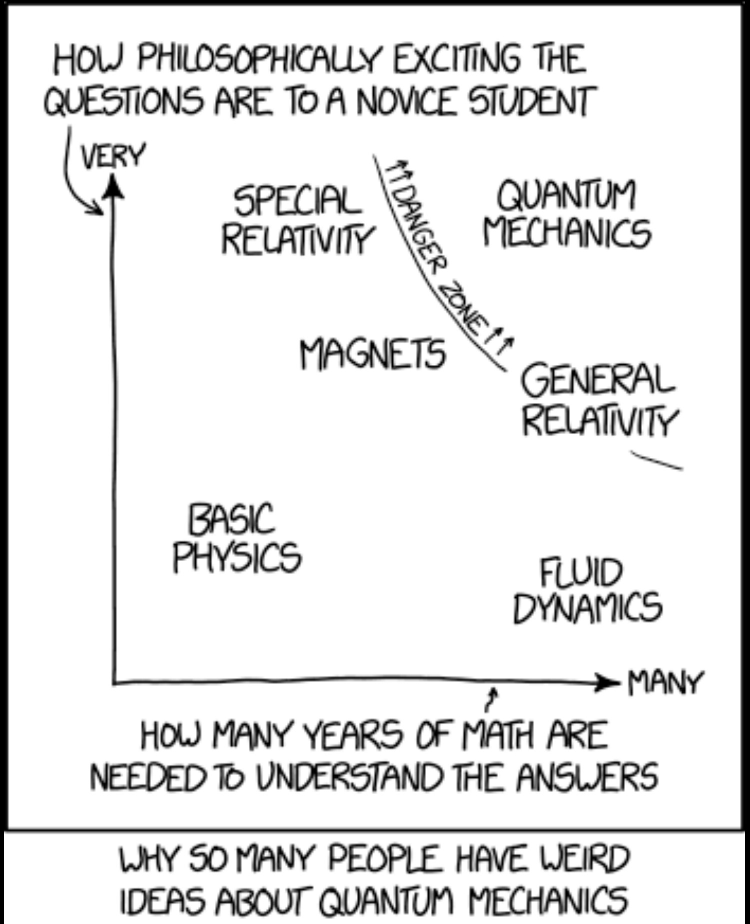
- next week part two:
  - intro to Qiskit
  - quantum computing
  - quantum computing IRL

Qiskit
Elements for building a quantum future



quantum mechanics





Qubit – basic unit of quantum information.
Superposition – being in „more than one state”

qubit

- the state of a qubit lives in a two-dimensional (Hilbert) vector space

$$\Psi = \alpha \,|0\rangle + \beta \,|1\rangle \qquad |\alpha|^2 + |\beta|^2 = 1$$

with basis vectors

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \qquad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$$

measurement

initial state

$$\Psi = \alpha \left|0\right\rangle + \beta \left|1\right\rangle$$

outcome probability

$$\left|\alpha\right|^2 \qquad \left|\beta\right|^2$$

system state after measurement

$$\left|0\right\rangle \qquad \left|1\right\rangle$$

two qubits

- the total state of two qubits lives in a tensor product of two qubit spaces spanned by basis vectors

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad |01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \qquad |11\rangle = |1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\Psi = \alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \delta\,|11\rangle \qquad |\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$$

entanglement

Arises for **two** quantum systems, A and B. In general, their total state is of the form

$$|\psi\rangle_{AB} = \sum_{i,j} c_{i,j} |i\rangle_A \otimes |j\rangle_B$$

$|i\rangle_A$, $|j\rangle_B$ - basis vectors of each system

If one can separate $|\psi\rangle_{AB}$ into $|\psi\rangle_{AB} = |\psi\rangle_A \otimes |\phi\rangle_B$ with $|\psi\rangle_A = \sum_i c_i^A |i\rangle_A$, $|\phi\rangle_B = \sum_j c_j^B |j\rangle_B$, the state is called a product state. Otherwise, it is **entangled**. This entails that it cannot be understood as two separate systems, but the full description of the composite system must be given.

entanglement

Consider an entangled state $$|\psi\rangle_{AB} = \frac{1}{\sqrt{2}}\left(|0\rangle_A \otimes |1\rangle_B - |1\rangle_A \otimes |0\rangle_B\right)$$

Now suppose an observer in system A measures her qubit and obtains $|0\rangle_A$. The composite system collapses then to the state $|0\rangle_A \otimes |1\rangle_B$, so an observer in system B measuring his state in the same basis will always measure $|1\rangle_B$.

For entangled states, the measurement results are **correlated**!

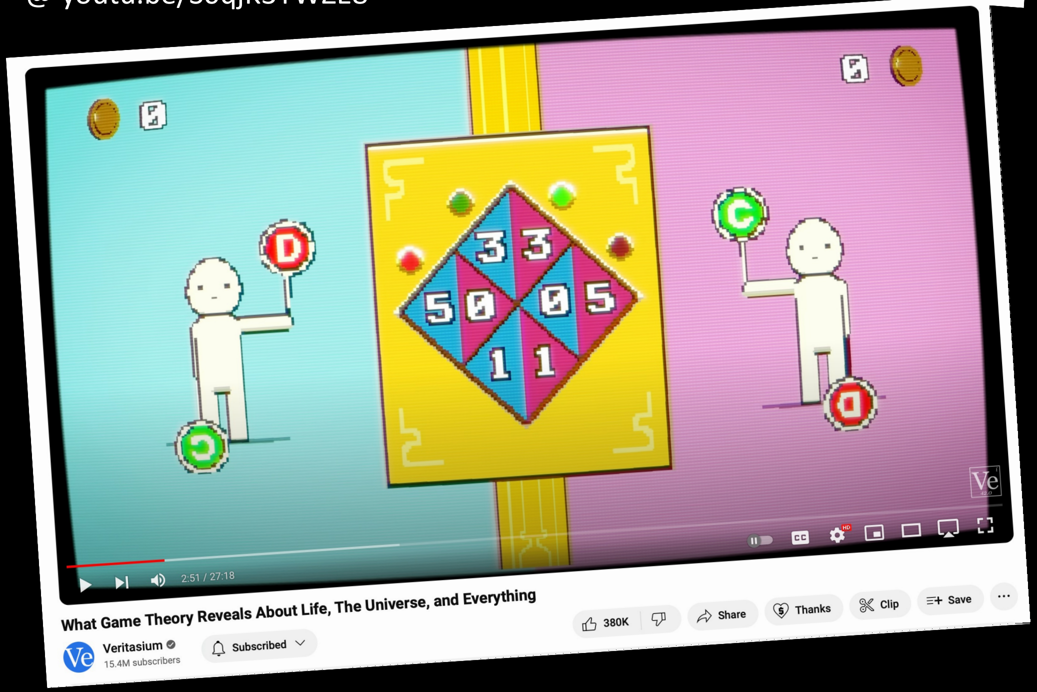
prisoner's dilemma recap

		Bob	
		cooperate	defect
Alice	cooperate	(3, 3) reward for mutual cooperation	(0, 5) sucker's payoff and temptation to defect
	defect	(5, 0) temptation to defect and sucker's payoff	(1, 1) punishment for mutual defection

@ Lecture 2: iterated PD

Nash equilibrium (not Pareto efficient)

@ youtu.be/S0qjK3TWZE8



@ youtu.be/mScpHTIi-kM

quantum prisoner's dilemma

Bonnie and Clyde now encounter their dilemma in a quantum setup. They both start in qubit state $|C\rangle$ and can decide what unitary operation to perform on their state. Then a referee measures their composite states in the basis $\{|CC\rangle, |CD\rangle, |DC\rangle, |DD\rangle\}$. Their expected payoff relies on the probability of measuring each outcome in the final state $|\psi_f\rangle$.

basis vectors

$$|C\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$ - cooperate

$$|D\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$ - defect

expected payoffs

$$\$_A = r\,P_{CC} + s\,P_{CD} + t\,P_{DC} + p\,P_{DD}$$

$$\$_B = r\,P_{CC} + t\,P_{CD} + s\,P_{DC} + p\,P_{DD}$$

$$r = 3, s = 0, t = 5, p = 1$$

Alice \ Bob	cooperate	defect
cooperate	(3, 3) reward for mutual cooperation	(0, 5) sucker's payoff and temptation to defect
defect	(5, 0) temptation to defect and sucker's payoff	(1, 1) punishment for mutual defection

quantum prisoner's dilemma

In the basic scheme, Alice and Bob use unitaries corresponding to classical decisions:

$$\hat{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \hat{D} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

cooperate ($|C\rangle$ unchanged) defect ($|C\rangle$ flipped)

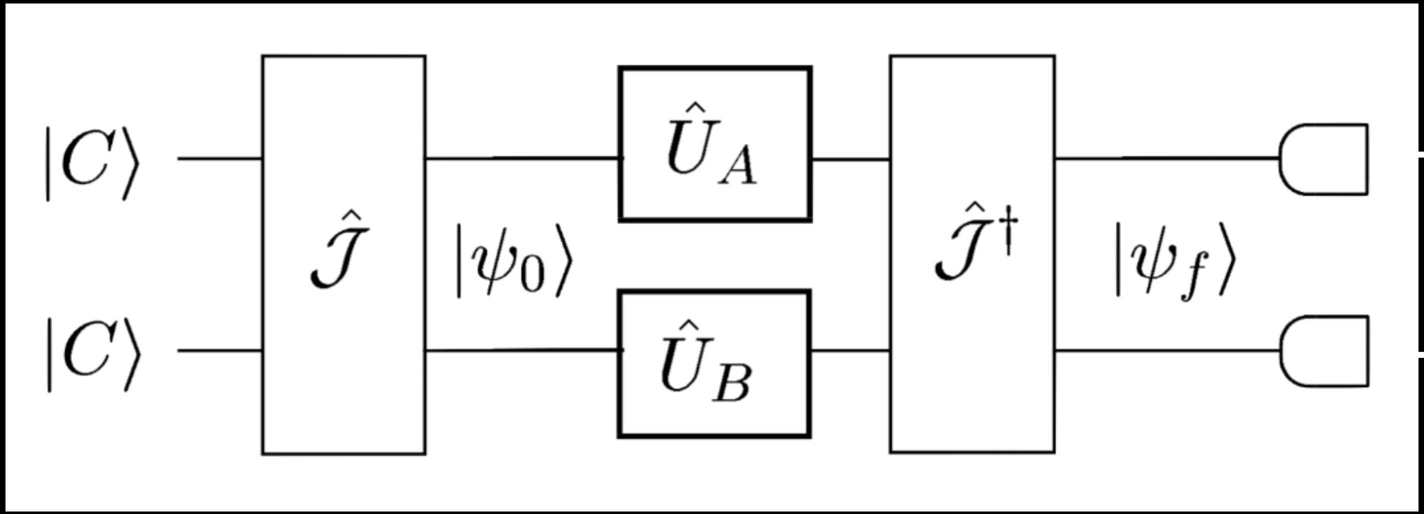
The classical PD is reproduced: $\hat{D} \otimes \hat{D}$ is the Nash equilibrium.

It gets interesting when the initial state is entangled!
For large values of entanglement, a Nash equilibrium
is obtained with a Pareto-efficient strategy $\hat{Q}$:

$$\hat{Q} = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}$$

the quantum circuit

measurement in
$\{|C\rangle, |D\rangle\}$



Alice's register

Bob's register

entangling operation

$$\hat{\mathcal{J}} = \exp(-i\,\gamma\,\hat{D} \otimes \hat{D}\,/\,2)$$

$0 \leq \gamma \leq \frac{\pi}{2}$ – degree of entanglement

unitary parametrization

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

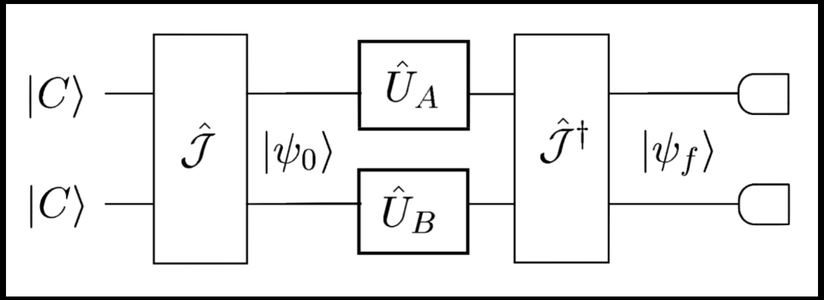
$\hat{C} = \hat{U}(0,0),\ \hat{D} = \hat{U}(\pi, 0)$ $\quad$ $\hat{U}(\theta, 0)$ - classical strategies

task 1: getting acquainted with computational QM (0.3 pts)

„Bonnie and Clyde have recently gotten into an argument about the philosophy of picking the correct side of a coin flip. Bonnie was raised by the moto ”Tails Never Fails”, while Clyde was taught ”Heads Has Best Creds”. Their friend suggests that they solve their disagreement with a series of coin flips, but Bonnie doesn’t trust any coin that Clyde owns, and vice versa for Clyde. Thus, they agree to use a qubit as their coin.”

1a) Let the user input a real value α using Python’s input() function. Take the state $|0\rangle$ and apply to it the unitary $\hat{U} = \begin{pmatrix} \cos\alpha/2 & \sin\alpha/2 \\ -\sin\alpha/2 & \cos\alpha/2 \end{pmatrix}$, which should result in a superposition $|\psi\rangle = \cos\alpha/2\,|0\rangle + \sin\alpha/2\,|1\rangle$. Simulate 100 measurements in the computational basis and plot the results compared with the theoretical prediction, for instance in the form of a bar chart.

task 2: QPD



Build a QPD circuit based on the scheme above. Compare how the Nash equilibrium strategies work for different values of entanglement. In particular:

2a (0.3 pts) Plot the payoff of one of the players for a range of entanglement parameters $\gamma \in [0, \pi/2]$ (for at least 100 values of γ) and a set of strategies: $\widehat{D}\otimes\widehat{D}$, $\widehat{D}\otimes\widehat{Q}$, $\widehat{Q}\otimes\widehat{D}$, $\widehat{Q}\otimes\widehat{Q}$, all on the same figure. You should find some interesting features of the plot for $\gamma_{th1} = \arcsin\left(\sqrt{1/5}\right)$ and $\gamma_{th2} = \arcsin\left(\sqrt{2/5}\right)$.

2b (0.2 pts) A taste of quantum supremacy. For maximal entanglement obtained with $\gamma = \pi/2$, plot Alice's pay-off as a function of θ when Bob applies $\widehat{U}(\theta, 0)$ and Alice applies $\hat{C}$, $\widehat{D}$ or the "miracle move" $\widehat{M} = 1/\sqrt{2}\begin{pmatrix} i & 1 \\ -1 & -i \end{pmatrix}$.

2c (0.2 pts) Plot $m = \max_{\widehat{U}_A} \min_{\widehat{U}_B \in \{\hat{C}, \widehat{D}\}} \$_A(\widehat{U}_A, \widehat{U}_B)$ as a function of γ.

extra tasks:

(0.2 pts) Reproduce Fig. 2 and 3 from https://arxiv.org/pdf/quant-ph/9806088.pdf with a manipulable 3D plot.

(0 pts and priceless experience) Read Lesson 1 on Qiskit from https://arxiv.org/abs/1903.04359 to get a head start on next week's class.

hints:

- tensor products in Python can be computed with `np.kron`:

```
psi_A, psi_B = …, …
Psi_i = np.kron(psi_A, psi_B)

U_A, U_B = …, …
U_AB = np.kron(U_A, U_B)

Psi_f = U_AB @ Psi_i
```

- one way to do 2c is to use the differential evolution algorithm:

```
def payoff_A(angles):
    theta, phi = angles
    payoff = …
    return -payoff # diff. evo. minimizes, so to maximize we minimize the negation

result = scipy.differential_evolution(payoff_A, bounds=[(0, np.pi), (0, np.pi/2))
```