Introduction to Quantum Computing

Presentation · December 2018

DOI: 10.13140/RG.2.2.29728.43524

1 author:

Ioan Valeriu Grossu
University of Bucharest

Introduction to Quantum Computing
Second Edition - Revised

Dr. I.V. Grossu

Linear Algebra Basics

Complex Numbers

Matrices

Vector Spaces

Quantum Mechanics Basics

de Broglie Wavelength

Wave Function Concept

Bohr Atomic Model

Electron Spin

Quantum Computing Basics

Qubit

Register

Quantum Algorithm

Measurement in Computational Basis

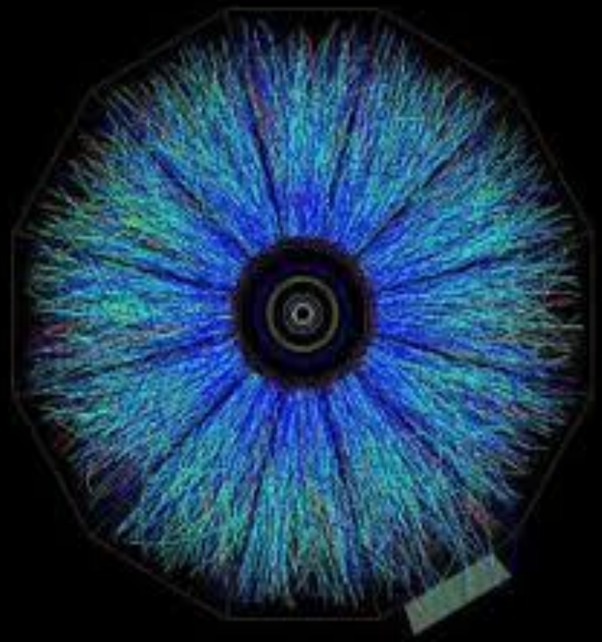
Partial Measurement

Quantum Entanglement

Quantum Algorithm Example (Grover)

Abstract

In this work I tried to create an intuitive introduction to the fascinating domain of Quantum Computing. Following this purpose, the first two chapters are conceived as an overview of the minimum Mathematics and Physics necessary notions, while the third chapter is dedicated to the main Quantum Computing concepts: Qubit, Quantum Register, Quantum Algorithm, Measurement, and Partial Measurement. Quantum entanglement is also discussed together with an intuitive analogy, and a concrete implementation on IBM Q Experience. In this context, I considered also a possible compromise between Bohr's and Einstein's interpretations. The fourth chapter is discussing Grover's algorithm in contrast with classical sequential search.



Particle collisions are an “eye” to quantum world.

Quantum Algorithm Example (Grover)

Quantum Entanglement

Partial Measurement

Measurement in Computational Basis

Quantum Algorithm

Register

Qubit

Quantum Computing Basics

Electron Spin

Bohr Atomic Model

Wave Function Concept

de Broglie Wavelength

Quantum Mechanics Basics

Vector Spaces

Matrices

Complex Numbers

Linear Algebra Basics

Linear Algebra Basics [1]

Complex Numbers

$z = (x, y)$; x, y in **R**

$z_1 + z_2 = (x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2)$

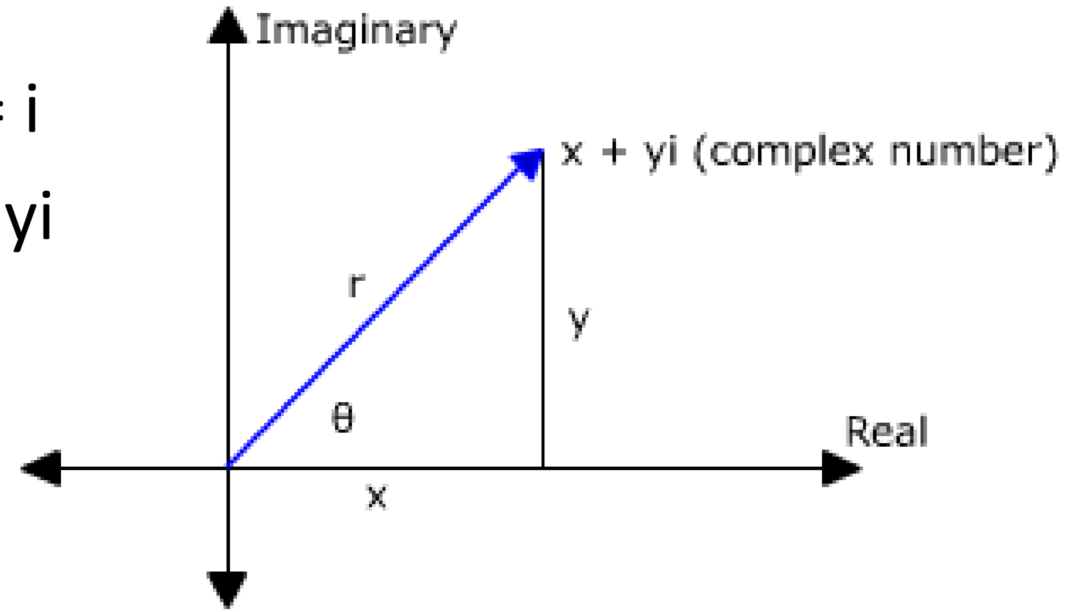
$z_1 * z_2 = (x_1x_2 - y_1y_2, x_1y_2 + x_2y_1)$

C – the set of complex numbers

$(0, 1) * (0, 1) = (-1, 0) \Rightarrow (0, 1) = \mathrm{SQRT}(-1) = i$

$(x, y) = (x, 0) + (0, y) = x(1, 0) + y(0, 1) = x + yi$

$|z| = \mathrm{SQRT}(x^2+y^2)$ - **Pythagoras**



Complex Numbers

z* = x – yi -> the <u>complex conjugate</u> of z

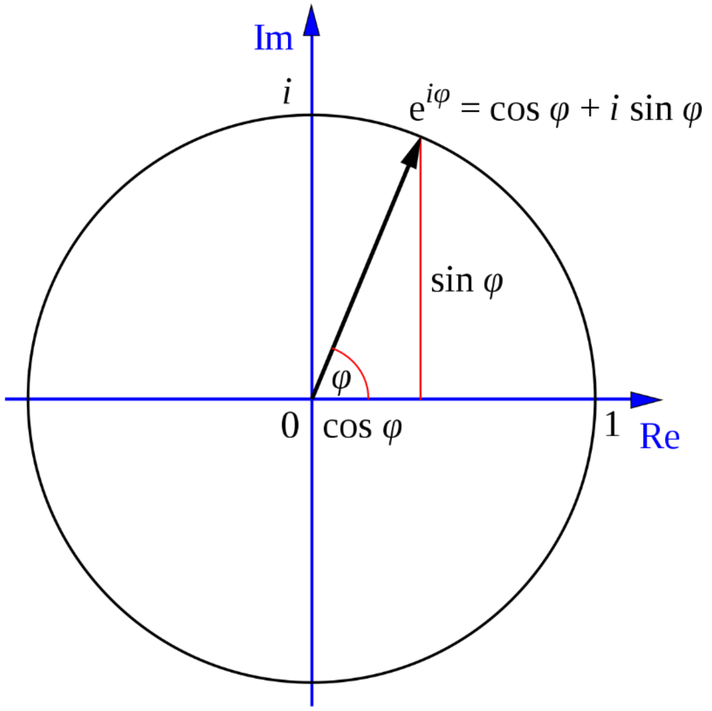
z z* = (x + yi)(x - yi)

= xx – xyi + yxi –yyii

= $x^2 + y^2 = |z|^2$

$$(\cos x + i \sin x)^n = \cos nx + i \sin nx$$

$e^{ix} = \cos x + i \sin x$ **Euler's Formula**

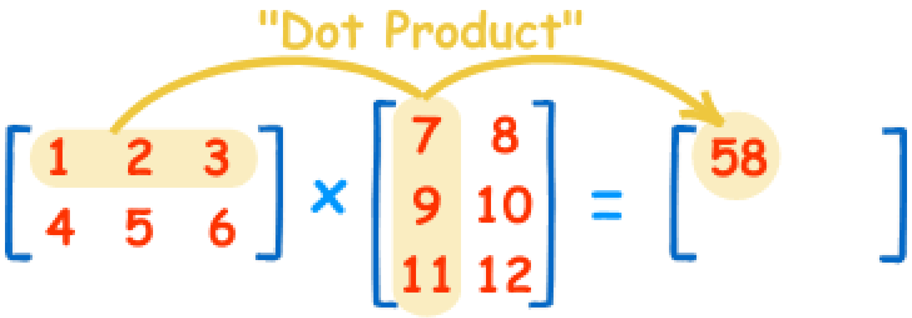


Matrices

Matrix Product

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \quad m \times n \; matrix \qquad B = \begin{bmatrix} b_{11} & \cdots & b_{1p} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{np} \end{bmatrix} \quad n \times p \; matrix$$

$$A \times B = \begin{bmatrix} \sum_{i=1}^{n} a_{1i} b_{i1} & \cdots & \sum_{i=1}^{n} a_{1i} b_{ip} \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} a_{mi} b_{i1} & \cdots & \sum_{i=1}^{n} a_{mi} b_{ip} \end{bmatrix} \quad m \times p \; matrix$$



Matrices

Examples:

$$\begin{bmatrix} 3 & 4 & 5 \end{bmatrix} \times \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} = 3 \times 3 + 4 \times 4 + 5 \times 5 = 50$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} = \begin{bmatrix} 12 \\ 12 \end{bmatrix}$$

$$\begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} \times \begin{bmatrix} 3 & 4 & 5 \end{bmatrix} = \begin{bmatrix} 9 & 12 & 15 \\ 12 & 16 & 20 \\ 15 & 20 & 25 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 11 & 12 & 13 \\ 21 & 22 & 23 \\ 31 & 32 & 33 \end{bmatrix} = \begin{bmatrix} 11 & 12 & 13 \\ 21 & 22 & 23 \\ 31 & 32 & 33 \end{bmatrix}$$

$$I_n \times A = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} \times A = A$$

Matrices

Examples

A^T => **transposed of A** => $a_{ij} = a^T{}_{ji}$

$$\mathbf{A} \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

$(A^*)^T$ - transpose-conjugate matrix:

$$\left(\begin{bmatrix} 1+1i & 1+2i & 1+3i \\ 2+1i & 2+2i & 2+3i \\ 3+1i & 3+2i & 3+3i \end{bmatrix}^*\right)^T = \begin{bmatrix} 1-1i & 2-1i & 3-1i \\ 1-2i & 2-2i & 3-2i \\ 1-3i & 2-3i & 3-3i \end{bmatrix}$$

Matrices

Kronecker Product

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \quad m \times n \; matrix \qquad B = \begin{bmatrix} b_{11} & \cdots & b_{1q} \\ \vdots & \ddots & \vdots \\ b_{p1} & \cdots & b_{pq} \end{bmatrix} \quad p \times q \; matrix$$

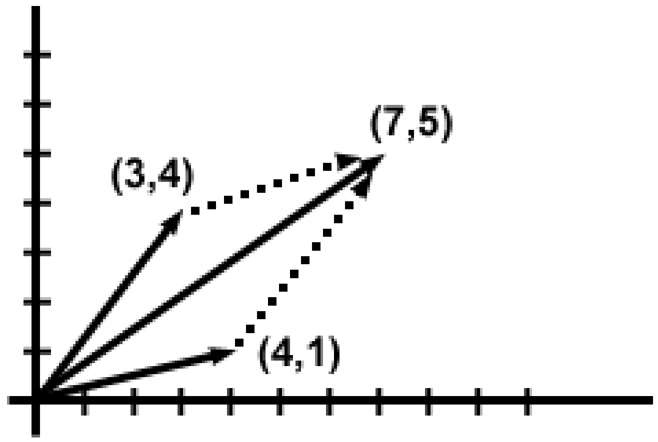
$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \quad mp \times nq \; matrix$$

$$\begin{bmatrix} 0 & 1 \\ 2 & 4 \end{bmatrix} \otimes \begin{bmatrix} 11 & 12 \\ 21 & 22 \end{bmatrix} = \begin{bmatrix} 0 \times \begin{bmatrix} 11 & 12 \\ 21 & 22 \end{bmatrix} & 1 \times \begin{bmatrix} 11 & 12 \\ 21 & 22 \end{bmatrix} \\ 2 \times \begin{bmatrix} 11 & 12 \\ 21 & 22 \end{bmatrix} & 4 \times \begin{bmatrix} 11 & 12 \\ 21 & 22 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 11 & 12 \\ 0 & 0 & 21 & 22 \\ 22 & 24 & 44 & 48 \\ 42 & 44 & 84 & 88 \end{bmatrix}$$

Vectors in Physics

Quantities with both magnitude and direction (x, y, z)

Examples: velocity, force, position

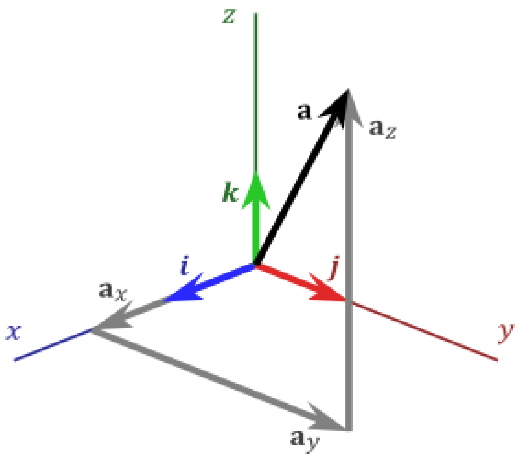


Multiplication with a scalar:

scalar * (x_1, y_1) = (scalar * x_1, scalar * y_1)

Sum (parallelogram's rule):

$\mathbf{v_1} + \mathbf{v_2} = (x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2)$



In 3D:

$|\mathbf{v}| = \mathrm{Sqrt}(x^2 + y^2 + z^2)$ => Pythagoras

$\vec{v} = x(1,0,0) + y(0,1,0) + z(0,0,1) = x\vec{\imath} + y\vec{\jmath} + z\vec{k}$

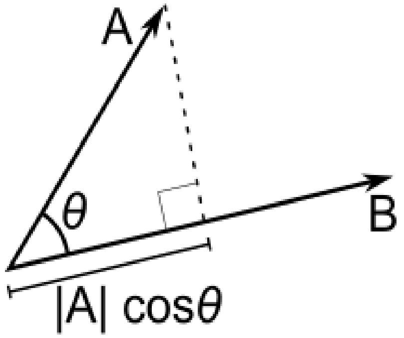
Vectors in Physics

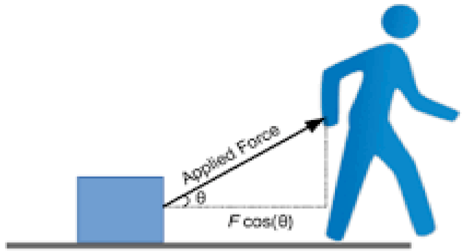
<u>Scalar Product</u>:

$\mathbf{v_1} \cdot \mathbf{v_2} = x_1x_2 + y_1y_2 = |\mathbf{v_1}||\mathbf{v_2}|\cos(\mathbf{v_1}\mathbf{v_2})$

$\mathbf{v_1} \cdot \mathbf{v_2} = 0 \Rightarrow \mathbf{v_1}$ rectangular on $\mathbf{v_2}$

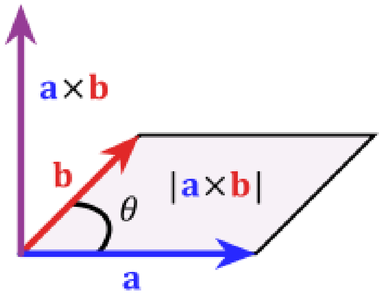
Example: The **work** of a force





<u>Vector product</u>:

$$\vec{a} \times \vec{b} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix}$$



Vectors could be also generalized to **ND**

$$\vec{v} = \sum_{k=1}^{N} x_k \vec{i}_k; \;\; \vec{i}_k . \vec{i}_j = \delta_{kj}$$

Linear Algebra

Vector Spaces

A Vector Space *V* over a field of complex numbers **C** is a **non-empty set** of elements on which **two operations** (functions) are defined (a **generalization of Vectors**):

Multiplication with a vector by a scalar in C

- associativity
- neutral element 1
- distributivity

Addition

- associative
- commutative
- neutral element: **0 vector**
- for each v in *V* exists -1 v => the inverse element

Linear Algebra

Hilbert Spaces

Hilbert Space = a Vector Space with an: **Inner Product (.,.) : *V X V -> C***

linear in second argument

$$v\sum_{i=1}^{n} a_i v_i = \sum_{i=1}^{n} a_i (v v_i)$$

$(v_1, v_2)=(v_2, v_1)^*$

(v, v) >= 0, equality only for v = **0**

in general, not linear in first argument

Vector Spaces in C^n

C^n

$$|\boldsymbol{v}> = v = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}, \quad v_i \in \boldsymbol{C};\ \boldsymbol{ket}$$

$< \boldsymbol{v}| \stackrel{\text{def}}{=} [v_1^* \cdots \quad v_n^*]$, dual vector; **bra**

$(\boldsymbol{v}, \boldsymbol{w}) = \sum_{i=1}^{n} \boldsymbol{v}_i^* \boldsymbol{w}_i = < \boldsymbol{v}|\boldsymbol{w} >$; **bracket**

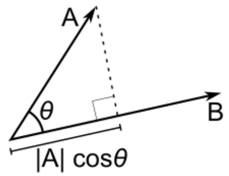
$$< \boldsymbol{v}|\boldsymbol{w} > = \boldsymbol{0} \Rightarrow \boldsymbol{v} \perp \boldsymbol{w}$$

$$\||\boldsymbol{v} >\| \stackrel{\text{def}}{=} \sqrt{< \boldsymbol{v}|\boldsymbol{v} >}$$

Physics

$$\vec{v} = (v_x, v_y, v_z), \quad v_{x,y,z} \in R$$

$$\vec{v}\vec{w} = v_x w_x + v_y w_y + v_z w_z$$
$$= vw\cos(\theta)$$



$$\vec{v}\vec{w} = 0 \Rightarrow \vec{v} \perp \vec{w} \quad (\cos\theta = 0)$$

$|\vec{v}| = \sqrt{\vec{v}\vec{v}}$ Pythagoras

Vector Spaces in C^n

C^n

$$|\boldsymbol{v}> = \begin{bmatrix} v_0 \\ \vdots \\ v_{n-1} \end{bmatrix}$$

$$= v_0 \begin{bmatrix} 1 \\ \vdots \\ 0 \end{bmatrix} + \cdots + v_{n-1} \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix} =$$

$$= v_0|0> + \cdots + v_{n-1}|n-1>$$

$$= \sum_{i=0}^{n-1} v_i \,|i>$$

Canonical Computational Basis in C^n (orthonormal)

$$|0>, \ldots, |n-1>, \qquad < i|j> = \delta_{ij}$$

Physics

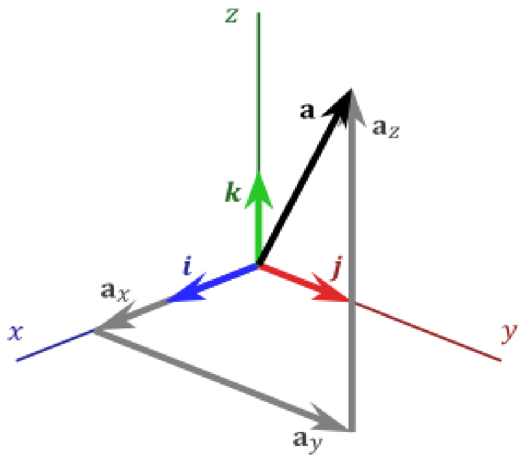
$$\vec{v} = (v_x, v_y, v_z)$$

$$= v_x(1,0,0) + v_y(0,1,0) + v_z(0,0,1)$$

$$= v_x\vec{i} + v_y\vec{j} + v_z\vec{k}$$

Orthonormal Basis:

$\vec{i}, \vec{j}, \vec{k}$



Quantum Mechanics Basics [1-3]

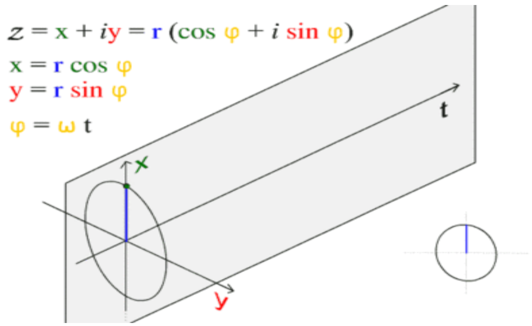
Waves

- Harmonic oscillator:

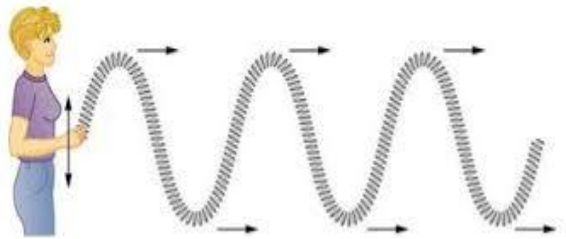
$$F = k\,y,\;\; y = A\,sin(\omega t)$$

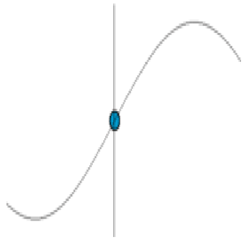
$\Psi(t) = Ae^{\,i\,\omega t}$ phasor (write more compact expressions)

$$E = 0.5\,k\,A^2 \sim A^2$$



- Wave –> propagation of a perturbation => at distance x, the same oscillation has the delay x/c:



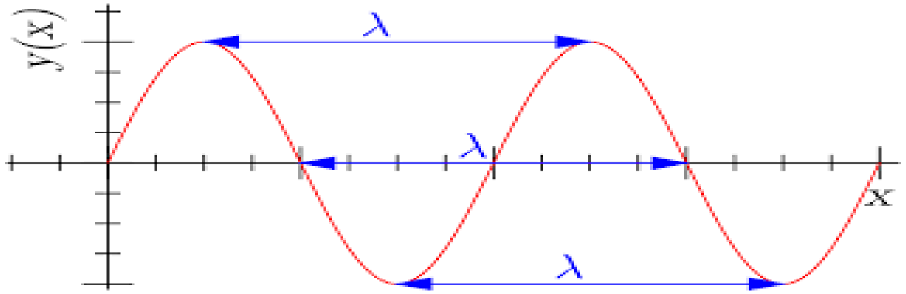


$$\nu = 1/\mathrm{T};\quad c = \lambda/\mathrm{T} = \lambda\nu;\ \omega = 2\pi/\mathrm{T} = 2\pi\ \nu$$

$$\Psi(x,t) = A\sin[2\pi(\frac{x}{\lambda} - \frac{t}{T})]$$

$e^{ix} = \cos x + i\sin x$ **Euler's Formula**

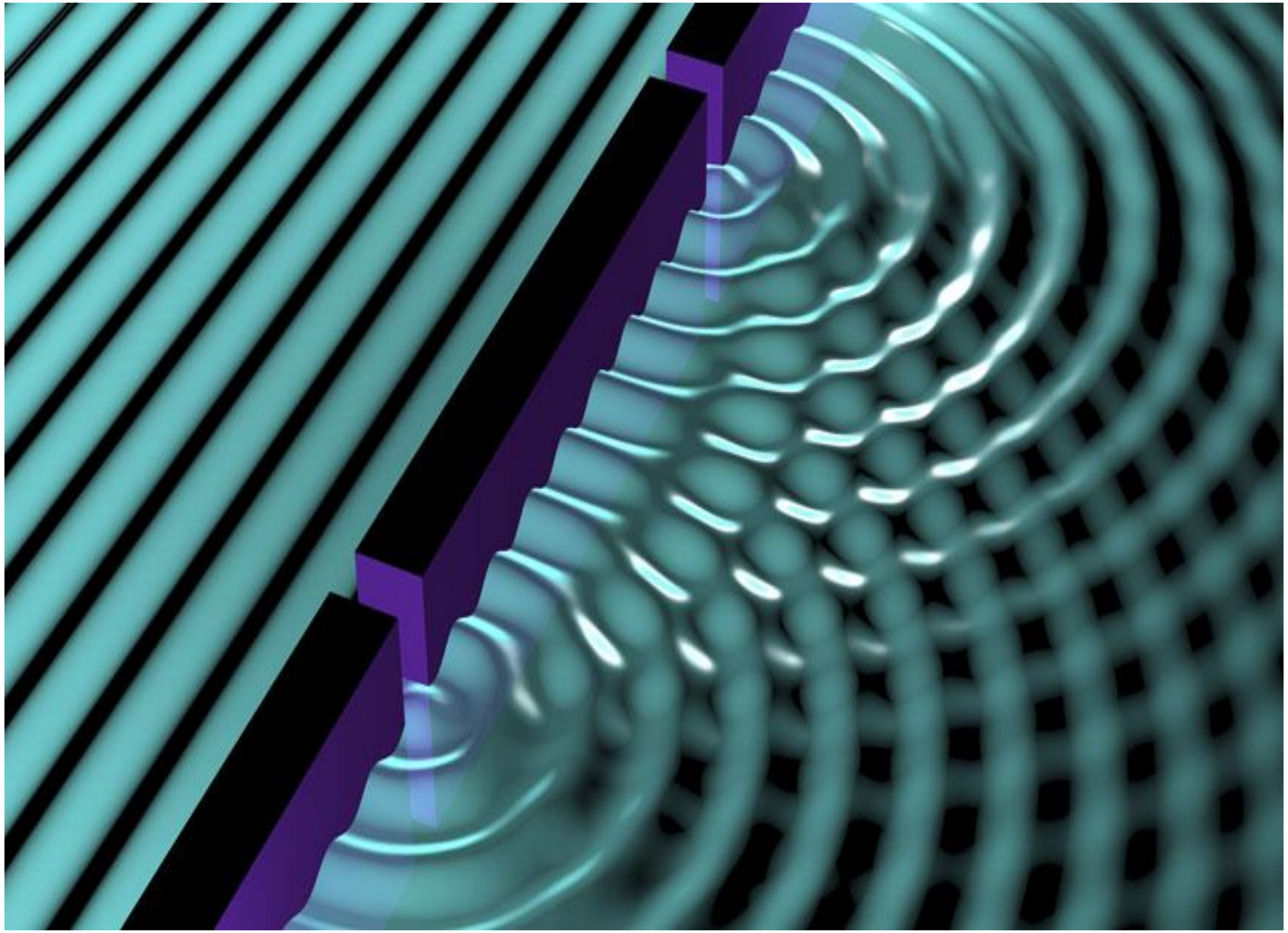
$$\Psi(x,t) = Ae^{i2\pi(x/\lambda - t/T)}$$

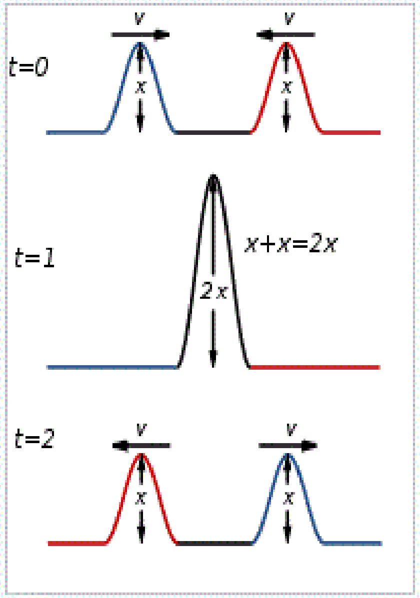


Waves

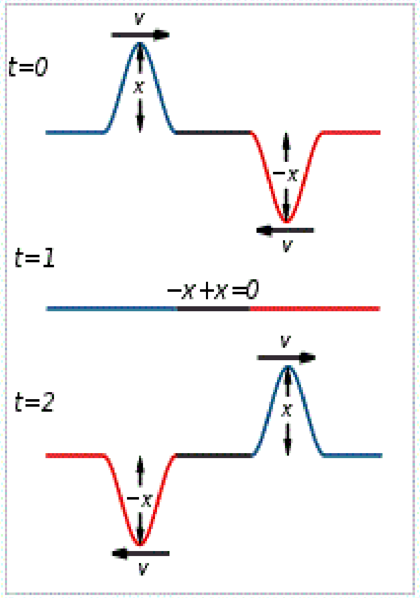
Huygens Principle

In 1678 Huygens proposed a model where each point on a wave front may be regarded as a source of waves expanding from that point.



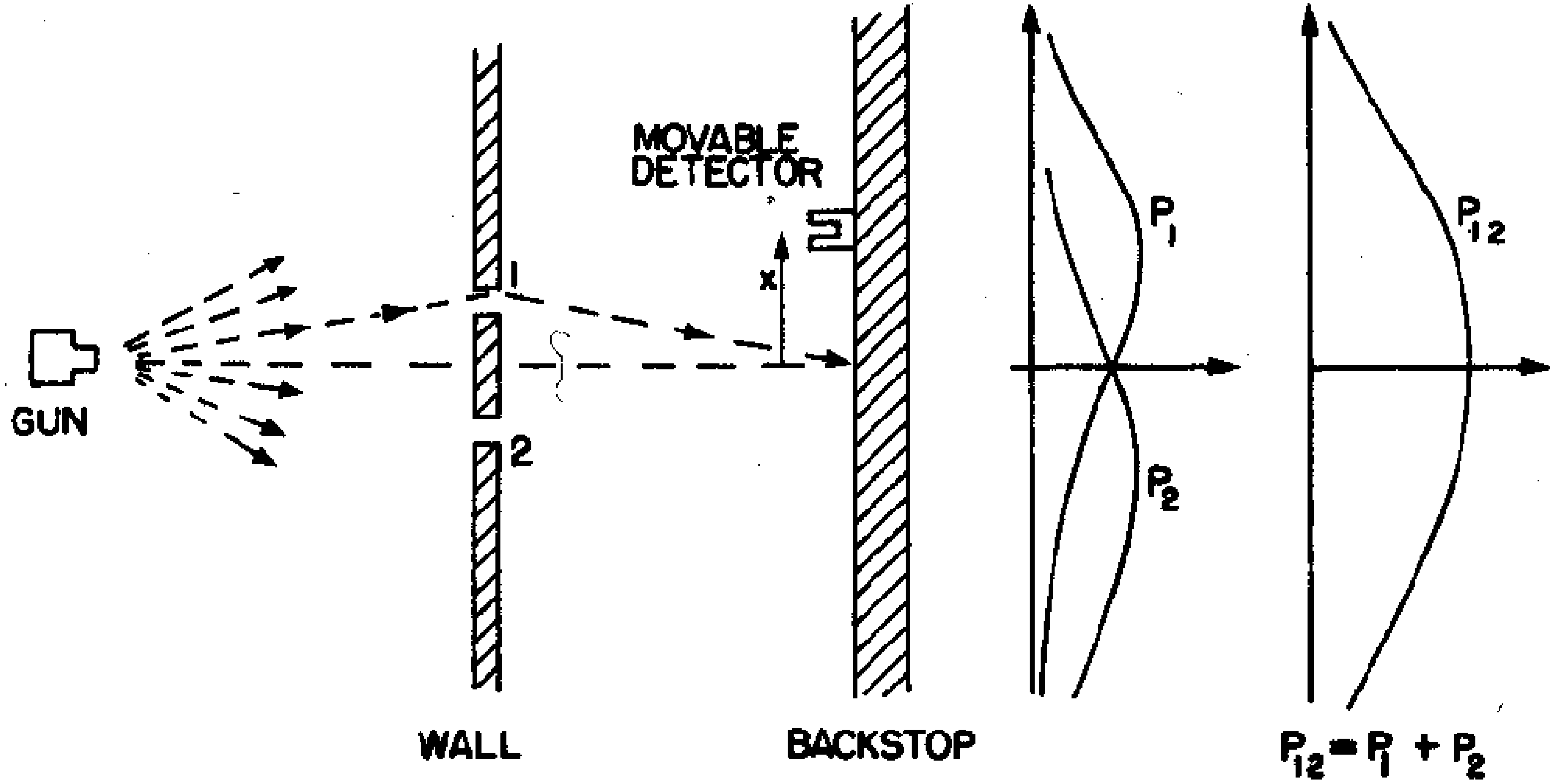


Constructive interference: When the tops of two waves meet in the middle a taller wave is seen.



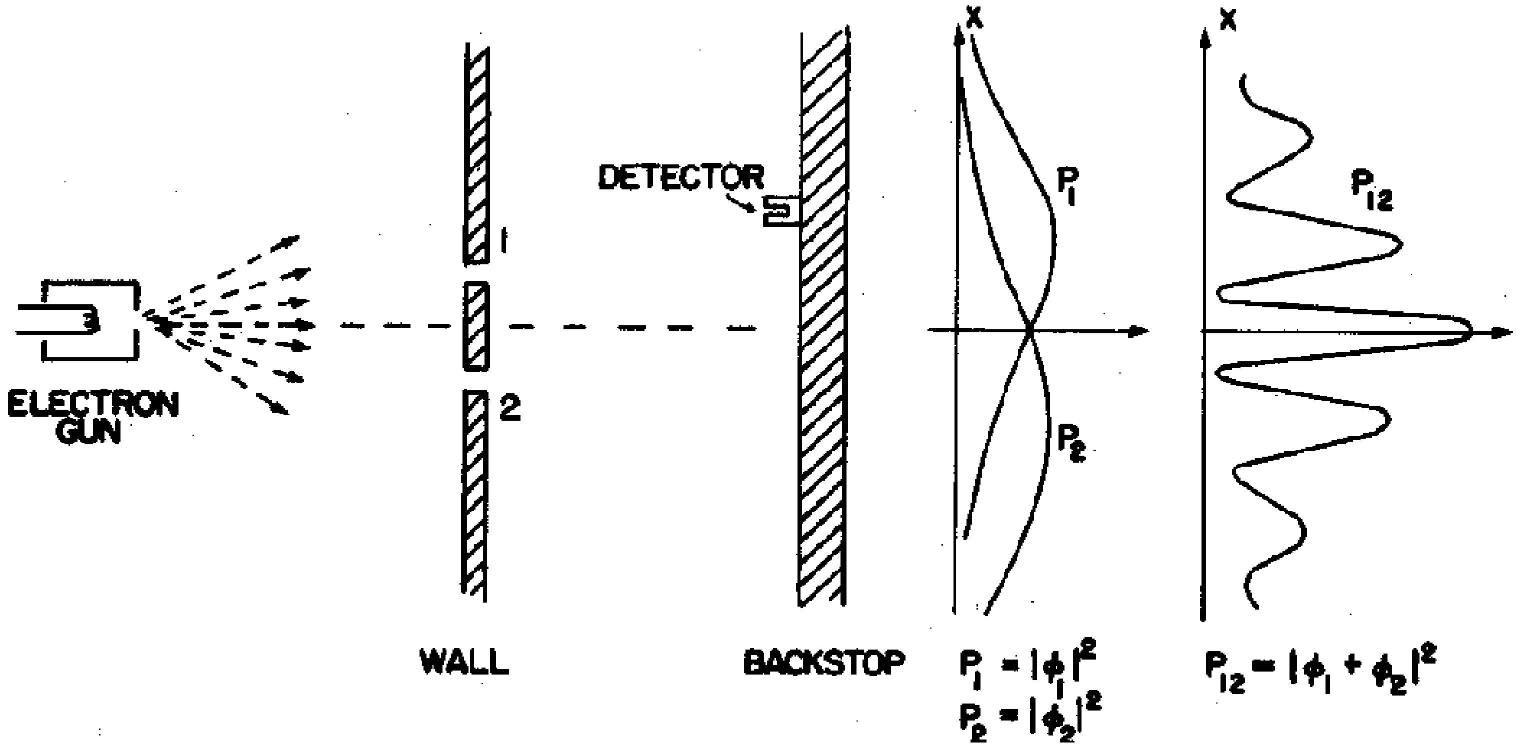
Destructive interference: When the top of one wave meets the trough (bottom) of another wave in the middle, the water stays flat.

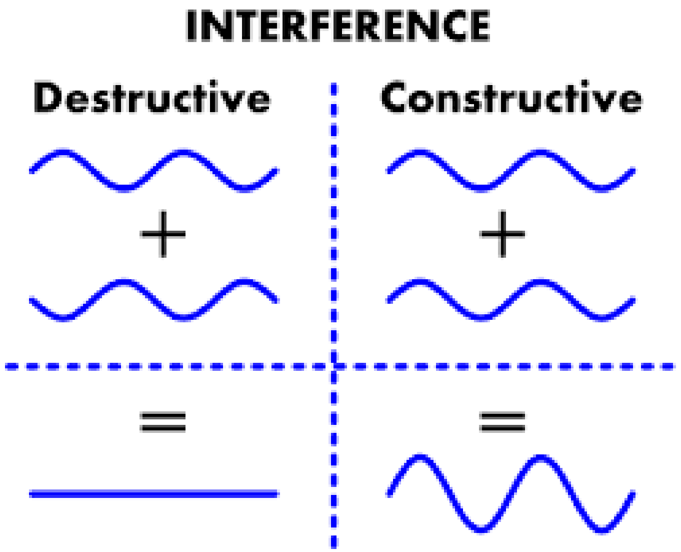
de Broglie Wavelength



Electron (particle?) => classical (e.g. tennis balls) expected result

de Broglie Wavelength





Experimental result (e.g. Davisson-Germer) – Electron’s diffraction

Wave Function Concept

- Diffraction is present even for very low intensities (one particle at a time => the electron “interferes with itself”)
- Where did the electron pass through?... **Both slits!**
- This, difficult to understand, non classical behavior is available as long as the system is isolated.
- The measurement process affects the system (“wave function collapse”)

  => only one state can be found by measuring => **The main limitation in Quantum Computing!**

- One interpretation: the wave function is all that one could know about the particle. It is a truth that we should not try to explain by other concepts.
- Another interpretation: there are “hidden variables” inaccessible in present.

***$h = 6.626 \times 10^{-34}\ m^2Kg/s$* – Planck’s constant**

$$\boldsymbol{\lambda = \frac{h}{mv} = \frac{h}{p}}$$ **de Broglie wavelength,** ***E = hv***

$$\Psi(x,t) = Ae^{i2\pi(x/\lambda - t/T)} = Ae^{i(px-Et)/\hbar}$$ wave function example

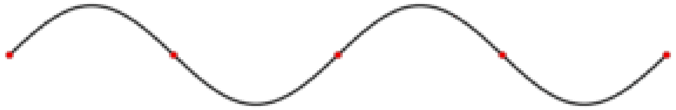
$$\rho(x,t) = \Psi(x,t)\Psi^*(x,t) = |\Psi(x,t)|^2$$ -> **probability density**

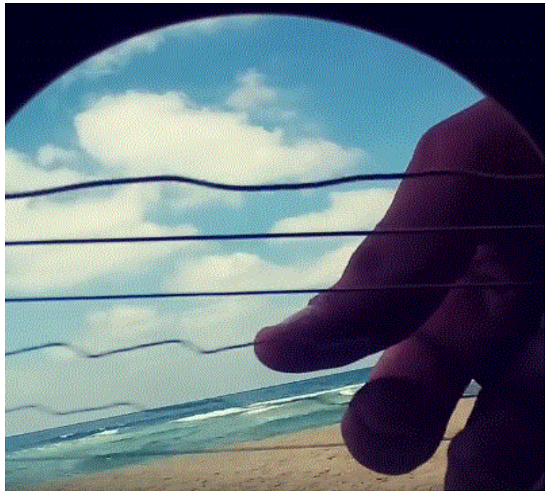
$$\int_{all\ space} |\Psi(x,t)|^2 dx = 1$$ -> normalization

Bohr Atomic Model

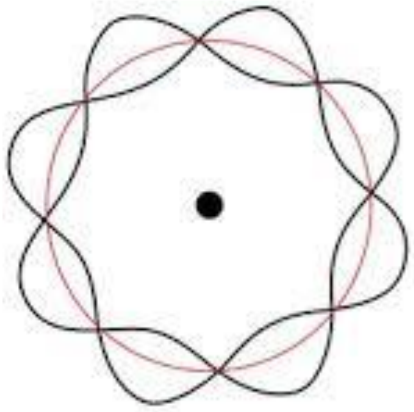
- Hydrogen spectra is discrete
- Bohr atomic model: only the states corresponding to **stationary waves** are allowed => **angular momentum quantification**

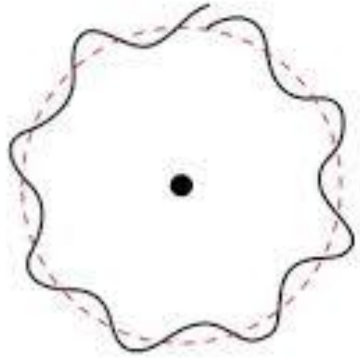
$$2\pi r = n\lambda = n\frac{h}{p} \Rightarrow rp = n\frac{h}{2\pi} = n\hbar$$

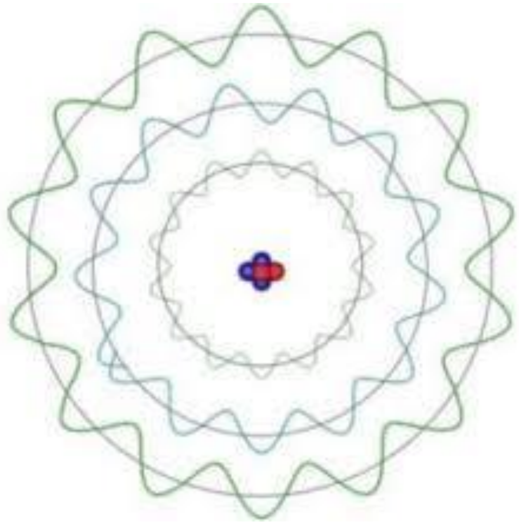


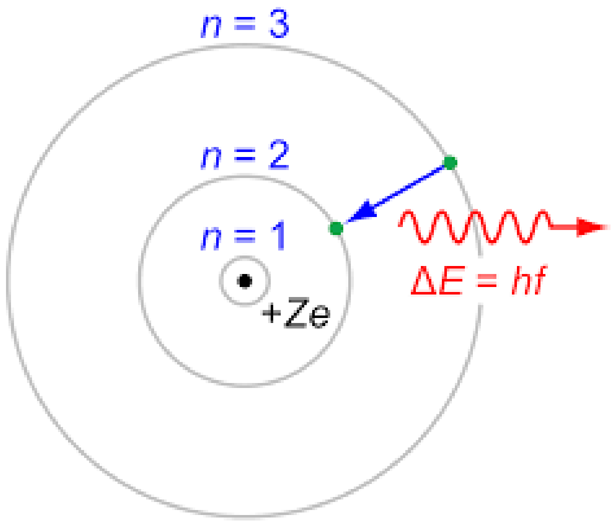


https://www.facebook.com/hossamibrahemm/posts/2021133031242089



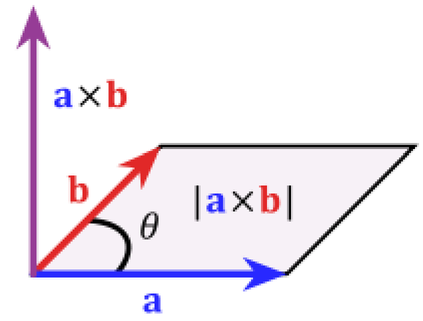


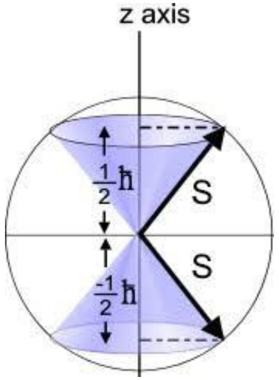




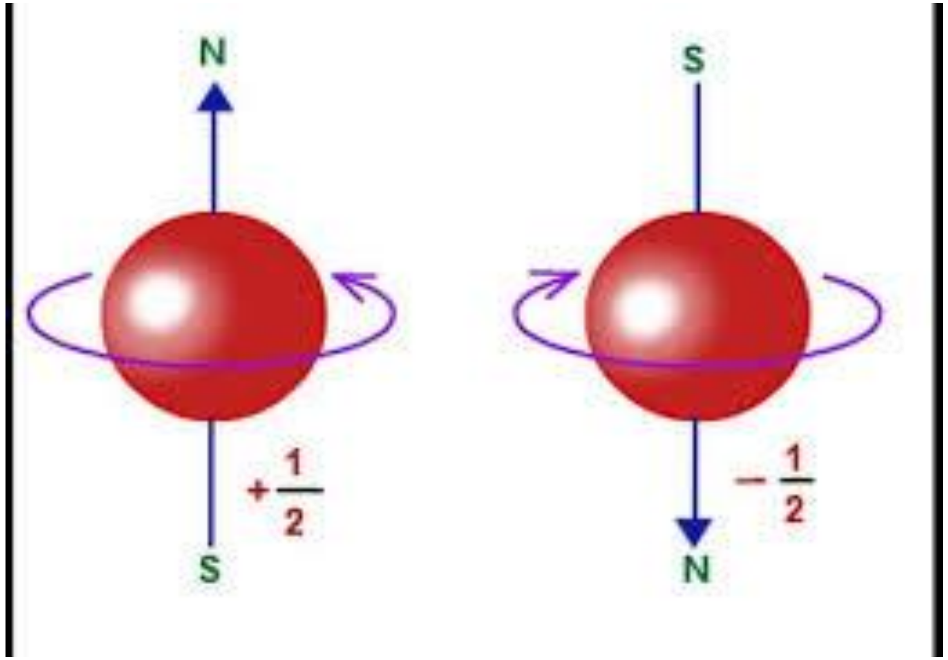
Electron Spin

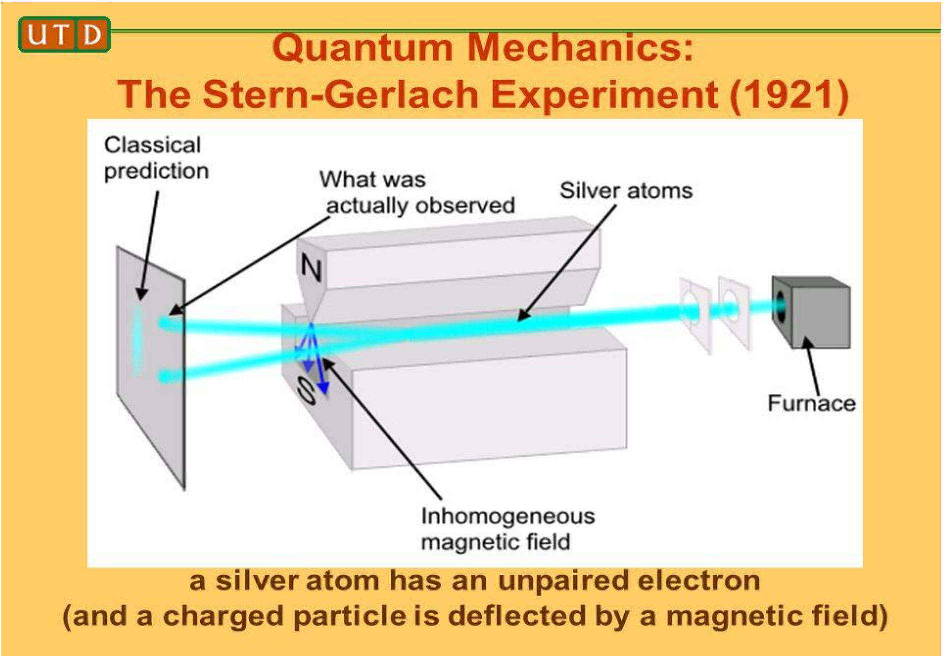
- Not only the angular momentum is quantified, but also its projection in a magnetic (**Zeeman Effect**) or electric field (**Stark Effect**)
- The elecron has an intrinsic magnetic momentum [3] (**intuitively** related to an intrinsic angular momentum, spin), with only 2 possible projections.
- An isolated electron is a superposition of both states (no classical equivalent).





Quantification of electron spin projections





Quantum Computing Basics [1,4-8]

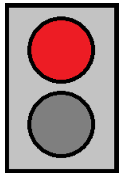
QUBIT - Analogy

- Consider a semaphore which **randomly** switches from red to green and back in such a way that the probabilities of finding it (e.g. by taking a picture) in one specific color are: *r* (for red), and *g* (for green). Obviously:

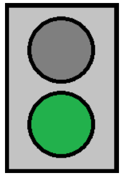
$$r + g = 1$$

- Supposing the semaphore switches enough fast from one color to another, one could approximate it is both red and green in the same time (e.g. the eye will perceive both lights are on, but with different intensities) .
- Another example: at each moment of time, tossing an unusual, unfair coin (with unequal probabilities associated to its sides).

$$|0> = \begin{bmatrix}1\\0\end{bmatrix}$$



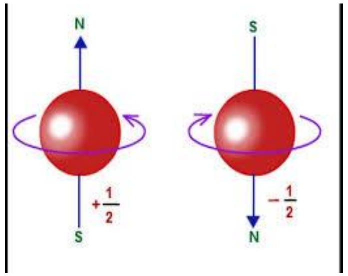
$$|1> = \begin{bmatrix}0\\1\end{bmatrix}$$



QUBIT

- ***State-Space Postulate***: *An isolated physical system* ***has an associated Hilbert space****, called the state space. The state of the system is fully described by a unit vector, called the state vector in that Hilbert space.*
- **QUBIT**: A **unit vector** in ***C²*** (e.g. electron's spin)
- Canonical base in ***C²***:

$$|0> = \begin{bmatrix}1\\0\end{bmatrix}, \ |1> = \begin{bmatrix}0\\1\end{bmatrix}$$ ORTHONORMAL BASIS



$$\Psi = \alpha|0> \ + \beta|1>, \quad |\alpha|^2 + |\beta|^2 = 1$$

$$||\Psi||^2 = (< 0|\alpha^* + < 1|\beta^*)(\alpha|0 > + \beta|1 >) =$$
$$= \ \alpha^*\alpha < 0|0 > \ + \ \alpha^*\beta < 0|1 > + \beta^*\alpha < 1|0 > + \beta^*\beta < 1|1 >$$
$$= \alpha^*\alpha + \beta^*\beta = \ |\alpha|^2 + |\beta|^2 = 1$$

- **Superposition of "opposite movements", without classical equivalent**.
- The supperpositions applies to one single particle, not to a statistic set.
- The superposition applies to **isolated** systems.
- The postulate does not give details on how to construct the corresponding Hilbert space
- By measuring we'll find a single value (with the corresponding probabilities: $|\alpha|^2$, $|\beta|^2$). More details will be discussed in the following chapters.

Quantum Register - Analogy

- In the frame of the same "semaphores analogy", one could consider two independent semaphores, randomly switching, enough fast, from red to green and back, with the corresponding probabilities:

  (r_1, g_1), *and* (r_2, g_2)

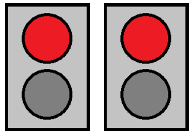
- Another example: at each moment of time, tossing two unfair, independent coins.
- The probabilities of compound states could be obtained by multiplication of individual probabilities:

  $r_1 * r_2,\ r_1 * g_2,\ g_1 * r_2,\ g_1 * g_2$

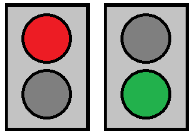
- One could also use Kronecker product:

$$\begin{bmatrix} r_1 \\ g_1 \end{bmatrix} \otimes \begin{bmatrix} r_2 \\ g_2 \end{bmatrix} = \begin{bmatrix} r_1 \begin{bmatrix} r_2 \\ g_2 \end{bmatrix} \\ g_1 \begin{bmatrix} r_2 \\ g_2 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} r_1 r_2 \\ r_1 g_2 \\ g_1 r_2 \\ g_1 g_2 \end{bmatrix}$$

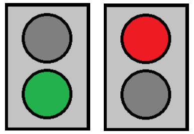
$$p = r_1 r_2, \qquad \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$



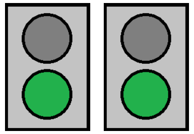
$$p = r_1 g_2, \qquad \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$



$$p = g_1 r_2, \qquad \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$



$$p = g_1 g_2, \qquad \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$



$$r_1 r_2 + r_1 g_2 + g_1 r_2 + g_1 g_2 = (r_1 + g_1)\,(r_2 + g_2) = 1$$

Quantum Register

- ***Composite Systems Postulate***: *The state space of a composite system is the tensor product of the state space of the components.*

- => the composite space canonical base is obtained by **Kronecker multiplication** of component spaces canonical bases.

- This postulate gives the “**recipe**” **for constructing the Hilbert space of a composite system**.

- **Register**: Set of qubits treated as a composite system (e.g. 2 QUBITS => ***C4***)

$$|0> = |0,0> = |0> \otimes |0> = \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1\begin{bmatrix}1\\0\end{bmatrix}\\0\begin{bmatrix}1\\0\end{bmatrix}\end{bmatrix} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}$$

$$|1> = |0,1> = |0> \otimes |1> = \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}1\begin{bmatrix}0\\1\end{bmatrix}\\0\begin{bmatrix}0\\1\end{bmatrix}\end{bmatrix} = \begin{bmatrix}0\\1\\0\\0\end{bmatrix}$$

$$|2> = |1,0> = |1> \otimes |0> = \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}0\begin{bmatrix}1\\0\end{bmatrix}\\1\begin{bmatrix}1\\0\end{bmatrix}\end{bmatrix} = \begin{bmatrix}0\\0\\1\\0\end{bmatrix}$$

$$|3> = |1,1> = |1> \otimes |1> = \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\begin{bmatrix}0\\1\end{bmatrix}\\1\begin{bmatrix}0\\1\end{bmatrix}\end{bmatrix} = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

$$\Psi = \sum_{i=0}^{2^n-1} a_i|i>; \sum_{i=0}^{2^n-1} |a_i|^2 = 1$$

Unitary Operators

Linear Operator: function A_{op}:*V->W* (V, W - vector spaces)

$$A_{op}(\sum_i a_i \, |v_i >) = \sum_i a_i A_{op}(|v_i >)$$

This is **equivalent with** considering **a corresponding matrix** *A*:

$A_{op}(|v>) = A \; |v>$ (matrix multiplication)

Adjoint operator: => $A^{+} = (A^{*})^{T}$ (transpose-conjugate matrix)

A

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

Unitary Operator U: $U^{+}U = UU^{+} = I$ => $\| U|v >\| = \| |v >\|$

Unitary operators **are preserving the norm**

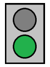
Quantum Algorithm

***Evolution Postulate**: The time evolution of an isolated quantum system is described by a unitary transformation.*

- **Quantum algorithm:** a prescription of a sequence of unitary operators applied to an initial state:

$$|\Psi_n> = U_n \dots U_1|\Psi_1>$$

- **A quantum logic gate** (or simply quantum gate) is a basic quantum circuit operating on a small number of qubits.
- Unlike many classical logic gates, quantum logic gates are reversible (unitary operators).



- In "semaphores analogy", an unitary operator is a device capable of changing the red/green probability distribution.

Unitary Operator Examples:

Pauli Matrices:

$$\sigma_0 = I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \sigma_1 = X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_2 = Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma_3 = Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$X|0> = |1>, \quad X|1> = |0>, \quad Z|0> = |0>, \quad Z|1> = -|1>$$

Unitary Operator Examples

$$\text{Hadamard} = \text{H} = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$

$$H|0> = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix} = \begin{bmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 0\end{bmatrix} + \frac{1}{\sqrt{2}}\begin{bmatrix}0\\ 1\end{bmatrix} = \frac{|0> \ + |1>}{\sqrt{2}} = | +>, \qquad H|1> = \frac{|0> \ - |1>}{\sqrt{2}} = | ->$$

$$H \otimes I = \begin{bmatrix}1/\sqrt{2} & 1/\sqrt{2}\\ 1/\sqrt{2} & -1/\sqrt{2}\end{bmatrix} \otimes \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} = \begin{bmatrix}1/\sqrt{2} \times \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} & 1/\sqrt{2} \times \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\\ 1/\sqrt{2} \times \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} & -1/\sqrt{2} \times \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\end{bmatrix} = \begin{bmatrix}1/\sqrt{2} & 0 & 1/\sqrt{2} & 0\\ 0 & 1/\sqrt{2} & 0 & 1/\sqrt{2}\\ 1/\sqrt{2} & 0 & -1/\sqrt{2} & 0\\ 0 & 1/\sqrt{2} & 0 & -1/\sqrt{2}\end{bmatrix}$$

0

$$SWAP = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$ swaps two qubits => $$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}ac\\ ad\\ bc\\ bd\end{bmatrix} = \begin{bmatrix}ac\\ bc\\ ad\\ bd\end{bmatrix} = \begin{bmatrix}ca\\ cb\\ da\\ db\end{bmatrix} = \begin{bmatrix}c\\ d\end{bmatrix} \otimes \begin{bmatrix}a\\ b\end{bmatrix}$$

$$(a|0> + b|1>) \otimes (c|0> + d|1>) = \begin{bmatrix}a\\ b\end{bmatrix} \otimes \begin{bmatrix}c\\ d\end{bmatrix} = \begin{bmatrix}ac\\ ad\\ bc\\ bd\end{bmatrix} \neq \begin{bmatrix}a\\ b\\ c\\ d\end{bmatrix}$$

Hermitian Operators

Adjoint operator: => $A^{+} = (A^{*})^{T}$ (transpose-conjugate matrix)

Hermitian (self-adjoint) operator: $O^{+}=O$

- *A Hermitian operator is **diagonalizable** => there exists an orthonormal basis v_i of V such that:*

$$O = \sum_{i=1}^{n} \lambda_i \, |v_i >< v_i|$$

λ_i - eigenvalues, $|v_i>$ - eigenvectors

- ***The eigenvalues of a Hermitian operator are real numbers***
- A real symmetric matrix is Hermitian

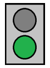
Measurement

Measurement Postulate: *A projective measurement is described by a Hermitian operator O, called observable in the state space of the system being measured. The observable O has a diagonal representation*:

$$O = \sum_{i=1}^{n} \lambda_i \, |v_i >< v_i| = \sum_{\lambda} \lambda P_\lambda$$

- P_λ takes into account the fact that the same eigenvalue λ could correspond to many eigenvectors.
- The measurement process is irreversible.
- The possible results of measurement of the observable O are the eigenvalues λ.
- The probability of obtaining the result λ is:

$$p_\lambda = \| P_\lambda |\Psi > \|^2 =< \Psi | P_\lambda | \Psi >$$



- In “semaphores analogy”, measurement is associated with a device capable of stopping systems’ oscillations, which results in “freezing” the semaphores into one compound state.

Measurement in Computational Basis

$$|\Psi> = \sum_{k=0}^{2^n-1} a_k |k>, \qquad \sum_k |a_k|^2 = 1$$

$$O = \sum_{k=0}^{2^n-1} k\ |k >< k|$$

The measurement result is **an integer value *0 <= k <= 2ⁿ-1***

The probability of obtaining the value *k* is:

$$p_k = ||k >< k|\Psi >|^2 = |a_k|^2$$

Partial Measurement

Measure only some qubits => for 2 subsystems: A, B => measuring the observable: $O_A \otimes I_B$

For m qubits and n qubits one could write:

$$|\Psi> = \sum_{i=0}^{2^m-1} \sum_{j=0}^{2^n-1} a_{ij} \, |i, j>$$

Measuring the m qubits, the probability to obtain the result *k* is:

$$p_k = \sum_{j=0}^{2^n-1} |a_{kj}|^2$$

If the result is *k*, the state immediately after is:

$$\frac{1}{\sqrt{p_k}} |\mathrm{k}> \left(\sum_{j=0}^{2^n-1} a_{kj} |j> \right)$$

Partially measurement results in affecting the measurement result of remaining qubits.

Partial Measurement

Example 1:

$$|\Psi > = \frac{1}{\sqrt{3}}(|0,0 > -i\ |0,1 > \ + |1,1 >)$$

Measuring the first qubit => $p_0 = 2/3$, $p_1 = 1/3$

If the result is 0, the state immediately after is (the state of the second qubit is a superposition):

$$\frac{1}{\sqrt{2/3}}|0 > \left(\frac{|0 > \ - i\ |1 >}{\sqrt{3}}\right)$$

Example 2:

$$|\Psi > = \frac{|0,0 > \ + |1,1 >}{\sqrt{2}}$$

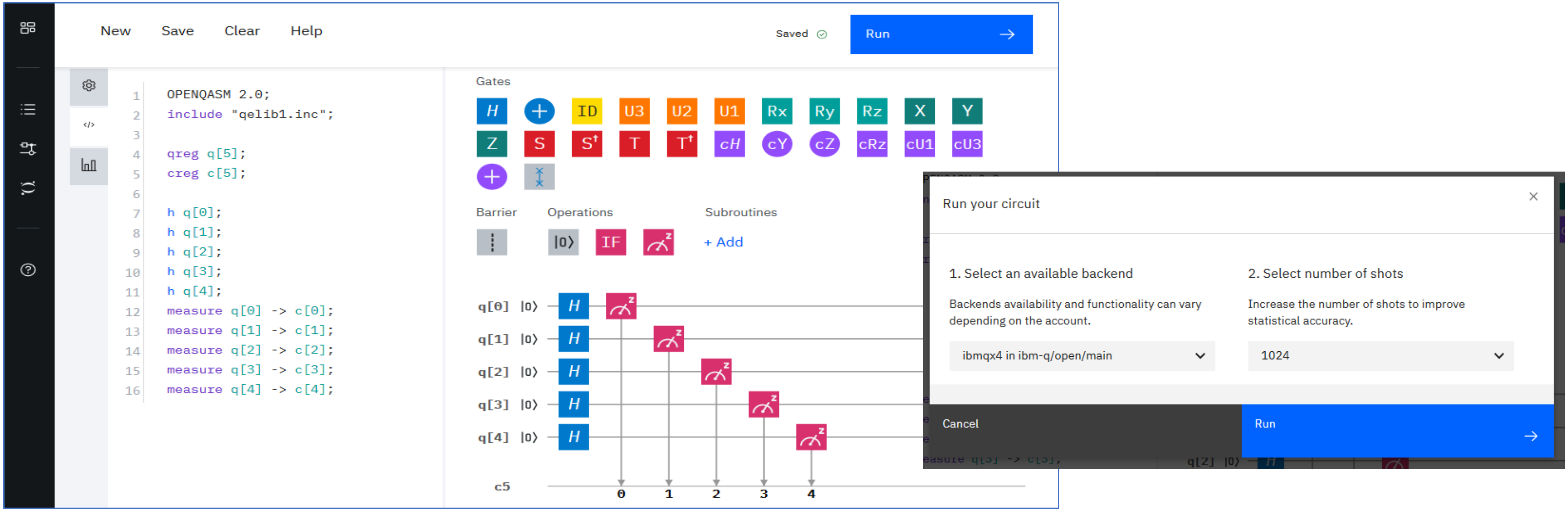
If the result is 0, respectively 1, the states immediately after measurements are:

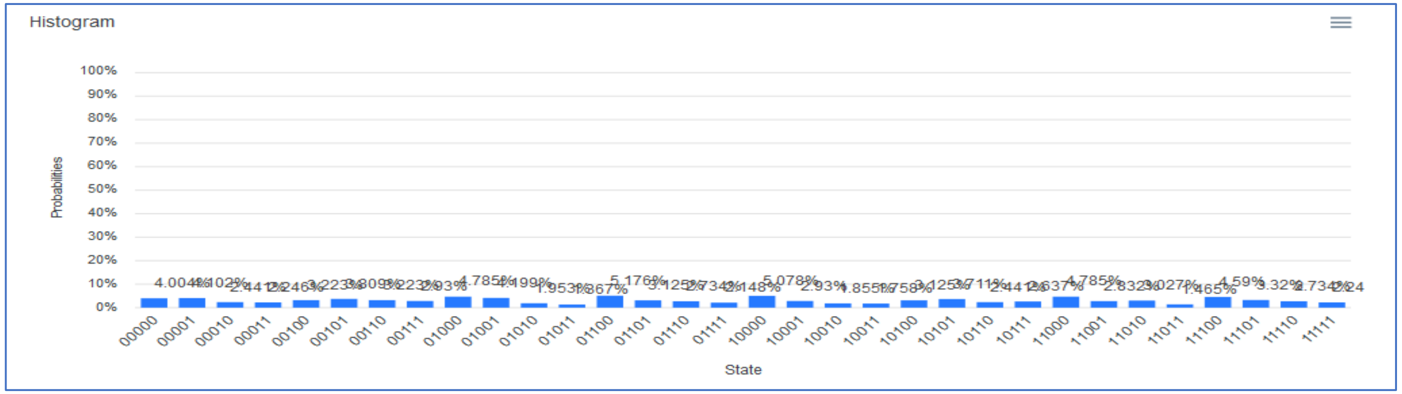
$$\frac{\mathbf{1}}{\sqrt{\mathbf{1/2}}}|\mathbf{0} > \left(\frac{|\mathbf{0} >}{\sqrt{\mathbf{2}}}\right), \qquad \frac{\mathbf{1}}{\sqrt{\mathbf{1/2}}}|\mathbf{1} > \left(\frac{|\mathbf{1} >}{\sqrt{\mathbf{2}}}\right)$$

Measuring one qubit is instantaneously (?) affecting the other qubit too.

Application

A simple random number generator implemented on IBM Q Experience: https://quantum-computing.ibm.com/





Quantum Entanglement

2-qubit register => ***C*[4]**

Consider the following unit vector:

$$|\Psi> = \frac{|0,0> + |1,1>}{\sqrt{2}}$$

Which is the state of each qubit in this case?

$$|\Psi> = (a|0> + b|1>) \otimes (c|0> + d|1>) = \begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c \\ d \end{bmatrix}$$ (Kronecker product)

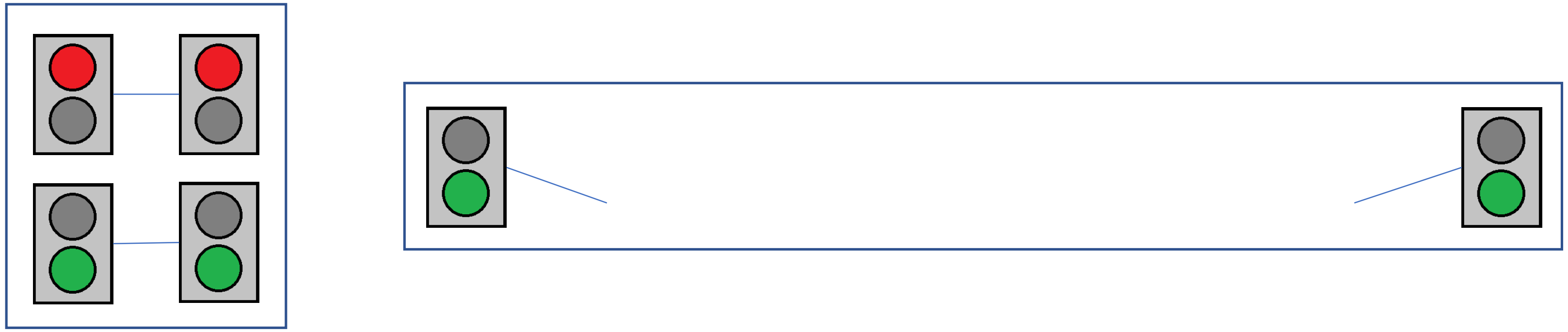
But, the corresponding system of equations **has no solution**! $\begin{bmatrix} ac \\ ad \\ bc \\ bd \end{bmatrix} = \begin{bmatrix} \sqrt{2} \\ 0 \\ 0 \\ \sqrt{2} \end{bmatrix}$

The quantum state of the composite system as a whole, but we cannot attribute the state of the parts.

A single qubit can be in a superposed state, but it cannot be entangled, as its state is not composed of subsystems.

Quantum Entanglement - Analogy

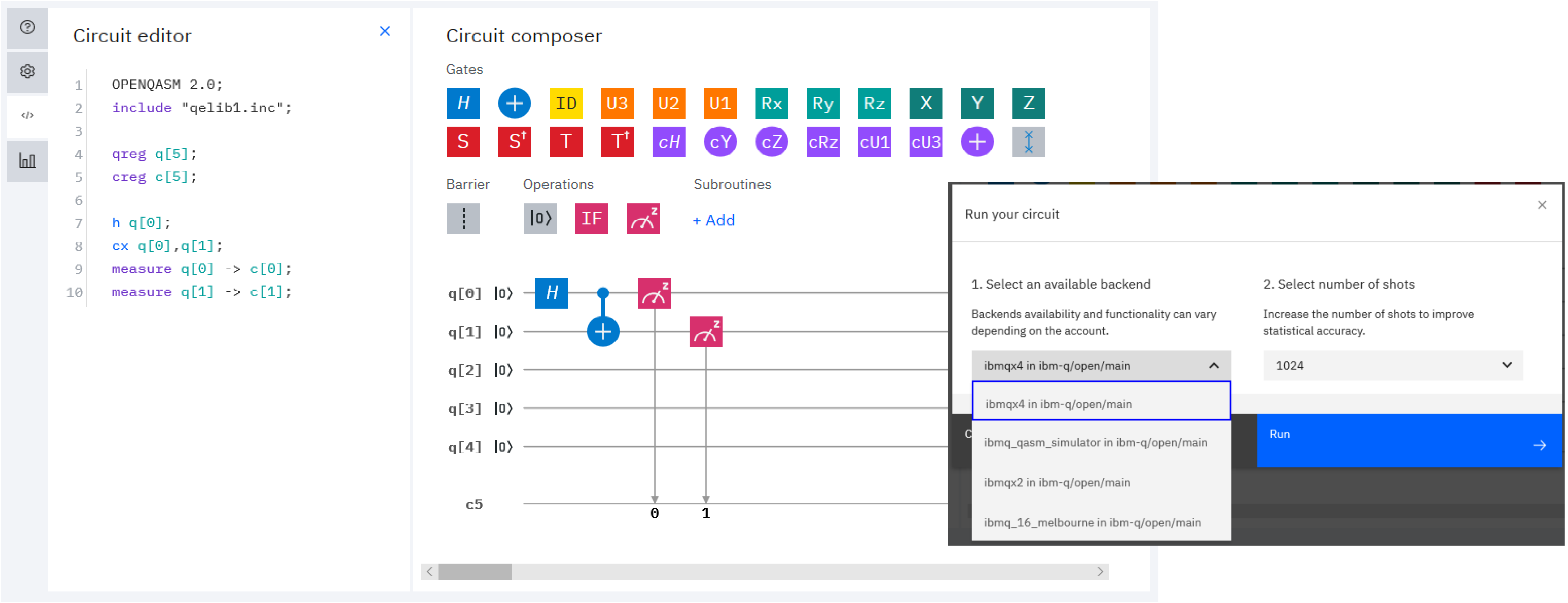
- In semaphores analogy, one could consider two **synchronized** semaphores with: $r_1 = g_1 = 0.5$, *and* $r_2 = g_2 = 0.5$
- As the two semaphores are not independent, it is not possible any more to obtain the compound states probabilities by multiplying individual probabilities (Kronecker product).
- Another example: at each moment of time, tossing two **tied** coins.
- As result of existing bindings, stopping one semaphore results in “freezing” the other one in the same light. What looks quite strange in quantum mechanics is that this should happen instantaneously, as if the measurement effect is “propagating” with an infinity velocity. This was one of the famous Bohr-Einstein debate subjects. I would suggest also that Bohr’s interpretation might be applicable with a space/time limitation (compromise between the two points of view). Thus, after an enough long time, the system might equally probable “fall” (the “constraint thread” breaks) into only one of its compound states (either both semaphores red, or both green).



Quantum Entanglement – IBM Q Experience

Implementation on IBM Q Experience: https://quantum-computing.ibm.com/

1. **Hadamard:** applied on first qubit: q[0] => $|\Psi> = \frac{|0,0> + |1,0>}{\sqrt{2}}$
2. **Conditional not:** applied on qubits q[0], q[1] (change the second qubit only when the first one is 1) => $|\Psi> = \frac{|0,0> + |1,1>}{\sqrt{2}}$
3. **Partial measurement:** the results are stored in the classical register c[0], c[1]



Quantum Algorithm Example [1,5,6]

Quantum Algorithm Example

Search algorithm :

f is an unknown function (e.g. provided by a developer in a dll):

$$f:\{0,\dots,2^n-1\},\qquad N=2^n,\qquad f(x)=\begin{cases}1 & x=x_0\\ 0 & otherwise\end{cases}$$

Classical sequential search O(N):

```
for x = 0 to N-1
        if f(x) = 1 then
                print x
                stop
        end if
end for
```

Quantum Algorithm Example

Grover's algorithm [1,5-6] $\boldsymbol{O}(\sqrt{\boldsymbol{N}})$:

- Initialize the register to the **diagonal state**:

$$|D> = \sum_{i=0}^{N-1} \frac{1}{\sqrt{N}} |i>, \qquad \sum_{i=0}^{N-1} \left(\frac{1}{\sqrt{N}}\right)^2 = N\frac{1}{N} = 1$$

- Apply ***U=R_DR_f*** *for* ***t* times**, where $t = \frac{\pi}{4}\sqrt{N}$:

R_f -> we are provided with access to a **black-box subroutine**:

$$R_f|x> = \begin{cases} -|x> & for\ x = x_0, f(x) = 1 \\ |x> & for\ x \neq x_0, f(x) = 0 \end{cases}$$

R_D **-> Grover's diffusion operator**:

$$R_D = 2|D><D| - I_N = 2\begin{bmatrix} \frac{1}{\sqrt{N}} \\ \vdots \\ \frac{1}{\sqrt{N}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{N}} & \cdots & \frac{1}{\sqrt{N}} \end{bmatrix} - I_N = 2\begin{bmatrix} 1/n & \cdots & 1/n \\ \vdots & \ddots & \vdots \\ 1/n & \cdots & 1/n \end{bmatrix} - \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}$$

- Measure the register in computational basis. The result is **x_0, with probability** $\boldsymbol{1 - \frac{1}{N}}$

Quantum Algorithm Example

Key points for understanding Grover's algorithm:

- One solution for overcoming the probabilistic nature of quantum computing could be **to check the result validity** (if this does not involve significant costs). For instance, it is easy to check a value is satisfying a polynomial equation.
- Another solution is to **run the program multiple times and apply statistics**.
- For a concrete implementation on **IBM's 5-qubit computer [7,8]**, consult reference [5]
- $\boldsymbol{R_f}$ is, in fact:

$$R_f = I_N - 2|x_0 >< x_0|$$

$$R_f|x >= |x > -2|x_0 >< x_0|x >= |x >, \; x \neq x_0$$

$$R_f|x_0 >= |x_0 > -2|x_0 >< x_0|x_0 >= -|x_0 >$$

- The action of R_D to an arbitrary state results in flipping the amplitude of each state about the mean amplitude

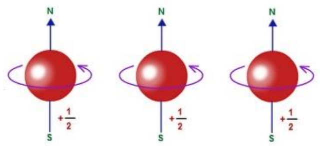
$$R_D \sum_i a_i|i >= \left(2\begin{bmatrix} \frac{1}{n} & \cdots & \frac{1}{n} \\ \vdots & \ddots & \vdots \\ \frac{1}{n} & \cdots & \frac{1}{n} \end{bmatrix} - \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} \right) \begin{bmatrix} a_0 \\ \vdots \\ a_1 \end{bmatrix} = \begin{bmatrix} 2 < a > -a_0 \\ \vdots \\ 2 < a > -a_{n-1} \end{bmatrix}; \;\; < a >= \frac{\sum_i a_i}{n}$$

- *Rf* => the amplitude of $|x_0>$ becomes below <a>, while all other states above <a>.
- R_D => the amplitude of $|x_0>$ becomes above the mean, while all other states below
- **Applying *U* repeatedly will increase the amplitude of $|x_0>$**

Quantum Algorithm Example

```csharp
// C# .Net Simulation
public static void Test()
{
    var register = Grover(3, 5);
}

// Grover's Algorithm Test
public static Matrix Grover(int qubits, int x0)
{
    var n = (int)Math.Pow(2, qubits);
    var oracle = MatrixFactory.NewGroverOracleMatrix(n, x0);
    var grover = MatrixFactory.NewGroverDiffusionMatrix(n);
    var register = MatrixFactory.NewDiagonalState(n);
    var steps = (int)((Math.PI / 4.0) * Math.Sqrt(n));
    for(int step = 1; step <= steps; step++)
    {
        register = oracle * register;
        register = grover * register;
    }
    return register;
}
```



$$R_f = Oracle = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_D = Grover = \begin{bmatrix} -0.75 & \cdots & 0.25 \\ \vdots & \ddots & \vdots \\ 0.25 & \cdots & -0.75 \end{bmatrix} \qquad Register = |D> = \begin{bmatrix} \frac{1}{\sqrt{8}} \\ \vdots \\ \frac{1}{\sqrt{8}} \end{bmatrix}$$

$$step = 0 \begin{bmatrix} 0.35 \\ 0.35 \\ 0.35 \\ 0.35 \\ 0.35 \\ 0.35 \\ 0.35 \\ 0.35 \end{bmatrix} \quad step = 1 \begin{bmatrix} 0.17 \\ 0.17 \\ 0.17 \\ 0.17 \\ 0.17 \\ 0.88 \\ 0.17 \\ 0.17 \end{bmatrix} \quad step = 2 \begin{bmatrix} -0.08 \\ -0.08 \\ -0.08 \\ -0.08 \\ -0.08 \\ 0.97 \\ -0.08 \\ -0.08 \end{bmatrix} \Rightarrow \begin{aligned} p(0) &= 0.0064 \\ p(1) &= 0.0064 \\ p(2) &= 0.0064 \\ p(3) &= 0.0064 \\ p(4) &= 0.0064 \\ p(5) &= 0.9409 \\ p(6) &= 0.0064 \\ p(7) &= 0.0064 \end{aligned}$$

0.0064 * 7 + 0.9409 ≈ 1 (rounding issues)

References

1. R. Portugal, Quantum Walks and Search Algorithms, Quantum Science and Technology, Springer, New York 2013, DOI 10.1007/978-1-4614-6336-8
2. Cursul de fizica Berkeley, vol. IV, Cuantica, Editura Didactica si Pedagogica, 1983
3. Florin Popescu, Florin Marica, Fizica Atomica – Note de curs, Universitatea din Bucuresti, 1998
4. https://quantummechanics.ucsd.edu/ph130a/130_notes/node68.html
5. Patrick J. Coles et. al., Quantum Algorithm Implementations for Beginners, https://arxiv.org/pdf/1804.03719.pdf
6. https://en.wikipedia.org/wiki/Grover%27s_algorithm
7. https://www.research.ibm.com/ibm-q/
8. https://quantumexperience.ng.bluemix.net/qx/tutorial?sectionId=beginners-guide&page=introduction