SummerSoC 2018

Quantum Computing: Physics, Math, Basics

Frank Leymann

Institute of Architecture of Application Systems (IAAS)
University of Stuttgart

Basics in Quantum Physics

Photoelectric Effekt

- By shining electromagnetic waves (i.e. light) on a surface of material, electrons are emitted
- …even by lowest intensity of the light!
- Dislodgement of electrons and their resulting energy only depends on frequency of the incoming light!
- This energy is an integer multitude of hν (ν: frequency of the light)

Einstein's Light Quantum Hypothesis:
Light itself is made of individual packets or quanta (*photons*).
Their energy is hν, where ν is the frequency of the light

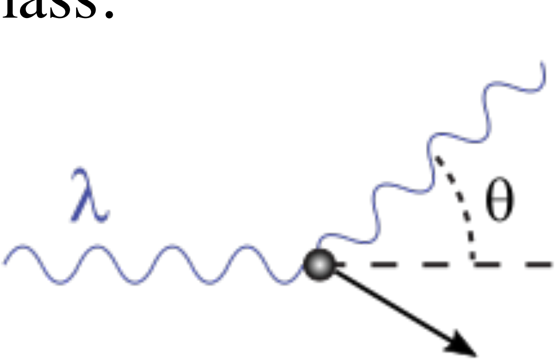
$$\boxed{E = h \cdot \nu}$$

Compton Effect

Light exchanges momentum.

Thus, photons have mass.

$$\boxed{E = m \cdot c^2}$$ from relativity theory



$$\boxed{m = \frac{E}{c^2} = \frac{h\nu}{c^2} \overbrace{=}^{c=\lambda\nu} \frac{h}{c\lambda}}$$

https://upload.wikimedia.org/wikipedia/commons/thumb/e/e3/Compton-scattering.svg/239px-Compton-scattering.svg.png

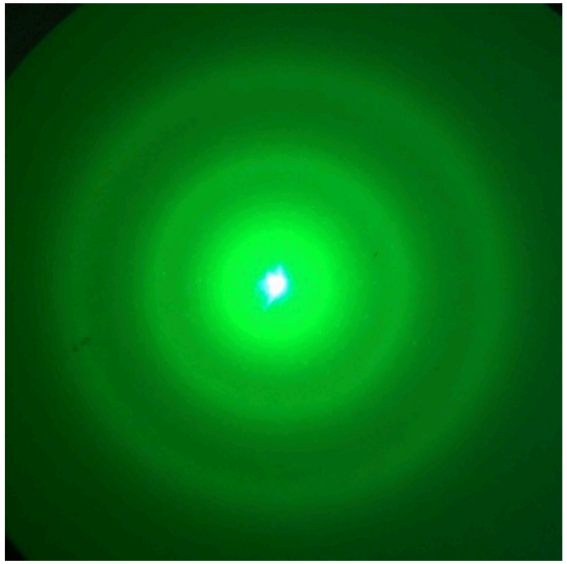
Wave-Particle Duality

In experiments, light behaves sometimes as a wave,
sometimes as a particle

If light is sometimes a wave and sometimes a particle,
can "normal matter" be considered as a wave?

Matter Waves

$$\boxed{m = \frac{h}{c\lambda}} \Rightarrow \boxed{\lambda = \frac{h}{mc}}$$



Interferenzmuster in Form von konzentrischen Kreisen

Diffraction of an electron beam
Verification of de-Broglie-Formula

Heisenberg's Uncertainty Principle

$$\boxed{\Delta x \cdot \Delta p \geq \frac{h}{4\pi}}$$

$$\boxed{\Delta E \cdot \Delta t \geq \frac{h}{4\pi}}$$

$$\begin{aligned} h &= 6{,}626\,070\,040(81) \cdot 10^{-34}\ \mathrm{Js} \\ &= 4{,}135\,667\,662(25) \cdot 10^{-15}\ \mathrm{eVs}, \end{aligned}$$

It is fundamentally not possible to determine the position and the momentum of a particle at the same time

Measuring the momentum of a particle implies a compulsory disturbance of its position and vice versa

It is impossible to prepare a quantum state in which position and momentum are defined with arbitrary precision

⇒ Superposition! (see next)

© Frank Leymann

State

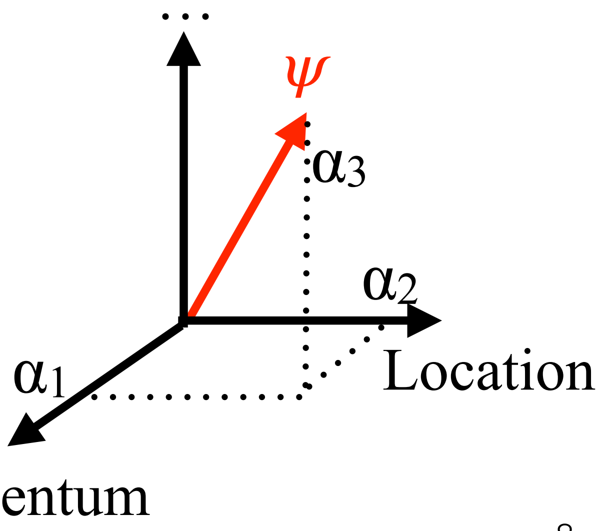
The actual state (e.g. momentum, location,…) of a physical system is a mathematical object

This mathematical object determines for

- each possible measurement of the system, and
- for each possible value of a measurement (momentum, location,…)

the probability that exactly the observed value occurs

<u>More precise</u>: The set of all possible states is a $\mathbb{C}$-Hilbert-Space (*state space*)



Superposition

The overall state ψ of a system is the superposition of individual states φ_i (momentum, location,…)

Mathematically, the state ψ is a linear combination of the individual states φ_i :

$$\psi = \sum_i c_i \varphi_i$$

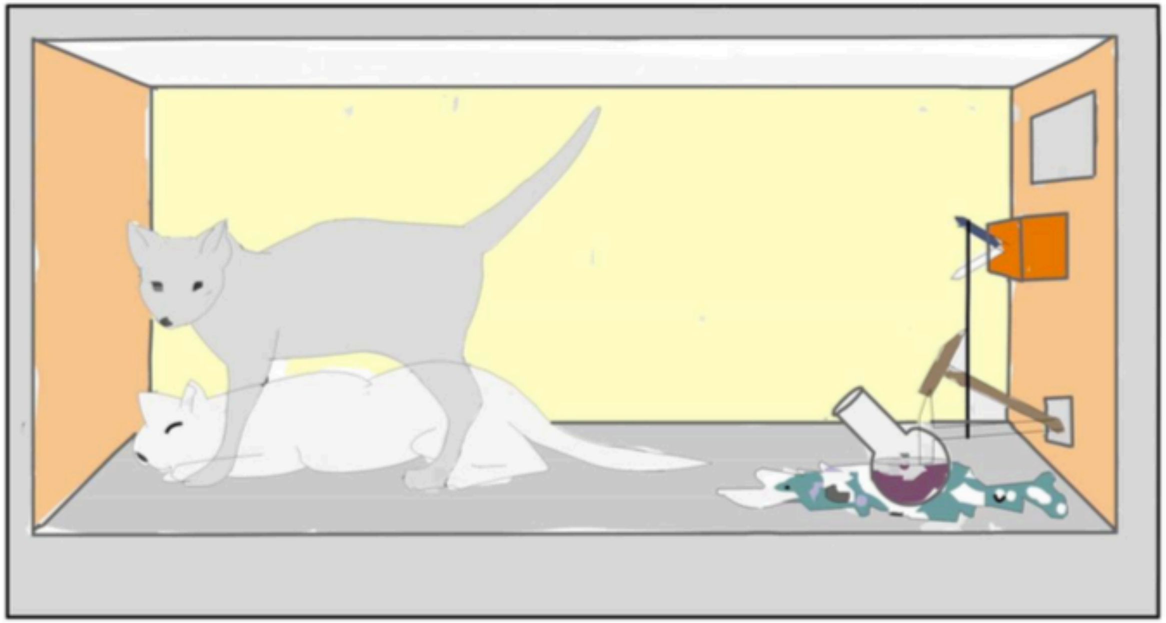
The φ_i's are orthogonal to each other (normed individual states)

The squared modulus $|c_i|^2$ of the amplitude c_i is the probability for measuring the state φ_i in a measurement specialized on this state

Measuring will always reveal a resulting state:

$$\sum_i \left| c_i \right|^2 = 1$$

Schrödinger’s Cat



Measuring delivers the result - it destroys superposition

Postulates of Quantum Mechanics

(von Neumann Postulates)

Every isolated physical system is associated with a *Hilbert-Space* over $\mathbb{C}$, the so-called state space of the system.

An *hermitian operator* H(t) (Hamilton Operator) exists that describes the temporal development of the system and that satisfies the following equation (Schrödinger Equation):

$$\frac{i \cdot h}{2\pi} \cdot \frac{\partial v(t)}{\partial t} = H(t) \cdot v(t)$$

Measuring is described by a hermitian operator *O* (observable) on the state space. Measuring always results in an *eigenvalue* of *O* as a result of the measurement.

According to the *Spectral Theorem:* $O = \sum_{\lambda \in \sigma(O)} \lambda \cdot P_\lambda$

where P_λ is the projection onto the *eigenspace* of *O* associated with the eigenvalue λ.

The state space of a composite system is the *tensor product* of the state spaces of the component systems.

So, We Need Some Math

Hilbert-Space

Hermitian

Operator

Unitary

Eigenspace

Eigenvalue

Spectral Theorem

Tensor Product

Vector Spaces

Example: Coordinate Space

Let $\mathbb{K}$ be a field (e.g. $\mathbb{R}$ or $\mathbb{C}$) and $n \in \mathbb{N}$

$\mathbb{K}^n = \mathbb{K} \times \ldots \times \mathbb{K} = \{(v_1,\ldots,v_n) \mid v_i \in \mathbb{K}\}$

is called *n-dimensional coordinate space* over $\mathbb{K}$

Coordinatewise Addition

$$\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} + \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} v_1 + w_1 \\ v_2 + w_2 \\ \vdots \\ v_n + w_n \end{pmatrix}$$

Coordinatewise Scalar Multiplication

$$a \cdot \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} a \cdot v_1 \\ a \cdot v_2 \\ \vdots \\ a \cdot v_n \end{pmatrix}$$

$(\mathbb{K}^n, +, \cdot)$ is a $\mathbb{K}$-vector space

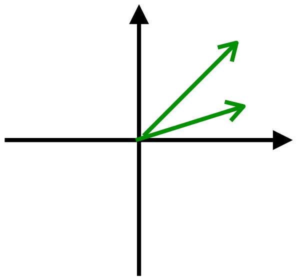
Linear Dependency

$v_1,\ldots,v_n \in V$ and $a_1,\ldots,a_n \in \mathbb{K}$

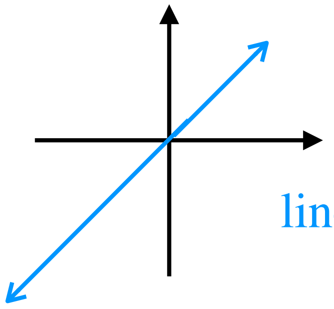
…let $a_1 v_1 + \ldots + a_n v_n = \sum_{i=1}^{n} a_i v_i = 0_V$

If this implies $a_1 = a_2 = \ldots = a_n = 0$, then $v_1,\ldots,v_n$ are called *linear independent*

Otherwise (i.e. $\exists 1 \leq i \leq n : a_i \neq 0$) we call $v_1,\ldots,v_n$ *linear dependent*



linear **<u>in</u>**dependent



linear dependent

Basis: Definition

Definition: A set $\{v_1,\ldots,v_n\} \subseteq V$ is called *basis* of V :⇔

(i) $\{v_1,\ldots,v_n\}$ are linear independent,

(ii) $L(\{v_1,\ldots,v_n\}) = V$

v_i is called *basis vector.*

Theorem: All bases of a vector space V have the same number of vectors.

Definition: The number of vectors of a basis of a vector space V is called *dimension* of V (**dim V**).

© Frank Leymann

Example

$$e_1 := \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, e_2 := \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \ldots, e_n := \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^n (\mathbb{C}^n)$$

is called *standard basis* of $\mathbb{R}^n$ ($\mathbb{C}^n$)

$$\Rightarrow \quad \dim \mathbb{R}^n = n$$

$$\Rightarrow \quad \dim \mathbb{C}^n = n$$

Definition

Let V, W be vector spaces over $\mathbb{K}$

A map $f : V \to W$ is called *lineare map* (or: *operator*) $:\Leftrightarrow$

$\forall x, y \in V \quad \forall a \in \mathbb{K}$:

(i) f(ax) = af(x)

(ii) f(x + y) = f(x) + f(y)

© Frank Leymann

Matrix of a Linear Map

Let $f: V \to W$ be a linear map

Let B={b$_1$,…,b$_n$} be a basis of V

Let E={e$_1$,…,e$_m$} be the standard basis of W

$$M_E^B(f) = \begin{pmatrix} f(b_1) & f(b_2) & \cdots & f(b_n) \end{pmatrix} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$

Take the images of a basis of V and put them as columns in a matrix.
This matrix is called *matrix of f* wrt bases B and E

Eigenvalues

Eigenvalues & Eigenvectors

$f : V \to V$ linear map (*endomorphism*)

$\lambda \in \mathbb{K} \backslash \{0\}$ is called *eigenvalue* of f :⇔

$$\exists v \in V \setminus \{0\} : f(v) = \lambda v$$

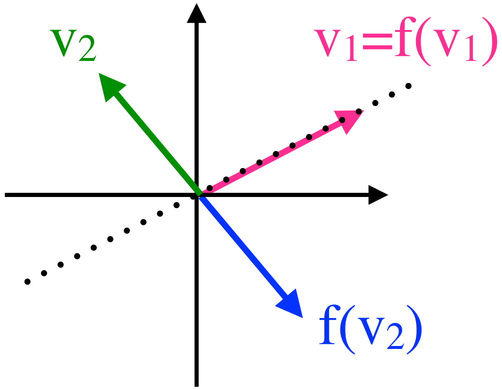
v is called an λ associated *eigenvector* of f

I.e. in the direction of an eigenvector, f is a stretching

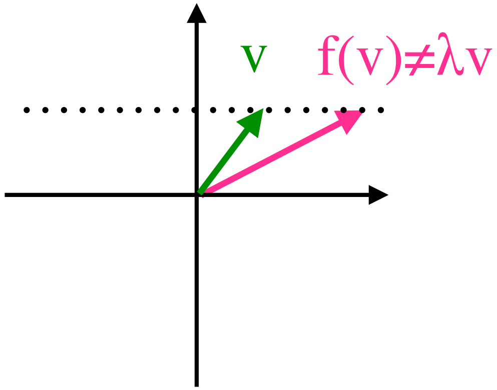
The set E_λ of all eigenvectors associated with the eigenvalue λ is called *eigenspace* of f of eigenwert λ

Measuring means observing eigenvalues

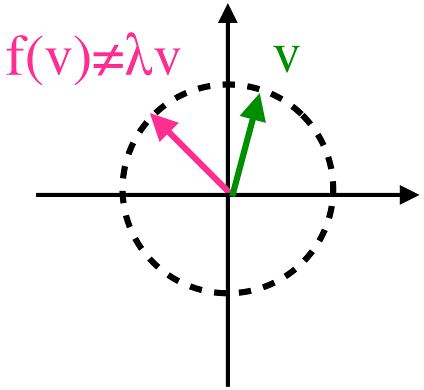
Examples



Reflection



Shear



Rotation, $0<\varphi<\pi$

v_1 and v_2 are eigenvectors with eigenvalues $\lambda_1=1$ and $\lambda_2=-1$

Only vectors $(x,0)$ (with $x\neq0$) are eigenvectors

No eigenvectors

Spectrum

The set of all Eigenvalues of f, denoted by σ(f),
is called *spectrum* of f (and of A=M(f))

In each ℂ-vector space V every map has at least one eigenvalue and the sum of the algebraic multiplicities is dimV.
(is a consequence of the Fundamental Theorem of Algebra)

This is not true for ℝ-vector spaces!

Determining Eigenvectors

Let $A = M(f)$

Let λ be an eigenvalue of f (from the characteristic polynomial)

I.e. there is an x with $f(x)=\lambda x \Leftrightarrow Ax=\lambda x$ $(= \lambda Ex)$

Thus, we have to solve the linear equation system $(A - \lambda E)x = 0$

Measuring transforms the system into an eigenvector (*collapse*)

Pre-Hilbert-Spaces

Inner Product aka Scalar Product

Let V be $\mathbb{K}$-vector space ($\mathbb{K} = \mathbb{C}$ or $\mathbb{R}$).

A *scalar product* (or: *inner product*) on V is a positiv definite, hermitian sesquilinearform:

$$\langle .,. \rangle : V \times V \to \mathbb{C}$$

$$\langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle$$
$$\langle \lambda x, y \rangle = \overline{\lambda} \langle x, y \rangle$$

Semililinear
(in the first argument)

$$\langle x, y \rangle = \overline{\langle y, x \rangle}$$

Hermitian

$$\langle x, x \rangle \geq 0$$
$$\langle x, x \rangle = 0 \Leftrightarrow x = 0$$

Positiv Definite

$$\langle x, y + z \rangle = \langle x, y \rangle + \langle x, z \rangle$$
$$\langle x, \lambda y \rangle = \lambda \langle x, y \rangle$$

Linear
(in the second argument)

Example: $\langle x, y \rangle := \sum_{i=1}^{n} \overline{x}_i y_i = \overline{x}_1 y_1 + \ldots + \overline{x}_n y_n$

A $\mathbb{K}$-vector space with a scalar product is called *Pre-Hilbert-Space*

Normed Vector Space

Let V be a $\mathbb{K}$-vector space ($\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$).

A map $\left\| \cdot \right\| : V \to \mathbb{R}_0^+$ is called *norm* on V :⇔

$\forall\ v, w \in V\ \forall\ \lambda \in \mathbb{K}:$

- $\left\|x\right\| = 0 \Leftrightarrow x = 0$
- $\left\|\lambda x\right\| = \left|\lambda\right| \cdot \left\|x\right\|$
- $\left\|x + y\right\| \leq \left\|x\right\| + \left\|y\right\|$

(V, ll . ll) is called *normed vector space*.

Theorem

A pre-Hilbert-space is a normed vector space.

Proof: $\|x\| := \sqrt{\langle x, x \rangle}$ is a norm on V

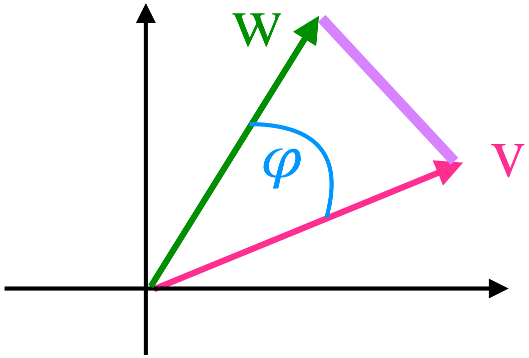
Angle

Let (V, <.,.>) be a pre-Hilbert-space

v≠0≠w . Then $\sphericalangle(v,w)=\varphi:=\arccos\frac{\langle v,w\rangle}{\|v\|\cdot\|w\|}$ is called *angle* between v and w

$$\boxed{v\perp w\Leftrightarrow\langle v,w\rangle=0}$$

v and w are called *orthogonal*



Orthogonality of Eigenspaces

Let (V, <.,.>) be a pre-Hilbert-space

Let λ, $\mu \in \sigma(f)$, $\lambda \neq \mu$, two different eigenvalues of the linear map f.
For $v \in E_\lambda$ and $w \in E_\mu$ it is $v \perp w$.

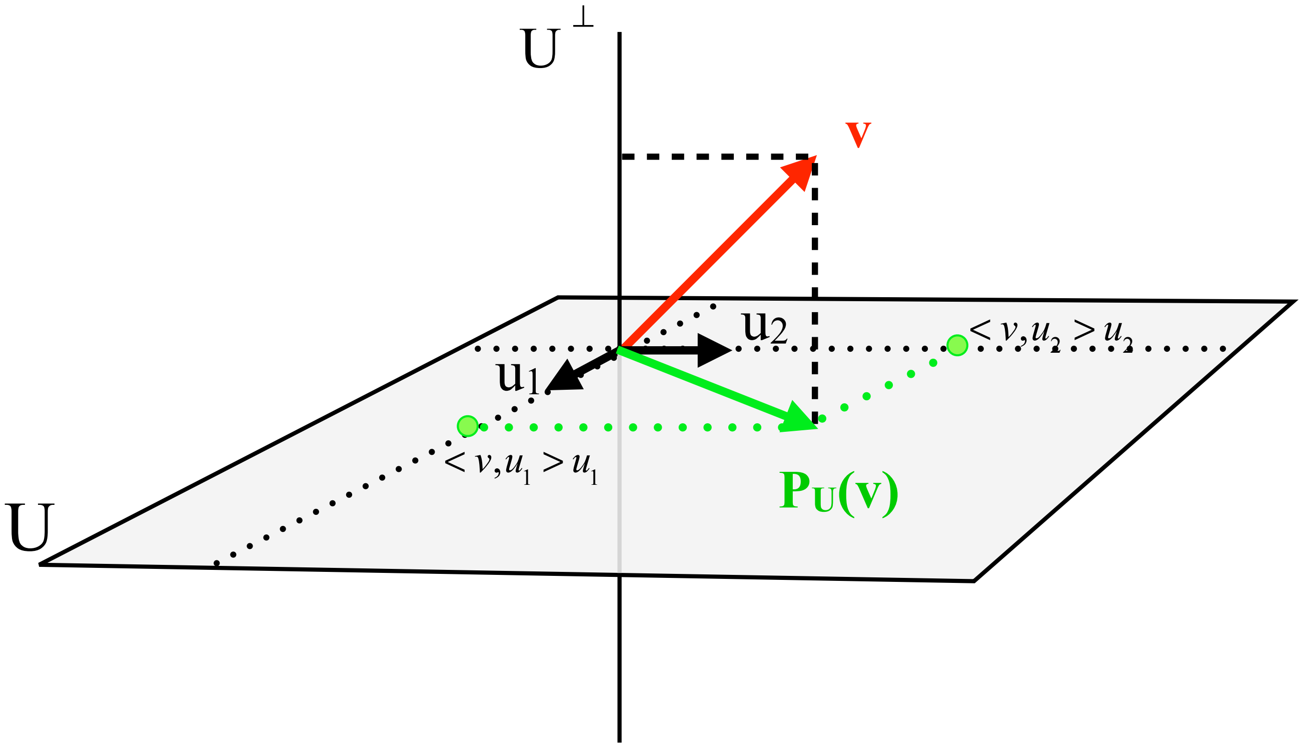
Eigenvectors of different eigenvalues are orthogonal.

Orthogonal Projection

U ⊆ V subspace and $\{u_1, \ldots, u_k\} \subseteq$ U an ON-basis of U.
Define $P_U : V \to U$ as

$$P_U(v) := \sum_{i=1}^{k} < v, u_i > u_i$$

Then, P_U is called *orthogonal projection* onto U.



Self-Adjoint and Hermitian Maps

Let V a pre-Hilbert-space over $\mathbb{K}$, dimV $< \infty$ and f : V → V linear.

f is called *self-adjoint* ($\mathbb{K}=\mathbb{R}$) or *hermitian* ($\mathbb{K}=\mathbb{C}$) :⇔

$$\forall u,v \in V: \; < f(u),v > \; = \; < u, f(v) >$$

Let f be hermitian $\Rightarrow \quad \forall \lambda \in \sigma(f): \; \lambda \in \mathbb{R}$
(all eigenvalues of hermitian maps are real numbers)

We always measure values that are real numbers

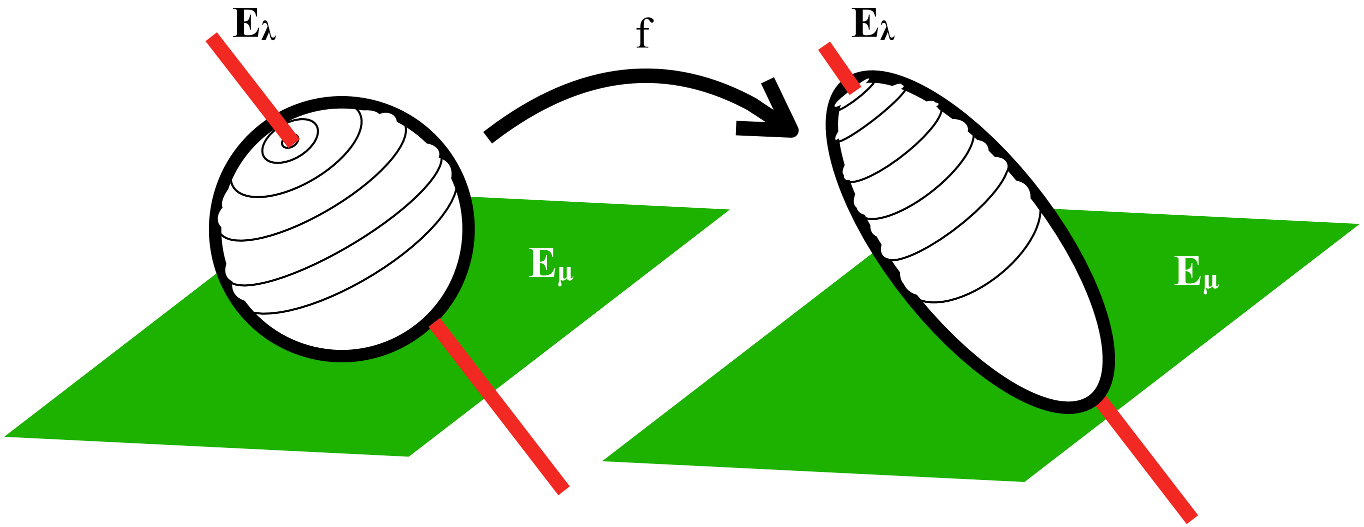
Spectral Theorem

Let f : V → V be self-adjoint or hermitian ⇒

$$f = \sum_{\lambda \in \sigma(f)} \lambda \cdot P_{E_\lambda}$$

Intuition:

1. if $\lambda, \mu \in \sigma(f)$, $\lambda \neq \mu$, then E_λ and E_μ are orthogonal.
2. f acts on E_λ by stretching ($\lambda > 1$) oder compression ($0 < \lambda < 1$)



Measurement is projection onto an eigenspace, result is corresponding eigenvalue

Orthogonal & Unitary Maps

Let V, W be two pre-Hilbert spaces over $\mathbb{K}$ and f : V → W.

f is *orthogonal* ($\mathbb{K}=\mathbb{R}$) or *unitary* ($\mathbb{K}=\mathbb{C}$) :⇔

$$\forall\ u,v \in V : \left\langle f(u), f(v) \right\rangle_W = \left\langle u,v \right\rangle_V \qquad \text{(}\textit{angle preserving}\text{)}$$

Some key properties:

① ker(f)={0}, f is injective Especially: f is invertible!

② || f(v) || = || v || (*length preserving*)

Quantum algorithm use unitary computation steps only,
i.e. each such step is reversible!
(in contrast to classical algorithms)

The Qbit

Dirac Notation

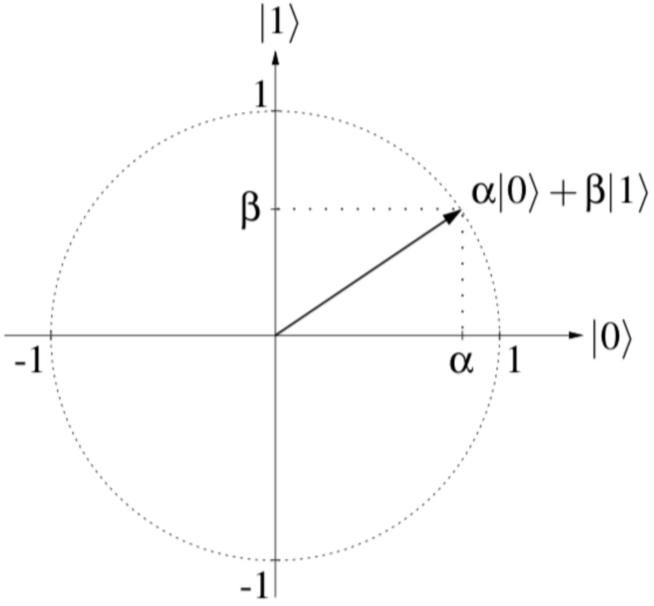
A state y is denoted by $\left|y\right\rangle$, so-called. *ket-notation*.
There is also a *bra-Notation*: $\left\langle y\right|$
See braket <.|.> 😆
Introduced by Paul Dirac.

Quantum bit (***Qbit***) is in the two classical states $\left|0\right\rangle$ or $\left|1\right\rangle$ at the same time (!): ***Superposition***

State of a qbit is

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

ie a linear combination of $\left|0\right\rangle$ and $\left|1\right\rangle$
$\alpha, \beta \in \mathbb{C}$ and $\mathbf{|\alpha|^2 + |\beta|^2 = 1}$.
Ie a quantum state is a vector on the unit circle S^1.
$\{\left|0\right\rangle,\left|1\right\rangle\}$ is a basis of the state space



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

Measurement

Classical bits can be read
⇒ You can find out <u>the **exact** state</u> (value 0 or 1) of the bit

Can't be done for qbits, their state is the superposition $\alpha|0\rangle + \beta|1\rangle$

Reading a qbit means measurement, and measuring **destroys** superposition!

<u>Corollary</u>: A qbit can be read only once.

Measuring $|x\rangle = \alpha|0\rangle + \beta|1\rangle$ destroys superposition and results in

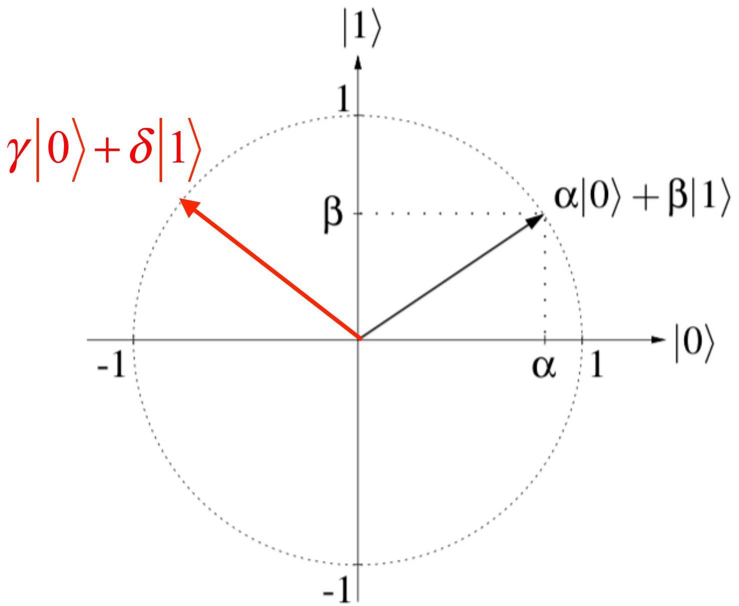
state $|0\rangle$ with probability $|\alpha|^2$

state $|1\rangle$ with probability $|\beta|^2$

Single Computation Steps

A *computation step* creates from a state (a vector from S^1) a new state (again a vector from S^1).

A computation step is a linear map from S^1 to S^1, thus, a unitary map.

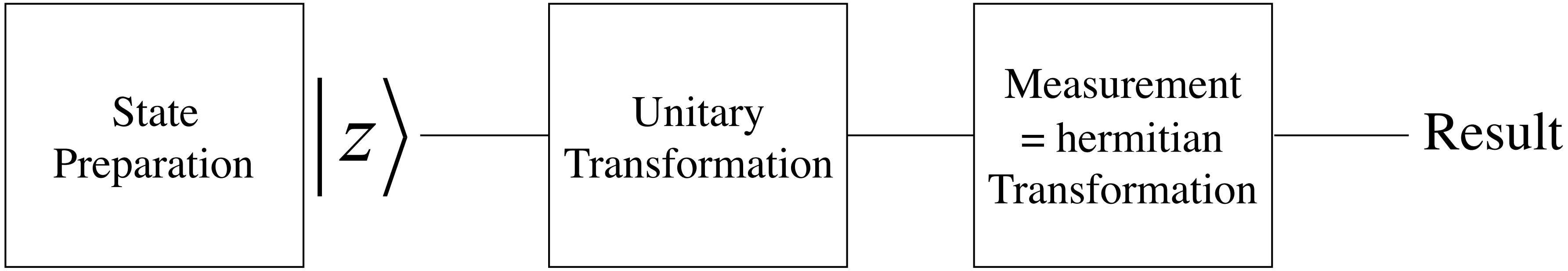


$|1\rangle$
1
$\gamma|0\rangle+\delta|1\rangle$
β
$\alpha|0\rangle+\beta|1\rangle$
-1
α
1
$|0\rangle$
-1

A computation step is represented by a unitary linear map

© Frank Leymann

Principle of a Quantum Algorithm



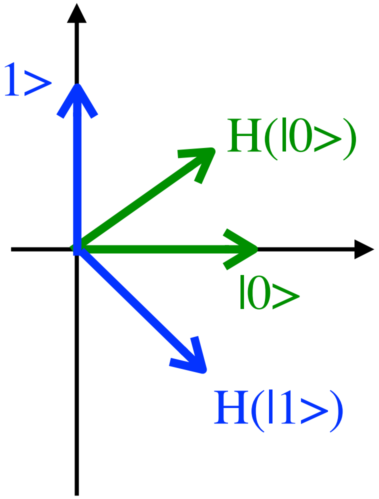
© Frank Leymann

Example: Coin Flipping

We want an algorithm, that results in $|0\rangle$ with probability 1/2,

and that results in $|1\rangle$ with probability 1/2

1. $|x\rangle \leftarrow |0\rangle$
2. $|x\rangle \leftarrow H|x\rangle$
3. Measure $|x\rangle$



Step 1: Qbit $|x\rangle$ is initialized in state $|0\rangle$

Step 2: Hadamard transformation H is applied to $|x\rangle$
thus, $|x\rangle$ transitions into state $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$

Step 3: Measuring gives the desired result

The algorithm produces a completely random bit, i.e. a random number:
Classical algorithms can only produce pseudo random numbers!

Quantum Register

Quantum Register: Informally

Quantum ***register*** is a series of n qbits

Classical register is a series of n bits

Quantum register with n qbits is the superposition of the corresponding 2^n states

$$|00...00\rangle,\ |00...01\rangle,\ |00...10\rangle,...,|11...11\rangle$$

Classical register with n bit → 1 value at a time

Quantum register with n bit → 2^n value at the **same** time

Quantum computer manipulates 2^n values at the same time (*Quantum Parallelism*)

2-Qbit Quantum Register: Formally

$$R = |x_1\rangle |x_0\rangle$$

This is a product!
("tensor product")

$$|x_0\rangle = \gamma_0 |0\rangle + \gamma_1 |1\rangle$$

$$|x_1\rangle = \beta_0 |0\rangle + \beta_1 |1\rangle$$

$$\begin{aligned} R &= |x_1\rangle \cdot |x_0\rangle \\ &= \left(\beta_0 |0\rangle + \beta_1 |1\rangle\right) \cdot \left(\gamma_0 |0\rangle + \gamma_1 |1\rangle\right) \\ &= \beta_0\gamma_0 |0\rangle|0\rangle + \beta_0\gamma_1 |0\rangle|1\rangle + \beta_1\gamma_0 |1\rangle|0\rangle + \beta_1\gamma_1 |1\rangle|1\rangle \end{aligned}$$

With $\alpha_{ij} = \beta_i \gamma_j$

$$\begin{aligned} R &= \alpha_{00} |0\rangle|0\rangle + \alpha_{01} |0\rangle|1\rangle + \alpha_{10} |1\rangle|0\rangle + \alpha_{11} |1\rangle|1\rangle \\ &= \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle \end{aligned}$$

Tensor Product of Vector Spaces

V and W vector spaces over $\mathbb{K}$ with basis $\{v_1,\ldots,v_n\}$ and $\{w_1,\ldots,w_m\}$

The *Tensor Produkt* $V \otimes W$ is an $(n \cdot m)$ dimensional vector space with basis $\{v_i \otimes w_j \mid 1 \leq i \leq n \text{ and } 1 \leq j \leq m\}$

$$\left\{ \begin{array}{cccc} v_1 \otimes w_1, & v_1 \otimes w_2, & \ldots, & v_1 \otimes w_m, \\ v_2 \otimes w_1, & v_2 \otimes w_2, & \ldots, & v_2 \otimes w_m, \\ \vdots & \vdots & \vdots & \vdots \\ v_n \otimes w_1, & v_n \otimes w_2, & \ldots, & v_n \otimes w_m \end{array} \right\}$$

Example

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} \otimes \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} \alpha_1\beta_1 \\ \alpha_1\beta_2 \\ \alpha_2\beta_1 \\ \alpha_2\beta_2 \end{pmatrix}$$

© Frank Leymann

2-Qbit Quantum Register as Tensor Produkt

Let $\{|0\rangle,|1\rangle\}$ be basis for the space of qbits $|x\rangle=(\alpha_0|0\rangle+\alpha_1|1\rangle)$ and $|y\rangle=(\beta_0|0\rangle+\beta_1|1\rangle)$

This space is denoted as $_2\mathbf{H}$ - it's a Hilbert-Space

The basis for the quantum register $R=|x\rangle\otimes|y\rangle$ is the set:

$$\{|0\rangle\otimes|0\rangle,\ |0\rangle\otimes|1\rangle,\ |1\rangle\otimes|0\rangle,\ |1\rangle\otimes|1\rangle\}$$

Simplified notation $\{|0\rangle|0\rangle,\ |0\rangle|1\rangle,\ |1\rangle|0\rangle,\ |1\rangle|1\rangle\}$

Yet even simpler $\{|00\rangle,\ |01\rangle,\ |10\rangle,\ |11\rangle\}$

Example: $|10\rangle=|1\rangle\otimes|0\rangle=\begin{pmatrix}0\\1\end{pmatrix}\otimes\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}0\\0\\1\\0\end{pmatrix}$

…In Summary:

$|00\rangle, |01\rangle, |10\rangle, |11\rangle$ is a basis of $\mathbb{C}^4 = {}_2H \otimes {}_2H$

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Quantum Register

Let ${}_2\mathrm{H}$ be the $\mathbb{C}$-vector space spanned by $\{|0\rangle, |1\rangle\}$

Then, $|\phi\rangle \in {}_2H^{\otimes n} := \underbrace{{}_2H \otimes \cdots \otimes {}_2H}_{n-times}$ with $\| \, |\phi\rangle \, \| = 1$ is called *state* of the n-qbit-*quantum register* $|x_{n-1}\rangle \otimes \cdots \otimes |x_0\rangle$

$$\mathbb{C}^{2^n} = {}_2H^{\otimes n}$$

Separable & Entangled States

$$\left|\phi\right\rangle \in H_1 \otimes \cdots \otimes H_n \text{ is called } \textit{separable} :\Leftrightarrow$$

$$\left|\phi\right\rangle = \left|\psi_1\right\rangle \otimes \cdots \otimes \left|\psi_n\right\rangle \text{ with } \left|\psi_i\right\rangle \in H_i,\ 1 \leq \mathrm{i} \leq \mathrm{n}$$

$$\left|\phi\right\rangle \text{ is called } \textit{entangled} :\Leftrightarrow \left|\phi\right\rangle \text{ is not separable}$$

Separable States

A state of a quantum register is called *separable*,
it it can be expressed as tensor product of the individual qbits

Example:

$$\frac{1}{2}\left(|00\rangle+|01\rangle+|10\rangle+|11\rangle\right)=\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)\otimes\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)$$

Entanglement

A state that is not separable is called *entangled*

$$\frac{1}{\sqrt{2}}\left(|00\rangle + |01\rangle\right)$$

Measuring the first qbit results in $|0\rangle$ with probability 1.
The second qbit will be measured as $|0\rangle$ or $|1\rangle$ with probability 1/2

$$\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$$

Measuring the first qbit results in $|0\rangle$ or $|1\rangle$ with equal probability.
After that the value of the second qbit is already determined!

Einstein–Podolsky–Rosen Paradox
(EPR Paradox)

Entanglement is a phenomenon unique to Quantum Computing!

Every computation that is not concerning entangled qbits,
can be performed with the same efficiency with classical computations.

Operators on Quantum Registers

1-Qbit Operators

A unitary map f : ${}_2H \rightarrow {}_2H$ is called *1-qbit operator* (…*Gate*)

Quantum NOT, Bit Flip

Phase Flip

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

X, Y, Z are called *Pauli-Matrices*

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \qquad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$$

Hadamard Matrix

Phase Matrix

π/8 Matrix

(yeah, strange: π/8 vs π/4; pure historical reasons!)

© Frank Leymann

1-Qbit Operators: Decomposition

A set $\mathcal{U}$ of 1-qbit operators is called *universal* :⇔
Each 1-qbit operator is a finite combination of operators from $\mathcal{U}$

Let U be a 1-bit operator. Then:

The set of Pauli-Operators are universal for 1-qbit operators

© Frank Leymann

Operators on Quantum Registers

Let n>1, ${}_2H^{\otimes n} = {}_2H \otimes \cdots \otimes {}_2H$

A unitary map f : ${}_2H^{\otimes n} \rightarrow {}_2H^{\otimes n}$ is called
n-qbit operator (or *quantum-register-operator* oder *quantum gate*)

A set $\mathcal{U}$ of quantum-register-operators is called *universal* :⇔
Every quantum-register-operator is a finite combination of operators from $\mathcal{U}$

Two-Level Operators

Let $f : V \rightarrow V$ be a unitary map

$$\text{f is called } \textit{two-level} :\Leftrightarrow \exists\, \text{U} \in \mathbb{C}^{2\times 2} : \text{M(f)} = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & U & & & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{pmatrix}$$

and U is unitary

A two-level map modifies at most two components of a vector
(i.e. two qbits of a quantum register)

Decomposition into Two-Level Operators

Let $f : V \to V$ be unitary, dim V = n.

Then, M(f) can be represented as product of r two-level matrices.

It is: $n - 1 \leq r \leq n \cdot (n - 1)/2$

The set of two-level operators on quantum registers is universal.

Problem: There is an infinite number of two-level operators. But the set of universal operators should be "small"!

CNOT
(Controlled Not)

$$\text{CNOT}: {}_2H \otimes {}_2H \to {}_2H \otimes {}_2H$$
$$\left|x,y\right\rangle \mapsto \left|x,x \oplus y\right\rangle$$

⊕	0	1
0	0	1
1	1	0

$\oplus : \{0,1\} \to \{0,1\}$ with $x \oplus y \mapsto x+y \bmod 2$

I.e. if x=1 then y will be negated; otherwise, y is not changed at all
(x is called *control*-qbit, y is called *target*-qbit)

CNOT is unitary

Decomposition into CNOT and 1-Qbit Operators

Let $f: V \rightarrow V$ be unitary, dim V = d.

M(f) can be represented as a product of 1-qbit operators and CNOT operators.

The set of 1-qbit Operators and CNOT is universal.

Reminder: The Pauli-Matrices is a set of universal 1-qbit operators

For an n qbit quantum register it is $d=2^n$.
The number of required 1-qbit operators and CNOTs is $O(n^2 \cdot 4^n)$

Problem: This is not an efficient implementation of quantum register operators

Approximation

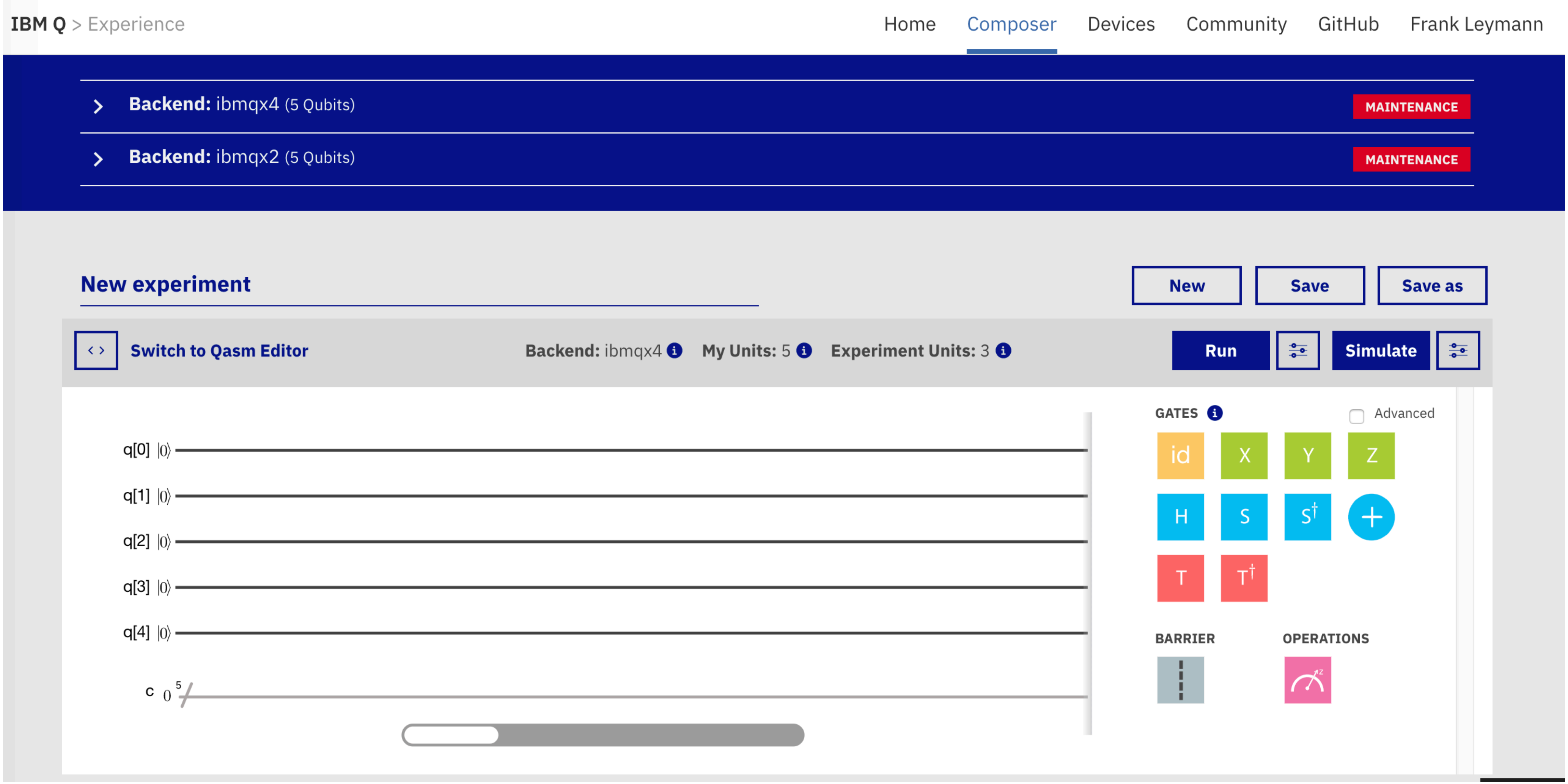
Using {Hadamard, Phase, CNOT, $\pi/8$},
each operator U on a quantum register
can be approximated with arbitrary precision.

Solovay-Kitaev Theorem

Here, *approximation with precision* ε means,
that the probabilities of measurements of results of U deviates at most by ε
from measurements of the results of the composition.
("The statistics of measurements don't really differ.")

Assumption: This implementation is acceptable.

Tools



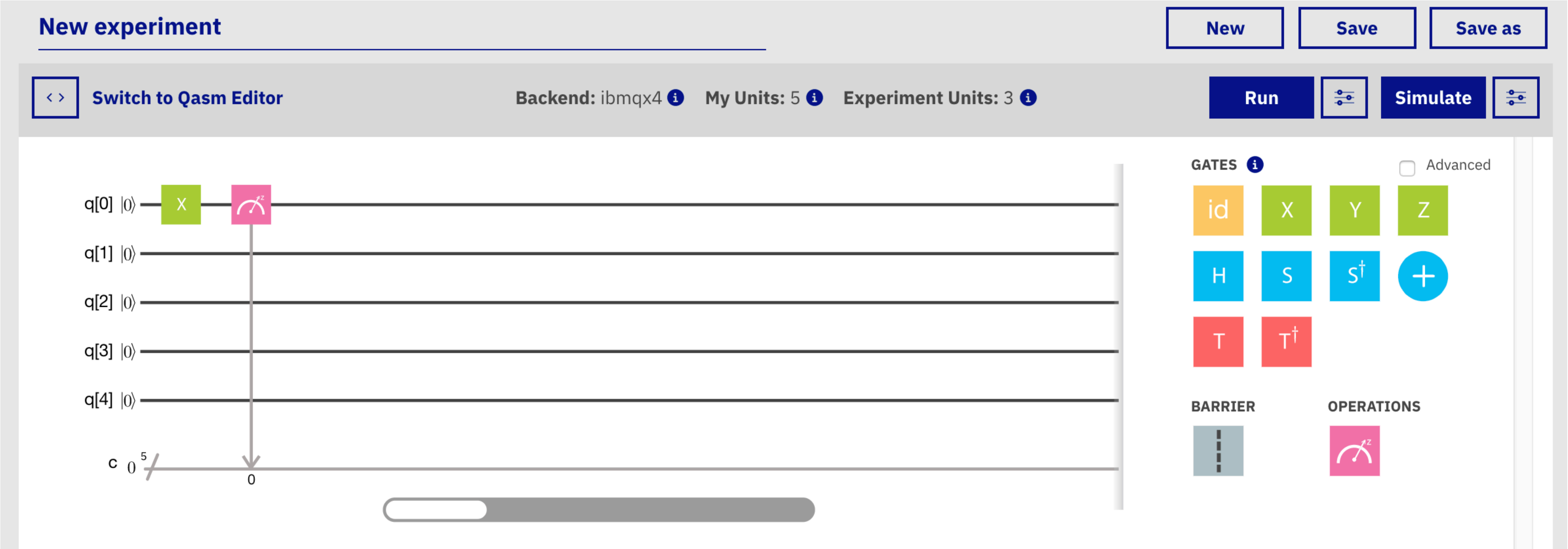
Backend: ibmqx4 (5 Qubits) MAINTENANCE

Backend: ibmqx2 (5 Qubits) MAINTENANCE

Backend: ibmqx4

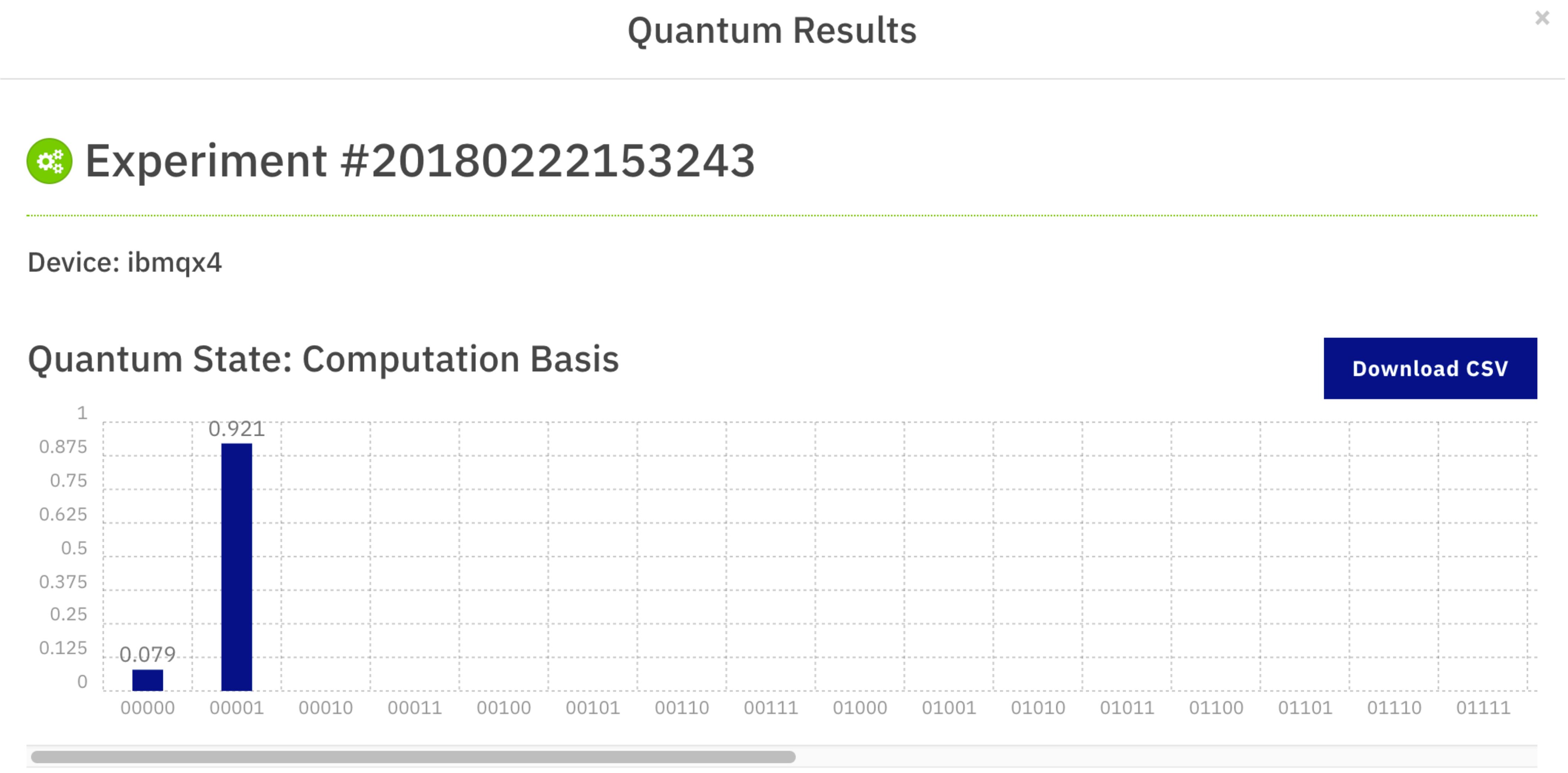
My Units: 5

Experiment Units: 3



New experiment

Backend: ibmqx4 My Units: 5 Experiment Units: 3



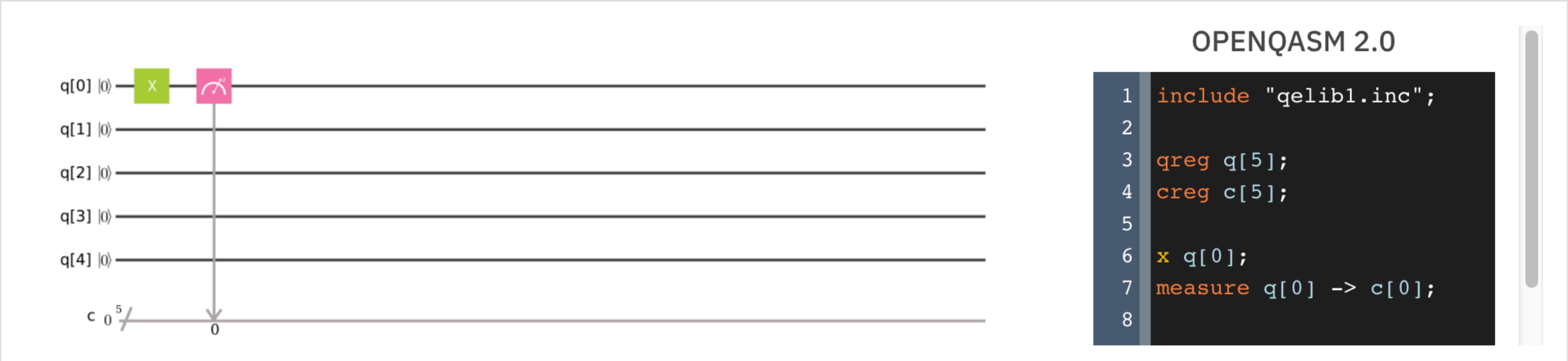
Quantum Results

Experiment #20180222153243

Device: ibmqx4

Quantum State: Computation Basis

Quantum Circuit



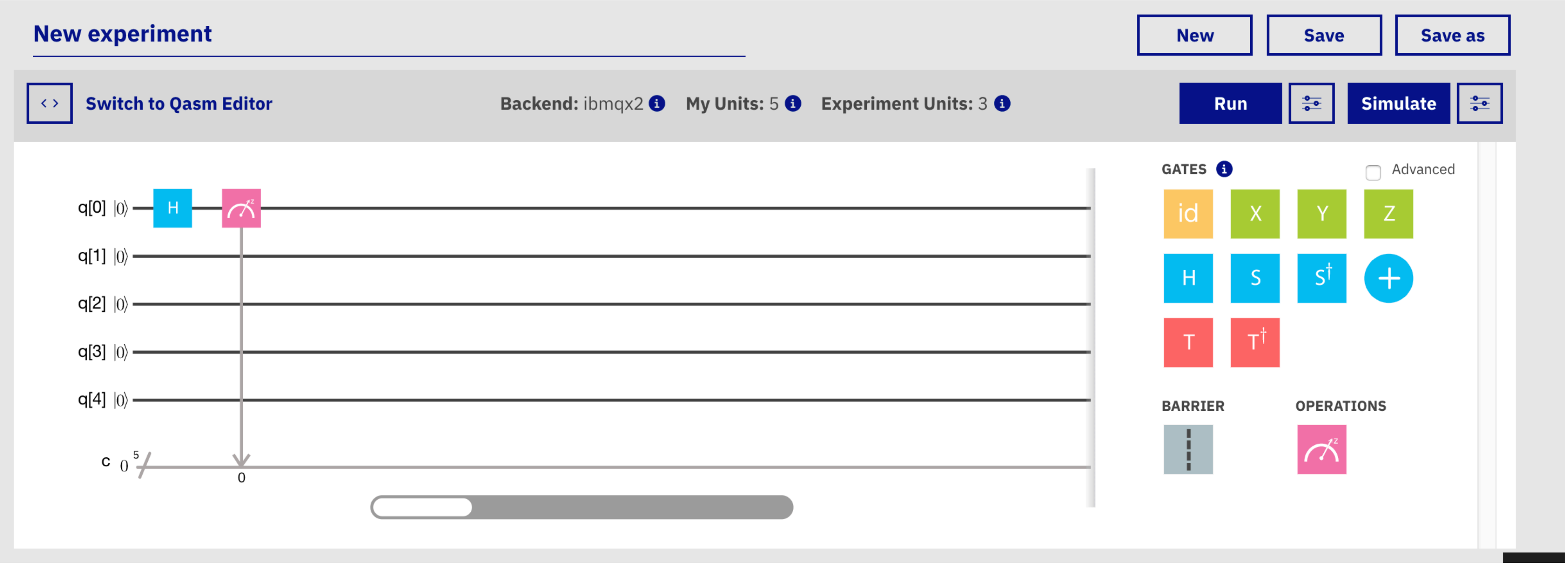
OPENQASM 2.0

```
include "qelib1.inc";

qreg q[5];
creg c[5];

x q[0];
measure q[0] -> c[0];

```

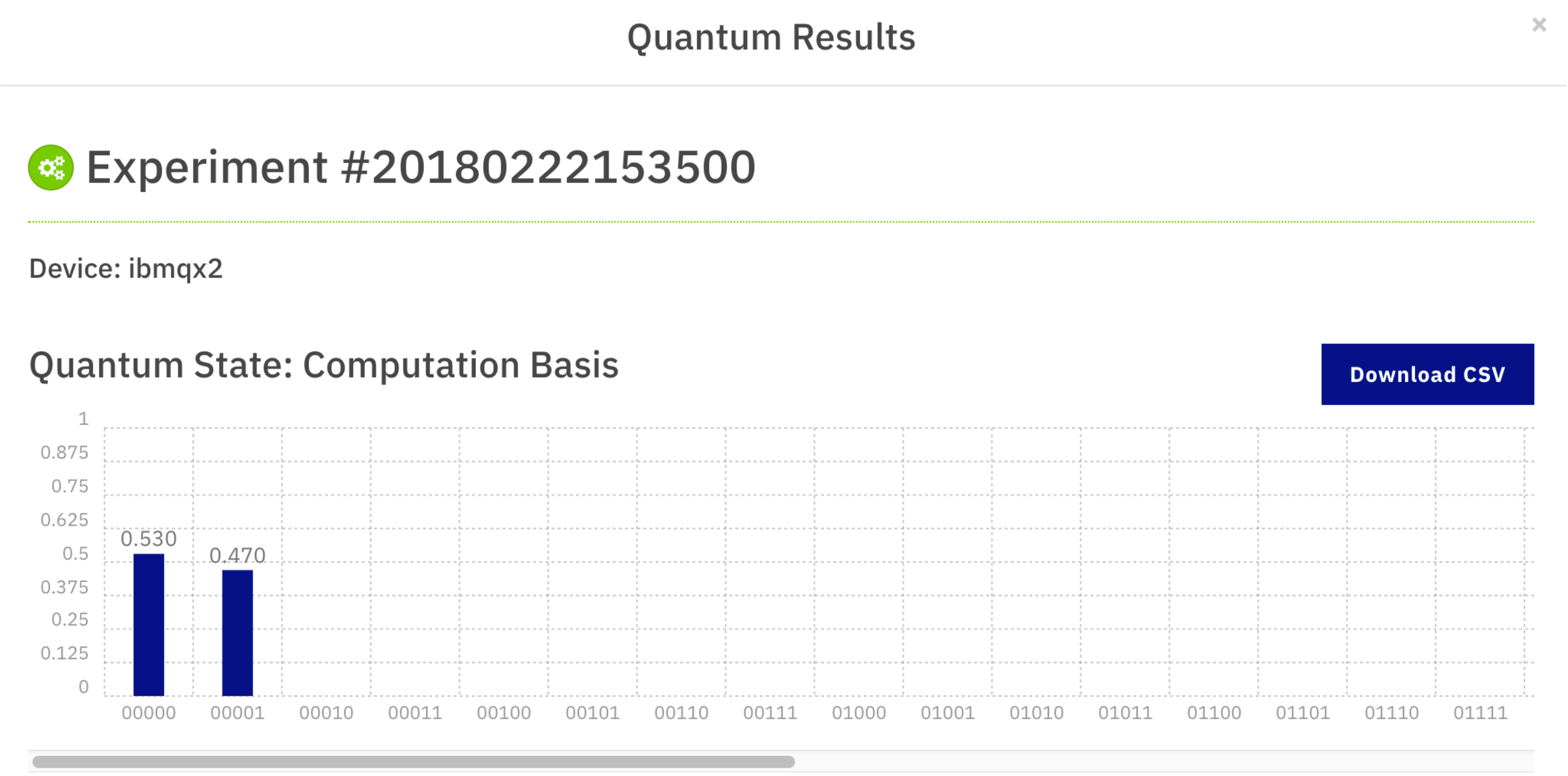


Backend: ibmqx2

My Units: 5

Experiment Units: 3

□ Advanced



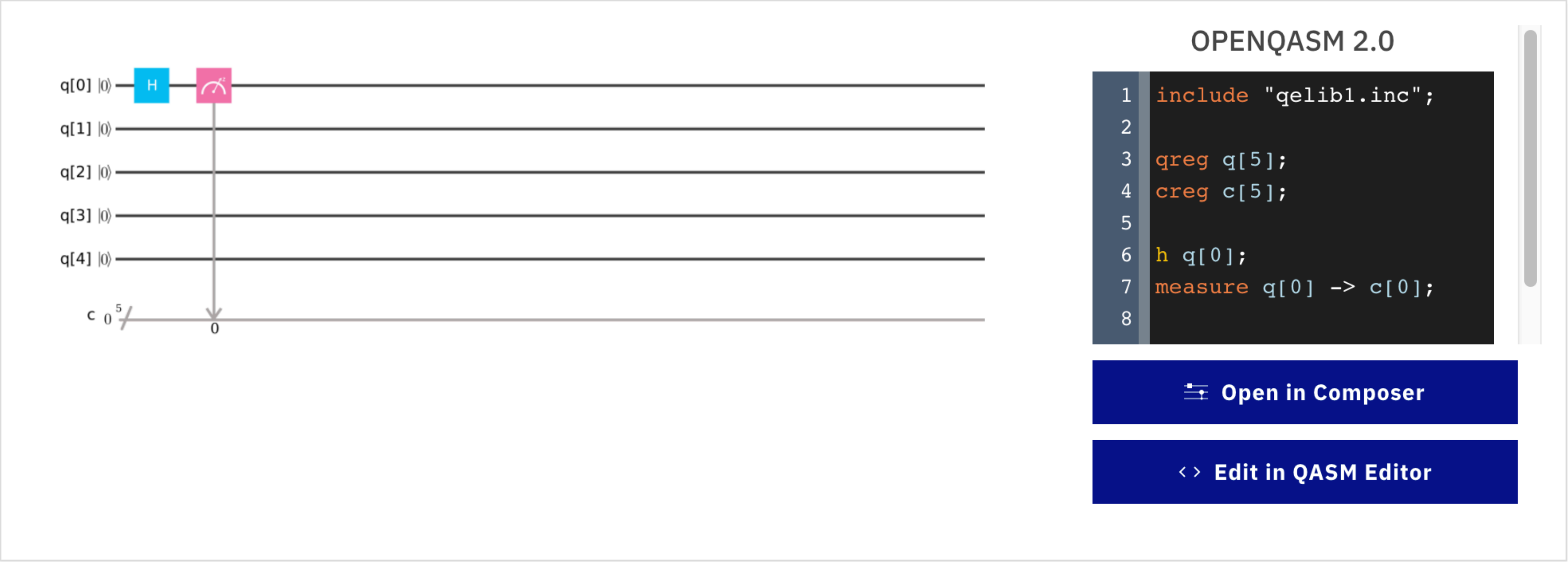
Quantum Results

Experiment #20180222153500

Device: ibmqx2

Quantum State: Computation Basis

Quantum Circuit



```
include "qelib1.inc";

qreg q[5];
creg c[5];

h q[0];
measure q[0] -> c[0];

```

IBM: openQASM

https://quantumexperience.ng.bluemix.net/qx/editor

https://github.com/QISKit/ibmqx-user-guides

https://github.com/QISKit/qiskit-tutorial/blob/master/index.ipynb

Microsoft: Q#

https://docs.microsoft.com/en-us/quantum/index?view=qsharp-preview

https://docs.microsoft.com/en-us/quantum/quantum-qr-intro?view=qsharp-preview

https://docs.microsoft.com/de-de/quantum/quantum-installconfig?view=qsharp-preview&tabs=tabid-vscode

Google

http://www.quantumplayground.net/#/home

https://github.com/quantumlib/OpenFermion

Algorithm of Deutsch

A Problem

You have coin…

…and you want to find out whether or not it’s bogus….

(i.e. whether it has both, head and tail, or only heads or only tails)

Can you find that out by flipping the coin <u>exactly once</u>?

In a classical world you can’t do this!

Problem Abstraction

$$f : \{0,1\} \to \{0,1\}$$

Such a function is either *constant* (i.e. f(0) = f(1))
or it is *balanced* (i.e. f(0) ≠ f(1))

Is it possible to evaluate the function f <u>once</u>
to determine whether f is constant or balanced?

A New Operator

$$U_f : |x, y\rangle \mapsto |x, y \oplus f(x)\rangle$$

$\mathbf{U_f}$ is unitary

Algorithm of David Deutsch

Step 1: Prepare the register

$|x\rangle|y\rangle \leftarrow |0\rangle|1\rangle$

Step 2: Apply Hadamard transformation

$|x\rangle|y\rangle \leftarrow H|x\rangle H|y\rangle$

Step 3: Evaluate f

$|x\rangle|y\rangle \leftarrow U_f\big(|x\rangle|y\rangle\big)$

U_f is performed only once!

Step 4: Apply Hadamard transformation

$|x\rangle|y\rangle \leftarrow H|x\rangle H|y\rangle$

Step 5: Measure the register

$|x\rangle|y\rangle = |0\rangle|1\rangle \Rightarrow$ f is constant

$|x\rangle|y\rangle = |1\rangle|1\rangle \Rightarrow$ f is balanced

Generalization

$$f:\{0,1\}^n \rightarrow \{0,1\}$$

f *constant* $:\Leftrightarrow \quad \forall\, x,y \in \{0,1\}^n : f(x) = f(y)$

f *balanced* $:\Leftrightarrow \quad \text{card}\, f^{-1}(0) = 2^{n-1} = \text{card}\, f^{-1}(1)$

(f maps half of the domain to 0, the other half to 1)

Problem:
Determine with a minimum number of evaluations of f whether f is constant or balanced!

Classical Case

In the classical case, even after having read half (i.e. 2^{n-1}) values it's not clear whether f is constant or balanced

Example:
All values read are 0, but the next value
(i.e. the $(2^{n-1}+1)$-th value) is 1 ⇒ f balanciert;
or the next value is 0 ⇒ f konstant.

I.e. a classical (deterministic) algorithm requires (worst case) $2^{n-1}+1$ evaluations of f

An "Oracle"

$$U_f : {}_2H^{\otimes n} \otimes {}_2H \rightarrow {}_2H^{\otimes n} \otimes {}_2H$$

$$|x, y> \;\; \mapsto \;\; |x, y \oplus f(x) >$$

(|x ⟩ is an n-Qbit-Quatum Register)

U_f is unitary

Algorithm of Deutsch-Jozsa

Step 1: Initialize the register

$$|x\rangle|y\rangle \leftarrow |0\rangle^{\otimes n}|1\rangle$$

Step 2: Apply the Hadamard Transformation

$$|x\rangle|y\rangle \leftarrow H^{\otimes(n+1)}\left(|0\rangle^{\otimes n}|1\rangle\right)$$

Step 3: Evaluate f

$$|x\rangle|y\rangle \leftarrow U_f\left(|x\rangle|y\rangle\right)$$

U_f will be executed exactly once!

Step 4: Apply the Hadamard Transformation

$$|x\rangle \leftarrow H^{\otimes n}|x\rangle$$

Step 5: Measure

$|x\rangle = |0\cdots 0\rangle \Rightarrow$ f is constant

$|x\rangle \neq |0\cdots 0\rangle \Rightarrow$ f is balanced

Meaning

The algorithm evaluates f exactly once!

In the classical case, f has to be evaluated (worst case) $2^{n-1}+1$ times!

The quantum algorithm of Deutsch-Jozsa results in an exponential speedup!

Quantum Parallelism

$f:\{0,1\}\to\{0,1\}$ → a function with a domain of cardinality 2

Classically, you need to invoke the function twice
to get the values f(0) and f(1)

$U_f : |x,y\rangle \mapsto |x, y\oplus f(x)\rangle$ → unitary map ("oracle")

$$U_f\Big(\|0\rangle+|1\rangle,|0\rangle\rangle\rangle\Big) = U_f\big(|0,0\rangle\big)+U_f\big(|1,0\rangle\big)$$ → U_f unitary, i.e. linear

$$= \left|0, f(0)\right\rangle + \left|1, f(1)\right\rangle$$

{ Single invocation of U_f
delivers all values of f!

Can be obviously extended to functions with finite domain

This is "quantum parallelism"

Searching

Unstructured Search

We want to find out to whom a certain phone number belongs. Alphabetic order of phone book doesn’t help!

Simple Solution: Inspect each of the N phone numbers until you found number at hand, then read the owner field of this record (this is *O(N)*)

Algorithm of Grover

There is a quantum algorithm that solves the problem in

$$G(N) = \frac{\pi}{4}\sqrt{N} = O\left(\sqrt{N}\right)$$

Classical unstructured search is O(N)

⇒ Quantum search results in quadratic speedup!

Application

Quantum search can speed-up (selective*) NP-problems

"Just" list all possible solutions and build a "database" out of them

Then use Grover algorithm to determine in $O(\sqrt{N})$ the solution

(*) You can define an oracle function for the problem
(which can be done for cracking keys, traveling salesman,…)

Quantum Information

No-Cloning Theorem

There can be no algorithm,
which can copy each arbitrary state of a system.

Formalization:

There exists no unitary transformation $U : H \to H$

such that for a chosen $\left|c\right\rangle \in H$ (the state receiving the copy)

and an arbitrary state $\left|\psi\right\rangle \in H$ holds: $\left(id \otimes U\right)\left(\left|\psi\right\rangle \otimes \left|c\right\rangle\right) = \left|\psi\right\rangle \otimes \left|\psi\right\rangle$

Orthogonal States

A quantum algorithm that can copy $|\psi\rangle$,
can at most copy states that are orthogonal to $|\psi\rangle$.

Let x be a classical bit. Then: $x = a|0\rangle + b|1\rangle \Rightarrow (a=1 \wedge b=0) \vee (a=0 \wedge b=1)$

Let x, y be two classical bits. Then: $x=y$ or $x \perp y$

⇒ Classical bits can be copied without limitations!

Quantum Teleportation

Problem:

Location A has a certain qbit $\left|\Psi\right\rangle_A = \alpha\left|0\right\rangle_A + \beta\left|1\right\rangle_A$.
We want to transmit classical bits from location A to location B
such that the qbit at A can be reconstructed at B.

This can be done!

A and B must agree in advance
on a certain entangled 2-qbit state,
and A and B hold 1-qbit of this entangled state.

A will create a mixed state from $\left|\Psi\right\rangle_A$ and the shared qbit
and compute the classical bits to transmitted to B from this mixed state.

B will use the received classical bits to lookup a unitary operator
to compute from his shared qbit the state $\left|\Psi\right\rangle_A$.

Quantum Teleportation

The "magic" is entanglement here!

At location A,
manipulation of the entangled state and following measurement
results in information about the part of the entangled qbit
at location B.

This information is used to tell location B
what to do to reconstruct the qbit at location B.

Conclusion

Summary

- Hardware of quantum computers is rapidly evolving ⇒ In the next few years deep problems will likely become solvable

- Quantum algorithms are based on the postulates of quantum mechanics ⇒ You have to understand a bit of this to work with quantum computers

- The current state of the art of software for implementing quantum computing is at the assembler level

Quote by Enrico Fermi

I am still confused…

…but at a higher level!

End