

Structural programming paradigm

IN QUANTUM COMPUTATION



mgr inż. Kamil Wereszczyński,
prof. dr hab. inż. Andrzej Polański

Katedra Grafiki,
Wizji Komputerowej
i Systemów Cyfrowych

*“**Nature isn't classical**, dammit,
and if you want to make
a simulation of nature, you'd better
make it quantum mechanical,
and by golly **it's** a **wonderful**
problem, because it
doesn't look so easy.”*

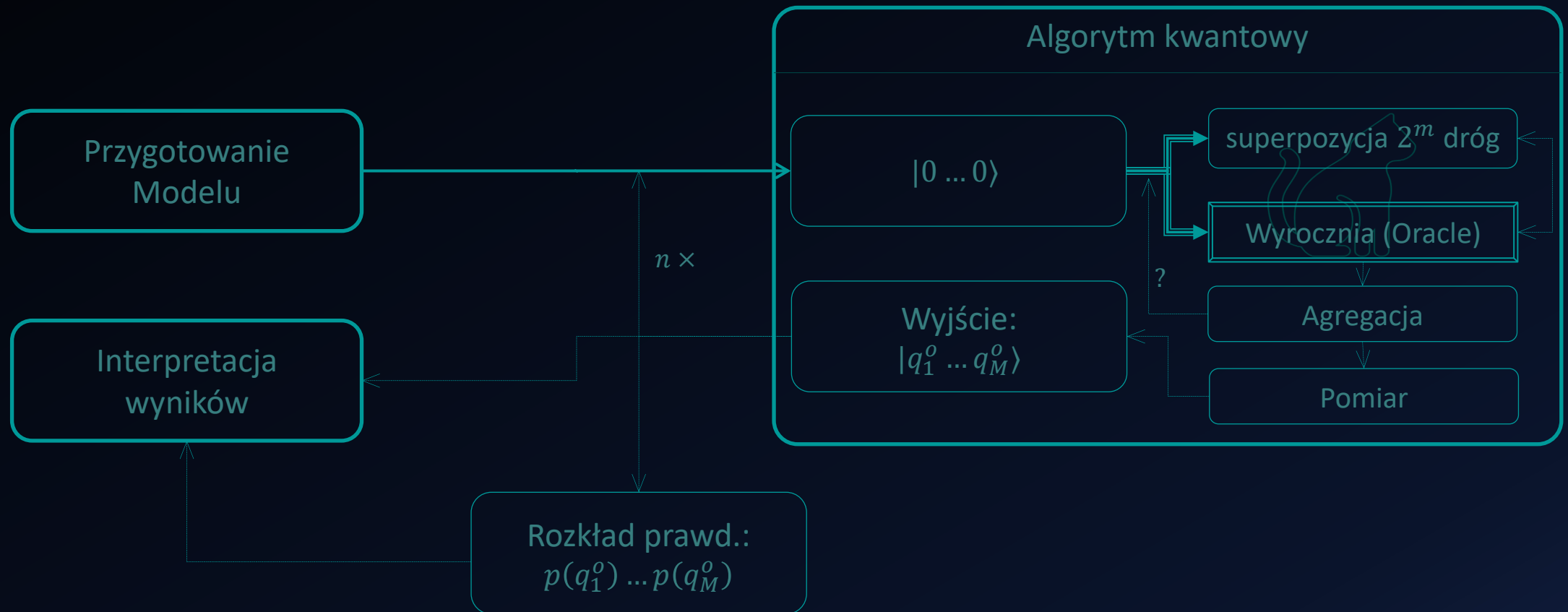
Richard Feynman.



Plan prezentacji

1. Krótki wstęp
2. Kwantowy bit – qubit
 - a. Definicja
 - b. Ewolucja i pomiar
 - c. Bramki (superpozycja) i obwody
 - d. Modele funkcji (core)
3. Systemy wielo-qubitowe
 - a. System dwuqubitowy
 - b. Baza obliczeniowa i pomiarowa
 - c. System n-qubitowy
4. Kompozyty wyroczni
 - a. Dodawanie (add)
 - b. Mnożenie (mul)
 - c. Porównanie (eq)
5. Zastosowania
 - a. Całkowanie
 - b. Klasyfikacja
6. Wnioski
7. Zakończenie

Kwantowe przetwarzanie danych



Algorytm kwantowy



Kwantowy bit - Qubit

- Z punktu widzenia informatyki kwantowej **qubit** jest to (abstrakcyjny) stan kwantowy ewoluujący wg operatora ewolucji i mierzony za pomocą obserwacji o dwóch różnych wartościach własnych.
 - **Abstrakcyjny** oznacza, że implementacja fizyczna nie jest istotna.
 - **Stan kwantowy** oznacza, że qubit podlega wszystkim postulatam mechaniki kwantowej.
 - Jest dwuwymiarowym wektorem nad ciałem liczb zespolonych
 - Jego norma musi być równa jeden.
 - Na wejściu układu może być równy tylko i wyłącznie jednemu z wektorów bazowych.
 - W czasie obliczeń qubit ewoluuje zgodnie z zadaniem unitarnym **operatorem ewolucji** ($A^\dagger A = \mathbf{1}$) i równaniem Schroedingera.
 - W czasie obliczeń może być dowolnym wektorem (spełniającym powyższe reguły).
 - Pomiar jest związany z **obserwacją**, czyli wielkością która może być zmierzona a jest reprezentowana przez **operator hermitowski** ($A^\dagger = A$).

Qubit – notacja

- Notacja: $|q\rangle, |q_0 q_1 \dots q_m\rangle = |q_0\rangle \otimes |q_1\rangle \otimes \dots \otimes |q_m\rangle$
- $|q\rangle = [\alpha, \beta]^T$, takim że $\alpha, \beta \in \mathbb{C}, \|\alpha\|^2 + \|\beta\|^2 = 1$
- $\langle q| = [\bar{\alpha}, \bar{\beta}]$. Zachodzi: $\|\bar{\alpha}\|^2 + \|\bar{\beta}\|^2 = \|\alpha\|^2 + \|\beta\|^2 = 1$
 - Jest z przestrzeni dualnej.
- α, β – amplitudy prawdopodobieństwa
- $\langle \Phi | \Psi \rangle = \|\overline{\Phi_\alpha} \Psi_\alpha\| + \|\overline{\Phi_\beta} \Psi_\beta\|$ - iloczyn skalarny.
- Stąd norma: $\langle q | q \rangle = \|\bar{\alpha} \alpha\| + \|\bar{\beta} \beta\| = 1$

Qubit - baza

- Macierz Pauliego $\sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$:
 - Wektory własne: $[0,1]^T, [1,0]^T$, wartości własne: $-1, 1$.
 - σ_z jest hermitowska, zatem jest obserwabłą.
- Wektory własne – baza pomiaru (ang. *Measurement basis*).
- Uznaje się (najwygodniej), że:
 - $|0\rangle = [1,0]^T = 1 \cdot |0\rangle + 0 \cdot |1\rangle$ - wartość mierzona 1.
 - $|1\rangle = [0,1]^T = 0 \cdot |0\rangle + 1 \cdot |1\rangle$ - wartość mierzona -1 .
- Inne obserwable (wartości własne zawsze $1, -1$):
 - $\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$
 - $\sigma_y = \begin{bmatrix} 0 & i \\ -i & 0 \end{bmatrix}, \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}, \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix}$

Jak ewoluuje w czasie jeden qubit?

- $|q\rangle = \alpha|0\rangle + \beta|1\rangle = \cos\frac{\theta}{2}|0\rangle + \sin\frac{\theta}{2}e^{i\varphi}|1\rangle$
- Sfera Blocha
- Najbardziej ogólna postać operatora dla jednego qubit:
$$\begin{bmatrix} \cos(\theta/2) & -e^{i\lambda}\sin(\theta/2) \\ e^{i\varphi}\sin(\theta/2) & e^{i\lambda+i\varphi}\cos(\theta/2) \end{bmatrix}$$
- Operator $\sim$ bramka

Przykłady bramek jedno-qubitowych: swap gate

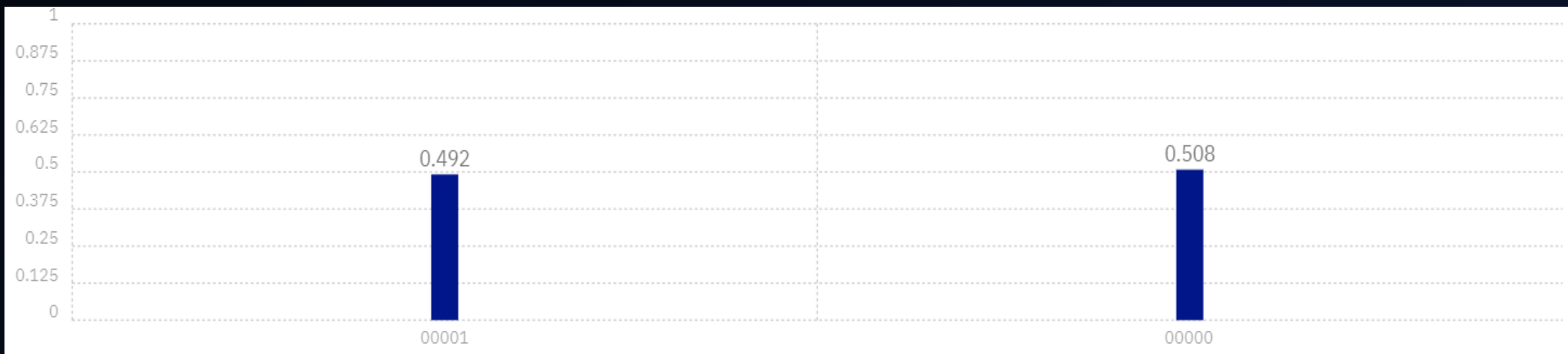
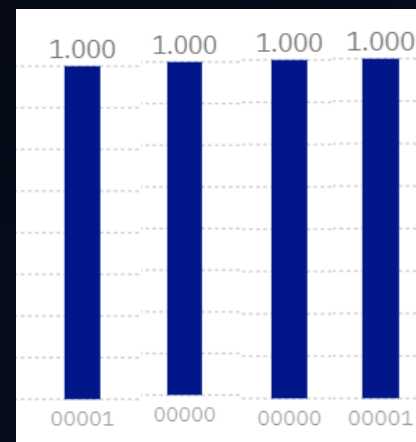
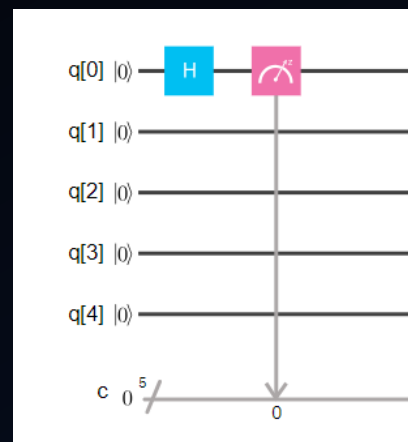
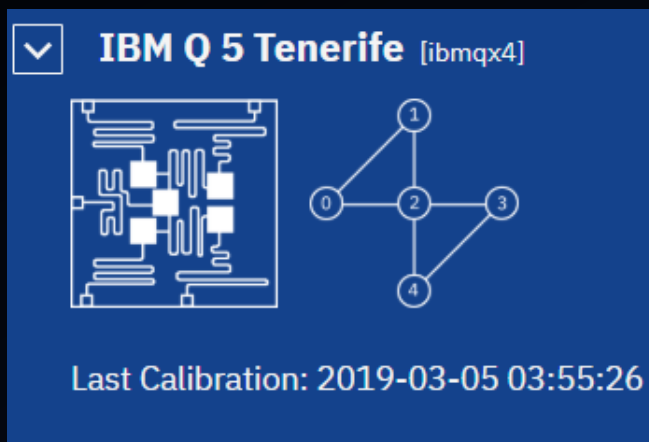
- $X = \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ (SWAP, Pauli X-gate)
- $X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$
- $X|1\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle$

Superpozycja: Hadamard gate

- $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ (Hadamard $\Leftarrow$ Clifford gate)
- $H|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$
- $H|1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$
- $\langle 0|H|0\rangle = \frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}} = \frac{1}{2}$, $\langle 1|H|0\rangle = \frac{1}{2}$
- $\langle 1|H|1\rangle = \frac{1}{2}$, $\langle 1|H|1\rangle = -\frac{1}{\sqrt{2}} \left(-\frac{1}{\sqrt{2}} \right) = \frac{1}{2}$
- **Superpozycja jest super!**

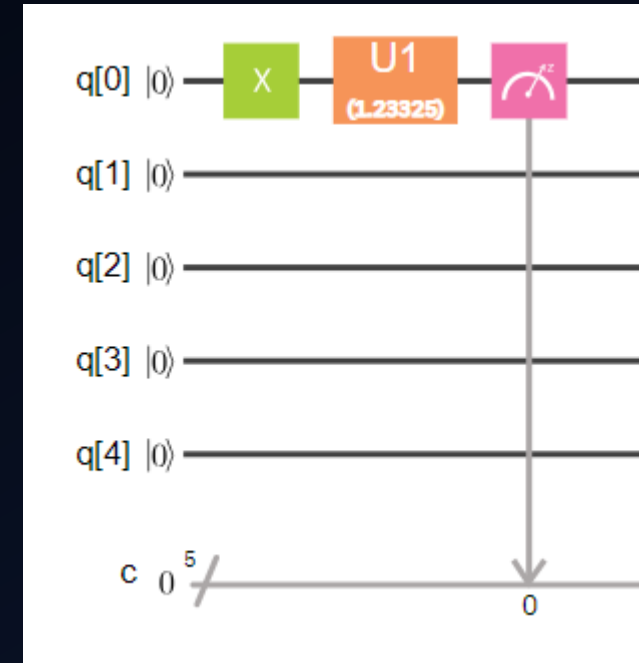


Experyment z bramką Hadamarda



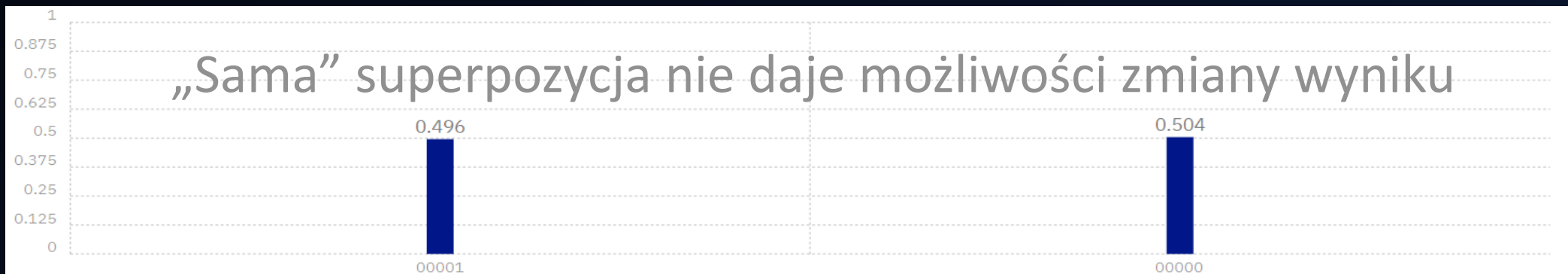
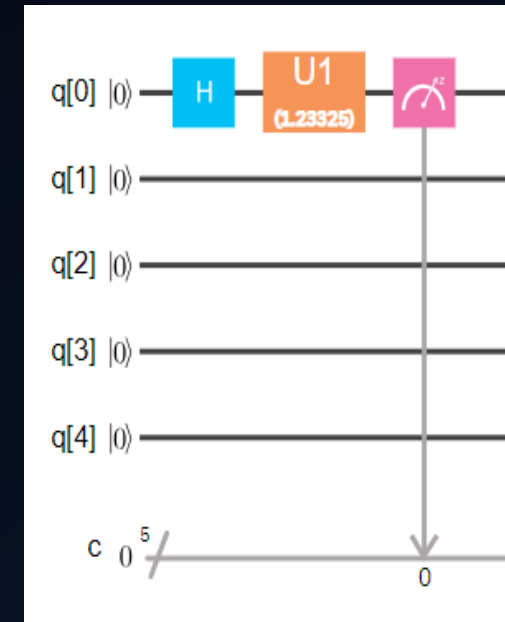
Przykłady bramek jedno-qubitowych: phase shift

- $R_\phi = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix}$
- $R_\phi |0\rangle = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle$
- $R_\phi |1\rangle = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ e^{i\phi} \end{bmatrix} = e^{i\phi} |1\rangle,$
- $\langle 0 | e^{i\phi} |1\rangle = 0, \langle 1 | e^{i\phi} |1\rangle = 1$

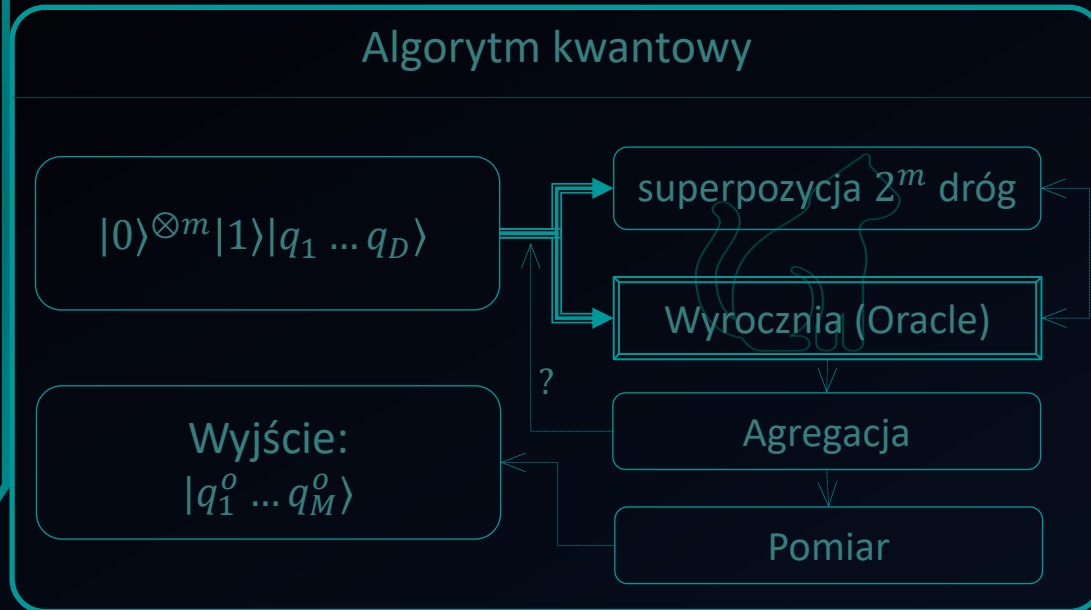


Zaczynamy obliczenia – obwody kwantowe

- $|\Psi\rangle = R_\phi H|0\rangle$ (kolejność!)
- $|\Psi\rangle = R_\phi \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}} (|0\rangle + e^{i\phi} |1\rangle)$
- $\langle 0|\Psi\rangle = \frac{1}{2}$
- $\langle 1|\Psi\rangle = \frac{1}{\sqrt{2}} e^{i\phi} \frac{1}{\sqrt{2}} e^{i\phi} = \frac{e^{i\phi} e^{-i\phi}}{2} = \frac{1}{2}$



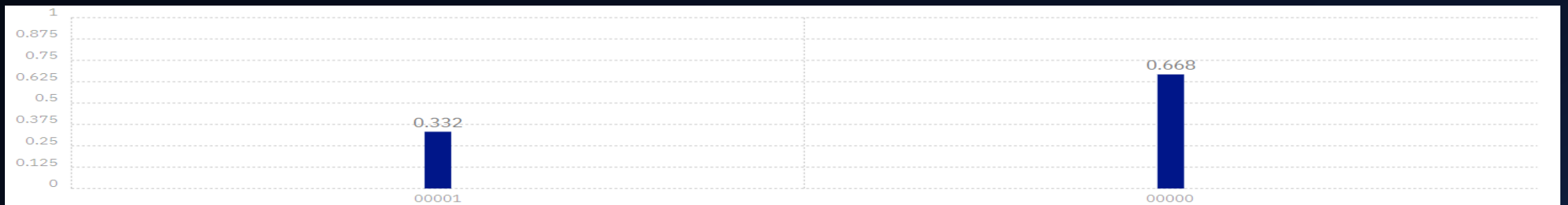
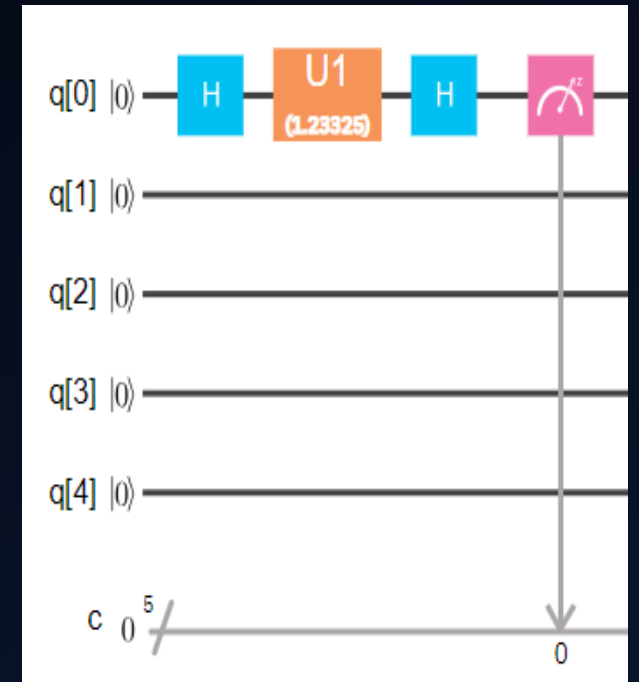
Dlaczego to nie działa?



- Mamy stan wejściowy $|0\rangle$
- Mamy superpozycję H
- Mamy coś na kształt wyroczni R_ϕ
- Brakuje agregacji!

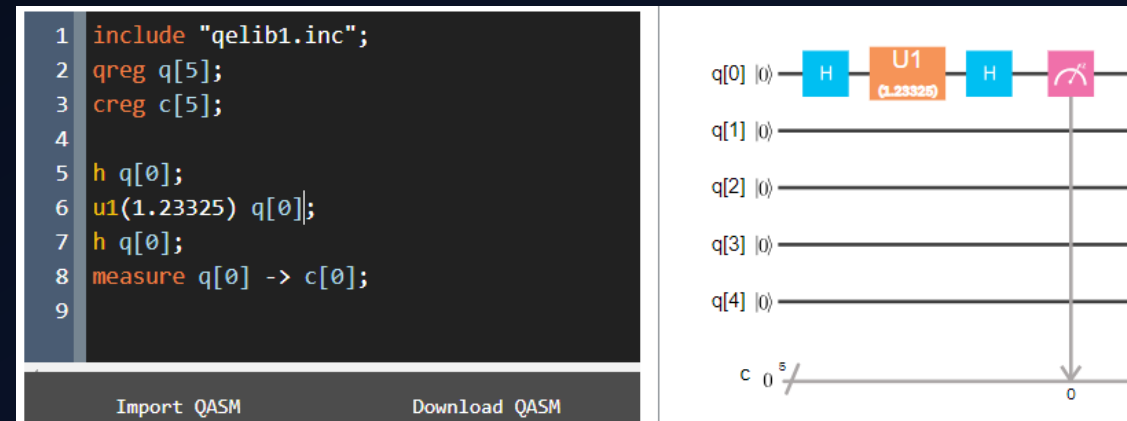
Zaczynamy obliczenia – obwody kwantowe

- $|\Psi_\phi\rangle = HR_\phi H|0\rangle$
- $|\Psi_\phi\rangle = H \frac{1}{\sqrt{2}} (|0\rangle + e^{i\phi}|1\rangle) = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ e^{i\phi} \end{bmatrix}$
 $= \frac{1}{2} \begin{bmatrix} 1 + e^{i\phi} \\ 1 - e^{i\phi} \end{bmatrix} = \frac{1}{2} [(1 + e^{i\phi})|0\rangle + (1 - e^{i\phi})|1\rangle]$
- $\langle 0|\Psi_\phi\rangle = \frac{(1+e^{i\phi})(1+e^{-i\phi})}{4} = \frac{1+\cos(\phi)}{2} = 0.66558$



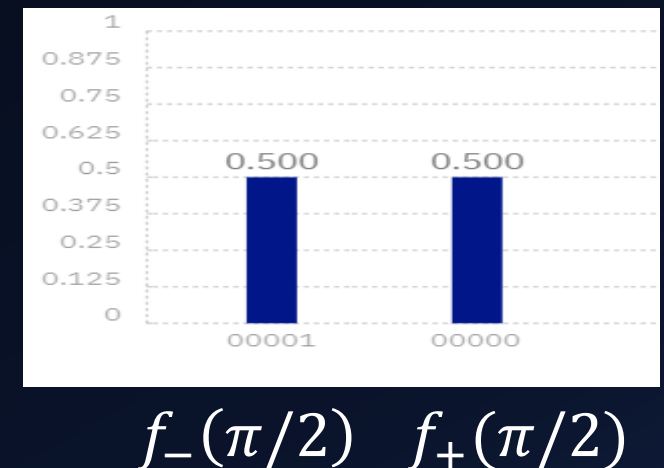
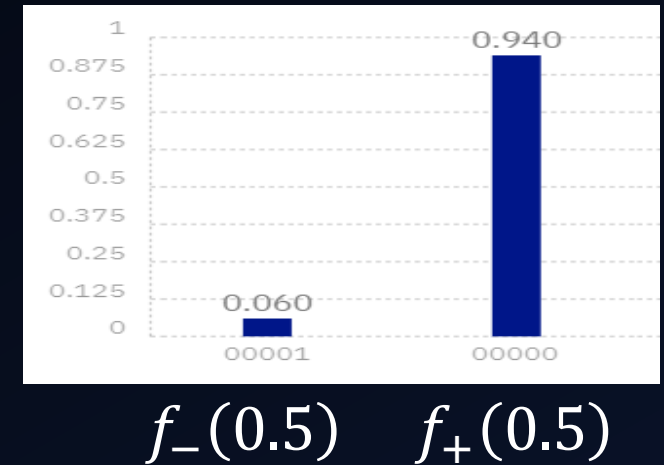
Obwód kwantowy

- **Obwód kwantowy** jest to z punktu widzenia informatyki kwantowej, abstrakcyjny operator ewolucji systemu qubitów w wybranym kanale (czasie), zbudowany z bramek kwantowych.
 - Jest to operator unitarny
 - Podobieństwo do bramki
 - W przypadku jednego qubitu jest wynikiem mnożenia operatorów (składania bramek)
- $HR_\phi H$ - obwód kwantowy dla jednego qubita
- QASM – „assembler” do budowania obwodów kwantowych



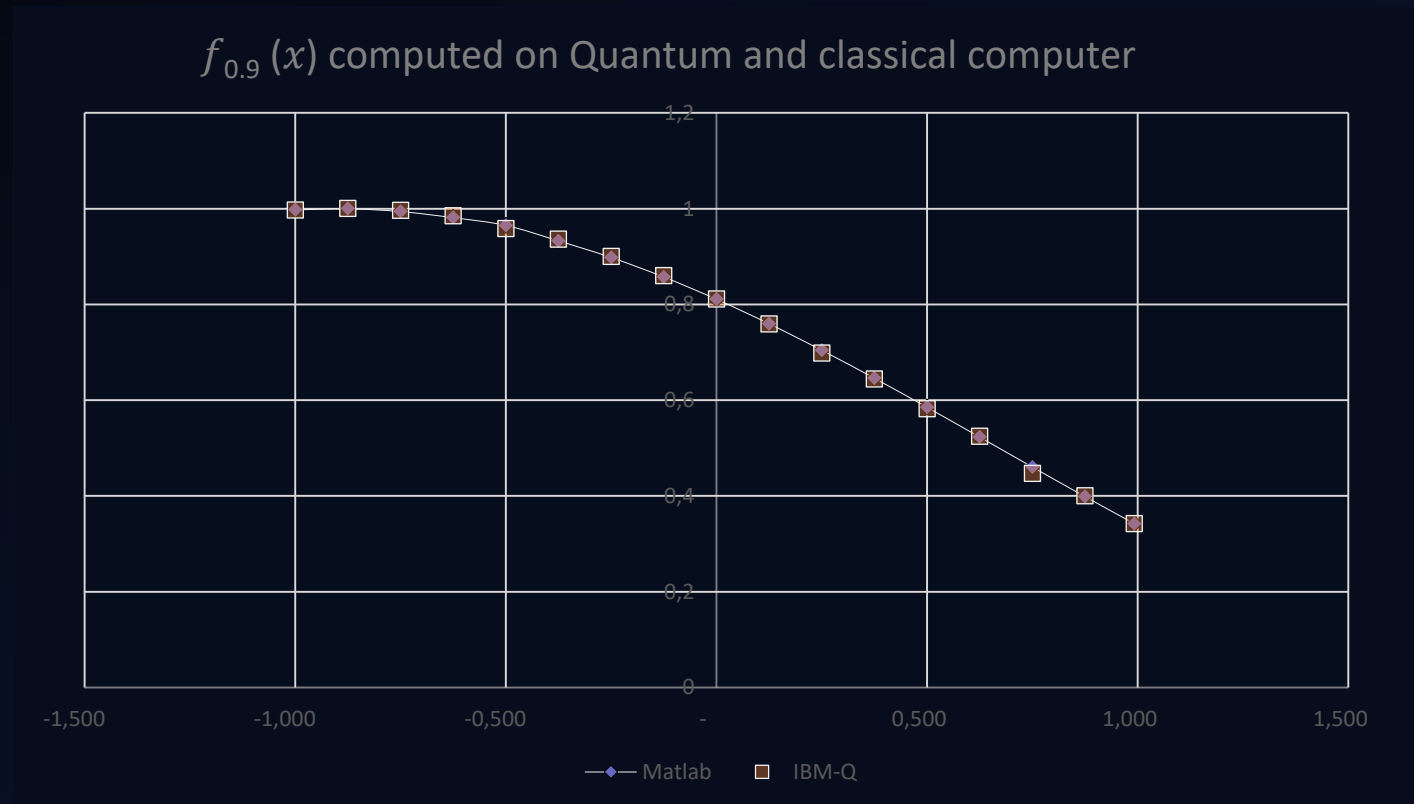
Model funkcji w oparciu o $|\Psi\rangle$

- $\Psi_\phi = HR_\phi H$, $|\Psi_\phi\rangle = \Psi_\phi|0\rangle$
 - $\langle 0|\Psi\rangle = \frac{1+\cos(\phi)}{2}$
- Model funkcji $f_+ : \mathbb{R} \rightarrow \mathbb{R}$, $f_+(x) = \frac{1+\cos(x)}{2}$:
 - Wykonaj **dostateczną** ilość razy obwód Ψ_x
 - $f_+(\phi) = \frac{\#\{|\Psi_\phi\rangle=|0\rangle\}}{\#\{|\Psi_\phi\rangle\}}$
- Model funkcji $f_- : \mathbb{R} \rightarrow \mathbb{R}$, $f_-(x) = \frac{1-\cos(x)}{2}$:
 - Wykonaj **dostateczną** ilość razy obwód $|\Psi_x\rangle$
 - $f_-(x) = \frac{\#\{|\Psi_\phi\rangle=|1\rangle\}}{\#\{|\Psi_\phi\rangle\}}$



Model 3

- $\Gamma_\lambda = HR_x R_\lambda H$ $|\Gamma_\lambda\rangle = \frac{1}{2} \left[(1 + e^{i(x+\lambda)})|0\rangle + (1 - e^{i(x+\lambda)})|1\rangle \right]$
- $\langle 0|\Gamma_\lambda\rangle = \frac{1+\cos(x+\lambda)}{2}$
- Model funkcji $f_\lambda: \mathbb{R} \rightarrow \mathbb{R}$
$$f_\lambda(x) = \frac{1+\cos(x+\lambda)}{2}$$
 - $f_\lambda(x) = \frac{\#\{|\Gamma_\lambda\rangle=|0\rangle\}}{\#\{|\Gamma_\lambda\rangle\}}$



Model 4

- $\Gamma_{n,\lambda} = HR_x^n R_\lambda H \quad |\Gamma_\lambda\rangle = \frac{1}{2} [(1 + e^{i(nx+\lambda)})|0\rangle + (1 - e^{i(nx+\lambda)})|1\rangle]$
- $\langle 0 | \Gamma_{n,\lambda} \rangle = \frac{1 + \cos(nx + \lambda)}{2}$
- Model funkcji $f_{n,\lambda}: \mathbb{R} \rightarrow \mathbb{R} \quad f_{n,\lambda}(x) = \frac{1 + \cos(nx + \lambda)}{2}$
 - $f_{n,\lambda}(x) = \frac{\#\{|\Gamma_{n,\lambda}\rangle = |0\rangle\}}{\#\{|\Gamma_{n,\lambda}\rangle\}}$
- core – podstawa obliczeń.

Aproksymacja / interpolacja funkcji

- Spostrzeżenie: skończony (co najwyżej przeliczalny) zbiór $\mathcal{F} = \{f_{n,\lambda} : f_{m,\gamma} \in \mathcal{F} \Rightarrow n \neq m \vee \lambda \neq \gamma\}$ jest liniowo niezależny
 - Stąd: $\text{span}(\mathcal{F})$ jest skończone wymiarową podprzestrzenią $(\mathcal{L}_2, \mathbb{R})$
 - Każdą funkcję $\mathbb{R} \rightarrow \mathbb{R}$ można przybliżyć kombinacją liniową z $\mathcal{F}$.
- $$\langle 0 | \Gamma_\lambda \rangle = \frac{1}{4} (1 + e^{i(nx+\lambda)}) (1 + e^{-i(nx+\lambda)}) =$$
$$\frac{1}{4} (1 + e^{i\lambda} e^{inx}) (1 + e^{-i\lambda} e^{-inx}) = \frac{1}{4} (1 + c \cdot e^{inx}) (1 + \bar{c} \cdot e^{-inx})$$
- Zatem λ gra rolę współczynnika ustalanego przy aproksymacji (czy też interpolacji) – odbywa się to po „stronie kwantowej”.
- Zatem potrzebujemy wprowadzić operację dodawania dwóch funkcji $f_{n,\lambda} + f_{m,\gamma}$
- Nie da się tego zrobić jednym qubitem – potrzebujemy dwóch.

System dwóch qubitów

- Jest to iloczyn tensorowy dwóch qubitów:
 - $|q_0 q_1\rangle = |q_0\rangle \otimes |q_1\rangle$
 - niech $|q_i\rangle = [\alpha_i, \beta_i]^T$:
 $|q_0 q_1\rangle = [\alpha_0, \beta_0]^T \otimes [\alpha_1, \beta_1]^T = [\alpha_0 \alpha_1, \alpha_0 \beta_1, \beta_0 \alpha_1, \beta_0 \beta_1]^T$
- 4 wektory bazowe: $|00\rangle = |0\rangle, |01\rangle = |1\rangle, |10\rangle = |2\rangle, |11\rangle = |3\rangle$
 - Ogólnie: $|k\rangle, k = 0, \dots, 3$.
- Przestrzeń $\mathbb{C}^4$.
- Można budować z dowolnej bazy σ_j
 - Tutaj: $\sigma_z \otimes \sigma_z$

Bramki dla dwóch qubitów

Dwa rodzaje bramek:

- Iloczyn tensorowy bramek jedno-qubitowych.
- Bramki, których nie da się uzyskać z iloczynu tensorowego dwóch bramek, np. Controlled NOT – CNOT cX (ale też $cR_\phi, cH, Hc, \dots$):

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} a \\ b \\ d \\ c \end{bmatrix}$$

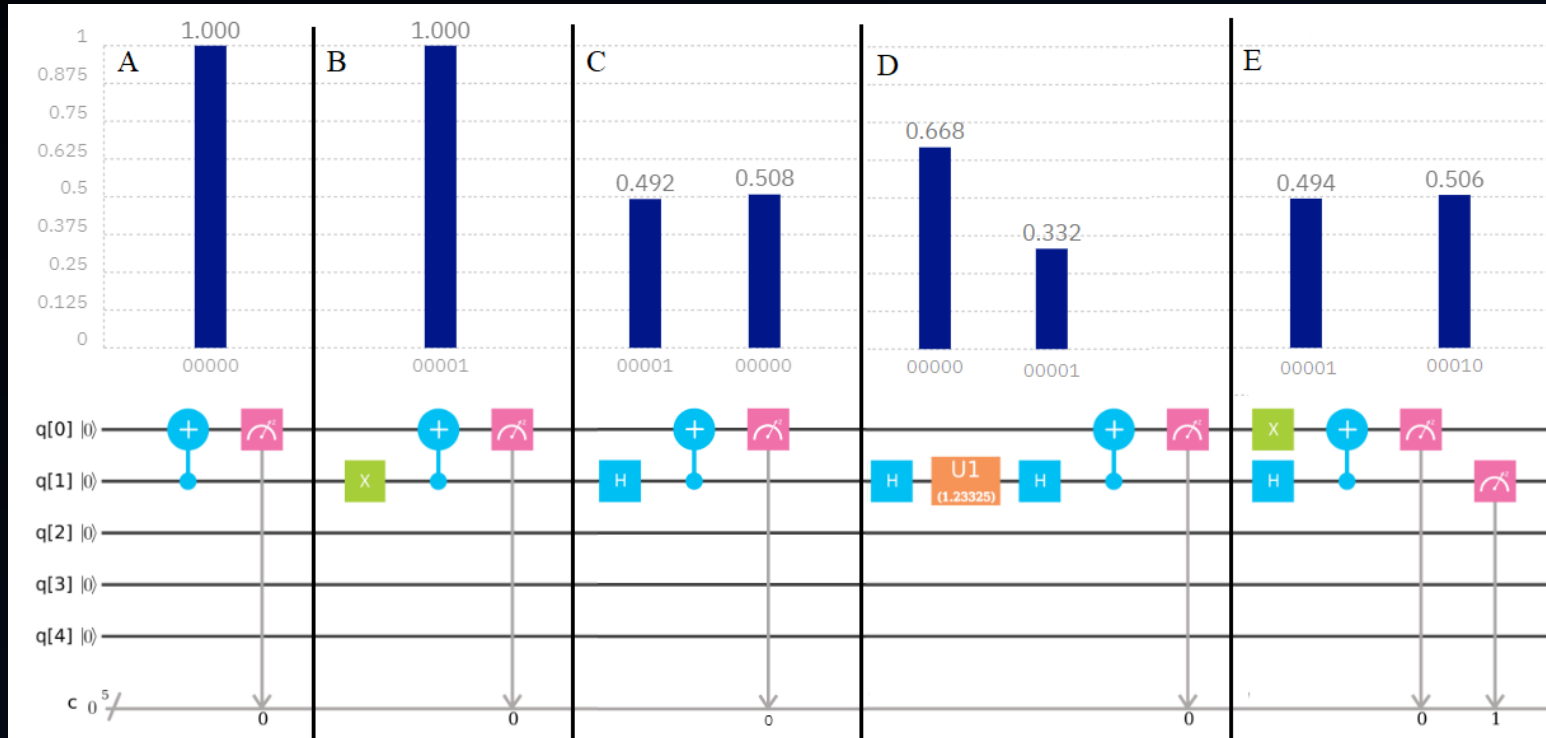
$$cX(a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle) = (a|00\rangle + b|01\rangle + d|10\rangle + c|11\rangle)$$

Jeżeli pierwszy qubit jest równy $|1\rangle$
to na drugi qubit działa bramka SWAP

Pomiar w systemach dwu-qubitowych

- Mierzmy dwa qubity, baza: $|k\rangle \in \{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$
- System $|q_0q_1\rangle = (a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle)$
możemy zmierzyć „jednym qubitem” $|j\rangle$ z prawej, lub z lewej strony:
 - $\langle q_0q_1|j\rangle = \bar{a} \langle 0|\langle 0|j\rangle + \bar{b} \langle 0|\langle 1|j\rangle + \bar{c} \langle 1|\langle 0|j\rangle + \bar{d} \langle 1|\langle 1|j\rangle$
 - $\|\langle q_0q_1|0\rangle\| = \|\bar{a} \langle 0|\langle 0|0\rangle + \bar{b} \langle 0|\langle 1|0\rangle + \bar{c} \langle 1|\langle 0|0\rangle + \bar{d} \langle 1|\langle 1|0\rangle\| = \|\bar{a}\langle 0| + \bar{c}\langle 1|\| = \sqrt{\|a\|^2 + \|c\|^2}$
 - $\|\langle q_0q_1|1\rangle\| = \sqrt{\|b\|^2 + \|d\|^2}$
 - Prawdopodobieństwo uzyskania $|0\rangle$ ($|1\rangle$) jest normą dwuwymiarowego wektora o współrzędnych od tych wektorów bazowych (czterowymiarowych) które mają po prawej stronie 0 (1).
 - Tylko w „wygodnej bazie” ☺
 - Analogicznie pomiar z prawej strony

Bramka CNOT w praktyce - splątanie



- A, B: Zmierzono zawsze $|0\rangle$ lub jeden w zależności od wartości qubita kontrolnego
- C: Przeniesienie superpozycji
- D: Przeniesienie amplitudy

$|q_0\rangle = |1\rangle \rightarrow |q_1\rangle = |0\rangle$ **Ala** bierze $|q_0\rangle$ i leci na **Marsa**
 $|q_0\rangle = |0\rangle \rightarrow |q_1\rangle = |1\rangle$ **Bolek** zostaje z $|q_1\rangle$ na **Ziemi**

Koszmarne oddziaływanie na odległość = komunikacja kwantowa



n-qubitów

- Baza obliczeniowa ma wymiar $2^n: \{|k\rangle\}_{k=0}^{2^n-1}$
 - Zapis w postaci binarnej bardziej informatywny $|17\rangle = |010001\rangle$
- Baza obliczeniowa może mieć mniej wymiarów niż wynika to z ilości qubitów zaangażowanych w obliczenia, jedna zawsze musi to być 2, 4, 8, 16, ... - dlatego że mierzymy 1, 2, 3, 4, ... qubity.
- Pomiar lokalny w podbazie ortonormalnej $\{|\xi_k\rangle\}_{k=0}^m$:
 - $|\Psi\rangle = \sum_k |\xi_k\rangle \otimes |\eta_k\rangle \quad \langle Q|\Psi\rangle = \sum_{k=0}^m \langle Q|\xi_k\rangle |\eta_k\rangle$
 - $|q_0q_1\rangle = (a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle) = |0\rangle \otimes (a|0\rangle + b|1\rangle) + |1\rangle \otimes (c|0\rangle + d|1\rangle)$
 - $\langle q|q_0q_1\rangle = \langle q|0\rangle(a|0\rangle + b|1\rangle) + \langle q|1\rangle(c|0\rangle + d|1\rangle)$
 - $\langle 0|q_0q_1\rangle = \langle 0|0\rangle(a|0\rangle + b|1\rangle) + \langle 0|1\rangle(c|0\rangle + d|1\rangle)$

Pomiar w bazie lokalnej

- W „wygodnej” reprezentacji qubitów prawdopodobieństwo uzyskania wektora $|j\rangle$ z bazy pomiarowej uzyskujemy licząc normę całego wektora ale zerując te współrzędne dla których końcowe „bity” z bazy obliczeniowej nie zgadzają się z $|j\rangle$. Np.:
 - $|\Psi\rangle = \frac{1}{\sqrt{3}} |001001110\rangle + \frac{1}{\sqrt{3}} |111001110\rangle + \frac{1}{\sqrt{3}} |001001111\rangle$
 - mierząc w bazie 8-wymiarowej:
 - $\|\langle\Psi|110\rangle\| = \frac{2}{3}$, $\|\langle\Psi|111\rangle\| = \frac{1}{3}$, pozostałych: 0 (z prawej strony)
 - $\|\langle 001|\Psi\rangle\| = \frac{2}{3}$, $\|\langle 111|\Psi\rangle\| = \frac{1}{3}$, pozostałych: 0 (z lewej strony)

Dodawanie dwóch funkcji (add)

- Dane: $\Psi = R_x^n R_\lambda H = R_{nx+\lambda} H = R_\phi H$, i dwa stany $|\Psi_p\rangle$ oraz $|\Psi_q\rangle$
- $|\Psi_p\rangle = \frac{1}{\sqrt{2}} [1, e^{ip}]^T$, $|\Psi_q\rangle = [1, e^{iq}]^T$ zatem
- $|\Psi_p\rangle \otimes |\Psi_q\rangle = \frac{1}{2} [1, e^{iq}, e^{ip}, e^{ip} e^{iq}]^T$ - ostatnia współrzędna „psuj”:
 - Bo jak byśmy nie agregowali, ten człon nie ma sensu. Dlatego trzeba go wyeliminować:
- $cR_{-p} cR_{-q} |\Psi_p \Psi_q\rangle = \frac{1}{2} [1, e^{iq}, e^{ip}, 1]^T$ - pomiar $\Leftrightarrow pr = \frac{1}{4}$
- $|A_{pq}\rangle = IHcR_{-p} cR_{-q} |\Psi_p \Psi_q\rangle = \frac{1}{2\sqrt{2}} [1 + e^{iq}, 1 - e^{iq}, 1 + e^{ip}, e^{ip} - 1]^T$

Splątanie!

Dodawanie dwóch funkcji - cd

$ 00\rangle$	$\frac{1 + e^{iq}}{2\sqrt{2}}$
$ 01\rangle$	$\frac{1 - e^{iq}}{2\sqrt{2}}$
$ 10\rangle$	$\frac{1 + e^{ip}}{2\sqrt{2}}$
$ 11\rangle$	$\frac{e^{ip} - 1}{2\sqrt{2}}$

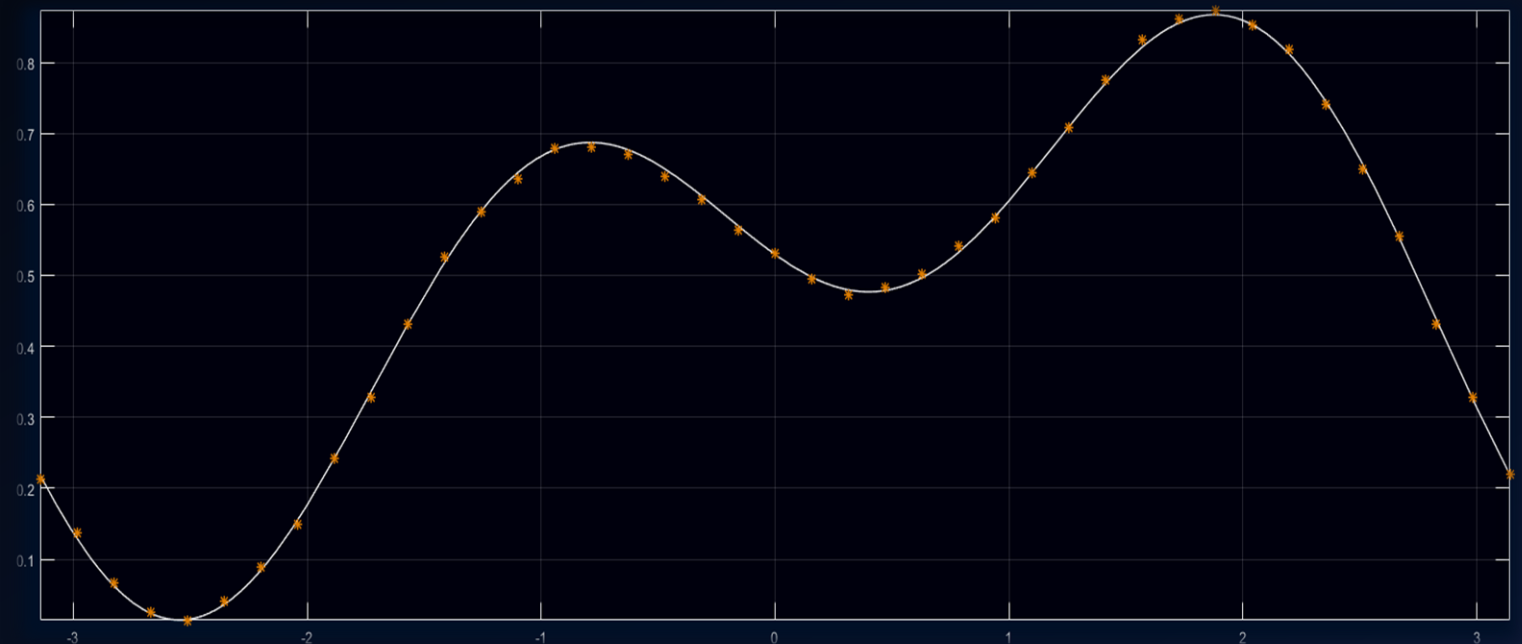
- Współrzędne które tworzą sumę stoją przy tych wektorach bazowych, które mają 0 na końcu
- Zatem dokonując pomiaru w bazie jedno-qubitowej (z prawej strony) dla wektora $|0\rangle$:
- $\langle A_{pq} | 0 \rangle$ - to jest wektor
- $\langle A_{pq} | 0 \rangle = \frac{1+e^{-iq}}{2\sqrt{2}} |0\rangle + \frac{1+e^{-ip}}{2\sqrt{2}} |1\rangle$
- $||\langle A_{pq} | 0 \rangle|| = \frac{1+\cos(q)}{4} + \frac{1+\cos(p)}{4}$

Wyniki eksperymentalne

```
include "qelib1.inc";
qreg q[10];
creg c[10];
gate core (n, x, l) q {
    h q;
    u1 (n*x) q;
    u1 (l) q;
}
gate cRR (n, m, x, l1, l2) q0,q1 {
    cu1 (-(n*x)-l1) q0,q1;
    cu1 (-(m*x)-l2) q0,q1;
}
gate iH ctrl,q {
    id ctrl;
    h q;
}
gate add (n, m, x, l1, l2) q0, q1 {
    core(n, x, l1) q0;
    core(m, x, l2) q1;
    cRR(n, m, x, l1, l2) q0,q1;
    iH q0,q1;
}

add(1, 2, 1.3, -0.9, 2.1) q[0], q[1];
measure q[1] -> c[0];
```

$$f(x) = \frac{1 + \cos(x - 0.9)}{4} + \frac{1 + \cos(2x + 2.1)}{4}$$

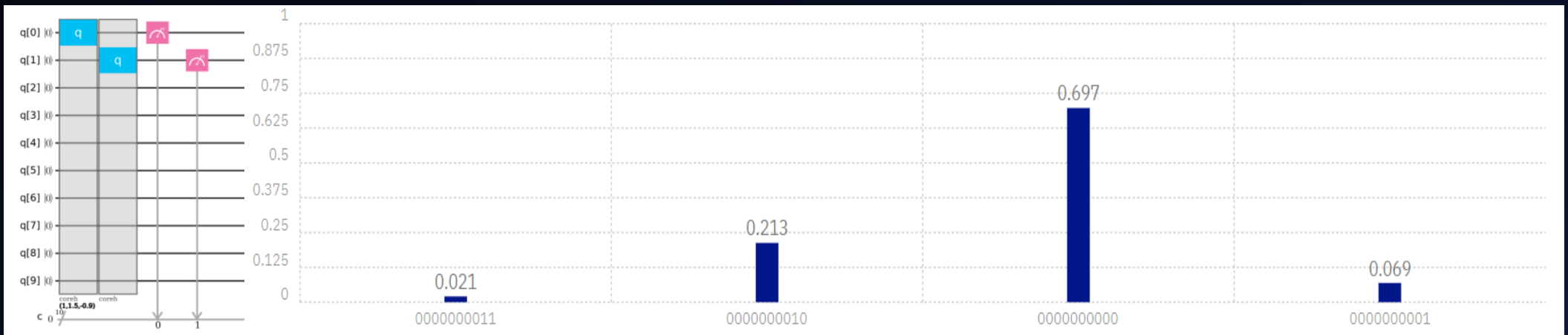


Mnożenie (mul)

- Dane: $\Psi = HR_x^n R_\lambda H = HR_{nx+\lambda} H = HR_\phi H$,
 - dwa stany $|\Psi_p\rangle$ oraz $|\Psi_q\rangle$

- $|\Psi_p\rangle \otimes |\Psi_q\rangle = \frac{1}{16} [(1 + e^{ip})(1 + e^{iq}), (1 + e^{ip})(1 - e^{iq}), (1 + e^{iq}) \dots]^T$

- $\|\langle 00 | \Psi_p \Psi_q \rangle\| = \frac{(1 + \cos p)(1 + \cos q)}{4} = 0.6952$



Mnożenie (mul)

$ 000\rangle$	0
$ 001\rangle$	$\frac{(1 + e^{ip})(1 + e^{iq})}{16}$
$ 010\rangle$	$\frac{(1 + e^{ip})(1 - e^{iq})}{16}$
$ 011\rangle$	0
$ 100\rangle$	$\frac{(1 - e^{ip})(1 + e^{iq})}{16}$
$ 101\rangle$	0
$ 110\rangle$	$\frac{(1 - e^{ip})(1 - e^{iq})}{16}$
$ 111\rangle$	0

- Przy pomiarze jednego qubitu uzyskamy zawsze dwa składniki.
- $\otimes |0\rangle$ - dostajemy 8 współrzędnych
- Dostaniemy $\langle 0|S\rangle = 0, \langle 0|S\rangle = 1$
- Macierz permutacji – unitarna
- Permutacja (0,1).
- $\langle M_{pq}|1\rangle = \frac{(1+e^{ip})(1+e^{iq})}{16} |00\rangle$
- $\|\langle M_{pq}|1\rangle\| = \frac{1+\cos(q)}{4} \cdot \frac{1+\cos(p)}{4}$

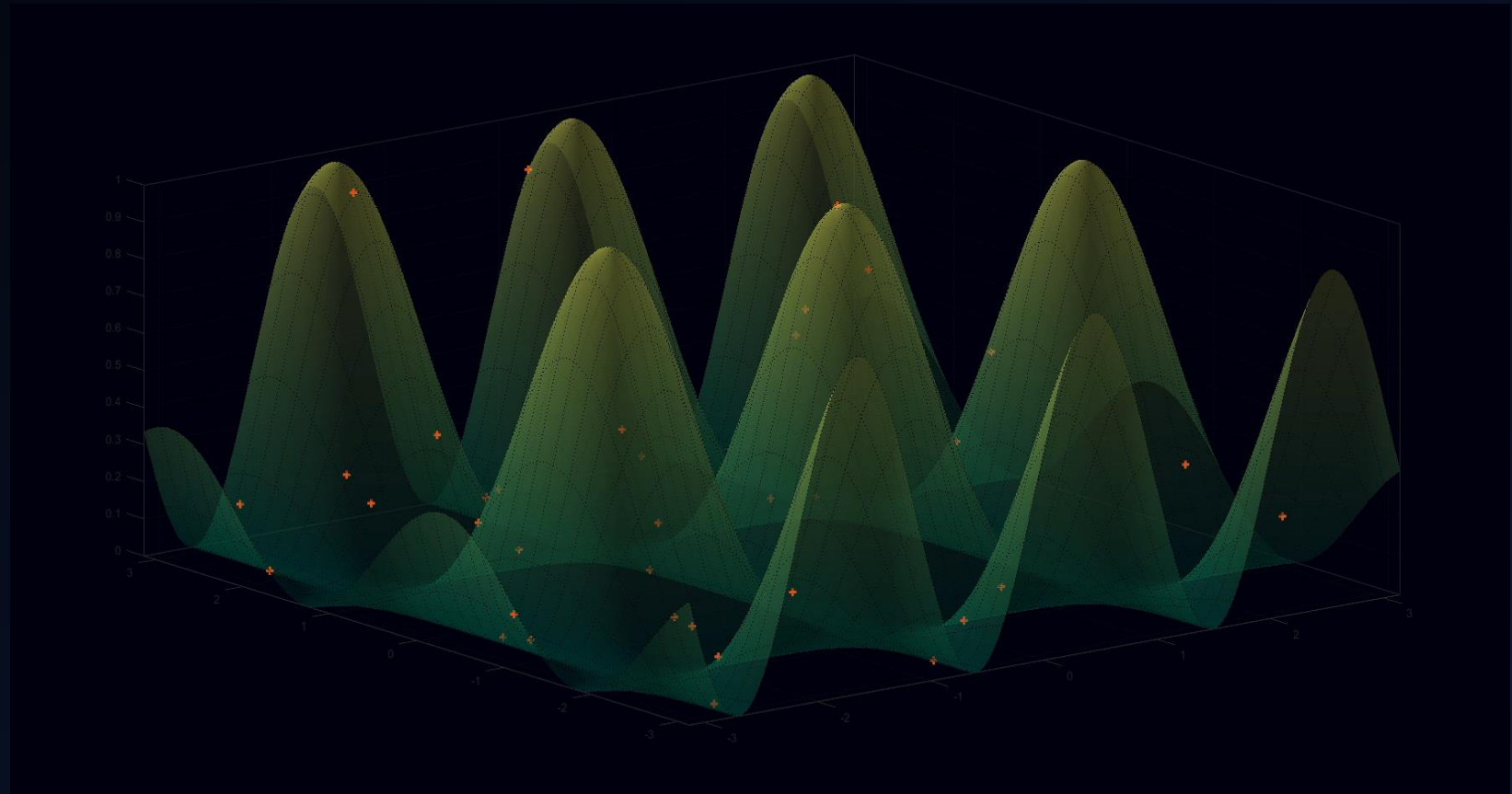
Wyniki eksperymentalne - mnożenie

```
include "qelib1.inc";
qreg q[10];
creg c[10];
gate coreh (n, x, 1) q {
    h q;
    u1 (n*x) q;
    u1 (1) q;
    h q;
}

gate perm01 q0,q1,q2{
    x q0;
    x q1;
    x q2;
    ccx q0,q1,q2;
    x q0;
    x q1;
    x q2;
}

coreh(1,1.5,-1.3) q[0];
coreh(2,-0.54,0.6) q[1];
perm01 q[0], q[1], q[2];
measure q[0] -> c[0];
measure q[1] -> c[1];
```

$$f(x, y) = \frac{1 + \cos(x - 1.3)}{4} \frac{1 + \cos(2y + 0.6)}{4}$$



Porównywanie

- $\Psi = R_x^n R_\lambda H = R_{nx+\lambda} H = R_\phi H$, i stany $|\Psi_p\rangle$ oraz $|\Psi_q\rangle = \Psi X|q\rangle$
- Jeśli $p = q = x$, to $|\Psi_p\rangle = [1, e^{ix}]$, $|\Psi_q\rangle = [1, -e^{ix}]$
- Wystarczy zatem dodać te dwie wartości. Wynik „0” będzie oznaczał, że $p = q$; każdy inny że $p \neq q$.
- Im większa będzie różnica między eksponentami, tym większa różnica między liczbami – zatem nie jest to wartość typu logicznego, ale raczej miara (nie)podobieństwa pomiędzy liczbami.
 - Co może również mieć zastosowanie.

Porównywanie - cd

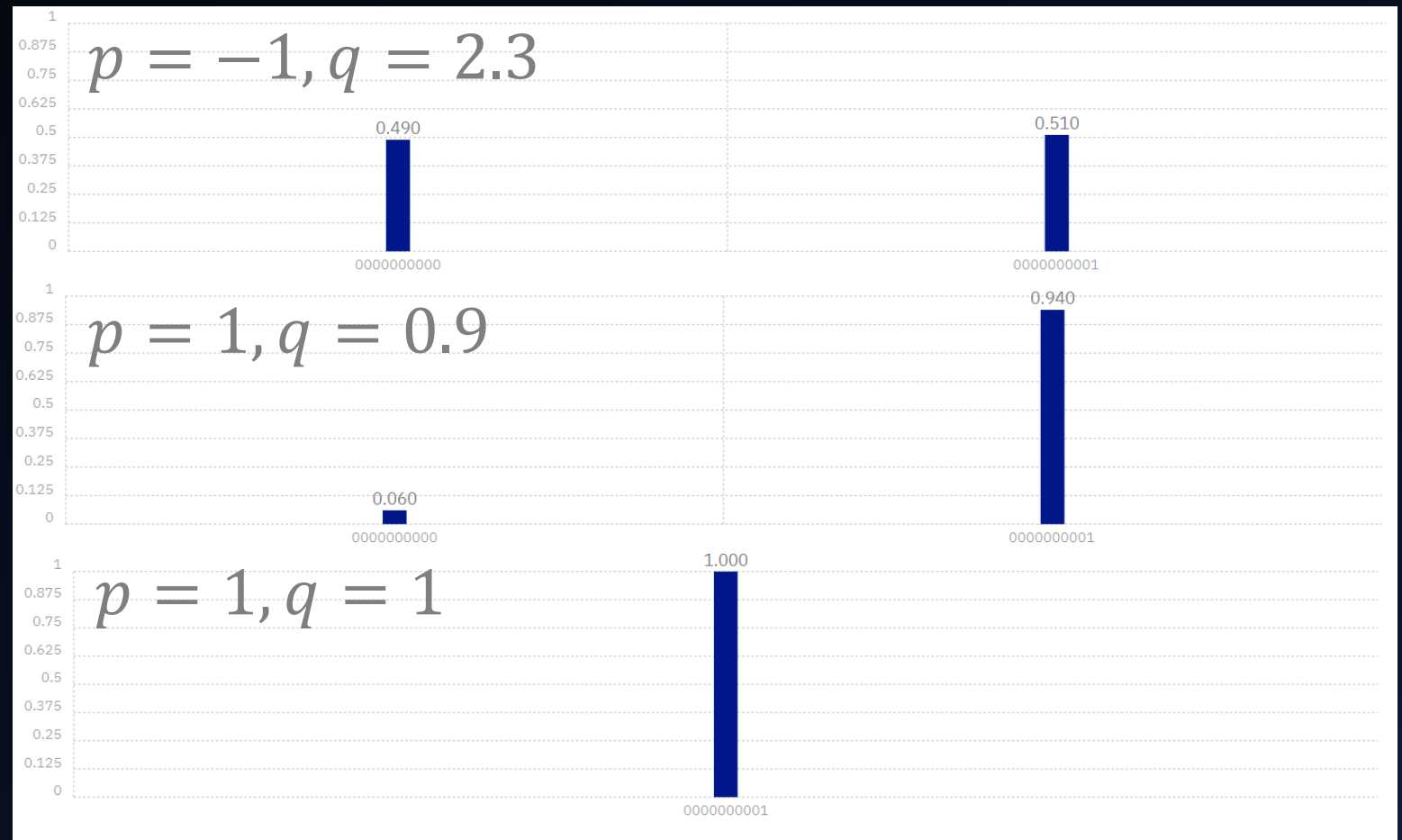
$ 00\rangle$	$\frac{e^{iq} - e^{ip}}{4}$
$ 01\rangle$	$\frac{2 - e^{iq} - e^{ip}}{4}$
$ 10\rangle$	$\frac{e^{ip} - e^{iq}}{4}$
$ 11\rangle$	$\frac{2 + e^{ip} + e^{iq}}{4}$

$$\begin{cases} \|\langle E_{pq} | 0 \rangle\| = 0 \\ \|\langle E_{pq} | 1 \rangle\| = 1 \end{cases}$$

- Podobnie do dodawania: $cR_{-p}cR_{-q} |\Psi_p \Psi_q\rangle = \frac{1}{2} [1, e^{iq}, -e^{ip}, -1]^T$
- Ale nie możemy użyć dodawania, bo minus się zredukuje przy obliczaniu modułu i nie otrzymamy zera. Dążymy do tego, żeby **amplituda** się wyzerowała.
- $H^{\otimes 2} cR_{-p}cR_{-q} |\Psi_p \Psi_q\rangle$
- $|E_{pq}\rangle = cX H^{\otimes 2} cR_{-p}cR_{-q} |\Psi_p \Psi_q\rangle$
- $\|\langle E_{pq} | 0 \rangle\| = \frac{1 - \cos(q-p)}{4}$

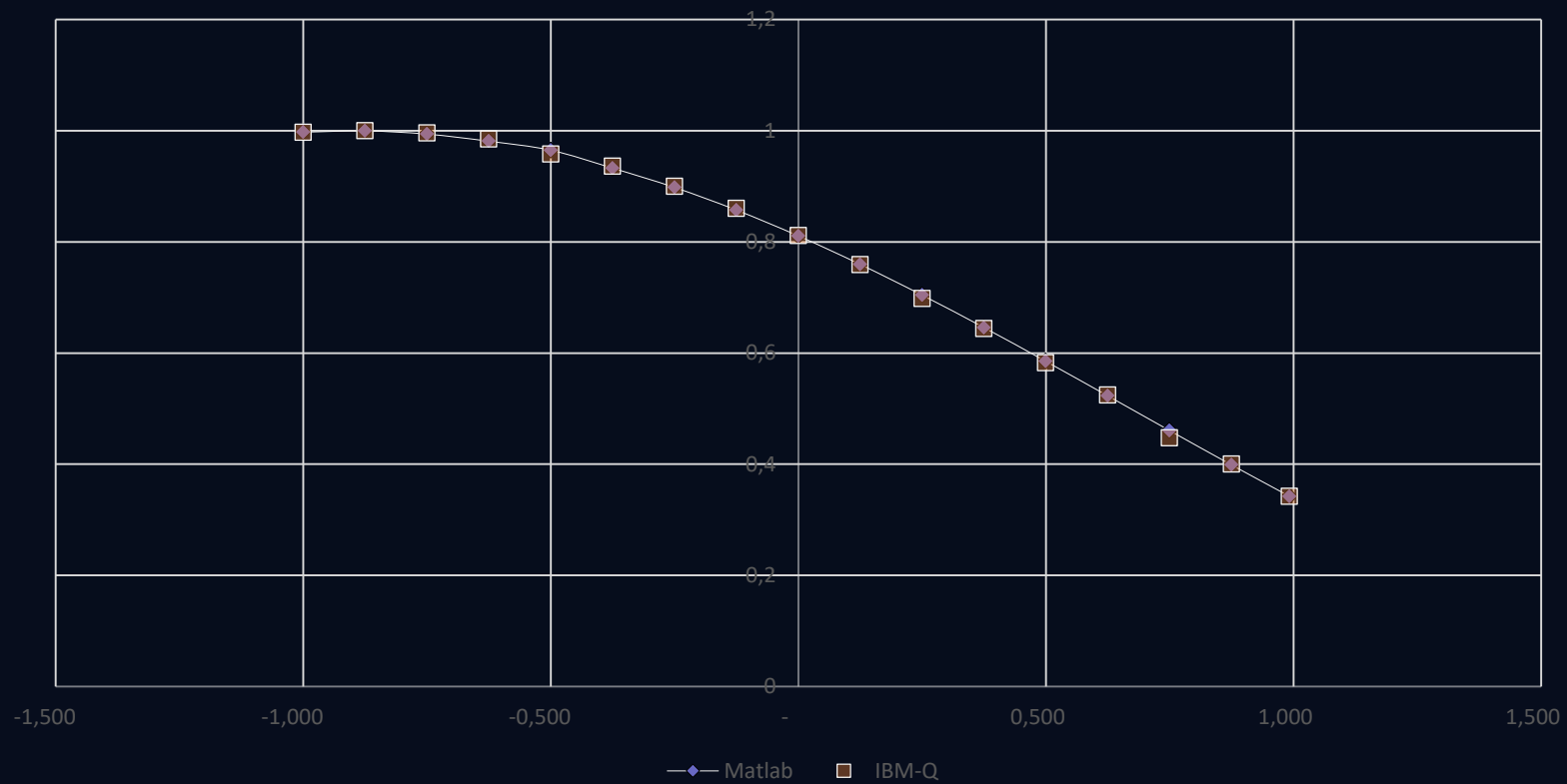
Wyniki eksperymentalne

```
include "qelib1.inc";
qreg q[10];
creg c[10];
gate core (n, x, l) q {
  h q;
  u1 (n*x) q;
  u1 (l) q;
}
gate eq (x,y) q0, q1{
  x q0;
  core(1,x,0.9) q0;
  core(1,y,0.9) q1;
  cu1(-0.9-x) q0,q1;
  cu1(-0.9-y) q0,q1;
  h q0;
  h q1;
  cx q0, q1;
}
eq (1,1) q[0], q[1];
measure q[1] -> c[1];
```

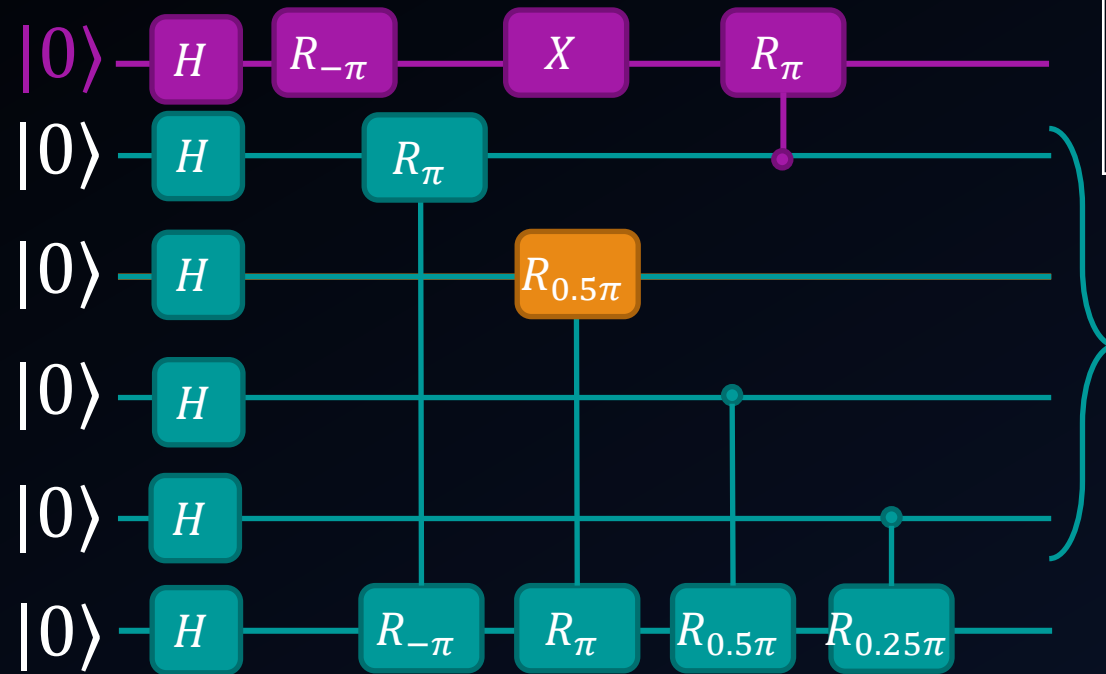


Ładowanie danych

$f_{0.9}(x)$ computed on Quantum and classical computer



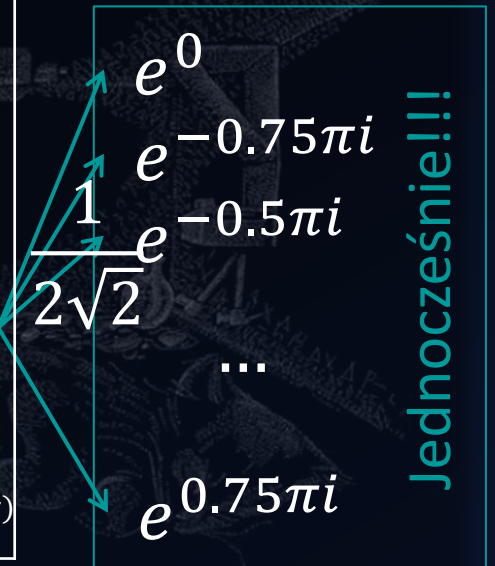
Zastosowania: całkowanie



$ 00\rangle$	0.5
$ 01\rangle$	0.5
$ 10\rangle$	0.5
$ 11\rangle$	$0.5e^{i\pi}$

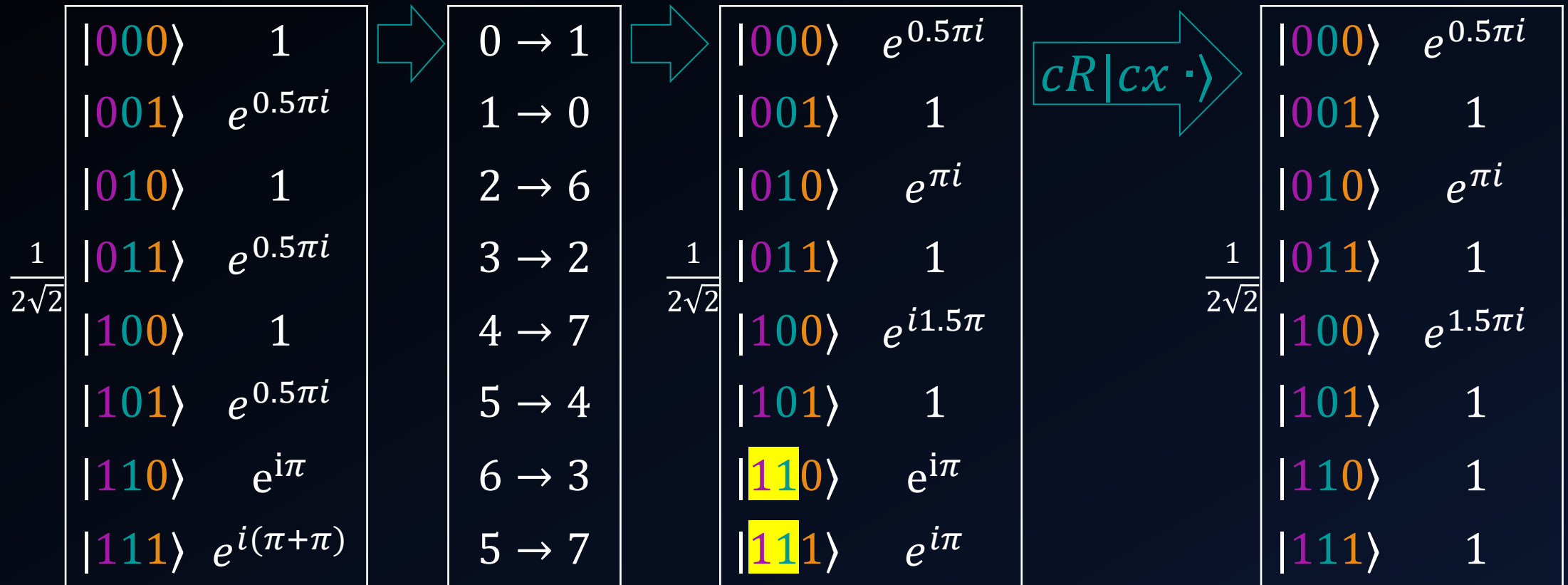
superpozycja

$ 000\rangle$	1
$ 001\rangle$	$e^{0.5\pi i}$
$ 010\rangle$	1
$ 011\rangle$	$e^{0.5\pi i}$
$ 100\rangle$	1
$ 101\rangle$	$e^{0.5\pi i}$
$ 110\rangle$	$e^{i\pi}$
$ 111\rangle$	$e^{i(\pi+0.5\pi)}$



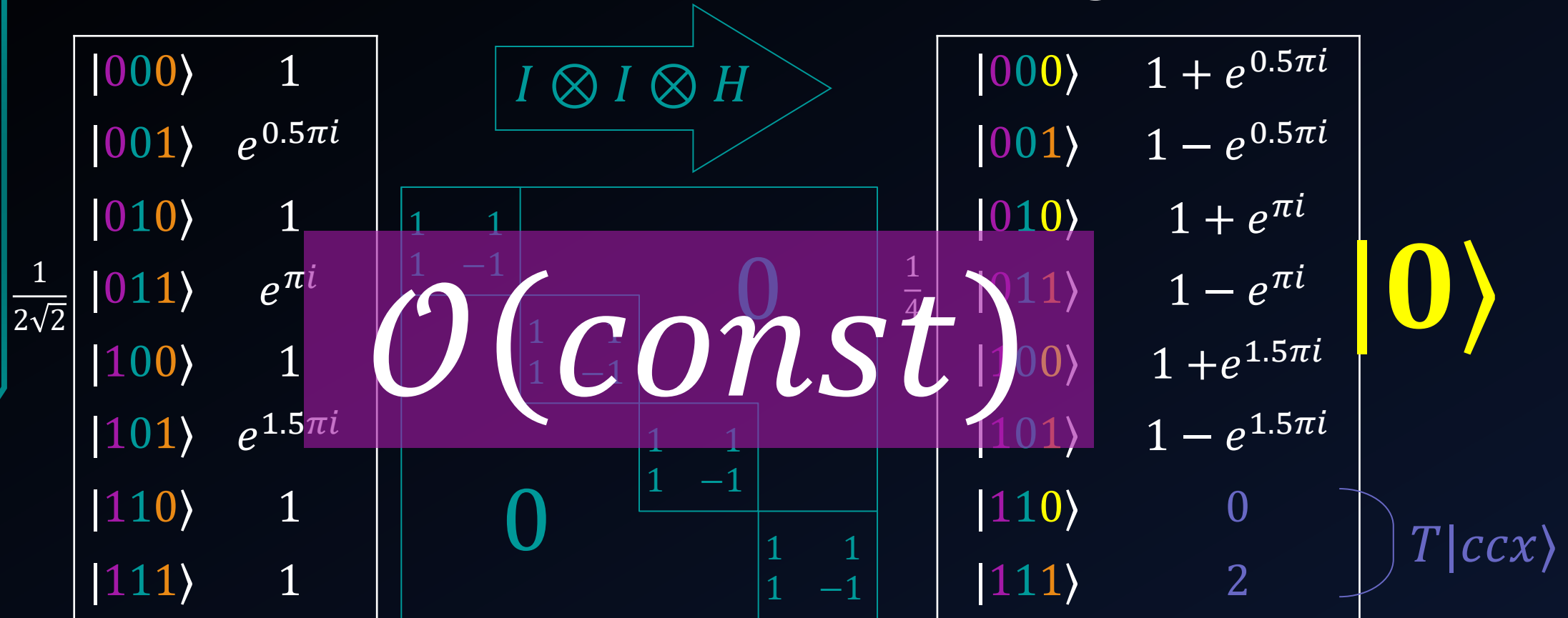
- Idea na podstawie QFT (ang. *Quantum Fourier Transformation*).
- W trakcie obliczeń istnieje wszystkie 2^n wartości funkcji jednocześnie!
- Wystarczy je dodać.

Zastosowanie: całkowanie → permutacje



Po następnym kroku będzie 3+4=7 wyrazów istotnych, czyli będzie 9 jedynek
 Po tym kroku 0.625 jedynek, po następnym: 0.5625 – maleje (dokładanie).

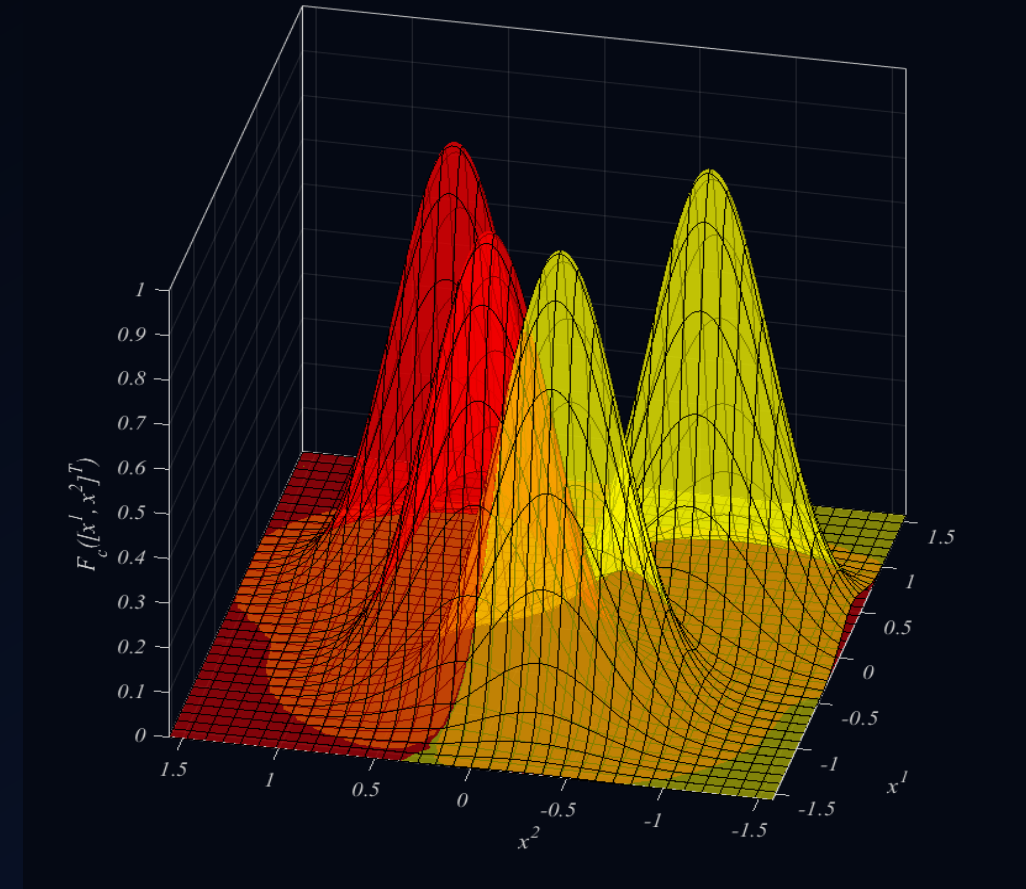
Zastosowania: całkowanie – do brzegu



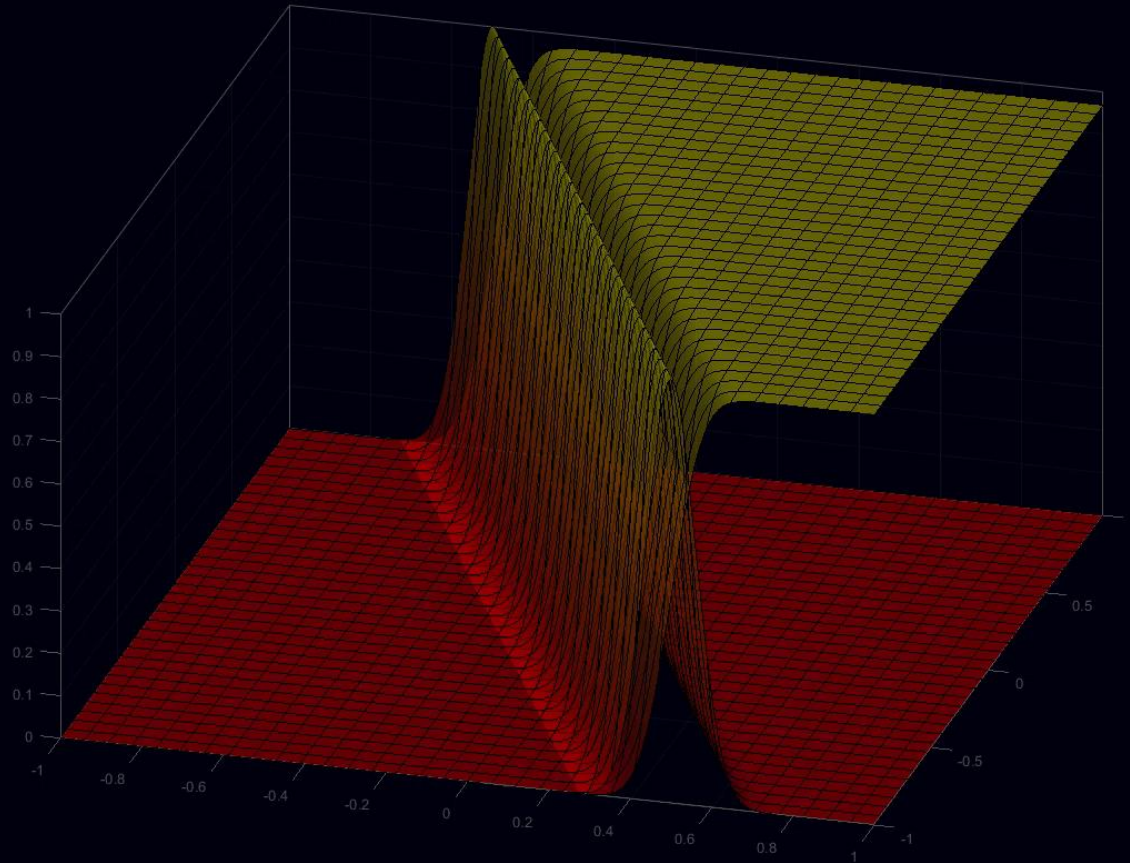
$$\frac{1}{8} \left[\frac{1 + \cos(0.5\pi)}{2} + \frac{1 + \cos(\pi)}{2} + \frac{1 + \cos(1.5\pi)}{2} \right] = \frac{1}{8} \sum_k f(x_k) \approx \frac{1}{8} \int_{-\pi}^{\pi} f(x) dx$$

Zastosowanie: klasyfikacja

- Przestrzeń cech: $\mathbb{R}^n$
- Funkcja przynależności do klasy:
- $f(\mathbf{c}) = 1, g(\mathbf{c}) = 0$ dla czerwonych
- $f(\mathbf{c}) = 0, g(\mathbf{c}) = 1$ dla żółtych
- $$f(x) = \sum_{j=1}^K \left(\frac{1 + \cos(x + \lambda_j)}{2} \right)^{n_j}$$
 - K - ilość punktów
 - λ_j - współrzędna j -tego punktu
 - Szukamy n_j tak żeby „pasowało” do danych
 - Algorytm Grovera



qSVM



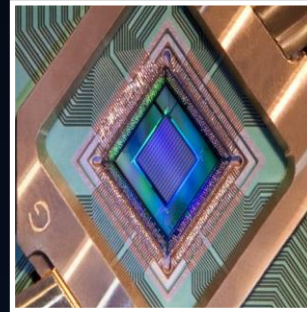
- $f(x, y) = \left(\frac{1 + \cos(nx + my)}{2} \right)^K$
- $2\pi \int_{-1}^x \left(\frac{1 + \cos(nx + my)}{2} \right)^K dx$
- Szukamy:
 - n, m – wyznaczają nachylenie hiperpłaszczyzny
 - K - margines

Narzędzia



IBMQ

- <https://quantumexperience.ng.bluemix.net/qx/editor>
- Dostęp do komputerów
 - 5 qubitów – composer
 - Do 16 –python QISKIT (Terra, Aqua,...)
 - 20 – po uzgodnieniach „płatne”

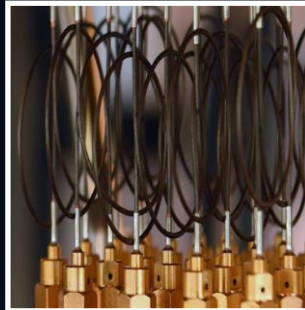


Dwave

- Adiatyczny – oparty o model wyżarzania (Isinga)
- Inne podejście – jeden układ – potrzeba mapowania problemu
- Jest dostęp z poziomu pythona

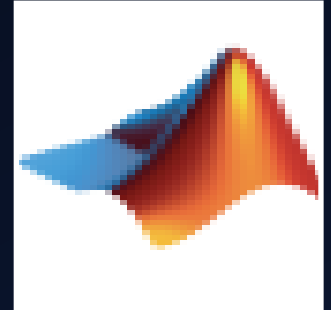
Microsoft Quantum

- <https://www.microsoft.com/en-us/quantum/>
- Integracja z Azure
- Język Q#
- Tylko hasłowo – dla mnie „niezbadane”



Matlab

- Symbolic Math Toolbox
- `[de2bi(0:7,3, 'left-msb'), kron(I,kron(I,H))*PRMT*kron(Rp(x), P1(y))]`
- Mogę udostępnić kod, który pozwala oswoić się z prezentowanymi zagadnieniami.



Wnioski

- Idea wyroczeni kompozytowej – zbudowanej z modułów
- Moduł – układ kwantowy będący częścią wyroczeni, którego działanie ma jasne przełożenie na efekt końcowy widoczny w postaci rozkładu prawdopodobieństwa wyników, np.:
 - **core** produkuje funkcję $\frac{1+\cos(nx+\lambda)}{2}$,
 - **add** – dodaje takie dwie funkcje
 - **mul** – mnoży je a
 - **eq** służy do produkowania wyjścia wyroczeni: TAK / NIE
- Biblioteka – powstaje rozszerzenie do QISKIT; być może do Q#
- Łatwiejsze tworzenie algorytmów kwantowych
- Umożliwi budowanie większych systemów – analogia do języków programowania wyższego poziomu abstrakcji
 - Analogicznie do przejścia z ASM na C/C++/C# ...

Wnioski - Kwantowa interpolacja

- Nasze podejście: Kalnau, K. Quantum interpolation of polynomials. Quantum Information & Computation 11, 1 (January 2011), 95-103

- Wejście – zbiór punktów

- Wymiarowość $\Rightarrow$ model

- Oracle: Ajoy, A., Yi-Yang, L., Saha, K., Marseglia, L., Jaskula, J.-C., Bissbort, U., Cappellaro, P.: Quantum interpolation for high-resolution sensing. Proceedings of the National Academy of Sciences Feb 2017, 114 (9) 2149-2153; DOI: 10.1073/pnas.1610835114

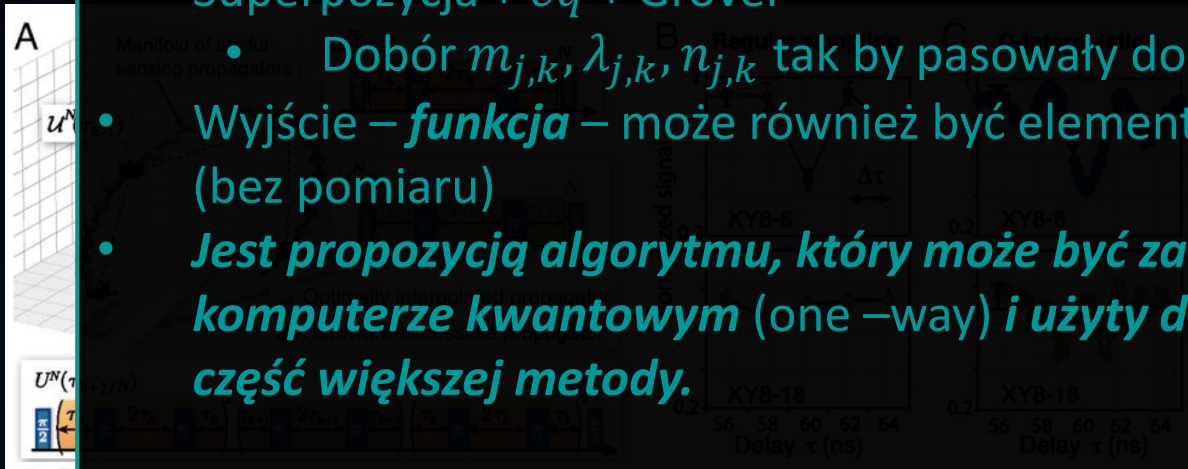
- $$f(x) = \sum_j \prod_k \left(\frac{1 + \cos(m_{j,k}x + \lambda_{j,k})}{2} \right)^{n_{j,k}}$$

- Superpozycja + *eq* + Grover

- Dobór $m_{j,k}, \lambda_{j,k}, n_{j,k}$ tak by pasowały do punktów wejściowych

- Wyjście – **funkcja** – może również być elementem dalszych obliczeń kwantowych (bez pomiaru)

- **Jest propozycją algorytmu, który może być zaimplementowany na dowolnym komputerze kwantowym (one –way) i użyty dla dowolnych danych również jako część większej metody.**



Wnioski - całkowanie

- **Nasze podejście:**
 - Wejście – znana funkcja wyrażona za pomocą wyroczni
 - **Może być wynikiem innych algorytmów (np. interpolacji)**
 - Oracle:
$$f(x) = \sum_j \prod_k \left(\frac{1 + \cos(m_{j,k}x + \lambda_{j,k})}{2} \right)^{n_{j,k}}$$
 - Superpozycja + pomiar (odpowiedni)
 - Wyjście – **wartość całki w zadanym przedziale** – może również być elementem dalszych obliczeń kwantowych (bez pomiaru)
 - Proponujemy **sposób aproksymacji funkcji do całkowania**
 - Przedstawiamy **kompletną budowę** algorytmu do całkowania
 - Celem jest **implementacja na komputerze kwantowym**

Operator

z Q_f definiują

postać funkcji

„przepisu” na nie.

Wynik jest sumą prawdopodobieństw dla kolejnych punktów

$$A(f)(C) = \sum_{\varphi(x_0, \dots, x_{k-1}) \in C} P_{A,f}(x_0, \dots, x_{k-1}).$$

Wnioski - klasyfikacja

Nasze podejście:

- Lokalność – trenowanie przeprowadzane na bieżąco w lokalnym otoczeniu punktu dla którego szukamy klasy. Superpozycja umożliwia takie podejście.
- Dlatego możliwa implementacja na współczesnych komputerach.
- Proponujemy nowy klasyfikator oparty o funkcję prawdopodobieństwa przynależności do klasy:
 - $f(x) = \sum_j \left(\frac{1 + \cos(x + \lambda_j)}{2} \right)^{n_j}$
 - Przyjmując 8 różnych wartości n_j dla zakodowania potrzeba 3 qubity + 1 na obliczenia – łącznie 4 qubity. Zatem możemy prowadzić obliczenia dla 4 punktów z otoczenia na 16 qubitach, 5 dla 20 qubitów.
- Implementacja SVM – w całości kwantowa:
 - $f(x, y) = 2\pi \int_{-1}^x \left(\frac{1 + \cos(nx + y)}{2} \right)^K dx$
 - Przeznaczając na K 3 qubity, na $n - 6$ zostaje 7 albo 11 qubitów na kodowanie punktów z otoczenia.
- Możliwość rozszerzania na więcej niż dwa wymiary – elastyczność.



Co przed nami?

- Prace implementacyjne na komputerze kwantowym (IBM-Q?)
 - dowolnej permutacji $\Rightarrow$ całkowanie (nowy kompozyt **perm**?)
 - interpolacji
 - Klasyfikacja (jw. + qSVM)
- Microsoft (?)
- Pomysły, pomysły:
 - Segmentacja zdjęć –(D-Wave vs IBM-Q)
 - Ray tracing
 - SIMONA CARAIMAN: QUANTUM COMPUTER GRAPHICS ALGORITHMS, Universitatea Tehnică „Gheorghe Asachi” din Iași Tomul LVIII (LXII), Fasc. 4, 2012
 - Współpraca z przemysłem ?

Literatura

- Masahito Hayashi:
 - Introduction to Quantum Information Theory
 - Quantum Information Theory
 - Group Theoretic Approach to Quantum Information
- Leonard Susskind:
 - YouTube: Lecture Collection – The Theoretical Minimum: Quantum Mechanics
 - YouTube: Advanced Quantum Mechanics
- Frederick Schuller
 - YouTube: Lectures on Quantum Mechanics – są do tych lekcji skrypty. Trudne.



Dziękuję za uwagę

PROSZĘ O KRYTYKĘ / PYTANIA

ZAPRASZAMY DO WSPÓŁPRACY 😊

