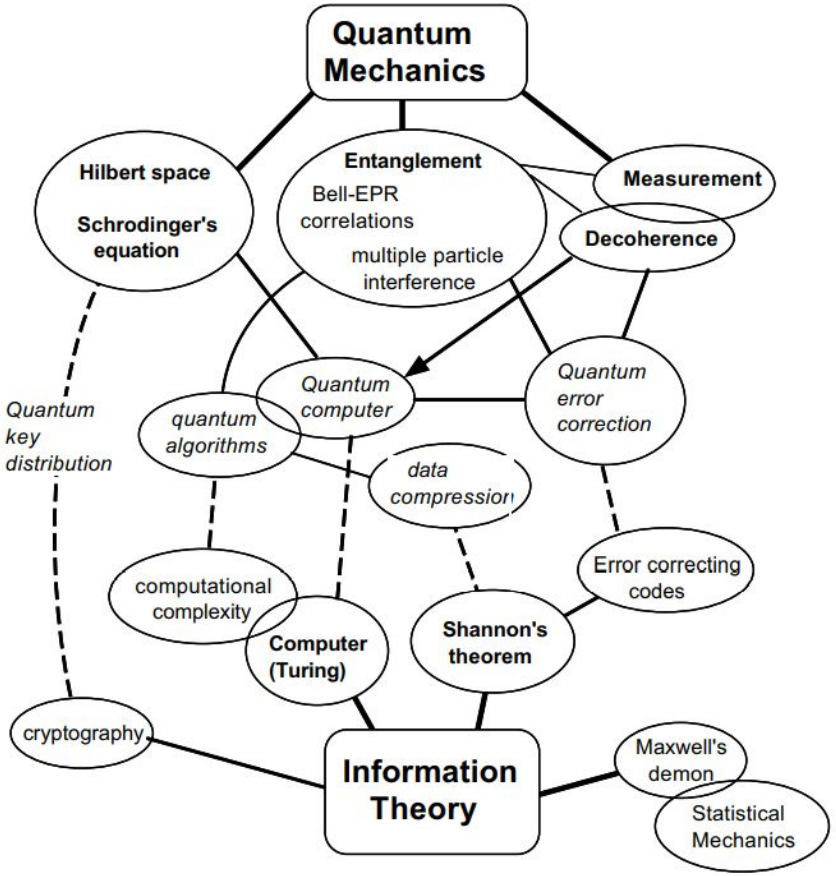
Présentation du TIPE

Informatique Quantique et une application à la cryptographie

Plan de présentation:

1. Introduction.
2. Notions fondamentales relatives à l’ordinateur quantique.
   a. Mécanique quantique.
   b. Informatique théorique.
3. Qubits, Portes Quantiques.
4. Paradoxe EPR, Théorème de Bell.
5. Algorithme de Shor.


Quantum Mechanics
Hilbert space
Schrodinger's equation
Entanglement
Bell-EPR correlations
multiple particle interference
Measurement
Decoherence
Quantum key distribution
quantum algorithms
Quantum computer
Quantum error correction
data compression
computational complexity
Computer (Turing)
Shannon's theorem
Error correcting codes
cryptography
Information Theory
Maxwell's demon
Statistical Mechanics

MCOT[6]

Mécanique Quantique

- Un système: totalement caractérisé par une fonction d'onde, ψ.
- On définit alors un espace d'états, dont les vecteurs sont notés avec la notation de Dirac:
- $|\psi\rangle$ ou "ket-psi".
- Espace d'état: espace de Hilbert.
- En ayant une base orthonormée de cet espace: $|\psi\rangle=\Sigma_k a_k|k\rangle$: superposition d' états.

- La projection orthogonale sur $|\psi\rangle$ est notée $\langle\psi|$ ou bien "bra-psi".
- $\langle\psi| = \Sigma_k a^*_k \langle k|$.
- Equation de Schrodinger:

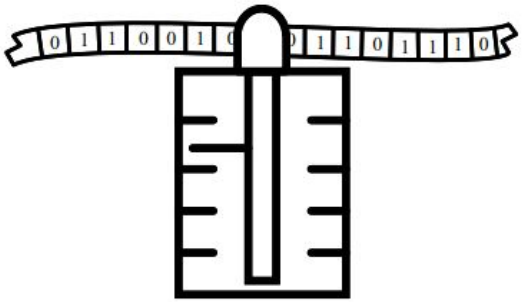
$$i\hbar\frac{d}{dt}|\Psi(t)\rangle = \hat{H}|\Psi(t)\rangle$$

- Tous les vecteurs d'états obéissent à cette équation fondamentale
- $|\langle\psi|\psi\rangle|^2=1$.
- Réunion de systèmes: produit tensoriel entre les états de bases des systèmes séparés donne la nouvelle base de cette réunion.

- Opérateurs Adjoints: U*: |x⟩,|y⟩, ⟨x|U|y⟩=(⟨y|U*|x⟩)*
- Opérateur Unitaire: U.U*=Identité
- Opérateur Autoadjoint: U=U*
- Ĥ est autoadjoint.
- Approximations à prendre vis-à-vis les systèmes réels.
- Principe d’incertitude de Heisenberg.
- Principe de réduction du paquet d’onde et problème de mesure.

Informatique théorique

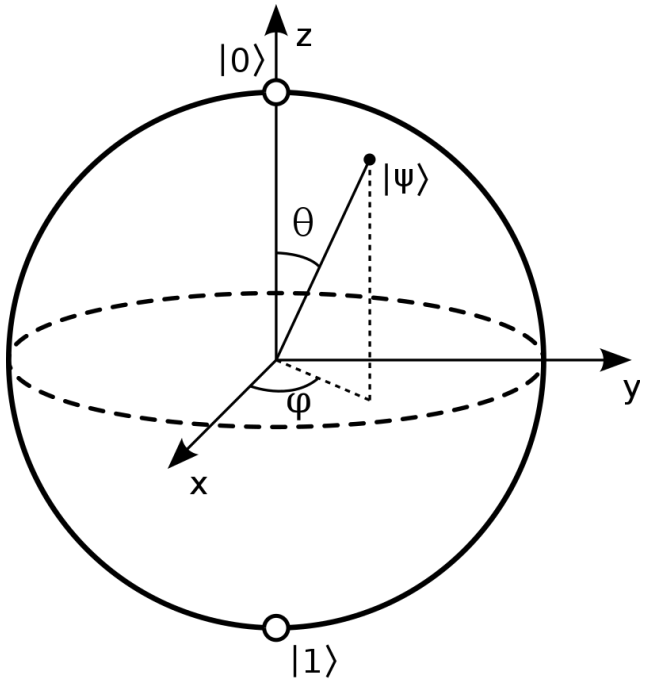
- Machine de Turing: un quintuplet (Q, Γ,q_0,F,□).
- Machine Universelle de Turing: simulation du comportement de toute autre machine de Turing.
- Complexité: nombre d'étapes pour terminer un algorithme.
- Classes: P, NP, NP-difficile, NP-complet.
- Question ouverte très importante: **P versus NP.**



MCOT [6]

Qubit, Portes Logiques

- Qubit: système à deux états (particule à spin demi-entier par exemple).
- $|0\rangle$ et $|1\rangle$: Etats de base (base orthonormée de l'espace des états). Donc un qubit est $|\psi\rangle=\alpha|0\rangle+\beta|1\rangle$, normalisé.
- Représentation dans une sphère de Bloch: $|\psi\rangle=\cos(\theta)|0\rangle+e^{i\phi}.\sin(\theta)|1\rangle$:
  Premières différences avec un bit classique.



- La réunion de deux qubit est un système à quatres états, donc la réunion de n-qubits est un système à 2^n états: registre quantique.
- $|\psi\rangle=\Sigma_k \alpha_k |k\rangle$: Un registre quantique contient une information sur 2^n états, mais pas toute cette information est accessible lors d'une mesure.
- 2 qubits ayant chacun des états de base $|0\rangle$ et $|1\rangle$, leur réunion est dans les états $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$, ou les produits de Kroenecker possibles entre $|0\rangle$ et $|1\rangle$.

- Portes quantique: des opérateurs unitaires qui agissent sur un ou plusieurs qubits (équivalents des portes logiques classiques).
- Classiquement, NOT est l'unique porte unaire. Quantiquement, Il en existe plusieurs: Pauli-X; Pauli-Y; Pauli-Z; Phase Shift; Hadamard…
- Si un qubit est $|\psi\rangle=\alpha|0\rangle+\square|1\rangle$, donc la colonne $\begin{pmatrix}\alpha\\ \beta\end{pmatrix}$,
- Pauli-X: possède l'action d'un NOT, Hadamard: Transformation de base.

Pauli-X: $\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$

Pauli-Y: $\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}$

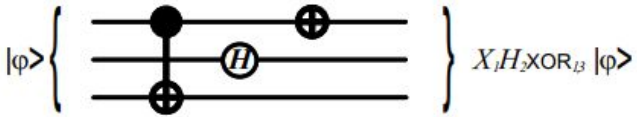
Pauli-Z: $\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$

Hadamard: $(1/\sqrt{2}).\begin{pmatrix}1 & 1\\ -1 & 1\end{pmatrix}$

- Portes contrôlées:
  Portes binaires, donc des matrices 4*4.
- U porte quantique unaire, Controlled-U:

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes U$$

- U=Pauli-X, alors C-U est appelée C-NOT.



MCOT[6]

Propriété sur les qubits: pas de clonage.

- Opérateur d'évolution U(t,t0).
- A un système, $|\psi\rangle$ son état initial.
- B un autre système dans le même espace d'états, $|0\rangle$ son état initial. Donc l'état initial de l'ensemble est $|\psi\rangle|0\rangle$.
- Cloner $|\psi\rangle$: $U.\ |\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$
- Mais cette évolution serait indépendante de ψ, donc pour $|\psi'\rangle$, $U.|\psi'\rangle|0\rangle=|\psi'\rangle|\psi'\rangle$
- Pour l'état $|\psi''\rangle=(1/\sqrt{2}).(|\psi\rangle+|\psi'\rangle)$: $U|\psi''\rangle|0\rangle=(1/\sqrt{2}).(|\psi\rangle|0\rangle+|\psi'\rangle|0\rangle)$ qui n'est pas $|\psi''\rangle|\psi''\rangle$.
- Ainsi cloner un état inconnu est impossible.

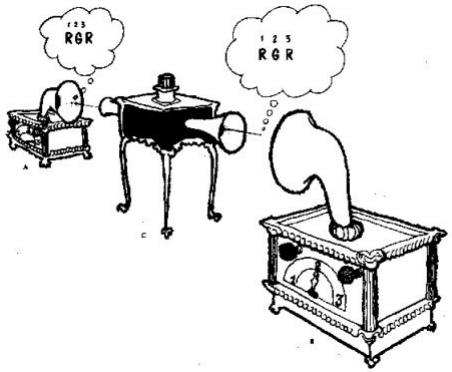
- Si une évolution permet de cloner un certain état, alors on peut montrer que les autres états clonables seront soit identiques, soit orthogonaux, de toute façon connus.

Paradoxe EPR, Théorème de Bell.

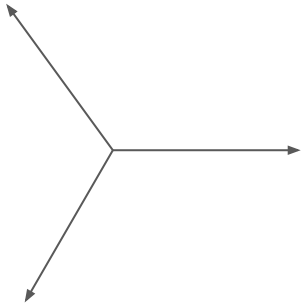
- Paradoxe EPR: un singulet (intrication), on sépare les électrons et on les lance chacun vers un récepteur distinct (électron A vers Alice, électron B vers Bob).
- $|\psi\rangle=(1/\sqrt{2}).(|\uparrow\downarrow\rangle-|\downarrow\uparrow\rangle)$, $\uparrow$ désigne un spin dans le sens positif, $\downarrow$ un spin dans le sens négatif.
- Hypothèses de réalisme et localité.

- Une mesure du spin de A selon une direction, révèle que le spin B est dans dans la même direction dans le sens opposé.(*)
- Les composantes du spin (par exemple Sx et Sz dans la base (**x,z)**) ne sont pas compatibles: le principe de Heisenberg s'applique.
- Alice mesure son électron: l'information (*) pour l'électron de Bob doit être transmise.
- A une distance assez éloignée, l'information passe plus rapidement que la lumière.

- Proposition d'Einstein: incomplétude de la théorie, nécessité de variables locales cachées.
- Bell: La mécanique quantique ne peut pas obéir à des variables cachées locales:
- On fournit à Alice et à Bob 3 axes dans le plan (**x,z**) (**y** est l'axe de propagation des électrons) dont l'angle entre chaque 2 est de $2\pi/3$.
- Chacun mesure suivant un axe indépendamment des choix de l'autre, signale "+" si le spin est UP suivant cet axe, "-" sinon.



MCOT[7]



- Pour l'hypothèse des variables cachées locales: le choix des spin suivant chaque axe s'effectue au moment de séparation des électrons.
- On traite les cas où "A" a décidé (↑↑↑) et (↑↑↓), B est son opposé à chaque fois.
- Dans le premier cas, il n'y a pas de chances d'avoir Alice et Bob choisir des signes égaux, dans le second cas, seulement 4/9 de fois est le cas. En tout, il est moins probable d'avoir des signes égaux qu'avoir des signes opposés.
- Pour le travail de Bell: P("avoir des signes égaux")=P("avoir des signes égaux et Alice et Bob sélectionnent des axes différents). Vu que les axes sont interchangeables:

- P("avoir des signes égaux")=6xP("avoir des signes égaux et Alice sélectionne l'axe 1 et Bob sélectionne l'axe 2") .
  Donc, en notant p=P("avoir des signes égaux"):
  p=6xP("Alice sélectionne 1, Bob sélectionne 2")xP("avoir des signes égaux"|"Alice sélectionne 1, Bob sélectionne 2")
  P("Alice sélectionne 1, Bob sélectionne 2")=1/9, et P("avoir des signes égaux"|"Alice sélectionne 1, Bob sélectionne 2") est prouvable ne dépendre que sur l'angle entre les deux axes et vaut $\sin^2((\theta_1-\theta_2)/2)$, donc ¾ dans ce cas. p=1/2

- Cette différence dans la valeur de p indique évidemment que la mécanique quantique ne peut pas obéir aux variables cachées comme les définissent Einstein.
- Hypothèses: Interprétation de Copenhague, Variables cachées non locales, l'information traverse l'axe du temps du futur vers le passé…
- Mise en évidence que les ordinateurs classiques n'espèrent pas simuler des phénomènes qui violent l'inégalité de Bell: donc les ordinateurs quantiques sont dans certaines application plus puissants.

$$1 + P(\vec{b}, \vec{c}) \geq |P(\vec{a}, \vec{b}) - P(\vec{a}, \vec{c})|$$

Algorithme de Shor

- Algorithme qui permet une décomposition d'un entier N en facteurs premiers en un temps polynomial en log(N): les cryptages à base clé publique deviendront vulnérables
- RSA: méthode de cryptage antisymétrique. Clé publique constituée d'un entier produit de deux nombres premiers.
- Décryptage de RSA est très difficile pour les ordinateurs classiques, mais plus efficace pour un ordinateur quantique.

- 2 parties: classique et quantique
- Classiquement:
  un choix aléatoire de $x<N$, calcul du PGCD(N,x), intérêt au cas ou N et x sont premiers entre eux, l'ensemble des entiers inférieurs à N et premiers avec lui (mod N), essentiellement Z/NZ* est un groupe pour la loi ., c'est aussi fini, donc l'ordre de x existe. La recherche de cet ordre est l'étape quantique.

- Supposons trouvé l'ordre s'il est pair: $(x^{r/2}-1).(x^{r/2}+1)$ est un multiple de N. Le faite que r est l'ordre de x dans Z/NZ*, N ne divise pas $x^{r/2}-1$. Si $x^{r/2}+1$ n'est pas un multiple de N, alors on est certain que pgcd(N, $x^{r/2}-1$) ou pgcd(N,$x^{r/2}+1$) est non trivial. Dans les cas d'échec on retourne à l'étape de la sélection de x.
- Dans le cas de RSA, on sait que N=p.q, p et q deux entiers premiers, alors les diviseurs non triviaux de N sont p et q, ainsi trouver r dans les bonnes conditions permet à accéder p et q.

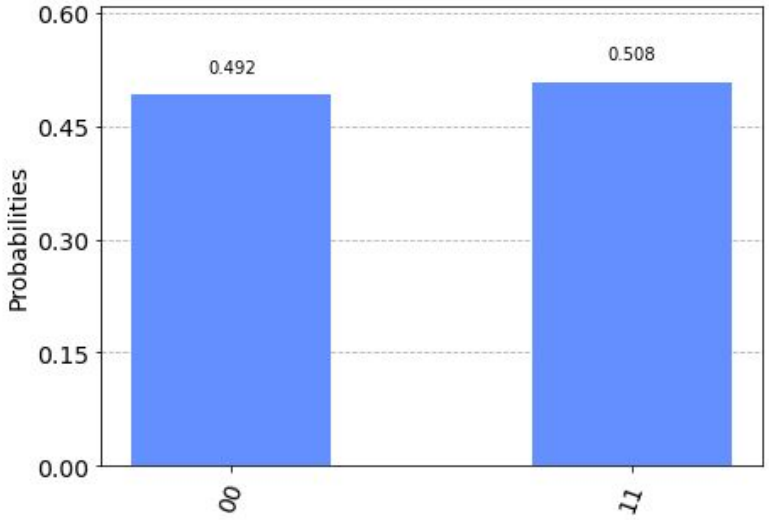
- Partie quantique:
  Recherche de l'ordre est la recherche de la période de f: $y \rightarrow x^y$ modN.
- Transformée de Fourier quantique: pour un entier q ayant des facteurs premiers petits (le travail avec des qubits inspire l'utilisation de $q=2^n$).
  $U_{QFT}|a\rangle=(1/q^{1/2}).\sum_0^{q-1}\exp[(2i\pi ka)/q]]|k\rangle$.
  Cet opérateur est unitaire, et Shor prouve que son implémentation polynomiale sur un ordinateur quantique est possible[8].

- Soit f notre fonction de période r qu'on souhaite déterminer. On considère pour un certain N, q tel que $q>N^2$ (On peut prendre $q=2^n$, $n=E(2.\log(N))$.
- 2 registres quantiques chacun de n qubits sont créés. Le système total des deux registres est initialisé à $(1/\sqrt{q}).\sum_0^{q-1}|l\rangle|0\rangle$.
- (*)Evoluer le système, vers l'état $(1/\sqrt{q}).\sum_0^{q-1}|l\rangle|f(l)\rangle$, (on suppose que f est bijective sur [k.r, k+1r[, ce qui est le cas pour notre fonction).
- (**)Appliquer de la transformée de Fourier quantique sur les $|l\rangle$. Résultat: $(1/q).\sum_{l=0}^{q-1}\sum_{k=0}^{q-1}\exp(2ikl\pi/q)|k\rangle|f(l)\rangle$
- On peut regrouper les f(l) selon les valeurs que f peut prendre (entre 0 et N-1). Résultat: $(1/q).\sum_{j=0}^{N-1}\sum_{k=0}^{q-1}[\sum_{f(l)=j}\exp(2ikl\pi/q)]|k\rangle|j\rangle$
- Soit l_{0j} le plus petit entier tel que f(l)=j. On aura que l_{0j} est inférieur à r. f(l)=j si et seulement si $l=l_{0j}+b.r$, et vu que l est entre 0 et q-1, b est entre 0 et $(q-l_{0j}-1)/r$. Ainsi, pour $p=E[(q-l_{0j}-1/r)]+1$, la somme se réécrit: $(1/q).\sum_{j=0}^{N-1}\sum_{k=0}^{q-1}[\sum_{l=0}^{p-1}\exp(2ik(l_{0j}+l.r)\pi/q)]|k\rangle|j\rangle$.
- On effectue une mesure sur les deux registres, la probabilité d'obtenir un certain état $|k\rangle|j\rangle$ est: $1/q^2|\sum_{l=0}^{p-1}\exp(2ik(l_{0j}+l.r)\pi/q)]|^2$

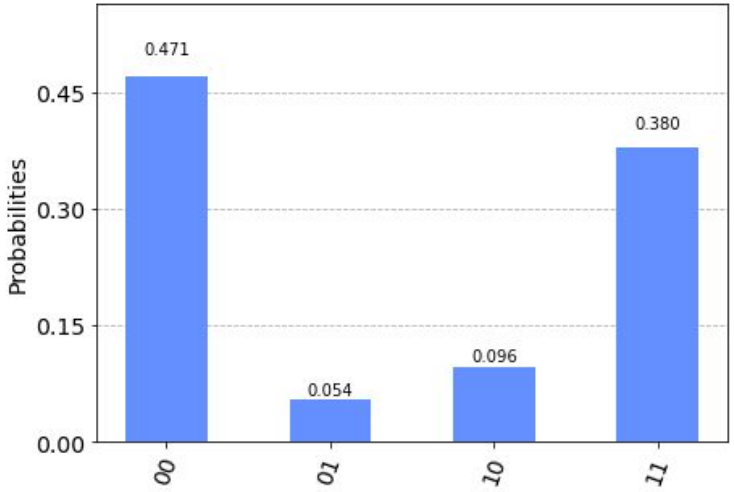
- La quantité est une somme géométrique qui se simplifie en: $(1/q^2).[\sin(p.r.k.\pi/q)/\sin(r.k.\pi/q)]^2$.
- Cette quantité est la plus grande lorsque r.k/q est un entier, ou proche d'un entier. Ainsi rk/q est proche d'un certain entier, c, d'où k/q est proche de c/r. En cherchant alors des rationnels proches de k/q irréductibles, on obtient des candidats de r(les dénominateurs de ces rationnels), et on vérifie classiquement si ces candidats ou leurs multiples sont les périodes de f.
- L'échec de l'algorithme pousse à revenir à l'étape de sélection de x.

Implémentation de l’algorithme de Shor: qiskit

- qiskit: une librairie python qui permet la construction de circuit quantiques, les exécuter soit sur un simulateur virtuel (qasm_simulator), ou sur la plateforme IBM des ordinateur quantiques cloud.



Exécution avec qasm_simulator



Exécution avec une machine IBM

- L’algorithme de Shor sur qiskit est préexistent sous qiskit.aqua.algorithms, donc implémentable facilement.
- La manière dont on peut manuellement l’implémenter nécessite la construction d’une évolution comme (*), ainsi que la transformée de Fourier(**).
- Première étape: l’étape la plus coûteuse et l’implémenter dépend parfois du paramètre x.
- Seconde étape: transformée de Fourier quantique: responsable pour l’accélération exponentielle de la recherche de période.
- Pour implémenter cette transformée, il faut examiner la formule pour dégager les composantes du circuit à employer.

- $U_{QFT}|a\rangle=(1/q^{1/2}).\sum_0^{q-1}\exp[(2i\pi ka)/q]|k\rangle$.
- $k=[(k_j)]$, j parcourant le nombre de bits n de q et les k_j sont des zéros et des 1.
- $k=\sum_1^n k_j 2^{n-i}$
- $|k\rangle$ est pratiquement le produit tensoriel de tout les $|k_i\rangle$ dans l’ordre de 1 à n.
- Donc injectant cette forme de k dans l’exponentielle et dans les vecteurs d’état, ainsi que transformer la somme de 0 à q-1 à n sommes des k_j de 0 à 1, on sépare chaque qubit j à un état qui serait: $(½)^{½}.(|0\rangle+\exp[(2\pi i.a)/2^j]\ |1\rangle)$
- En écrivant $a=\sum_1^n a_m 2^{n-m}$, l’exponentielle devient un produit d’exponentielles, où figurent les a_m à partir de j.

- Pour construire ce circuit, il faut utiliser deux portes quantiques: une porte de Hadamard qui devrait transformer un qubit $|a_j\rangle$ en $(½)^{½}.(|0\rangle+\exp[(2i\pi a_j)/2]|1\rangle)$
- Une suite de portes de déphasages contrôlés par les a_m pour m>j, le déphasage étant de $2\pi/2^{m-j+1}$
- Ceci donne par exemple pour a_1 la transformation nécessaire à a_n. Donc aussi il faut inverser les qubits j et n-j+1

Observations et conclusion

- L’algorithme de Shor accélère exponentiellement la rapidité du calcul des périodes de fonctions, mais pas encore pratique.
- Reflète l’état de l’informatique quantique en ce moment: puissante mais limitée par la faisabilité moderne.

Annexe de codes:

Code utilisé pour les 2 premiers graphes:

```
from qiskit import *

reg_q=QuantumRegister(2)

reg_cl=ClassicalRegister(2)

circuit=QuantumCircuit(reg_q,reg_cl)

circuit.h(reg_q[0])

circuit.cx(reg_q[0],reg_q[1])

circuit.measure(reg_q,reg_cl)

simulator=Aer.get_backend('qasm_simulator')

result = execute(circuit, backend=simulator).result()

from qiskit.tools.visualization import plot_histogram

plot_histogram(result.get_counts(circuit))
```

```
IBMQ.load_account()

provider=IBMQ.get_provider('ibm-q')

qcomp=provider.get_backend('ibmq_16_melbourne')

job=execute(circuit, backend=qcomp)

from qiskit.tools.monitor import job_monitor

job_monitor(job)

result=job.result()

plot_histogram(result.get_counts(circuit))
```

L’implémentation de QFT:

```
from qiskit import *
import numpy as np
def QuantumFourier(n):
    circuit= QuantumCircuit(n)
    for q in range(n):
        circuit.h(q)
        for q_dapres in range(q+1,n):
            circuit.cp(np.pi/2**q_dapres-q, q_dapres, q)
    return circuit
display(QuantumFourier(4).draw())
```