

Plan

- 40 Programmation dans l'environnement Pulser de PASQAL

100

100

- importance du phénomène de *blocage de Rydberg*
- une centaine d'atomes utilisables dans la version livrée au CEA, sur une structure plane

0.1

THESE ARE THE TERMS AND CONDITIONS OF THE SALE:

Concepts mathématiques - théorie des graphes

Le problème que peut traiter la machine est PASQAL est le *Maximum Independant Set*

- considérons un graphe $(G; E)$
 - G est un ensemble de points, ou sommets (ou *vertices*) dans un espace à n dimensions
 - E est un ensemble de segments, ou arêtes (ou *edges*) qui relient deux points de G
- on cherche le, ou les, plus grand(s) ensemble de sommets dont les membres ne sont pas connectés entre eux.

Dans le cas des machines PASQUAL, on considère des graphes **unitaires**

- les sommets dont la distance est inférieure à un rayon R donné sont connectés
- deux sommets connectés sont forcément plus proche que R

Le problème MIS est dual du problème MCS (*Minimum Converging set*)

- identifier l'ensemble le plus petit possible de sommets où poser des "caméras" pour voir toutes les arêtes du graphe
- on démontre facilement que les solutions de MIS et MVS sont complémentaires

Les bases physiques

Pour résoudre MIS, la machine PASQAL met en oeuvre différents mécanismes

- des atomes de rubidium monovalents émis par une ampoule dans une chambre à vides, ralentis et positionnés par des lasers
- le niveau de valence le plus bas et le niveau le plus haut encode les états $|0\rangle$ et $|1\rangle$
- les atomes sont intriqués via le phénomène de blocage de Rydberg (*Rydberg blockade*)
- on sait éjecter les atomes dans l'état $|1\rangle$ (mais on sait parfaitement où ils étaient)
- on sait voir les autres par fluorescence, on sait donc mesurer les états de chaque atome

Par ailleurs

- ne sont intriqués que des atomes dont la distance est inférieure au rayon de Rydberg
- deux atomes intriqués sont dans l'état $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$

Utilisation de lasers

La machine PASQAL utilise les lasers pour deux propos

- ralentir et positionner les atomes de rubidium émis par l'ampoule
- faire sortir les atomes de l'état $|0\rangle$ et déclencher des blocages de Rydberg

L'hamiltonien qui correspond au système est formellement très proche de la formulation du problème de Ising.

Programmation Pulser

La programmation avec le framework Pulser en Python est assez simple

- On calcule le rayon de Rydberg
- on place les atomes pour qu'ils soient les sommets d'un graphe uniaire dont le rayon correspond au rayon de Rydberg
- on décrit la forme d'une émission laser globale qui va éclairer tous les atomes et initier des blocages de Rydberg
- tire les lasers à plusieurs reprises
- on réalise les mesures pour faire ressortir les états les plus probables finals

Exemple de code - Initialisation

On initialise l'environnement de programmation comme suit

```
import numpy as np
from pulser import Pulse, Sequence, Register
from pulser_simulation import Simulation
from pulser.devices import MockDevice
from pulser.waveforms import RampWaveform, ConstantWaveform

import matplotlib.pyplot as plt
import qutip
```

Exemple de code - description du graphe

```
# Define a dictionary where each key is the name of the qubit,
# and each value is the qubit's position (in um)
```

```
qubit_positions = {
    'q0': (0, 0),
    'q1': (3, 5.2),
    'q2': (6, 0),
    'q3': (9, -5.2)
}
```

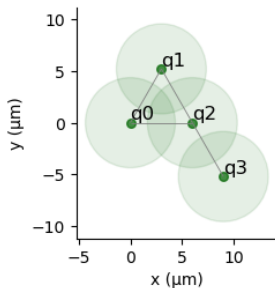
Ce dictionnaire est ensuite transformé en registre.

```
# Arrangements of qubits on the machine are called a register
# Define a register in Pulser by passing the qubit dictionary
reg = Register(qubit_positions)
reg.draw()
```


References

13 1 1 11 2 5

1 13 1 1 11 13 1 1 11 1



Définir l'impulsion laser

```
# A Sequence is the object that contains all
# the info about the quantum evolution
seq = Sequence(reg, MockDevice)

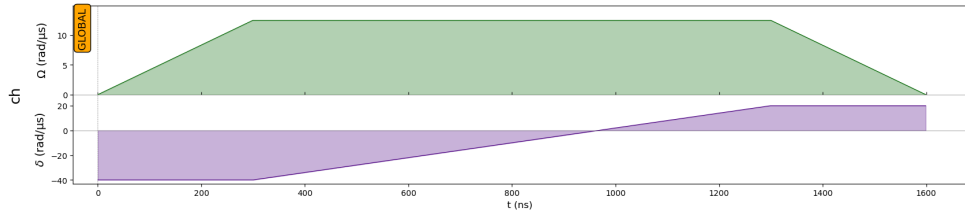
# Now we want to fill the channel with pulses
# First we need to define the waveforms for the pulses
# First ramp
omega_wf_1 = RampWaveform(300, 0, Omega_max)

#arguments are duration (ns), detuning (rad/us)
delta_wf_1 = ConstantWaveform(300, -40)

first_pulse = Pulse(omega_wf_1, delta_wf_1, 0)
seq.add(first_pulse, 'ch')
seq.draw()
```

Forme de l'impulsion

L'impulsion laser aura ainsi la forme suivante



Pulser - fin du code

```
# The sequence is ready now for simulation

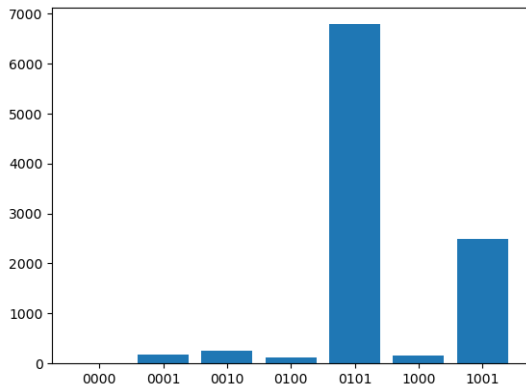
sim = Simulation(seq)
results = sim.run()

# The result can be sampled
samples = results.sample_final_state(10000)

# And the sampling can be visualized
plt.bar(samples.keys(), samples.values())
```

Résultat

On voit le résultat suivant



On voit deux solutions : $|0101\rangle$ et $|1001\rangle$. ce sont les solutions du problème MIS.

Les problèmes NP-Complets

Certains problèmes NP-Complets sont particulièrement d'un point de vue de l'informatique quantique

- le problème d'optimisation QUBO (voir slide suivant)
- le problème **MaxCut** (couper un graphe en deux ensembles complémentaires qui maximise les coupes des arêtes)
- le problème **Maximum Independant Set (MIS)** et son dual le problème **Maximum Vertices Cover (MVC)**
- les problèmes de coloration de graphes
- le problème SAT et sa variante **3-SAT** (satisfaisabilité booléenne)
- le problème du voyageur de commerce (**Travelling Sales Person (TSP)**) et le problème du *cycle hamiltonien*
- les problèmes de détection de clique maximum
- le problème du sac à dos

Le problème QUBO

L'acronyme *QUBO* signifie *Quadratic Unconstrained Binary Optimisation*. Il permet de résoudre des problèmes d'optimisation qui se ramènent, à la recherche d'optima d'une forme quadratique.

L'énoncé de QUBO est le suivant : étant donné un entier $n \in \mathbb{N}$, étant donné $\mathbb{B}^n = \{0; 1\}^n$, l'ensemble des vecteurs de taille n formés de 0 et de 1, étant donné une forme quadratique f_Q représentée par une matrice $Q \in \mathbb{R}^{n \times n}$, quel est la valeur $x^* \in \mathbb{B}^n$ qui minimise $f_Q(x) = x^T \cdot Q \cdot x$. D'une manière générale, on ajoute parfois au terme quadratique un terme linéaire, l'énoncé devient alors

$Q \in \mathbb{R}^{n \times n}, c \in \mathbb{R}^n$, trouver la valeur x^* qui minimise $f(x) = x^T \cdot Q \cdot x + c^T \cdot x$

Fonctionnalités futures

Dans les prochaines versions de leur hardware, PASQAL sera capable

- de gérer plus d'atomes (quelques centaines)
- de disposer les atomes en 3 dimensions (actuellement on fait juste de la 2D)
- d'éclairer certains atomes différemment les uns des autres

Ce dernier point est important, il permet d'implémenter le problème *Weighted MIS (MWIS)* ou "MIS à poids".

La formulation MWIS est important : un problème en $O(N^6)$ en MIS devient $O(N^2)$ en MWIS. Il est nécessaire de mettre en place beaucoup de middleware pour bien exploiter cette conversion.