Pasqal

Optimization use cases with Pasqal software stack

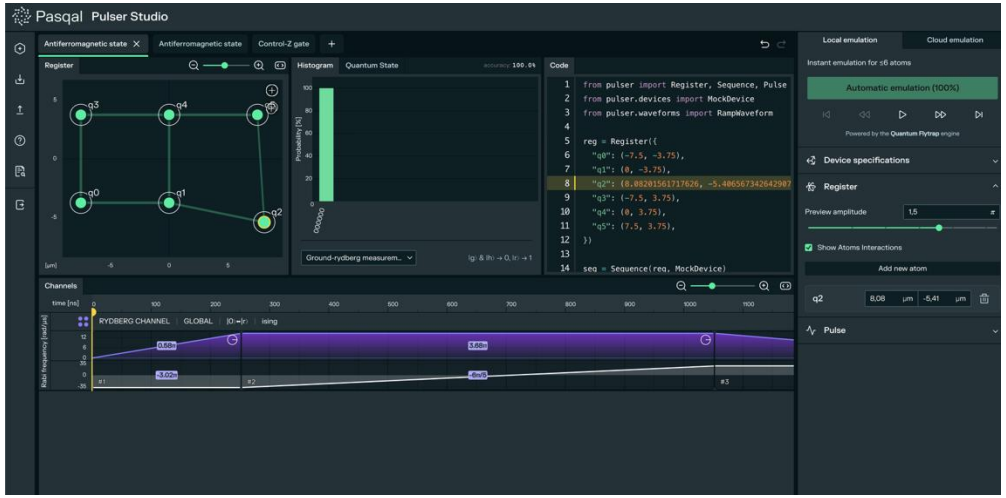


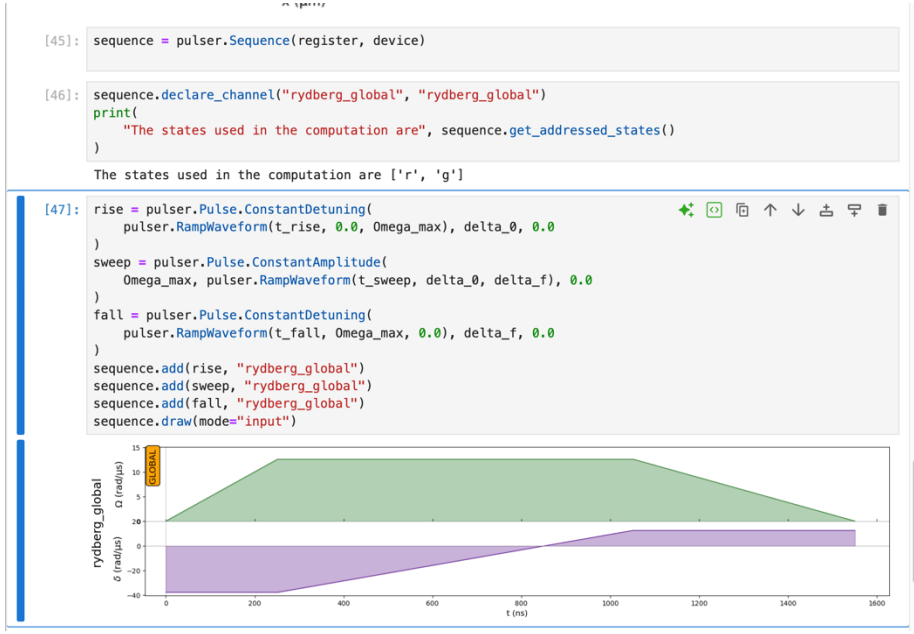
Goals of this session

Pasqal

- Use the full Pasqal software stack on optimization problems
- Understand what can be done and how to do it
- See that quantum computing can have practical and useful applications
- Links to the notebook and the doc at the end

Multiple ways to experiment





Agenda

Pasqal

01

Qoolqit and Pulser side by side example (QUBO)

02

Pulser with a specific device for MWIS

03

QUBO and MIS libraries

Pasqal Defining the quantum reality

Qoolqit and Pulser side by side example (QUBO)

Reminders

Pasqal

Pulser

- Low level
- More complicated
- High degree of control and customization
- Physical unit for parameters

Qoolqit

- Higher level
- Easier to use
- Some degree of automatisation
- Adimensionnality

Reminder: Quadratic Unconstrained Binary Optimisation (QUBO)

Pasqal

A QUBO problem is a combinatorial optimization problem with a wide range of applications from finance and economics to machine learning.

Mathematically, a QUBO instance consists of a symmetric matrix Q of size $N * N$, and the optimization problem associated with it is to find the bitstring $z = (z_1, \dots, z_N) \in \{0,1\}^N$ that minimizes the quantity $\mathrm{f(z)} = z^T Q z$

QUBO is a structurally simple, yet computationally hard optimization problem (NP hard), once again complicated to solve with classical computers.

Some forms of a QUBO can be naturally encoded in Pasqal's hardware

Constraints to encode a QUBO problem on a neutral atom machine

Pasqal

In this tutorial, we will demonstrate how this QUBO instance can be mapped and solved using neutral atoms with pulser and qoolqit with the following matrix:

```
Q = np.array(
    [
        [-10.0, 19.7365809, 19.7365809, 5.42015853, 5.42015853],
        [19.7365809, -10.0, 20.67626392, 0.17675796, 0.85604541],
        [19.7365809, 20.67626392, -10.0, 0.85604541, 0.17675796],
        [5.42015853, 0.17675796, 0.85604541, -10.0, 0.32306662],
        [5.42015853, 0.85604541, 0.17675796, 0.32306662, -10.0],
    ]
)
```

This QUBO instance is particularly amenable to embedding on a neutral-atom device since:

- All the off-diagonal terms are positive : the off-diagonal terms are encoded via the Rydberg interaction between atoms, thus this interaction can only be positive or zero. Work around are possible when negative terms.

- The diagonal terms are all equal : the diagonal terms of Q are understood as a detuning (δ channel mentionned in previous webinars) to apply on each atom. Since the diagonal terms are all the same, we can use the same detuning for all the atoms. If that was not the case, some kind of local addressability capabilities would be necessary (see second tutorial)

With Pulser

Pasqal

With Qoolqit

Pasqal

Pasqal Defining the quantum reality

Pulser with a specific device for MWIS

Introduction of the tuto

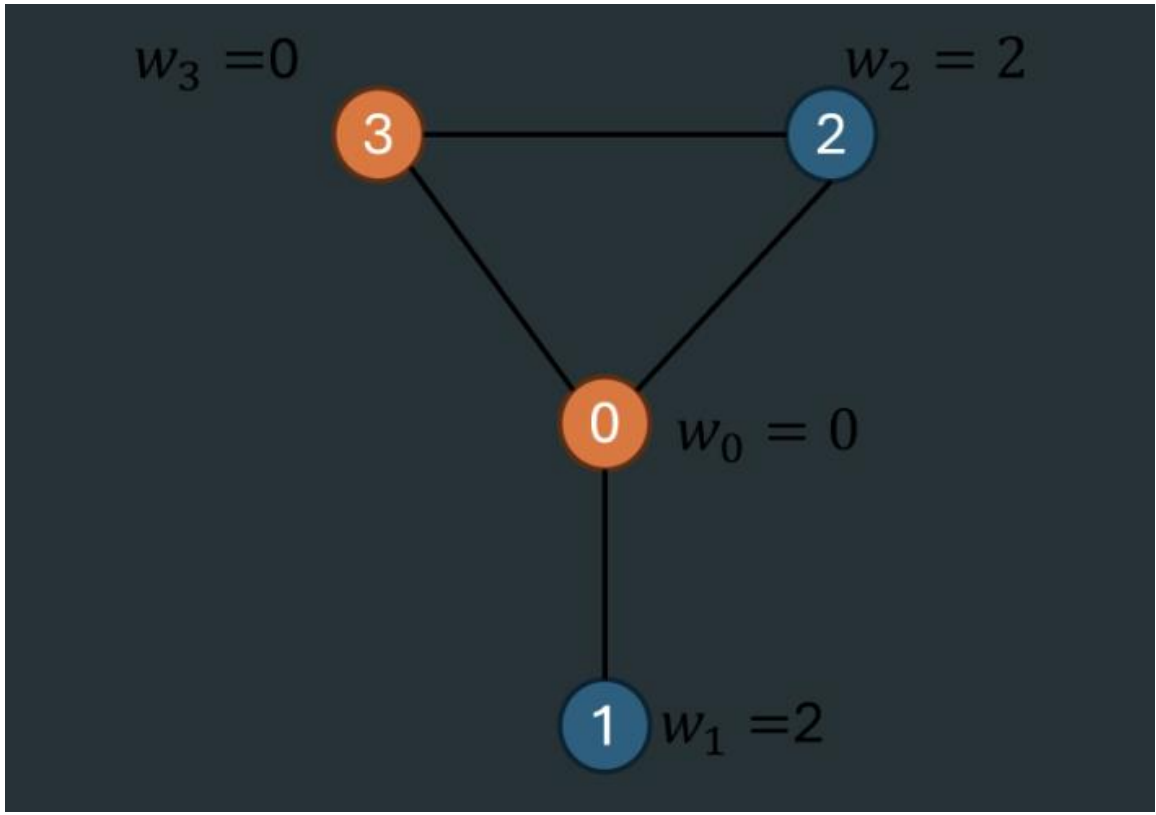
Pasqal

In this tutorial, we illustrate how to solve a Maximally Weighted Independent Set (MWIS) problem using an ensemble of Rydberg atoms in analog mode.

MWIS arise in many applications, including resource allocation, scheduling and staffing problems, error-correcting coding, complex system analysis and optimization, logistics and transportation, communication networks…

In the MWIS problem, we consider an undirected graph with nodes and edges where each node is associated with a positive weight. The problem is to find a maximum weighted independent set, i.e., select a set of nodes in graph where:
-there is no edge between any pair of selected nodes.
-the sum of the weights of the nodes for this set is maximised.

What is the MWIS here?



Pasqal Defining the quantum reality

QUBO library

Features in development

Pasqal

- The two libraries that we are going to see were released in October and are not final products. There might be issues when using them, if you encounter any please notify us, for example via the Pasqal community.

- The QUBO library is more developed and fully fledged than the MIS one (0.5… vs 0.3…). As such it is more user friendly and should be considered first when trying to do your first experiments.

- Contrary to Qoolqit and Pulser, these libraries won't be available on the Ruby machine right now. It indeed necessitates some softwares updates validated by TGCC.

- You can still test them locally on your machine to see what the emulators are capable of. If you feel you would like to use them on the Ruby machine please tell us! Your feedback helps us set priorities in features.

Examples of the QUBO solver

Pasqal

Useful links

Pasqal

- Link to the gitub with the note books used during the session https://github.com/VgfMartin/Pasqal-software-stack-tutorials/tree/main/Optimisation%20webinar%20(formation%20GENCI%204)
- Pulser doc https://docs.pasqal.com/pulser/
- Qoolqit doc https://docs.pasqal.com/qoolqit/
- QUBO solver libray doc https://pasqal-io.github.io/qubo-solver/latest/tutorial/00-a-tour-of-qubo/