

AQuRA

Development of an analog quantum computer

We design a novel analog quantum computer (AQuRA) based on continuous quantum variables. For this purpose, we develop the concept and the quantum algorithms for AQuRA together with a corresponding simulation software for classical computers.

- Analog computing
- Quantum simulations
- Modeling continuous quantum systems
- Optimization problems

Continuous-variable quantum computing

Quantum computing with continuous quantum systems is based on manipulating and measuring real-valued properties like the position/ momentum of a particle or the quadratures of light.

Compared to discrete variables, they may offer some advantages in terms of:

- Information content of a single quantum state
- Calculations with real-valued numbers

In the project **AQuRA**, we investigate the encoding of several qubits into a single continuous-variable quantum state and its utilization for quantum computing aside or together with discrete quantum states. Together with our industry partner anabrid, we aim to bring the classical analog computing into the quantum domain.

Concept

- **Conceptual design** of an analog quantum computer based on **continuous quantum variables**.

Applications

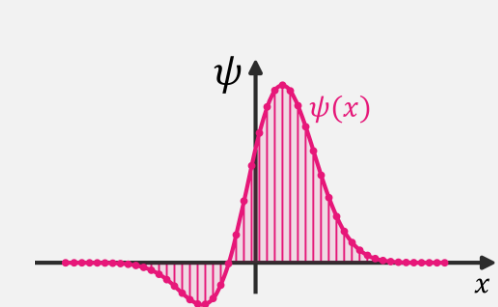
- Development of suitable **quantum algorithms**.
- **Identification of potential applications**.

Software

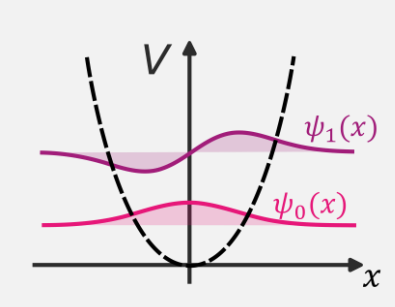
- **Software development** and **simulation** of AQuRA with classical computers and HPC-Systems.

Available basis states

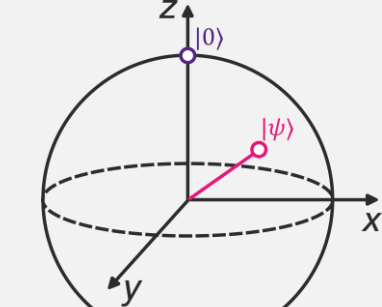
Position basis



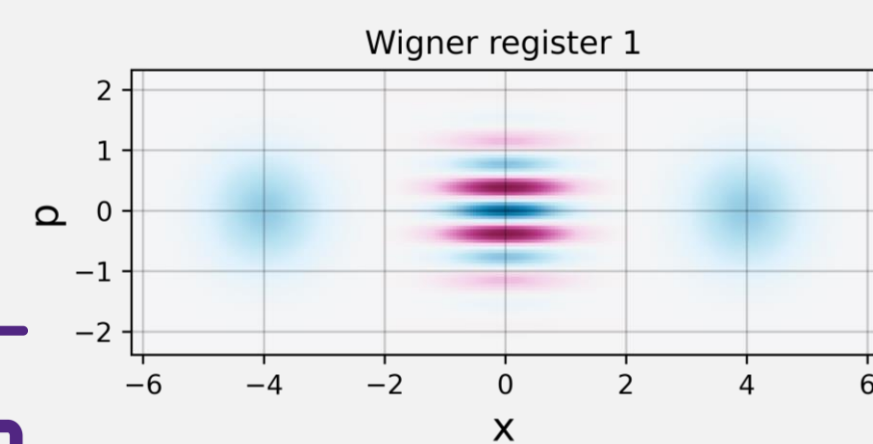
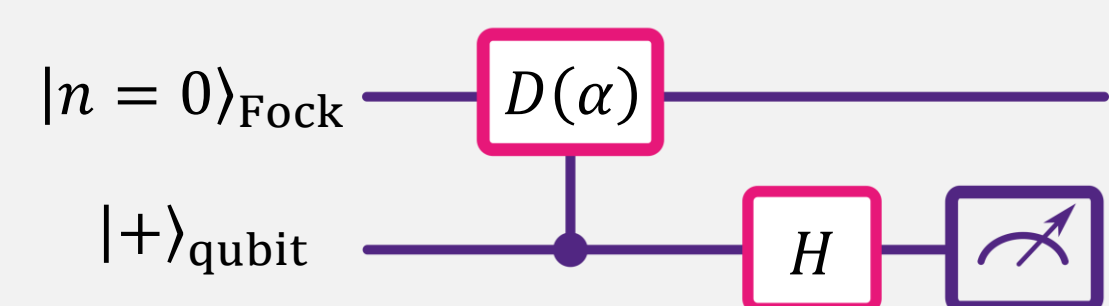
Fock basis



Qubit basis



Exemplary algorithm



```
import aqura as aq
import numpy as np
```

```
qc = aq.QuantumCircuit()
```

```
qc.add_register(aq.FockRegister(200))
```

```
qc.add_register(aq.QubitRegister())
```

```
qc.add_operation(aq.FockInitOperation(), [0])
```

```
qc.add_operation(aq.QubitInitOperation(aq.Qubit.Plus), [1])
```

```
qc.add_operation(aq.DVControlledDisplacement(4/np.sqrt(2)), [1, 0])
```

```
qc.add_operation(aq.HGateOperation(), [1])
```

```
meas = qc.add_measurement(1)
```

```
# Initialize Quantum Circuit
```

```
# Select register
```

```
# Initialize register
```

```
# Apply controlled displacement
```

```
# Apply Hadamard-gate on qubit
```

```
# Measure qubit register
```

Concept

1. Encoding of information

- Implementing qubits or qudits as continuous-variable quantum states, such as GKP states or Cat-Codes. These states can have special properties, for instance, a resource-efficient error correction.
- Encoding real-valued numbers into quantum states.

2. Manipulation

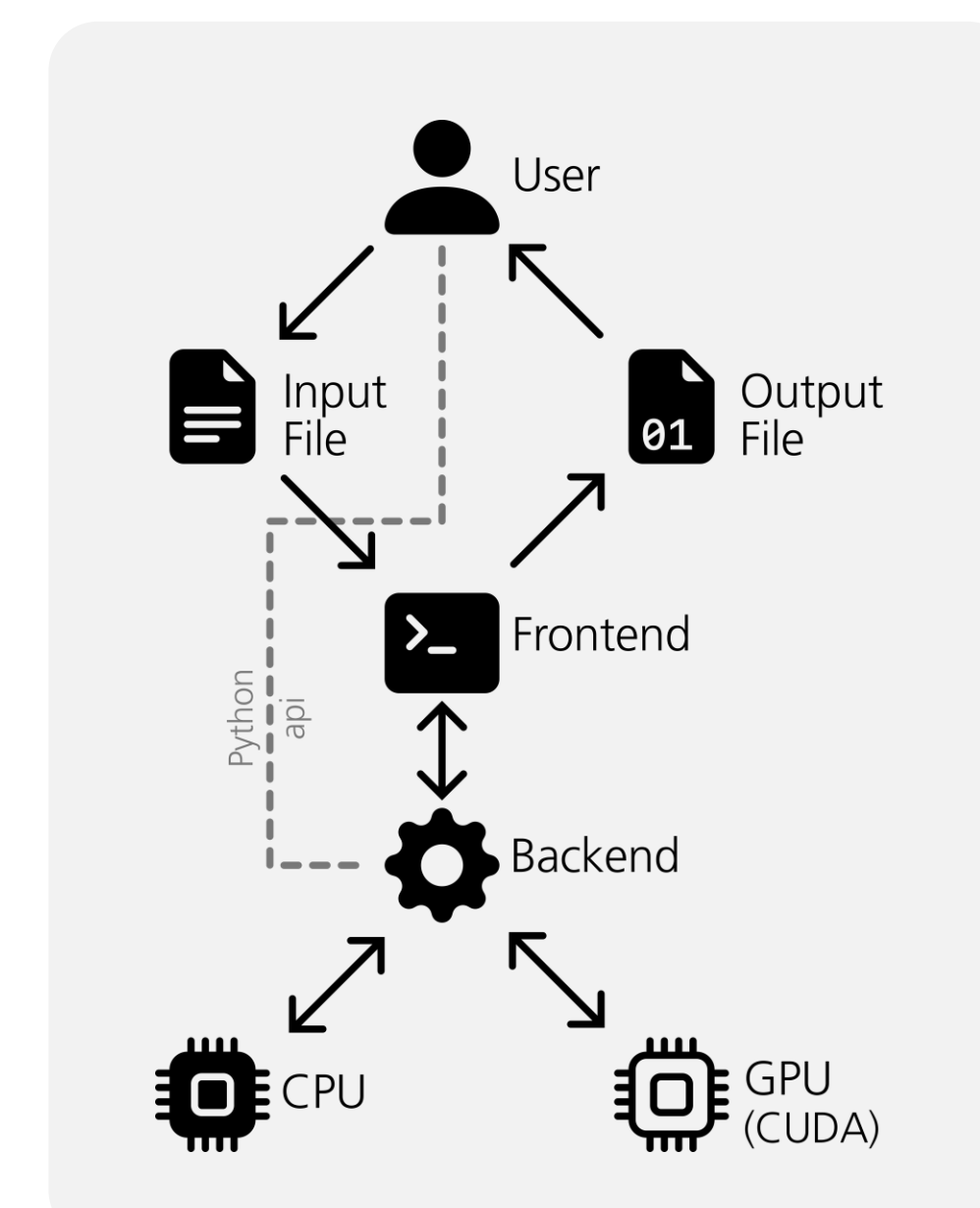
- Universal gate set consisting of phase space manipulations, e.g. displacements, rotation and squeezing, cubic operators as well as coupling operations.

3. Read-Out

- Measurement of the position or photon number of quantum states.

Software functionality

- Initialization, manipulation, coupling and read-out of arbitrary states.
- Calculations are executed in Fock or position basis.
- Visualization of quantum state in terms of Wigner functions.
- AQuRA can be imported as a Python package.
- Possibility to distribute calculations on the GPU.



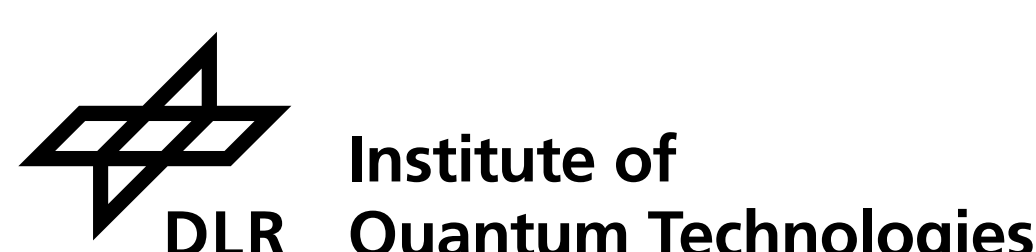
Potential applications

- **Optimization** of objective functions with real-valued input variables or in a combination with discrete decision variables.
- **Simulation** of continuous-variable quantum systems as well as composite systems of discrete and continuous variables.
- **Quantum information network** scenarios with continuous variables.

Cooperation with anabrid GmbH

- Computing paradigms and computing topologies.
- Modeling of quantum systems and real-world problems with the help of suitable analytical, numerical and analog methods.
- Simulation of use-cases and comparison of the (theoretical) performance of the different computing architectures.

A project of



Collaborator



Contact

Matthias Zimmermann
Matthias.zimmermann@dlr.de

More information about the project on our website



With funding from the:

