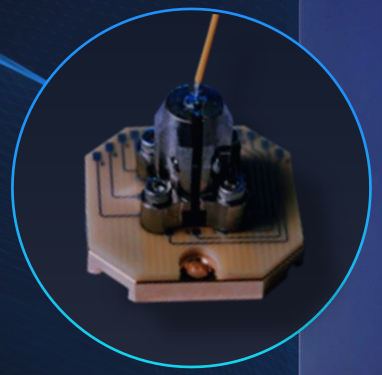
QUANDELA

Modular Optical Quantum Computing



Teratec 15th of June 2022

45 people dedicated to Quantum Computing

QUANDELA

More than 30 PhDs and engineers in algorithms, semiconductors, optical technologies and computer science

Offices, laboratories, clean-room facility in

(FR) PARIS, MASSY, PALAISEAU
(DE) MUNICH

Partnerships

With EU industrial actors to access and develop state-of-the-art products

Partnership with

TOPTICA PHOTONICS
attocube WITTENSTEIN Group
QUBIG
Swabian instruments
Pixel Photonics
leti CEA tech
LIGENTEC
Quantum Opus
QUIX QUANTUM

with several research institutions at the forefront of material and quantum science

Partnership with

IFW Leibniz-Institut für Festkörper- und Werkstoffforschung Dresden
The University Of Queensland AUSTRALIA
SAPIENZA Università di Roma
C2N
Inria
cnrs

Full-stack development

MOSAIQ

- Software dev.
- Theory & Algorithms
- Opto-electronics & modules
- Cryogenics & Integration
- Semicond. & sources

Development chain

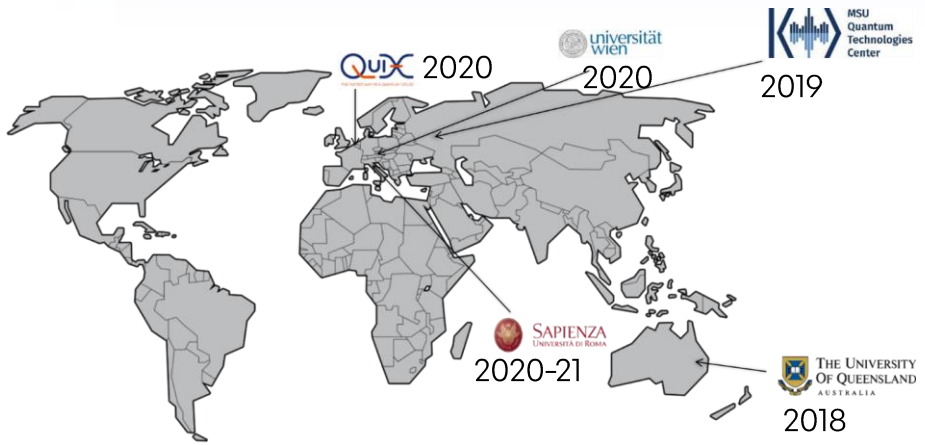
- User-access
- Software + algorithms stack
- Interconnection
- Fabrication & Production

Commercialization activity driven by the needs of the scientific community

Since 2018 we commercialize reliable
- single photon source devices
- opto-electronics modules

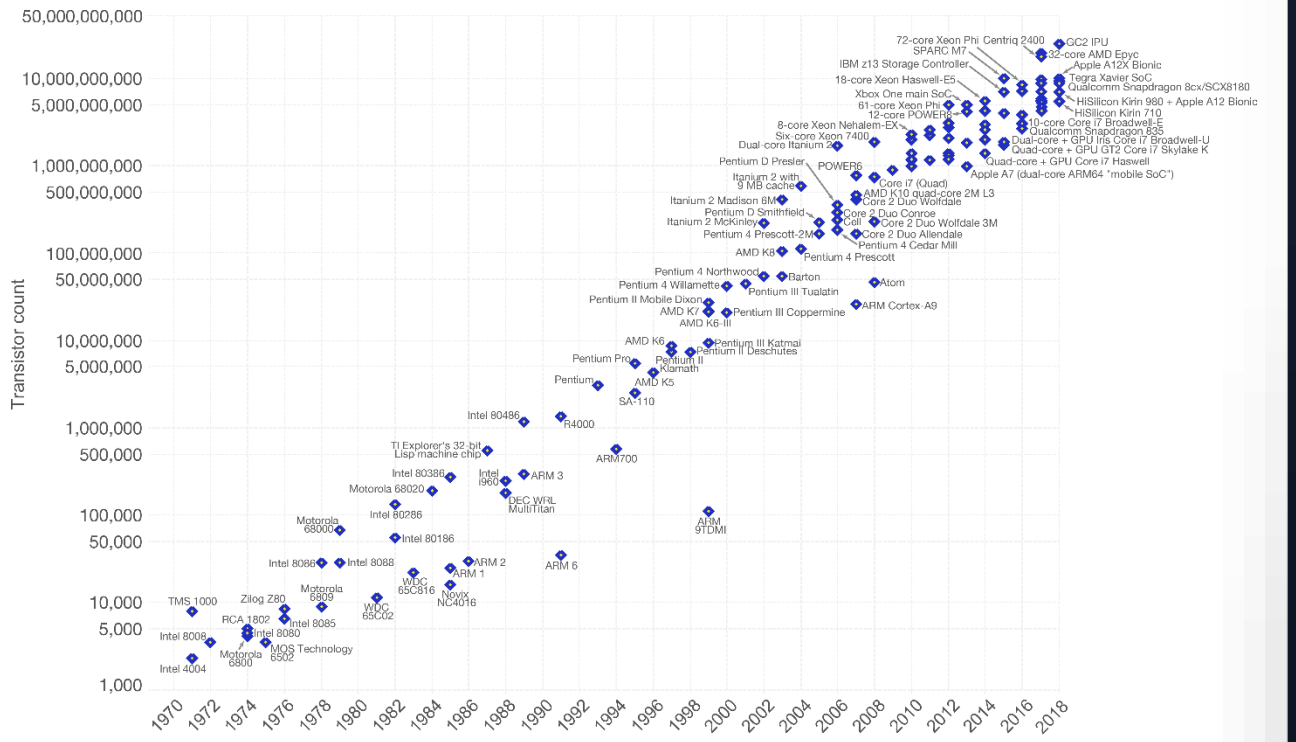
to enable the emerging of q-technologies

Clients:



Quantum computing, the next game changer

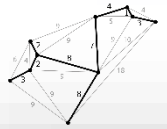
Classical computers based on electronics are reaching their limits due to the presence of a minimum processor size.



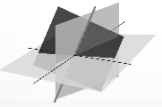
The next space race is the implementation of real-world applications on scalable Quantum Computers

Quantum Computers have the potential to find faster, more cost-effective and better-tailored solutions than classical machines.

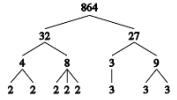
Some examples



Combinatorial Optimization



Linear Algebra

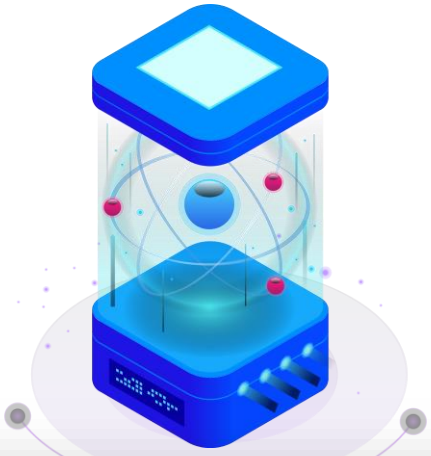


Factorization

Differential Equations

But scalability and decoherence makes it challenging

Number of qubits on chip is limited



Quantum Processors must be connected to each others

Scalability limited and high energy consumption due to dilution fridge

Quantum bits are subject to decoherence

SCALABILITY : is there an architecture for creating 100, 1000 qubits, where next hundreds are also possible.

→ The increase in “size” does not imply an exponential increase in errors and resources

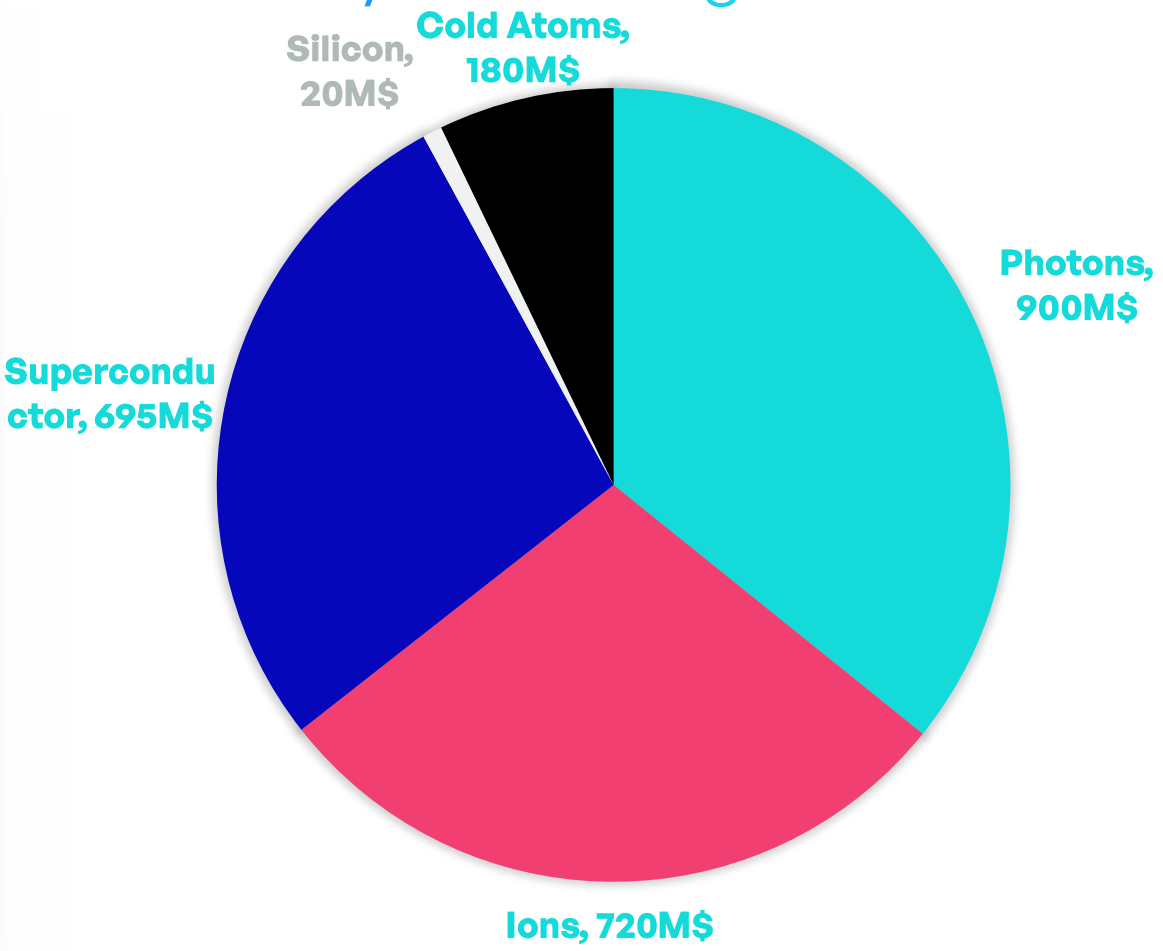
Light-based Vs matter-based qubits for digital QC

	Photons light based	Ions matter based	Superconducting matter based	Cold atoms matter based
Technology Readiness Level	5-6	5-6	5-6	5-6
Cryostats	Information processing at room temperature (source and detectors 4K)	N.A.	15mK	N.A.
Universality	✓	✓	✓	✗
Coherence time	11.5 billion years (but they can be lost)	~1 mn	~100 µs	~100µs
Platforms interconnectivity	✓	✗	✗	✗
Proven Quantum advantage	✓	✗	✓	✗

Photonic Quantum Computing Ecosystem...

PHOTONIC QUBITS		Specificities
PsiQ	Raised 665M$ 215M$ in 2020 + 450M$ in 2021	Monolithic approach based on thousands of sources detectors and switches
XANADU	Raised 100M$ in May 2021	Quantum advantage via Gaussian boson sampling
	Raised 15M$ in Oct 2021	High performing sources and opto-electronic modules
ORCA Computing	Leads a 12M$ project to build a quantum-safe data center funded by the UK government	Develops Quantum memories
QANT	Leads PhoQuant, a 50M€ project funded by the German Ministry of education and research (BMBF) to build a photonic quantum computer	
	Prof. J. W Pan & Prof. C. Lu's group	Showed quantum advantage

Leading the amount of investment by technologies



This is a result of a Quandela study. It accounts all the investment made in QC start-ups during the last five years.

QUANDELA

Optical Quantum Computing

Different Methods for Quantum Computing based on Light

Discrete Variables

Particles

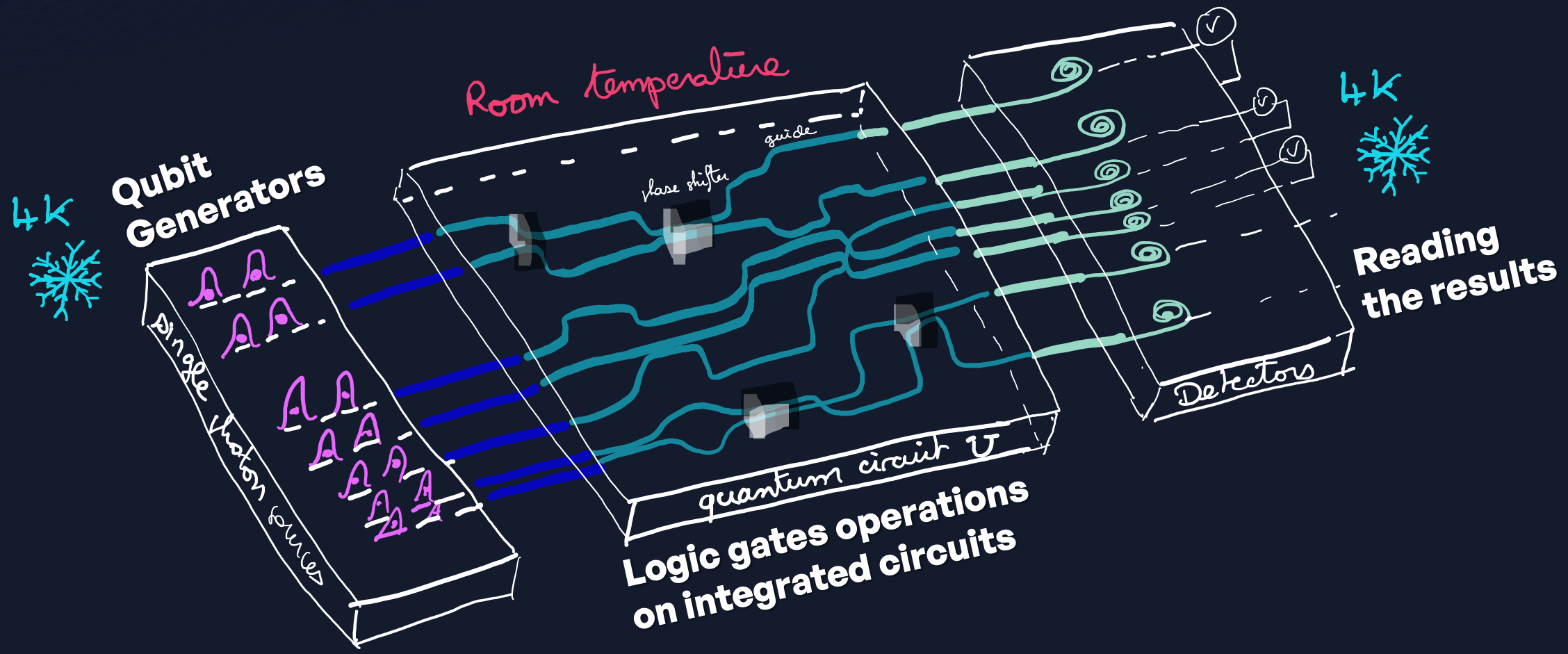
- Information is encoded on single photons
- Since 2001, universal scheme KLM for Linear Optical Quantum Computing (LOQC)
- Roadmap for scaling by using Measurement Based Quantum Computing (MBQC)
- Laboratory demonstration of calculations up to 20 qubits (2019)

Continuous Variables

Wave

- Information is encoded on field characteristics (amplitude, phase,...) - QUMODES
- Very well suited for boson sampling calculations (GBS)
- Protocols are very sensitive to optical losses
- Fault-tolerant protocols still under development and theoretical validation

Component of the Optical Quantum Computer



SWOT of Linear Optical Quantum Computing

Positive

Negative

Internal

External

Strengths

- **« Infinite » coherence time**
- Information processing at **room temperature**
- **Flying qubits**: modularity and interconnection

Opportunities

- **Photonic industry is developed** (lasers, detectors, photonic on chip)
- Photonic quantum computer is compatible with a **quantum network**

Weaknesses

- 2-qubit gates are **probabilistic**
- Requires **ultra-fast electronics** and **commuters**

Threats

- **Lack of financial support in EU** vs China and US
- Requires **compatibility of assemblers and compilers** with other QC platforms

QUANDELA

Quandela Quantum Computing vision

Optical Quantum computing Quandela's vision

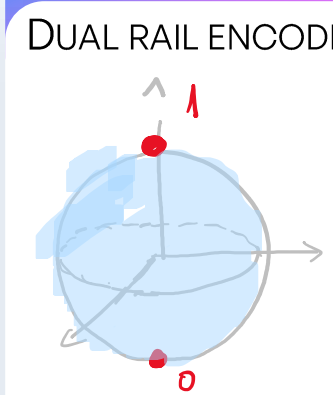
MAIN INGREDIENTS

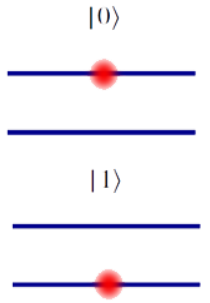
Type	Discrete variables: single photons
Encoding	Dual rail: integrated photonic Time: fibre loops & delay
Scheme	KLM: linear optical circuits – NISQ (up to tens of qubits) MBQC inspired: operation on cluster states, fusion gates – towards fault-tolerant
Approach	Efficient modules interconnected by ultra-low loss fibre links

PHOTONIC QUBITS

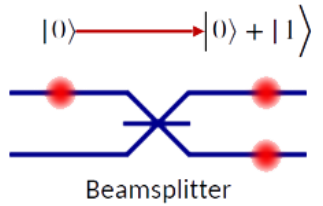
Quandela's Competitive Advantages

DUAL RAIL ENCODING

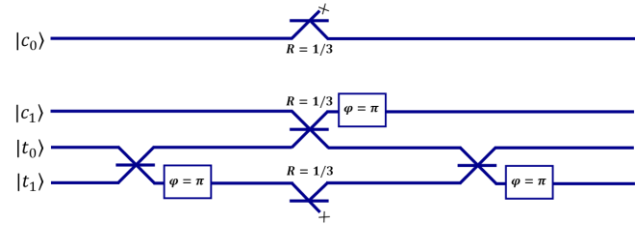


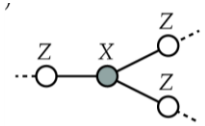


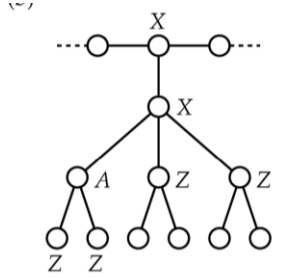
ONE-QUBIT GATE



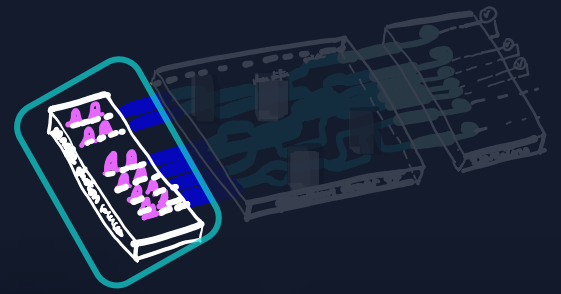
TWO-QUBIT GATE (CNOT)





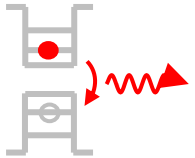


Single-photon sources : two approaches

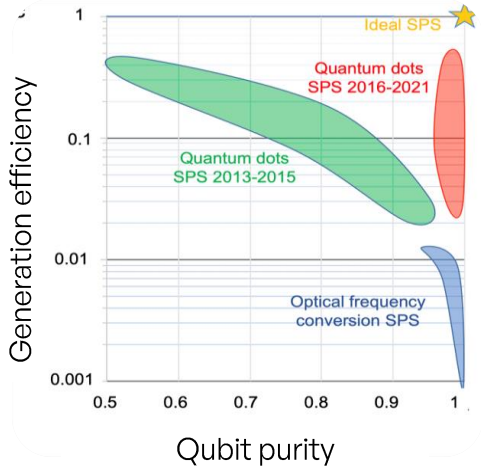


QUBIT GENERATOR

Deterministic (atomic) process



No limit for emission **efficiency** and **qubit purity**



KLM / NISQ

single-photon emission

source footprint **~ few µm²** in 1 device (2 cm2)

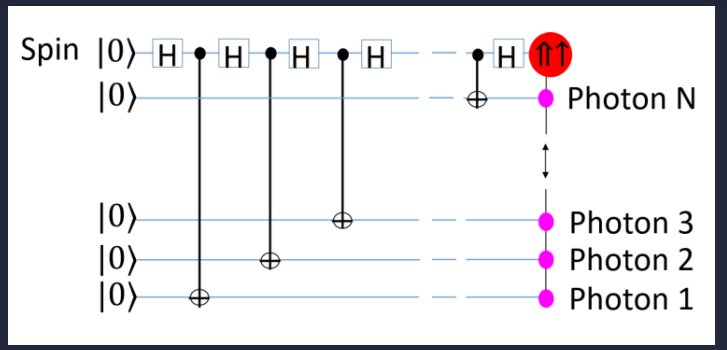
Current emission efficiency up **to 50 %**

MBQC / fault tolerant

linear cluster state emission

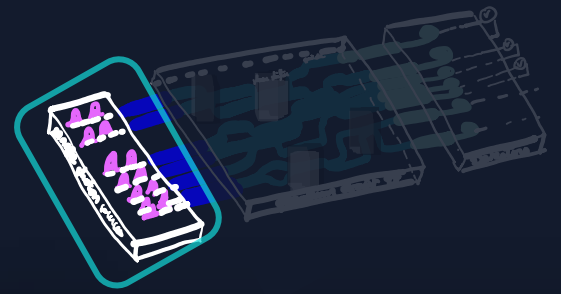
source footprint **~ few µm²** in 1 device (2 cm2) **+ SPIN & mT**

Laboratory demonstration of **3-photons cluster state**



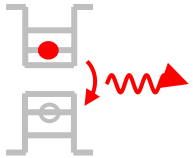
Linder, Rudolph, PRL, 103, 113602 (2009)

Single-photon sources : two approaches

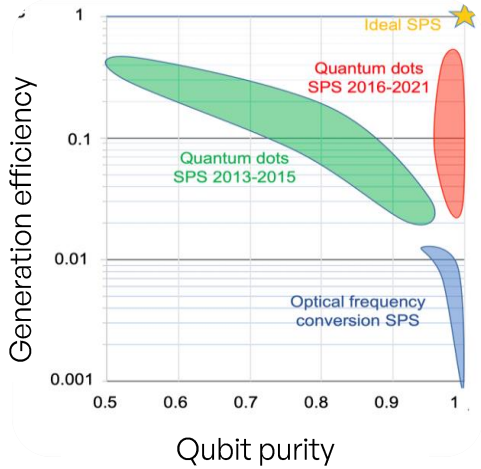


QUBIT GENERATOR

Deterministic (atomic) process



No limit for emission **efficiency** and **qubit purity**



KLM / NISQ

single-photon emission

source footprint **~ few µm²** in 1 device (2 cm2)

Current emission efficiency up **to 50 %**

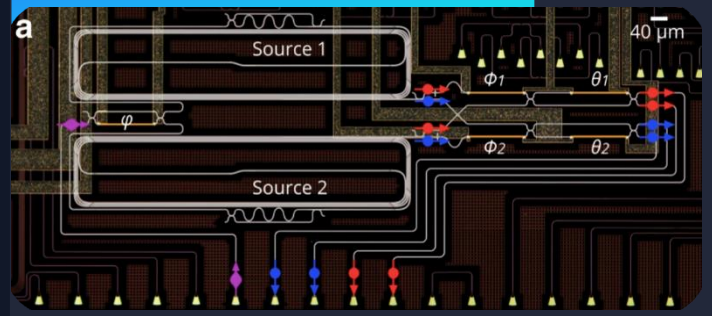
MBQC / fault tolerant

linear cluster state emission

source footprint **~ few µm²** in 1 device (2 cm2) **+ SPIN & mT**

Laboratory demonstration of **3-photons cluster state**

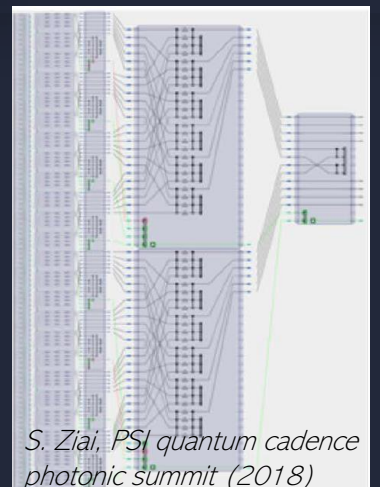
PROBABILISTIC SOURCE



Probabilistic processes (4wave-mix or SPDC)

Emission efficiency **bounded to few %, with high qubit purity.**

Require Multiplexing
Cascading of millions of components

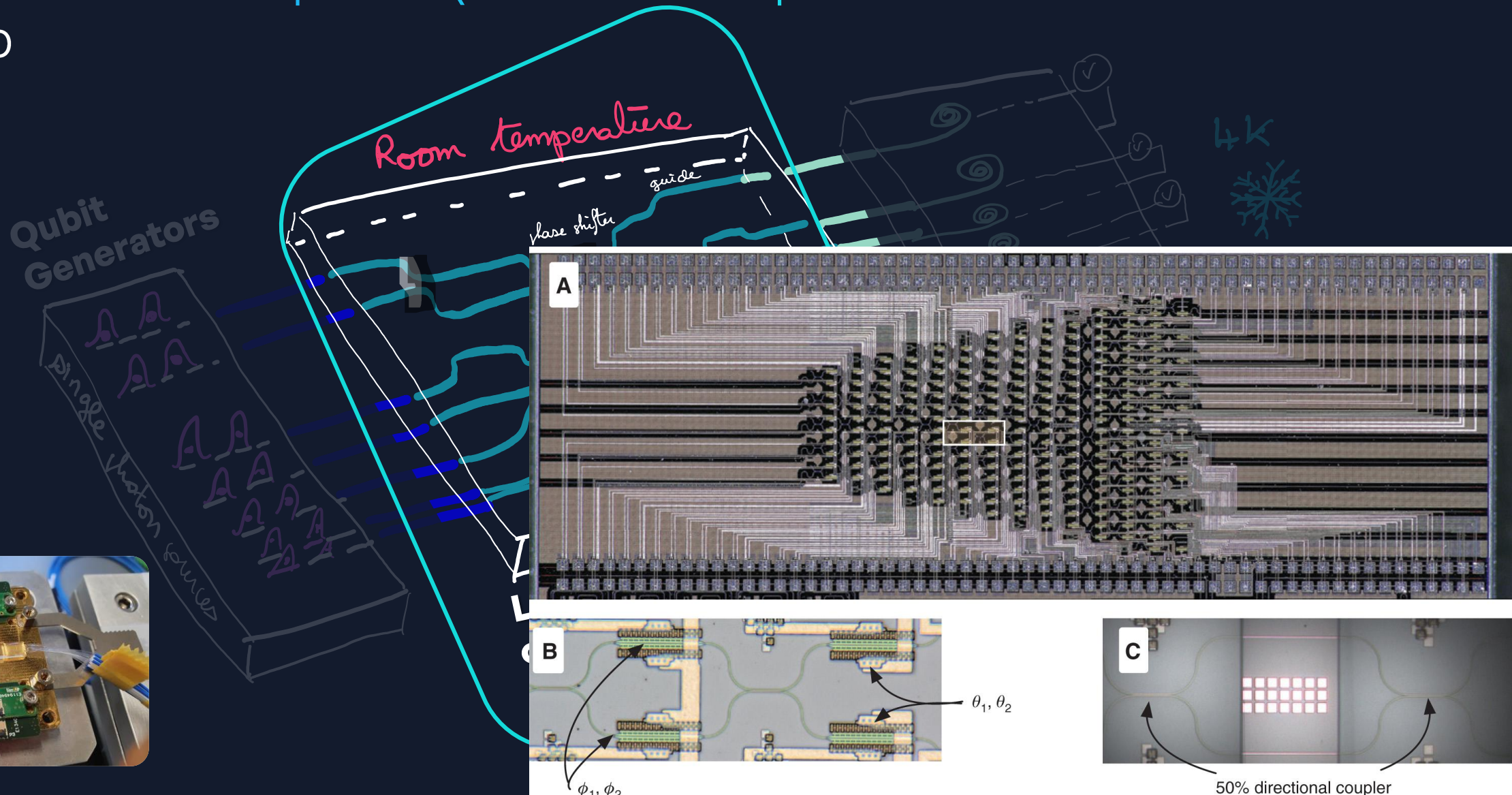


S. Ziai, PSI quantum cadence photonic summit (2018)

1 photon: Footprint ~dm²

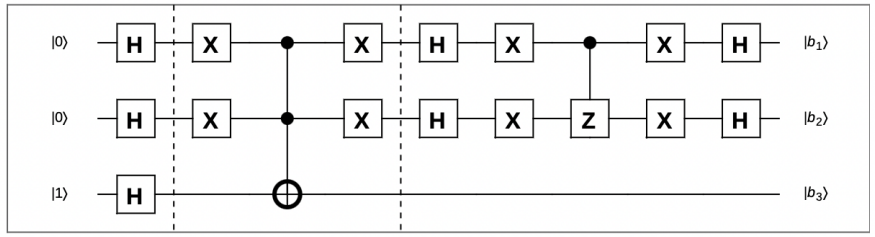
3-photons cluster: Footprint ~ full 300mm wafer

Component of the Optical Quantum Computer:
The chip



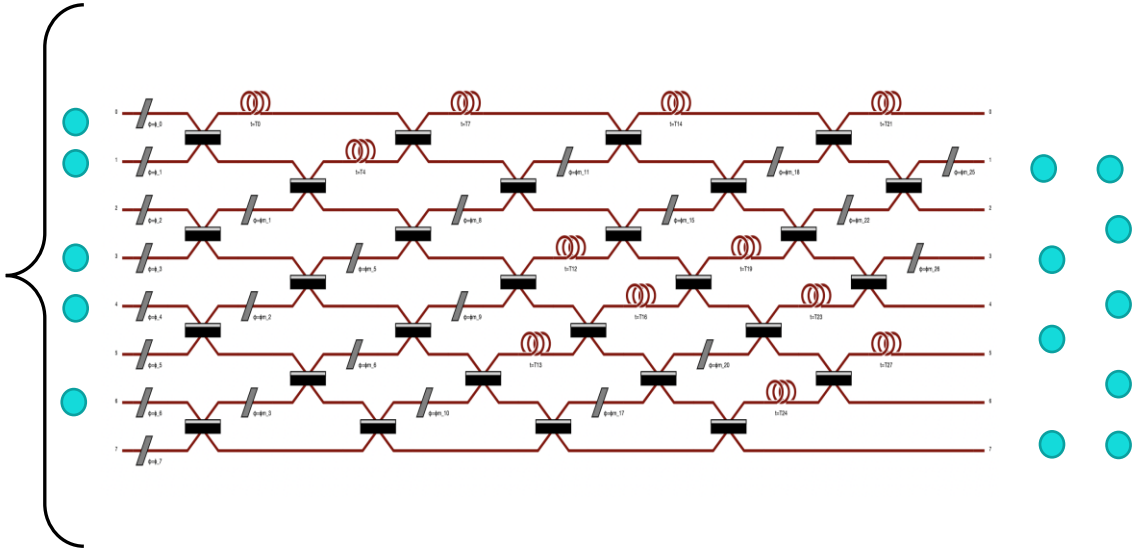
Programming a photonic chip

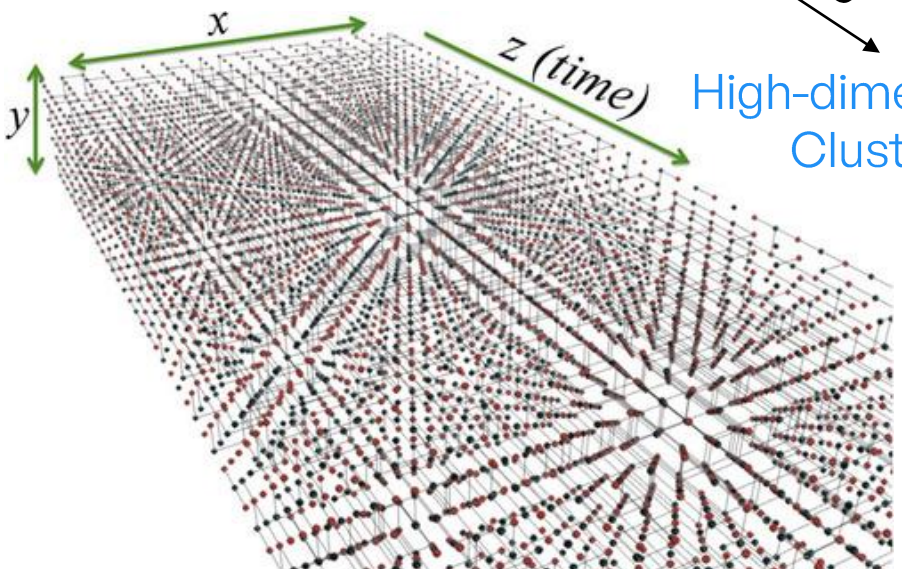
FROM GATE-BASED TO LO CIRCUITS



Quantum Circuit → NISQ → Photonic Circuit

Quantum Circuit → MBQC → High-dimensional Cluster state





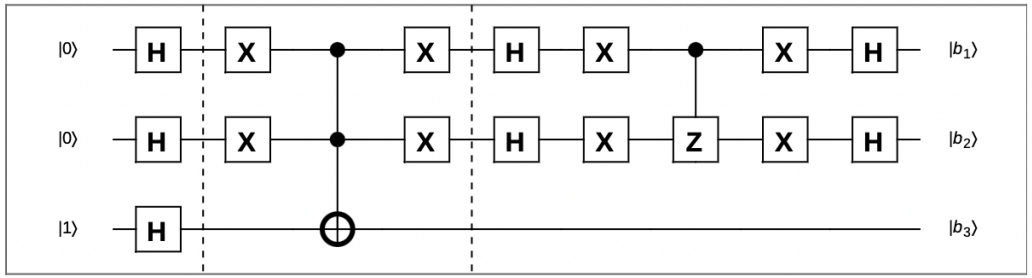
NISQ

Compile quantum circuit directly to photonic circuit

Large-scale fault-tolerant via MBQC

- Evaluate quantum circuit by measurements on cluster state via 1-qubit measurements

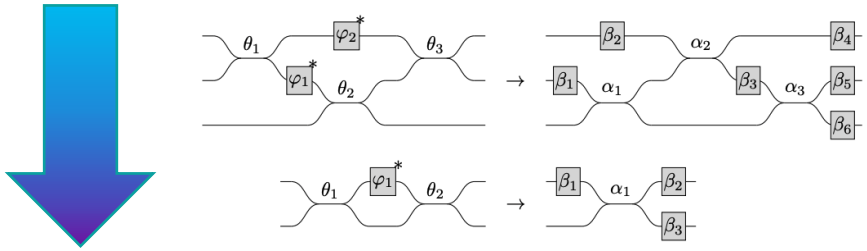
Every gate-based circuit can be translated into a photonic circuit to leverage the power of PQC

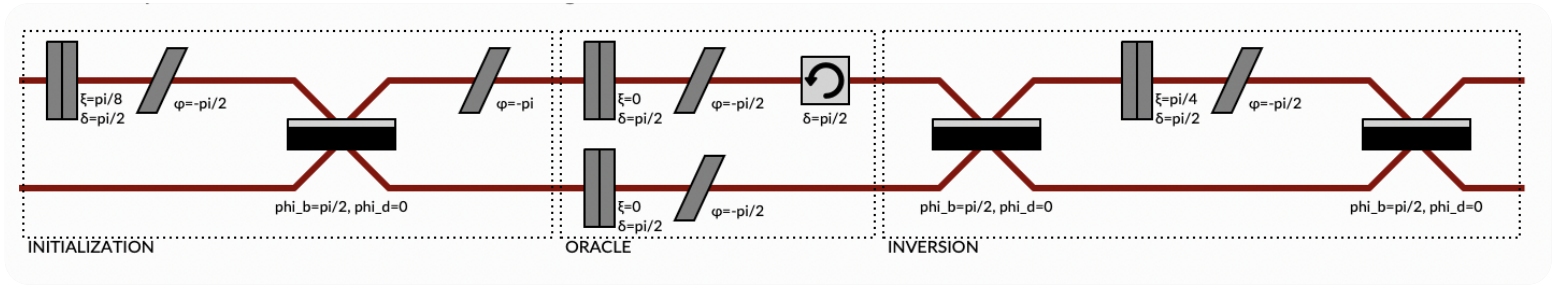


LOv-Calculus

We developed the first **graphical language for photonic processors**

Clément et al., Lov-Calculus: a graphical language for linear optical quantum circuits (2022)





Hilbert space (H)

|1,1,1>, |1,0,0>, |1,0,1>, |0,0,1>, |0,1,0>, ...

Fock space (space of photons) **has higher dimension** than Hilbert space (space of qubits)
-100% equivalency maintained-

Fock space ($F = H^n$)

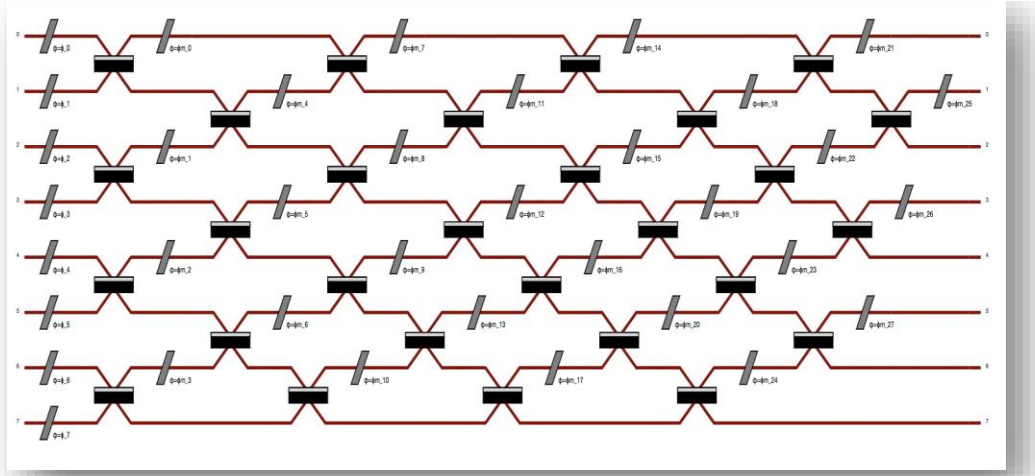
|1,1,1>, |1,2,0>, |1,0,2>, |2,0,1>, |2,1,0>, |0,1,2>, |1,0,2>, |3,0,0>, |0,3,0>, |0,0,3> , ...

Chip processor module

INTEGRATED PHOTONIC CIRCUITS

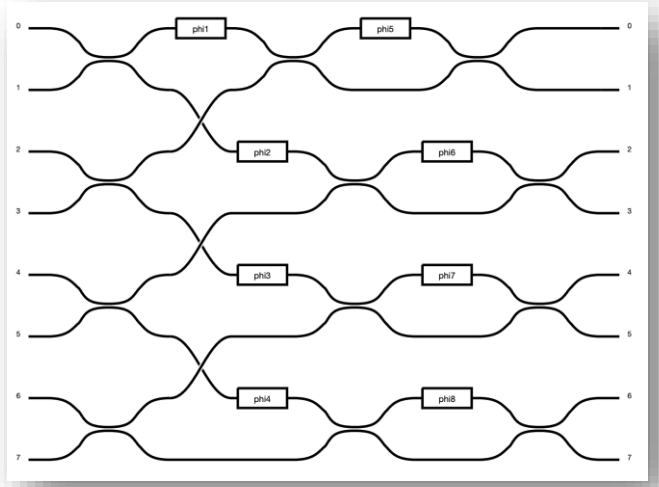
Generic Interferometer where each single phase is controlled and can be individually set

```
import perceval as pcvl
import perceval.lib.phys as phys

N=8
c = pcvl.Circuit.generic_interferometer(
    N,
    lambda idx: phys.BS() // (0, phys.PS(phi=pcvl.P("ϕm_%d" % idx))),
    depth=N,
    phase_shifter_fun_gen=lambda idx: phys.PS(phi=pcvl.P("ϕ_%d" % idx))
)

pcvl.pdisplay(c)
```



Specialized circuit implementing a specific function

```
ghz_chip = pcvl.Circuit(m=8, name="GHZChip")

for i in range(4):
    ghz_chip.add((2 * i, 2 * i + 1), symb.BS())
for i in range(3):
    ghz_chip.add((2 * i + 1, 2 * i + 2), symb.PERM([1, 0]))
for i in range(4):
    ghz_chip.add(2 * i, symb.PS(phases[i]))
for i in range(4):
    ghz_chip.add((2 * i, 2 * i + 1), symb.BS())
for i in range(4):
    ghz_chip.add(2 * i, symb.PS(phases[i + 4]))
for i in range(4):
    ghz_chip.add((2 * i, 2 * i + 1), symb.BS())
```

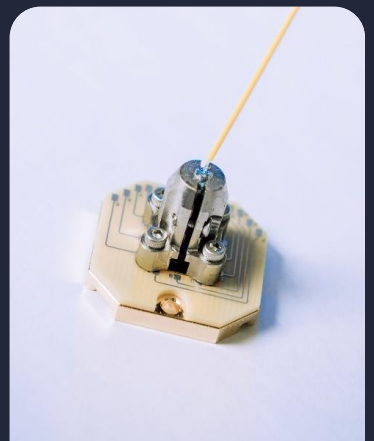


GHZ-Generator

QUANDELA

Quandela's QC solutions and use-case examples

We build
the first full-stack optical QC



MosaiQ

End-users — Cloud access or on-premise

Front end — Python Libs, REST API, Visual/Graphical interface, Integration with existing platforms

Compiler — Logical Qubits <> Photon encoding

Assembler

Calibration, Machine Language

Hardware Modules

Electronics, FPGA, Voltage Sequence

Semiconductors

(sources, photonics integrated chips, detectors,...)

Perceval Optical QC Simulator

(< 20 qubits)

1. Main computer
2. Lasers & Electronics
3. Photonic Integrated on Chip (PIC)
4. Qbit-controller module
5. Photonic Qubit Demultiplexer (DMX)
6. Cryogenically cooled qubit generator

Available products & services

DOUBLE the number of qubits each year.

Modularity and scalability for reaching universal quantum computing and fault tolerant prototypes in 2024

Current market and user traction

MBDA · ONERA · EDF · OVHcloud

MosaiQ

2 qubits	Certified QRNG Available on premise Soon on cloud	☁
4 qubits	Cluster (GHZ) state generator and analyzer Laboratory prototype	
6 qubits	Reconfigurable QPU From Q3 2022 Running a set of protocols available by users, on the cloud	☁
Perceval	LOC simulator Available open-source (15 qubits) PRO version via cloud (20 qubits)	☁

Quantum emitter devices and modules

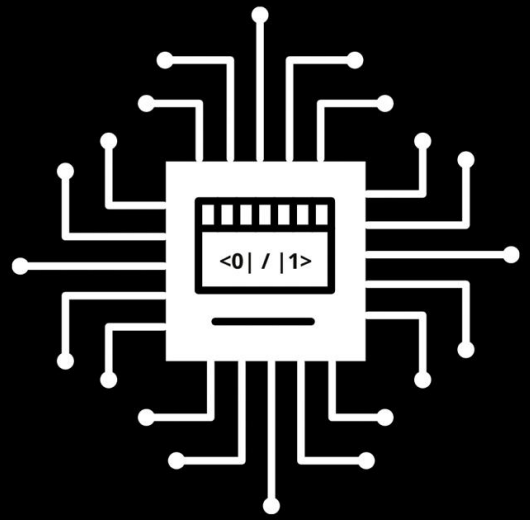
Prometheus

Available
Turn-key single-photon source system

6-QDMX

Available
compact 6 single-photon router. 12 photon version under development

Perceval: A Software Platform for Discrete Variable Photonic Quantum Computing



Simulation

Design

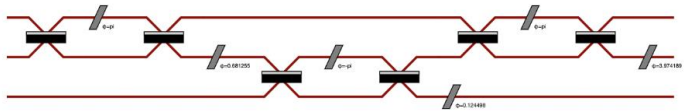
Experimentation

https://perceval.quandela.net/

OVHcloud

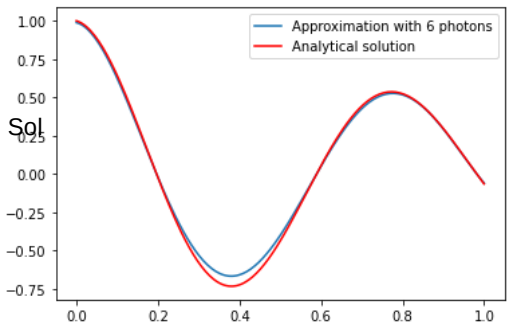
Perceval, Open-source programming framework for Quantum Photonics

SIMULATION

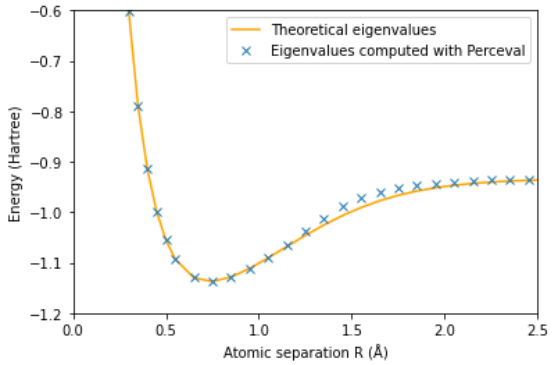


Shor's and Grover's Algorithms on a photonic processor

Up to 15 (20) photons on a laptop (supercomputer)



Solving Differential Equations



Simulation of Variational Quantum Eigensolvers

- **Simple to use**, both for physicists and for computer scientists
- **Does not constrain** the user **to any framework-specific** conventions that are theoretically equivalent
- Provides **state-of-the-art optimized algorithms**
- Provides access to symbolic calculations for finding analytical solutions
- Provides companion tools, like a unitary matrix toolkit, LateX or HTML rendering of algorithms
- Incorporates realistic, parametrizable error and noise modelling

Simulation (Amplitude Estimation)

Option Pricing
Portfolio Risk

Optimization (Variational Algorithms, Quantum Semi-Definite Programming)

Portfolio Optimization, Diversifications, Issuance

Quantum Machine Learning (HHL, Quantum Super-Vector Machine)

Financial forecasting
Credit Scoring
Fraud Detection and Money-laundering

Heurtel et al., Perceval: A Software Platform for Discrete Variable Photonic Quantum Computing, arXiv (2022)

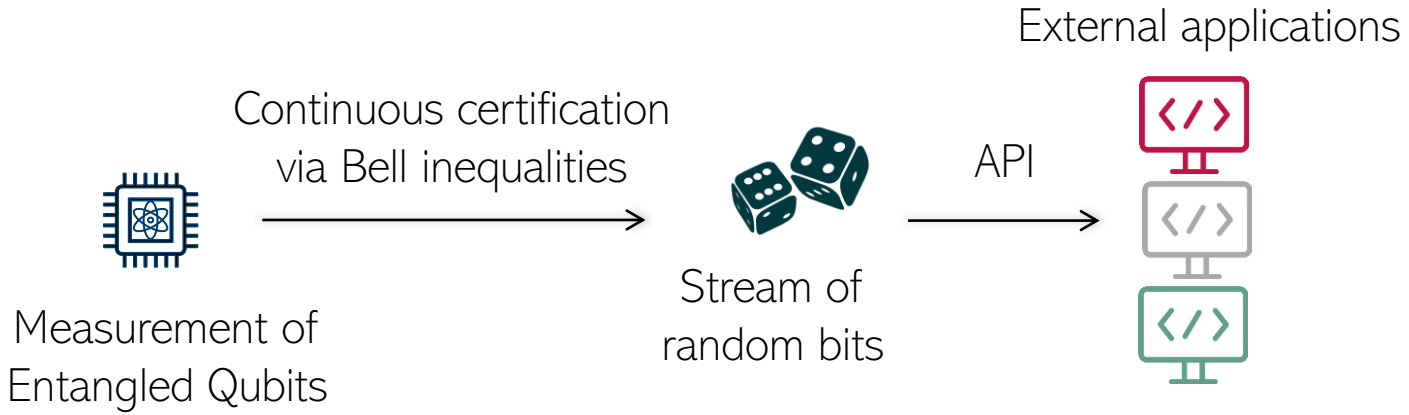
Collaborative & Open Source

Entropy – First useful application of a 2-qubit QPU

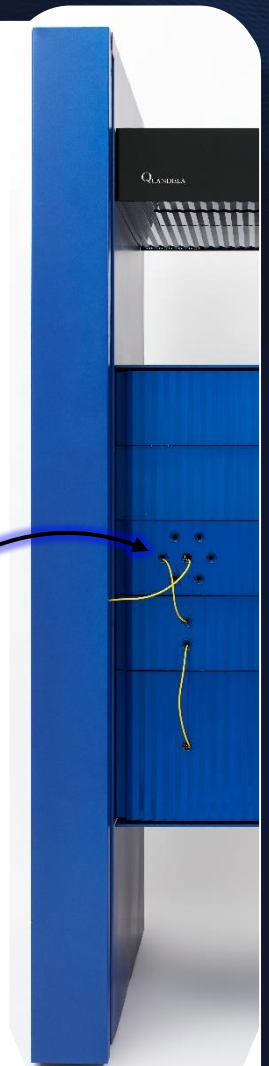
For cybersecurity and Monte-carlo simulations

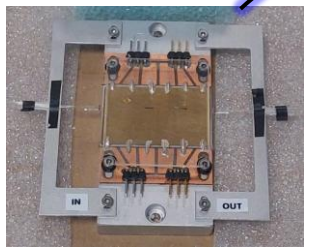
MOSAIQ

Quandela provides data-center compatible 2-qubit processor unit with a proprietary protocol to extract quantum-certified randomness



- Upgradable
- Remote (cloud)





5 cm

BBC NEWS

NSA 'altered random-number generator'

11 September 2013

US intelligence agency the NSA subverted a standards process to be able to break encryption more easily, according to leaked documents.

Applications in Cyber, Montecarlo simulations, etc…

QUANDELA

We build the European Photonic Quantum Computer

Technical specification – bonus slide

MOSAIQ

Specifications	MosaiQ
Technology	Optical quantum computer based on bright single-photon sources
Starting time	6 hours (cooling down to 4K) – Max. 1 day
Compressor	Water-cooled (16A) Optional: Air-cooled 1 phase, 220V, 50/60Hz
User Interface	Wire access or online access
Technical Interface	Fully automated control of the different modules using the central computer
Power Consumption & Electrical Connections	Power: max. 10kW Voltage: 220V AC (Air-cooled)
Physical Dimensions	Whole structure: 180x76x84cm (x2) Weight: max. 250kg (x2)

Temperature Variations	- +/- 3°C: no effect - +/- 7°C: monitoring and auto stabilization mechanism - +/- 10°C: short intervention (a few hours)
Hygrometric Constraint	No
Electromagnetic Constraint	No
Vibration Constraint	No
Power Outages	Cryostat temperature rises and QC stops working 1 day to cool down the system Easy procedure
Laser Risk	No (closed modules & fibered equipment)
Fluid Related Risk	No (closed circuit & small quantities)

NO NEED of specialized technicians for maintenance

Discretee variable QC

THE LOC APPROACH (2000)

Efficient Linear Optics Quantum Computation

E. Knill,[1,*] R. Laflamme,[1,†] and G. Milburn[2,‡]

[1]*Los Alamos National Laboratory, MS B265, Los Alamos, New Mexico 87545*
[2]*Center for Quantum Computer Technology, Department of Physics, University of Queensland, St. Lucia, Australia*

- Universal for Fault tolerant computation
- Based on the manipulation of single photons and their 2-photon interference on BS
- Use of linear optics to create logic gates and entanglement
- Quick to implement
- Hard to scale: due to photon losses and probabilistic two-qubits gates

THE MBQC APPROACH (2009)

Measurement-based quantum computation

H. J. Briegel[1,2], D. E. Browne[3], W. Dür[1,2], R. Raussendorf[4] and M. Van den Nest[2,5]

- Universal for Fault tolerant computation
- Based on the manipulation of cluster state of entangled photons
- Generation of the cluster state “off-line”
- Computation via sequences of “1-qubit” measurement + feedforward
- The path to scale: 1 qubit gates are “trivial”, existence of error correction protocols
- Hard to generate cluster states

Perceval a Software Platform for Discrete Variable Photonic Quantum Computing : pip install, create linear optical circuits and display them (1/5)

0. Installing:

```
pip install perceval-quandela
```

1. Starting with import

```
import perceval as pcvl
import numpy as np
import perceval.lib.phys as plib
import perceval.lib.symb as symb
```

2. Testing symbolic components:

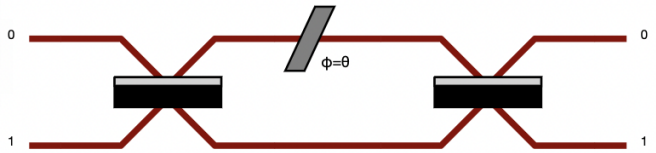
```
bs=plib.BS(theta=pcvl.P("θ"))   #beamsplitter
ps=plib.PS(pcvl.P("φ"))         #phase shifter
```

```
pcvl.pdisplay(bs)               #display the component
pcvl.pdisplay(ps)
pcvl.pdisplay(bs.U)             #display the matrix
pcvl.pdisplay(ps.U)
pcvl.pdisplay(bs.definition())  #display the defintion/convention
```

3. Testing numeric components:

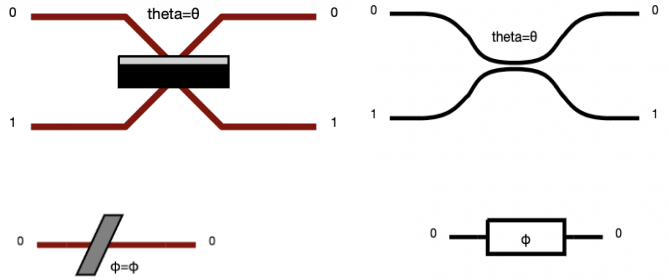
```
#Mach-Zehnder Inteferometer
mzi= pcvl.Circuit(2) // plib.BS(R=1/2) // plib.PS(phi=pcvl.P("θ")) // plib.BS(R=1/2)

pcvl.pdisplay(mzi)
mzi.U
```

3.1 Outcome:



$$\begin{bmatrix} 0.5e^{i\theta}+0.5 & 0.5e^{i\theta}-0.5 \\ 0.5e^{i\theta}-0.5 & 0.5e^{i\theta}+0.5 \end{bmatrix}$$

2.1 Outcome:



$$\begin{bmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{bmatrix}$$

$$\begin{bmatrix} e^{i\varphi} \end{bmatrix}$$

$$\begin{bmatrix} e^{i\phi_a}\cos(\theta) & ie^{i\phi_b}\sin(\theta) \\ ie^{i(\phi_a-\phi_b+\phi_d)}\sin(\theta) & e^{i\phi_d}\cos(\theta) \end{bmatrix}$$

[Heurtel et al., Perceval: Software Platform for Discrete Variable Photonic Quantum Computing. France, (2022)]

Perceval a Software Platform for Discrete Variable Photonic Quantum Computing : choose backend, define states and simulate outcome probabilities (2/5)

List of backends

```
pcvl.BackendFactory().list_backend()
```

```
['Naive', 'CliffordClifford2017', 'SF', 'SLOS', 'Stepper']
```

1. Implementing a circuit in 'Naive':

```
backend = pcvl.BackendFactory().get_backend("Naive")
circuit=plib.BS(R=1/2)
simulator=backend(circuit)

pcvl.pdisplay(circuit)
```

1.1:

2. Defining states and evolving:

```
input10=pcvl.BasicState("|1,0>")
input01=pcvl.BasicState("|0,1>")
input11=pcvl.BasicState("|1,1>")

# Strong simulation

print("output de input10: ",simulator.evolve(input10))
print("output de input01: ",simulator.evolve(input01))
print("output de input11: ",simulator.evolve(input11))
```

2.1:

```
output de input10:  sqrt(2)/2*|1,0>+sqrt(2)/2*|0,1>
output de input01:  sqrt(2)/2*|1,0>-sqrt(2)/2*|0,1>
output de input11:  sqrt(2)/2*|2,0>-sqrt(2)/2*|0,2>
```

3. Computing probabilities and amplitudes:

```
simulator.prob(input10,pcvl.BasicState("|0,1>"))
```

```
simulator.probampli(input10,pcvl.BasicState("|0,1>"))
```

3.1:

```
0.5000000000000001
(0.7071067811865476+4.329780281177467e-17j)
```

[Heurtel et al., Perceval: Software Platform for Discrete Variable Photonic Quantum Computing. France, (2022)]

Perceval a Software Platform for Discrete Variable Photonic Quantum Computing : sampling (3/5)

Boson Sampling

1. Defining circuit

```
n = 9    #number of photons at the input
m = 30   #number of modes
N = 10   #number of samplings
```

```
Unitary_10 = pcvl.Matrix.random_unitary(m)
```

Decomposition (optional, to verify)

```
mzi = (symb.BS() // (0, symb.PS(phi=pcvl.Parameter("φ_a")))
       // symb.BS() // (1, symb.PS(phi=pcvl.Parameter("φ_b"))))
pcvl.pdisplay(mzi)
```

```
Linear_Circuit_10 = pcvl.Circuit.decomposition(Unitary_10, mzi,
                                               phase_shifter_fn=symb.PS,
                                               shape="triangle")
```

2. Backend selection

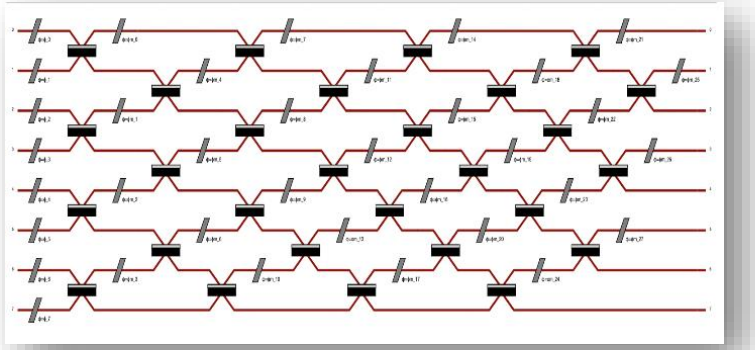
```
Sampling_Backend = pcvl.BackendFactory().
                        get_backend("CliffordClifford2017")
```

3. Choosing input

```
input_state = Generating_Input(n,m)
print("The input state:\n", input_state)
```

4. Outputting samples

```
print("The sampled outputs are:")
for _ in range(N):
    print(Sampling_Backend(Unitary_10).sample(input_state))
```



3.1:

```
The input state:
 |0,0,0,1,0,0,0,0,1,1,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1>
```

4.1:

```
The sampled outputs are:
|0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,3,0,0,0,2,0,0,0,0,0,0,0,3,0>
|0,0,1,0,1,0,1,0,0,0,0,1,1,0,1,0,1,0,0,0,0,0,1,0,0,0,1,0,0,0>
|0,1,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,2>
|0,0,1,1,0,2,0,0,0,0,3,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0>
|0,1,1,1,0,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0>
|0,1,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,1,0,1,2,0,0>
|0,0,1,0,0,0,2,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,1,1,0,0,1>
|0,0,0,0,0,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,1,0,2,0,1>
|0,0,0,2,0,0,1,0,0,0,1,0,0,1,0,0,0,0,1,1,0,1,0,0,0,0,0,0,0,1>
|0,0,2,0,0,0,0,0,0,0,0,1,1,0,0,1,0,0,0,0,1,0,0,1,0,0,1,0,1,0>
```

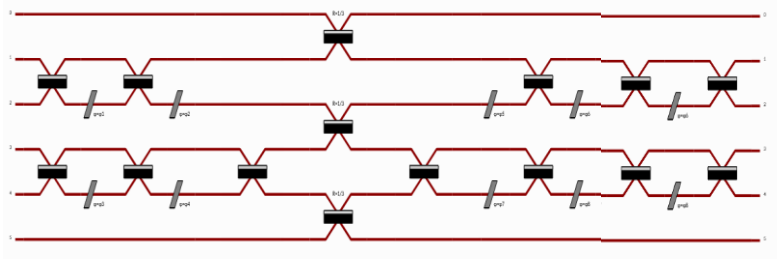
Perceval a Software Platform for Discrete Variable Photonic Quantum Computing : Variational Quantum Eigensolver, create circuit and define loss function (4/5)

Variational Quantum Eigen solver

1. Defining circuit

```
#List of the parameters φ1,φ2,...,φ8
List_Parameters=[]

# VQE is a 6 optical mode circuit
VQE=pcvl.Circuit(6)

VQE.add((1,2), phys.BS(R=1/2))
VQE.add((3,4), phys.BS(R=1/2))
List_Parameters.append(pcvl.Parameter("ϕ1"))
VQE.add((2,),phys.PS(phi=List_Parameters[-1]))
List_Parameters.append(pcvl.Parameter("ϕ3"))
VQE.add((4,),phys.PS(phi=List_Parameters[-1]))
VQE.add((1,2), phys.BS(R=1/2))
VQE.add((3,4), phys.BS(R=1/2))
List_Parameters.append(pcvl.Parameter("ϕ2"))
VQE.add((2,),phys.PS(phi=List_Parameters[-1]))
List_Parameters.append(pcvl.Parameter("ϕ4"))
VQE.add((4,),phys.PS(phi=List_Parameters[-1]))
```

...



2. Define loss function

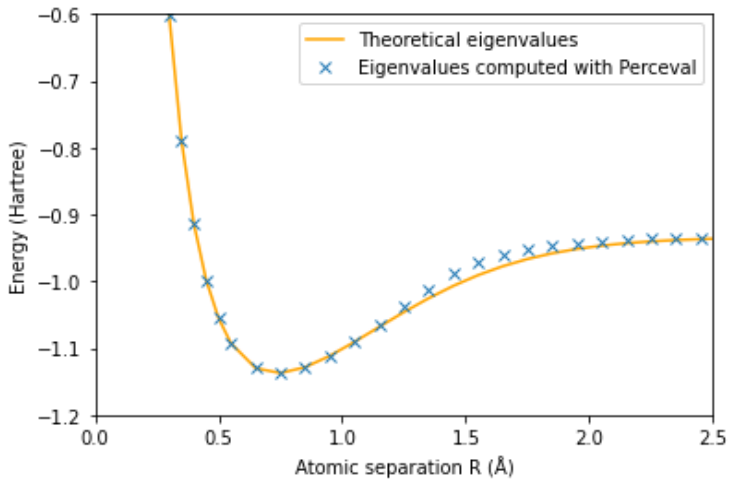
```
def minimize_loss(lp=None):
    # Updating the parameters on the chip
    for idx, p in enumerate(lp):
        List_Parameters[idx].set_value(p)

    #Simulation, Quantum processing part of the VQE
    s_VQE = simulator_backend(VQE.compute_unitary(use_symbolic = False))

    # Collecting the output state of the circuit
    psi = []
    for input_state in input_states:
        for output_state in output_states: #|00>,|01>,|10>,|11>
            psi.append(s_VQE.probampli(input_state,output_state))

        #Evaluating the mean value of the Hamiltonian.  # The Hamiltonians H is defined in the following block
    psi_prime=np.dot(H[R][1],psi)
    loss = np.real(sum(sum(np.conjugate(psi)*np.array(psi_prime[0])))/(sum([i*np.conjugate(i) for i in psi]))
    loss=np.real(loss)

    tq.set_description('%g / %g  loss function=%g' % (R, len(H), loss))
    return(loss)
```

Heurtel et al., Perceval: A Software Platform for Discrete Variable Photonic Quantum Computing, arXiv (2022)

Perceval a Software Platform for Discrete Variable Photonic Quantum Computing : Variational Quantum Eigensolver, minimize loss function (5/5)

3. Optimizing circuit parameters using a classical algorithm

```
tq = tqdm(desc='Minimizing...') #New progress bar
radius3=[]
E3=[]
init_param=[]

H=H3

for R in range(len(H)):            #We try to find the ground state eigenvalue for each radius R
    radius3.append(H[R][0])
    if (init_param==[]):           #
            init_param = [2*(np.pi)*random.random() for _ in List_Parameters]
    else:
        for i in range(len(init_param)):
            init_param[i]=VQE.get_parameters()[i]._value

    # Finding the ground state eigen value for each H(R)
    result=minimize(minimize_loss,init_param,method='Nelder-Mead')

    E3.append(result.get('fun'))
    tq.set_description('Finished' )
```



Finding the eigenstate of a two-atom system

For full code and other examples see :

Heurtel et al., Perceval: A Software Platform for Discrete Variable Photonic Quantum Computing, arXiv (2022)
https://perceval.quandela.net/docs/notebooks/Variational%20Quantum%20Eigensolver.html