Comparison of Bosonic-Qiskit, Strawberry Fields, and Perceval for Continuous-Variable Quantum Computing

Masoud Hakimi Heris

July 30, 2025

Abstract

This report provides a brief comparison of three quantum software frameworks: `bosonic-qiskit` (bq), `Strawberry Fields` (sf), and `Perceval` by Quandela, focusing on their capabilities and differences in implementing continuous-variable (CV) and discrete-variable (DV) quantum computations.

To gain practical insight into these frameworks, I have experimented with their programming models and backend support. For `Strawberry Fields`, developed by Xanadu, my exploratory work and encountered issues are documented in the GitHub repository: `SFrepo`. Similarly, for `Perceval` by Quandela, my implementation attempts and detailed notes are available at: `Quandelarepo`. These repositories contain detailed code samples, observations, and challenges encountered during the use of each platform's toolchain and backends. Also, because we could have a comparison between SF and BQ (as they are able to both handle CV), I have simulated an example of CV QAOA from SF CV-QAOA notebook on GitHub which was the first start for Xanadu evaluations as well, using BQ in this repository. I can conclude that so far the only available platform for hybrid is BQ, but SF (or even pennylane which seems to be able to handle hybrid cv-dv, however, it doesn't support controlled cv operations) only supports bosonic cv and perceval (quandela) only supports bosonic dv.

1 Bosonic-Qiskit (bq)

Bosonic-Qiskit extends the IBM Qiskit framework to support hybrid continuous-variable (CV) and discrete-variable (DV) quantum circuits. While it offers a range of bosonic gates and supports the construction of complex CV-DV circuits, it lacks certain native CV operations, particularly the *quadratic phase gate* (also known as the squeezing gate) and the direct *position* and *momentum displacement* gates that are available in Strawberry Fields.

Instead, bosonic-qiskit provides hybrid gates that combine CV and DV operations, allowing flexibility in algorithm design. However, this means that some fundamental CV gates must be synthesized or approximated using these hybrid operations.

2 Strawberry Fields (sf)

Strawberry Fields, developed by Xanadu, is a comprehensive quantum software platform dedicated primarily to continuous-variable quantum computing. It natively supports a rich set of CV gates, including:

- Position displacement $\hat{D}(x)$,
- Momentum displacement $\hat{D}(p)$,

- Squeezing (quadratic phase gate),
- Rotation and beam splitter operations.

This native support for fundamental CV gates makes Strawberry Fields particularly well-suited for simulating photonic CV systems and implementing algorithms that require fine-grained CV gate control.

3 Perceval (Quandela)

Perceval, developed by Quandela, is primarily focused on discrete-variable photonic quantum computing but with bosonic encoding. It supports bosonic qubits and discrete-variable photon-number states rather than fully continuous-variable encodings.

Consequently, Perceval does not support full CV operations like position or momentum displacements but rather targets DV bosonic modes. This design choice fits well with certain photonic hardware architectures and complements the approaches taken by bosonic-qiskit and Strawberry Fields.

Notably, there exists a converter that translates Qiskit circuits into Perceval programs, facilitating interoperability between these frameworks. However, this converter explicitly highlights that Perceval only supports discrete-variable (DV) photonic models and does not provide continuous-variable (CV) gate support.

Reference:

- `https://perceval.quandela.net/docs/v0.12/reference/qiskit_converter.html`

Summary

- **Bosonic-Qiskit** provides hybrid CV-DV gates but lacks native quadratic phase, position, and momentum displacement gates.
- **Strawberry Fields** offers full native support for fundamental CV gates, making it ideal for CV-specific algorithms.
- **Perceval** focuses on discrete-variable bosonic encodings, supporting photon-number states without continuous-variable gate sets.

This comparison highlights the complementary roles of these software packages in advancing quantum computing research across CV, DV, and hybrid regimes.

4 Comparison of StrawberryFields and Bosonic-Qiskit

Therefore, we can only compare SF and BQ in CV as SF only supports CV. Even it we make use of PennyLane which supports DV like qiskit, and it makes use of SF as an add-on, it still suffers from lacking hybrid gates like conditional-displacement. Thus, BQ is better in terms of working hybrid CV-DV quantum computing.

Strawberry Fields Gate and Operation Set

Single-mode Gates

Gate	Description
`Dgate(r,` ϕ`)`	Displacement gate in phase space
`Xgate(x)`	Position displacement
`Zgate(p)`	Momentum displacement
`Sgate(r,` ϕ`)`	Squeezing gate
`Rgate(`θ`)`	Rotation gate
`Pgate(s)`	Quadratic phase gate
`Vgate(`γ`)`	Cubic phase gate
`Fouriergate()`	Fourier transform

Two-mode Gates

Gate	Description
`BSgate(`θ`,` ϕ`)`	Beam splitter gate
`MZgate(`$\phi_{\mathtt{in}}$`,` $\phi_{\mathtt{ex}}$`)`	Mach-Zehnder interferometer
`S2gate(r,` ϕ`)`	Two-mode squeezing gate
`CXgate(s)`	Controlled addition in position basis
`CZgate(s)`	Controlled phase gate in position basis
`CKgate(`κ`)`	Cross-Kerr interaction

State Preparation

Operation	Description
`Vacuum()`	Vacuum state
`Coherent(r,` ϕ`)`	Coherent state
`Squeezed(r, p)`	Squeezed vacuum state
`DisplacedSqueezed(...)`	Displaced squeezed state
`Thermal(n)`	Thermal state
`Fock(n)`	Fock state
`Catstate(...)`	Cat state
`GKP(...)`	GKP state (finite energy)
`Ket(state)`	Custom Fock-basis ket state
`DensityMatrix(state)`	Custom Fock-basis density matrix
`Gaussian(V)`	Gaussian state with covariance matrix

Measurements

Measurement	Description
`MeasureFock()`	Photon counting (Fock basis)
`MeasureThreshold()`	Threshold detector (0 or $\geq$)
`MeasureHomodyne(`ϕ`)`	Homodyne (quadrature) measurement
`MeasureHeterodyne()`	Heterodyne measurement
`MeasureX`	Homodyne in x
`MeasureP`	Homodyne in p
`MeasureHD`	Heterodyne (same as above)

Channels and Decompositions

Operation	Description
`LossChannel(T)`	Loss channel
`ThermalLossChannel(T, nbar)`	Thermal loss
`MSgate(...)`	Measurement-based squeezing
`PassiveChannel(T)`	Multimode passive operation
`Interferometer(U)`	General interferometer
`GraphEmbed(A)`	Graph embedding
`BipartiteGraphEmbed(A)`	Bipartite graph embedding
`GaussianTransform(S)`	Gaussian symplectic transformation

Bosonic Qiskit (C2QA) Gate and Operation Set

Core CV Gates and Operations

Method	Description
`cv_d`	Displacement gate
`cv_r`	Rotation gate
`cv_sq`	Squeezing gate
`cv_sq2`	Two-mode squeezing
`cv_sq3`	Three-mode squeezing
`cv_bs`	Beam splitter gate
`cv_c_bs`	Controlled phase beam splitter
`cv_sum`	Two-mode sum gate
`cv_c_sum`	Controlled sum gate
`cv_snap`	Selective Number-dependent Arbitrary Phase (SNAP)
`cv_c_multiboson_sampling`	SNAP variant for multi-boson sampling

Qubit-Mode Hybrid Gates

Method	Description
`cv_jc`	Jaynes–Cummings interaction
`cv_ajc`	Anti-Jaynes–Cummings interaction
`cv_ecd`	Echoed controlled displacement
`cv_rb`	Rabi interaction gate
`cv_c_r`	Qubit-controlled phase rotation
`cv_c_rx`, `cv_c_ry`	Qubit-controlled phase space rotations (σ_x, σ_y)
`cv_c_d`	Conditional displacement
`cv_c_sq`	Conditional squeezing
`cv_c_schwinger`	General controlled Schwinger gate

Other Functions

Method	Description
`cv_initialize`	Initialize qumode to arbitrary state
`cv_measure`	Generic CV + qubit measurement
`cv_delay`	Identity operation (for timing / noise modeling)
`cv_gate_from_matrix`	Apply arbitrary matrix as gate
`cv_eswap`	Exponential SWAP gate
`cv_snapshot`	Store simulator snapshot for Wigner visualization
`measure_x/y/z`	Qubit measurements via probe qubits

Gate Set Comparison

Operation Type	SF	BQ (C2QA)	Notes
Displacement (Dgate)	✓	✓	Both support full complex α
Rotation	✓	✓	Similar
Squeezing	✓	✓	SF supports ϕ, BQ usually only r
Cubic Phase	✓	✓	Both support non-Gaussianity
Beam Splitter	✓	✓	Both support 2-mode mixing
Fourier Transform	✓	~	SF has native gate
Kerr / Cross-Kerr	✓	~	BQ lacks native Kerr gate
SNAP	✗	✓	Only in BQ
Cat State, GKP, Thermal, etc.	✓	~	SF richer in state prep
Measurements (X, P, Fock)	✓	~	SF supports richer set
Interferometers	✓	~	SF has higher-level decompositions
Hybrid qubit-mode ops	~	✓	C2QA focus area

4.1 Comparison of Continuous-Variable Gate Sets: Strawberry Fields vs. Bosonic Qiskit

At first glance, the continuous-variable (CV) gate set provided by Strawberry Fields (SF) appears more comprehensive than that of Bosonic Qiskit (BQ). SF offers a broad range of single- and two-mode gates with explicit support for phase-space displacements, rotations, squeezing, and higher-order nonlinearities such as cubic phase gates.

However, many of these SF gates can be realized or approximated within BQ by suitable transformations and decompositions. In particular, several fundamental SF gates, such as the Z-gate, P-gate, and X-gate, correspond to specific displacement and rotation operations that can be implemented in BQ's framework. The following relations highlight how these SF gates map to BQ operations:

- **Momentum Displacement (Zgate):**

$$Z(p) = \exp(-ip\hat{x}) = D(ip)$$

  In BQ, the Z-gate corresponds to a displacement operator with purely imaginary amplitude $\alpha = ip$.

- **Quadratic Phase Gate (Pgate):**

$$P(s) = \exp\left(i\frac{s}{2}\hat{x}^2\right)$$

  This gate applies a quadratic phase shift in the position basis. It is a non-Gaussian operation when viewed from the Hamiltonian perspective and appears in many CV protocols, including GKP state preparation and CV quantum error correction. Although the $P(s)$ gate is native to Strawberry Fields (SF), it is not a built-in primitive in Bosonic Qiskit (BQ). However, it can be synthesized using the squeezing and rotation gates as follows.

  To derive this decomposition, we start by expressing the position operator $\hat{x}$ in terms of the ladder operators:

$$\hat{x} = \frac{1}{\sqrt{2}}(\hat{a} + \hat{a}^\dagger) \Rightarrow \hat{x}^2 = \frac{1}{2}\left(\hat{a}^2 + \hat{a}^{\dagger 2} + \hat{a}^\dagger\hat{a} + \hat{a}\hat{a}^\dagger\right)$$

  Recall the bosonic commutation relation: $[\hat{a}, \hat{a}^\dagger] = 1$, which implies:

$$\hat{a}\hat{a}^\dagger = \hat{a}^\dagger\hat{a} + 1 \Rightarrow \hat{x}^2 = \frac{1}{2}\left(\hat{a}^2 + \hat{a}^{\dagger 2} + 2\hat{a}^\dagger\hat{a} + 1\right)$$

Therefore,

$$P(s) = \exp\left(i\frac{s}{4}\left(\hat{a}^2 + \hat{a}^{\dagger 2} + 2\hat{n} + 1\right)\right)$$

where $\hat{n} = \hat{a}^\dagger\hat{a}$ is the number operator. Let us now factor this exponential (approximately, since these operators don't all commute) using the **Baker–Campbell–Hausdorff** (BCH) approximation:

$$P(s) \approx e^{i\frac{s}{4}\hat{a}^2} \cdot e^{i\frac{s}{4}\hat{a}^{\dagger 2}} \cdot e^{i\frac{s}{2}\hat{n}} \cdot e^{i\frac{s}{4}}$$

Each of the terms in this decomposition corresponds to gates available or constructible in BQ:

- $e^{i\frac{s}{4}\hat{a}^2}$ and $e^{i\frac{s}{4}\hat{a}^{\dagger 2}}$ are components of a **squeezing operator**:

  $$S(r) = \exp\left(\frac{1}{2}(r\hat{a}^2 - r^*\hat{a}^{\dagger 2})\right)$$

  With appropriate phase rotation, one can simulate combinations of these terms using squeezing and rotations.
- $e^{i\frac{s}{2}\hat{n}}$ is a **phase rotation gate** (also called the $R(\theta)$ gate):

  $$R(\theta) = \exp(i\theta\hat{n})$$

- The global phase term $e^{i\frac{s}{4}}$ is physically irrelevant in most contexts.

Therefore, while $P(s)$ is not primitive in BQ, one can approximate it by composing:

1. A squeezing gate with carefully chosen phase,
2. Followed by a rotation gate $R(\theta)$,
3. Possibly combined with gate conjugation sequences.

Alternative method (Hamiltonian engineering): In simulation mode or analog quantum devices, one may directly implement the Hamiltonian $\hat{H} = \hat{x}^2$ and evolve under it for time t, effectively generating $P(s)$ via:

$$U(t) = \exp\left(-it\hat{H}\right) = \exp\left(-it\hat{x}^2\right) = P(-2t)$$

This approach can be taken in BQ if custom Hamiltonians are supported.

- **Position Displacement (Xgate):**

  $$X(x) = \exp(-ix\hat{p}) = D(x)$$

  The X-gate corresponds to a displacement operator with purely real amplitude $\alpha = x$ in BQ.

- **Rotation (Rgate):**

  $$R(\theta) = \exp(i\theta\hat{a}^\dagger\hat{a})$$

  Both SF and BQ support phase space rotations as native gates.

These relations show that the seemingly richer gate set of SF can often be expressed through compositions of displacement, rotation, and squeezing gates available in BQ. This mapping enables users to leverage BQ's framework for simulating CV operations, despite the difference in native gate nomenclature and implementation details.

In summary, while SF provides an explicit and broad CV gate set tailored for photonic quantum computing, BQ's gate set, although initially appearing more limited, is flexible enough to approximate many of these operations through transformations and gate decompositions.

5 Conclusion

I can conclude that so far the only available platform for hybrid is BQ, but SF (or even pennylane which seems to be able to handle hybrid cv-dv, however, it doesn't support controlled cv operations) only supports bosonic cv and perceval (quandela) only supports bosonic dv.