



QUANTINUUM  
GUPPY

**Pythonic Quantum-Classical Programming**

Agustín Borgna et al.

# A next generation programming language

- Pythonic syntax
- Rich control flow
- Hybrid runtime

$V_3 = (I + 2iZ) / \sqrt{5}$   
arXiv:1311.1074

```
@guppy
def v3_rus(q: qubit) → int:
    attempt = 0
    while attempt < comptime(MAX_ATTEMPTS):
        attempt += 1
        a, b = plus_q(), plus_q()

        toffoli(a, b, q)
        s(q)
        toffoli(a, b, q)

        a_x, b_x = x_measure(a), x_measure(b)

        if not (a_x or b_x):
            break

        z(q)

    return attempt
```

# Familiar type **abstractions**

- Expressive type system
- User-defined structs
- Generic types

```
@guppy.struct
class PauliString(Generic[n]):
    xs: array[bool, n]
    zs: array[bool, n]

@guppy
def __eq__(self: PauliString[n], other: PauliString[n]) → bool:
    same_xs = array_eq(self.xs, other.xs)
    same_zs = array_eq(self.zs, other.zs)
    return same_xs and same_zs
```



# Safe and friendly

- Static types
- Linear type system
- Errors caught at compile time
- Rust-inspired ownership system

```
@guppy
def drop_qubit() → bool:
    _q = qubit()
    return True

drop_qubit.check()
```

```
Error: Drop violation (at /Users/aborgna/src/guppylang/demo.py:74:4)
|
72 | @guppy
73 | def drop_qubit() -> bool:
74 |     _q = qubit()
    |     ^^ Variable `_q` with non-droppable type `qubit` is leaked

Help: Make sure that `_q` is consumed or returned to avoid the leak

Guppy compilation failed due to 1 previous error
```

# Hybrid emulation with **Selene**

- Run and debug Guppy kernels
- Estimate program resources



# Thanks!



 [guppylang.org](https://guppylang.org)

 [CQCL/guppylang](https://github.com/CQCL/guppylang)