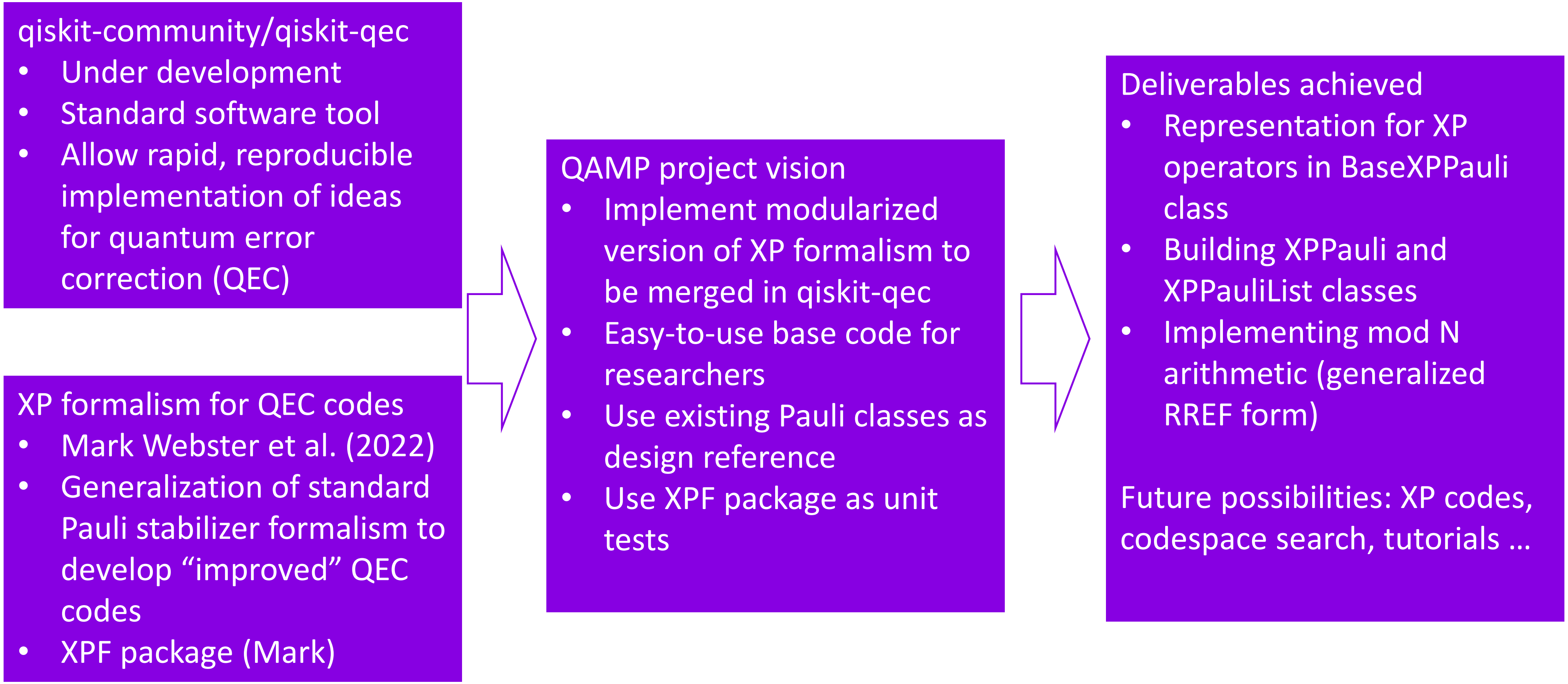
Qiskit

QAMP Fall 2022
Building out Qiskit-QEC: XP Formalism

Members: **Dhruv Bhatnagar**, Ruihao Li

Mentors: Grace Harper (IBM), Drew Vandeth (IBM)

Big Picture



qiskit-community/qiskit-qec

- Under development
- Standard software tool
- Allow rapid, reproducible implementation of ideas for quantum error correction (QEC)

XP formalism for QEC codes

- Mark Webster et al. (2022)
- Generalization of standard Pauli stabilizer formalism to develop “improved” QEC codes
- XPF package (Mark)

QAMP project vision

- Implement modularized version of XP formalism to be merged in qiskit-qec
- Easy-to-use base code for researchers
- Use existing Pauli classes as design reference
- Use XPF package as unit tests

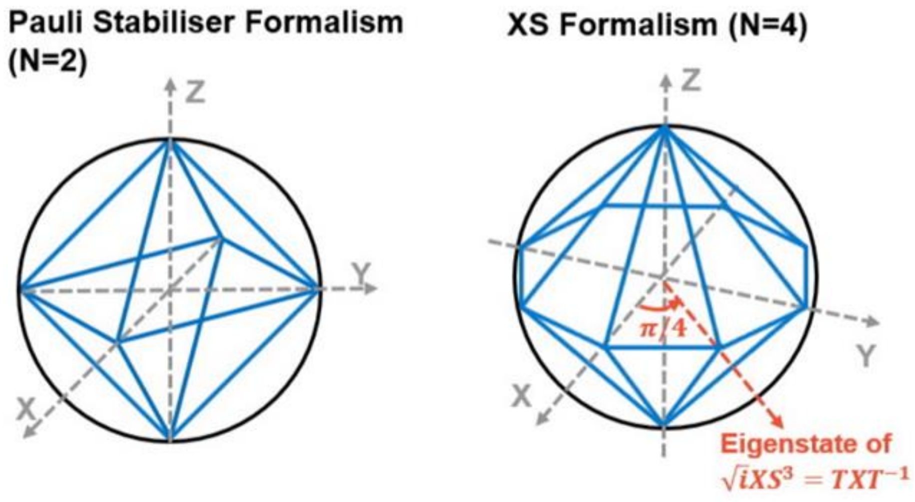
Deliverables achieved

- Representation for XP operators in BaseXPPauli class
- Building XPPauli and XPPauliList classes
- Implementing mod N arithmetic (generalized RREF form)

Future possibilities: XP codes, codespace search, tutorials …

Quick overview: XP stabilizer formalism

M.A. Webster, B.J. Brown, and S.D. Bartlett. Quantum 6, 815 (2022).
https://github.com/m-webster/XPFpackage



Pauli Stabiliser Formalism (N=2)

XS Formalism (N=4)

Eigenstate of $\sqrt{i}XS^3 = TXT^{-1}$

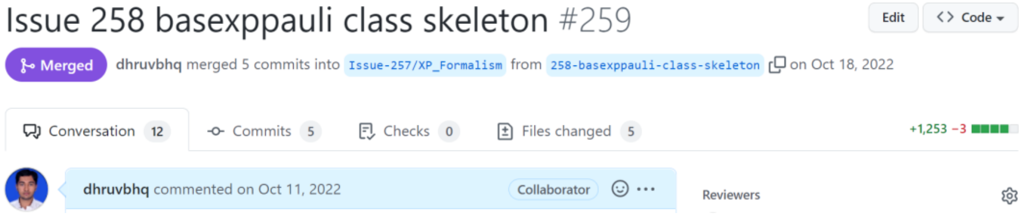
- Pauli Stabilizer Formalism

$$\langle iI, X, Z\rangle^{\otimes n}$$

- XP Formalism: To construct new QEC codes using fractional Z rotations to generate the stabilizer group

$$\langle iI, X, P\rangle^{\otimes n}, \omega = e^{i\pi/N}, P = diag(1, \omega^2)$$

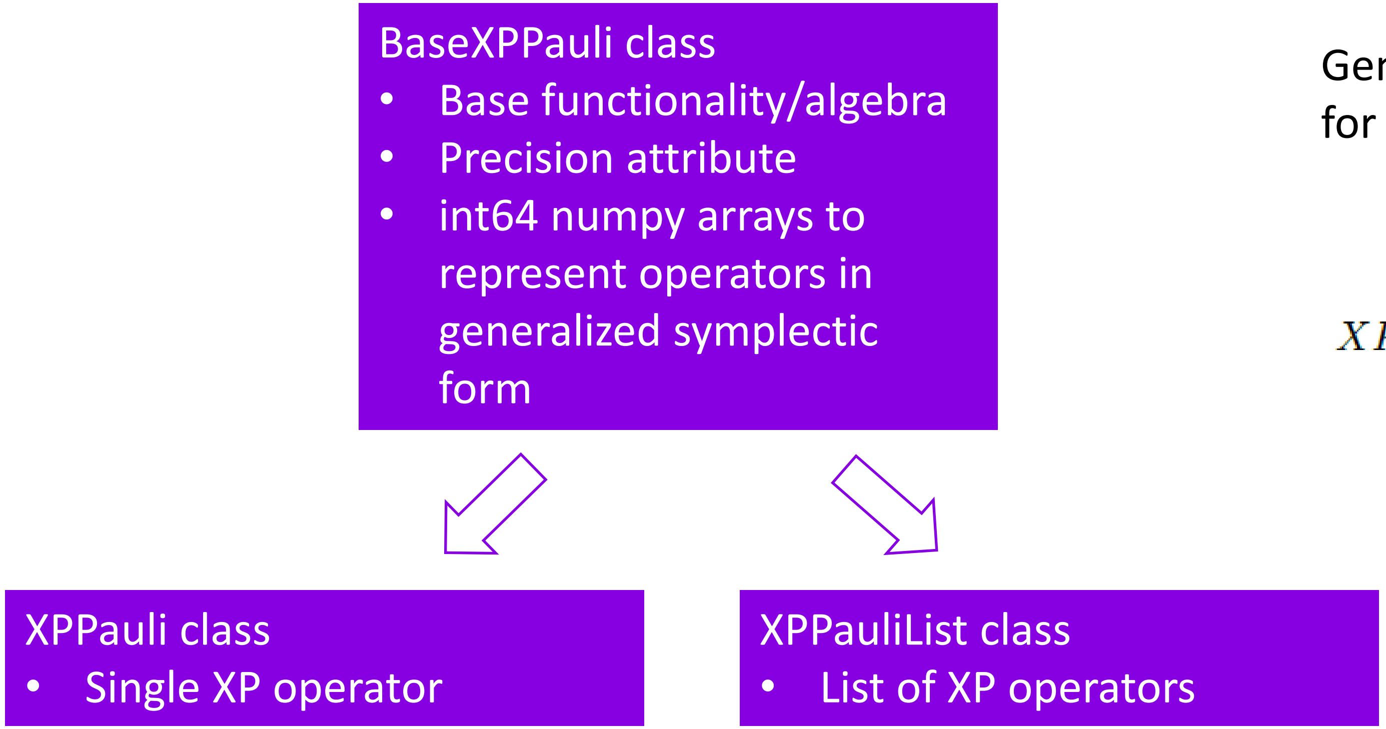
XP operators: BaseXPPauli class



Issue 258 basexppauli class skeleton #259

dhruvbhq merged 5 commits into `Issue-257/XP_Formalism` from `258-basexppauli-class-skeleton` on Oct 18, 2022

dhruvbhq commented on Oct 11, 2022



BaseXPPauli class

- Base functionality/algebra
- Precision attribute
- int64 numpy arrays to represent operators in generalized symplectic form

XPPauli class

- Single XP operator

XPPauliList class

- List of XP operators

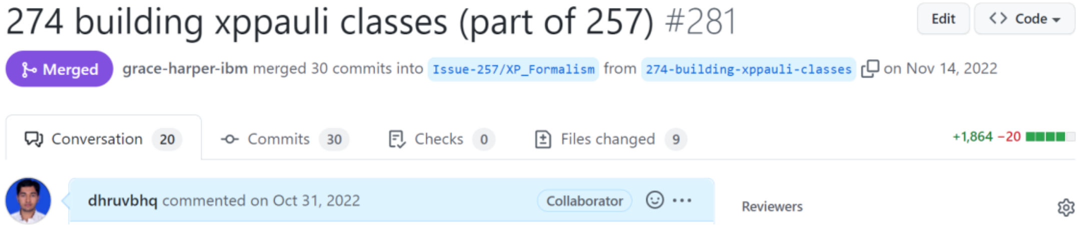
Generalized symplectic vector representation for XP operators, with an example.

$$XP_N(\mathbf{u}) := \omega^p \bigotimes_{0\leq i<n} X^{\mathbf{x}[i]} P^{\mathbf{z}[i]}$$

$$XP_N(p|\mathbf{x}|\mathbf{z}) = XP_N(p \bmod 2N|\mathbf{x} \bmod 2|\mathbf{z} \bmod N)$$

$$A = XP_8(12|1110000|0040000)$$

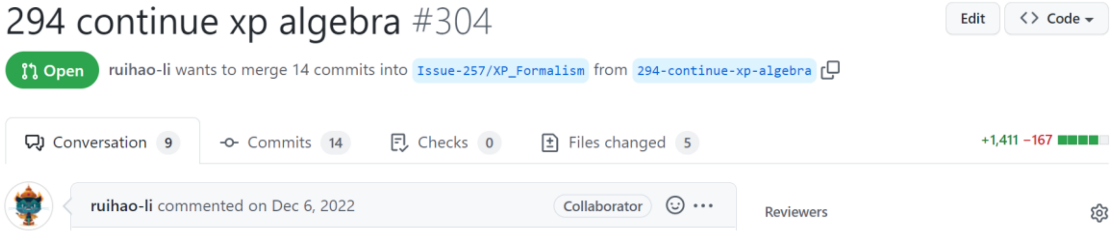
XP operator algebra implementation



274 building xppauli classes (part of 257) #281

Merged grace-harper-ibm merged 30 commits into `Issue-257/XP_Formalism` from `274-building-xppauli-classes` on Nov 14, 2022

dhruvbhq commented on Oct 31, 2022



294 continue xp algebra #304

Open ruihao-li wants to merge 14 commits into `Issue-257/XP_Formalism` from `294-continue-xp-algebra`

ruihao-li commented on Dec 6, 2022

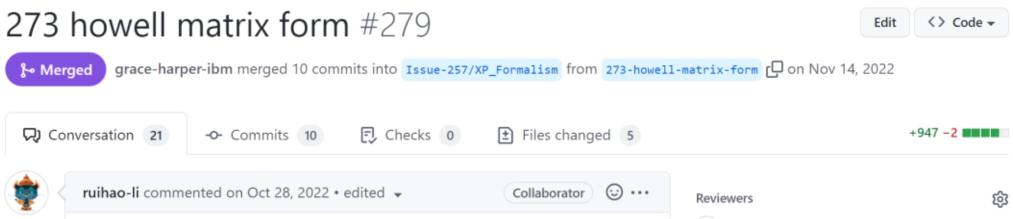
To compute stuff using XP operators

- Determine unique vector representation of an XP operator
- Determine if the operator is diagonal
- Rescale operator precision
- Calculate products, inverses, commutation relations of XP operators, and more

Guiding principles

- Higher level python method for the user
  - Checks and easy to use
- Lower level python method
  - Functionality without checks
- in-place option
- Exceptions
  - e.g. Rescaling precision
- Rewriting/refactoring XPF package
- Tests

Modulo N arithmetic. String representations



273 howell matrix form #279

Edit | <> Code

Merged grace-harper-ibm merged 10 commits into `Issue-257/XP_Formalism` from `273-howell-matrix-form` on Nov 14, 2022

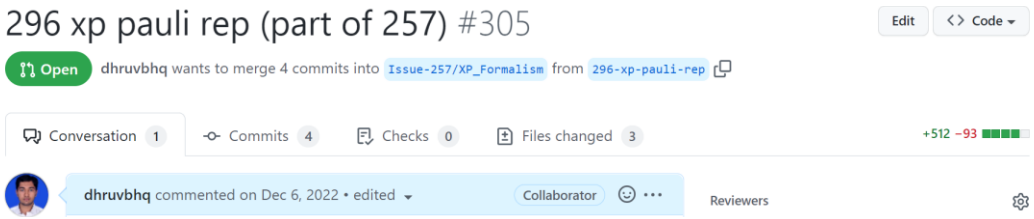
Conversation 21 | Commits 10 | Checks 0 | Files changed 5 | +947 −2

ruihao-li commented on Oct 28, 2022 • edited | Collaborator

Reviewers

Modular arithmetics on ring Z/nZ: reside in qiskit_qec/arithmetic/modn.py

- Extended Euclidean algorithm for finding the greatest common divisor (gcd_ext)
- Quotient (quo)
- Divisor (div)



296 xp pauli rep (part of 257) #305

Edit | <> Code

Open dhruvbhq wants to merge 4 commits into `Issue-257/XP_Formalism` from `296-xp-pauli-rep`

Conversation 1 | Commits 4 | Checks 0 | Files changed 3 | +512 −93

dhruvbhq commented on Dec 6, 2022 • edited | Collaborator

Reviewers

XP operator to string representation: reside in qiskit_qec/utils/xp_pauli_rep.py

- `INDEX_SYNTAX` : `'XP8((w,12)(X(P,4))2(X)1(X)0)'`
- `XP_SYMPLECTIC_SYNTAX` : `'XP8(12|1 1 1 0 0 0 0|0 0 4 0 0 0 0)'`
- `PRODUCT_SYNTAX` : `'XP8((w,12)(I)(I)(I)(I)(X(P,4))(X)(X))'`
- `LATEX_SYNTAX` : `'XP_{8}((w,12)(XP^{4})_{2}(X)_{1}(X)_{0})'`

Summary

The current implementation achieves:

- Representing XP operators in the style of existing framework of qiskit-qec
- Algebra of XP operators
- Modulo N arithmetic, useful for further algorithms for XP codes

This serves as a base of methods to enable further implementation, like algorithms from XPF package for:

- Whether a given set of generators identify a valid codespace, dimension of codespace, codewords
- Which sets of operators stabilize the same codespace?
- How do we find all transversal logical operators for a given code?

References

- M.A. Webster, B.J. Brown, and S.D. Bartlett. Quantum 6, 815 (2022).
- https://github.com/m-webster/XPFpackage
- https://github.com/qiskit-advocate/qamp-fall-22/issues/15
- https://www.qiskit.org