Qudit Hypergraphs and Function Encoding

Alexander J. Heilman

Lebanon Valley College

Moravian College
24 February 2018

with support from LVC Arnold research grants

Why Quantum Computers?

- Improve communications security

Why Quantum Computers?

- Improve communications security
- Improve efficiency of large computational tasks

Why Quantum Computers?

- Improve communications security
- Improve efficiency of large computational tasks
- Possibly solve new problems

Information Processing

In a classical computer, each 'bit' of information holds one of two values, either 1 or 0.

$$|\psi\rangle = 0 \text{ or } 1$$

Information Processing

In a classical computer, each 'bit' of information holds one of two values, either 1 or 0.

$$|\psi\rangle = 0 \text{ or } 1$$

Classical computers process well-defined bit strings and yield bits,

$$0110100100 \underset{process}{\overset{Computer}{\longrightarrow}} 1$$

Information Processing

In a classical computer, each 'bit' of information holds one of two values, either 1 or 0.

$$|\psi\rangle = 0 \text{ or } 1$$

Classical computers process well-defined bit strings and yield bits,

$$0110100100 \xrightarrow[\textit{process}]{\textit{Computer}} 1$$

Quantum bits (qu**b**its) differ from bits as they can also be in a superposition of the two states:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

Information Processing

In a classical computer, each 'bit' of information holds one of two values, either 1 or 0.

$$|\psi\rangle = 0 \text{ or } 1$$

Classical computers process well-defined bit strings and yield bits,

$$0110100100 \xrightarrow[process]{Computer} 1$$

Quantum bits (qu**b**its) differ from bits as they can also be in a superposition of the two states:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$$

Further, qu**d**its can be in a superposition of d different states:

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \alpha_2 |2\rangle + \ldots + \alpha_{d-1} |d-1\rangle$$

Information Processing (cont.)

The collective system of physically close states is described by the tensor-product of their states. A two qubit state would look something like:

$$(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) = |00\rangle + |01\rangle + |10\rangle + |11\rangle$$

Information Processing (cont.)

The collective system of physically close states is described by the tensor-product of their states. A two qubit state would look something like:

$$(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) = |00\rangle + |01\rangle + |10\rangle + |11\rangle$$

A quantum computer processes these superpositions of states and yield some bit of information,

$$|00\rangle + |01\rangle + |10\rangle + |11\rangle \xrightarrow[\textit{process}]{\textit{Quantum}} 0 \text{ or } 1$$

Information Processing (cont.)

The collective system of physically close states is described by the tensor-product of their states. A two qubit state would look something like:

$$(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) = |00\rangle + |01\rangle + |10\rangle + |11\rangle$$

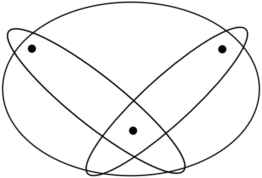
A quantum computer processes these superpositions of states and yield some bit of information,

$$|00\rangle + |01\rangle + |10\rangle + |11\rangle \xrightarrow[\textit{process}]{\textit{Quantum}} 0 \text{ or } 1$$

One class of quantum states that has proven useful for quantum computation are hypergraph states

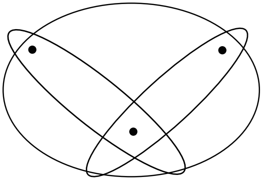
Hypergraph States

Hypergraphs are a combination of vertices (dots) and edges (circles).



Hypergraph States

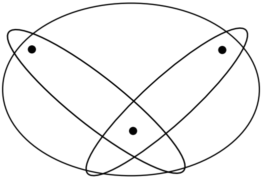
Hypergraphs are a combination of vertices (dots) and edges (circles).



Each dot represents a qubit and each circle represents a gate action on the enclosed qubits, applying some phase ω ($e^{2\pi i/d}$) to certain elements of the superposition.

Hypergraph States

Hypergraphs are a combination of vertices (dots) and edges (circles).

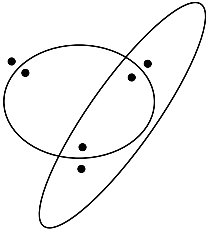


Each dot represents a qubit and each circle represents a gate action on the enclosed qubits, applying some phase ω ($e^{2\pi i/d}$) to certain elements of the superposition.

$$|\psi\rangle = |000\rangle + |001\rangle + |010\rangle - |011\rangle + |100\rangle + |101\rangle - |110\rangle - |111\rangle$$

Function Encoding

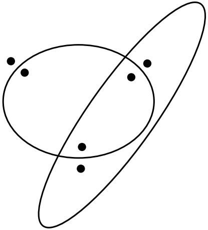
Qudit hypergraph states can easily become cumbersome to describe in their entirety.



For a state with just three qutrits ($d = 3$), there are 81 possible state vectors required to describe the state:

Function Encoding

Qudit hypergraph states can easily become cumbersome to describe in their entirety.

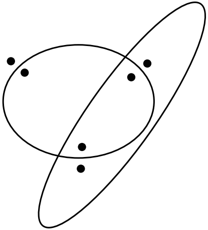


For a state with just three qutrits ($d = 3$), there are 81 possible state vectors required to describe the state:

$$= |000\rangle + |001\rangle + |002\rangle + |010\rangle + |011\rangle + |012\rangle + |020\rangle + |021\rangle + |022\rangle + |100\rangle + \omega |101\rangle + \omega |102\rangle + |110\rangle + \omega^2 |111\rangle + |112\rangle + |120\rangle + |121\rangle + \omega^2 |122\rangle + |200\rangle + \omega |201\rangle + \omega |202\rangle + |210\rangle + |211\rangle + \omega^2 |212\rangle + |220\rangle + \omega^2 |221\rangle + |222\rangle$$

Function Encoding (cont.)

Alternatively, these states can be described with uniquely encoded functions (mod d).

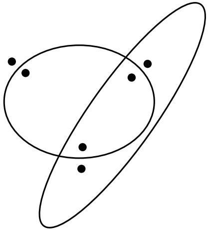


Thus, the previous state is more eloquently described:

$$f(x, y, z) = xyz + x^2y^2$$

Function Encoding (cont.)

Alternatively, these states can be described with uniquely encoded functions (mod d).



Thus, the previous state is more eloquently described:

$$f(x, y, z) = xyz + x^2y^2$$

The function, f, determines the phase for each individual element of the superposition. For example:

$$f(1, 2, 2) = 1 \cdot 2 \cdot 2 + 1^2 \cdot 2^2 = 2 \longrightarrow \omega^2 |122\rangle$$

Encoding Process

The general map for encoding a hypergraph as a function is as follows:

$$\overset{state}{|\psi\rangle} \underset{log\omega}{\longrightarrow} \overset{c-vec}{|c\rangle} \underset{S^{-1}matrix}{\longrightarrow} \overset{a-vec}{|a\rangle}$$

Where the a vector ($|a\rangle$) describes the coefficients of the state's function. Returning to the state is similar,

$$|\psi\rangle \underset{exp\omega}{\longleftarrow} |c\rangle \underset{S\,matrix}{\longleftarrow} |a\rangle$$

Encoding Process

The general map for encoding a hypergraph as a function is as follows:

$$\overset{state}{|\psi\rangle} \underset{log\omega}{\longrightarrow} \overset{c-vec}{|c\rangle} \underset{S^{-1}matrix}{\longrightarrow} \overset{a-vec}{|a\rangle}$$

Where the a vector ($|a\rangle$) describes the coefficients of the state's function. Returning to the state is similar,

$$|\psi\rangle \underset{exp\omega}{\longleftarrow} |c\rangle \underset{S\,matrix}{\longleftarrow} |a\rangle$$

The step from $|c\rangle$ to $|\psi\rangle$ is an entry-wise operation on the elements of a vector, while step $|a\rangle$ to $|c\rangle$ is achieved by multiplication of the S matrix

Encoding Process (cont.)

For example:

$$|\psi\rangle = |00\rangle + |01\rangle + |02\rangle + |10\rangle + \omega\,|11\rangle + \omega^2\,|12\rangle + |20\rangle + \omega^2\,|21\rangle + \omega\,|22\rangle$$

$$\downarrow$$

$$|\psi\rangle = \begin{bmatrix}1 & 1 & 1 & 1 & \omega & \omega^2 & 1 & \omega^2 & \omega\end{bmatrix}$$

$$\downarrow log_\omega$$

$$|c\rangle = \begin{bmatrix}0 & 0 & 0 & 0 & 1 & 2 & 0 & 2 & 1\end{bmatrix}$$

$$\downarrow S^{-1} Matrix$$

$$|a\rangle = \begin{bmatrix}0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\end{bmatrix}$$

$x^0y^0 \mid x^0y^1 \mid x^0y^2 \mid x^1y^0 \mid x^1y^1 \mid x^1y^2 \mid x^2y^0 \mid x^2y^1 \mid x^2y^2$

$$f = xy$$

S Matrix

The S matrix has the form of a mod d Vandermonde matrix:

$$S_d = \begin{bmatrix} 0^0 & 0^1 & 0^2 & \cdots & 0^{d-1} \\ 1^0 & 1^1 & 1^2 & \cdots & 1^{d-1} \\ 2^0 & 2^1 & 2^2 & \cdots & 2^{d-1} \\ 3^0 & 3^1 & 3^2 & \cdots & 3^{d-1} \\ \vdots & \vdots & \vdots & \ddots & \\ (d-1)^0 & (d-1)^1 & (d-1)^2 & & (d-1)^{d-1} \end{bmatrix} \mod d$$

S Matrix

The S matrix has the form of a mod d Vandermonde matrix:

$$S_d = \begin{bmatrix} 0^0 & 0^1 & 0^2 & \cdots & 0^{d-1} \\ 1^0 & 1^1 & 1^2 & \cdots & 1^{d-1} \\ 2^0 & 2^1 & 2^2 & \cdots & 2^{d-1} \\ 3^0 & 3^1 & 3^2 & \cdots & 3^{d-1} \\ \vdots & \vdots & \vdots & \ddots & \\ (d-1)^0 & (d-1)^1 & (d-1)^2 & & (d-1)^{d-1} \end{bmatrix} \bmod d$$

For example, for d=5

$$S_5 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 3 & 1 \\ 1 & 3 & 4 & 2 & 1 \\ 1 & 4 & 1 & 4 & 1 \end{bmatrix} \bmod 5$$

Generalized Local Pauli Matrices

The Paulis are a basis for $d \times d$ operators (actions) on individual qudits. They can be used to describe all local actions on a state. For qubits ($d = 2$), they are standard:

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Generalized Local Pauli Matrices

The Paulis are a basis for $d \times d$ operators (actions) on individual qudits. They can be used to describe all local actions on a state. For qubits ($d = 2$), they are standard:

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

We've considered one possible generalization of the Paulis for local d dimensional actions, the Ps (generalized X) and Qs (generalized Z). For example:

$$P_{01} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} Q_0 = \begin{bmatrix} \omega & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Generalized Local Pauli Matrices (cont.)

The Ps act as subspace permutations and the Qs act as local phase shifts on the state vector ($|\psi\rangle$).

$$P_{24} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad Q_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \omega & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

For any dimension d, there are $d! - 1$ Ps and d Qs.

Generalized Local Pauli Matrices (cont.)

The Ps act as subspace permutations and the Qs act as local phase shifts on the state vector ($|\psi\rangle$).

$$P_{24} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad Q_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \omega & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

For any dimension d, there are $d! - 1$ Ps and d Qs.

Pauli operators are important for error-correcting applications.

GLP Actions on Encoded Functions

In order to see the effects of the generalized Paulis on encoded functions, we must consider the map of encoding,

$$|a'\rangle \leftarrow S^{-1} \leftarrow log_{\omega} \leftarrow |\psi\rangle \leftarrow exp_{\omega} \leftarrow S \leftarrow |a\rangle$$

GLP Actions on Encoded Functions

In order to see the effects of the generalized Paulis on encoded functions, we must consider the map of encoding,

$$|a'\rangle \leftarrow S^{-1} \leftarrow log_\omega \leftarrow |\psi\rangle \leftarrow exp_\omega \leftarrow S \leftarrow |a\rangle$$

As P is a permutation that only swaps subspaces and doesn't perform any other action, the actions of the Ps on functions is described by

$$S^{-1}PS$$

GLP Actions on Encoded Functions

In order to see the effects of the generalized Paulis on encoded functions, we must consider the map of encoding,

$$|a'\rangle \leftarrow S^{-1} \leftarrow log_\omega \leftarrow |\psi\rangle \leftarrow exp_\omega \leftarrow S \leftarrow |a\rangle$$

As P is a permutation that only swaps subspaces and doesn't perform any other action, the actions of the Ps on functions is described by

$$S^{-1}PS$$

For example,

$$P_{01} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 1 & 1 \\ 0 & 1 & 2 & 1 & 1 \\ 0 & 1 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} P_{24} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 2 & 1 & 0 \\ 0 & 4 & 0 & 2 & 0 \\ 0 & 4 & 4 & 3 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

GLP Actions on Encoded Functions (cont.)

The Ps effectively transmute a variable into a collection of different powers of that variable:

GLP Actions on Encoded Functions (cont.)

The Ps effectively transmute a variable into a collection of different powers of that variable:

$$P_{01}\left|x\right\rangle \to \left|x^3 + x^2 + 2x + 1\right\rangle$$
$$P_{24}\left|x^3\right\rangle \to \left|x + 2x^2 + 3x^3\right\rangle$$

GLP Actions on Encoded Functions (cont.)

The Ps effectively transmute a variable into a collection of different powers of that variable:

$$P_{01}\left|x\right\rangle \rightarrow \left|x^3 + x^2 + 2x + 1\right\rangle$$
$$P_{24}\left|x^3\right\rangle \rightarrow \left|x + 2x^2 + 3x^3\right\rangle$$

The Qs perform an entirely different operation on the functions, adding single variable terms to the function:

GLP Actions on Encoded Functions (cont.)

The Ps effectively transmute a variable into a collection of different powers of that variable:

$$P_{01}\,|x\rangle \rightarrow \left|x^3 + x^2 + 2x + 1\right\rangle$$
$$P_{24}\,\left|x^3\right\rangle \rightarrow \left|x + 2x^2 + 3x^3\right\rangle$$

The Qs perform an entirely different operation on the functions, adding single variable terms to the function:

$$Q_0\,\left|x^2\right\rangle \rightarrow \left|1 + x^2 + 4x^4\right\rangle$$
$$Q_3\,|1\rangle \rightarrow \left|1 + 3x + x^2 + 2x^3 + x^4\right\rangle$$

GLP Actions on Encoded Functions (cont.)

The Ps effectively transmute a variable into a collection of different powers of that variable:

$$P_{01} \left|x\right\rangle \to \left|x^3 + x^2 + 2x + 1\right\rangle$$
$$P_{24} \left|x^3\right\rangle \to \left|x + 2x^2 + 3x^3\right\rangle$$

The Qs perform an entirely different operation on the functions, adding single variable terms to the function:

$$Q_0 \left|x^2\right\rangle \to \left|1 + x^2 + 4x^4\right\rangle$$
$$Q_3 \left|1\right\rangle \to \left|1 + 3x + x^2 + 2x^3 + x^4\right\rangle$$

These actions can then be used to identify states that are equivalent under local actions.

GLP Actions on Encoded Functions (cont.)

The Ps effectively transmute a variable into a collection of different powers of that variable:

$$P_{01}\left|x\right\rangle \rightarrow \left|x^3 + x^2 + 2x + 1\right\rangle$$
$$P_{24}\left|x^3\right\rangle \rightarrow \left|x + 2x^2 + 3x^3\right\rangle$$

The Qs perform an entirely different operation on the functions, adding single variable terms to the function:

$$Q_0\left|x^2\right\rangle \rightarrow \left|1 + x^2 + 4x^4\right\rangle$$
$$Q_3\left|1\right\rangle \rightarrow \left|1 + 3x + x^2 + 2x^3 + x^4\right\rangle$$

These actions can then be used to identify states that are equivalent under local actions.For two qutrits (d=3), there are 9 equivalence classes:

$$f(x,y) \equiv 0,\ xy,\ xy^2,\ x^2y,\ xy^2 + x^2y,\ x^2y^2,$$
$$2x^2y^2,\ x^2y^2 + xy,\ 2x^2y^2 + xy$$

GLP Actions on Encoded Functions (cont.)

The Ps effectively transmute a variable into a collection of different powers of that variable:

$$P_{01} |x\rangle \rightarrow \left|x^3 + x^2 + 2x + 1\right\rangle$$
$$P_{24} \left|x^3\right\rangle \rightarrow \left|x + 2x^2 + 3x^3\right\rangle$$

The Qs perform an entirely different operation on the functions, adding single variable terms to the function:

$$Q_0 \left|x^2\right\rangle \rightarrow \left|1 + x^2 + 4x^4\right\rangle$$
$$Q_3 |1\rangle \rightarrow \left|1 + 3x + x^2 + 2x^3 + x^4\right\rangle$$

These actions can then be used to identify states that are equivalent under local actions.For two qutrits (d=3), there are 9 equivalence classes:

$$f(x,y) \equiv 0,\ xy,\ xy^2,\ x^2y,\ xy^2 + x^2y,\ x^2y^2,$$
$$2x^2y^2,\ x^2y^2 + xy,\ 2x^2y^2 + xy$$

All states n=2, d=3 can be transformed into one of the above functions with the GLP

GLP Stabilizers

Some functions are left unaffected by certain operators. These operators are said to stabilize the function.

GLP Stabilizers

Some functions are left unaffected by certain operators. These operators are said to stabilize the function.

$$P_{24} \left|x + x^2 + x^3\right\rangle \rightarrow \left|x + x^2 + x^3\right\rangle \quad \text{(d=5)}$$

GLP Stabilizers

Some functions are left unaffected by certain operators. These operators are said to stabilize the function.

$$P_{24} \left|x + x^2 + x^3\right\rangle \rightarrow \left|x + x^2 + x^3\right\rangle \quad \text{(d=5)}$$

Qs cannot form stabilizers on their own, but can if multiplied with Ps.

GLP Stabilizers

Some functions are left unaffected by certain operators. These operators are said to stabilize the function.

$$P_{24} \left|x + x^2 + x^3\right\rangle \to \left|x + x^2 + x^3\right\rangle \text{ (d=5)}$$

Qs cannot form stabilizers on their own, but can if multiplied with Ps.

$$Q_2 * P_{012} \otimes Q_1^2 * P_{021} \left|xy^2 + x^2y\right\rangle \to \left|xy^2 + x^2y\right\rangle \text{ (d=3)}$$

GLP Stabilizers

Some functions are left unaffected by certain operators. These operators are said to stabilize the function.

$$P_{24} \left|x + x^2 + x^3\right\rangle \rightarrow \left|x + x^2 + x^3\right\rangle \quad \text{(d=5)}$$

Qs cannot form stabilizers on their own, but can if multiplied with Ps.

$$Q_2 * P_{012} \otimes Q_1^2 * P_{021} \left|xy^2 + x^2y\right\rangle \rightarrow \left|xy^2 + x^2y\right\rangle \quad \text{(d=3)}$$

Known stabilizers can then be used to create families of stabilized states.

GLP Stabilizers

Some functions are left unaffected by certain operators. These operators are said to stabilize the function.

$$P_{24} \left|x + x^2 + x^3\right\rangle \rightarrow \left|x + x^2 + x^3\right\rangle \text{ (d=5)}$$

Qs cannot form stabilizers on their own, but can if multiplied with Ps.

$$Q_2 * P_{012} \otimes Q_1^2 * P_{021} \left|xy^2 + x^2y\right\rangle \rightarrow \left|xy^2 + x^2y\right\rangle \text{ (d=3)}$$

Known stabilizers can then be used to create families of stabilized states.

$$P_{01} \left|x + 2x^2\right\rangle = \left|x + 2x^2\right\rangle \text{ (d=3) ,}$$

GLP Stabilizers

Some functions are left unaffected by certain operators. These operators are said to stabilize the function.

$$P_{24} \left|x + x^2 + x^3\right\rangle \to \left|x + x^2 + x^3\right\rangle \text{ (d=5)}$$

Qs cannot form stabilizers on their own, but can if multiplied with Ps.

$$Q_2 * P_{012} \otimes Q_1^2 * P_{021} \left|xy^2 + x^2y\right\rangle \to \left|xy^2 + x^2y\right\rangle \text{ (d=3)}$$

Known stabilizers can then be used to create families of stabilized states.

$$P_{01} \left|x + 2x^2\right\rangle = \left|x + 2x^2\right\rangle \text{ (d=3) },$$
$$P_{02} \left|x + x^2\right\rangle = \left|x + x^2\right\rangle \text{ (d=3) },$$

GLP Stabilizers

Some functions are left unaffected by certain operators. These operators are said to stabilize the function.

$$P_{24} \left|x + x^2 + x^3\right\rangle \rightarrow \left|x + x^2 + x^3\right\rangle \ \text{(d=5)}$$

Qs cannot form stabilizers on their own, but can if multiplied with Ps.

$$Q_2 * P_{012} \otimes Q_1^2 * P_{021} \left|xy^2 + x^2y\right\rangle \rightarrow \left|xy^2 + x^2y\right\rangle \ \text{(d=3)}$$

Known stabilizers can then be used to create families of stabilized states.

$$P_{01} \left|x + 2x^2\right\rangle = \left|x + 2x^2\right\rangle \ \text{(d=3)} ,$$
$$P_{02} \left|x + x^2\right\rangle = \left|x + x^2\right\rangle \ \text{(d=3)} ,$$
$$P_{12} \left|x^2\right\rangle = \left|x^2\right\rangle \ \text{(d=3)}$$

GLP Stabilizers

Some functions are left unaffected by certain operators. These operators are said to stabilize the function.

$$P_{24} \left|x + x^2 + x^3\right\rangle \rightarrow \left|x + x^2 + x^3\right\rangle \text{ (d=5)}$$

Qs cannot form stabilizers on their own, but can if multiplied with Ps.

$$Q_2 * P_{012} \otimes Q_1^2 * P_{021} \left|xy^2 + x^2y\right\rangle \rightarrow \left|xy^2 + x^2y\right\rangle \text{ (d=3)}$$

Known stabilizers can then be used to create families of stabilized states.

$$P_{01} \left|x + 2x^2\right\rangle = \left|x + 2x^2\right\rangle \text{ (d=3) ,}$$
$$P_{02} \left|x + x^2\right\rangle = \left|x + x^2\right\rangle \text{ (d=3) ,}$$
$$P_{12} \left|x^2\right\rangle = \left|x^2\right\rangle \text{ (d=3)}$$

$$\mathrm{P}_{01} \otimes P_{02} \otimes P_{12} \left|(x + 2x^2)(y + y^2)(z^2)\right\rangle = \left|(x + 2x^2)(y + y^2)(z^2)\right\rangle$$

GLP Stabilizers

Some functions are left unaffected by certain operators. These operators are said to stabilize the function.

$$P_{24} \left|x + x^2 + x^3\right\rangle \rightarrow \left|x + x^2 + x^3\right\rangle \quad \text{(d=5)}$$

Qs cannot form stabilizers on their own, but can if multiplied with Ps.

$$Q_2 * P_{012} \otimes Q_1^2 * P_{021} \left|xy^2 + x^2y\right\rangle \rightarrow \left|xy^2 + x^2y\right\rangle \quad \text{(d=3)}$$

Known stabilizers can then be used to create families of stabilized states.

$$P_{01} \left|x + 2x^2\right\rangle = \left|x + 2x^2\right\rangle \quad \text{(d=3)} ,$$
$$P_{02} \left|x + x^2\right\rangle = \left|x + x^2\right\rangle \quad \text{(d=3)} ,$$
$$P_{12} \left|x^2\right\rangle = \left|x^2\right\rangle \quad \text{(d=3)}$$

$$\mathrm{P}_{01} \otimes P_{02} \otimes P_{12} \left|(x + 2x^2)(y + y^2)(z^2)\right\rangle = \left|(x + 2x^2)(y + y^2)(z^2)\right\rangle$$

Stabilizers are important in application and theory.

GLP Stabilizers (cont.)

One family of stabilizers found to work for any d or number of qudits is the 'Virginia Reel'

$$P_{VR} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ 1 & 0 & 0 & 0 & & 0 \end{bmatrix}$$

GLP Stabilizers (cont.)

One family of stabilizers found to work for any d or number of qudits is the 'Virginia Reel'

$$\mathrm{P}_{VR} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ 1 & 0 & 0 & 0 & & 0 \end{bmatrix}$$

And its transpose,

$$\mathrm{P}_{VR}^{T} = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & 0 & & 0 \end{bmatrix}$$

GLP Stabilizers (cont.)

One family of stabilizers found to work for any d or number of qudits is the 'Virginia Reel'

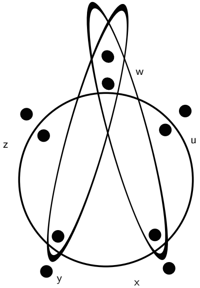
$$\mathrm{P}_{VR} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ 1 & 0 & 0 & 0 & & 0 \end{bmatrix}$$

And its transpose,

$$\mathrm{P}_{VR}^{T} = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & 0 & & 0 \end{bmatrix}$$

These will stabilize any function of the form f=(x+y)·(other variables)

GLP Stabilizers (cont.)



$$f(x, y, z, w, u) = (x + y)(zu + w^2)$$

The Virginia Reel is exponentiated from a hermitian matrix (M) and a certain constant $2\pi i/d$ ($P_{VR}=e^{(2\pi i/d)M}$). Its transpose/inverse is exponentiated from the same hermitian matrix and the negative of the same constant ($P_{VR}^T=e^{(-2\pi i/d)M}$). However, for any real t, $e^{itM} \otimes e^{-itM}$ is also a stabilizer for functions of the form f=(x+y)·(other variables). Thus, there is an infinite family of stabilizers for such functions.

Thank You!

Lebanon Valley College Mathematical Physics Research Group
http://quantum.lvc.edu/mathphys/