NAME

codequest − compute the quantum subsystem code implemented by the given operators

SYNOPSIS

man [−**fhw**] [**--version**] [**--**] [*filename*]

DESCRIPTION

codequest computes a quantum subsystem code that can be implemented by the given measurement operators. If you specify **-o** then the computed code is optimized, and the optimal qubit distances are output with the code. Be aware that the code optimization can be a very expensive calculation, which is why it is not performed by default.

In both the optimal and non-optimal case, note that the code that is output by **codequest** is not unique; in particular, a code with equivalent properties can be obtained by multiplying the logical and gauge qubit by one of the stabilizers.

codequest accepts one of two different formats, *dense* and *sparse* for specifying the measurement operators. See below for a description of these two formats.

If you specify a filename, then codequest reads the list of measurement operators from that file. Otherwise it reads them from the standard input.

OPTIONS

−**f input_format** or −−**format input_format**
: Specify the input format, which can be either *dense* or *sparse* as described below. The default is *dense*.

−**h** or −−**help**
: Print a help message and then exit.

−**o** or −−**optimize**
: Optimize the subsytem code (very expensive!) and output the optimal logical qubit distances. Note that this will result in a different set of logical operators compared to running **codequest** without this option.

−−**version**
: Display the version and then exit.

−− or −−**ignore_rest**
: Ignores the rest of the labeled arguments following this flag.

[filename]
: If *filename* is present, then **codequest** reads the list of operators from it. Otherwise, it reads them from standard input.

INPUT FORMATS

DENSE

In the dense format, n-qubit operators are specified in tensor product form as n-character strings consisting of the letters X, Y, and Z (case insensitive) to specify a Pauli operator and either spaces or periods to specify the identity operators. For example, the stabilizers of the 5-qubit Kitaev could be specified as follows:

```
XZZX.
.XZZX
X.XZZ
ZX.XZ
```

SPARSE

In the sparse format, each operator is specified by a line containing pairs of the form (L) P, where L is a label enclosed in parenthesis and P is the Pauli operator (i.e., X, Y, or Z, case insensitive) associated with that label. The label can be an arbitrary (case sensitive) string, and will be interpreted as the name of a qubit. The order of the pairs is not important, though no label may appear twice. Not all qubit names need appear in every operator. If the name of a qubit does not appear in an operator, then this is interpreted as meaning that there is an identity operator at that qubit's location in the tensor product. Whitespace will be trimmed around the Pauli operators and around the label, so that (A) and (A) are treated as equivalent labels but (AA) and (A A) are not.

For example, the following could be a specification of operators in a four-qubit system:

```
(A) X (B) Y
(C) Z (A) Z
(B) X (D) X
```

The first operator is a tensor product with identities at the locations of C and D, the X Pauli operator at the location of qubit A, and the Y Pauli operator at the location of qubit B. And so on.

If appropriate, the labels will be intepreted as coordinates. Specifically, labels which are arbitrary integers are interpreted as one-dimensional coordinates, and labels which are pairs of arbitrary integers separated by a comma are interpreted as as two-dimensional coordinates. The only effect that this has is on how the output is formatted; the operators of the subsystem code will be displayed using either one-dimensional or two-dimensional picture representations of the operators with the physical qubits drawn at locations corresponding to the coordinates specified for them in the input operators.

For example, if one inputs the following operators:

```
(0,0) X (1,0) X
(0,0) Z (0,1) Z
(0,1) X (1,1) X
(1,0) Z (1,1) Z
```

Then the X logical qubit of the code is displayed as

```
.X
.X
```

and the Z logical qubit of the code is displayed as

```
ZZ
..
```

There need not be a qubit located at every coordinate; those coordinates between coordinates where qubits are located will have spaces drawn, such as follows:

```
 . Y
. Y .
 . .
```

If any label cannot be intepreted as a one- or two-dimensional coordinate, then all labels will be treated as strings instead. In this case, **codequest** places an ordering on the labels and displays the output operators as one-dimensional strings. Note that, for example, there is nothing stopping one for specifying labels in the form of *three*-dimensional coordinates, but they will be interpreted by codequest as string labels for the

purpose of displaying the output.

AUTHOR

Gregory M. Crosswhite is the author of **codequest**. He can be contacted at *gcross@phys.washington.edu*.