CalcPauliTransferMatrix

```
SetDirectory @ NotebookDirectory[];
Import["../Link/QuESTlink.m"];
```

Doc

```
? CalcPauliTransferMatrix
```

Symbol

CalcPauliTransferMatrix[circuit] returns a PTM operator equivalent to the given circuit. CalcPauliTransferMatrix accepts optional argument AssertValidChannels.

```
? AssertValidChannels
```

Symbol

Optional argument to CalcCircuitMatrix, GetCircuitSuperoperator and CalcPauliTransferMatrix (default True), specifying whether to simplify their outputs by asserting that all channels therein are completely-positive and trace-preserving. For example, this asserts that the argument to a damping channel lies between 0 and 1 (inclusive).

```
? PTM
```

Symbol

PTM[matrix] is a Pauli-transfer matrix representation of an operator or channel. The subscript indices specify which Paulis of a Pauli string are operated upon.

Correctness

```
CalcPauliTransferMatrix @ Rz_5[x]
MatrixForm @ First @ %
```

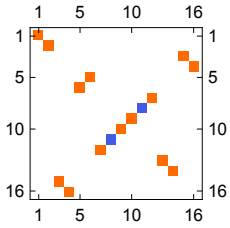
$$\text{PTM}_5\left[\text{SparseArray}\left[\text{Specified elements: 6, Dimensions: \{4, 4\}}\right]\right]$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \text{Cos}[x] & -\text{Sin}[x] & 0 \\ 0 & \text{Sin}[x] & \text{Cos}[x] & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
CalcPauliTransferMatrix @ C0[X1]
MatrixPlot @ First @ %
```

$PTM_{1,0}\left[\text{SparseArray}\left[\right]\right]$

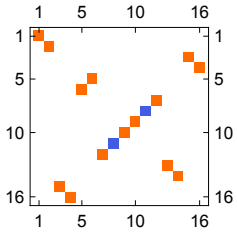
Specified elements: 16
Dimensions: {16, 16}




```
CalcPauliTransferMatrix @ C1[X0]
MatrixPlot @ First @ %
```

$PTM_{0,1}\left[\text{SparseArray}\left[\right]\right]$

Specified elements: 16
Dimensions: {16, 16}




```
CalcPauliTransferMatrix @ C5,7[R[x, X4 Y2]]
```

$PTM_{4,2,5,7}\left[\text{SparseArray}\left[\right]\right]$

Specified elements: 1056
Dimensions: {256, 256}

AssertValidChannels -> False

```
CalcPauliTransferMatrix[Rz0[x], AssertValidChannels → False]
MatrixForm @ First @ %
```

$PTM_0\left[\text{SparseArray}\left[\right]\right]$

Specified elements: 6
Dimensions: {4, 4}

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}\left(e^{-i x}+e^{i x}\right) & \frac{1}{2}\left(-i\, e^{-i x}+i\, e^{i x}\right) & 0 \\ 0 & \frac{1}{2}\left(i\, e^{-i x}-i\, e^{i x}\right) & \frac{1}{2}\left(e^{-i x}+e^{i x}\right) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
CalcPauliTransferMatrix[Damp0[x], AssertValidChannels → False]
Normal @ First @ %
```

$$\text{PTM}_0\left[\text{SparseArray}\left[\right]\right]$$

Specified elements: 8
Dimensions: {4, 4}

$$\left\{\left\{\frac{1}{2}\left(1+\sqrt{1-x}\ \text{Conjugate}\left[\sqrt{1-x}\right]+\sqrt{x}\ \text{Conjugate}\left[\sqrt{x}\right]\right),\right.\right.$$
$$\left.0,\ 0,\ \frac{1}{2}\left(1-\sqrt{1-x}\ \text{Conjugate}\left[\sqrt{1-x}\right]-\sqrt{x}\ \text{Conjugate}\left[\sqrt{x}\right]\right)\right\},$$
$$\left\{0,\ \frac{1}{2}\left(\sqrt{1-x}+\text{Conjugate}\left[\sqrt{1-x}\right]\right),\ \frac{1}{2}\left(i\sqrt{1-x}-i\ \text{Conjugate}\left[\sqrt{1-x}\right]\right),\ 0\right\},$$
$$\left\{0,\ \frac{1}{2}\left(-i\sqrt{1-x}+i\ \text{Conjugate}\left[\sqrt{1-x}\right]\right),\ \frac{1}{2}\left(\sqrt{1-x}+\text{Conjugate}\left[\sqrt{1-x}\right]\right),\ 0\right\},$$
$$\left\{\frac{1}{2}\left(1-\sqrt{1-x}\ \text{Conjugate}\left[\sqrt{1-x}\right]+\sqrt{x}\ \text{Conjugate}\left[\sqrt{x}\right]\right),\right.$$
$$\left.\left.0,\ 0,\ \frac{1}{2}\left(1+\sqrt{1-x}\ \text{Conjugate}\left[\sqrt{1-x}\right]-\sqrt{x}\ \text{Conjugate}\left[\sqrt{x}\right]\right)\right\}\right\}$$

Errors

```
CalcPauliTransferMatrix[]
```

CalcPauliTransferMatrix : Invalid arguments. See ?CalcPauliTransferMatrix

```
$Failed
```

```
CalcPauliTransferMatrix[X0, BadOption → Yes]
```

OptionValue : Unknown option BadOption for CalcPauliTransferMatrix.

```
$Failed
```

```
CalcPauliTransferMatrix[Blob0]
```

CalcPauliTransferMatrix : Circuit contained an unrecognised or unsupported gate: Blob_0

```
$Failed
```

```
CalcPauliTransferMatrix @ U0[x]
```

CalcPauliTransferMatrix : Circuit contained an unrecognised or unsupported gate: $U_0[x]$

```
$Failed
```

```
CalcPauliTransferMatrix[{G[x], Fac[2]}]
```

CalcPauliTransferMatrix : Circuit must explicitly target at least one qubit.

```
$Failed
```

```
CalcPauliTransferMatrix[Rx0[]]
```

CalcPauliTransferMatrix : Invalid arguments. See ?CalcPauliTransferMatrix

```
$Failed
```

```
CalcPauliTransferMatrix[{}]
```

CalcPauliTransferMatrix : Circuit must explicitly target at least one qubit.

```
$Failed
```