

GetCircuitConjugated

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

Doc

? GetCircuitConjugated

Symbol

GetCircuitConjugated[circuit] returns a circuit describing the complex conjugate operation of the given circuit. This is not the conjugate-transpose; instead, each operator is replaced with one or more operators described by Z-basis matrices equal to the complex conjugate of the original operator's matrix.

Accepts optional argument AssertValidChannels->False which relaxes the assumption that the circuit's operators are completely-positive and trace-preserving (CPTP). This permits canonical operator parameters (such as rotation strengths and channel probabilities) to be arbitrary complex values.

See ?AssertValidChannels.

Correctness

```
test[circ_, assumps_:{}, opts___] := Module[  
  {out, matr1, matr2, error},  
  matr1 = Conjugate @ CalcCircuitMatrix[circ, opts];  
  out = GetCircuitConjugated[circ, opts];  
  matr2 = CalcCircuitMatrix[out, opts];  
  error = matr1 - matr2 // N // Abs // Chop // Max;  
  error = FullSimplify[error, assumps];  
  Echo[out, "output: "];  
  Echo[error, "error: "];  
  If[error != 0 && error != 0., Style["ERRONEOUS CONJUGATION!", Red]]
```

Gates

Self-conjugate

```

test @  $H_0$ 
test @  $\text{Id}_3$ 
test @  $X_0$ 
test @  $Z_1$ 
test @  $\text{SWAP}_{0,2}$ 

» output: {  $H_0$  }
» error: 0
» output: {  $\text{Id}_3$  }
» error: 0
» output: {  $X_0$  }
» error: 0
» output: {  $Z_1$  }
» error: 0
» output: {  $\text{SWAP}_{0,2}$  }
» error: 0

test @  $C_1 @ H_0$ 
test @  $C_{0,2} @ \text{Id}_3$ 
test @  $C_{1,3} @ X_0$ 
test @  $C_0 @ Z_1$ 
test @  $C_1 @ \text{SWAP}_{0,2}$ 

» output: {  $C_1 [H_0]$  }
» error: 0
» output: {  $C_{0,2} [\text{Id}_3]$  }
» error: 0
» output: {  $C_{1,3} [X_0]$  }
» error: 0
» output: {  $C_0 [Z_1]$  }
» error: 0
» output: {  $C_1 [\text{SWAP}_{0,2}]$  }
» error: 0

```

Un-parameterised

```

test @  $S_0$ 
test @  $T_3$ 

```

```

» output:  $\left\{ \text{Ph}_0 \left[ -\frac{\pi}{2} \right] \right\}$ 
» error: 0
» output:  $\left\{ \text{Ph}_3 \left[ -\frac{\pi}{4} \right] \right\}$ 
» error: 0

test @ C1@ S0
test @ C0,1@ T3
» output:  $\left\{ C_1 \left[ \text{Ph}_0 \left[ -\frac{\pi}{2} \right] \right] \right\}$ 
» error: 0
» output:  $\left\{ C_{0,1} \left[ \text{Ph}_3 \left[ -\frac{\pi}{4} \right] \right] \right\}$ 
» error: 0

test @ Y3
test @ C2[Y1]
test @ C0,2[Y1]
» output:  $\{ U_3 [ \{ \{ 0, i \}, \{ -i, 0 \} \} ] \}$ 
» error: 0
» output:  $\{ C_2 [ U_1 [ \{ \{ 0, i \}, \{ -i, 0 \} \} ] ] \}$ 
» error: 0
» output:  $\{ C_{0,2} [ U_1 [ \{ \{ 0, i \}, \{ -i, 0 \} \} ] ] \}$ 
» error: 0

```

Parameterised (Real)

```

test[G[x], x ∈ Reals]
test[Fac[x]]
» output: {G[-x]}
» error: 0.
» output: {Fac[Conjugate[x]]}
» error: 0

GetCircuitConjugated[C0,1@G[x]]
{C0,1[G[-x]]}

test[C1,2@Fac[x]]
» output: {C1,2[Fac[Conjugate[x]]]}
» error: 0

test[Ph0[x], x ∈ Reals]
test[Ph0,2,1[x], x ∈ Reals]
test[C3,1@Ph0,2[x], x ∈ Reals]

```

```

» output: {Ph0[-x]}
» error: 0.
» output: {Ph0,2,1[-x]}
» error: 0.
» output: {C3,1[Ph0,2[-x]]}
» error: 0.

test[Rx0[x], x ∈ Reals]
test[Ry0[x], x ∈ Reals]
test[Rz0[x], x ∈ Reals]
» output: {Rx0[-x]}
» error: 0.
» output: {Ry0[x]}
» error: 0.
» output: {Rz0[-x]}
» error: 0.

test[Rx0,2[x], x ∈ Reals]
test[Ry1,3[x], x ∈ Reals]
test[Rz0,2,4[x], x ∈ Reals]
» output: {Rx0,2[-x]}
» error: 0.
» output: {Ry1,3[-x]}
» error: 0.
» output: {Rz0,2,4[-x]}
» error: 0.

test[C1@Rx0[x], x ∈ Reals]
test[C2@Ry0[x], x ∈ Reals]
test[C1,2@Rz0[x], x ∈ Reals]
test[C1@Rx0,2[x], x ∈ Reals]
test[C2,0@Ry1,3[x], x ∈ Reals]
test[C1@Rz0,2,4[x], x ∈ Reals]

```

```

» output: {C1[Rx0[-x]]}
» error: 0.
» output: {C2[Ry0[x]]}
» error: 0.
» output: {C1,2[Rz0[-x]]}
» error: 0.
» output: {C1[Rx0,2[-x]]}
» error: 0.
» output: {C2,0[Ry1,3[-x]]}
» error: 0.
» output: {C1[Rz0,2,4[-x]]}
» error: 0.

test[R[x, X0], x ∈ Reals]
test[R[x, Y1], x ∈ Reals]
test[R[x, Z2], x ∈ Reals]
test[C1@R[x, X0], x ∈ Reals]
test[C2,3@R[x, Y1], x ∈ Reals]
test[C0,1,3@R[x, Z2], x ∈ Reals]

» output: {R[-x, X0]}
» error: 0.
» output: {R[x, Y1]}
» error: 0.
» output: {R[-x, Z2]}
» error: 0.
» output: {C1[R[-x, X0]]}
» error: 0.
» output: {C2,3[R[x, Y1]]}
» error: 0.
» output: {C0,1,3[R[-x, Z2]]}
» error: 0.

test[R[x, X0 X1 Z2], x ∈ Reals]
test[C2,3@R[x, X0 X1 Z4], x ∈ Reals]

» output: {R[-x, X0 X1 Z2]}
» error: 0.
» output: {C2,3[R[-x, X0 X1 Z4]]}
» error: 0.

test[R[x, Y0 Y1], x ∈ Reals]
test[R[x, Y0 Y1 Y3], x ∈ Reals]

```

» output: $\{R[-x, Y_0 Y_1]\}$

» error: 0.

» output: $\{R[x, Y_0 Y_1 Y_3]\}$

» error: 0.

test[$R[x, Y_0 Z_1 Y_3 Z_4]$, $x \in \text{Reals}$]

test[$C_2 @ R[x, Y_0 Z_1 Y_3 Z_4]$, $x \in \text{Reals}$]

» output: $\{R[-x, Y_0 Y_3 Z_1 Z_4]\}$

» error: 0.

» output: $\{C_2[R[-x, Y_0 Y_3 Z_1 Z_4]]\}$

» error: 0.

Matrices

test @ $U_0 @ \{\{a, b\}, \{c, d\}\}$

test @ $C_{1,2} @ U_0 @ \{\{a, b\}, \{c, d\}\}$

» output: $\{U_0[\{\{\text{Conjugate}[a], \text{Conjugate}[b]\}, \{\text{Conjugate}[c], \text{Conjugate}[d]\}\}]\}$

» error: 0

» output: $\{C_{1,2}[U_0[\{\{\text{Conjugate}[a], \text{Conjugate}[b]\}, \{\text{Conjugate}[c], \text{Conjugate}[d]\}\}]\}$

» error: 0

test @ $U_{0,1} @ \text{RandomVariate} @ \text{CircularUnitaryMatrixDistribution} @ 4$

» output: $\{U_{0,1}[\{\{0.372528 - 0.566851 i,$
 $0.0809632 - 0.427203 i, 0.322425 - 0.137389 i, -0.47746 - 0.00656809 i\},$
 $\{0.163431 + 0.0864828 i, -0.258165 + 0.0657788 i, 0.655039 - 0.460613 i,$
 $0.498109 - 0.0740396 i\}, \{-0.012772 - 0.217894 i, 0.72287 + 0.405375 i,$
 $0.117806 - 0.160343 i, 0.12799 + 0.457733 i\}, \{0.069987 + 0.673182 i,$
 $0.0849192 - 0.214207 i, -0.0053987 - 0.443068 i, -0.419402 + 0.341461 i\}\}\}$

» error: 0

test @ $\text{UNonNorm}_0 @ \text{RandomComplex}[\{-1 - i, 1 + i\}, \{2, 2\}]$

» output: $\{\text{UNonNorm}_0[\{\{-0.883097 - 0.869113 i, 0.257308 + 0.591233 i\},$
 $\{-0.305792 - 0.20885 i, 0.718841 + 0.380704 i\}\}\}$

» error: 0

test @ $\text{UNonNorm}_{0,1,2} @ \text{RandomComplex}[\{-1 - i, 1 + i\}, \{8, 8\}]$

» output: {UNonNorm_{0,1,2}[{{0.9882 + 0.880376 i, 0.157184 - 0.0405589 i,
-0.191796 - 0.276161 i, 0.0291038 - 0.461392 i, -0.994763 - 0.961721 i,
0.712357 + 0.351834 i, -0.338665 + 0.200641 i, -0.184606 + 0.39828 i},
{0.884389 + 0.711152 i, -0.455944 + 0.163629 i, 0.655264 + 0.722777 i,
0.700113 - 0.121047 i, 0.464735 - 0.273893 i, 0.422873 + 0.437635 i,
-0.0401868 - 0.516622 i, 0.378875 - 0.343279 i},
{-0.0576798 + 0.772925 i, 0.367361 - 0.947076 i, 0.981778 - 0.866671 i,
-0.0587411 - 0.0257073 i, 0.581773 + 0.530006 i,
0.372268 + 0.969882 i, 0.231179 + 0.977732 i, 0.597868 - 0.643527 i},
{-0.888734 - 0.596695 i, -0.307319 - 0.316016 i, 0.690254 - 0.163767 i,
0.350565 - 0.229672 i, 0.630041 - 0.478496 i, -0.078318 - 0.306743 i,
-0.607411 - 0.363953 i, 0.281212 + 0.774483 i},
{0.0586227 + 0.0304034 i, 0.778395 - 0.0395657 i, 0.0136174 - 0.348595 i,
-0.5713 - 0.942165 i, 0.0498884 - 0.41103 i, 0.623614 + 0.200821 i,
-0.518105 - 0.418063 i, 0.266606 + 0.71744 i},
{-0.162868 - 0.117616 i, 0.62186 + 0.910843 i, -0.586799 - 0.141926 i,
-0.183634 + 0.700061 i, -0.932599 + 0.811292 i,
0.246315 + 0.393509 i, 0.979046 - 0.737228 i, 0.209746 + 0.857651 i},
{0.762315 + 0.636204 i, 0.207592 + 0.17342 i, 0.969639 - 0.809773 i,
0.294947 - 0.780985 i, -0.0956733 - 0.728371 i, 0.386882 - 0.608546 i,
-0.770708 + 0.373223 i, 0.424447 - 0.0423626 i},
{0.395386 - 0.484563 i, -0.360512 + 0.350551 i, 0.251552 - 0.0711501 i,
-0.414203 + 0.249229 i, 0.165518 - 0.7877 i, -0.200341 + 0.319659 i,
0.887626 - 0.0818245 i, 0.204408 + 0.688087 i}}}]}

» error: 0

test @ Matr_{0,2} @ RandomComplex[{-1 - i, 1 + i}, {2, 2}]

» output: {Matr_{0,2}[{{-0.908255 - 0.458092 i, -0.128121 + 0.000029823 i},
{-0.406197 - 0.083583 i, -0.480934 - 0.93448 i}}}]}

» error: 0

Channels

test[Deph₀[x], x ∈ Reals]
test[Deph_{0,1}[x], x ∈ Reals]

» output: {Deph₀[x]}

» error: 0.

» output: {Deph_{0,1}[x]}

» error: 0.

test[Depol₀[x], x ∈ Reals]
test[Depol_{0,1}[x], x ∈ Reals]

» output: {Depol₀[x]}

» error: 0.

» output: {Depol_{0,1}[x]}

» error: 0.

```

test[Damp2[x], 0 ≤ x ≤ 1]
» output: {Damp2[x]}
» error: 0.

m = Table[RandomComplex[{-1 - i, 1 + i}, {2, 2}], 5];
test[KrausNonTP0[m]]
» output:
{KrausNonTP0[{{{-0.0958492 - 0.60373 i, -0.922535 + 0.135156 i}, {0.808911 + 0.601336 i,
-0.452922 - 0.869196 i}}, {{0.0667436 + 0.932678 i, 0.155401 - 0.129154 i},
{0.628402 + 0.292134 i, 0.439744 - 0.195828 i}}, {{0.14143 - 0.781209 i,
-0.844258 - 0.785011 i}, {-0.209284 + 0.221074 i, -0.486959 + 0.0683674 i}},
{{-0.959909 - 0.0341837 i, 0.857763 - 0.196977 i}, {0.23409 + 0.0688866 i,
0.626097 - 0.115888 i}}, {{0.78889 + 0.255592 i, 0.78734 + 0.755442 i},
{-0.220219 + 0.0595561 i, -0.540675 + 0.0740567 i}}]}]}
» error: 0

```

Circuits

```

circ = Circuit[C0[Ry2[x]] SWAP1,2 Deph0,1[y] Damp2[z] C2[Y1] R[x, X0 Y1 Z2];
test[circ, {x ∈ Reals, 0 ≤ y ≤ 1, 0 ≤ z ≤ 1}]
» output: {C0[Ry2[x]], SWAP1,2, Deph0,1[y], Damp2[z], C2[U1[{{0, i}, {-i, 0}}]], R[x, X0 Y1 Z2]}
» error: 0.

```

AssertValidChannels -> False

```

GetCircuitConjugated[Damp0[x]]
GetCircuitConjugated[Damp0[x], AssertValidChannels → False]
{Damp0[x]}
{Damp0[Conjugate[x]]}

```

Operators

```

test[G[x], {}, AssertValidChannels → False]
test[Fac[x], {}, AssertValidChannels → False]
» output: {G[-Conjugate[x]]}
» error: 0
» output: {Fac[Conjugate[x]]}
» error: 0

GetCircuitConjugated[C0,1@G[x], AssertValidChannels → False]
{C0,1[G[-Conjugate[x]]]}

test[C1,2@Fac[x], {}, AssertValidChannels → False]

```



```

» output: {C1,2[Fac[Conjugate[x]]]}
» error: 0

test[Ph0[x], {}, AssertValidChannels → False]
test[Ph0,2,1[x], {}, AssertValidChannels → False]
test[C3,1@Ph0,2[x], {}, AssertValidChannels → False]
» output: {Ph0[-Conjugate[x]]}
» error: 0
» output: {Ph0,2,1[-Conjugate[x]]}
» error: 0
» output: {C3,1[Ph0,2[-Conjugate[x]]]}
» error: 0

test[Rx0[x], {}, AssertValidChannels → False]
test[Ry0[x], {}, AssertValidChannels → False]
test[Rz0[x], {}, AssertValidChannels → False]
» output: {Rx0[-Conjugate[x]]}
» error: 0
» output: {Ry0[Conjugate[x]]}
» error: 0
» output: {Rz0[-Conjugate[x]]}
» error: 0

test[Rx0,2[x], {}, AssertValidChannels → False]
test[Ry1,3[x], {}, AssertValidChannels → False]
test[Rz0,2,4[x], {}, AssertValidChannels → False]
» output: {Rx0,2[-Conjugate[x]]}
» error: 0
» output: {Ry1,3[-Conjugate[x]]}
» error: 0
» output: {Rz0,2,4[-Conjugate[x]]}
» error: 0

test[C1@Rx0[x], {}, AssertValidChannels → False]
test[C2@Ry0[x], {}, AssertValidChannels → False]
test[C1,2@Rz0[x], {}, AssertValidChannels → False]
test[C1@Rx0,2[x], {}, AssertValidChannels → False]
test[C2,0@Ry1,3[x], {}, AssertValidChannels → False]
test[C1@Rz0,2,4[x], {}, AssertValidChannels → False]

```

```

» output: {C1[Rx0[-Conjugate[x]]]}
» error: 0
» output: {C2[Ry0[Conjugate[x]]]}
» error: 0
» output: {C1,2[Rz0[-Conjugate[x]]]}
» error: 0
» output: {C1[Rx0,2[-Conjugate[x]]]}
» error: 0
» output: {C2,0[Ry1,3[-Conjugate[x]]]}
» error: 0
» output: {C1[Rz0,2,4[-Conjugate[x]]]}
» error: 0

test[R[x, X0], {}, AssertValidChannels → False]
test[R[x, Y1], {}, AssertValidChannels → False]
test[R[x, Z2], {}, AssertValidChannels → False]
test[C1@R[x, X0], {}, AssertValidChannels → False]
test[C2,3@R[x, Y1], {}, AssertValidChannels → False]
test[C0,1,3@R[x, Z2], {}, AssertValidChannels → False]
» output: {R[-Conjugate[x], X0]}
» error: 0
» output: {R[Conjugate[x], Y1]}
» error: 0
» output: {R[-Conjugate[x], Z2]}
» error: 0
» output: {C1[R[-Conjugate[x], X0]]}
» error: 0
» output: {C2,3[R[Conjugate[x], Y1]]}
» error: 0
» output: {C0,1,3[R[-Conjugate[x], Z2]]}
» error: 0

test[R[x, X0 X1 Z2], {}, AssertValidChannels → False]
test[C2,3@R[x, X0 X1 Z4], {}, AssertValidChannels → False]
» output: {R[-Conjugate[x], X0 X1 Z2]}
» error: 0
» output: {C2,3[R[-Conjugate[x], X0 X1 Z4]]}
» error: 0

test[R[x, Y0 Y1], {}, AssertValidChannels → False]
test[R[x, Y0 Y1 Y3], {}, AssertValidChannels → False]

```

```

» output: {R[-Conjugate[x], Y0 Y1]}
» error: 0
» output: {R[Conjugate[x], Y0 Y1 Y3]}
» error: 0

test[R[x, Y0 Z1 Y3 Z4], {}, AssertValidChannels → False]
test[C2@R[x, Y0 Z1 Y3 Z4], {}, AssertValidChannels → False]
» output: {R[-Conjugate[x], Y0 Y3 Z1 Z4]}
» error: 0
» output: {C2[R[-Conjugate[x], Y0 Y3 Z1 Z4]]}
» error: 0

```

Circuit

```

circ = Circuit[C0[Ry2[x]] SWAP1,2 Deph0,1[y] Damp2[z] C2[Y1] R[x, X0 Y1 Z2]];
conj = GetCircuitConjugated[circ, AssertValidChannels → False];
subs = {x → RandomComplex[], y → RandomComplex[], z → RandomComplex[]};

Conjugate @ CalcCircuitMatrix @ circ - CalcCircuitMatrix @ conj;
% /. subs // Abs // Max
0.

```

Errors

GetCircuitConjugated[M₀]

GetCircuitConjugated: Cannot obtain conjugate of unrecognised or unsupported operator: M₀

\$Failed

GetCircuitConjugated[GlampSchmwamp₀]

GetCircuitConjugated: Cannot obtain conjugate of unrecognised or unsupported operator: GlampSchmwamp₀

\$Failed

GetCircuitConjugated[X₂, badoption → False]

OptionValue: Unknown option badoption for GetCircuitConjugated.

\$Failed

GetCircuitConjugated[eh]

GetCircuitConjugated: Invalid arguments. See ?GetCircuitConjugated

\$Failed