Quick Guide to

Quantavo: Maple Toolbox for Linear Quantum Optics

Alvaro Feito Boirac[1,2]

[1]Institute for Mathematical Sciences, Imperial College London,

53 Prince's Gate, South Kensington, London, SW7 2PG, UK

[2]QOLS, The Blackett Lab. Imperial College London,

Prince Consort Road, London SW7 2BW, UK

Objects	Appearance	Declaration
vec	$\psi = \begin{bmatrix} 1 & [0,0,0] \\ \lambda & [1,1,0] \\ \lambda^2 & [2,2,0] \\ \lambda^3 & [3,3,0] \end{bmatrix}$	vec:=`CoherentState`(K,d,α) vec:=`SqueezedVac`(K,d,λ) vac:=`Vac`(K); vec1:=`TensorVac`(vec,m) vec1:=`TensorProduct`(V,[1,2],W,[3,4]) vec:=Trim(V)
mat	$\rho = \begin{bmatrix} 0 & [0,0,0] & [1,1,0] \\ [0,0,0] & 1 & \bar{\lambda} \\ [1,1,0] & \lambda & \lambda\bar{\lambda} \end{bmatrix}$	mat:=`vec2mat`(vec) mat:=`matcol2mat`(matcol) or direct declaration of the Matrix
matcol	$\rho_{matcol} = \begin{bmatrix} 1 & [0,0,0] & [0,0,0] \\ \bar{\lambda} & [0,0,0] & [1,1,0] \\ \lambda & [1,1,0] & [0,0,0] \\ \lambda\bar{\lambda} & [1,1,0] & [1,1,0] \\ \lambda^2\bar{\lambda}^2 & [2,2,0] & [2,2,0] \end{bmatrix}$	matcol:=`vec2matcol`(vec) matcol:=`mat2matcol`(mat) or direct declaration of the 3 column Matrix

$$\text{ready made states are:}\quad \begin{array}{lcl} \texttt{CoherentState}(K,d,\alpha) & \sim & \displaystyle\sum_{n=0}^{d-1}\frac{\alpha^n}{\sqrt{n!}}|n\rangle^{\otimes K} \\ \texttt{SqueezedVac}(K,d,\lambda) & \sim & \displaystyle\sum_{n=0}^{d-1}\lambda^n|n\rangle^{\otimes K} \\ \texttt{TensorVac}(\mathbf{vec},m) & : & |\phi\rangle \to |\phi\rangle\otimes|0\rangle^{\otimes m} \\ \texttt{IdentityState(d,K)} & \sim & \mathbb{I}_{d^K\times d^K} \end{array}$$

Common Procedures:

Linear Optics:	Measurements
`BS`(**vec/matcol**, i, j) `myBS`(**vec/matcol**, i, j, t, r) `PS`(**vec/matcol**, i, ϕ) `BuildUnitary`([*List of Lists*]) `UnitaryEvolution`(U, **vec/matcol**)	`Project`(**vec/matcol**, [*List*], **vec/matcol**) `Probability`(**vec/matcol**, [*List*]**vec/matcol**) `Traceout`(**mat/matcol**, i) `POVMresult`(**matcol**, [*List*], **vec/matcol**) `APD` ($\{0, 1\}, r, d + 1$)

Display	Algebraic Operations
`Dstate`(**vec/matcol/mat**) `PlotState`(**vec/matcol/mat**, width, height)	`StateApprox`(**vec/matcol**, *list*, N) `StateMultiply`(**vec/matcol**, **vec/matcol**) `StateComplexConjugate`(**mat/matcol**) `StateNorm`(**vec**) `StateNormalize`(**vec/mat/matcol**) `StateTrace`(**mat/matcol**) `Traceout`(**mat/matcol**, i) `TensorProduct`(**vec/matcol**, *list*, **vec/matcol**, *list*) `StatePartialTranspose`(**mat/matcol**, i) `deltaK`(i, j)

State Properties:	Entanglement & Energy
`IsHermitian`(**matcol/mat**) `IsNormalized`(**vec/matcol/mat**) `StateNorm`(**vec**) `StateSort`(**vec/matcol**) `FindKnd`(**vec/matcol/mat**)	`Negativity`(**vec/matcol/mat**) `LogNegativity`(**vecmatcol/mat**) `Entropy`(**vec/matcol**) `Energy`(**vec/matcol**)

For further details see dictionary of procedures in the Quantavo_manual.pdf