QI Package for Mathematica 7.0 (version 0.4.36)

Jarosław Adam Miszczak Piotr Gawron Zbigniew Puchała
The Institute of Theoretical and Applied Informatics
Polish Academy of Sciences,
Bałtycka 5, 44-100 Gliwice, Poland

February 4, 2012

QI is a package of functions for Mathematica computer algebra system, which implements number of functions used in the analysis of quantum states and quantum operations. In contrast to many available packages for symbolic and numerical simulation of quantum computation presented package is focused on geometrical aspects of quantum information theory.

- **ApplayChannel**[f,ρ] - apply channel f, given as a pure function, to the input state ρ. See also: **ApplyUnitary**, **ApplyKraus**.
- **ApplyKraus**[ck,ρ] - apply channel ck, given as a list of Kraus operators, to the input state ρ. See also: **ApplyUnitary**, **ApplyChannel**.
- **BaseMatrices**[n] returns a list with the canonical basis in $n$xn-dimensional Hilbert-Schmidt space of matrices. See also: BaseVectors.
- **BaseVectors**[n] returns a list with the canonical basis in n-dimensional Hilbert space. See also: BaseMatrices.
- **BlochToState**[v] - returns a matrix of appropriate dimension from Bloch vector, i.e. coefficients treated as coefficients from expansion on normalized generalized Pauli matrices. See also: **GeneralizedPauliMatrices**.
- **ChannelToMatrix**[E,d] returns matrix representation of a channel E acting on d-dimensional state space. First argument should be a pure function E such that $E[\rho]$ transforms input state according to the channel definition.
- *cnot* - Controlled not matrix for two qubits.
- **ComplexToPoint**[z] returns a real and an imaginary parts of a complex number z as a pair of real numbers.
- **DynamicalMatrix** returns a dynamical matrix of quantum channel
  DynamicalMatrix[ch] - operates on a quantum channel given as a list of Kraus operators.
  DynamicalMatrix[fun,dim] - operates on a a function fun acting on dim-dimensional space.
  See also: **Superoperator**, **ChannelToMatrix**.
- **Fidelity**[ρ_1,ρ_2] returns the quantum fidelity between states ρ_1 and ρ_2 calculated using a simplified formula as $(\sum\lambda_i)^2$, where λ_i are the eigenvalues of $\rho_1\rho_2$.
- **GateFidelity**[U, V] is equivalent to $1/d\mathrm{tr}|\mathrm{UV}^\dagger|$.

- **GeneralizedPauliMatrices**[n] returns list of generalized Pauli matrices for $SU(n)$. For $n = 2$ these are just Pauli matrices and for $n = 3$ - Gell-Mann matrices. Note that identity matrix is not included in the list.
- **GinibreMatrix**[m,n] returns complex matrix of dimension m×n with the standard normal distribution of real and imaginary parts.
- *id* - Identity matrix for one qubit. See also: IdentityMatrix.
- **Id**[n] returns an identity matrix of dimension n. This is equivalent to IdentityMatrix[n].
- **Jamiolkowski**[K] gives the image of the Jamiolkowski isomorphism for the channel given as the list of Karus operators K.
  Jamiolkowski[fun,dim] gives the image of the Jamiolkowski isomorphism for the channel given as a function fun action on dim-dimensional space. See also: **Superoperator**, **ChannelToMatrix**, **DynamicalMatrix**.
- **Lambda1**[i, j, n] generalized Pauli matrix. For example **Lambda1**[$1, 2, 2$] is equal to Pauli σx. See also: **GeneralizedPauliMatrices**.
- **Lambda2**[i, j, n] generalized Pauli matrix. For example **Lambda2**[$1, 2, 2$] is equal to Pauli σy. See also: **GeneralizedPauliMatrices**.
- **Lambda3**[i, n] generalized Pauli matrix. For example **Lambda3**[$2, 2$] is equal to Pauli σz. See also: **GeneralizedPauliMatrices**.
- **MatrixAbs**[A] returns absolute value for matrix A defined as **MatrixSqrt**[$A.A^\dagger$]. See also: **MatrixSqrt**.
- **MatrixIm**[A] returns an antyhermitian part of the matrix A i.e. $1/2(A - A^\dagger)$.
- **MatrixRe**[A] returns a hermitian part of the matrix A i.e. $1/2(A + A^\dagger)$.
- **MatrixSqrt**[A] returns square root for the matrix A.
- **PartialTrace**[ρ,sys] - Returns the partial trace of an operator ρ acting on a bipartite (d×d)-dimensional system, assuming that matrix ρ is $(d^2 \times d^2)$ dimensional. In this case the system specification can be given as a integer 1 or 2, by the list of integers consisting only 1 or 2 or by the empty list.
  PartialTrace[ρ,$dims$,sys] - Returns the partial trace of an operator ρ acting on a composite system with subsystem dimensions given in the list $dims$. List sys specifies systems to be discarded.
- **PartialTranspose**[ρ,dim,sys] - Returns the partial transpose, according to systems sys, of density matrix ρ composed of subsystems of dimensions dim.
- **ProductSuperoperator**[Ψ,Φ] computes a product superoperator of superoperatos Ψ and Φ.
- **Proj**[v] returns projector of the vector v.
- **QubitKet**[α,β] parametrization of the pure state (as a state vector) for one qubit as (Cos[α] Exp[iβ], Sin[α]). This is equivalent to **StateVector**[α,β]. See also: **QubitPureState**, **StateVector**.
- **QubitPureState**[α,β] - a parametrization of the pure state as a density matrix for one qubit. This is just a alias for **Proj**[**QubitKet**[α,β]]. See also: **QubitKet**.
- **RandomDynamicalMatrix**[d,k] returns dynamical matrix of operation acting on d-dimensional states with k eigenvalues equal to 0. Thanks to Wojtek Bruzda. see Random Quantum Operations DOI[10.1016/j.physleta.2008.11.043].

- **RandomKet**[*d*] - for integer *d* returns a random ket vector in *d*-dimensional space. See: T. Radtke, S. Fritzsche, Comp. Phys. Comm., Vol. 179, No. 9, p. 647-664.
  RandomKet[*d*1,*d*2, *type*] - for integers *d*1, *d*2 returns a random ket vector in *d*1*d*2-dimensional space with distribution specified by *type*.
  The parameter *type* can be:
  "Sep" - in this case function returns separable random vectors,
  "MaxEnt" - in this case function returns maximally entangled random vectors,
  l list of positive numbers summing up to 1, the length of the list must be less or equal to the Min[*d*1,*d*2] - in this case function returns random vectors with fixed Schmidt numbers given by *l*.

- **RandomOrthogonal**[*d*] returns a random orthogonal matrix of size *d* using QR decomposition. See: F. Mezzadri, NOTICES of the AMS, Vol. 54 (2007), 592-604.

- **RandomSimplex**[*d*] generates a point on a *d*-dimensional simplex according to the uniform distibution.

- **RandomSpecialUnitary**[*d*] returns a random special unitary matrix of size *d*. See **RandomUnitary**.

- **RandomState**[*d*,*dist*] - random density matrix of dimension *d*. Argument *dist* can be "*HS*" (default value), "*Bures*" or an integer K.
  "*HS*" - gives uniform distribution with respect to the Hilbert-Schmidt measure.
  "*Bures*" - gives a random state distributed according to Bures measure.
  Integer K - gives a random state generated with respect to induced measure with an ancilla system od dimension K.

- **RandomUnitary**[*d*] returns a random unitary matrix of size *d* using QR decomposition. See: F. Mezzadri, NOTICES of the AMS, Vol. 54 (2007), 592-604.

- **Res**[*A*] is equivalent to **Vec**[Transpose[*A*]]. Reshaping mapsmatrix A into a vector row by row. Note, that this is different then thereshape operation in *Matlab* or *GNUOctave*.

- **Reshuffle**[ρ, *drows*, *dcols*] for a matrix of dimensions (drows[[1]]×drows[[2]])×(dcols[[1]]×dcols[[2]]) returns a reshuffled matrix with dimensions (drows[[1]]×dcols[[1]])×(drows[[2]]×dcols[[2]]).
  Parameters *drows*,*dcols* can be ommited for a square matrix of dimension n^2×n^2.

- **SchmidtDecomposition**[*x*,*dim*] - accepts a vector or a matrix as a first argument and returns apropriate Schmidt decomposition. The second argument is optional and specifies the dimensions of subsystems.
  If *x* is a vector
  SchmidtDecomposition[*x*] assumes that *x* is (n^2)-dimensional,
  SchmidtDecomposition[*x*,*n*,*m*] assumes that the vector is a *nxm*-dimensional.
  If *x* is a matrix, this function can be used in three different ways.
  SchmidtDecomposition[*x*] assumes that *x* is (n^2xn^2)-dimensional,
  SchmidtDecomposition[*x*,*n*,*m*] assume that the matrix is a *nmxnm* square matrix and
  SchmidtDecomposition[*x*,*r*1,*r*2,*c*1,*c*2]
  For example, for a matrix *mtx* of dimension *r*1*r*2*xc*1*c*2 one can obtain a Schmidt decomposition on *r*1*c*1(*x*)*r*2*c*2 system as
  sd = **SchmidtDecomposition**[*mtx*, *r*1, *r*2, *c*1, *c*2];
  and reconstruct the original matrix as
  mtx == Sum[*sd*[[*i*,1]]*KroneckerProduct[*sd*[[*i*,2]], *sd*[[*i*,3]]], *i*, Length[*sd*]];

- **SpecialUnitary**[*d*,*params*] returns the special unitary matrix of size *d* with Euler parameters given in *params*. *params* must be a list of length $d^2 - 1$.

- **SpecialUnitary2**[β,γ,δ] returns a parametrization of $SU(2)$. This is equivalent to **Unitary2**[0,β,γ,δ].

- **SquareMatrixQ**[*A*] returns True only if *A* is a square matrix, and gives False otherwise.

- **StateToBloch**$[A]$ - for a square matrix A returns a vector of coefficients obtained from expansion on normed generalized Pauli matrices. See also: **GeneralizedPauliMatrices**.
- **StateVector**$[\theta_1,...,\theta_n,\phi_{n+1},...,\phi_{2n}]$ returns pure n+1-dimensional pure state (ket vector) constructed form probability distribution parametrize by numbers $\theta_1,...,\theta_n$ and phases $\phi_1,...,\phi_n$. See also: **SymbolicVector**.
- **Subfidelity**$[\rho_1,\rho_2]$ returns subfidelity between states ρ_1 and ρ_2 See: J.A. Miszczak et al., Quantum Information & Computation, Vol.9 No.1&2 (2009).
- **Superfidelity**$[\rho_1,\rho_2]$ calculates superfidelity between ρ_1 and ρ_2 defined as $\mathrm{tr}[\rho_1\rho_2]+\sqrt{(1-\mathrm{tr}\,[\rho_1^2])\,(1-\mathrm{tr}\,[\rho_2^2])}$ See: J.A. Miszczak et al., Quantum Information & Computation, Vol.9 No.1&2 (2009).
- **Superoperator**$[kl]$ returns matrix representation of quantum channel given as a list of Kraus operators. **Superoperator**$[fun,dim]$ is just am alternative name for **ChannelToMatrix**$[fun,dim]$ and returns matrix representation of quantum channel, given as a pure function, acting on *dim*-dimensional space. See also: **ChannelToMatrix**.
- **SuperoperatorToKraus**$[m]$ returns Kraus operators for a given super operator m.
- sx - Pauli matrix sx.
- sy - Pauli matrix sy.
- **SymbolicBistochasticMatrix**$[sym,dim]$ produces symbolic bistochastic matrix size dim. See also: **SymbolicMatrix, SymbolicVector**.
- **SymbolicHermitianMatrix**$[sym,n]$ produces a n×n Hermitian matrix. See also: **SymbolicMatrix, SymbolicVector**.
- **SymbolicMatrix**$[a,m,n]$ returns m×n-matrix with elements $a[i,j], i=1,...,m, j=1,...,n$.If the third argument is ommited this function returns square m×m matrix. This functions can save you some keystrokes and, thanks to TeXForm function, its results can be easily incorporated in LaTeX documents.
- **SymbolicVector**$[a,n]$ returns a vector with n elements $a[i], i=1,...,n$.
- sz - Pauli matrix sz.
- **TPChannelQ**$[ck]$ performs some checks on Kraus operators ck. Use this if you want to check if they represent quantum channel.
- **TraceDistance**$[\rho_1,\rho_2]$ returns the trace distance between matrices ρ_1 and ρ_2, which is defined as $\frac{1}{2}\mathrm{tr}|\rho_1-\rho_2|$.
- **TraceNorm**$[A]=\sum\sigma_i$, where σ_i are the singular values of A. See also: **TraceDistance**.
- **Unitary2**$[\alpha,\beta,\gamma,\delta]$ returns a parametrization of $U(2)$.
- **Unitary2**$[\alpha,\beta,\gamma,\delta]$ returns the Euler parametrization of $U(2)$.
- **Unres**$[v,c]$ - de-reshaping of the vector into a matrix with c columns. If the second parameter is omitted, then it is assumed that v can be mapped into a square matrix. See also: **Unvec**, **Res**.
- **Unvec**$[v,c]$ - de-vectorization of the vector into the matrix with c columns. If the second parameter is omitted then it is assumed that v can be mapped into square matrix. See also: **Unres, Vec**.
- **Vec**$[A]$ - vectorization of the matrix A column by column. See also: **Res**.
- wh - Hadamard gate for one qubit.