

qpp
0.1

Generated by Doxygen 1.8.5

Sun Apr 13 2014 22:52:58

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Namespace Documentation	9
5.1	qpp Namespace Reference	9
5.1.1	Function Documentation	13
5.1.1.1	absm	13
5.1.1.2	adjoint	13
5.1.1.3	anticomm	14
5.1.1.4	channel	14
5.1.1.5	choi	15
5.1.1.6	choi2kraus	15
5.1.1.7	comm	16
5.1.1.8	conjugate	16
5.1.1.9	cosm	16
5.1.1.10	det	17
5.1.1.11	disp	17
5.1.1.12	disp	17
5.1.1.13	disp	17
5.1.1.14	disp	17
5.1.1.15	displn	17
5.1.1.16	displn	18
5.1.1.17	displn	18
5.1.1.18	displn	18

5.1.1.19	entanglement	19
5.1.1.20	evals	19
5.1.1.21	evects	20
5.1.1.22	expandout	20
5.1.1.23	expm	21
5.1.1.24	fun	21
5.1.1.25	funm	21
5.1.1.26	gconcurrency	22
5.1.1.27	grams	23
5.1.1.28	grams	23
5.1.1.29	hevals	24
5.1.1.30	hevects	24
5.1.1.31	kron	24
5.1.1.32	kronlist	25
5.1.1.33	kronpow	25
5.1.1.34	load	25
5.1.1.35	loadMATLABmatrix	25
5.1.1.36	loadMATLABmatrix	25
5.1.1.37	loadMATLABmatrix	25
5.1.1.38	logm	26
5.1.1.39	multiidx2n	26
5.1.1.40	mutualinfo	27
5.1.1.41	n2multiidx	27
5.1.1.42	norm	28
5.1.1.43	powm	28
5.1.1.44	proj	29
5.1.1.45	ptrace	29
5.1.1.46	ptrace2	30
5.1.1.47	ptranspose	30
5.1.1.48	rand	31
5.1.1.49	rand	31
5.1.1.50	rand	31
5.1.1.51	rand	31
5.1.1.52	randH	31
5.1.1.53	randket	31
5.1.1.54	randkraus	32
5.1.1.55	randn	32
5.1.1.56	randn	32
5.1.1.57	randn	32
5.1.1.58	randn	32

5.1.1.59	randrho	33
5.1.1.60	randU	33
5.1.1.61	randV	33
5.1.1.62	renyi	33
5.1.1.63	renyi_inf	34
5.1.1.64	reshape	34
5.1.1.65	save	34
5.1.1.66	saveMATLABmatrix	34
5.1.1.67	saveMATLABmatrix	35
5.1.1.68	saveMATLABmatrix	35
5.1.1.69	schmidtcoeff	35
5.1.1.70	schmidtprob	36
5.1.1.71	schmidtU	36
5.1.1.72	schmidtV	37
5.1.1.73	shannon	37
5.1.1.74	sinm	38
5.1.1.75	spectralpowm	38
5.1.1.76	sqrtm	38
5.1.1.77	sum	39
5.1.1.78	super	39
5.1.1.79	syspermute	40
5.1.1.80	trace	40
5.1.1.81	transpose	41
5.1.1.82	tsallis	41
5.1.2	Variable Documentation	41
5.1.2.1	gt	41
5.1.2.2	rdevs	41
5.2	qpp::ct Namespace Reference	41
5.2.1	Function Documentation	42
5.2.1.1	omega	42
5.2.2	Variable Documentation	42
5.2.2.1	chop	42
5.2.2.2	ee	42
5.2.2.3	eps	42
5.2.2.4	ii	42
5.2.2.5	pi	42
5.3	qpp::internal Namespace Reference	42
5.3.1	Function Documentation	43
5.3.1.1	_check_col_vector	43
5.3.1.2	_check_dims	43

5.3.1.3	_check_dims_match_cvect	43
5.3.1.4	_check_dims_match_mat	43
5.3.1.5	_check_dims_match_rvect	43
5.3.1.6	_check_eq_dims	43
5.3.1.7	_check_nonzero_size	43
5.3.1.8	_check_perm	43
5.3.1.9	_check_row_vector	43
5.3.1.10	_check_square_mat	43
5.3.1.11	_check_subsys	43
5.3.1.12	_check_vector	43
5.3.1.13	_multiidx2n	43
5.3.1.14	_n2multiidx	43
5.3.1.15	_ptranspose_worker	43
5.3.1.16	_syspermute_worker	44
5.4	qpp::types Namespace Reference	44
5.4.1	Typedef Documentation	44
5.4.1.1	bra	44
5.4.1.2	cmat	44
5.4.1.3	cplx	44
5.4.1.4	dmat	44
5.4.1.5	DynMat	44
5.4.1.6	fmat	44
5.4.1.7	imat	44
5.4.1.8	ket	44
6	Class Documentation	45
6.1	qpp::DiscreteDistribution Class Reference	45
6.1.1	Constructor & Destructor Documentation	45
6.1.1.1	DiscreteDistribution	45
6.1.1.2	DiscreteDistribution	45
6.1.1.3	DiscreteDistribution	45
6.1.2	Member Function Documentation	45
6.1.2.1	probabilities	45
6.1.2.2	sample	46
6.1.3	Member Data Documentation	46
6.1.3.1	_d	46
6.2	qpp::DiscreteDistributionAbsSquare Class Reference	46
6.2.1	Constructor & Destructor Documentation	46
6.2.1.1	DiscreteDistributionAbsSquare	47
6.2.1.2	DiscreteDistributionAbsSquare	47

6.2.1.3	DiscreteDistributionAbsSquare	47
6.2.1.4	DiscreteDistributionAbsSquare	48
6.2.2	Member Function Documentation	48
6.2.2.1	cplx2weights	48
6.2.2.2	probabilities	48
6.2.2.3	sample	48
6.2.3	Member Data Documentation	48
6.2.3.1	_d	48
6.3	qpp::Exception Class Reference	48
6.3.1	Member Enumeration Documentation	50
6.3.1.1	Type	50
6.3.2	Constructor & Destructor Documentation	51
6.3.2.1	Exception	51
6.3.2.2	Exception	51
6.3.2.3	~Exception	51
6.3.3	Member Function Documentation	51
6.3.3.1	_construct_exception_msg	51
6.3.3.2	what	51
6.3.4	Member Data Documentation	51
6.3.4.1	_custom	51
6.3.4.2	_msg	51
6.3.4.3	_type	51
6.3.4.4	_where	51
6.4	qpp::Gates Class Reference	52
6.4.1	Constructor & Destructor Documentation	53
6.4.1.1	Gates	53
6.4.1.2	Gates	53
6.4.1.3	~Gates	53
6.4.2	Member Function Documentation	53
6.4.2.1	CTRL	53
6.4.2.2	Fd	54
6.4.2.3	getInstance	54
6.4.2.4	Id	54
6.4.2.5	operator=	54
6.4.2.6	Rtheta	54
6.4.2.7	Xd	54
6.4.2.8	Zd	54
6.4.3	Member Data Documentation	55
6.4.3.1	b00	55
6.4.3.2	b01	55

6.4.3.3	b10	55
6.4.3.4	b11	55
6.4.3.5	CNOTab	55
6.4.3.6	CNOTba	55
6.4.3.7	CS	55
6.4.3.8	CZ	55
6.4.3.9	FRED	55
6.4.3.10	H	55
6.4.3.11	Id2	55
6.4.3.12	pb00	55
6.4.3.13	pb01	55
6.4.3.14	pb10	55
6.4.3.15	pb11	55
6.4.3.16	px0	55
6.4.3.17	px1	55
6.4.3.18	py0	55
6.4.3.19	py1	55
6.4.3.20	pz0	55
6.4.3.21	pz1	55
6.4.3.22	S	55
6.4.3.23	SWAP	55
6.4.3.24	T	55
6.4.3.25	TOF	55
6.4.3.26	X	55
6.4.3.27	x0	55
6.4.3.28	x1	56
6.4.3.29	Y	56
6.4.3.30	y0	56
6.4.3.31	y1	56
6.4.3.32	Z	56
6.4.3.33	z0	56
6.4.3.34	z1	56
6.5	qpp::NormalDistribution Class Reference	56
6.5.1	Constructor & Destructor Documentation	56
6.5.1.1	NormalDistribution	56
6.5.2	Member Function Documentation	56
6.5.2.1	sample	56
6.5.3	Member Data Documentation	57
6.5.3.1	_d	57
6.6	qpp::Qudit Class Reference	57

6.6.1	Constructor & Destructor Documentation	57
6.6.1.1	Qudit	57
6.6.1.2	~Qudit	57
6.6.2	Member Function Documentation	57
6.6.2.1	getD	58
6.6.2.2	getRho	58
6.6.2.3	measure	58
6.6.2.4	measure	58
6.6.3	Member Data Documentation	58
6.6.3.1	_D	58
6.6.3.2	_rho	58
6.7	qpp::RandomDevices Class Reference	59
6.7.1	Constructor & Destructor Documentation	59
6.7.1.1	RandomDevices	59
6.7.1.2	RandomDevices	59
6.7.1.3	~RandomDevices	59
6.7.2	Member Function Documentation	59
6.7.2.1	getInstance	59
6.7.2.2	operator=	59
6.7.3	Member Data Documentation	59
6.7.3.1	_rd	59
6.7.3.2	_rng	59
6.8	qpp::Timer Class Reference	59
6.8.1	Constructor & Destructor Documentation	60
6.8.1.1	Timer	60
6.8.1.2	~Timer	60
6.8.2	Member Function Documentation	60
6.8.2.1	seconds	60
6.8.2.2	tic	60
6.8.2.3	toc	60
6.8.3	Friends And Related Function Documentation	60
6.8.3.1	operator<<	60
6.8.4	Member Data Documentation	60
6.8.4.1	_end	60
6.8.4.2	_start	60
6.9	qpp::UniformRealDistribution Class Reference	60
6.9.1	Constructor & Destructor Documentation	61
6.9.1.1	UniformRealDistribution	61
6.9.2	Member Function Documentation	61
6.9.2.1	sample	61

6.9.3	Member Data Documentation	61
6.9.3.1	_d	61
7	File Documentation	63
7.1	include/channels.h File Reference	63
7.2	include/classes/exception.h File Reference	64
7.3	include/classes/gates.h File Reference	65
7.4	include/classes/qudit.h File Reference	66
7.5	include/classes/randevs.h File Reference	67
7.6	include/classes/stat.h File Reference	68
7.7	include/classes/timer.h File Reference	69
7.8	include/constants.h File Reference	70
7.9	include/entanglement.h File Reference	72
7.10	include/entropies.h File Reference	73
7.11	include/functions.h File Reference	74
7.12	include/internal.h File Reference	77
7.13	include/io.h File Reference	78
7.14	include/matlab.h File Reference	80
7.15	include/qpp.h File Reference	81
7.16	include/random.h File Reference	81
7.17	include/types.h File Reference	83

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

qpp	9
qpp::ct	41
qpp::internal	42
qpp::types	44

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

qpp::DiscreteDistribution	45
qpp::DiscreteDistributionAbsSquare	46
exception	
qpp::Exception	48
qpp::Gates	52
qpp::NormalDistribution	56
qpp::Qudit	57
qpp::RandomDevices	59
qpp::Timer	59
qpp::UniformRealDistribution	60

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

qpp::DiscreteDistribution	45
qpp::DiscreteDistributionAbsSquare	46
qpp::Exception	48
qpp::Gates	52
qpp::NormalDistribution	56
qpp::Qudit	57
qpp::RandomDevices	59
qpp::Timer	59
qpp::UniformRealDistribution	60

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

include/channels.h	63
include/constants.h	70
include/entanglement.h	72
include/entropies.h	73
include/functions.h	74
include/internal.h	77
include/io.h	78
include/matlab.h	80
include/qpp.h	81
include/random.h	81
include/types.h	83
include/classes/exception.h	64
include/classes/gates.h	65
include/classes/qudit.h	66
include/classes/randevs.h	67
include/classes/stat.h	68
include/classes/timer.h	69

Chapter 5

Namespace Documentation

5.1 qpp Namespace Reference

Namespaces

- [ct](#)
- [internal](#)
- [types](#)

Classes

- class [Exception](#)
- class [Gates](#)
- class [Qudit](#)
- class [RandomDevices](#)
- class [NormalDistribution](#)
- class [UniformRealDistribution](#)
- class [DiscreteDistribution](#)
- class [DiscreteDistributionAbsSquare](#)
- class [Timer](#)

Functions

- [types::cmat channel](#) (const [types::cmat](#) &rho, const std::vector< [types::cmat](#) > &Ks)
- [types::cmat super](#) (const std::vector< [types::cmat](#) > &Ks)
- [types::cmat choi](#) (const std::vector< [types::cmat](#) > &Ks)
- std::vector< [types::cmat](#) > [choi2kraus](#) (const [types::cmat](#) &A)
- template<typename Derived >
[types::cmat schmidtcoeff](#) (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &dims)
- template<typename Derived >
[types::cmat schmidtU](#) (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &dims)
- template<typename Derived >
[types::cmat schmidtV](#) (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &dims)
- template<typename Derived >
[types::cmat schmidtprob](#) (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &dims)
- template<typename Derived >
double [entanglement](#) (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &dims)
- template<typename Derived >
double [gconcurrency](#) (const Eigen::MatrixBase< Derived > &A)

- `template<typename Derived >`
`double shannon (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`double renyi (const double alpha, const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`double renyi_inf (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`double tsallis (const double alpha, const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`double mutualinfo (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &subsys, const std::vector< size_t > &dims)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > transpose (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > conjugate (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > adjoint (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`Derived::Scalar trace (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`Derived::Scalar det (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`Derived::Scalar sum (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`double norm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat evals (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat evecs (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat hevals (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat hevecs (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat funm (const Eigen::MatrixBase< Derived > &A, types::cplx(*f)(const types::cplx &))`
- `template<typename Derived >`
`types::cmat sqrtm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat absm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat expm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat logm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat sinm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat cosm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat spectralpowm (const Eigen::MatrixBase< Derived > &A, const types::cplx z)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > powm (const Eigen::MatrixBase< Derived > &A, size_t n)`

- `template<typename OutputScalar , typename Derived >`
`types::DynMat< OutputScalar > fun (const Eigen::MatrixBase< Derived > &A, OutputScalar(*f)(const type-`
`name Derived::Scalar &))`
- `template<typename Derived1 , typename Derived2 >`
`types::DynMat< typename`
`Derived1::Scalar > kron (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase< Derived2 >`
`&B)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > kronlist (const std::vector< types::DynMat< typename Derived::Scalar > > &As)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > kronpow (const Eigen::MatrixBase< Derived > &A, size_t n)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > reshape (const Eigen::MatrixBase< Derived > &A, size_t rows, size_t cols)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > syspermute (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &perm,`
`const std::vector< size_t > &dims)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > ptrace2 (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &dims)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > ptrace (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &subsys,`
`const std::vector< size_t > &dims)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > ptranspose (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &sub-`
`sys, const std::vector< size_t > &dims)`
- `template<typename Derived1 , typename Derived2 >`
`types::DynMat< typename`
`Derived1::Scalar > comm (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase< Derived2`
`> &B)`
- `template<typename Derived1 , typename Derived2 >`
`types::DynMat< typename`
`Derived1::Scalar > anticomm (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase< De-`
`rived2 > &B)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > proj (const Eigen::MatrixBase< Derived > &V)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > expandout (const Eigen::MatrixBase< Derived > &A, size_t pos, const std::vector< size_t`
`> &dims)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > grams (const std::vector< types::DynMat< typename Derived::Scalar > > &Vs)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > grams (const Eigen::MatrixBase< Derived > &A)`
- `std::vector< size_t > n2multiidx (size_t n, const std::vector< size_t > &dims)`
- `size_t multiidx2n (const std::vector< size_t > &midx, const std::vector< size_t > &dims)`
- `template<typename T >`
`void disp (const T &x, const std::string &separator, const std::string &start="[", const std::string &end="]",`
`std::ostream &os=std::cout)`

- `template<typename T >`
`void displn (const T &x, const std::string &separator, const std::string &start="["`, const std::string &end=""]",
`std::ostream &os=std::cout)`
- `template<typename T >`
`void disp (const T *x, const size_t n, const std::string &separator, const std::string &start="["`, const std::string
&end=""]", std::ostream &os=std::cout)
- `template<typename T >`
`void displn (const T *x, const size_t n, const std::string &separator, const std::string &start="["`, const std-
::string &end=""]", std::ostream &os=std::cout)
- `template<typename Derived >`
`void disp (const Eigen::MatrixBase< Derived > &A, double chop=ct::chop, std::ostream &os=std::cout)`
- `template<typename Derived >`
`void displn (const Eigen::MatrixBase< Derived > &A, double chop=ct::chop, std::ostream &os=std::cout)`
- `void disp (const types::cplx c, double chop=ct::chop, std::ostream &os=std::cout)`
- `void displn (const types::cplx c, double chop=ct::chop, std::ostream &os=std::cout)`
- `template<typename Derived >`
`void save (const Eigen::MatrixBase< Derived > &A, const std::string &fname)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > load (const std::string &fname)`
- `template<typename Derived >`
`Derived loadMATLABmatrix (const std::string &mat_file, const std::string &var_name)`
- `template<>`
`types::dmat loadMATLABmatrix (const std::string &mat_file, const std::string &var_name)`
- `template<>`
`types::cmat loadMATLABmatrix (const std::string &mat_file, const std::string &var_name)`
- `template<typename Derived >`
`void saveMATLABmatrix (const Eigen::MatrixBase< Derived > &A, const std::string &mat_file, const std-`
::string &var_name, const std::string &mode)
- `template<>`
`void saveMATLABmatrix (const Eigen::MatrixBase< typename types::dmat > &A, const std::string &mat_file,`
`const std::string &var_name, const std::string &mode)`
- `template<>`
`void saveMATLABmatrix (const Eigen::MatrixBase< typename types::cmat > &A, const std::string &mat_file,`
`const std::string &var_name, const std::string &mode)`
- `template<typename Derived >`
`Derived rand (size_t rows, size_t cols, double a=0, double b=1)`
- `template<>`
`types::dmat rand (size_t rows, size_t cols, double a, double b)`
- `template<>`
`types::cmat rand (size_t rows, size_t cols, double a, double b)`
- `double rand (double a=0, double b=1)`
- `template<typename Derived >`
`Derived randn (size_t rows, size_t cols, double mean=0, double sigma=1)`
- `template<>`
`types::dmat randn (size_t rows, size_t cols, double mean, double sigma)`
- `template<>`
`types::cmat randn (size_t rows, size_t cols, double mean, double sigma)`
- `double randn (double mean=0, double sigma=1)`
- `types::cmat randU (size_t D)`
- `types::cmat randV (size_t Din, size_t Dout)`
- `std::vector< types::cmat > randkraus (size_t n, size_t D)`
- `types::cmat randH (size_t D)`
- `types::ket randket (size_t D)`
- `types::cmat randrho (size_t D)`

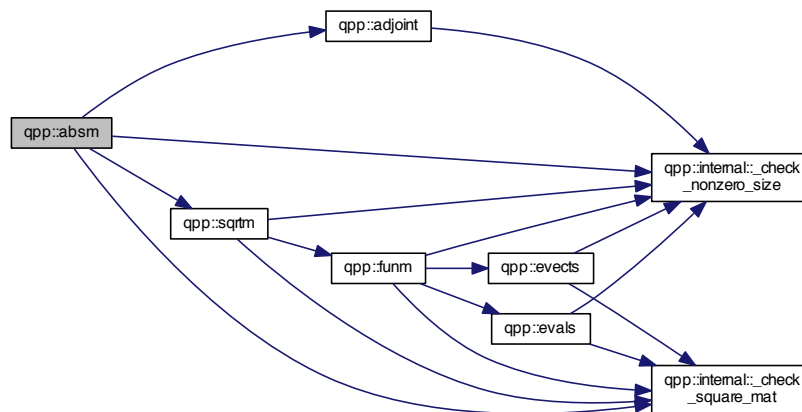
Variables

- `RandomDevices` & `rdevs` = `RandomDevices::getInstance()`
- `const Gates` & `gt` = `Gates::getInstance()`

5.1.1 Function Documentation

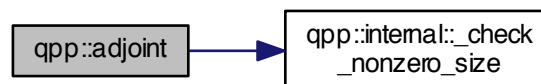
5.1.1.1 `template<typename Derived > types::cmat qpp::absm (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



5.1.1.2 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::adjoint (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



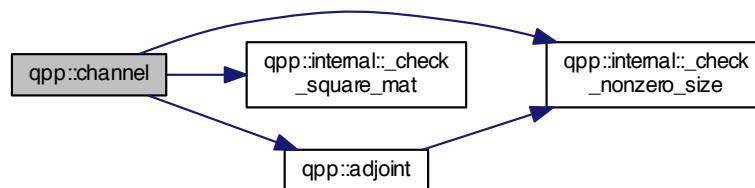
5.1.1.3 `template<typename Derived1 , typename Derived2 > types::DynMat<typename Derived1::Scalar> qpp::anticomm (const Eigen::MatrixBase< Derived1 > & A, const Eigen::MatrixBase< Derived2 > & B)`

Here is the call graph for this function:



5.1.1.4 `types::cmat qpp::channel (const types::cmat & rho, const std::vector< types::cmat > & Ks)`

Here is the call graph for this function:



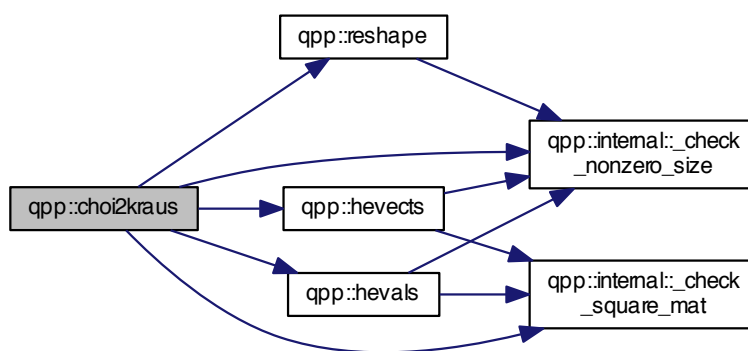
5.1.1.5 `types::cmat qpp::choi (const std::vector< types::cmat > & Ks)`

Here is the call graph for this function:



5.1.1.6 `std::vector<types::cmat> qpp::choi2kraus (const types::cmat & A)`

Here is the call graph for this function:



5.1.1.7 `template<typename Derived1 , typename Derived2 > types::DynMat<typename Derived1::Scalar> qpp::comm (const Eigen::MatrixBase< Derived1 > & A, const Eigen::MatrixBase< Derived2 > & B)`

Here is the call graph for this function:



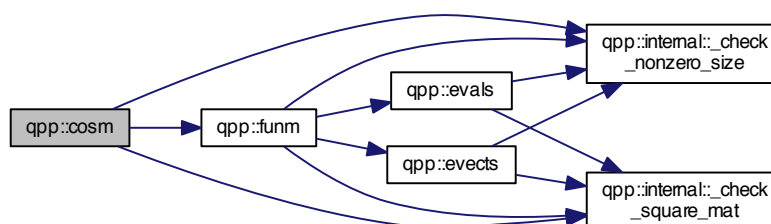
5.1.1.8 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::conjugate (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



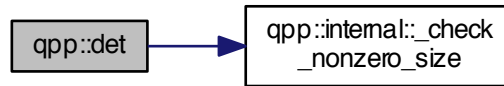
5.1.1.9 `template<typename Derived > types::cmat qpp::cosm (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



5.1.1.10 `template<typename Derived > Derived::Scalar qpp::det (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



5.1.1.11 `template<typename T > void qpp::disp (const T & x, const std::string & separator, const std::string & start = " [", const std::string & end = "] ", std::ostream & os = std::cout)`

5.1.1.12 `template<typename T > void qpp::disp (const T * x, const size_t n, const std::string & separator, const std::string & start = " [", const std::string & end = "] ", std::ostream & os = std::cout)`

5.1.1.13 `template<typename Derived > void qpp::disp (const Eigen::MatrixBase< Derived > & A, double chop = ct::chop, std::ostream & os = std::cout)`

5.1.1.14 `void qpp::disp (const types::cplx c, double chop = ct::chop, std::ostream & os = std::cout)`

Here is the call graph for this function:



5.1.1.15 `template<typename T > void qpp::displn (const T & x, const std::string & separator, const std::string & start = " [", const std::string & end = "] ", std::ostream & os = std::cout)`

Here is the call graph for this function:



5.1.1.16 `template<typename T> void qpp::displn (const T * x, const size_t n, const std::string & separator, const std::string & start = " [", const std::string & end = "] ", std::ostream & os = std::cout)`

Here is the call graph for this function:



5.1.1.17 `template<typename Derived> void qpp::displn (const Eigen::MatrixBase< Derived> & A, double chop = ct::chop, std::ostream & os = std::cout)`

Here is the call graph for this function:



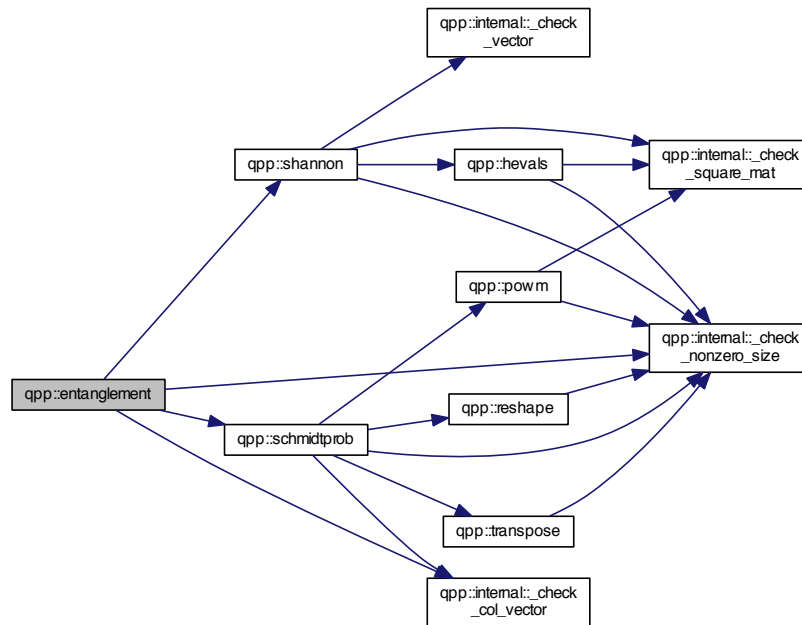
5.1.1.18 `void qpp::displn (const types::cplx c, double chop = ct::chop, std::ostream & os = std::cout)`

Here is the call graph for this function:



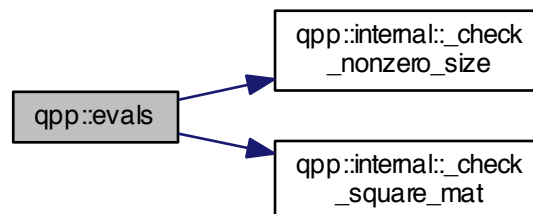
5.1.1.19 `template<typename Derived> double qpp::entanglement (const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims)`

Here is the call graph for this function:



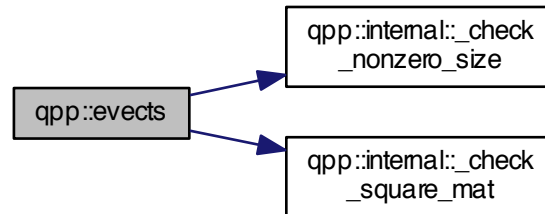
5.1.1.20 `template<typename Derived> types::cmat qpp::evals (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



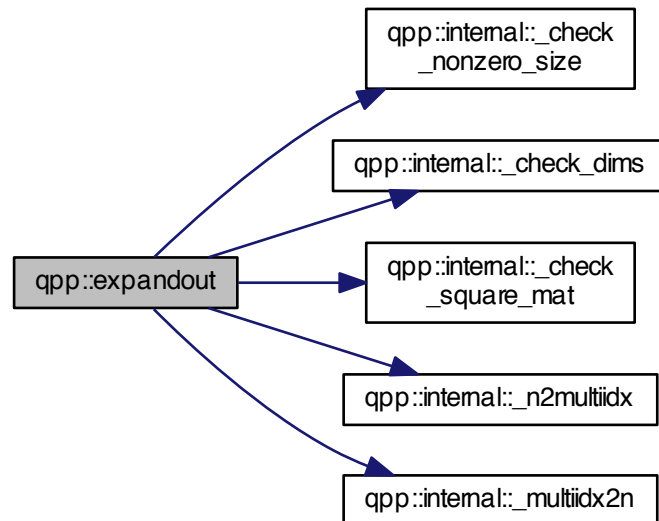
5.1.1.21 `template<typename Derived> types::cmat qpp::evects (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



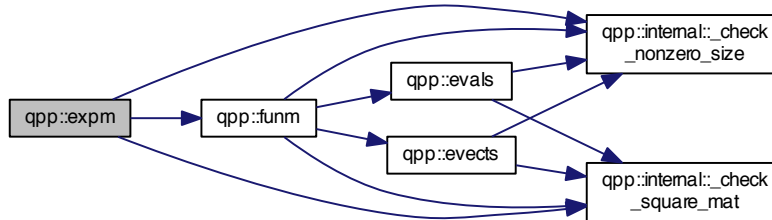
5.1.1.22 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::expandout (const Eigen::MatrixBase< Derived> & A, size_t pos, const std::vector< size_t> & dims)`

Here is the call graph for this function:



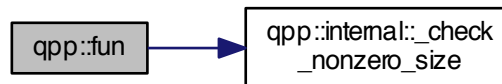
5.1.1.23 `template<typename Derived > types::cmat qpp::expm (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



5.1.1.24 `template<typename OutputScalar , typename Derived > types::DynMat<OutputScalar> qpp::fun (const Eigen::MatrixBase< Derived > & A, OutputScalar*)(const typename Derived::Scalar &) f)`

Here is the call graph for this function:



5.1.1.25 `template<typename Derived > types::cmat qpp::funm (const Eigen::MatrixBase< Derived > & A, types::cplx*)(const types::cplx &) f)`

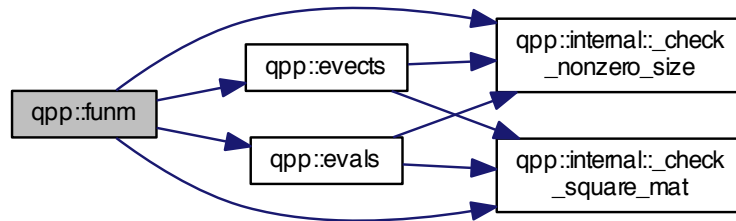
Parameters

<i>A</i>	input matrix
<i>f</i>	function pointer

Returns

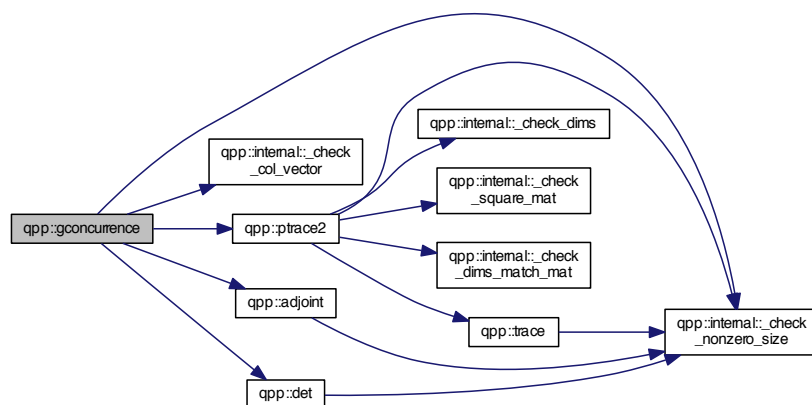
[types::cmat](#)

Here is the call graph for this function:



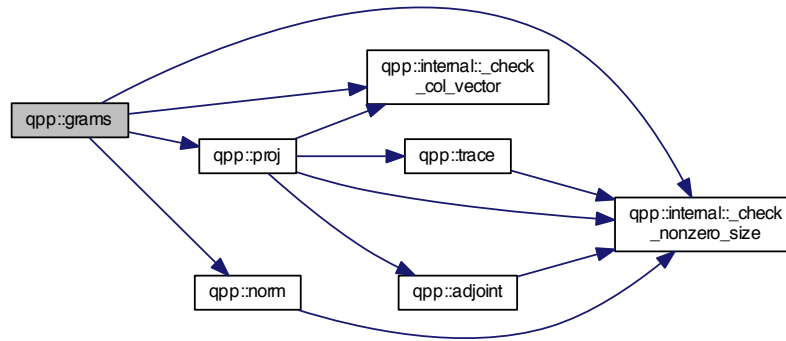
5.1.1.26 `template<typename Derived> double qpp::gconcurrency (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



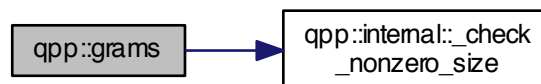
5.1.1.27 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::grams (const std::vector<types::DynMat<typename Derived::Scalar>> & Vs)`

Here is the call graph for this function:



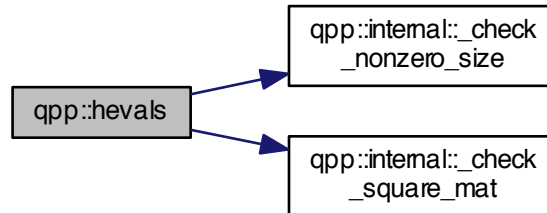
5.1.1.28 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::grams (const Eigen::MatrixBase<Derived> & A)`

Here is the call graph for this function:



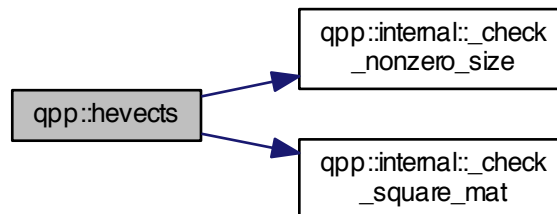
5.1.1.29 `template<typename Derived> types::cmat qpp::hevals (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



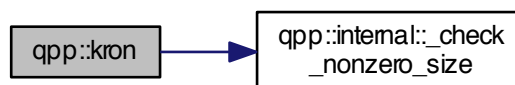
5.1.1.30 `template<typename Derived> types::cmat qpp::hevects (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



5.1.1.31 `template<typename Derived1, typename Derived2> types::DynMat<typename Derived1::Scalar> qpp::kron (const Eigen::MatrixBase< Derived1 > & A, const Eigen::MatrixBase< Derived2 > & B)`

Here is the call graph for this function:



5.1.1.32 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::kronlist (const std::vector< types::DynMat< typename Derived::Scalar> > & As)`

Here is the call graph for this function:



5.1.1.33 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::kronpow (const Eigen::MatrixBase< Derived> & A, size_t n)`

Here is the call graph for this function:



5.1.1.34 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::load (const std::string & fname)`

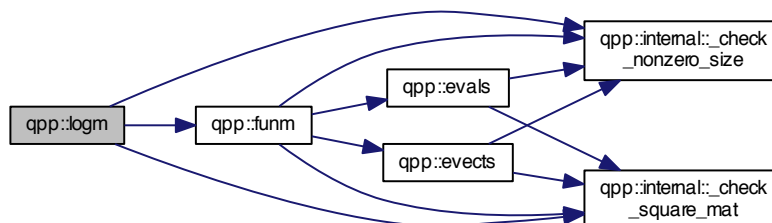
5.1.1.35 `template<typename Derived> Derived qpp::loadMATLABmatrix (const std::string & mat_file, const std::string & var_name)`

5.1.1.36 `template<> types::dmat qpp::loadMATLABmatrix (const std::string & mat_file, const std::string & var_name)`

5.1.1.37 `template<> types::cmat qpp::loadMATLABmatrix (const std::string & mat_file, const std::string & var_name)`

5.1.1.38 `template<typename Derived> types::cmat qpp::logm (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



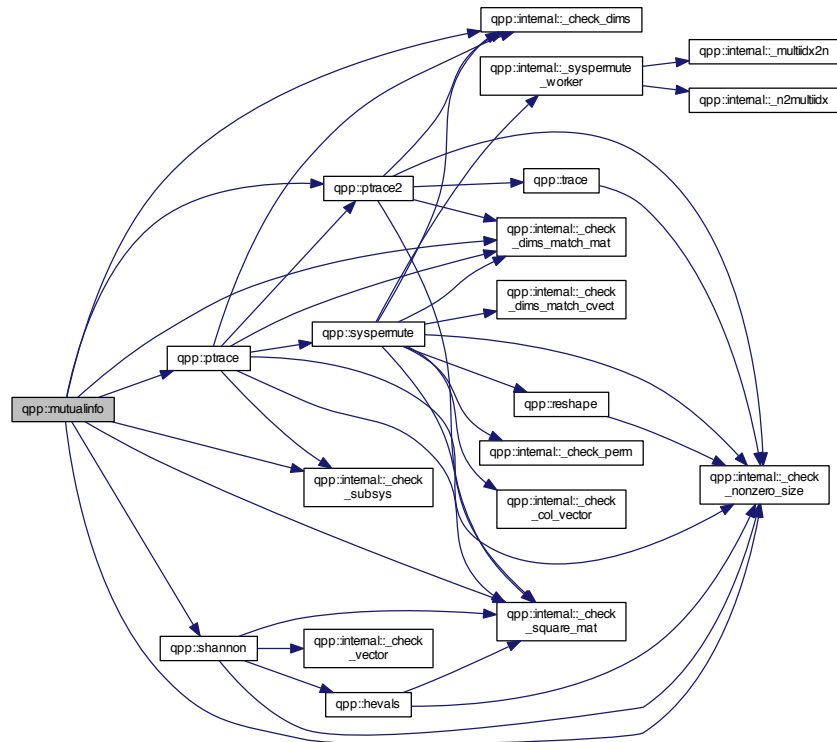
5.1.1.39 `size_t qpp::multiidx2n (const std::vector< size_t> & midx, const std::vector< size_t> & dims)`

Here is the call graph for this function:



5.1.1.40 `template<typename Derived> double qpp::mutualinfo (const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & subsys, const std::vector< size_t> & dims)`

Here is the call graph for this function:



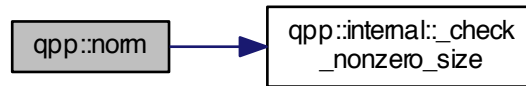
5.1.1.41 `std::vector<size_t> qpp::n2multidx (size_t n, const std::vector< size_t> & dims)`

Here is the call graph for this function:



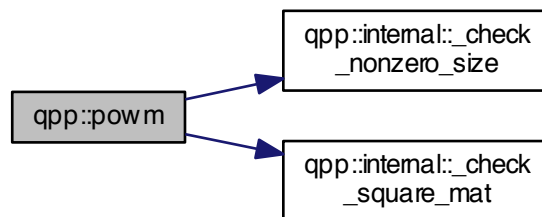
5.1.1.42 `template<typename Derived > double qpp::norm (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



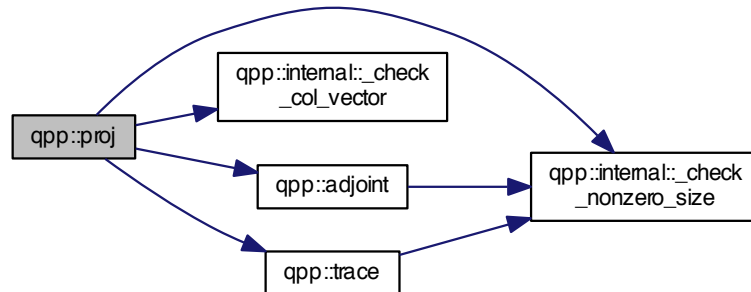
5.1.1.43 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::powm (const Eigen::MatrixBase< Derived > & A, size_t n)`

Here is the call graph for this function:



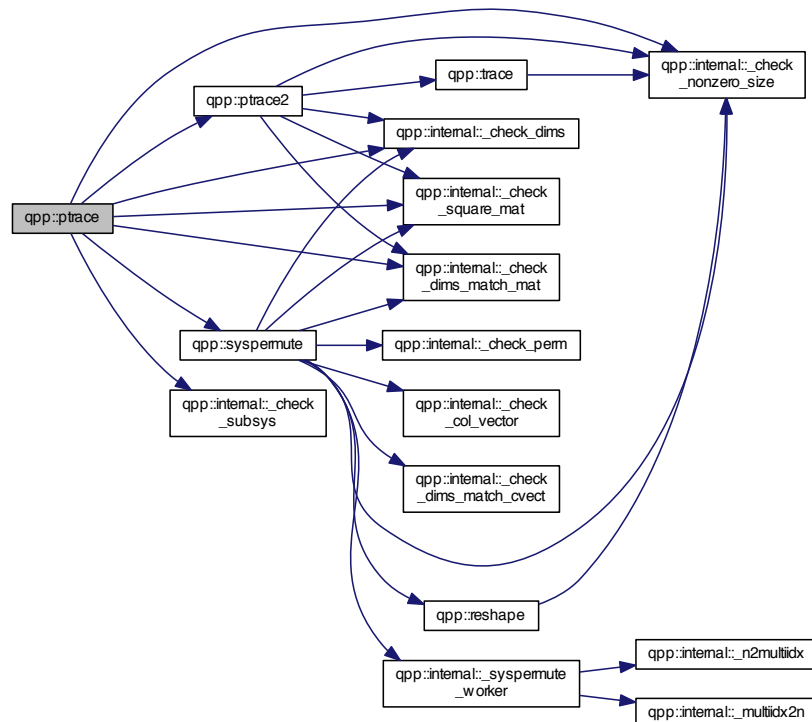
5.1.1.44 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::proj (const Eigen::MatrixBase<Derived > & V)`

Here is the call graph for this function:



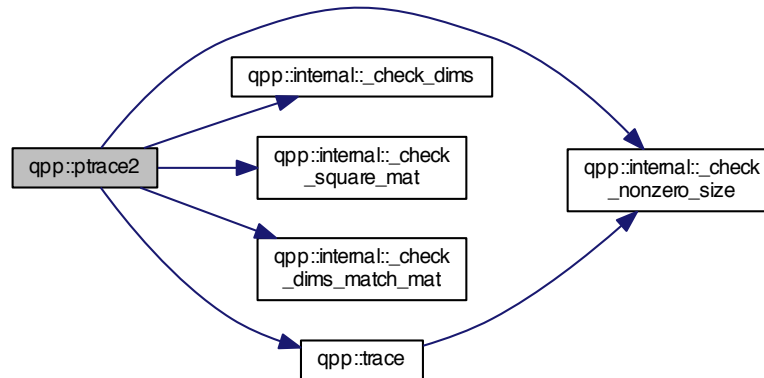
5.1.1.45 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::ptrace (const Eigen::MatrixBase<Derived > & A, const std::vector<size_t> & subsys, const std::vector<size_t> & dims)`

Here is the call graph for this function:



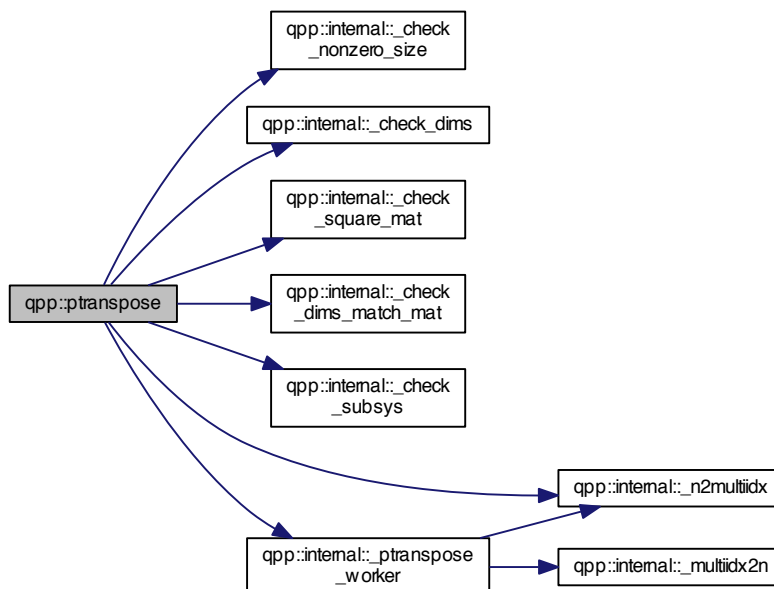
5.1.1.46 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::ptrace2 (const Eigen::MatrixBase< Derived > & A, const std::vector< size_t > & dims)`

Here is the call graph for this function:



5.1.1.47 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::ptrtranspose (const Eigen::MatrixBase< Derived > & A, const std::vector< size_t > & subsys, const std::vector< size_t > & dims)`

Here is the call graph for this function:



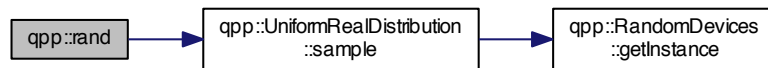
5.1.1.48 `template<typename Derived > Derived qpp::rand (size_t rows, size_t cols, double a = 0, double b = 1)`

5.1.1.49 `template<> types::dmat qpp::rand (size_t rows, size_t cols, double a, double b)`

5.1.1.50 `template<> types::cmat qpp::rand (size_t rows, size_t cols, double a, double b)`

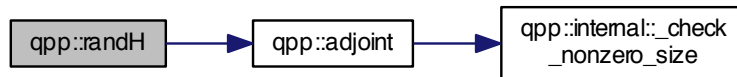
5.1.1.51 `double qpp::rand (double a = 0, double b = 1)`

Here is the call graph for this function:



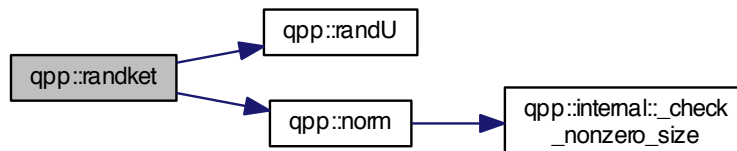
5.1.1.52 `types::cmat qpp::randH (size_t D)`

Here is the call graph for this function:



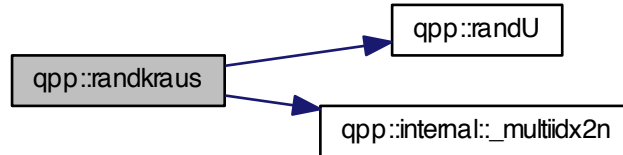
5.1.1.53 `types::ket qpp::randket (size_t D)`

Here is the call graph for this function:



5.1.1.54 `std::vector<types::cmat> qpp::randkraus (size_t n, size_t D)`

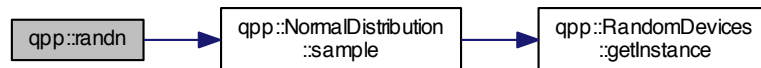
Here is the call graph for this function:



5.1.1.55 `template<typename Derived > Derived qpp::randn (size_t rows, size_t cols, double mean = 0, double sigma = 1)`

5.1.1.56 `template<> types::dmat qpp::randn (size_t rows, size_t cols, double mean, double sigma)`

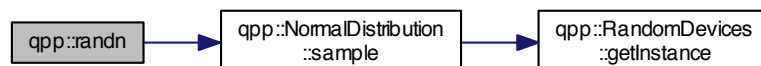
Here is the call graph for this function:



5.1.1.57 `template<> types::cmat qpp::randn (size_t rows, size_t cols, double mean, double sigma)`

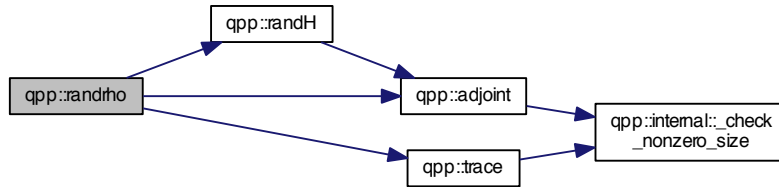
5.1.1.58 `double qpp::randn (double mean = 0, double sigma = 1)`

Here is the call graph for this function:



5.1.1.59 `types::cmat qpp::randrho (size_t D)`

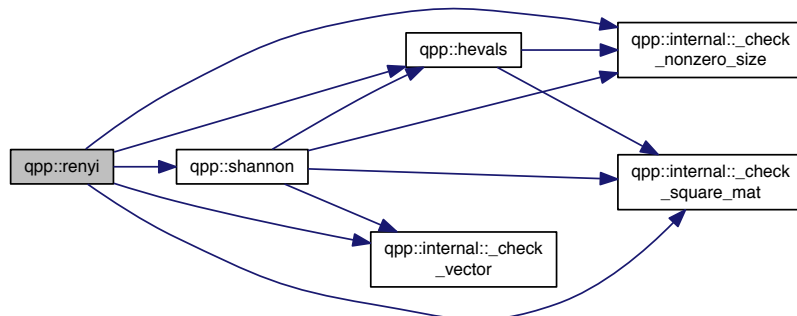
Here is the call graph for this function:

5.1.1.60 `types::cmat qpp::randU (size_t D)`5.1.1.61 `types::cmat qpp::randV (size_t Din, size_t Dout)`

Here is the call graph for this function:

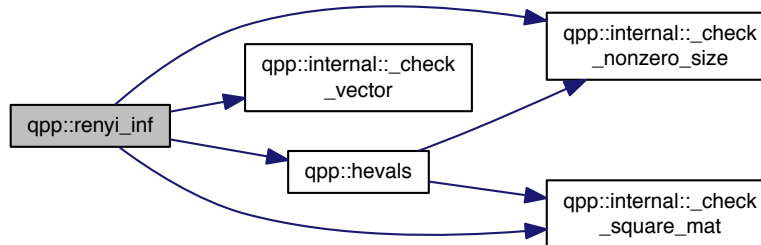
5.1.1.62 `template<typename Derived> double qpp::renyi (const double alpha, const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



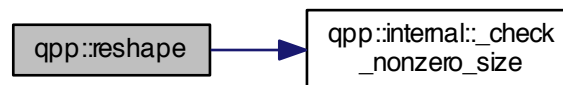
5.1.1.63 `template<typename Derived> double qpp::renyi_inf (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



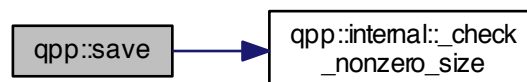
5.1.1.64 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::reshape (const Eigen::MatrixBase< Derived> & A, size_t rows, size_t cols)`

Here is the call graph for this function:



5.1.1.65 `template<typename Derived> void qpp::save (const Eigen::MatrixBase< Derived> & A, const std::string & fname)`

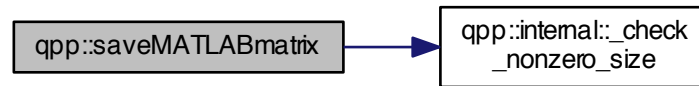
Here is the call graph for this function:



5.1.1.66 `template<typename Derived> void qpp::saveMATLABmatrix (const Eigen::MatrixBase< Derived> & A, const std::string & mat_file, const std::string & var_name, const std::string & mode)`

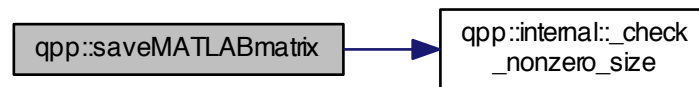
5.1.1.67 `template<> void qpp::saveMATLABmatrix (const Eigen::MatrixBase< typename types::dmat > & A, const std::string & mat_file, const std::string & var_name, const std::string & mode)`

Here is the call graph for this function:



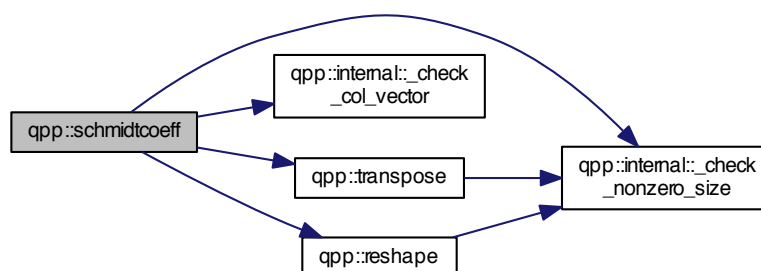
5.1.1.68 `template<> void qpp::saveMATLABmatrix (const Eigen::MatrixBase< typename types::cmat > & A, const std::string & mat_file, const std::string & var_name, const std::string & mode)`

Here is the call graph for this function:



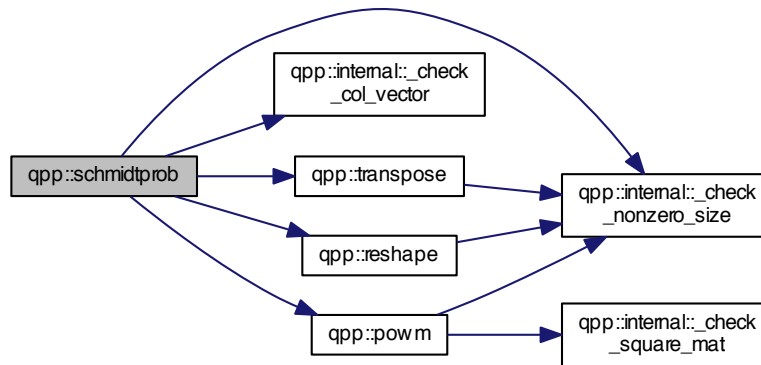
5.1.1.69 `template<typename Derived> types::cmat qpp::schmidtcoeff (const Eigen::MatrixBase< Derived > & A, const std::vector< size_t > & dims)`

Here is the call graph for this function:



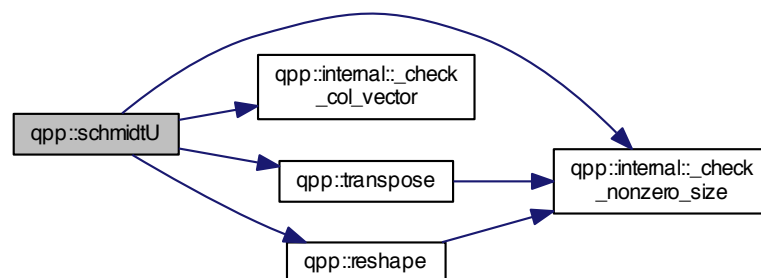
5.1.1.70 `template<typename Derived> types::cmat qpp::schmidtprob (const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims)`

Here is the call graph for this function:



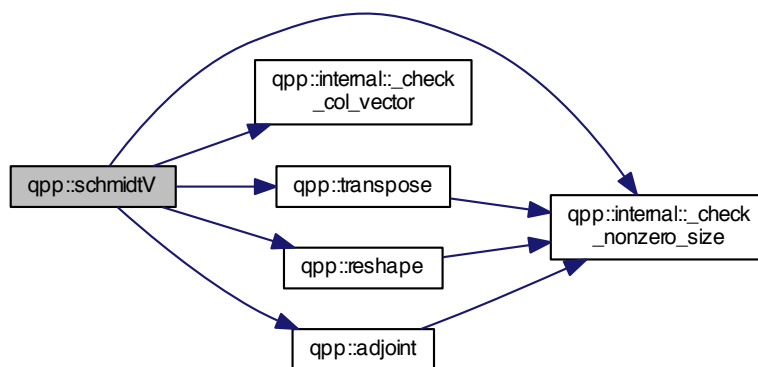
5.1.1.71 `template<typename Derived> types::cmat qpp::schmidtU (const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims)`

Here is the call graph for this function:



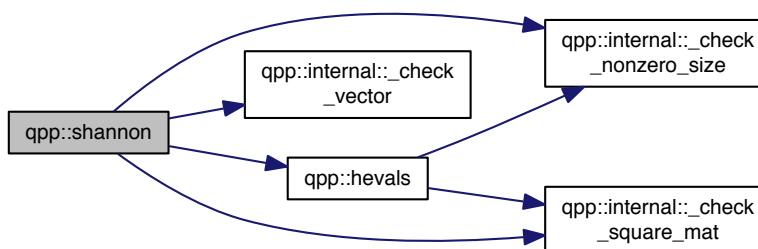
5.1.1.72 `template<typename Derived> types::cmat qpp::schmidtV (const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims)`

Here is the call graph for this function:



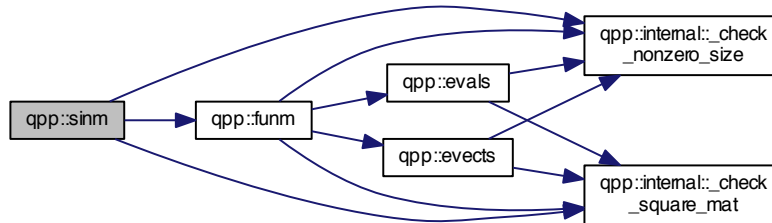
5.1.1.73 `template<typename Derived> double qpp::shannon (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



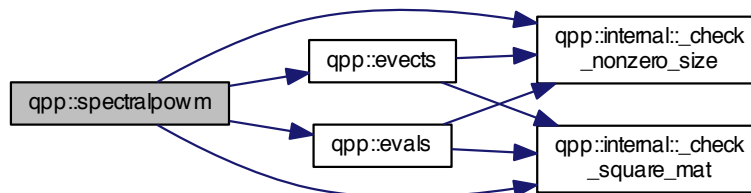
5.1.1.74 `template<typename Derived> types::cmat qpp::sinm (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



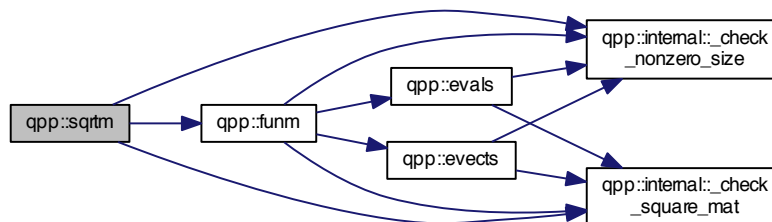
5.1.1.75 `template<typename Derived> types::cmat qpp::spectralpowm (const Eigen::MatrixBase< Derived> & A, const types::cplx z)`

Here is the call graph for this function:



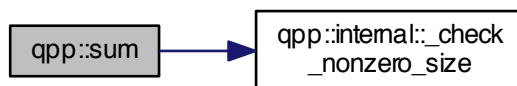
5.1.1.76 `template<typename Derived> types::cmat qpp::sqrtm (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



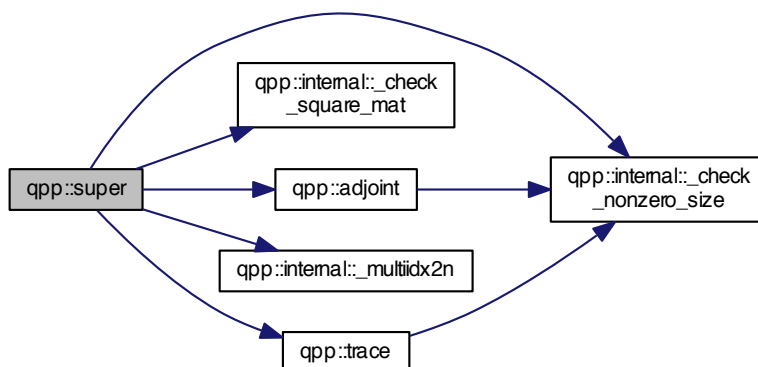
5.1.1.77 `template<typename Derived > Derived::Scalar qpp::sum (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



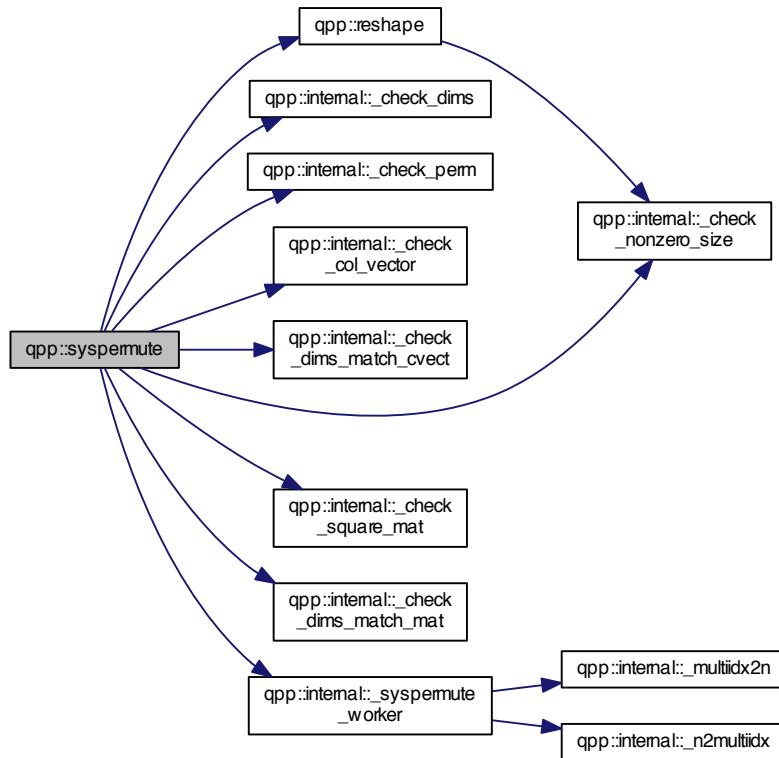
5.1.1.78 `types::cmat qpp::super (const std::vector< types::cmat > & Ks)`

Here is the call graph for this function:



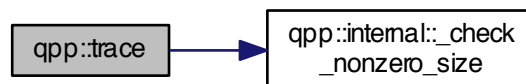
5.1.1.79 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::syspermute (const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & perm, const std::vector< size_t> & dims)`

Here is the call graph for this function:



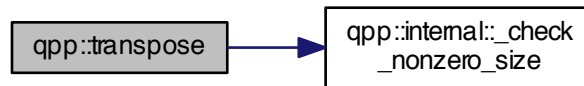
5.1.1.80 `template<typename Derived> Derived::Scalar qpp::trace (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



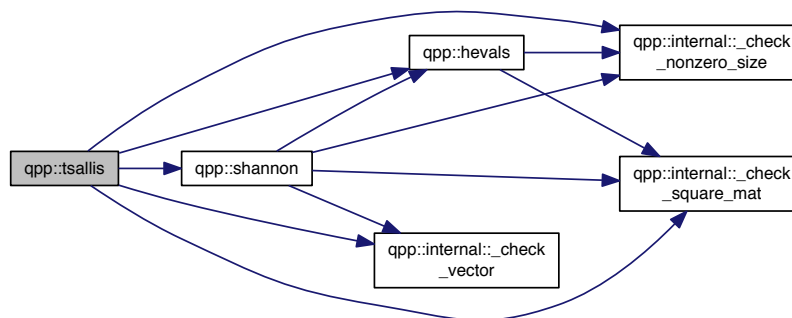
5.1.1.81 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::transpose (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



5.1.1.82 `template<typename Derived> double qpp::tsallis (const double alpha, const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



5.1.2 Variable Documentation

5.1.2.1 `const Gates& qpp::gt = Gates::getInstance()`

5.1.2.2 `RandomDevices& qpp::rdevs = RandomDevices::getInstance()`

5.2 qpp::ct Namespace Reference

Functions

- `std::complex< double> omega (size_t D)`

Variables

- `const double chop = 1e-10`
- `const double eps = 1e-12`
- `const std::complex< double> ii = { 0, 1 }`

- const double `pi` = 3.141592653589793238462643383279502884
- const double `ee` = 2.718281828459045235360287471352662497

5.2.1 Function Documentation

5.2.1.1 `std::complex<double> qpp::ct::omega (size_t D)`

5.2.2 Variable Documentation

5.2.2.1 `const double qpp::ct::chop = 1e-10`

5.2.2.2 `const double qpp::ct::ee = 2.718281828459045235360287471352662497`

5.2.2.3 `const double qpp::ct::eps = 1e-12`

5.2.2.4 `const std::complex<double> qpp::ct::ii = { 0, 1 }`

5.2.2.5 `const double qpp::ct::pi = 3.141592653589793238462643383279502884`

5.3 qpp::internal Namespace Reference

Functions

- void `_n2multiidx` (size_t *n*, size_t *numdims*, const size_t **dims*, size_t **result*)
- size_t `_multiidx2n` (const size_t **midx*, size_t *numdims*, const size_t **dims*)
- template<typename *Derived* >
bool `_check_square_mat` (const Eigen::MatrixBase< *Derived* > &*A*)
- template<typename *Derived* >
bool `_check_vector` (const Eigen::MatrixBase< *Derived* > &*A*)
- template<typename *Derived* >
bool `_check_row_vector` (const Eigen::MatrixBase< *Derived* > &*A*)
- template<typename *Derived* >
bool `_check_col_vector` (const Eigen::MatrixBase< *Derived* > &*A*)
- template<typename *T* >
bool `_check_nonzero_size` (const *T* &*x*)
- bool `_check_dims` (const std::vector< size_t > &*dims*)
- template<typename *Derived* >
bool `_check_dims_match_mat` (const std::vector< size_t > &*dims*, const Eigen::MatrixBase< *Derived* > &*A*)
- template<typename *Derived* >
bool `_check_dims_match_cvect` (const std::vector< size_t > &*dims*, const Eigen::MatrixBase< *Derived* > &*V*)
- template<typename *Derived* >
bool `_check_dims_match_rvect` (const std::vector< size_t > &*dims*, const Eigen::MatrixBase< *Derived* > &*V*)
- bool `_check_eq_dims` (const std::vector< size_t > &*dims*, size_t *dim*)
- bool `_check_subsys` (const std::vector< size_t > &*subsys*, const std::vector< size_t > &*dims*)
- bool `_check_perm` (const std::vector< size_t > &*perm*, const std::vector< size_t > &*dims*)
- template<typename *Scalar* >
void `_syspermute_worker` (size_t *numdims*, const size_t **cdims*, const size_t **cperm*, size_t *i*, size_t &*iperm*, const types::DynMat< *Scalar* > &*V*, types::DynMat< *Scalar* > &*result*)
- template<typename *Scalar* >
void `_ptranspose_worker` (const size_t **midxc*, size_t *numdims*, size_t *numsubsys*, const size_t **cdims*, const size_t **csubsys*, size_t *i*, size_t *j*, size_t &*iperm*, size_t &*jperm*, const types::DynMat< *Scalar* > &*A*, types::DynMat< *Scalar* > &*result*)

5.3.1 Function Documentation

5.3.1.1 `template<typename Derived > bool qpp::internal::_check_col_vector (const Eigen::MatrixBase< Derived > & A)`

5.3.1.2 `bool qpp::internal::_check_dims (const std::vector< size_t > & dims)`

5.3.1.3 `template<typename Derived > bool qpp::internal::_check_dims_match_cvect (const std::vector< size_t > & dims, const Eigen::MatrixBase< Derived > & V)`

5.3.1.4 `template<typename Derived > bool qpp::internal::_check_dims_match_mat (const std::vector< size_t > & dims, const Eigen::MatrixBase< Derived > & A)`

5.3.1.5 `template<typename Derived > bool qpp::internal::_check_dims_match_rvect (const std::vector< size_t > & dims, const Eigen::MatrixBase< Derived > & V)`

5.3.1.6 `bool qpp::internal::_check_eq_dims (const std::vector< size_t > & dims, size_t dim)`

5.3.1.7 `template<typename T > bool qpp::internal::_check_nonzero_size (const T & x)`

5.3.1.8 `bool qpp::internal::_check_perm (const std::vector< size_t > & perm, const std::vector< size_t > & dims)`

5.3.1.9 `template<typename Derived > bool qpp::internal::_check_row_vector (const Eigen::MatrixBase< Derived > & A)`

5.3.1.10 `template<typename Derived > bool qpp::internal::_check_square_mat (const Eigen::MatrixBase< Derived > & A)`

5.3.1.11 `bool qpp::internal::_check_subsys (const std::vector< size_t > & subsys, const std::vector< size_t > & dims)`

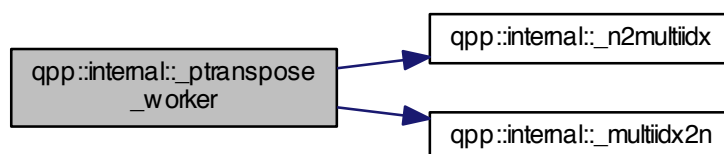
5.3.1.12 `template<typename Derived > bool qpp::internal::_check_vector (const Eigen::MatrixBase< Derived > & A)`

5.3.1.13 `size_t qpp::internal::_multiidx2n (const size_t * midx, size_t numdims, const size_t * dims)`

5.3.1.14 `void qpp::internal::_n2multiidx (size_t n, size_t numdims, const size_t * dims, size_t * result)`

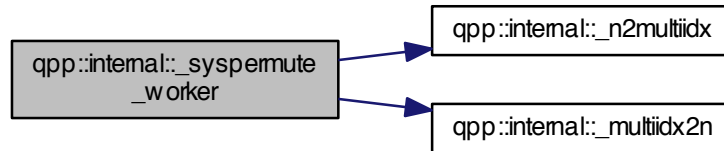
5.3.1.15 `template<typename Scalar > void qpp::internal::_pttranspose_worker (const size_t * midxcol, size_t numdims, size_t numsubsys, const size_t * cdims, const size_t * csubsys, size_t i, size_t j, size_t iperm, size_t jperm, const types::DynMat< Scalar > & A, types::DynMat< Scalar > & result)`

Here is the call graph for this function:



5.3.1.16 `template<typename Scalar > void qpp::internal::_syspermute_worker (size_t numdims, const size_t * cdims, const size_t * cperm, size_t i, size_t & iperm, const types::DynMat< Scalar > & V, types::DynMat< Scalar > & result)`

Here is the call graph for this function:



5.4 qpp::types Namespace Reference

Typedefs

- `typedef std::complex< double > cplx`
- `typedef Eigen::MatrixXcd cmat`
- `typedef Eigen::MatrixXd dmat`
- `typedef Eigen::MatrixXf fmat`
- `typedef Eigen::MatrixXi imat`
- `typedef Eigen::Matrix< cplx, Eigen::Dynamic, 1 > ket`
- `typedef Eigen::Matrix< cplx, 1, Eigen::Dynamic > bra`
- `template<typename Scalar > using DynMat = Eigen::Matrix< Scalar, Eigen::Dynamic, Eigen::Dynamic >`

5.4.1 Typedef Documentation

5.4.1.1 `typedef Eigen::Matrix<cplx, 1, Eigen::Dynamic> qpp::types::bra`

5.4.1.2 `typedef Eigen::MatrixXcd qpp::types::cmat`

5.4.1.3 `typedef std::complex<double> qpp::types::cplx`

5.4.1.4 `typedef Eigen::MatrixXd qpp::types::dmat`

5.4.1.5 `template<typename Scalar > using qpp::types::DynMat = typedef Eigen::Matrix<Scalar, Eigen::Dynamic, Eigen::Dynamic>`

5.4.1.6 `typedef Eigen::MatrixXf qpp::types::fmat`

5.4.1.7 `typedef Eigen::MatrixXi qpp::types::imat`

5.4.1.8 `typedef Eigen::Matrix<cplx, Eigen::Dynamic, 1> qpp::types::ket`

Chapter 6

Class Documentation

6.1 qpp::DiscreteDistribution Class Reference

```
#include <stat.h>
```

Public Member Functions

- `template<typename InputIterator >`
`DiscreteDistribution` (InputIterator first, InputIterator last)
- `DiscreteDistribution` (std::initializer_list< double > weights)
- `DiscreteDistribution` (std::vector< double > weights)
- `size_t sample ()`
- `std::vector< double > probabilities ()`

Protected Attributes

- `std::discrete_distribution`
`< size_t > _d`

6.1.1 Constructor & Destructor Documentation

6.1.1.1 `template<typename InputIterator > qpp::DiscreteDistribution::DiscreteDistribution (InputIterator first, InputIterator last)` [inline]

6.1.1.2 `qpp::DiscreteDistribution::DiscreteDistribution (std::initializer_list< double > weights)` [inline]

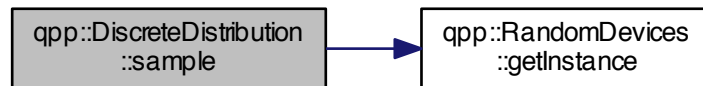
6.1.1.3 `qpp::DiscreteDistribution::DiscreteDistribution (std::vector< double > weights)` [inline]

6.1.2 Member Function Documentation

6.1.2.1 `std::vector<double> qpp::DiscreteDistribution::probabilities ()` [inline]

6.1.2.2 `size_t qpp::DiscreteDistribution::sample () [inline]`

Here is the call graph for this function:



6.1.3 Member Data Documentation

6.1.3.1 `std::discrete_distribution<size_t> qpp::DiscreteDistribution::_d [protected]`

The documentation for this class was generated from the following file:

- [include/classes/stat.h](#)

6.2 `qpp::DiscreteDistributionAbsSquare` Class Reference

```
#include <stat.h>
```

Public Member Functions

- `template<typename InputIterator >`
`DiscreteDistributionAbsSquare` (InputIterator first, InputIterator last)
- `DiscreteDistributionAbsSquare` (std::initializer_list< [types::cplx](#) > amplitudes)
- `DiscreteDistributionAbsSquare` (std::vector< [types::cplx](#) > amplitudes)
- `DiscreteDistributionAbsSquare` (const [types::cmat](#) &V)
- `size_t sample ()`
- `std::vector< double > probabilities ()`

Protected Member Functions

- `template<typename InputIterator >`
`std::vector< double > cplx2weights` (InputIterator first, InputIterator last)

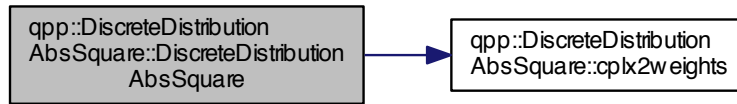
Protected Attributes

- `std::discrete_distribution`
`< size_t > _d`

6.2.1 Constructor & Destructor Documentation

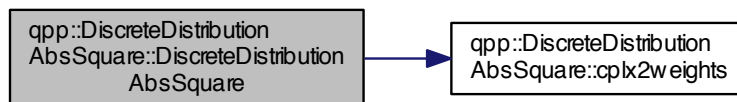
6.2.1.1 `template<typename InputIterator> qpp::DiscreteDistributionAbsSquare::DiscreteDistributionAbsSquare (InputIterator first, InputIterator last) [inline]`

Here is the call graph for this function:



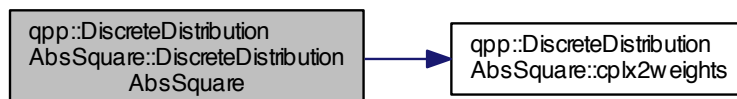
6.2.1.2 `qpp::DiscreteDistributionAbsSquare::DiscreteDistributionAbsSquare (std::initializer_list< types::cplx > amplitudes) [inline]`

Here is the call graph for this function:



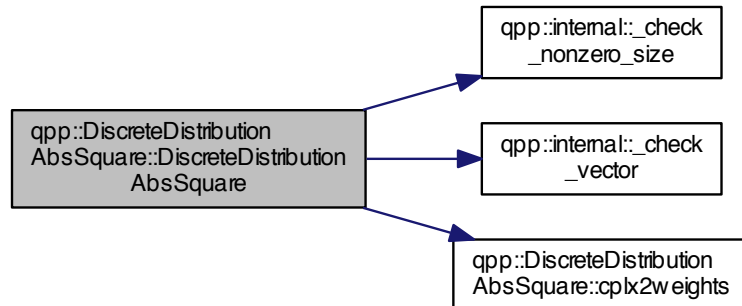
6.2.1.3 `qpp::DiscreteDistributionAbsSquare::DiscreteDistributionAbsSquare (std::vector< types::cplx > amplitudes) [inline]`

Here is the call graph for this function:



6.2.1.4 `qpp::DiscreteDistributionAbsSquare::DiscreteDistributionAbsSquare (const types::cmat & V) [inline]`

Here is the call graph for this function:



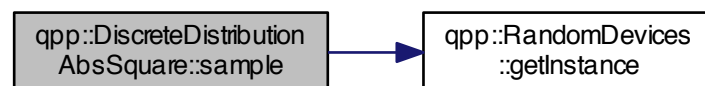
6.2.2 Member Function Documentation

6.2.2.1 `template<typename InputIterator> std::vector<double> qpp::DiscreteDistributionAbsSquare::cplx2weights (InputIterator first, InputIterator last) [inline],[protected]`

6.2.2.2 `std::vector<double> qpp::DiscreteDistributionAbsSquare::probabilities () [inline]`

6.2.2.3 `size_t qpp::DiscreteDistributionAbsSquare::sample () [inline]`

Here is the call graph for this function:



6.2.3 Member Data Documentation

6.2.3.1 `std::discrete_distribution<size_t> qpp::DiscreteDistributionAbsSquare::_d [protected]`

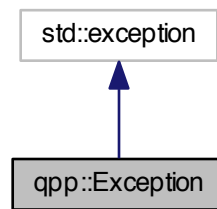
The documentation for this class was generated from the following file:

- [include/classes/stat.h](#)

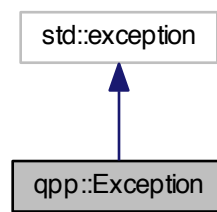
6.3 qpp::Exception Class Reference

```
#include <exception.h>
```

Inheritance diagram for qpp::Exception:



Collaboration diagram for qpp::Exception:



Public Types

- enum `Type` {
`Type::UNKNOWN_EXCEPTION = 0`, `Type::ZERO_SIZE`, `Type::MATRIX_NOT_SQUARE`, `Type::MATRIX_NOT_CVECTOR`,
`Type::MATRIX_NOT_RVECTOR`, `Type::MATRIX_NOT_VECTOR`, `Type::MATRIX_NOT_SQUARE_OR_CVECTOR`, `Type::MATRIX_NOT_SQUARE_OR_RVECTOR`,
`Type::MATRIX_NOT_SQUARE_OR_VECTOR`, `Type::DIMS_INVALID`, `Type::DIMS_NOT_EQUAL`, `Type::DIMS_MISMATCH_MATRIX`,
`Type::DIMS_MISMATCH_CVECTOR`, `Type::DIMS_MISMATCH_RVECTOR`, `Type::DIMS_MISMATCH_VECTOR`, `Type::SUBSYS_MISMATCH_DIMS`,
`Type::PERM_MISMATCH_DIMS`, `Type::NOT_QUBIT_GATE`, `Type::NOT_QUBIT_SUBSYS`, `Type::NOT_BIPARTITE`,
`Type::OUT_OF_RANGE`, `Type::UNDEFINED_TYPE`, `Type::TYPE_MISMATCH`, `Type::CUSTOM_EXCEPTION` }

Public Member Functions

- `Exception` (const std::string &where, const `Type` &type)
- `Exception` (const std::string &where, const std::string &custom)
- virtual const char * `what` () const noexcept override
- virtual `~Exception` () noexcept

Private Member Functions

- `std::string _construct_exception_msg ()`

Private Attributes

- `std::string _where`
- `std::string _msg`
- `Type _type`
- `std::string _custom`

6.3.1 Member Enumeration Documentation

6.3.1.1 `enum qpp::Exception::Type` `[strong]`

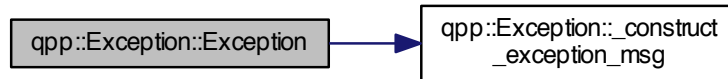
Enumerator

UNKNOWN_EXCEPTION
ZERO_SIZE
MATRIX_NOT_SQUARE
MATRIX_NOT_CVECTOR
MATRIX_NOT_RVECTOR
MATRIX_NOT_VECTOR
MATRIX_NOT_SQUARE_OR_CVECTOR
MATRIX_NOT_SQUARE_OR_RVECTOR
MATRIX_NOT_SQUARE_OR_VECTOR
DIMS_INVALID
DIMS_NOT_EQUAL
DIMS_MISMATCH_MATRIX
DIMS_MISMATCH_CVECTOR
DIMS_MISMATCH_RVECTOR
DIMS_MISMATCH_VECTOR
SUBSYS_MISMATCH_DIMS
PERM_MISMATCH_DIMS
NOT_QUBIT_GATE
NOT_QUBIT_SUBSYS
NOT_BIPARTITE
OUT_OF_RANGE
UNDEFINED_TYPE
TYPE_MISMATCH
CUSTOM_EXCEPTION

6.3.2 Constructor & Destructor Documentation

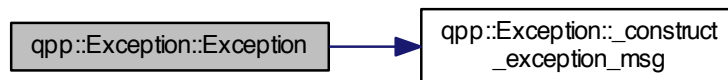
6.3.2.1 `qpp::Exception::Exception (const std::string & where, const Type & type)` `[inline]`

Here is the call graph for this function:



6.3.2.2 `qpp::Exception::Exception (const std::string & where, const std::string & custom)` `[inline]`

Here is the call graph for this function:



6.3.2.3 `virtual qpp::Exception::~~Exception ()` `[inline]`, `[virtual]`, `[noexcept]`

6.3.3 Member Function Documentation

6.3.3.1 `std::string qpp::Exception::_construct_exception_msg ()` `[inline]`, `[private]`

6.3.3.2 `virtual const char* qpp::Exception::what () const` `[inline]`, `[override]`, `[virtual]`, `[noexcept]`

6.3.4 Member Data Documentation

6.3.4.1 `std::string qpp::Exception::_custom` `[private]`

6.3.4.2 `std::string qpp::Exception::_msg` `[private]`

6.3.4.3 `Type qpp::Exception::_type` `[private]`

6.3.4.4 `std::string qpp::Exception::_where` `[private]`

The documentation for this class was generated from the following file:

- [include/classes/exception.h](#)

6.4 qpp::Gates Class Reference

```
#include <gates.h>
```

Public Member Functions

- [Gates](#) (const [Gates](#) &)=delete
- [Gates](#) & [operator=](#) (const [Gates](#) &)=delete
- virtual [~Gates](#) ()=default
- [types::cmat Rtheta](#) (double theta) const
- [types::cmat Id](#) (size_t D) const
- [types::cmat Zd](#) (size_t D) const
- [types::cmat Fd](#) (size_t D) const
- [types::cmat Xd](#) (size_t D) const
- [types::cmat CTRL](#) (const [types::cmat](#) &A, const std::vector< size_t > &ctrl, const std::vector< size_t > &gate, size_t n, size_t D=2) const

Static Public Member Functions

- static const [Gates](#) & [getInstance](#) ()

Public Attributes

- [types::cmat Id2](#)
- [types::cmat H](#)
- [types::cmat X](#)
- [types::cmat Y](#)
- [types::cmat Z](#)
- [types::cmat S](#)
- [types::cmat T](#)
- [types::cmat CNOTab](#)
- [types::cmat CZ](#)
- [types::cmat CS](#)
- [types::cmat CNOTba](#)
- [types::cmat SWAP](#)
- [types::cmat TOF](#)
- [types::cmat FRED](#)
- [types::ket x0](#)
- [types::ket x1](#)
- [types::ket y0](#)
- [types::ket y1](#)
- [types::ket z0](#)
- [types::ket z1](#)
- [types::cmat px0](#)
- [types::cmat px1](#)
- [types::cmat py0](#)
- [types::cmat py1](#)
- [types::cmat pz0](#)
- [types::cmat pz1](#)
- [types::ket b00](#)
- [types::ket b01](#)
- [types::ket b10](#)
- [types::ket b11](#)

- [types::cmat pb00](#)
- [types::cmat pb01](#)
- [types::cmat pb10](#)
- [types::cmat pb11](#)

Private Member Functions

- [Gates \(\)](#)

6.4.1 Constructor & Destructor Documentation

6.4.1.1 `qpp::Gates::Gates () [inline], [private]`

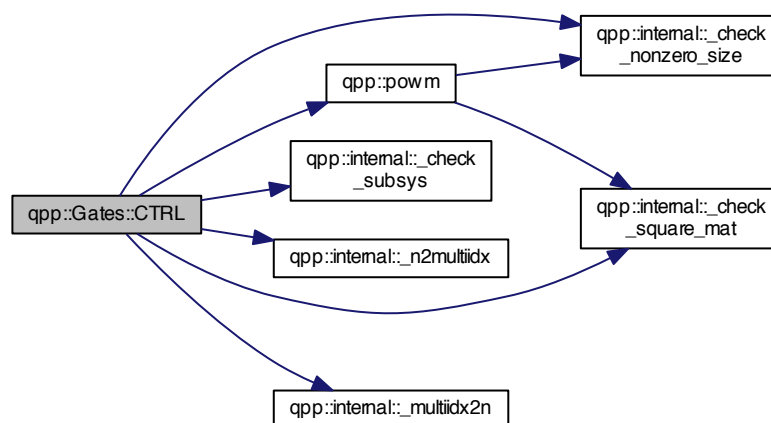
6.4.1.2 `qpp::Gates::Gates (const Gates &) [delete]`

6.4.1.3 `virtual qpp::Gates::~~Gates () [virtual], [default]`

6.4.2 Member Function Documentation

6.4.2.1 `types::cmat qpp::Gates::CTRL (const types::cmat & A, const std::vector< size_t > & ctrl, const std::vector< size_t > & gate, size_t n, size_t D = 2) const [inline]`

Here is the call graph for this function:



6.4.2.2 `types::cmat qpp::Gates::Fd (size_t D) const` `[inline]`

Here is the call graph for this function:



6.4.2.3 `static const Gates& qpp::Gates::getInstance ()` `[inline]`, `[static]`

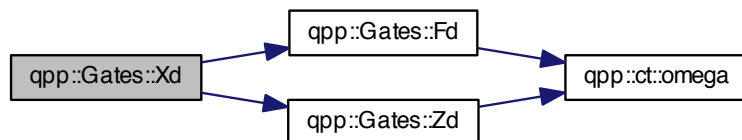
6.4.2.4 `types::cmat qpp::Gates::Id (size_t D) const` `[inline]`

6.4.2.5 `Gates& qpp::Gates::operator= (const Gates &)` `[delete]`

6.4.2.6 `types::cmat qpp::Gates::Rtheta (double theta) const` `[inline]`

6.4.2.7 `types::cmat qpp::Gates::Xd (size_t D) const` `[inline]`

Here is the call graph for this function:



6.4.2.8 `types::cmat qpp::Gates::Zd (size_t D) const` `[inline]`

Here is the call graph for this function:



6.4.3 Member Data Documentation

6.4.3.1 `types::ket qpp::Gates::b00`

6.4.3.2 `types::ket qpp::Gates::b01`

6.4.3.3 `types::ket qpp::Gates::b10`

6.4.3.4 `types::ket qpp::Gates::b11`

6.4.3.5 `types::cmat qpp::Gates::CNOTab`

6.4.3.6 `types::cmat qpp::Gates::CNOTba`

6.4.3.7 `types::cmat qpp::Gates::CS`

6.4.3.8 `types::cmat qpp::Gates::CZ`

6.4.3.9 `types::cmat qpp::Gates::FRED`

6.4.3.10 `types::cmat qpp::Gates::H`

6.4.3.11 `types::cmat qpp::Gates::Id2`

6.4.3.12 `types::cmat qpp::Gates::pb00`

6.4.3.13 `types::cmat qpp::Gates::pb01`

6.4.3.14 `types::cmat qpp::Gates::pb10`

6.4.3.15 `types::cmat qpp::Gates::pb11`

6.4.3.16 `types::cmat qpp::Gates::px0`

6.4.3.17 `types::cmat qpp::Gates::px1`

6.4.3.18 `types::cmat qpp::Gates::py0`

6.4.3.19 `types::cmat qpp::Gates::py1`

6.4.3.20 `types::cmat qpp::Gates::pz0`

6.4.3.21 `types::cmat qpp::Gates::pz1`

6.4.3.22 `types::cmat qpp::Gates::S`

6.4.3.23 `types::cmat qpp::Gates::SWAP`

6.4.3.24 `types::cmat qpp::Gates::T`

6.4.3.25 `types::cmat qpp::Gates::TOF`

6.4.3.26 `types::cmat qpp::Gates::X`

6.4.3.27 `types::ket qpp::Gates::x0`

6.4.3.28 `types::ket qpp::Gates::x1`

6.4.3.29 `types::cmat qpp::Gates::Y`

6.4.3.30 `types::ket qpp::Gates::y0`

6.4.3.31 `types::ket qpp::Gates::y1`

6.4.3.32 `types::cmat qpp::Gates::Z`

6.4.3.33 `types::ket qpp::Gates::z0`

6.4.3.34 `types::ket qpp::Gates::z1`

The documentation for this class was generated from the following file:

- [include/classes/gates.h](#)

6.5 qpp::NormalDistribution Class Reference

```
#include <stat.h>
```

Public Member Functions

- [NormalDistribution](#) (double *mean*=0, double *sigma*=1)
- double [sample](#) ()

Protected Attributes

- `std::normal_distribution _d`

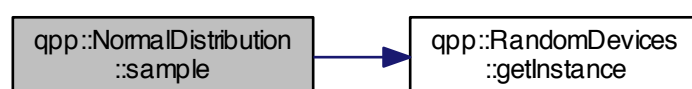
6.5.1 Constructor & Destructor Documentation

6.5.1.1 `qpp::NormalDistribution::NormalDistribution (double mean = 0, double sigma = 1)` `[inline]`

6.5.2 Member Function Documentation

6.5.2.1 `double qpp::NormalDistribution::sample ()` `[inline]`

Here is the call graph for this function:



6.5.3 Member Data Documentation

6.5.3.1 `std::normal_distribution` `qpp::NormalDistribution::_d` `[protected]`

The documentation for this class was generated from the following file:

- `include/classes/stat.h`

6.6 qpp::Qudit Class Reference

```
#include <qudit.h>
```

Public Member Functions

- `Qudit` (`const types::cmat &rho=Gates::getInstance().pz0`)
- `size_t measure` (`const types::cmat &U, bool destructive=false`)
- `size_t measure` (`bool destructive=false`)
- `types::cmat getRho` (`() const`)
- `size_t getD` (`() const`)
- `virtual ~Qudit` (`()=default`)

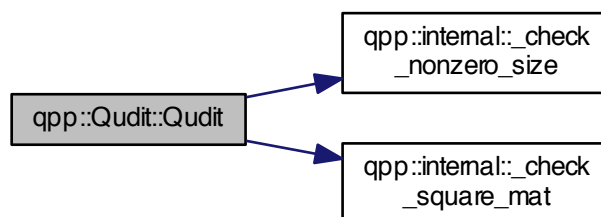
Private Attributes

- `types::cmat _rho`
- `size_t _D`

6.6.1 Constructor & Destructor Documentation

6.6.1.1 `qpp::Qudit::Qudit (const types::cmat & rho = Gates::getInstance().pz0) [inline]`

Here is the call graph for this function:



6.6.1.2 `virtual qpp::Qudit::~~Qudit () [virtual],[default]`

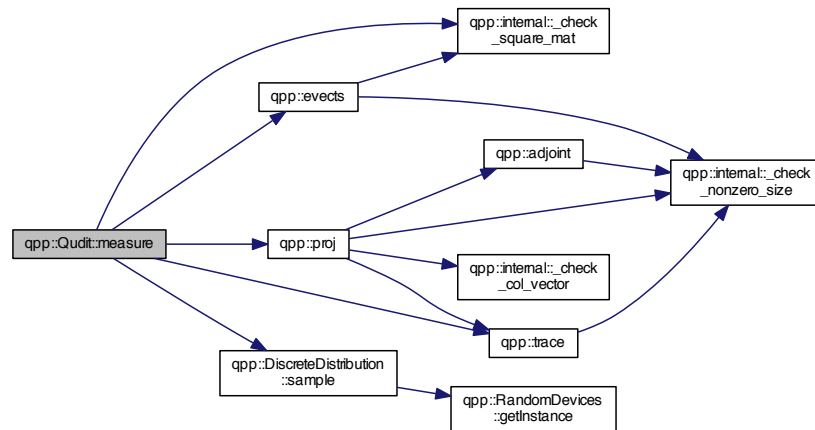
6.6.2 Member Function Documentation

6.6.2.1 `size_t qpp::Qudit::getD () const [inline]`

6.6.2.2 `types::cmat qpp::Qudit::getRho () const [inline]`

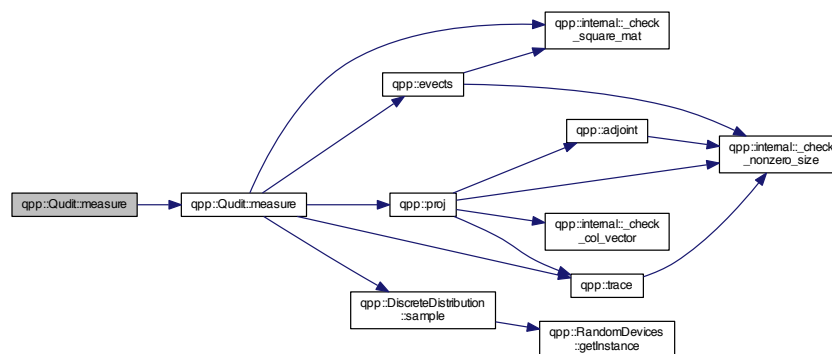
6.6.2.3 `size_t qpp::Qudit::measure (const types::cmat & U, bool destructive = false) [inline]`

Here is the call graph for this function:



6.6.2.4 `size_t qpp::Qudit::measure (bool destructive = false) [inline]`

Here is the call graph for this function:



6.6.3 Member Data Documentation

6.6.3.1 `size_t qpp::Qudit::_D [private]`

6.6.3.2 `types::cmat qpp::Qudit::_rho [private]`

The documentation for this class was generated from the following file:

- [include/classes/qudit.h](#)

6.7 qpp::RandomDevices Class Reference

```
#include <randevs.h>
```

Public Member Functions

- [RandomDevices](#) (const [RandomDevices](#) &)=delete
- [RandomDevices](#) & [operator=](#) (const [RandomDevices](#) &)=delete
- virtual [~RandomDevices](#) ()=default

Static Public Member Functions

- static [RandomDevices](#) & [getInstance](#) ()

Public Attributes

- std::random_device [_rd](#)
- std::mt19937 [_rng](#)

Private Member Functions

- [RandomDevices](#) ()

6.7.1 Constructor & Destructor Documentation

6.7.1.1 [qpp::RandomDevices::RandomDevices](#) () [inline],[private]

6.7.1.2 [qpp::RandomDevices::RandomDevices](#) (const [RandomDevices](#) &) [delete]

6.7.1.3 virtual [qpp::RandomDevices::~~RandomDevices](#) () [virtual],[default]

6.7.2 Member Function Documentation

6.7.2.1 static [RandomDevices&](#) [qpp::RandomDevices::getInstance](#) () [inline],[static]

6.7.2.2 [RandomDevices&](#) [qpp::RandomDevices::operator=](#) (const [RandomDevices](#) &) [delete]

6.7.3 Member Data Documentation

6.7.3.1 std::random_device [qpp::RandomDevices::_rd](#)

6.7.3.2 std::mt19937 [qpp::RandomDevices::_rng](#)

The documentation for this class was generated from the following file:

- include/classes/[randevs.h](#)

6.8 qpp::Timer Class Reference

```
#include <timer.h>
```

Public Member Functions

- [Timer](#) ()
- void [tic](#) ()
- void [toc](#) ()
- double [seconds](#) () const
- virtual [~Timer](#) ()=default

Protected Attributes

- std::chrono::high_resolution_clock::time_point [_start](#)
- std::chrono::high_resolution_clock::time_point [_end](#)

Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [Timer](#) &rhs)

6.8.1 Constructor & Destructor Documentation

6.8.1.1 `qpp::Timer::Timer ()` [[inline](#)]

6.8.1.2 `virtual qpp::Timer::~~Timer ()` [[virtual](#)],[[default](#)]

6.8.2 Member Function Documentation

6.8.2.1 `double qpp::Timer::seconds ()` const [[inline](#)]

6.8.2.2 `void qpp::Timer::tic ()` [[inline](#)]

6.8.2.3 `void qpp::Timer::toc ()` [[inline](#)]

6.8.3 Friends And Related Function Documentation

6.8.3.1 `std::ostream& operator<< (std::ostream & os, const Timer & rhs)` [[friend](#)]

6.8.4 Member Data Documentation

6.8.4.1 `std::chrono::high_resolution_clock::time_point qpp::Timer::_end` [[protected](#)]

6.8.4.2 `std::chrono::high_resolution_clock::time_point qpp::Timer::_start` [[protected](#)]

The documentation for this class was generated from the following file:

- include/classes/[timer.h](#)

6.9 qpp::UniformRealDistribution Class Reference

```
#include <stat.h>
```

Public Member Functions

- [UniformRealDistribution](#) (double a=0, double b=1)
- double [sample](#) ()

Protected Attributes

- `std::uniform_real_distribution _d`

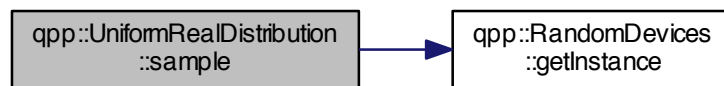
6.9.1 Constructor & Destructor Documentation

6.9.1.1 `qpp::UniformRealDistribution::UniformRealDistribution (double a = 0, double b = 1)` `[inline]`

6.9.2 Member Function Documentation

6.9.2.1 `double qpp::UniformRealDistribution::sample ()` `[inline]`

Here is the call graph for this function:



6.9.3 Member Data Documentation

6.9.3.1 `std::uniform_real_distribution qpp::UniformRealDistribution::_d` `[protected]`

The documentation for this class was generated from the following file:

- `include/classes/stat.h`

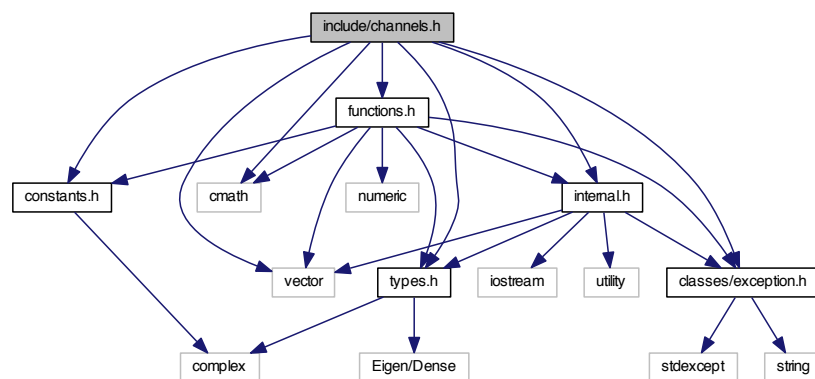
Chapter 7

File Documentation

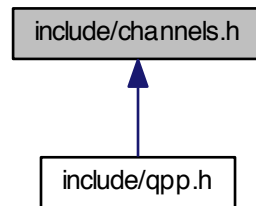
7.1 include/channels.h File Reference

```
#include <cmath>
#include <vector>
#include "constants.h"
#include "functions.h"
#include "internal.h"
#include "types.h"
#include "classes/exception.h"
```

Include dependency graph for channels.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [qpp](#)

Functions

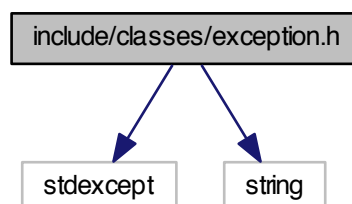
- `types::cmat qpp::channel (const types::cmat &rho, const std::vector< types::cmat > &Ks)`
- `types::cmat qpp::super (const std::vector< types::cmat > &Ks)`
- `types::cmat qpp::choi (const std::vector< types::cmat > &Ks)`
- `std::vector< types::cmat > qpp::choi2kraus (const types::cmat &A)`

7.2 include/classes/exception.h File Reference

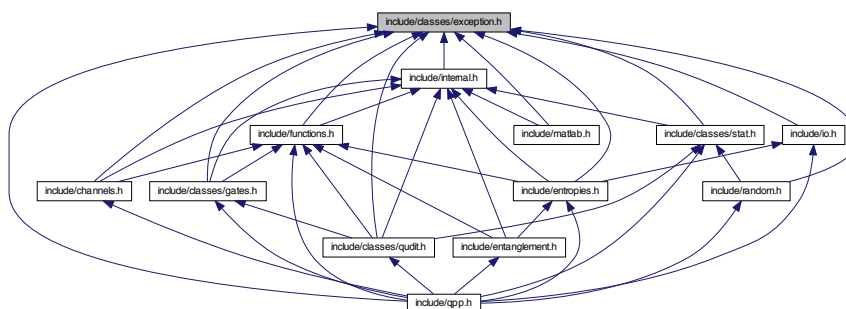
```
#include <stdexcept>
```

```
#include <string>
```

Include dependency graph for exception.h:



This graph shows which files directly or indirectly include this file:



Classes

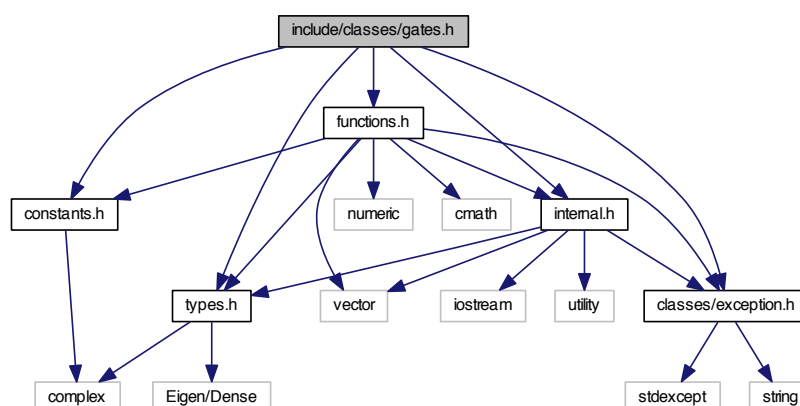
- class [qpp::Exception](#)

Namespaces

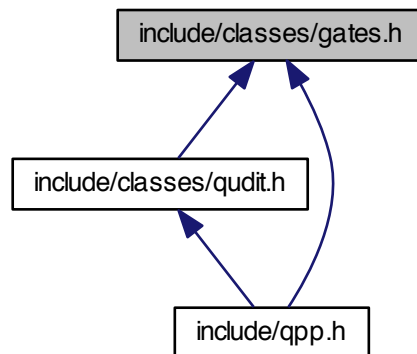
- [qpp](#)

7.3 include/classes/gates.h File Reference

```
#include "constants.h"
#include "functions.h"
#include "exception.h"
#include "internal.h"
#include "types.h"
Include dependency graph for gates.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `qpp::Gates`

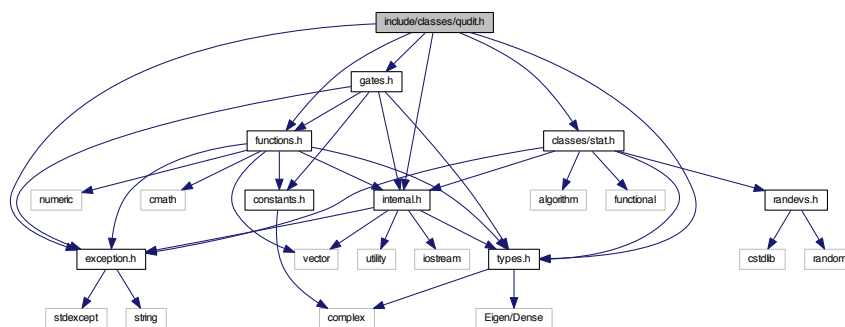
Namespaces

- `qpp`

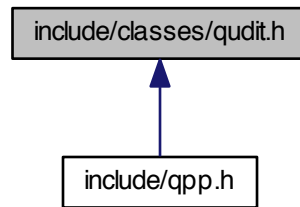
7.4 include/classes/qudit.h File Reference

```
#include "exception.h"
#include "functions.h"
#include "gates.h"
#include "internal.h"
#include "types.h"
#include "classes/stat.h"
```

Include dependency graph for `qudit.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `qpp::Qudit`

Namespaces

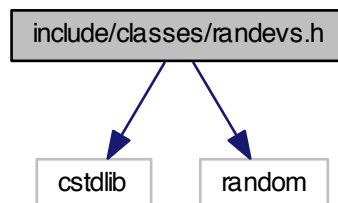
- `qpp`

7.5 include/classes/randevs.h File Reference

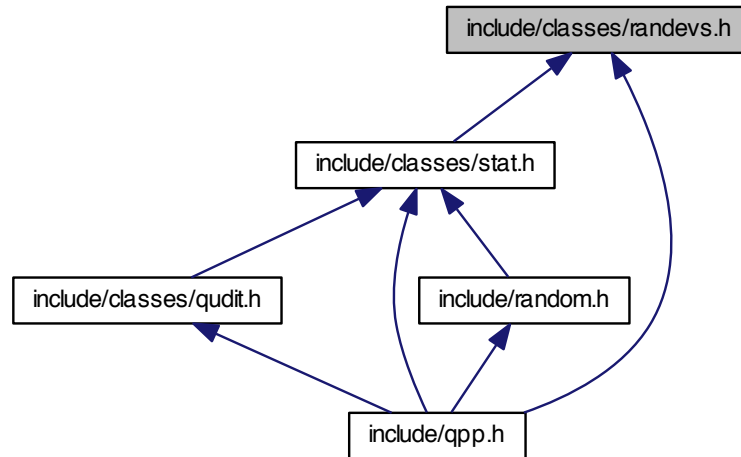
```
#include <cstdlib>
```

```
#include <random>
```

Include dependency graph for randevs.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [qpp::RandomDevices](#)

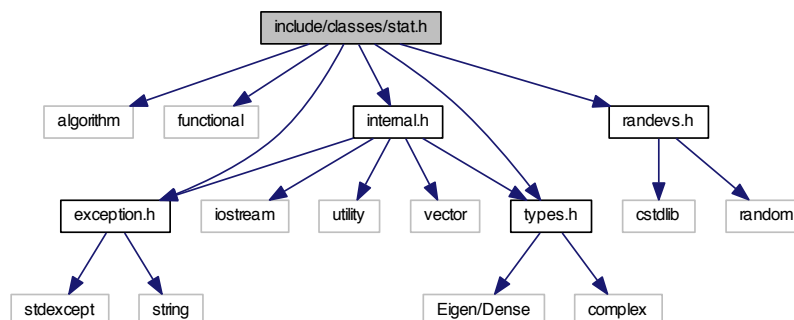
Namespaces

- [qpp](#)

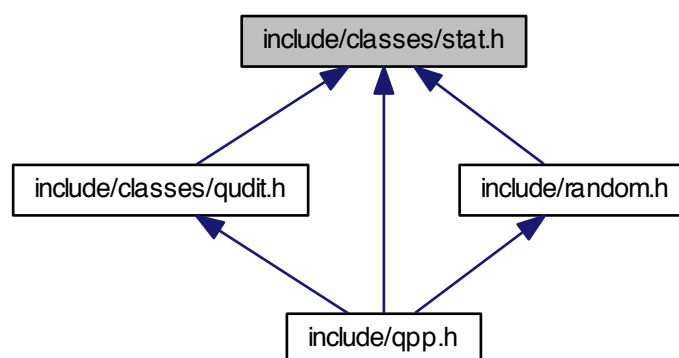
7.6 include/classes/stat.h File Reference

```
#include <algorithm>
#include <functional>
#include "exception.h"
#include "internal.h"
#include "randevs.h"
#include "types.h"
```

Include dependency graph for stat.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `qpp::NormalDistribution`
- class `qpp::UniformRealDistribution`
- class `qpp::DiscreteDistribution`
- class `qpp::DiscreteDistributionAbsSquare`

Namespaces

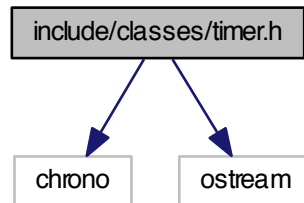
- `qpp`

7.7 include/classes/timer.h File Reference

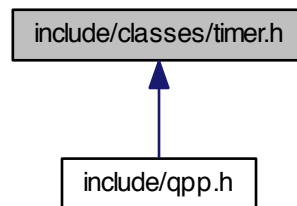
```
#include <chrono>
```

```
#include <ostream>
```

Include dependency graph for timer.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [qpp::Timer](#)

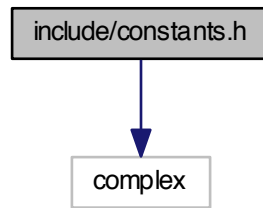
Namespaces

- [qpp](#)

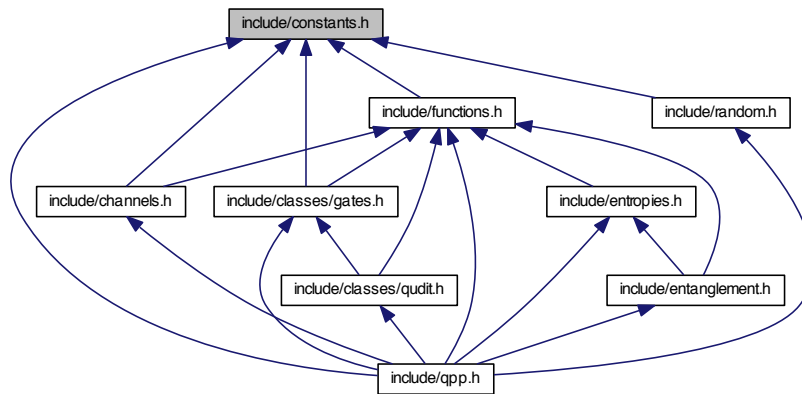
7.8 include/constants.h File Reference

```
#include <complex>
```


Include dependency graph for constants.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [qpp](#)
- [qpp::ct](#)

Functions

- `std::complex< double > qpp::ct::omega (size_t D)`

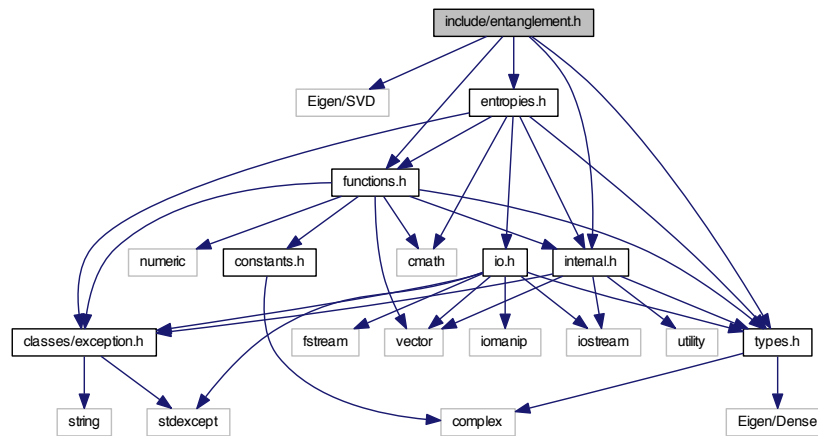
Variables

- `const double qpp::ct::chop = 1e-10`
- `const double qpp::ct::eps = 1e-12`
- `const std::complex< double > qpp::ct::ii = { 0, 1 }`
- `const double qpp::ct::pi = 3.141592653589793238462643383279502884`
- `const double qpp::ct::ee = 2.718281828459045235360287471352662497`

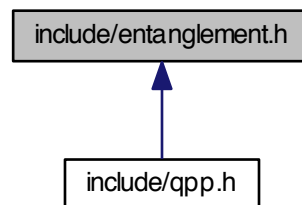
7.9 include/entanglement.h File Reference

```
#include <Eigen/SVD>
#include "entropies.h"
#include "functions.h"
#include "internal.h"
#include "types.h"
```

Include dependency graph for entanglement.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [qpp](#)

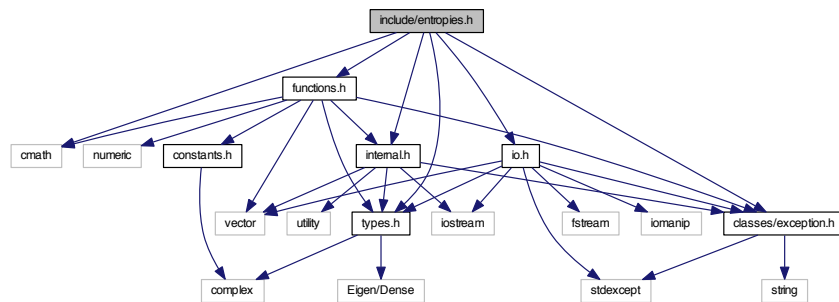
Functions

- `template<typename Derived >`
`types::cmat qpp::schmidtcoeff (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &dims)`
- `template<typename Derived >`
`types::cmat qpp::schmidtU (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &dims)`

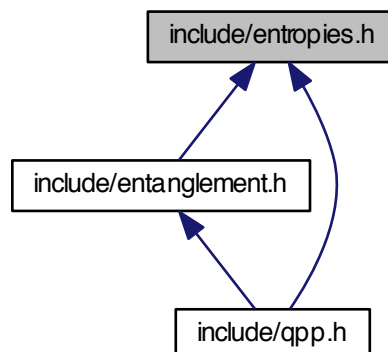
- `template<typename Derived >`
`types::cmat qpp::schmidtV (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &dims)`
- `template<typename Derived >`
`types::cmat qpp::schmidtprob (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &dims)`
- `template<typename Derived >`
`double qpp::entanglement (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &dims)`
- `template<typename Derived >`
`double qpp::gconcurrence (const Eigen::MatrixBase< Derived > &A)`

7.10 include/entropies.h File Reference

```
#include <cmath>
#include "functions.h"
#include "internal.h"
#include "types.h"
#include "classes/exception.h"
#include "io.h"
Include dependency graph for entropies.h:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- [qpp](#)

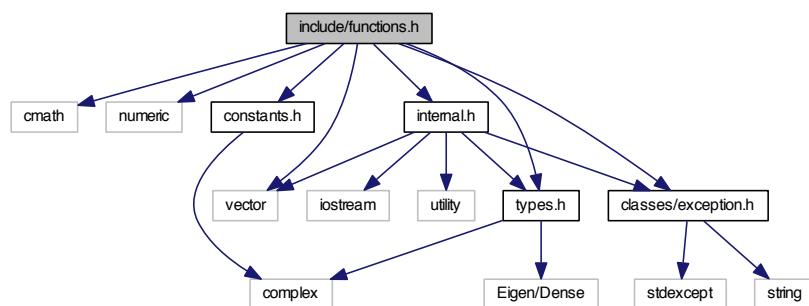
Functions

- `template<typename Derived >`
`double qpp::shannon (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`double qpp::renyi (const double alpha, const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`double qpp::renyi_inf (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`double qpp::tsallis (const double alpha, const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`double qpp::mutualinfo (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &subsys, const std::vector< size_t > &dims)`

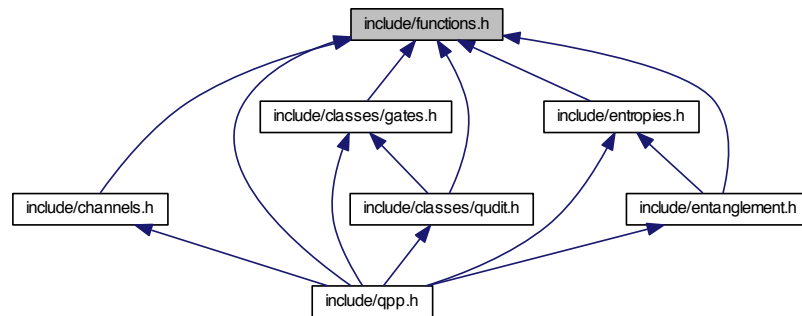
7.11 include/functions.h File Reference

```
#include <cmath>
#include <numeric>
#include <vector>
#include "constants.h"
#include "internal.h"
#include "types.h"
#include "classes/exception.h"
```

Include dependency graph for functions.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [qpp](#)

Functions

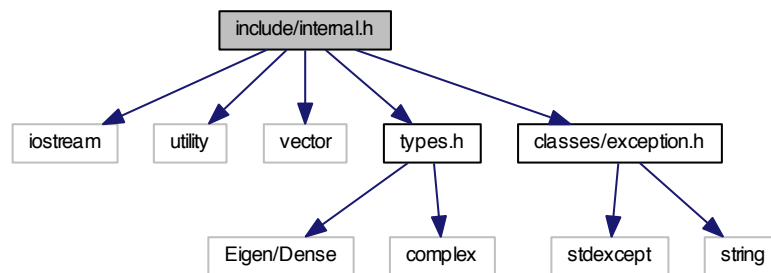
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::transpose (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::conjugate (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::adjoint (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`Derived::Scalar qpp::trace (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`Derived::Scalar qpp::det (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`Derived::Scalar qpp::sum (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`double qpp::norm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat qpp::evals (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat qpp::evecs (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat qpp::hevals (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat qpp::hevecs (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat qpp::funm (const Eigen::MatrixBase< Derived > &A, types::cplx(*f)(const types::cplx &))`
- `template<typename Derived >`
`types::cmat qpp::sqrtm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat qpp::absm (const Eigen::MatrixBase< Derived > &A)`

- `template<typename Derived >`
`types::cmat qpp::expm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat qpp::logm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat qpp::sinm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat qpp::cosm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat qpp::spectralpwm (const Eigen::MatrixBase< Derived > &A, const types::cplx z)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::pwm (const Eigen::MatrixBase< Derived > &A, size_t n)`
- `template<typename OutputScalar , typename Derived >`
`types::DynMat< OutputScalar > qpp::fun (const Eigen::MatrixBase< Derived > &A, OutputScalar(*f)(const`
`typename Derived::Scalar &))`
- `template<typename Derived1 , typename Derived2 >`
`types::DynMat< typename`
`Derived1::Scalar > qpp::kron (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase< De-`
`derived2 > &B)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::kronlist (const std::vector< types::DynMat< typename Derived::Scalar > > &As)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::kronpow (const Eigen::MatrixBase< Derived > &A, size_t n)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::reshape (const Eigen::MatrixBase< Derived > &A, size_t rows, size_t cols)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::syspermute (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t >`
`&perm, const std::vector< size_t > &dims)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::ptrace2 (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t >`
`&dims)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::ptrace (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &sub-`
`sys, const std::vector< size_t > &dims)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::ptranspose (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t >`
`&subsys, const std::vector< size_t > &dims)`
- `template<typename Derived1 , typename Derived2 >`
`types::DynMat< typename`
`Derived1::Scalar > qpp::comm (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase< De-`
`derived2 > &B)`
- `template<typename Derived1 , typename Derived2 >`
`types::DynMat< typename`
`Derived1::Scalar > qpp::anticomm (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase<`
`Derived2 > &B)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::proj (const Eigen::MatrixBase< Derived > &V)`

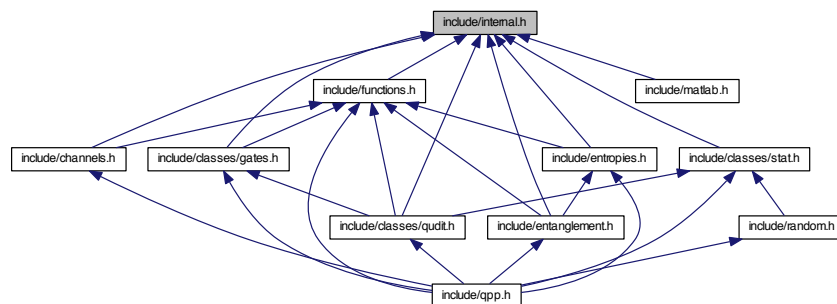
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::expandout (const Eigen::MatrixBase< Derived > &A, size_t pos, const std::vector<`
`size_t > &dims)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::grams (const std::vector< types::DynMat< typename Derived::Scalar > > &Vs)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::grams (const Eigen::MatrixBase< Derived > &A)`
- `std::vector< size_t > qpp::n2multiidx (size_t n, const std::vector< size_t > &dims)`
- `size_t qpp::multiidx2n (const std::vector< size_t > &midx, const std::vector< size_t > &dims)`

7.12 include/internal.h File Reference

```
#include <iostream>
#include <utility>
#include <vector>
#include "types.h"
#include "classes/exception.h"
Include dependency graph for internal.h:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- [qpp](#)
- [qpp::internal](#)

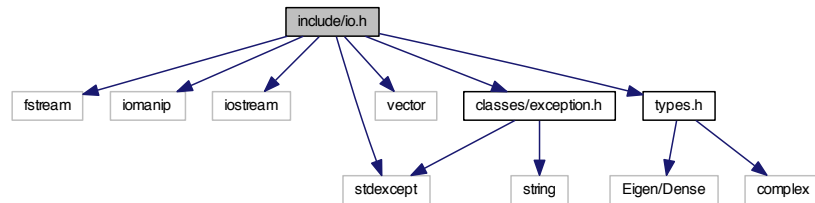
Functions

- void [qpp::internal::_n2multiidx](#) (size_t n, size_t numdims, const size_t *dims, size_t *result)
- size_t [qpp::internal::_multiidx2n](#) (const size_t *midx, size_t numdims, const size_t *dims)
- template<typename Derived >
bool [qpp::internal::_check_square_mat](#) (const Eigen::MatrixBase< Derived > &A)
- template<typename Derived >
bool [qpp::internal::_check_vector](#) (const Eigen::MatrixBase< Derived > &A)
- template<typename Derived >
bool [qpp::internal::_check_row_vector](#) (const Eigen::MatrixBase< Derived > &A)
- template<typename Derived >
bool [qpp::internal::_check_col_vector](#) (const Eigen::MatrixBase< Derived > &A)
- template<typename T >
bool [qpp::internal::_check_nonzero_size](#) (const T &x)
- bool [qpp::internal::_check_dims](#) (const std::vector< size_t > &dims)
- template<typename Derived >
bool [qpp::internal::_check_dims_match_mat](#) (const std::vector< size_t > &dims, const Eigen::MatrixBase< Derived > &A)
- template<typename Derived >
bool [qpp::internal::_check_dims_match_cvect](#) (const std::vector< size_t > &dims, const Eigen::MatrixBase< Derived > &V)
- template<typename Derived >
bool [qpp::internal::_check_dims_match_rvect](#) (const std::vector< size_t > &dims, const Eigen::MatrixBase< Derived > &V)
- bool [qpp::internal::_check_eq_dims](#) (const std::vector< size_t > &dims, size_t dim)
- bool [qpp::internal::_check_subsys](#) (const std::vector< size_t > &subsys, const std::vector< size_t > &dims)
- bool [qpp::internal::_check_perm](#) (const std::vector< size_t > &perm, const std::vector< size_t > &dims)
- template<typename Scalar >
void [qpp::internal::_syspermute_worker](#) (size_t numdims, const size_t *cdims, const size_t *cperm, size_t i, size_t &iperm, const types::DynMat< Scalar > &V, types::DynMat< Scalar > &result)
- template<typename Scalar >
void [qpp::internal::_ptranspose_worker](#) (const size_t *midxcoll, size_t numdims, size_t numsubsys, const size_t *cdims, const size_t *csubsys, size_t i, size_t j, size_t &iperm, size_t &jperm, const types::DynMat< Scalar > &A, types::DynMat< Scalar > &result)

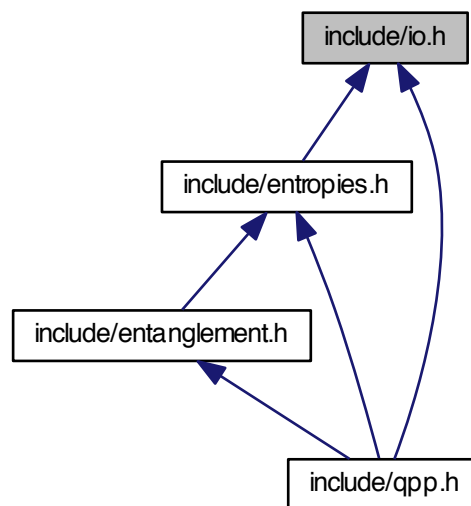
7.13 include/io.h File Reference

```
#include <fstream>
#include <iomanip>
#include <iostream>
#include <stdexcept>
#include <vector>
#include "types.h"
#include "classes/exception.h"
```


Include dependency graph for io.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [qpp](#)

Functions

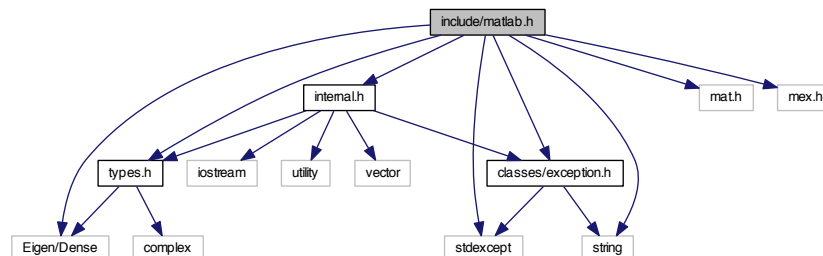
- `template<typename T >`
`void qpp::disp (const T &x, const std::string &separator, const std::string &start="[" , const std::string &end="]", std::ostream &os=std::cout)`
- `template<typename T >`
`void qpp::displn (const T &x, const std::string &separator, const std::string &start="[" , const std::string &end="]", std::ostream &os=std::cout)`
- `template<typename T >`
`void qpp::disp (const T *x, const size_t n, const std::string &separator, const std::string &start="[" , const std::string &end="]", std::ostream &os=std::cout)`

- `template<typename T >`
`void qpp::displn (const T *x, const size_t n, const std::string &separator, const std::string &start="[" , const std::string &end="]", std::ostream &os=std::cout)`
- `template<typename Derived >`
`void qpp::disp (const Eigen::MatrixBase< Derived > &A, double chop=ct::chop, std::ostream &os=std::cout)`
- `template<typename Derived >`
`void qpp::displn (const Eigen::MatrixBase< Derived > &A, double chop=ct::chop, std::ostream &os=std::cout)`
- `void qpp::disp (const types::cplx c, double chop=ct::chop, std::ostream &os=std::cout)`
- `void qpp::displn (const types::cplx c, double chop=ct::chop, std::ostream &os=std::cout)`
- `template<typename Derived >`
`void qpp::save (const Eigen::MatrixBase< Derived > &A, const std::string &fname)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::load (const std::string &fname)`

7.14 include/matlab.h File Reference

```
#include <Eigen/Dense>
#include <stdexcept>
#include <string>
#include "internal.h"
#include "types.h"
#include "classes/exception.h"
#include "mat.h"
#include "mex.h"
```

Include dependency graph for matlab.h:



Namespaces

- [qpp](#)

Functions

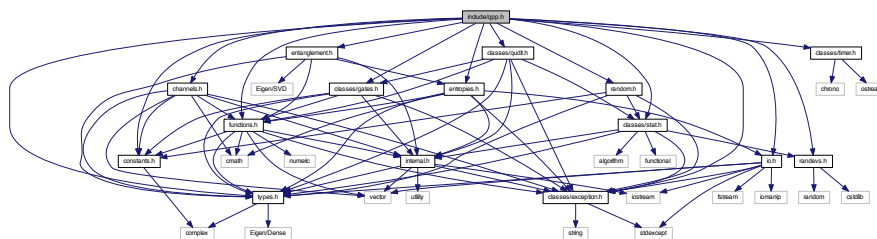
- `template<typename Derived >`
`Derived qpp::loadMATLABmatrix (const std::string &mat_file, const std::string &var_name)`
- `template<>`
`types::dmat qpp::loadMATLABmatrix (const std::string &mat_file, const std::string &var_name)`
- `template<>`
`types::cmat qpp::loadMATLABmatrix (const std::string &mat_file, const std::string &var_name)`
- `template<typename Derived >`
`void qpp::saveMATLABmatrix (const Eigen::MatrixBase< Derived > &A, const std::string &mat_file, const std::string &var_name, const std::string &mode)`

- `template<>`
`void qpp::saveMATLABmatrix (const Eigen::MatrixBase< typename types::dmat > &A, const std::string &mat_file, const std::string &var_name, const std::string &mode)`
- `template<>`
`void qpp::saveMATLABmatrix (const Eigen::MatrixBase< typename types::cmat > &A, const std::string &mat_file, const std::string &var_name, const std::string &mode)`

7.15 include/qpp.h File Reference

```
#include "channels.h"
#include "constants.h"
#include "entanglement.h"
#include "entropies.h"
#include "functions.h"
#include "io.h"
#include "random.h"
#include "types.h"
#include "classes/exception.h"
#include "classes/gates.h"
#include "classes/qudit.h"
#include "classes/randevs.h"
#include "classes/stat.h"
#include "classes/timer.h"
```

Include dependency graph for qpp.h:



Namespaces

- [qpp](#)

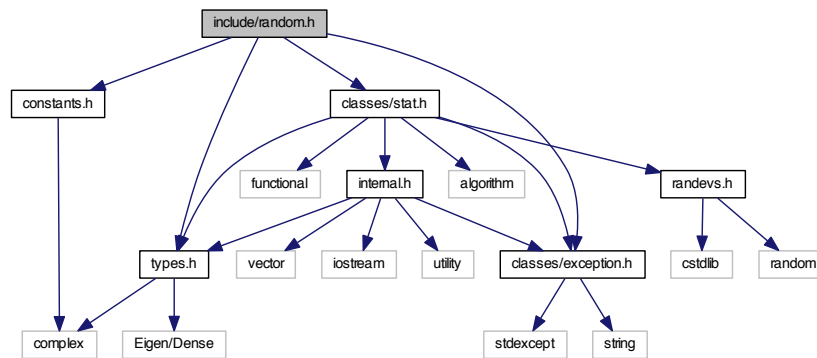
Variables

- `RandomDevices` & [qpp::rdevs](#) = `RandomDevices::getInstance()`
- `const Gates` & [qpp::gt](#) = `Gates::getInstance()`

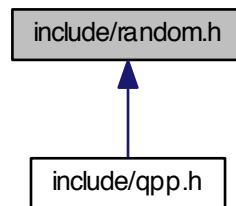
7.16 include/random.h File Reference

```
#include "constants.h"
#include "types.h"
#include "classes/exception.h"
#include "classes/stat.h"
```

Include dependency graph for random.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [qpp](#)

Functions

- `template<typename Derived >`
Derived [qpp::rand](#) (size_t rows, size_t cols, double a=0, double b=1)
- `template<>`
`types::dmat` [qpp::rand](#) (size_t rows, size_t cols, double a, double b)
- `template<>`
`types::cmat` [qpp::rand](#) (size_t rows, size_t cols, double a, double b)
- `double` [qpp::rand](#) (double a=0, double b=1)
- `template<typename Derived >`
Derived [qpp::randn](#) (size_t rows, size_t cols, double mean=0, double sigma=1)
- `template<>`
`types::dmat` [qpp::randn](#) (size_t rows, size_t cols, double mean, double sigma)
- `template<>`
`types::cmat` [qpp::randn](#) (size_t rows, size_t cols, double mean, double sigma)
- `double` [qpp::randn](#) (double mean=0, double sigma=1)

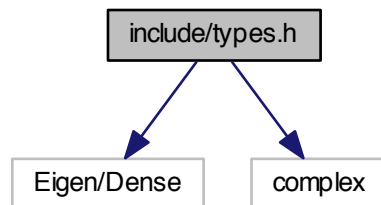
- `types::cmat` [qpp::randU](#) (size_t D)
- `types::cmat` [qpp::randV](#) (size_t Din, size_t Dout)
- `std::vector< types::cmat >` [qpp::randkraus](#) (size_t n, size_t D)
- `types::cmat` [qpp::randH](#) (size_t D)
- `types::ket` [qpp::randket](#) (size_t D)
- `types::cmat` [qpp::randrho](#) (size_t D)

7.17 include/types.h File Reference

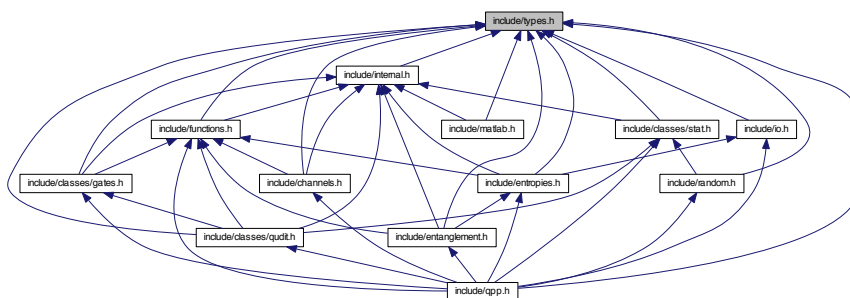
```
#include <Eigen/Dense>
```

```
#include <complex>
```

Include dependency graph for types.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [qpp](#)
- [qpp::types](#)

Typedefs

- `typedef std::complex< double >` [qpp::types::cplx](#)
- `typedef Eigen::MatrixXcd` [qpp::types::cmat](#)
- `typedef Eigen::MatrixXd` [qpp::types::dmat](#)

- typedef Eigen::MatrixXf [qpp::types::fmat](#)
- typedef Eigen::MatrixXi [qpp::types::imat](#)
- typedef Eigen::Matrix< cplx, Eigen::Dynamic, 1 > [qpp::types::ket](#)
- typedef Eigen::Matrix< cplx, 1, Eigen::Dynamic > [qpp::types::bra](#)
- template<typename Scalar >
using [qpp::types::DynMat](#) = Eigen::Matrix< Scalar, Eigen::Dynamic, Eigen::Dynamic >