

qpp
0.1

Generated by Doxygen 1.8.5

Thu May 1 2014 16:56:43

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	14
5.1.1.2	adjoint	14
5.1.1.3	anticomm	15
5.1.1.4	channel	15
5.1.1.5	channel	16
5.1.1.6	choi	16
5.1.1.7	choi2kraus	17
5.1.1.8	comm	17
5.1.1.9	compperm	18
5.1.1.10	conjugate	18
5.1.1.11	cosm	18
5.1.1.12	cwise	19
5.1.1.13	det	19
5.1.1.14	disp	19
5.1.1.15	disp	19
5.1.1.16	disp	19
5.1.1.17	disp	19
5.1.1.18	displn	20

5.1.1.19	displn	20
5.1.1.20	displn	20
5.1.1.21	displn	21
5.1.1.22	entanglement	21
5.1.1.23	evals	22
5.1.1.24	evecs	22
5.1.1.25	expandout	23
5.1.1.26	expm	23
5.1.1.27	funm	23
5.1.1.28	gate	25
5.1.1.29	gconcurrence	26
5.1.1.30	grams	26
5.1.1.31	grams	27
5.1.1.32	grams	27
5.1.1.33	hevals	27
5.1.1.34	hevecs	28
5.1.1.35	inverse	28
5.1.1.36	invperm	28
5.1.1.37	kron	28
5.1.1.38	kron	29
5.1.1.39	kron	29
5.1.1.40	kron	29
5.1.1.41	kronpow	30
5.1.1.42	load	30
5.1.1.43	loadMATLABmatrix	30
5.1.1.44	loadMATLABmatrix	30
5.1.1.45	loadMATLABmatrix	30
5.1.1.46	logdet	30
5.1.1.47	logm	31
5.1.1.48	mket	31
5.1.1.49	mket	31
5.1.1.50	mket	32
5.1.1.51	multiidx2n	32
5.1.1.52	n2multiidx	32
5.1.1.53	norm	33
5.1.1.54	powm	33
5.1.1.55	prj	34
5.1.1.56	ptrace	35
5.1.1.57	ptrace1	36
5.1.1.58	ptrace2	36

5.1.1.59	ptranspose	37
5.1.1.60	qmutualinfo	38
5.1.1.61	rand	38
5.1.1.62	rand	38
5.1.1.63	rand	38
5.1.1.64	rand	39
5.1.1.65	randH	39
5.1.1.66	randint	39
5.1.1.67	randket	40
5.1.1.68	randkraus	40
5.1.1.69	randn	40
5.1.1.70	randn	40
5.1.1.71	randn	40
5.1.1.72	randn	41
5.1.1.73	randperm	41
5.1.1.74	randrho	41
5.1.1.75	randU	41
5.1.1.76	randV	42
5.1.1.77	renyi	42
5.1.1.78	renyi_inf	42
5.1.1.79	reshape	43
5.1.1.80	save	43
5.1.1.81	saveMATLABmatrix	43
5.1.1.82	saveMATLABmatrix	43
5.1.1.83	saveMATLABmatrix	44
5.1.1.84	schmidtcoeff	44
5.1.1.85	schmidtprob	45
5.1.1.86	schmidtU	45
5.1.1.87	schmidtV	46
5.1.1.88	shannon	46
5.1.1.89	sinm	47
5.1.1.90	spectralpowm	47
5.1.1.91	sqrtm	47
5.1.1.92	sum	48
5.1.1.93	super	48
5.1.1.94	syspermute	49
5.1.1.95	trace	49
5.1.1.96	transpose	50
5.1.1.97	tsallis	50
5.1.2	Variable Documentation	50

5.1.2.1	gt	50
5.1.2.2	rdevs	50
5.1.2.3	st	50
5.2	qpp::ct Namespace Reference	50
5.2.1	Function Documentation	51
5.2.1.1	omega	51
5.2.2	Variable Documentation	51
5.2.2.1	chop	51
5.2.2.2	ee	51
5.2.2.3	eps	51
5.2.2.4	ii	51
5.2.2.5	maxn	51
5.2.2.6	pi	51
5.3	qpp::internal Namespace Reference	51
5.3.1	Function Documentation	52
5.3.1.1	_check_col_vector	52
5.3.1.2	_check_dims	52
5.3.1.3	_check_dims_match_cvect	52
5.3.1.4	_check_dims_match_mat	52
5.3.1.5	_check_dims_match_rvect	52
5.3.1.6	_check_eq_dims	52
5.3.1.7	_check_nonzero_size	52
5.3.1.8	_check_perm	52
5.3.1.9	_check_row_vector	52
5.3.1.10	_check_square_mat	52
5.3.1.11	_check_subsys_match_dims	52
5.3.1.12	_check_vector	52
5.3.1.13	_kron2	52
5.3.1.14	_multiidx2n	53
5.3.1.15	_n2multiidx	53
5.3.1.16	variadic_vector_emplace	53
5.3.1.17	variadic_vector_emplace	53
5.4	qpp::types Namespace Reference	53
5.4.1	Typedef Documentation	53
5.4.1.1	bra	53
5.4.1.2	cmat	53
5.4.1.3	cplx	53
5.4.1.4	dmat	53
5.4.1.5	DynMat	53
5.4.1.6	ket	53

6	Class Documentation	55
6.1	qpp::DiscreteDistribution Class Reference	55
6.1.1	Constructor & Destructor Documentation	55
6.1.1.1	DiscreteDistribution	55
6.1.1.2	DiscreteDistribution	55
6.1.1.3	DiscreteDistribution	55
6.1.2	Member Function Documentation	55
6.1.2.1	probabilities	55
6.1.2.2	sample	56
6.1.3	Member Data Documentation	56
6.1.3.1	_d	56
6.2	qpp::DiscreteDistributionAbsSquare Class Reference	56
6.2.1	Constructor & Destructor Documentation	56
6.2.1.1	DiscreteDistributionAbsSquare	57
6.2.1.2	DiscreteDistributionAbsSquare	57
6.2.1.3	DiscreteDistributionAbsSquare	57
6.2.1.4	DiscreteDistributionAbsSquare	58
6.2.2	Member Function Documentation	58
6.2.2.1	cplx2weights	58
6.2.2.2	probabilities	58
6.2.2.3	sample	58
6.2.3	Member Data Documentation	58
6.2.3.1	_d	58
6.3	qpp::Exception Class Reference	58
6.3.1	Member Enumeration Documentation	60
6.3.1.1	Type	60
6.3.2	Constructor & Destructor Documentation	60
6.3.2.1	Exception	61
6.3.2.2	Exception	61
6.3.2.3	~Exception	61
6.3.3	Member Function Documentation	61
6.3.3.1	_construct_exception_msg	61
6.3.3.2	what	61
6.3.4	Member Data Documentation	61
6.3.4.1	_custom	61
6.3.4.2	_msg	61
6.3.4.3	_type	61
6.3.4.4	_where	61
6.4	qpp::Gates Class Reference	62
6.4.1	Constructor & Destructor Documentation	62

6.4.1.1	Gates	62
6.4.1.2	Gates	62
6.4.1.3	~Gates	62
6.4.2	Member Function Documentation	62
6.4.2.1	CTRL	63
6.4.2.2	Fd	63
6.4.2.3	getInstance	63
6.4.2.4	Id	63
6.4.2.5	operator=	63
6.4.2.6	Rtheta	63
6.4.2.7	Xd	64
6.4.2.8	Zd	64
6.4.3	Member Data Documentation	64
6.4.3.1	CNOTab	64
6.4.3.2	CNOTba	64
6.4.3.3	CZ	64
6.4.3.4	FRED	64
6.4.3.5	H	64
6.4.3.6	Id2	64
6.4.3.7	S	64
6.4.3.8	SWAP	64
6.4.3.9	T	64
6.4.3.10	TOF	64
6.4.3.11	X	64
6.4.3.12	Y	65
6.4.3.13	Z	65
6.5	qpp::NormalDistribution Class Reference	65
6.5.1	Constructor & Destructor Documentation	65
6.5.1.1	NormalDistribution	65
6.5.2	Member Function Documentation	65
6.5.2.1	sample	65
6.5.3	Member Data Documentation	65
6.5.3.1	_d	65
6.6	qpp::Qudit Class Reference	66
6.6.1	Constructor & Destructor Documentation	66
6.6.1.1	Qudit	66
6.6.1.2	~Qudit	66
6.6.2	Member Function Documentation	66
6.6.2.1	getD	66
6.6.2.2	getRho	66

6.6.2.3	measure	67
6.6.2.4	measure	67
6.6.3	Member Data Documentation	67
6.6.3.1	_D	67
6.6.3.2	_rho	67
6.7	qpp::RandomDevices Class Reference	68
6.7.1	Constructor & Destructor Documentation	68
6.7.1.1	RandomDevices	68
6.7.1.2	RandomDevices	68
6.7.1.3	~RandomDevices	68
6.7.2	Member Function Documentation	68
6.7.2.1	getInstance	68
6.7.2.2	operator=	68
6.7.3	Member Data Documentation	68
6.7.3.1	_rd	68
6.7.3.2	_rng	68
6.8	qpp::States Class Reference	68
6.8.1	Constructor & Destructor Documentation	69
6.8.1.1	States	69
6.8.1.2	States	69
6.8.1.3	~States	69
6.8.2	Member Function Documentation	69
6.8.2.1	getInstance	69
6.8.2.2	operator=	70
6.8.3	Member Data Documentation	70
6.8.3.1	b00	70
6.8.3.2	b01	70
6.8.3.3	b10	70
6.8.3.4	b11	70
6.8.3.5	GHZ	70
6.8.3.6	pb00	70
6.8.3.7	pb01	70
6.8.3.8	pb10	70
6.8.3.9	pb11	70
6.8.3.10	pGHZ	70
6.8.3.11	pW	70
6.8.3.12	px0	70
6.8.3.13	px1	70
6.8.3.14	py0	70
6.8.3.15	py1	70

6.8.3.16	pz0	70
6.8.3.17	pz1	70
6.8.3.18	W	70
6.8.3.19	x0	70
6.8.3.20	x1	70
6.8.3.21	y0	70
6.8.3.22	y1	70
6.8.3.23	z0	70
6.8.3.24	z1	70
6.9	qpp::Timer Class Reference	71
6.9.1	Constructor & Destructor Documentation	71
6.9.1.1	Timer	71
6.9.1.2	~Timer	71
6.9.2	Member Function Documentation	71
6.9.2.1	seconds	71
6.9.2.2	tic	71
6.9.2.3	toc	71
6.9.3	Friends And Related Function Documentation	71
6.9.3.1	operator<<	71
6.9.4	Member Data Documentation	71
6.9.4.1	_end	71
6.9.4.2	_start	71
6.10	qpp::UniformRealDistribution Class Reference	71
6.10.1	Constructor & Destructor Documentation	72
6.10.1.1	UniformRealDistribution	72
6.10.2	Member Function Documentation	72
6.10.2.1	sample	72
6.10.3	Member Data Documentation	72
6.10.3.1	_d	72
7	File Documentation	73
7.1	include/channels.h File Reference	73
7.2	include/classes/exception.h File Reference	74
7.3	include/classes/gates.h File Reference	75
7.4	include/classes/qudit.h File Reference	76
7.5	include/classes/randevs.h File Reference	77
7.6	include/classes/stat.h File Reference	78
7.7	include/classes/states.h File Reference	79
7.8	include/classes/timer.h File Reference	80
7.9	include/constants.h File Reference	81

7.10	include/entanglement.h File Reference	83
7.11	include/entropies.h File Reference	84
7.12	include/functions.h File Reference	85
7.13	include/internal.h File Reference	88
7.14	include/io.h File Reference	90
7.15	include/matlab.h File Reference	92
7.16	include/qpp.h File Reference	92
7.17	include/random.h File Reference	93
7.18	include/types.h File Reference	95

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

qpp	9
qpp::ct	50
qpp::internal	51
qpp::types	53

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

qpp::DiscreteDistribution	55
qpp::DiscreteDistributionAbsSquare	56
exception	
qpp::Exception	58
qpp::Gates	62
qpp::NormalDistribution	65
qpp::Qudit	66
qpp::RandomDevices	68
qpp::States	68
qpp::Timer	71
qpp::UniformRealDistribution	71

Chapter 3

Class Index

3.1 Class List

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

qpp::DiscreteDistribution	55
qpp::DiscreteDistributionAbsSquare	56
qpp::Exception	58
qpp::Gates	62
qpp::NormalDistribution	65
qpp::Qudit	66
qpp::RandomDevices	68
qpp::States	68
qpp::Timer	71
qpp::UniformRealDistribution	71

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

include/channels.h	73
include/constants.h	81
include/entanglement.h	83
include/entropies.h	84
include/functions.h	85
include/internal.h	88
include/io.h	90
include/matlab.h	92
include/qpp.h	92
include/random.h	93
include/types.h	95
include/classes/exception.h	74
include/classes/gates.h	75
include/classes/qudit.h	76
include/classes/randevs.h	77
include/classes/stat.h	78
include/classes/states.h	79
include/classes/timer.h	80

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 [States](#)
- class [Timer](#)

Functions

- [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 channel](#) (const Eigen::MatrixBase< Derived > &rho, const std::vector< [types::cmat](#) > &Ks)
- template<typename Derived >
[types::cmat channel](#) (const Eigen::MatrixBase< Derived > &rho, const std::vector< [types::cmat](#) > &Ks,
const std::vector< size_t > &subsys, const std::vector< size_t > &dims)
- 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 qmutualinfo` (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 >`
`types::DynMat< typename`
`Derived::Scalar > inverse` (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 logdet` (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::dmat 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 > cwise` (const Eigen::MatrixBase< Derived > &A, OutputScalar (*)(const
typename Derived::Scalar &))
- `template<typename T >`
`types::DynMat< typename T::Scalar > kron` (const T &head)
- `template<typename T, typename... Args>`
`types::DynMat< typename T::Scalar > kron` (const T &head, const Args &...tail)
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > kron` (const std::vector< Derived > &As)
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > kron` (const std::initializer_list< Derived > &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 > ptrace1` (const Eigen::MatrixBase< Derived > &A, 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 > &subs, 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 > prj` (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 Derived1 , typename Derived2 >`
`types::DynMat< typename`
`Derived1::Scalar > gate` (const Eigen::MatrixBase< Derived1 > &state, const Eigen::MatrixBase< Derived2 > &A, const std::vector< size_t > &subsys, const std::vector< size_t > &dims)
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > grams` (const std::vector< Derived > &Vs)
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > grams` (const std::initializer_list< Derived > &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)
- `types::ket mket` (const std::vector< size_t > &mask)
- `types::ket mket` (const std::vector< size_t > &mask, const std::vector< size_t > &dims)
- `types::ket mket` (const std::vector< size_t > &mask, size_t d)
- `std::vector< size_t > invperm` (const std::vector< size_t > &perm)
- `std::vector< size_t > compperm` (const std::vector< size_t > &perm, const std::vector< size_t > &sigma)
- `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)
- long long `randint` (long long a, long long b)
- `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)
- `std::vector< size_t > randperm` (size_t n)

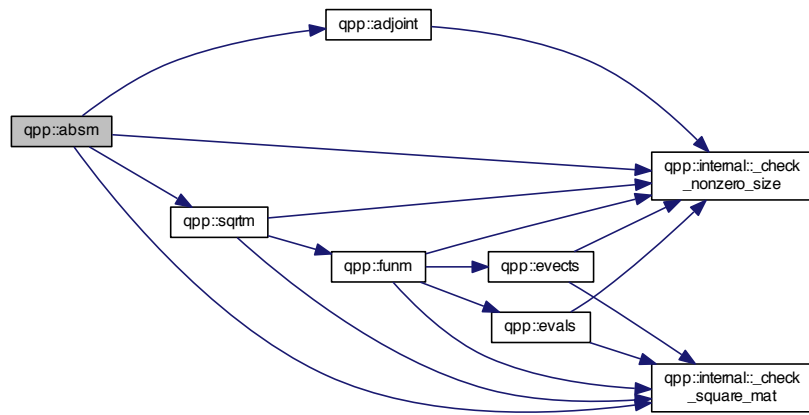
Variables

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

5.1.1 Function Documentation

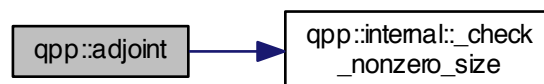
5.1.1.1 `template<typename Derived > types::cmat qpp::absm (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



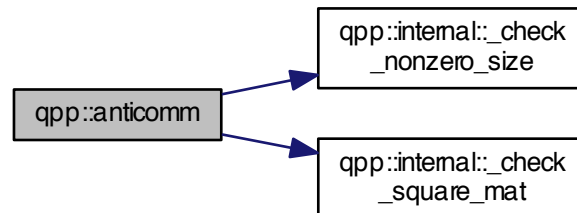
5.1.1.2 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::adjoint (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



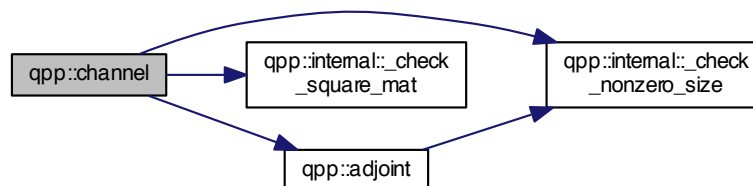
5.1.1.3 `template<typename Derived1 , typename Derived2 > types::DynMat<typename Derived1::Scalar> qpp::anticomm (const Eigen::MatrixBase< Derived1 > & A, const Eigen::MatrixBase< Derived2 > & B)`

Here is the call graph for this function:



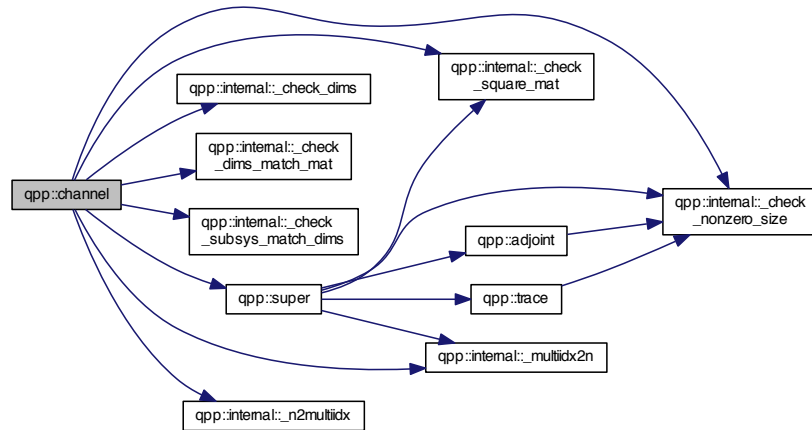
5.1.1.4 `template<typename Derived > types::cmat qpp::channel (const Eigen::MatrixBase< Derived > & rho, const std::vector< types::cmat > & Ks)`

Here is the call graph for this function:



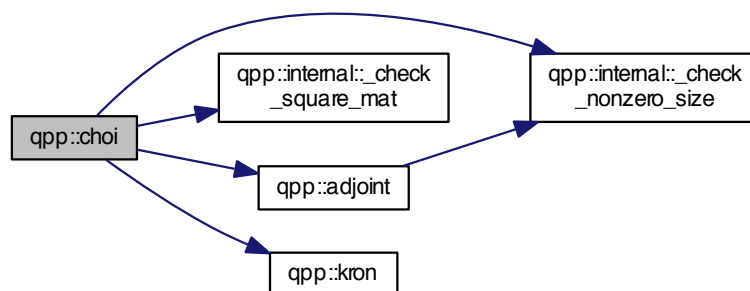
5.1.1.5 `template<typename Derived > types::cmat qpp::channel (const Eigen::MatrixBase< Derived > & rho, const std::vector< types::cmat > & Ks, const std::vector< size_t > & subsys, const std::vector< size_t > & dims)`

Here is the call graph for this function:



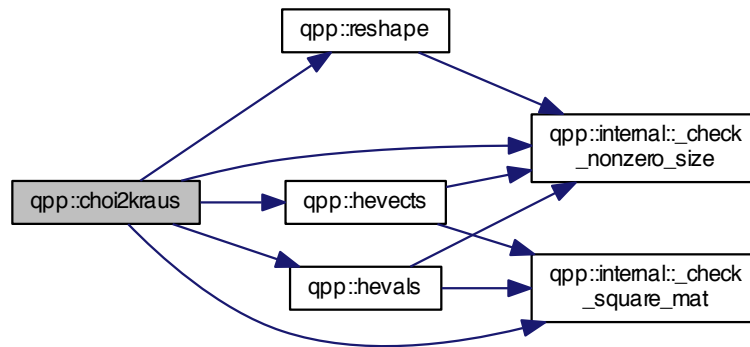
5.1.1.6 `types::cmat qpp::choi (const std::vector< types::cmat > & Ks)`

Here is the call graph for this function:



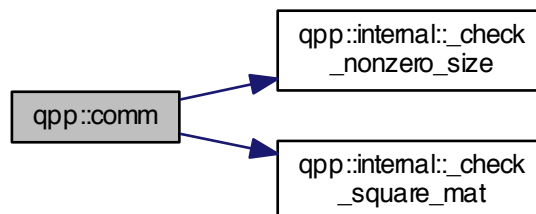
5.1.1.7 `std::vector<types::cmat> qpp::choi2kraus (const types::cmat & A)`

Here is the call graph for this function:



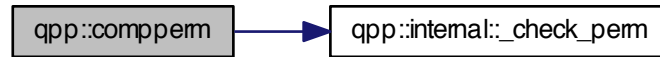
5.1.1.8 `template<typename Derived1, typename Derived2> types::DynMat<typename Derived1::Scalar> qpp::comm (const Eigen::MatrixBase< Derived1 > & A, const Eigen::MatrixBase< Derived2 > & B)`

Here is the call graph for this function:



5.1.1.9 `std::vector<size_t> qpp::compperm (const std::vector< size_t > & perm, const std::vector< size_t > & sigma)`

Here is the call graph for this function:



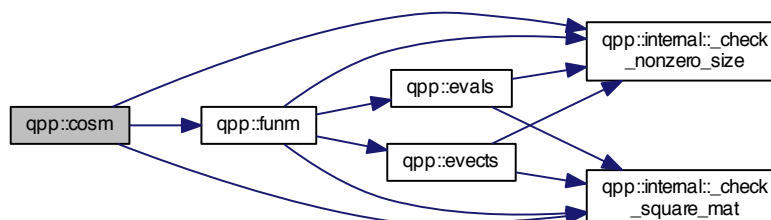
5.1.1.10 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::conjugate (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



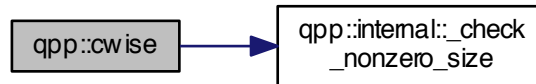
5.1.1.11 `template<typename Derived > types::cmat qpp::cosm (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



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

Here is the call graph for this function:



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

Here is the call graph for this function:



5.1.1.14 `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)`

5.1.1.15 `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)`

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

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

Here is the call graph for this function:



5.1.1.18 `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:



5.1.1.19 `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:



5.1.1.20 `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:



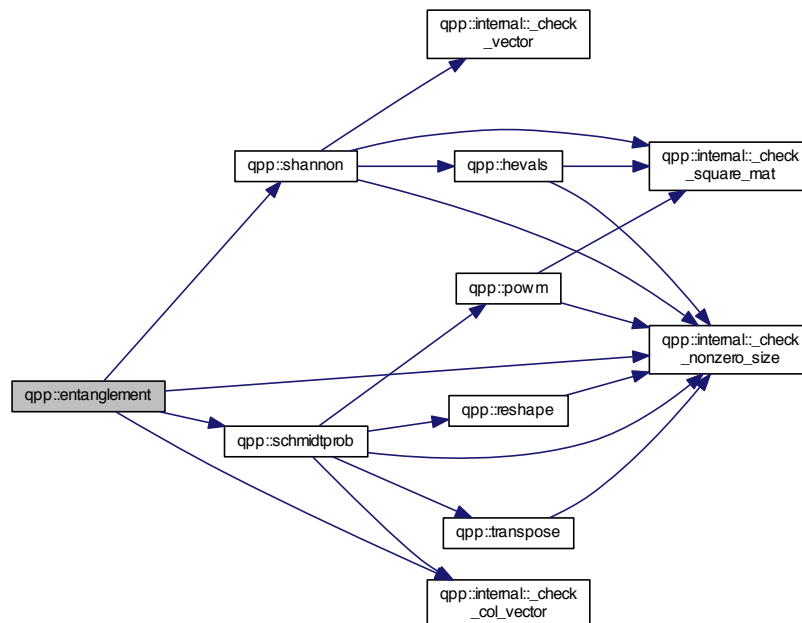
5.1.1.21 `void qpp::displn (const types::cplx c, double chop = ct:::chop, std::ostream & os = std:::cout)`

Here is the call graph for this function:



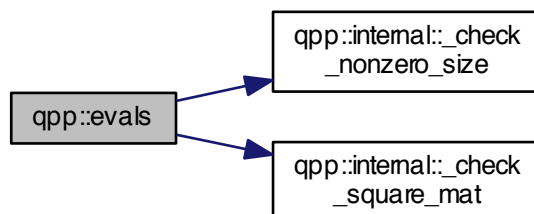
5.1.1.22 `template<typename Derived> double qpp::entanglement (const Eigen::MatrixBase< Derived > & A, const std::vector< size_t > & dims)`

Here is the call graph for this function:



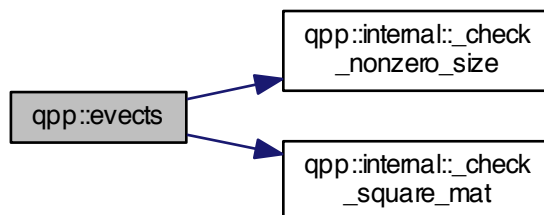
5.1.1.23 `template<typename Derived > types::cmat qpp::evals (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



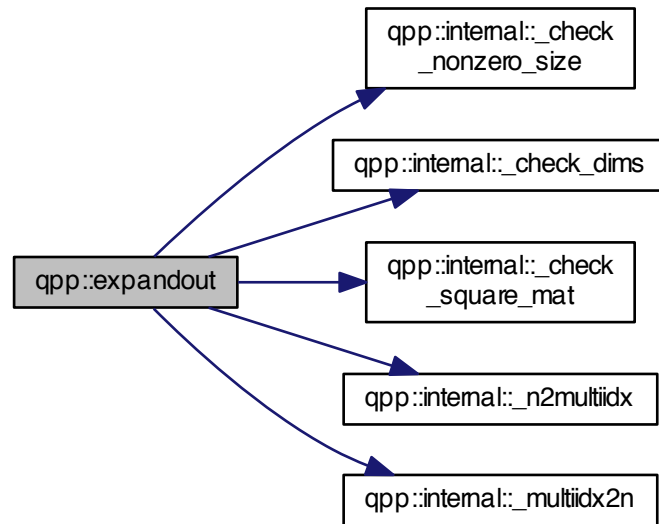
5.1.1.24 `template<typename Derived > types::cmat qpp::evecs (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



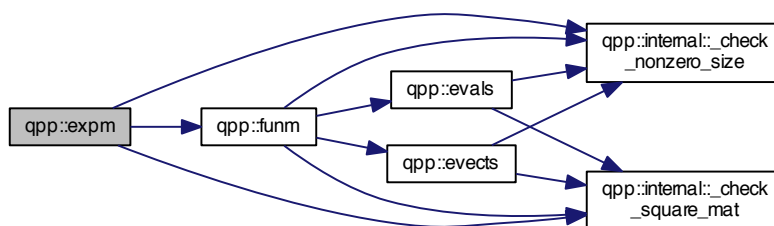
5.1.1.25 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::expandout (const Eigen::MatrixBase< Derived> & A, size_t pos, const std::vector< size_t> & dims)`

Here is the call graph for this function:



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

Here is the call graph for this function:



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

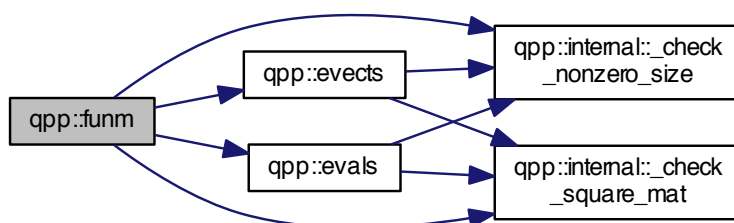
Parameters

A	input matrix
f	function pointer

Returns

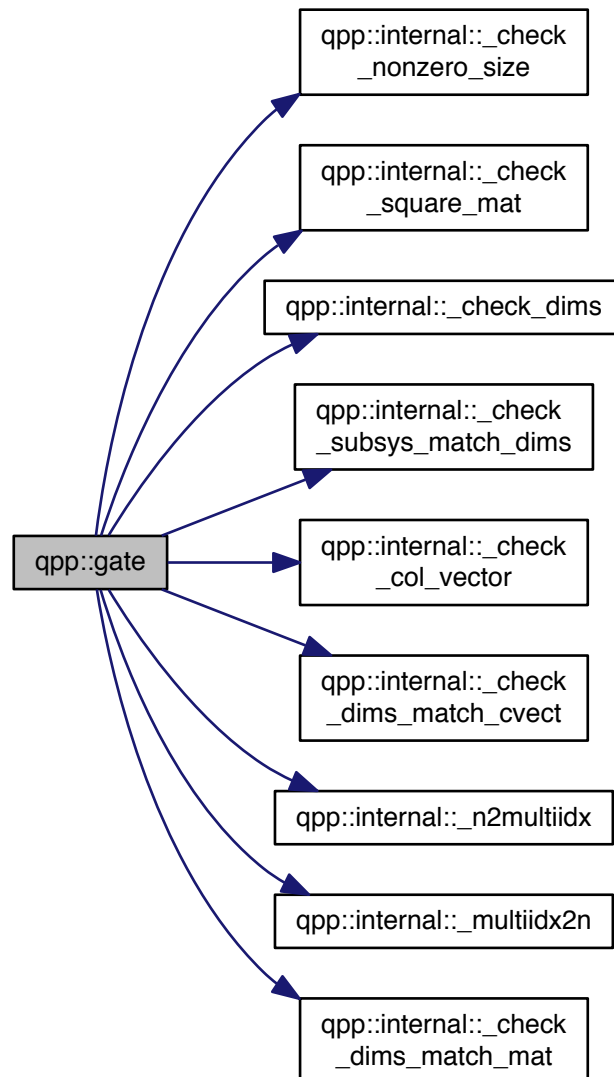
[types::cmat](#)

Here is the call graph for this function:



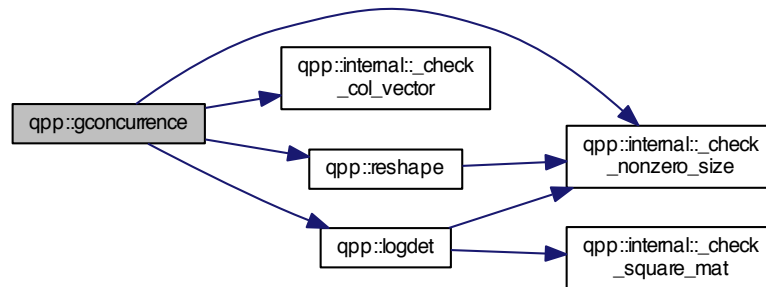
5.1.1.28 `template<typename Derived1 , typename Derived2 > types::DynMat<typename Derived1::Scalar> qpp::gate (const Eigen::MatrixBase< Derived1 > & state, const Eigen::MatrixBase< Derived2 > & A, const std::vector< size_t > & subsys, const std::vector< size_t > & dims)`

Here is the call graph for this function:



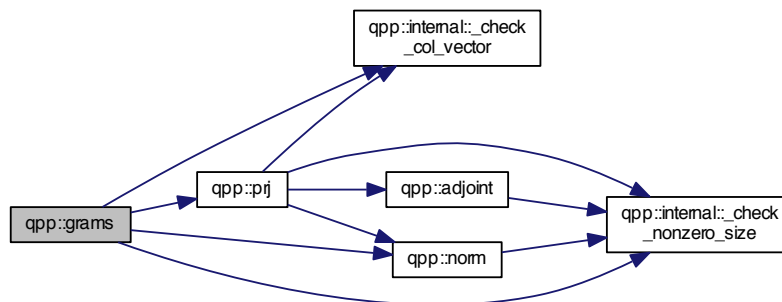
5.1.1.29 `template<typename Derived> double qpp::gconcurrency (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



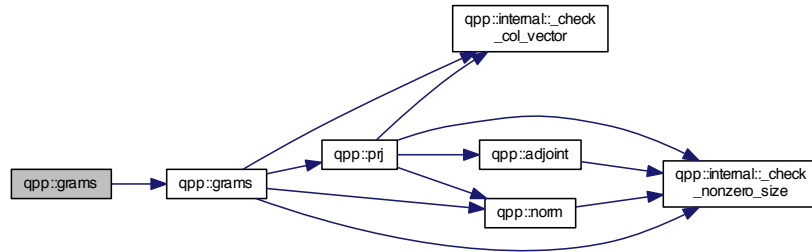
5.1.1.30 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::grams (const std::vector< Derived> & Vs)`

Here is the call graph for this function:



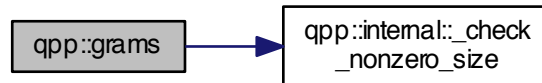
5.1.1.31 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::grams (const std::initializer_list< Derived> & Vs)`

Here is the call graph for this function:



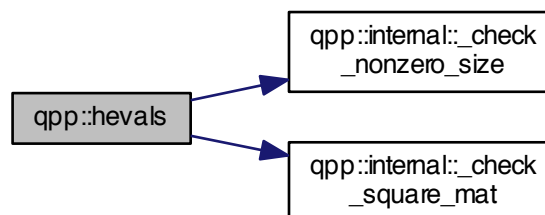
5.1.1.32 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::grams (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



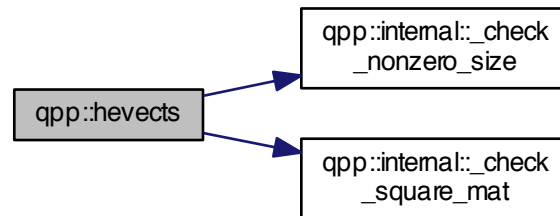
5.1.1.33 `template<typename Derived> types::dmat qpp::hevals (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



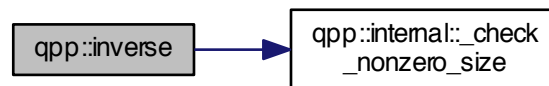
5.1.1.34 `template<typename Derived> types::cmat qpp::hevects (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



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

Here is the call graph for this function:



5.1.1.36 `std::vector<size_t> qpp::invperm (const std::vector< size_t > & perm)`

Here is the call graph for this function:



5.1.1.37 `template<typename T> types::DynMat<typename T::Scalar> qpp::kron (const T & head)`

5.1.1.38 `template<typename T , typename... Args> types::DynMat<typename T::Scalar> qpp::kron (const T & head, const Args &... tail)`

Here is the call graph for this function:



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

Here is the call graph for this function:



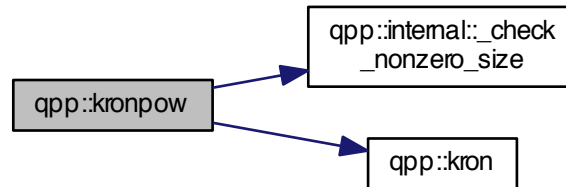
5.1.1.40 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::kron (const std::initializer_list< Derived > & As)`

Here is the call graph for this function:



5.1.1.41 `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:



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

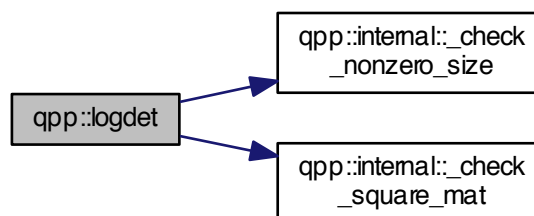
5.1.1.43 `template<typename Derived> Derived qpp::loadMATLABmatrix (const std::string & mat_file, const std::string & var_name)`

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

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

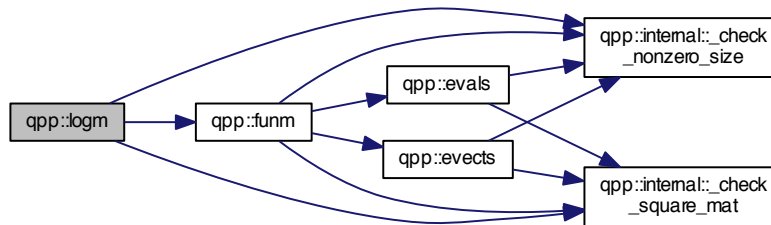
5.1.1.46 `template<typename Derived> Derived::Scalar qpp::logdet (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



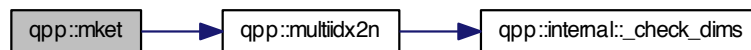
5.1.1.47 `template<typename Derived> types::cmat qpp::logm (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



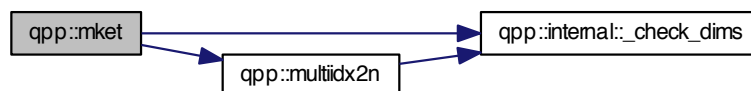
5.1.1.48 `types::ket qpp::mket (const std::vector< size_t> & mask)`

Here is the call graph for this function:



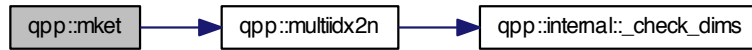
5.1.1.49 `types::ket qpp::mket (const std::vector< size_t> & mask, const std::vector< size_t> & dims)`

Here is the call graph for this function:



5.1.1.50 `types::ket qpp::mket (const std::vector< size_t > & mask, size_t d)`

Here is the call graph for this function:



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

Here is the call graph for this function:



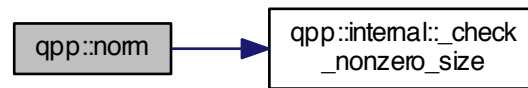
5.1.1.52 `std::vector<size_t> qpp::n2multiidx (size_t n, const std::vector< size_t > & dims)`

Here is the call graph for this function:



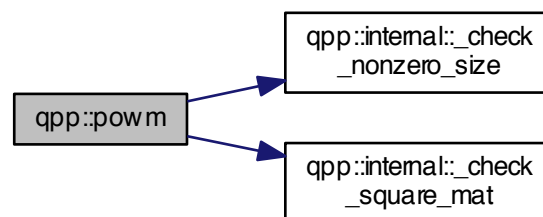
5.1.1.53 `template<typename Derived > double qpp::norm (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



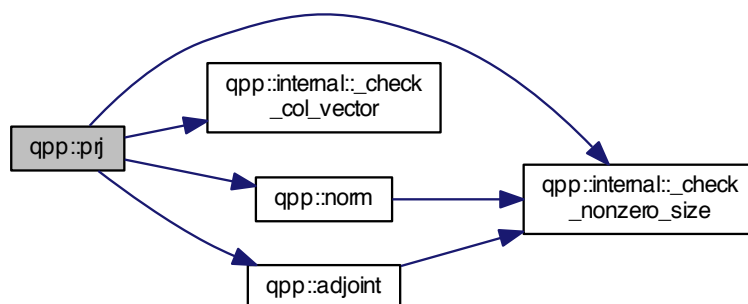
5.1.1.54 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::powm (const Eigen::MatrixBase< Derived > & A, size_t n)`

Here is the call graph for this function:



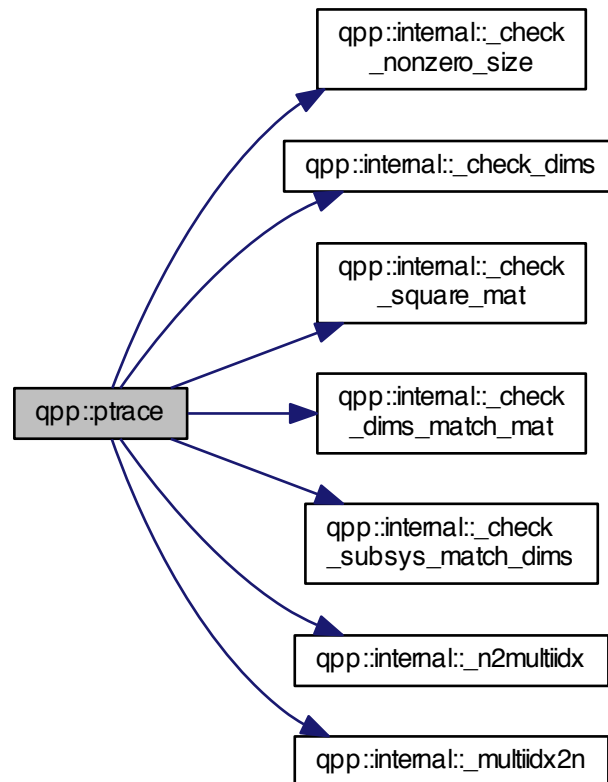
5.1.1.55 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::prj (const Eigen::MatrixBase<Derived> & V)`

Here is the call graph for this function:



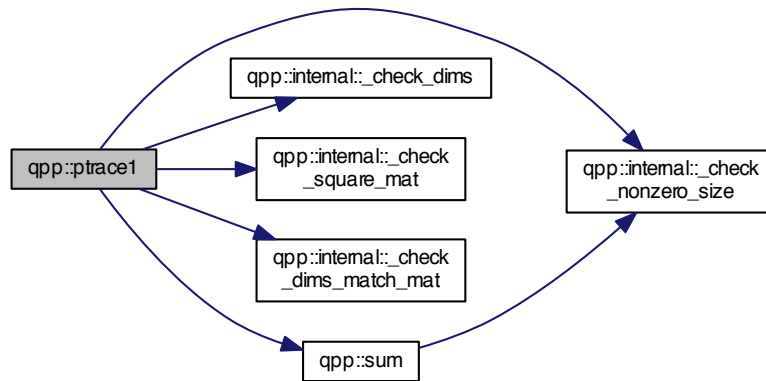
5.1.1.56 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::ptrace (const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & subsys, const std::vector< size_t> & dims)`

Here is the call graph for this function:



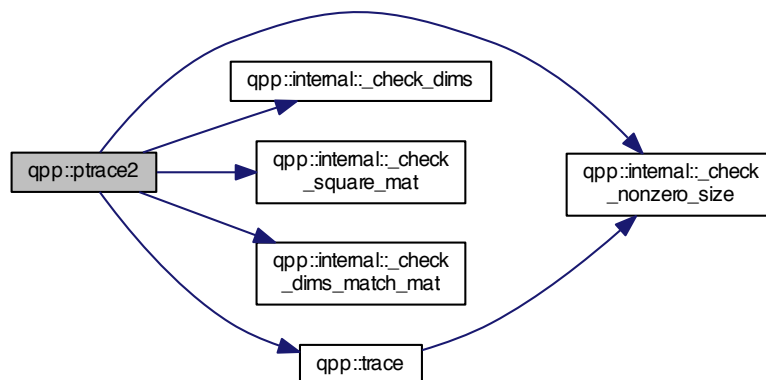
5.1.1.57 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::ptrace1 (const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims)`

Here is the call graph for this function:



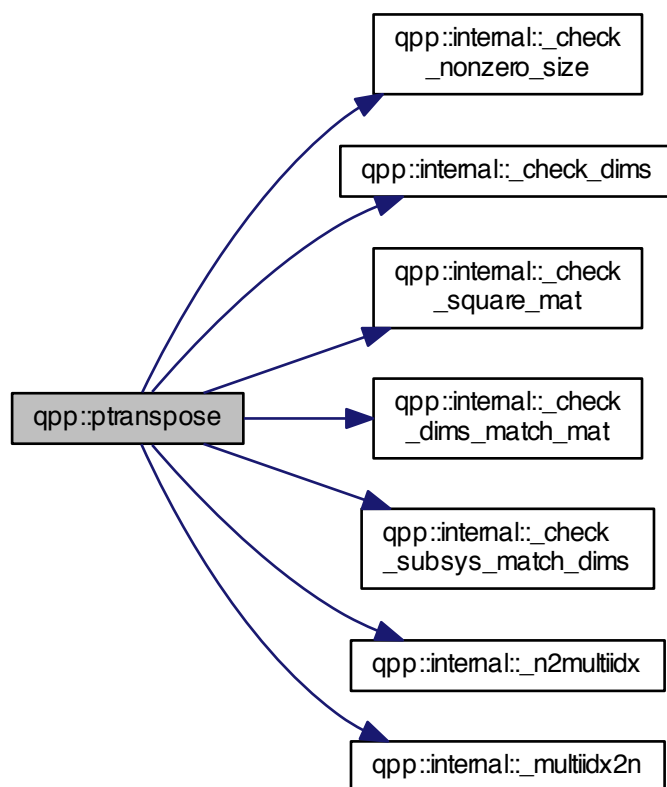
5.1.1.58 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::ptrace2 (const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims)`

Here is the call graph for this function:



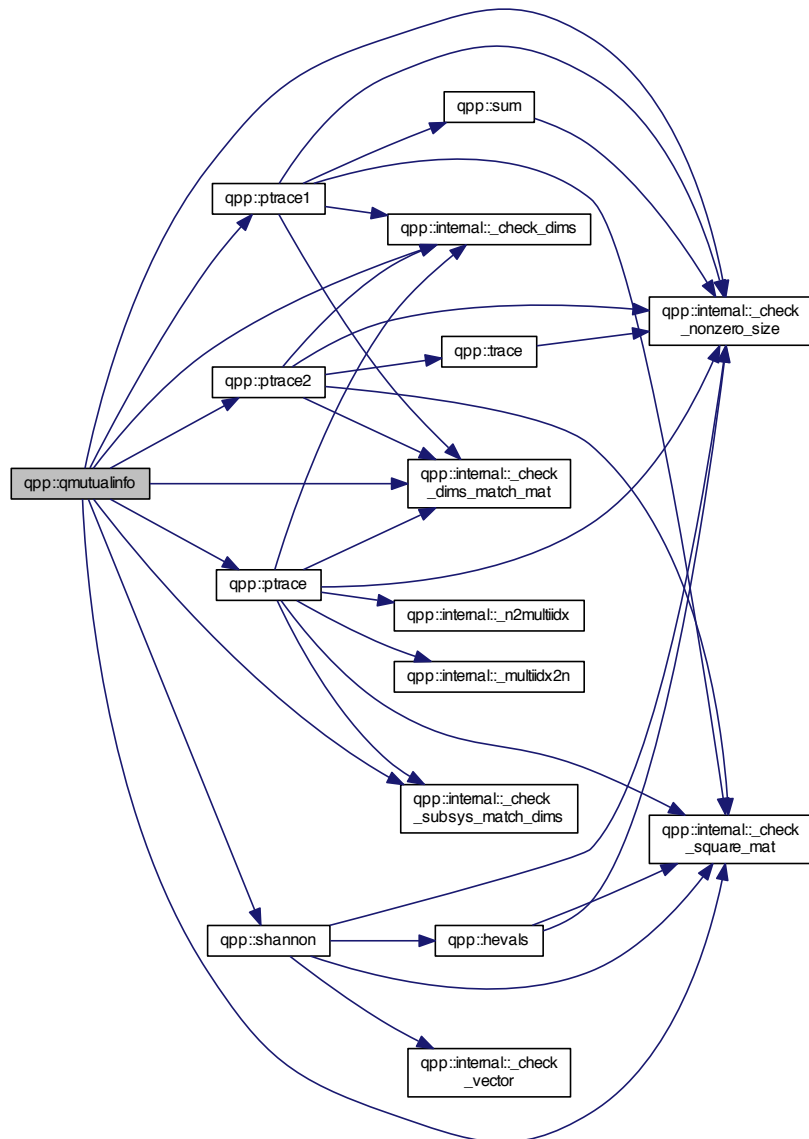
5.1.1.59 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::ptranspose (const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & subsys, const std::vector< size_t> & dims)`

Here is the call graph for this function:



5.1.1.60 `template<typename Derived > double qpp::qmutualinfo (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:



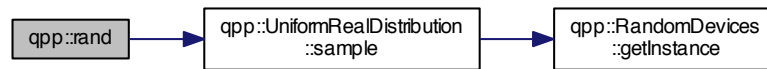
5.1.1.61 `template<typename Derived > Derived qpp::rand (size_t rows, size_t cols, double a = 0, double b = 1)`

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

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

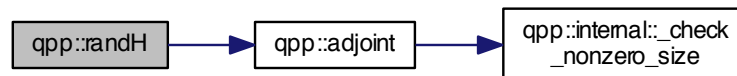
5.1.1.64 `double qpp::rand (double a = 0, double b = 1)`

Here is the call graph for this function:



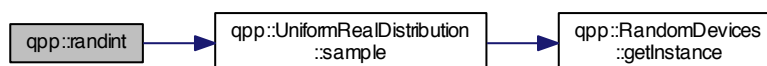
5.1.1.65 `types::cmat qpp::randH (size_t D)`

Here is the call graph for this function:



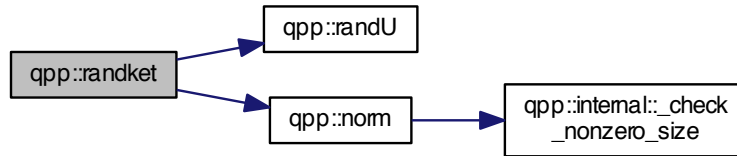
5.1.1.66 `long long qpp::randint (long long a, long long b)`

Here is the call graph for this function:



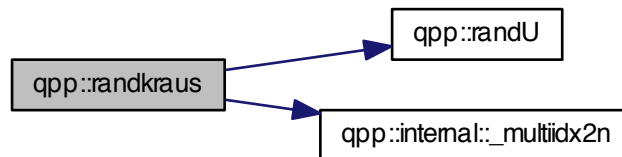
5.1.1.67 `types::ket qpp::randket (size_t D)`

Here is the call graph for this function:



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

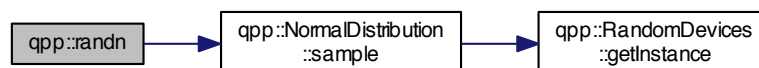
Here is the call graph for this function:



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

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

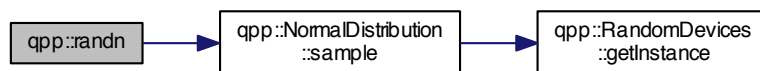
Here is the call graph for this function:



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

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

Here is the call graph for this function:



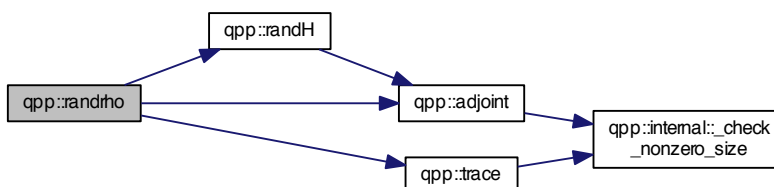
5.1.1.73 `std::vector<size_t> qpp::randperm (size_t n)`

Here is the call graph for this function:



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

Here is the call graph for this function:



5.1.1.75 `types::cmat qpp::randU (size_t D)`

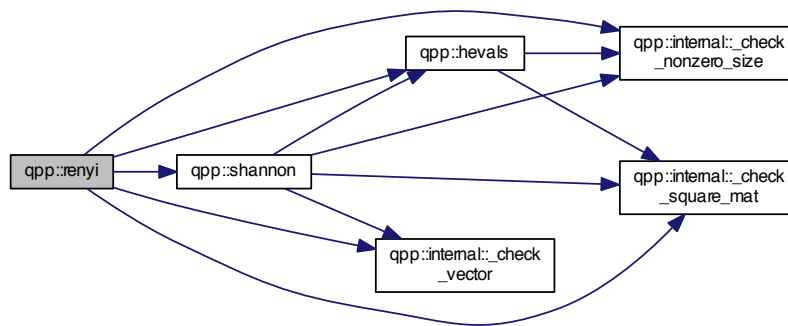
5.1.1.76 `types::cmat qpp::randV (size_t Din, size_t Dout)`

Here is the call graph for this function:



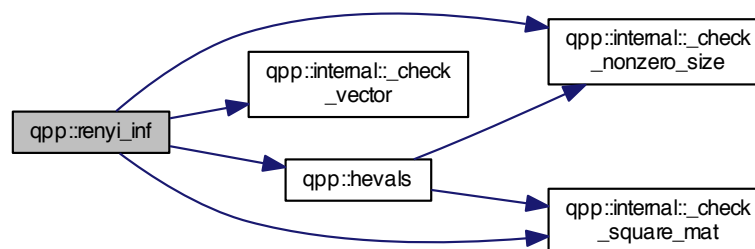
5.1.1.77 `template<typename Derived> double qpp::renyi (const double alpha, const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



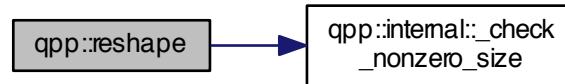
5.1.1.78 `template<typename Derived> double qpp::renyi_inf (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



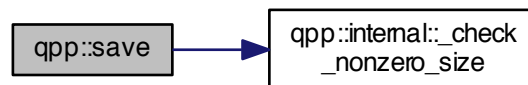
5.1.1.79 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::reshape (const Eigen::MatrixBase< Derived > & A, size_t rows, size_t cols)`

Here is the call graph for this function:



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

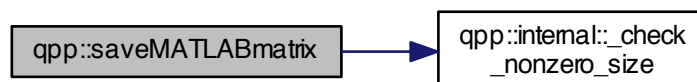
Here is the call graph for this function:



5.1.1.81 `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)`

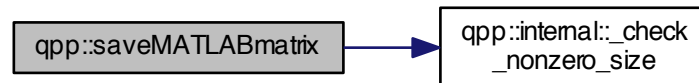
5.1.1.82 `template<> void qpp::saveMATLABmatrix (const Eigen::MatrixBase< typename types::dmat > & A, const std::string & mat_file, const std::string & var_name, const std::string & mode)`

Here is the call graph for this function:



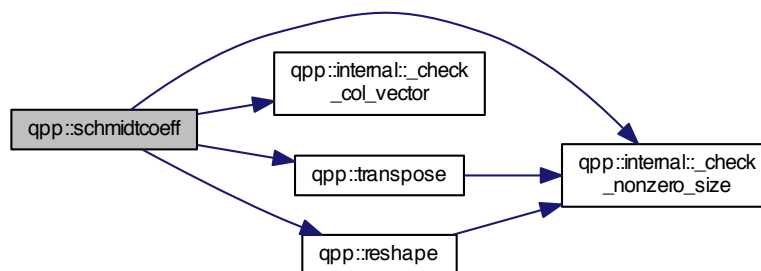
5.1.1.83 `template<> void qpp::saveMATLABmatrix (const Eigen::MatrixBase< typename types::cmat > & A, const std::string & mat_file, const std::string & var_name, const std::string & mode)`

Here is the call graph for this function:



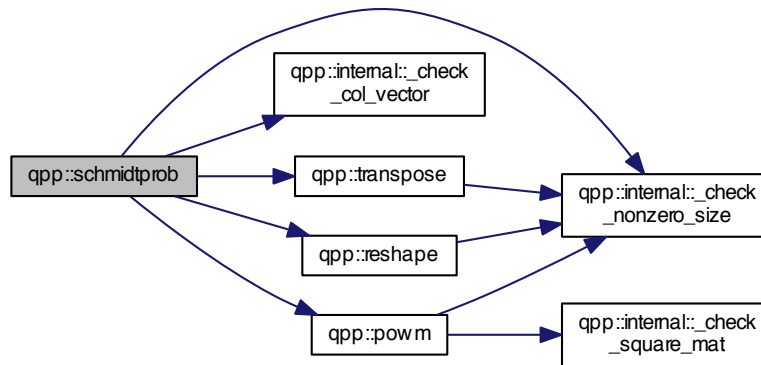
5.1.1.84 `template<typename Derived> types::cmat qpp::schmidtcoeff (const Eigen::MatrixBase< Derived > & A, const std::vector< size_t > & dims)`

Here is the call graph for this function:



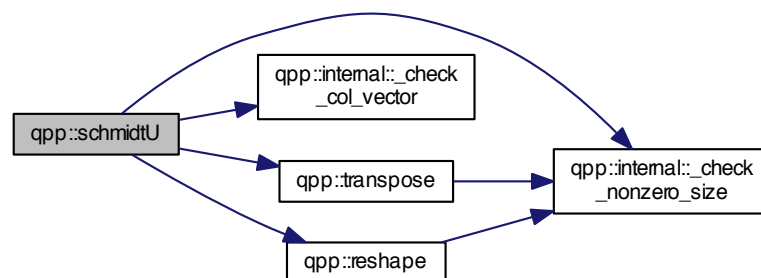
5.1.1.85 `template<typename Derived> types::cmat qpp::schmidtprob (const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims)`

Here is the call graph for this function:



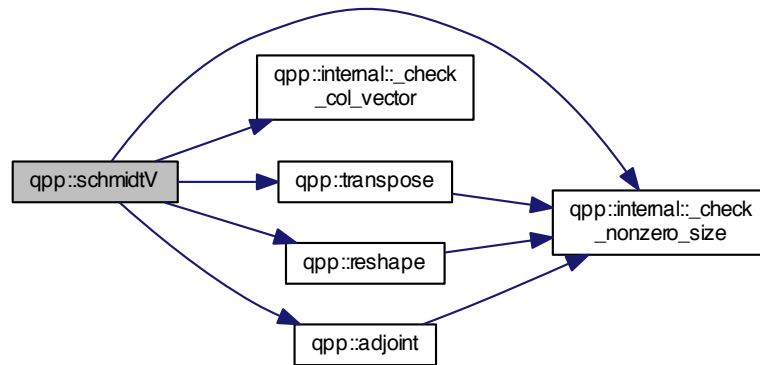
5.1.1.86 `template<typename Derived> types::cmat qpp::schmidtU (const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims)`

Here is the call graph for this function:



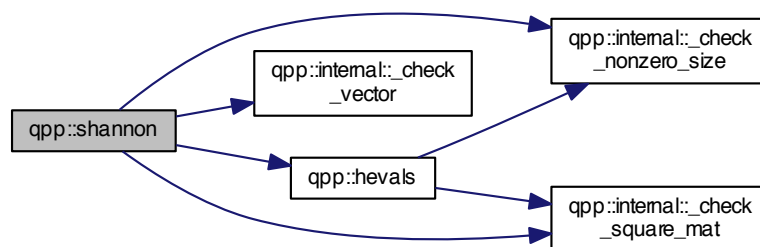
5.1.1.87 `template<typename Derived> types::cmat qpp::schmidtV (const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims)`

Here is the call graph for this function:



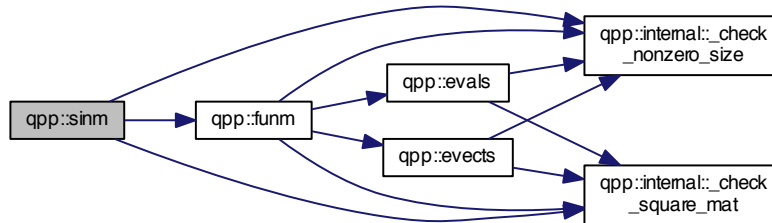
5.1.1.88 `template<typename Derived> double qpp::shannon (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



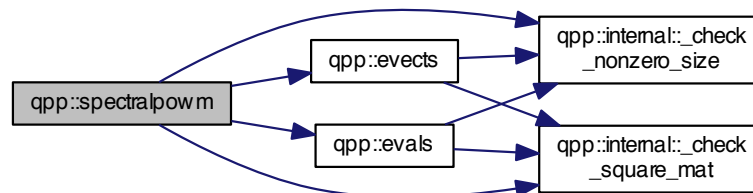
5.1.1.89 `template<typename Derived> types::cmat qpp::sinm (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



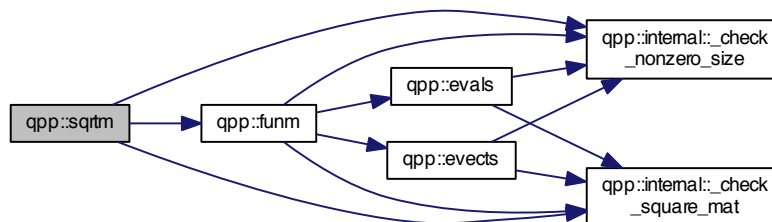
5.1.1.90 `template<typename Derived> types::cmat qpp::spectralpowm (const Eigen::MatrixBase< Derived> & A, const types::cplx z)`

Here is the call graph for this function:



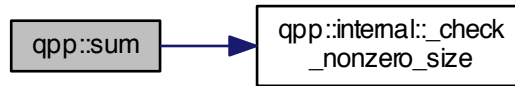
5.1.1.91 `template<typename Derived> types::cmat qpp::sqrtm (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



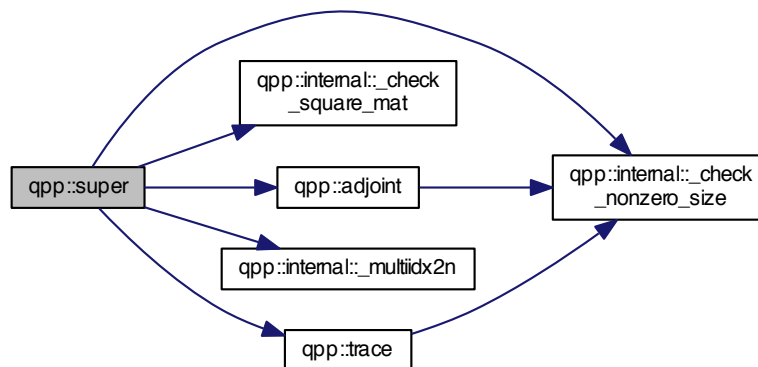
5.1.1.92 `template<typename Derived > Derived::Scalar qpp::sum (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



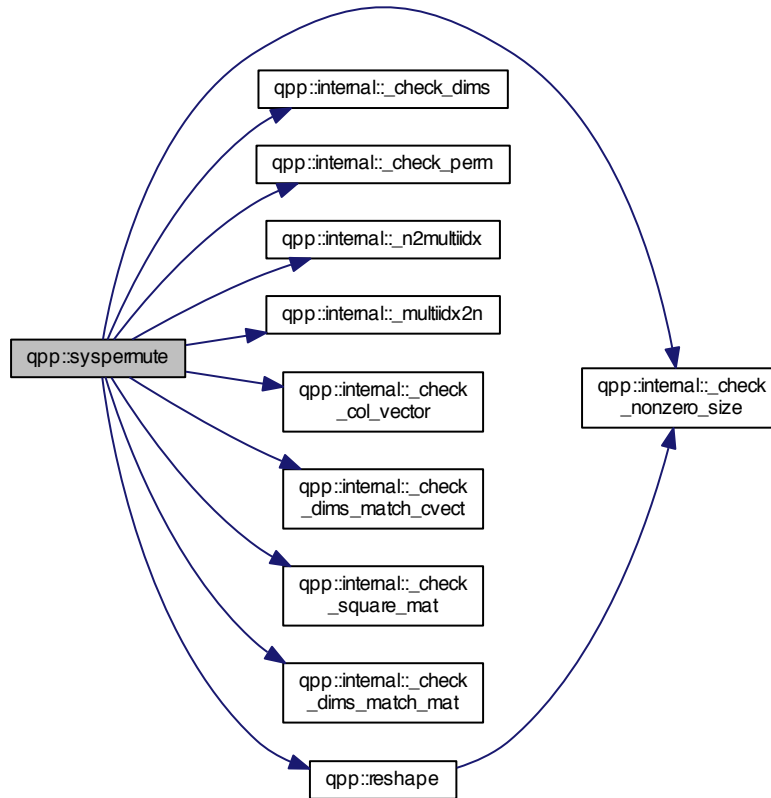
5.1.1.93 `types::cmat qpp::super (const std::vector< types::cmat > & Ks)`

Here is the call graph for this function:



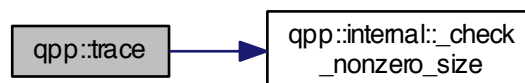
5.1.1.94 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::syspermute (const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & perm, const std::vector< size_t> & dims)`

Here is the call graph for this function:



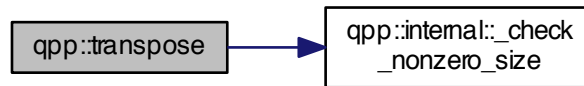
5.1.1.95 `template<typename Derived> Derived::Scalar qpp::trace (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



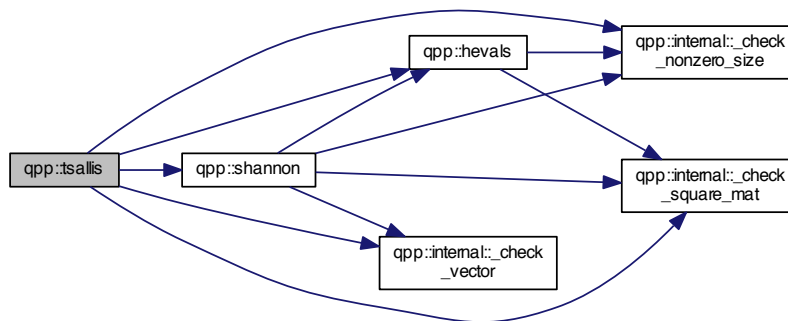
5.1.1.96 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::transpose (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



5.1.1.97 `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()`

5.1.2.3 `const States& qpp::st = States::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 size_t [maxn](#) = 64
- 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 size_t qpp::ct::maxn = 64`

5.2.2.6 `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_match_dims](#) (const std::vector< size_t > &subsys, const std::vector< size_t > &dims)
- bool [_check_perm](#) (const std::vector< size_t > &perm)

- `template<typename Derived1 , typename Derived2 >`
`types::DynMat< typename`
`Derived1::Scalar > _kron2 (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase< Derived2`
`> &B)`
- `template<typename T >`
`void variadic_vector_emplace (std::vector< T > &)`
- `template<typename T , typename First , typename... Args>`
`void variadic_vector_emplace (std::vector< T > &v, First &&first, Args &&...args)`

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)`

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

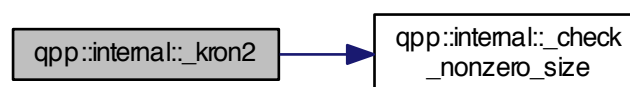
5.3.1.10 `template<typename Derived > bool qpp::internal::_check_square_mat (const Eigen::MatrixBase< Derived > & A)`

5.3.1.11 `bool qpp::internal::_check_subsys_match_dims (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 `template<typename Derived1 , typename Derived2 > types::DynMat<typename Derived1::Scalar>`
`qpp::internal::_kron2 (const Eigen::MatrixBase< Derived1 > & A, const Eigen::MatrixBase< Derived2 > & B)`

Here is the call graph for this function:



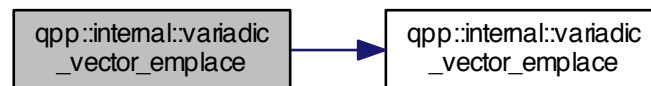
5.3.1.14 `size_t qpp::internal::_multiidx2n (const size_t * midx, size_t numdims, const size_t * dims)`

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

5.3.1.16 `template<typename T> void qpp::internal::variadic_vector_emplace (std::vector< T > &)`

5.3.1.17 `template<typename T, typename First, typename... Args> void qpp::internal::variadic_vector_emplace (std::vector< T > & v, First && first, Args &&... args)`

Here is the call graph for this function:



5.4 qpp::types Namespace Reference

Typedefs

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

5.4.1 Typedef Documentation

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

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

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

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

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

5.4.1.6 using `qpp::types::ket` = typedef `Eigen::Matrix<cplx, Eigen::Dynamic, 1>`

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]

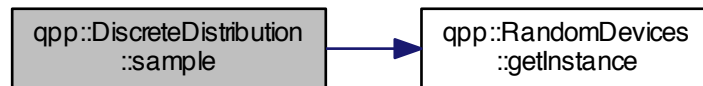
6.1.1.3 `qpp::DiscreteDistribution::DiscreteDistribution (std::vector< double > weights)` [inline]

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)

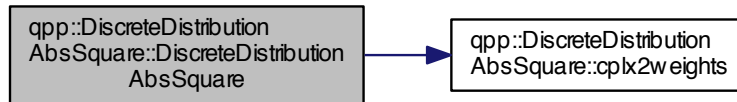
Protected Attributes

- `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:



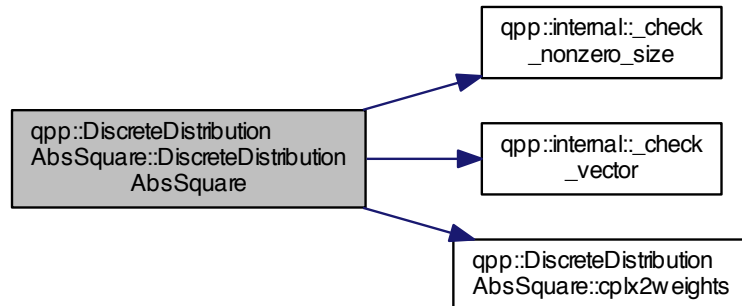
6.2.1.3 `qpp::DiscreteDistributionAbsSquare::DiscreteDistributionAbsSquare (std::vector< types::cplx > amplitudes) [inline]`

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:



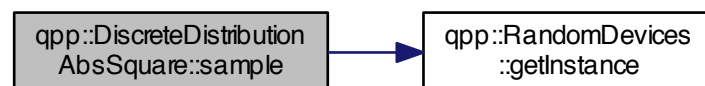
6.2.2 Member Function Documentation

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]`

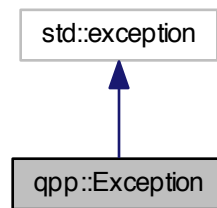
The documentation for this class was generated from the following file:

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

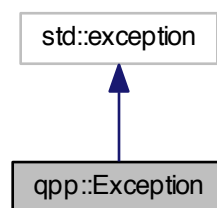
6.3 qpp::Exception Class Reference

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

Inheritance diagram for qpp::Exception:



Collaboration diagram for qpp::Exception:



Public Types

- enum `Type` {
`Type::UNKNOWN_EXCEPTION = 1`, `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_INVALID`, `Type::NOT_QUBIT_GATE`, `Type::NOT_QUBIT_SUBSYS`, `Type::NOT_BIPARTITE`,
`Type::OUT_OF_RANGE`, `Type::UNDEFINED_TYPE`, `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]

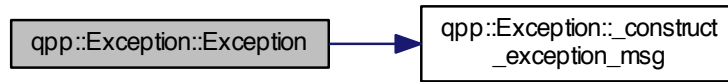
Enumerator

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_INVALID
NOT_QUBIT_GATE
NOT_QUBIT_SUBSYS
NOT_BIPARTITE
OUT_OF_RANGE
UNDEFINED_TYPE
CUSTOM_EXCEPTION

6.3.2 Constructor & Destructor Documentation

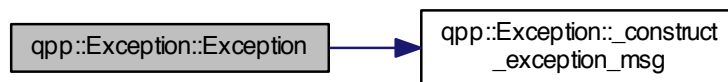
6.3.2.1 `qpp::Exception::Exception (const std::string & where, const Type & type)` `[inline]`

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 > &subsys, 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 CNOTba](#)
- [types::cmat SWAP](#)
- [types::cmat TOF](#)
- [types::cmat FRED](#)

Private Member Functions

- [Gates](#) ()

6.4.1 Constructor & Destructor Documentation

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

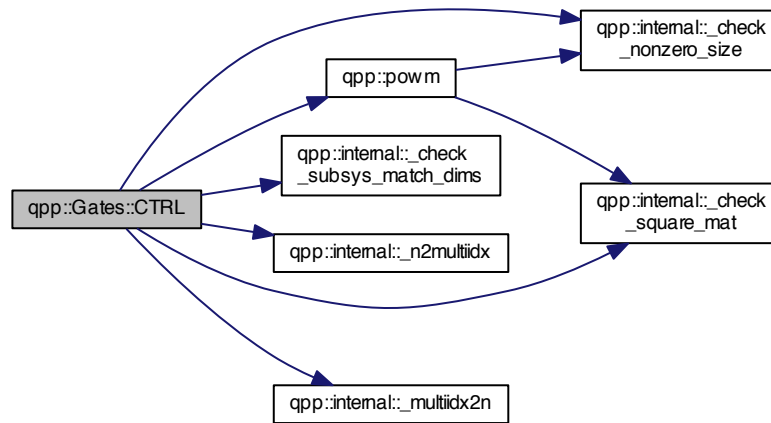
6.4.1.2 `qpp::Gates::Gates (const Gates &) [delete]`

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 > & subsys, 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]

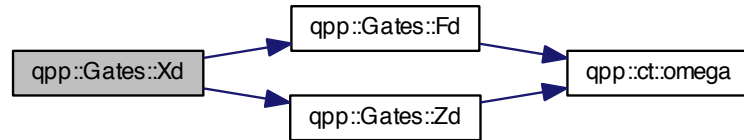
6.4.2.4 `types::cmat qpp::Gates::Id (size_t D) const` [inline]

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::cmat qpp::Gates::CNOTab`

6.4.3.2 `types::cmat qpp::Gates::CNOTba`

6.4.3.3 `types::cmat qpp::Gates::CZ`

6.4.3.4 `types::cmat qpp::Gates::FRED`

6.4.3.5 `types::cmat qpp::Gates::H`

6.4.3.6 `types::cmat qpp::Gates::Id2`

6.4.3.7 `types::cmat qpp::Gates::S`

6.4.3.8 `types::cmat qpp::Gates::SWAP`

6.4.3.9 `types::cmat qpp::Gates::T`

6.4.3.10 `types::cmat qpp::Gates::TOF`

6.4.3.11 `types::cmat qpp::Gates::X`

6.4.3.12 `types::cmat qpp::Gates::Y`

6.4.3.13 `types::cmat qpp::Gates::Z`

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`

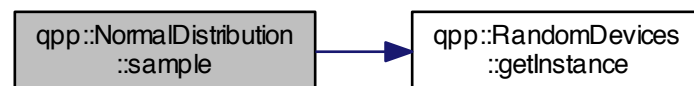
6.5.1 Constructor & Destructor Documentation

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=[States::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

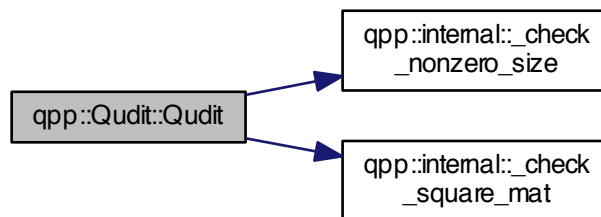
Private Attributes

- [types::cmat _rho](#)
- [size_t _D](#)

6.6.1 Constructor & Destructor Documentation

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

Here is the call graph for this function:



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

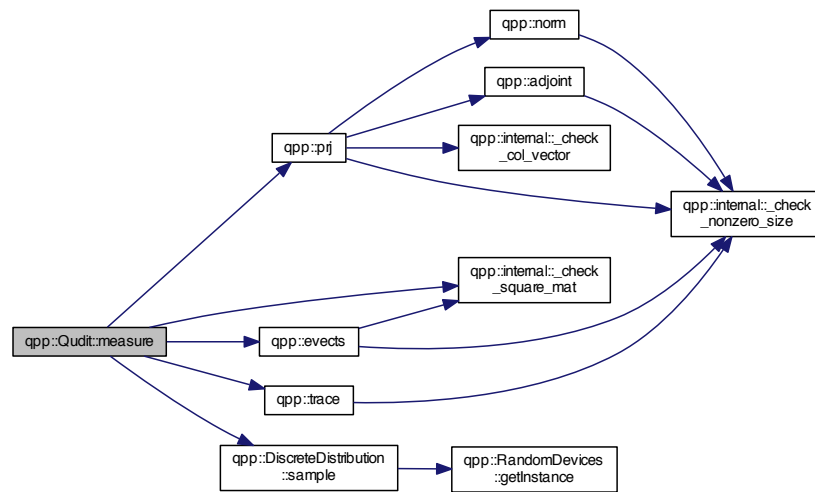
6.6.2 Member Function Documentation

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

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

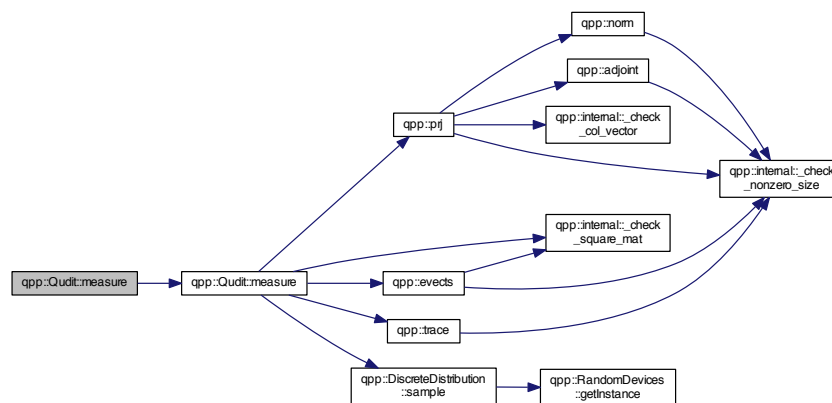
6.6.2.3 `size_t qpp::Qudit::measure (const types::cmat & U, bool destructive = false) [inline]`

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::States Class Reference

```
#include <states.h>
```


Public Member Functions

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

Static Public Member Functions

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

Public Attributes

- [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](#)
- [types::ket GHZ](#)
- [types::ket W](#)
- [types::cmat pGHZ](#)
- [types::cmat pW](#)

Private Member Functions

- [States](#) ()

6.8.1 Constructor & Destructor Documentation

6.8.1.1 `qpp::States::States ()` `[inline]`, `[private]`

6.8.1.2 `qpp::States::States (const States &)` `[delete]`

6.8.1.3 `virtual qpp::States::~~States ()` `[virtual]`, `[default]`

6.8.2 Member Function Documentation

6.8.2.1 `static const States& qpp::States::getInstance ()` `[inline]`, `[static]`

6.8.2.2 **States& qpp::States::operator= (const States &) [delete]**

6.8.3 Member Data Documentation

6.8.3.1 **types::ket qpp::States::b00**

6.8.3.2 **types::ket qpp::States::b01**

6.8.3.3 **types::ket qpp::States::b10**

6.8.3.4 **types::ket qpp::States::b11**

6.8.3.5 **types::ket qpp::States::GHZ**

6.8.3.6 **types::cmat qpp::States::pb00**

6.8.3.7 **types::cmat qpp::States::pb01**

6.8.3.8 **types::cmat qpp::States::pb10**

6.8.3.9 **types::cmat qpp::States::pb11**

6.8.3.10 **types::cmat qpp::States::pGHZ**

6.8.3.11 **types::cmat qpp::States::pW**

6.8.3.12 **types::cmat qpp::States::px0**

6.8.3.13 **types::cmat qpp::States::px1**

6.8.3.14 **types::cmat qpp::States::py0**

6.8.3.15 **types::cmat qpp::States::py1**

6.8.3.16 **types::cmat qpp::States::pz0**

6.8.3.17 **types::cmat qpp::States::pz1**

6.8.3.18 **types::ket qpp::States::W**

6.8.3.19 **types::ket qpp::States::x0**

6.8.3.20 **types::ket qpp::States::x1**

6.8.3.21 **types::ket qpp::States::y0**

6.8.3.22 **types::ket qpp::States::y1**

6.8.3.23 **types::ket qpp::States::z0**

6.8.3.24 **types::ket qpp::States::z1**

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

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

6.9 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)

6.9.1 Constructor & Destructor Documentation

6.9.1.1 [qpp::Timer::Timer](#) () [\[inline\]](#)

6.9.1.2 [virtual qpp::Timer::~~Timer](#) () [\[virtual\]](#),[\[default\]](#)

6.9.2 Member Function Documentation

6.9.2.1 [double qpp::Timer::seconds](#) () const [\[inline\]](#)

6.9.2.2 [void qpp::Timer::tic](#) () [\[inline\]](#)

6.9.2.3 [void qpp::Timer::toc](#) () [\[inline\]](#)

6.9.3 Friends And Related Function Documentation

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

6.9.4 Member Data Documentation

6.9.4.1 [std::chrono::high_resolution_clock::time_point qpp::Timer::_end](#) [\[protected\]](#)

6.9.4.2 [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](#)

6.10 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](#)

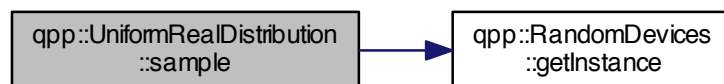
6.10.1 Constructor & Destructor Documentation

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

6.10.2 Member Function Documentation

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

Here is the call graph for this function:



6.10.3 Member Data Documentation

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

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

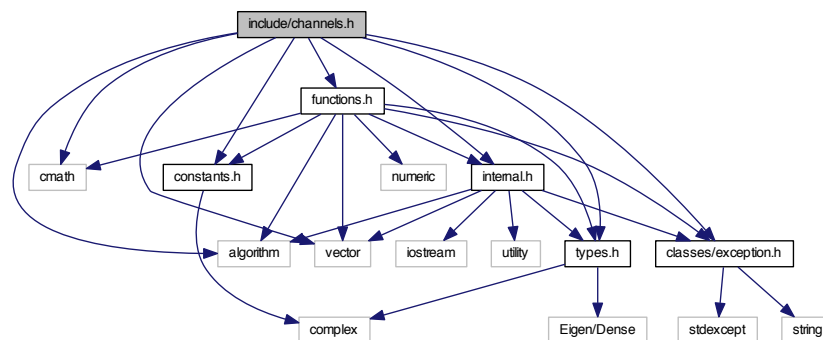
- `include/classes/stat.h`

Chapter 7

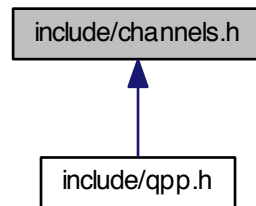
File Documentation

7.1 include/channels.h File Reference

```
#include <algorithm>
#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

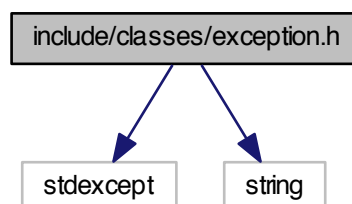
- `types::cmat qpp::super (const std::vector< types::cmat > &Ks)`
- `types::cmat qpp::choi (const std::vector< types::cmat > &Ks)`
- `std::vector< types::cmat > qpp::choi2kraus (const types::cmat &A)`
- `template<typename Derived >`
`types::cmat qpp::channel (const Eigen::MatrixBase< Derived > &rho, const std::vector< types::cmat > &Ks)`
- `template<typename Derived >`
`types::cmat qpp::channel (const Eigen::MatrixBase< Derived > &rho, const std::vector< types::cmat > &Ks,`
`const std::vector< size_t > &subsys, const std::vector< size_t > &dims)`

7.2 include/classes/exception.h File Reference

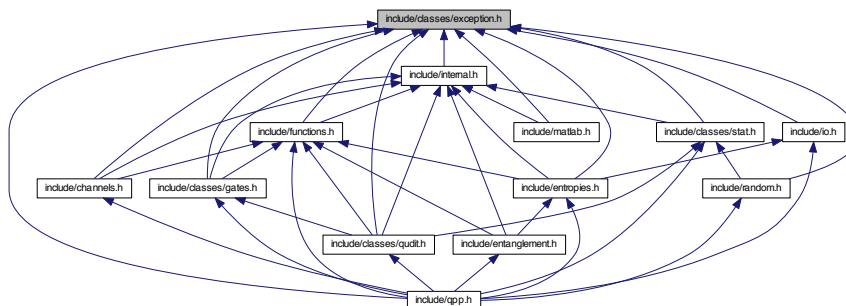
```
#include <stdexcept>
```

```
#include <string>
```

Include dependency graph for exception.h:



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



Classes

- class [qpp::Exception](#)

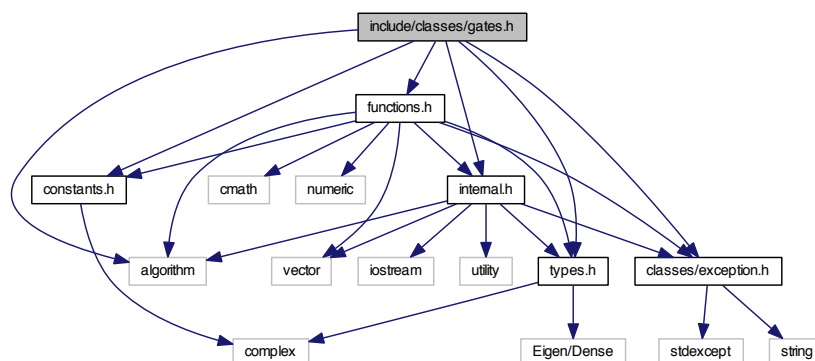
Namespaces

- [qpp](#)

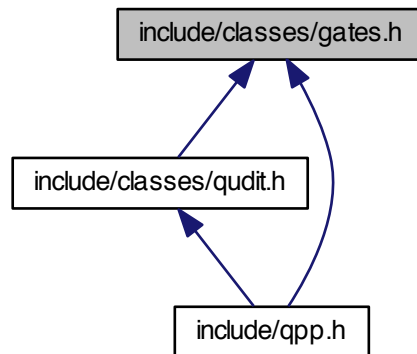
7.3 include/classes/gates.h File Reference

```
#include <algorithm>
#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`

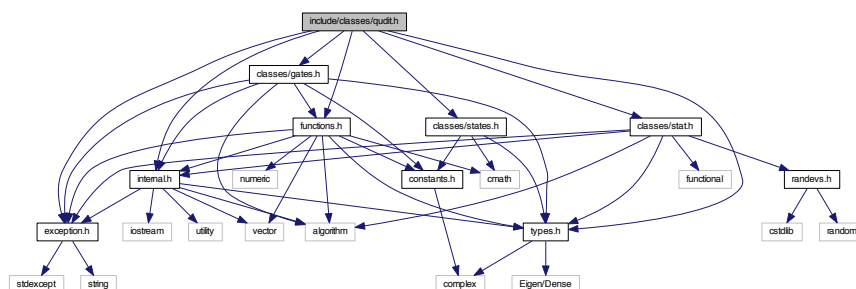
7.4 include/classes/qudit.h File Reference

```

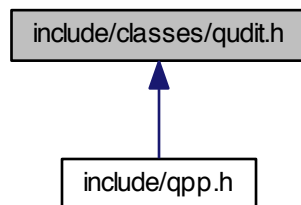
#include "exception.h"
#include "functions.h"
#include "internal.h"
#include "types.h"
#include "classes/gates.h"
#include "classes/stat.h"
#include "classes/states.h"

```

Include dependency graph for `qudit.h`:



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



Classes

- class `qpp::Qudit`

Namespaces

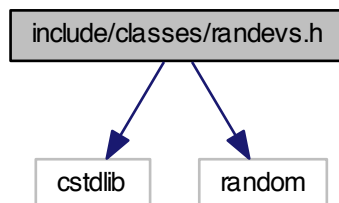
- `qpp`

7.5 include/classes/randevs.h File Reference

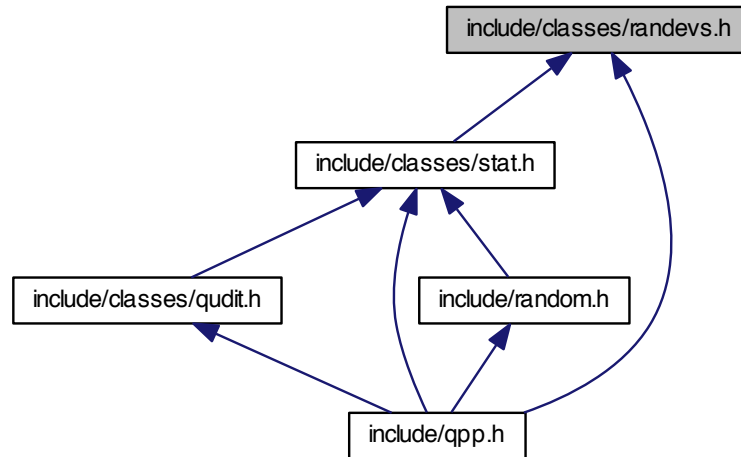
```
#include <cstdlib>
```

```
#include <random>
```

Include dependency graph for randevs.h:



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



Classes

- class [qpp::RandomDevices](#)

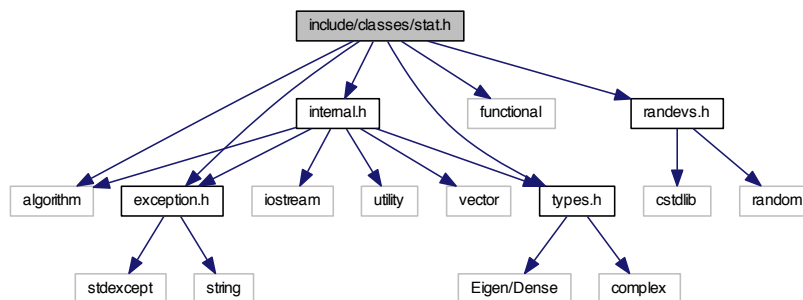
Namespaces

- [qpp](#)

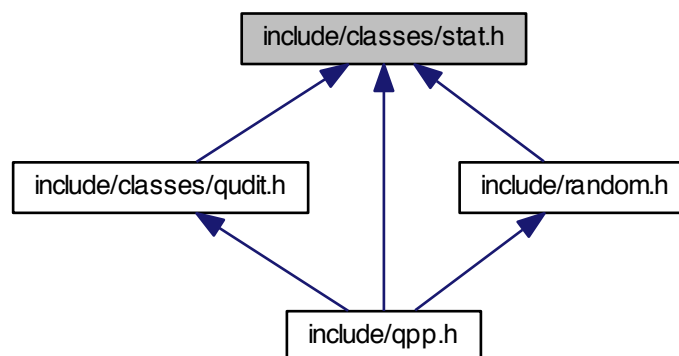
7.6 include/classes/stat.h File Reference

```
#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](#)

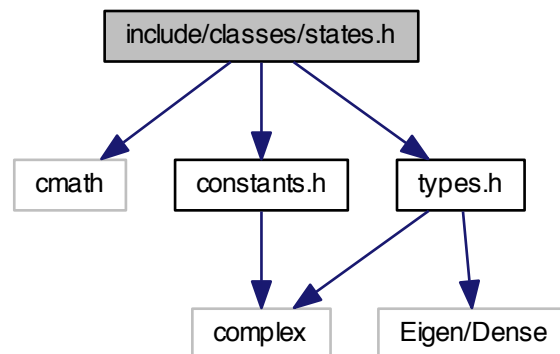
Namespaces

- [qpp](#)

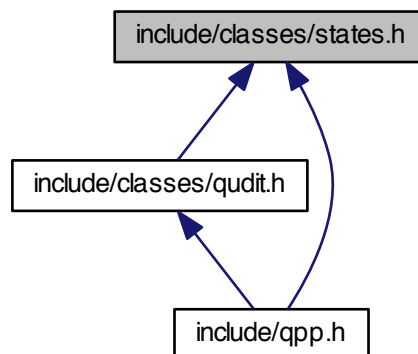
7.7 include/classes/stat.h File Reference

```
#include <cmath>
#include "constants.h"
#include "types.h"
```

Include dependency graph for states.h:



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



Classes

- class [qpp::States](#)

Namespaces

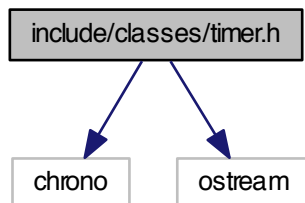
- [qpp](#)

7.8 include/classes/timer.h File Reference

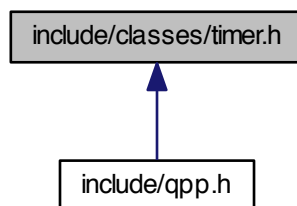
```
#include <chrono>
```

```
#include <ostream>
```

Include dependency graph for timer.h:



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



Classes

- class [qpp::Timer](#)

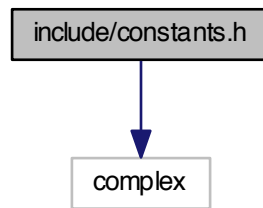
Namespaces

- [qpp](#)

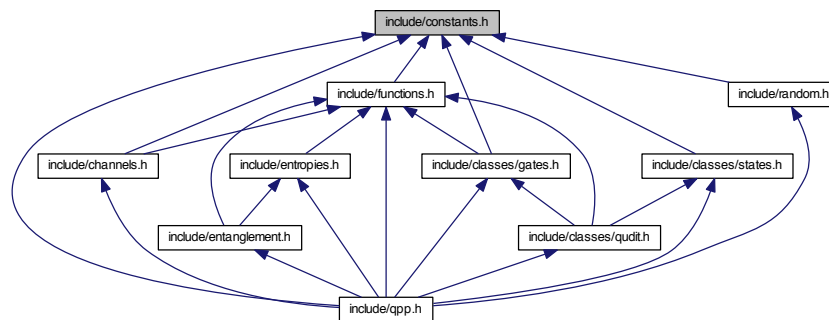
7.9 include/constants.h File Reference

```
#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)`

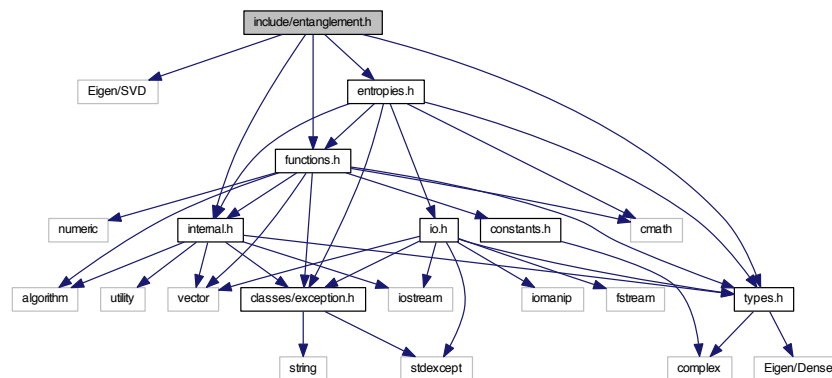
Variables

- `const double qpp::ct::chop = 1e-10`
- `const double qpp::ct::eps = 1e-12`
- `const size_t qpp::ct::maxn = 64`
- `const std::complex< double > qpp::ct::ii = { 0, 1 }`
- `const double qpp::ct::pi = 3.141592653589793238462643383279502884`
- `const double qpp::ct::ee = 2.718281828459045235360287471352662497`

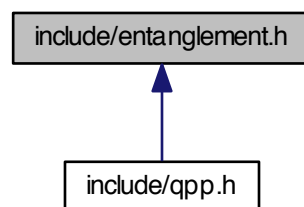
7.10 include/entanglement.h File Reference

```
#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

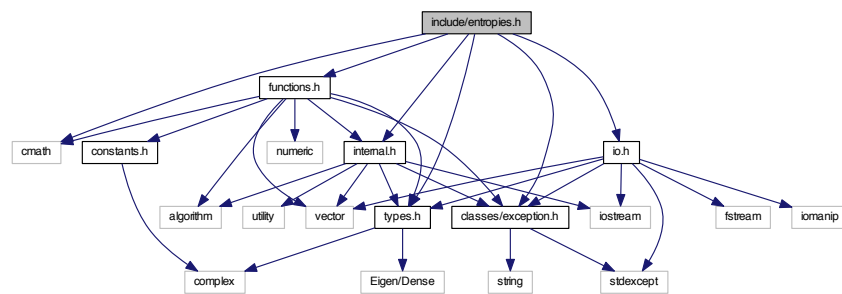
- `template<typename Derived >`
`types::cmat qpp::schmidtcoeff (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &dims)`
- `template<typename Derived >`
`types::cmat qpp::schmidtU (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &dims)`
- `template<typename Derived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

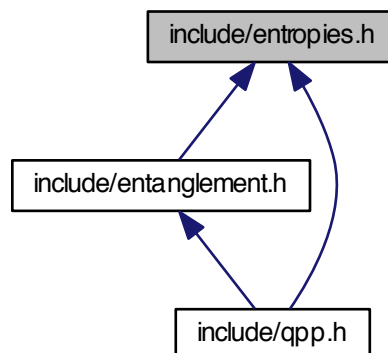
7.11 include/entropies.h File Reference

```
#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](#)

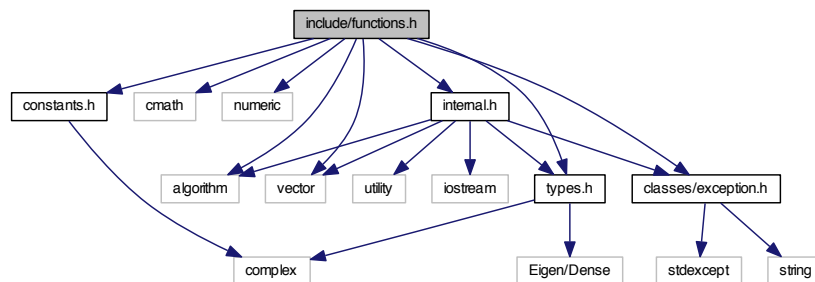
Functions

- `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::qmutualinfo (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &subsys, const std::vector< size_t > &dims)`

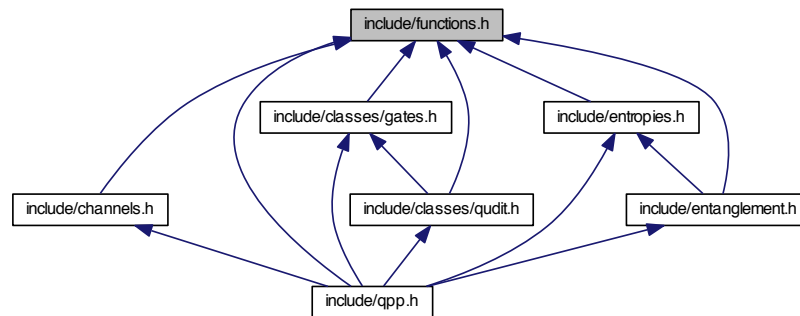
7.12 include/functions.h File Reference

```
#include <algorithm>
#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 >`
`types::DynMat< typename`
`Derived::Scalar > qpp::inverse (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::logdet (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::dmat 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 &))`

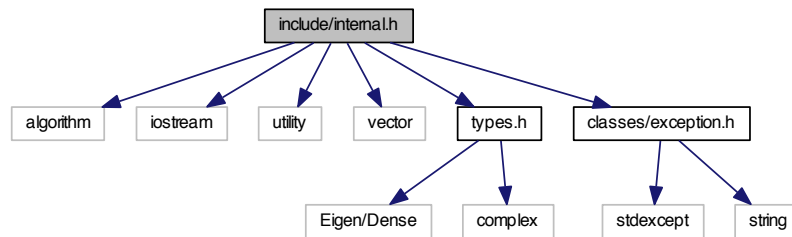
- `template<typename Derived >`
`types::cmat qpp::sqrtrm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat qpp::absm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`
`types::cmat qpp::expm (const Eigen::MatrixBase< Derived > &A)`
- `template<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::cwise (const Eigen::MatrixBase< Derived > &A, Output-`
`Scalar(*f)(const typename Derived::Scalar &))`
- `template<typename T >`
`types::DynMat< typename T::Scalar > qpp::kron (const T &head)`
- `template<typename T , typename... Args>`
`types::DynMat< typename T::Scalar > qpp::kron (const T &head, const Args &...tail)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::kron (const std::vector< Derived > &As)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::kron (const std::initializer_list< Derived > &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::ptrace1 (const Eigen::MatrixBase< Derived > &A, 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< 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::prj (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 Derived1 , typename Derived2 >`
`types::DynMat< typename`
`Derived1::Scalar > qpp::gate (const Eigen::MatrixBase< Derived1 > &state, const Eigen::MatrixBase< Derived2 > &A, const std::vector< size_t > &subsys, const std::vector< size_t > &dims)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::grams (const std::vector< Derived > &Vs)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::grams (const std::initializer_list< Derived > &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)`
- `types::ket qpp::mket (const std::vector< size_t > &mask)`
- `types::ket qpp::mket (const std::vector< size_t > &mask, const std::vector< size_t > &dims)`
- `types::ket qpp::mket (const std::vector< size_t > &mask, size_t d)`
- `std::vector< size_t > qpp::invperm (const std::vector< size_t > &perm)`
- `std::vector< size_t > qpp::compperm (const std::vector< size_t > &perm, const std::vector< size_t > &sigma)`

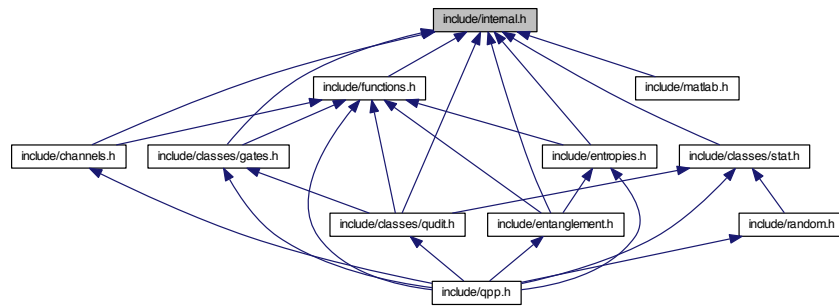
7.13 include/internal.h File Reference

```
#include <algorithm>
#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

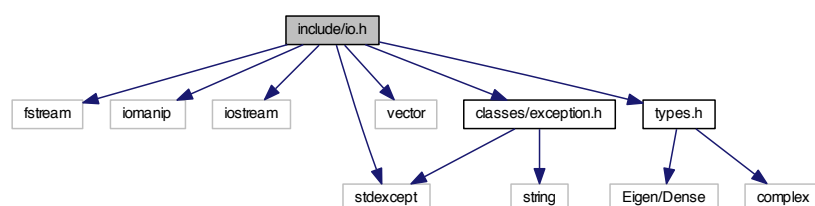
- void [qpp::internal::_n2multiidx](#) (size_t n, size_t numdims, const size_t *dims, size_t *result)
- size_t [qpp::internal::_multiidx2n](#) (const size_t *midx, size_t numdims, const size_t *dims)
- template<typename Derived >
bool [qpp::internal::_check_square_mat](#) (const Eigen::MatrixBase< Derived > &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_match_dims (const std::vector< size_t > &subsys, const std::vector< size_t > &dims)`
- `bool qpp::internal::_check_perm (const std::vector< size_t > &perm)`
- `template<typename Derived1 , typename Derived2 >`
`types::DynMat< typename`
`Derived1::Scalar > qpp::internal::_kron2 (const Eigen::MatrixBase< Derived1 > &A, const Eigen::Matrix-`
`Base< Derived2 > &B)`
- `template<typename T >`
`void qpp::internal::variadic_vector_emplace (std::vector< T > &)`
- `template<typename T , typename First , typename... Args>`
`void qpp::internal::variadic_vector_emplace (std::vector< T > &v, First &&first, Args &&...args)`

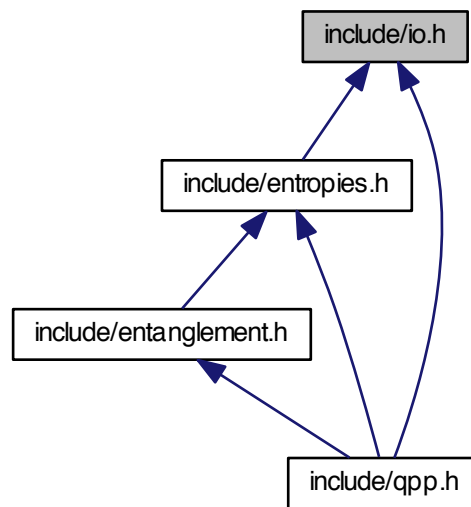
7.14 include/io.h File Reference

```
#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](#)

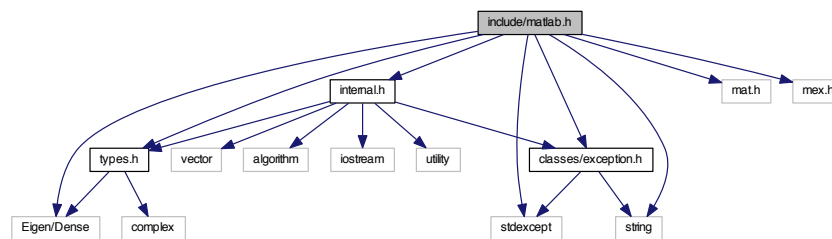
Functions

- `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)`

7.15 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)`

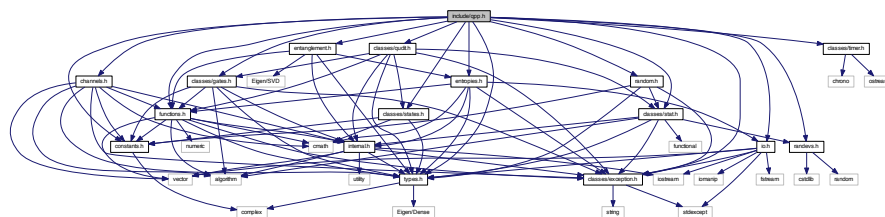
7.16 include/qpp.h File Reference

```
#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/states.h"
#include "classes/timer.h"
```

Include dependency graph for qpp.h:



Namespaces

- qpp

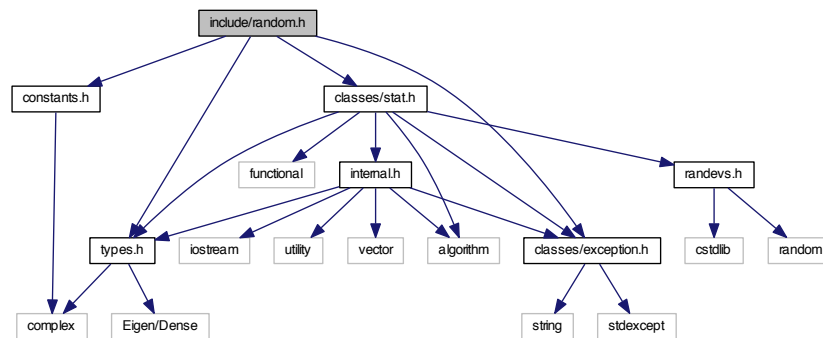
Variables

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

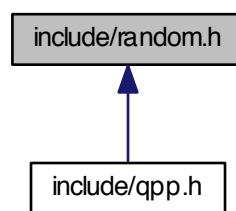
7.17 include/random.h File Reference

```
#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)
- `long long` [qpp::randint](#) (long long a, long long b)
- `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)

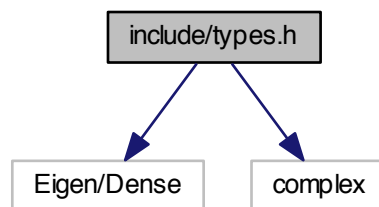
- `types::cmat` [qpp::randU](#) (`size_t D`)
- `types::cmat` [qpp::randV](#) (`size_t Din`, `size_t Dout`)
- `std::vector< types::cmat >` [qpp::randkraus](#) (`size_t n`, `size_t D`)
- `types::cmat` [qpp::randH](#) (`size_t D`)
- `types::ket` [qpp::randket](#) (`size_t D`)
- `types::cmat` [qpp::randrho](#)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

7.18 include/types.h File Reference

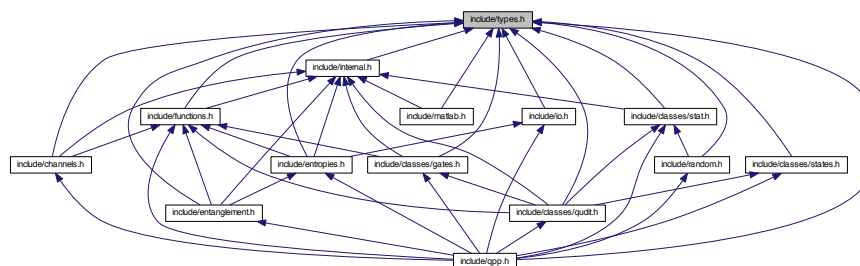
```
#include <Eigen/Dense>
```

```
#include <complex>
```

Include dependency graph for types.h:



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



Namespaces

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

Typedefs

- using [qpp::types::cplx](#) = `std::complex< double >`
- using [qpp::types::cmat](#) = `Eigen::MatrixXcd`
- using [qpp::types::dmat](#) = `Eigen::MatrixXd`
- using [qpp::types::ket](#) = `Eigen::Matrix< cplx, Eigen::Dynamic, 1 >`

- using `qpp::types::bra` = `Eigen::Matrix< cplx, 1, Eigen::Dynamic >`
- `template<typename Scalar >`
using `qpp::types::DynMat` = `Eigen::Matrix< Scalar, Eigen::Dynamic, Eigen::Dynamic >`