

qpp
0.1

Generated by Doxygen 1.8.7

Wed Oct 22 2014 14:01:53

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	14
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	20
5.1.1.15	disp	20
5.1.1.16	disp	20
5.1.1.17	disp	20
5.1.1.18	displn	21

5.1.1.19	displn	21
5.1.1.20	displn	21
5.1.1.21	displn	22
5.1.1.22	entanglement	22
5.1.1.23	evals	22
5.1.1.24	evects	23
5.1.1.25	expandout	24
5.1.1.26	expm	24
5.1.1.27	funm	25
5.1.1.28	gconcurrency	26
5.1.1.29	grams	27
5.1.1.30	grams	27
5.1.1.31	grams	27
5.1.1.32	hevals	28
5.1.1.33	hevects	28
5.1.1.34	inverse	29
5.1.1.35	invperm	30
5.1.1.36	kron	30
5.1.1.37	kron	30
5.1.1.38	kron	31
5.1.1.39	kron	31
5.1.1.40	kronpow	31
5.1.1.41	load	31
5.1.1.42	loadMATLABmatrix	32
5.1.1.43	loadMATLABmatrix	32
5.1.1.44	loadMATLABmatrix	32
5.1.1.45	logdet	32
5.1.1.46	logm	32
5.1.1.47	mket	33
5.1.1.48	mket	33
5.1.1.49	mket	34
5.1.1.50	multiidx2n	34
5.1.1.51	n2multiidx	34
5.1.1.52	norm	34
5.1.1.53	operator""_i	35
5.1.1.54	operator""_i	35
5.1.1.55	powm	35
5.1.1.56	prj	36
5.1.1.57	ptrace	37
5.1.1.58	ptrace1	38

5.1.1.59	ptrace2	38
5.1.1.60	ptranspose	39
5.1.1.61	qmutualinfo	40
5.1.1.62	rand	40
5.1.1.63	rand	40
5.1.1.64	rand	41
5.1.1.65	rand	41
5.1.1.66	randH	41
5.1.1.67	randint	41
5.1.1.68	randket	42
5.1.1.69	randkraus	42
5.1.1.70	randn	42
5.1.1.71	randn	42
5.1.1.72	randn	43
5.1.1.73	randn	43
5.1.1.74	randperm	43
5.1.1.75	randrho	44
5.1.1.76	randU	44
5.1.1.77	randV	44
5.1.1.78	renyi	44
5.1.1.79	renyi_inf	45
5.1.1.80	reshape	45
5.1.1.81	save	45
5.1.1.82	saveMATLABmatrix	45
5.1.1.83	saveMATLABmatrix	46
5.1.1.84	saveMATLABmatrix	46
5.1.1.85	schmidtcoeff	46
5.1.1.86	schmidtprob	47
5.1.1.87	schmidtU	47
5.1.1.88	schmidtV	48
5.1.1.89	shannon	48
5.1.1.90	sinm	48
5.1.1.91	spectralpowm	49
5.1.1.92	sqrtm	50
5.1.1.93	sum	51
5.1.1.94	super	52
5.1.1.95	syspermute	53
5.1.1.96	trace	53
5.1.1.97	transpose	54
5.1.1.98	tsallis	55

5.1.2	Variable Documentation	55
5.1.2.1	gt	55
5.1.2.2	rdevs	55
5.1.2.3	st	55
5.2	qpp::ct Namespace Reference	55
5.2.1	Function Documentation	55
5.2.1.1	omega	55
5.2.2	Variable Documentation	55
5.2.2.1	chop	55
5.2.2.2	ee	55
5.2.2.3	eps	55
5.2.2.4	maxn	56
5.2.2.5	pi	56
5.3	qpp::internal Namespace Reference	56
5.3.1	Function Documentation	56
5.3.1.1	_check_col_vector	56
5.3.1.2	_check_dims	56
5.3.1.3	_check_dims_match_cvect	56
5.3.1.4	_check_dims_match_mat	56
5.3.1.5	_check_dims_match_rvect	57
5.3.1.6	_check_eq_dims	57
5.3.1.7	_check_nonzero_size	57
5.3.1.8	_check_perm	57
5.3.1.9	_check_row_vector	57
5.3.1.10	_check_square_mat	57
5.3.1.11	_check_subsys_match_dims	57
5.3.1.12	_check_vector	57
5.3.1.13	_kron2	57
5.3.1.14	_multiidx2n	57
5.3.1.15	_n2multiidx	57
5.3.1.16	variadic_vector_emplace	57
5.3.1.17	variadic_vector_emplace	57
5.4	qpp::types Namespace Reference	58
5.4.1	Typedef Documentation	58
5.4.1.1	bra	58
5.4.1.2	cmat	58
5.4.1.3	cplx	58
5.4.1.4	dmat	58
5.4.1.5	DynMat	58
5.4.1.6	ket	58

6	Class Documentation	59
6.1	qpp::DiscreteDistribution Class Reference	59
6.1.1	Constructor & Destructor Documentation	59
6.1.1.1	DiscreteDistribution	59
6.1.1.2	DiscreteDistribution	59
6.1.1.3	DiscreteDistribution	59
6.1.2	Member Function Documentation	59
6.1.2.1	probabilities	59
6.1.2.2	sample	60
6.1.3	Member Data Documentation	60
6.1.3.1	_d	60
6.2	qpp::DiscreteDistributionAbsSquare Class Reference	60
6.2.1	Constructor & Destructor Documentation	61
6.2.1.1	DiscreteDistributionAbsSquare	61
6.2.1.2	DiscreteDistributionAbsSquare	61
6.2.1.3	DiscreteDistributionAbsSquare	61
6.2.1.4	DiscreteDistributionAbsSquare	61
6.2.2	Member Function Documentation	61
6.2.2.1	cplx2weights	61
6.2.2.2	probabilities	61
6.2.2.3	sample	61
6.2.3	Member Data Documentation	61
6.2.3.1	_d	61
6.3	qpp::Exception Class Reference	61
6.3.1	Member Enumeration Documentation	63
6.3.1.1	Type	63
6.3.2	Constructor & Destructor Documentation	64
6.3.2.1	Exception	64
6.3.2.2	Exception	64
6.3.3	Member Function Documentation	64
6.3.3.1	_construct_exception_msg	64
6.3.3.2	what	64
6.3.4	Member Data Documentation	64
6.3.4.1	_custom	64
6.3.4.2	_msg	64
6.3.4.3	_type	64
6.3.4.4	_where	64
6.4	qpp::Gates Class Reference	64
6.4.1	Constructor & Destructor Documentation	66
6.4.1.1	Gates	66

6.4.2	Member Function Documentation	66
6.4.2.1	apply	67
6.4.2.2	applyCTRL	67
6.4.2.3	CTRL	68
6.4.2.4	Fd	68
6.4.2.5	Id	68
6.4.2.6	Rn	68
6.4.2.7	Xd	69
6.4.2.8	Zd	69
6.4.3	Friends And Related Function Documentation	69
6.4.3.1	Singleton< const Gates >	69
6.4.4	Member Data Documentation	69
6.4.4.1	CNOTab	69
6.4.4.2	CNOTba	69
6.4.4.3	CZ	69
6.4.4.4	FRED	69
6.4.4.5	H	69
6.4.4.6	Id2	69
6.4.4.7	S	69
6.4.4.8	SWAP	69
6.4.4.9	T	69
6.4.4.10	TOF	70
6.4.4.11	X	70
6.4.4.12	Y	70
6.4.4.13	Z	70
6.5	qpp::NormalDistribution Class Reference	70
6.5.1	Constructor & Destructor Documentation	70
6.5.1.1	NormalDistribution	70
6.5.2	Member Function Documentation	70
6.5.2.1	sample	70
6.5.3	Member Data Documentation	70
6.5.3.1	_d	70
6.6	qpp::Qudit Class Reference	71
6.6.1	Constructor & Destructor Documentation	71
6.6.1.1	Qudit	71
6.6.2	Member Function Documentation	71
6.6.2.1	getD	71
6.6.2.2	getRho	71
6.6.2.3	measure	72
6.6.2.4	measure	72

6.6.3	Member Data Documentation	72
6.6.3.1	_D	72
6.6.3.2	_rho	72
6.7	qpp::RandomDevices Class Reference	73
6.7.1	Constructor & Destructor Documentation	74
6.7.1.1	RandomDevices	74
6.7.2	Friends And Related Function Documentation	74
6.7.2.1	Singleton< const RandomDevices >	74
6.7.3	Member Data Documentation	74
6.7.3.1	_rd	74
6.7.3.2	_rng	74
6.8	qpp::Singleton< T > Class Template Reference	74
6.8.1	Constructor & Destructor Documentation	74
6.8.1.1	Singleton	74
6.8.1.2	~Singleton	75
6.8.1.3	Singleton	75
6.8.2	Member Function Documentation	75
6.8.2.1	get_instance	75
6.8.2.2	operator=	75
6.9	qpp::States Class Reference	75
6.9.1	Constructor & Destructor Documentation	76
6.9.1.1	States	76
6.9.2	Friends And Related Function Documentation	76
6.9.2.1	Singleton< const States >	76
6.9.3	Member Data Documentation	76
6.9.3.1	b00	76
6.9.3.2	b01	76
6.9.3.3	b10	76
6.9.3.4	b11	77
6.9.3.5	GHZ	77
6.9.3.6	pb00	77
6.9.3.7	pb01	77
6.9.3.8	pb10	77
6.9.3.9	pb11	77
6.9.3.10	pGHZ	77
6.9.3.11	pW	77
6.9.3.12	px0	77
6.9.3.13	px1	77
6.9.3.14	py0	77
6.9.3.15	py1	77

6.9.3.16	pz0	77
6.9.3.17	pz1	77
6.9.3.18	W	77
6.9.3.19	x0	77
6.9.3.20	x1	77
6.9.3.21	y0	77
6.9.3.22	y1	77
6.9.3.23	z0	77
6.9.3.24	z1	77
6.10	qpp::Timer Class Reference	77
6.10.1	Constructor & Destructor Documentation	78
6.10.1.1	Timer	78
6.10.2	Member Function Documentation	78
6.10.2.1	seconds	78
6.10.2.2	tic	78
6.10.2.3	toc	78
6.10.3	Friends And Related Function Documentation	78
6.10.3.1	operator<<	78
6.10.4	Member Data Documentation	78
6.10.4.1	_end	78
6.10.4.2	_start	78
6.11	qpp::UniformIntDistribution Class Reference	78
6.11.1	Constructor & Destructor Documentation	79
6.11.1.1	UniformIntDistribution	79
6.11.2	Member Function Documentation	79
6.11.2.1	sample	79
6.11.3	Member Data Documentation	79
6.11.3.1	_d	79
6.12	qpp::UniformRealDistribution Class Reference	79
6.12.1	Constructor & Destructor Documentation	79
6.12.1.1	UniformRealDistribution	79
6.12.2	Member Function Documentation	79
6.12.2.1	sample	80
6.12.3	Member Data Documentation	80
6.12.3.1	_d	80
7	File Documentation	81
7.1	include/channels.h File Reference	81
7.2	include/classes/exception.h File Reference	82
7.3	include/classes/gates.h File Reference	82

7.4	include/classes/qudit.h File Reference	83
7.5	include/classes/randevs.h File Reference	83
7.6	include/classes/singleton.h File Reference	84
7.6.1	Macro Definition Documentation	84
7.6.1.1	CLASS_CONST_SINGLETON	84
7.6.1.2	CLASS_SINGLETON	84
7.7	include/classes/stat.h File Reference	85
7.8	include/classes/states.h File Reference	85
7.9	include/classes/timer.h File Reference	86
7.10	include/constants.h File Reference	86
7.11	include/entanglement.h File Reference	87
7.12	include/entropies.h File Reference	88
7.13	include/functions.h File Reference	89
7.14	include/internal.h File Reference	92
7.15	include/io.h File Reference	93
7.16	include/matlab.h File Reference	94
7.17	include/qpp.h File Reference	95
7.18	include/random.h File Reference	96
7.19	include/types.h File Reference	97
	Index	98

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

qpp	9
qpp::ct	55
qpp::internal	56
qpp::types	58

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

qpp::DiscreteDistribution	59
qpp::DiscreteDistributionAbsSquare	60
exception	
qpp::Exception	61
qpp::NormalDistribution	70
qpp::Qudit	71
qpp::Singleton< T >	74
qpp::Gates	64
qpp::RandomDevices	73
qpp::Singleton< const Gates >	74
qpp::Singleton< const RandomDevices >	74
qpp::Singleton< const States >	74
qpp::States	75
qpp::Timer	77
qpp::UniformIntDistribution	78
qpp::UniformRealDistribution	79

Chapter 3

Class Index

3.1 Class List

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

qpp::DiscreteDistribution	59
qpp::DiscreteDistributionAbsSquare	60
qpp::Exception	61
qpp::Gates	64
qpp::NormalDistribution	70
qpp::Qudit	71
qpp::RandomDevices	73
qpp::Singleton< T >	74
qpp::States	75
qpp::Timer	77
qpp::UniformIntDistribution	78
qpp::UniformRealDistribution	79

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

include/channels.h	81
include/constants.h	86
include/entanglement.h	87
include/entropies.h	88
include/functions.h	89
include/internal.h	92
include/io.h	93
include/matlab.h	94
include/qpp.h	95
include/random.h	96
include/types.h	97
include/classes/exception.h	82
include/classes/gates.h	82
include/classes/qudit.h	83
include/classes/randevs.h	83
include/classes/singleton.h	84
include/classes/stat.h	85
include/classes/states.h	85
include/classes/timer.h	86

Chapter 5

Namespace Documentation

5.1 qpp Namespace Reference

Namespaces

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

Classes

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

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< std::size_t > &subsys, const std::vector< std::size_t > &dims)
- constexpr std::complex< double > [operator""_i](#) (unsigned long long int x)
- constexpr std::complex< double > [operator""_i](#) (long double x)
- template<typename Derived >
[types::cmat schmidtcoeff](#) (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t > &dims)

- `template<typename Derived >`
`types::cmat schmidtU` (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t > &dims)
- `template<typename Derived >`
`types::cmat schmidtV` (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t > &dims)
- `template<typename Derived >`
`types::cmat schmidtprob` (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t > &dims)
- `template<typename Derived >`
`double entanglement` (const Eigen::MatrixBase< Derived > &A, const std::vector< std::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< std::size_t > &subsys, const std::vector< std::size_t > &dims)
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > transpose` (const Eigen::MatrixBase< Derived > &A)
Transpose.
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > conjugate` (const Eigen::MatrixBase< Derived > &A)
Complex conjugate.
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > adjoint` (const Eigen::MatrixBase< Derived > &A)
Adjoint.
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > inverse` (const Eigen::MatrixBase< Derived > &A)
Inverse.
- `template<typename Derived >`
`Derived::Scalar trace` (const Eigen::MatrixBase< Derived > &A)
Trace.
- `template<typename Derived >`
`Derived::Scalar det` (const Eigen::MatrixBase< Derived > &A)
Determinant.
- `template<typename Derived >`
`Derived::Scalar logdet` (const Eigen::MatrixBase< Derived > &A)
Logarithm of the determinant.
- `template<typename Derived >`
`Derived::Scalar sum` (const Eigen::MatrixBase< Derived > &A)
Element-wise sum.
- `template<typename Derived >`
`double norm` (const Eigen::MatrixBase< Derived > &A)
Trace norm.
- `template<typename Derived >`
`types::cmat evals` (const Eigen::MatrixBase< Derived > &A)
Eigenvalues.

- `template<typename Derived >`
`types::cmat evecs` (const Eigen::MatrixBase< Derived > &A)
Eigenvectors.
- `template<typename Derived >`
`types::dmat hevals` (const Eigen::MatrixBase< Derived > &A)
Hermitian eigenvalues.
- `template<typename Derived >`
`types::cmat hevecs` (const Eigen::MatrixBase< Derived > &A)
Hermitian eigenvectors.
- `template<typename Derived >`
`types::cmat funm` (const Eigen::MatrixBase< Derived > &A, `types::cplx`(*f)(const `types::cplx` &))
Functional calculus $f(A)$
- `template<typename Derived >`
`types::cmat sqrtm` (const Eigen::MatrixBase< Derived > &A)
Matrix square root.
- `template<typename Derived >`
`types::cmat absm` (const Eigen::MatrixBase< Derived > &A)
Matrix absolut value.
- `template<typename Derived >`
`types::cmat expm` (const Eigen::MatrixBase< Derived > &A)
Matrix exponential.
- `template<typename Derived >`
`types::cmat logm` (const Eigen::MatrixBase< Derived > &A)
Matrix logarithm.
- `template<typename Derived >`
`types::cmat sinm` (const Eigen::MatrixBase< Derived > &A)
Matrix sin.
- `template<typename Derived >`
`types::cmat cosm` (const Eigen::MatrixBase< Derived > &A)
Matrix cos.
- `template<typename Derived >`
`types::cmat spectralpowm` (const Eigen::MatrixBase< Derived > &A, const `types::cplx` z)
Matrix power.
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > powm` (const Eigen::MatrixBase< Derived > &A, std::size_t n)
Matrix power.
- `template<typename OutputScalar , typename Derived >`
`types::DynMat< OutputScalar > cwise` (const Eigen::MatrixBase< Derived > &A, OutputScalar(*f)(const
typename Derived::Scalar &))
Functor.
- `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, std::size_t n)

- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > reshape` (const Eigen::MatrixBase< Derived > &A, std::size_t rows, std::size_t cols)
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > syspermute` (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t > &perm, const std::vector< std::size_t > &dims)
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > ptrace1` (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t > &dims)
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > ptrace2` (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t > &dims)
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > ptrace` (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t > &subsys, const std::vector< std::size_t > &dims)
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > ptranspose` (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t > &subsys, const std::vector< std::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< Derived2 > &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, std::size_t pos, const std::vector< std::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< std::size_t > n2multiidx` (std::size_t n, const std::vector< std::size_t > &dims)
- `std::size_t multiidx2n` (const std::vector< std::size_t > &midx, const std::vector< std::size_t > &dims)
- `types::ket mket` (const std::vector< std::size_t > &mask)
- `types::ket mket` (const std::vector< std::size_t > &mask, const std::vector< std::size_t > &dims)
- `types::ket mket` (const std::vector< std::size_t > &mask, std::size_t d)
- `std::vector< std::size_t > invperm` (const std::vector< std::size_t > &perm)
- `std::vector< std::size_t > compperm` (const std::vector< std::size_t > &perm, const std::vector< std::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 std::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 std::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< 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 (std::size_t rows, std::size_t cols, double a=0, double b=1)`
- `template<>`
`types::dmat rand (std::size_t rows, std::size_t cols, double a, double b)`
- `template<>`
`types::cmat rand (std::size_t rows, std::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 (std::size_t rows, std::size_t cols, double mean=0, double sigma=1)`
- `template<>`
`types::dmat randn (std::size_t rows, std::size_t cols, double mean, double sigma)`
- `template<>`
`types::cmat randn (std::size_t rows, std::size_t cols, double mean, double sigma)`
- `double randn (double mean=0, double sigma=1)`
- `types::cmat randU (std::size_t D)`
- `types::cmat randV (std::size_t Din, std::size_t Dout)`
- `std::vector< types::cmat > randkraus (std::size_t n, std::size_t D)`
- `types::cmat randH (std::size_t D)`
- `types::ket randket (std::size_t D)`
- `types::cmat randrho (std::size_t D)`
- `std::vector< std::size_t > randperm (std::size_t n)`

Variables

- const [RandomDevices](#) & rdevs = [RandomDevices::get_instance\(\)](#)
- const [Gates](#) & gt = [Gates::get_instance\(\)](#)
- const [States](#) & st = [States::get_instance\(\)](#)

5.1.1 Function Documentation

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

Matrix absolute value.

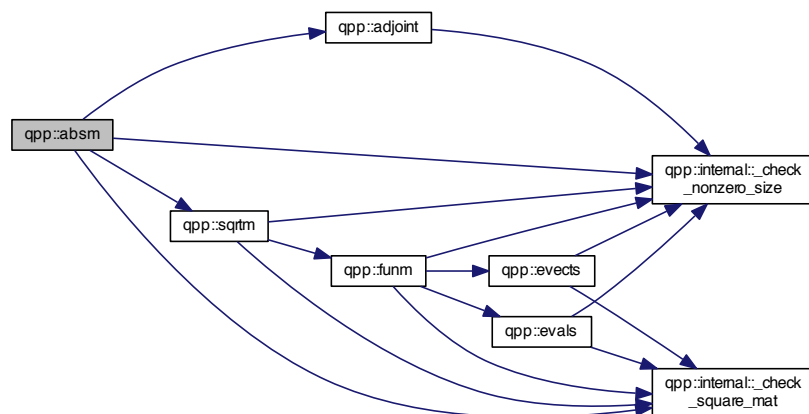
Parameters

<i>A</i>	Eigen expression
----------	------------------

Returns

Matrix absolut value of A, as a dynamic matrix over the complex field

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)`

Adjoint.

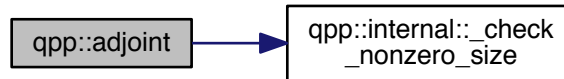
Parameters

<i>A</i>	Eigen expression
----------	------------------

Returns

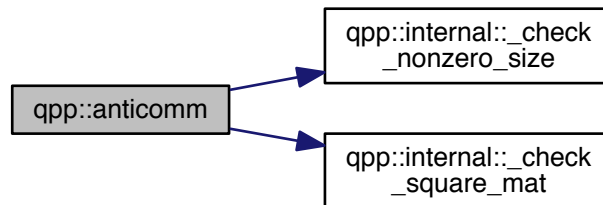
Adjoint (Hermitian conjugate) of A , as a dynamic matrix over the same scalar field

Here is the call graph for this function:



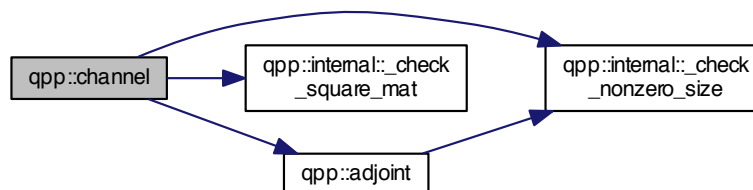
5.1.1.3 `template<typename Derived1 , typename Derived2 > types::DynMat<typename Derived1::Scalar> qpp::anticomm (const Eigen::MatrixBase< Derived1 > & A, const Eigen::MatrixBase< Derived2 > & B)`

Here is the call graph for this function:



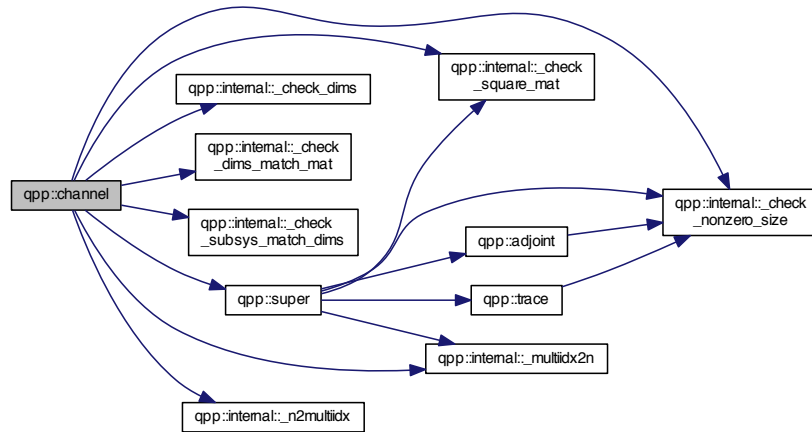
5.1.1.4 `template<typename Derived > types::cmat qpp::channel (const Eigen::MatrixBase< Derived > & rho, const std::vector< types::cmat > & Ks)`

Here is the call graph for this function:



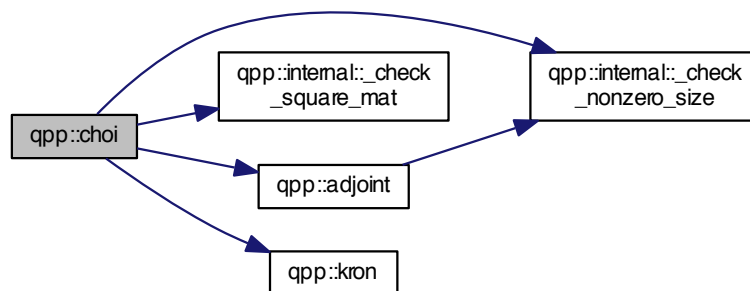
5.1.1.5 `template<typename Derived > types::cmat qpp::channel (const Eigen::MatrixBase< Derived > & rho, const std::vector< types::cmat > & Ks, const std::vector< std::size_t > & subsys, const std::vector< std::size_t > & dims)`

Here is the call graph for this function:



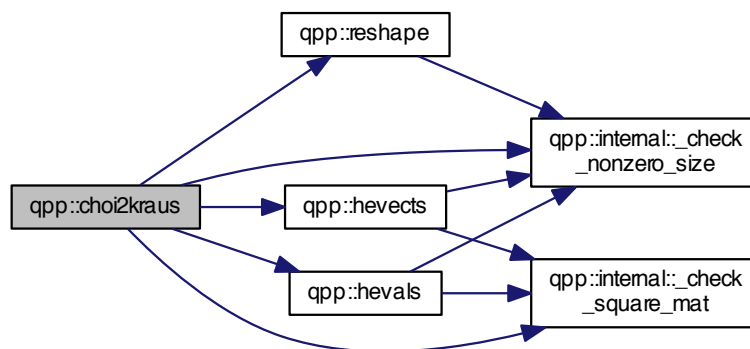
5.1.1.6 `types::cmat qpp::choi (const std::vector< types::cmat > & Ks)`

Here is the call graph for this function:



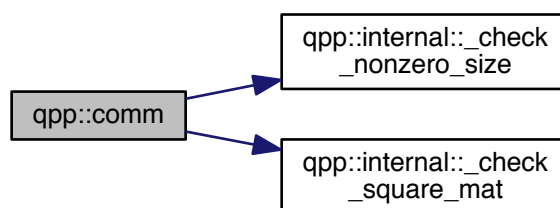
5.1.1.7 `std::vector<types::cmat> qpp::choi2kraus (const types::cmat & A)`

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

Complex conjugate.

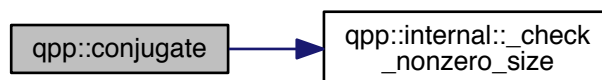
Parameters

A	Eigen expression
---	------------------

Returns

Complex conjugate of A, as a dynamic matrix over the same scalar field

Here is the call graph for this function:



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

Matrix cos.

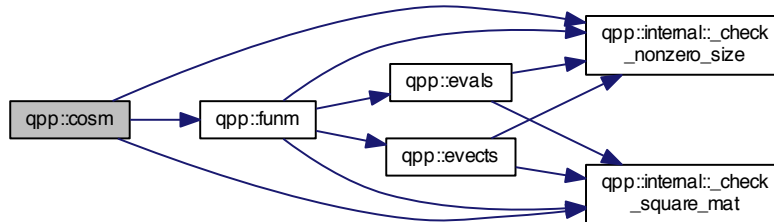
Parameters

A	Eigen expression
---	------------------

Returns

Matrix cosine of A , as a dynamic matrix over the complex field

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)`

Functor.

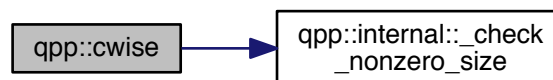
Parameters

A	Eigen expression
f	Pointer-to-function from scalars of A to OutputScalar

Returns

Component-wise $f(A)$, as a dynamic matrix over OutputScalar scalar field

Here is the call graph for this function:



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

Determinant.

Parameters

A	Eigen expression
-----	------------------

Returns

Determinant of A , as a dynamic matrix over the same scalar field

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 std::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 std::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:



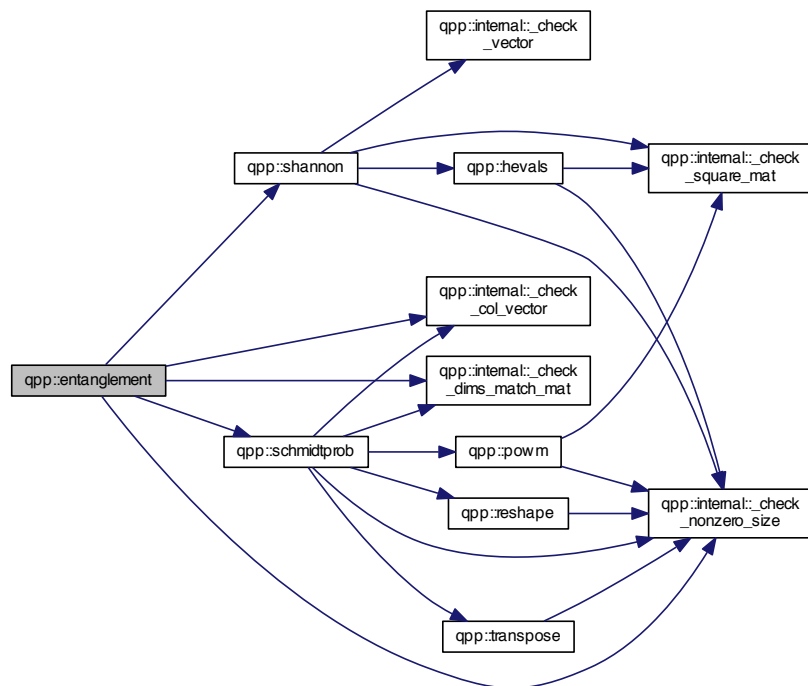
5.1.1.21 `void qpp::displn (const types::cplx c, double chop = ct:::chop, std::ostream & os = std:::cout)`

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< std::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)`

Eigenvalues.

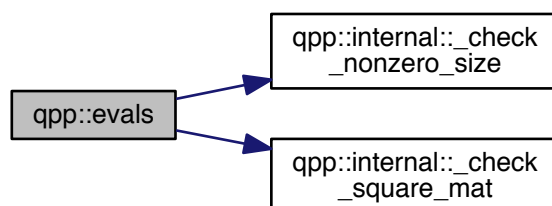
Parameters

A	Eigen expression
-----	------------------

Returns

Eigenvalues of A , as a dynamic matrix over the complex field

Here is the call graph for this function:



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

Eigenvectors.

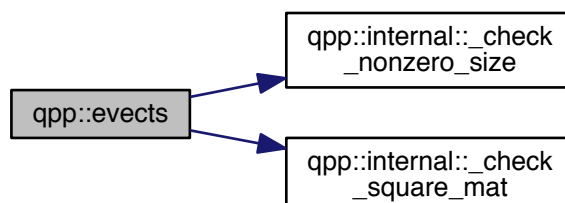
Parameters

A	Eigen expression
-----	------------------

Returns

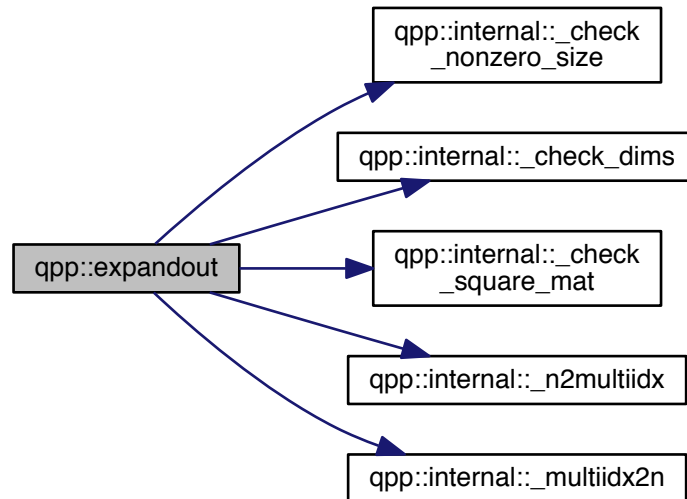
Eigenvectors of A , as a dynamic matrix over the complex field

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, std::size_t pos, const std::vector< std::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)`

Matrix exponential.

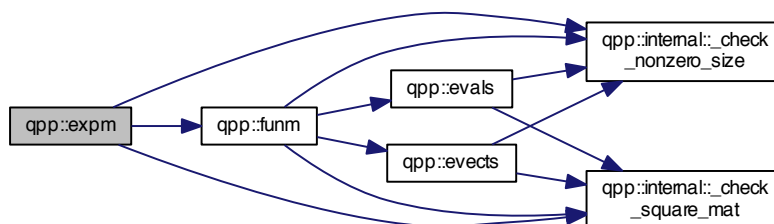
Parameters

A	Eigen expression
---	------------------

Returns

Matrix exponential of A, as a dynamic matrix over the complex field

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)`

Functional calculus $f(A)$

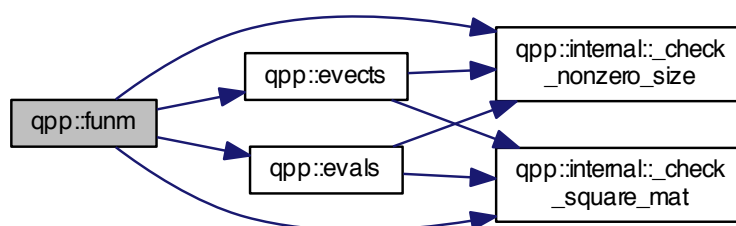
Parameters

A	Eigen expression
f	Pointer-to-function from complex to complex

Returns

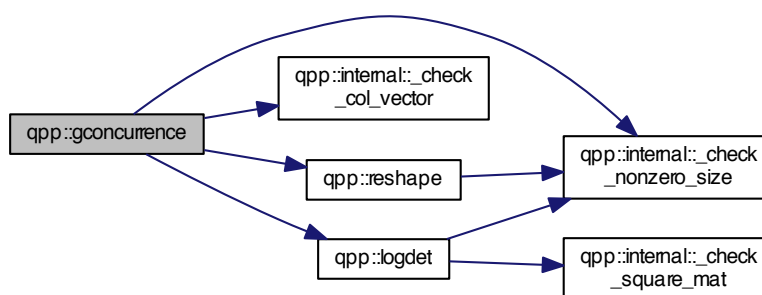
$f(A)$, as a dynamic matrix over the complex field

Here is the call graph for this function:



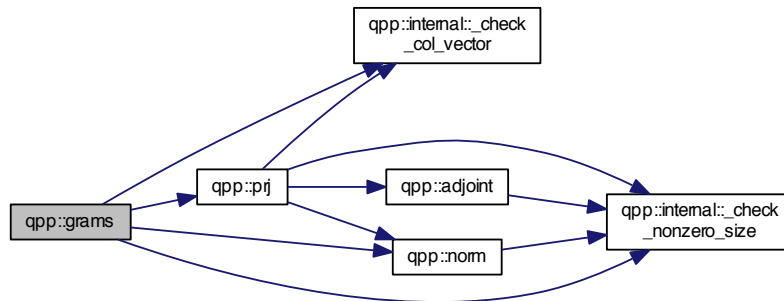
5.1.1.28 `template<typename Derived> double qpp::gconcurrency (const Eigen::MatrixBase< Derived> & A)`

Here is the call graph for this function:



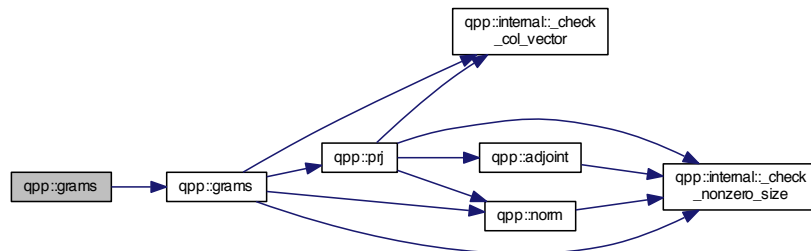
5.1.1.29 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::grams (const std::vector< Derived > & Vs)`

Here is the call graph for this function:



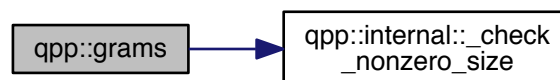
5.1.1.30 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::grams (const std::initializer_list< Derived > & Vs)`

Here is the call graph for this function:



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

Here is the call graph for this function:



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

Hermitian eigenvalues.

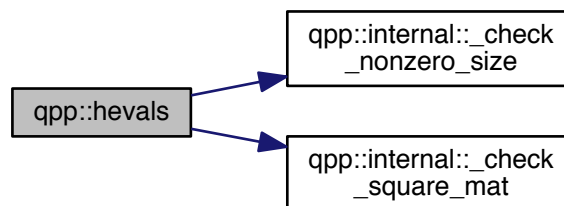
Parameters

A	Eigen expression
---	------------------

Returns

Eigenvalues of Hermitian A, as a dynamic matrix over the real field

Here is the call graph for this function:



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

Hermitian eigenvectors.

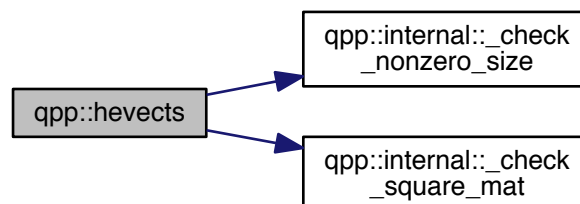
Parameters

A	Eigen expression
---	------------------

Returns

Eigenvectors of Hermitian A, as a dynamic matrix over the complex field

Here is the call graph for this function:



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

Inverse.

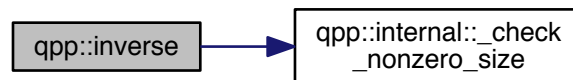
Parameters

A	Eigen expression
-----	------------------

Returns

Inverse of A , as a dynamic matrix over the same scalar field

Here is the call graph for this function:



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

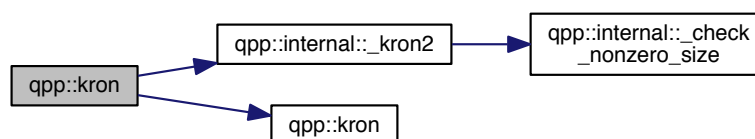
Here is the call graph for this function:



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

5.1.1.37 `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:



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

Here is the call graph for this function:



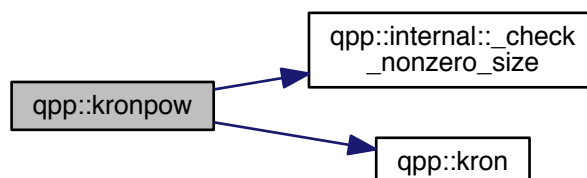
5.1.1.39 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::kron (const std::initializer_list<Derived> & As)`

Here is the call graph for this function:



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

Here is the call graph for this function:



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

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

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

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

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

Logarithm of the determinant.

Especially useful when determinant overflows/underflows

Returns +/- inf when determinant overflows/underflows

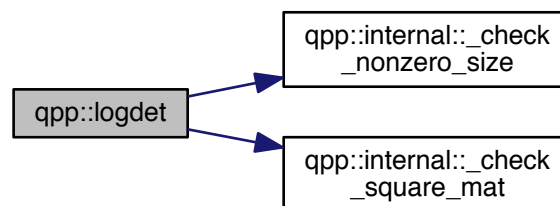
Parameters

<i>A</i>	Eigen expression
----------	------------------

Returns

Logarithm of the determinant of *A*, as a dynamic matrix over the same scalar field

Here is the call graph for this function:



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

Matrix logarithm.

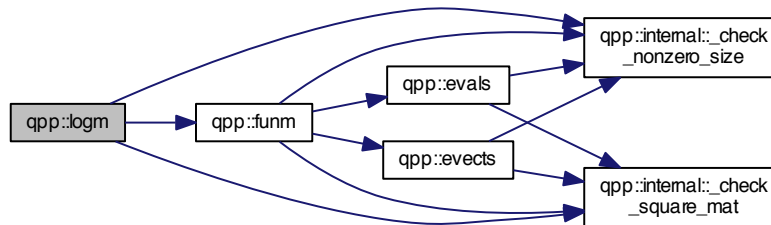
Parameters

<i>A</i>	Eigen expression
----------	------------------

Returns

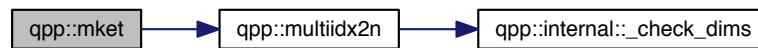
Matrix logarithm of A, as a dynamic matrix over the complex field

Here is the call graph for this function:



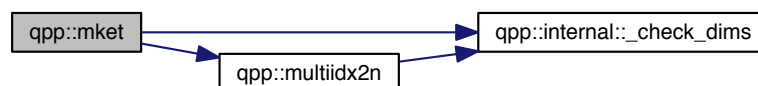
5.1.1.47 `types::ket qpp::mket (const std::vector< std::size_t > & mask)`

Here is the call graph for this function:



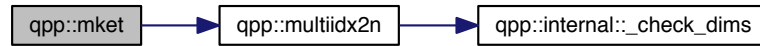
5.1.1.48 `types::ket qpp::mket (const std::vector< std::size_t > & mask, const std::vector< std::size_t > & dims)`

Here is the call graph for this function:



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

Here is the call graph for this function:



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

Here is the call graph for this function:



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

Here is the call graph for this function:



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

Trace norm.

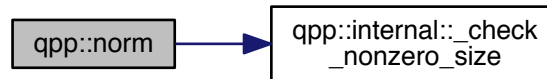
Parameters

<i>A</i>	Eigen expression
----------	------------------

Returns

Trace norm (or Frobenius norm) of A, as a real number

Here is the call graph for this function:



5.1.1.53 `constexpr std::complex<double> qpp::operator""_i (unsigned long long int x)`

5.1.1.54 `constexpr std::complex<double> qpp::operator""_i (long double x)`

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

Matrix power.

Defines $A^0 = \text{Identity}$

Explicitly multiplies the matrix with itself n times

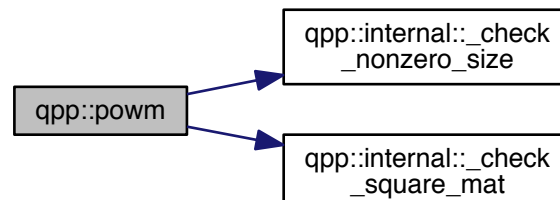
Parameters

A	Eigen expression
n	non-negative integer

Returns

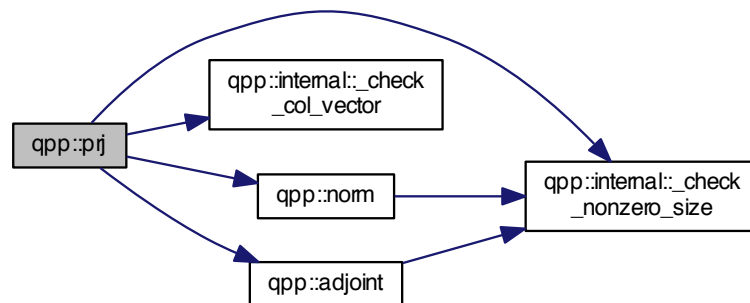
Matrix power of A, as a dynamic matrix over the same scalar field

Here is the call graph for this function:



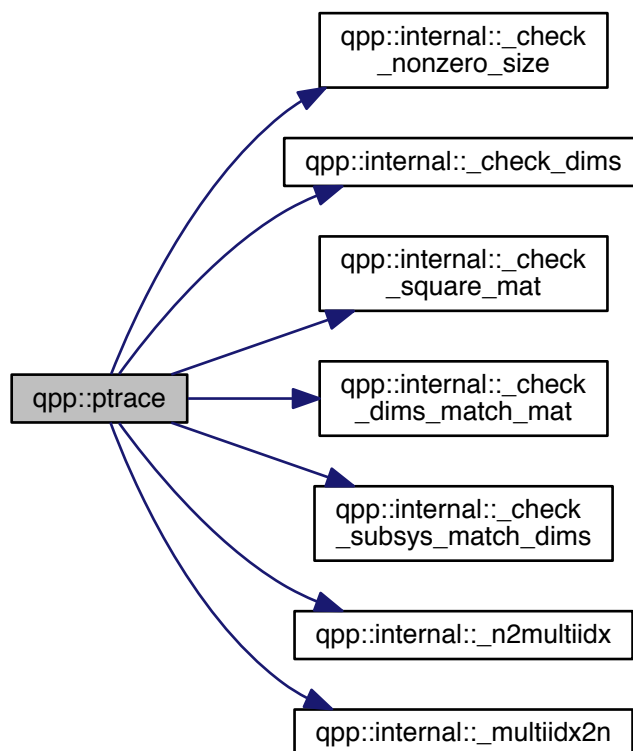
5.1.1.56 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::prj (const Eigen::MatrixBase<Derived> & V)`

Here is the call graph for this function:



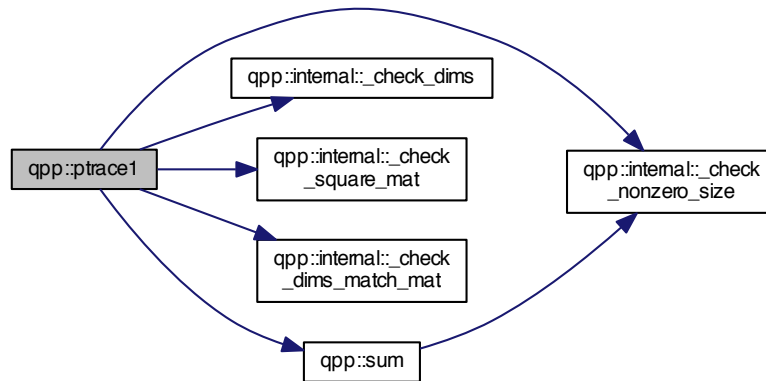
5.1.1.57 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::ptrace (const Eigen::MatrixBase< Derived> & A, const std::vector< std::size_t> & subsys, const std::vector< std::size_t> & dims)`

Here is the call graph for this function:



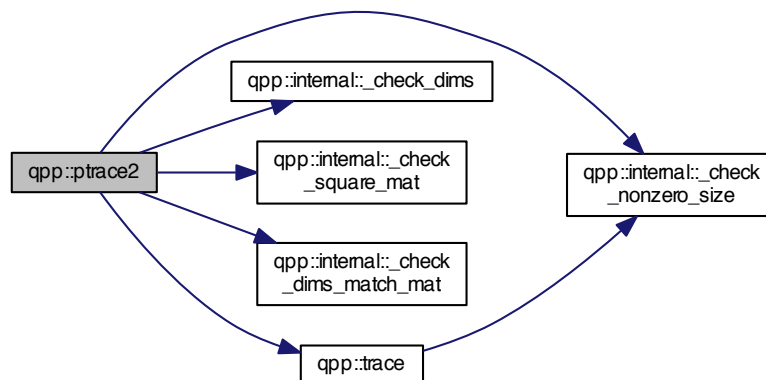
5.1.1.58 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::ptrace1 (const Eigen::MatrixBase< Derived > & A, const std::vector< std::size_t > & dims)`

Here is the call graph for this function:



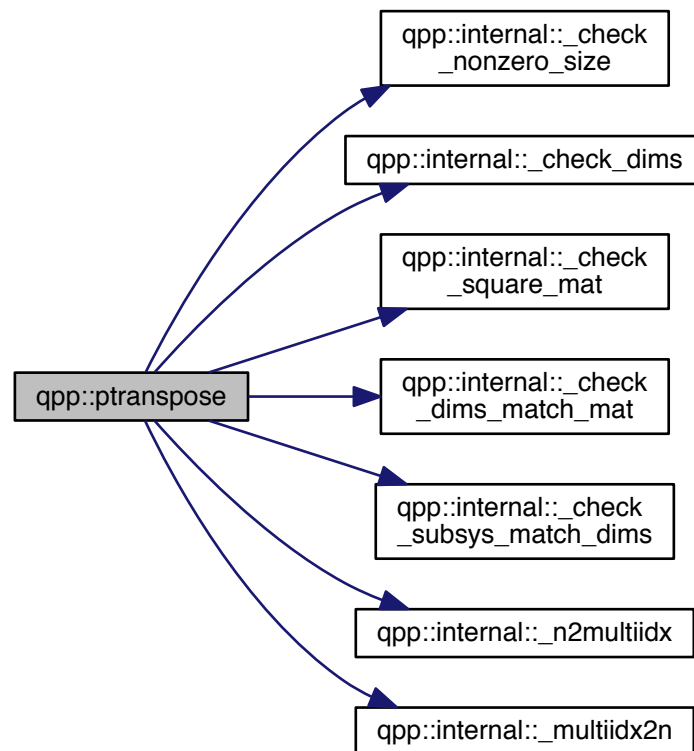
5.1.1.59 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::ptrace2 (const Eigen::MatrixBase< Derived > & A, const std::vector< std::size_t > & dims)`

Here is the call graph for this function:



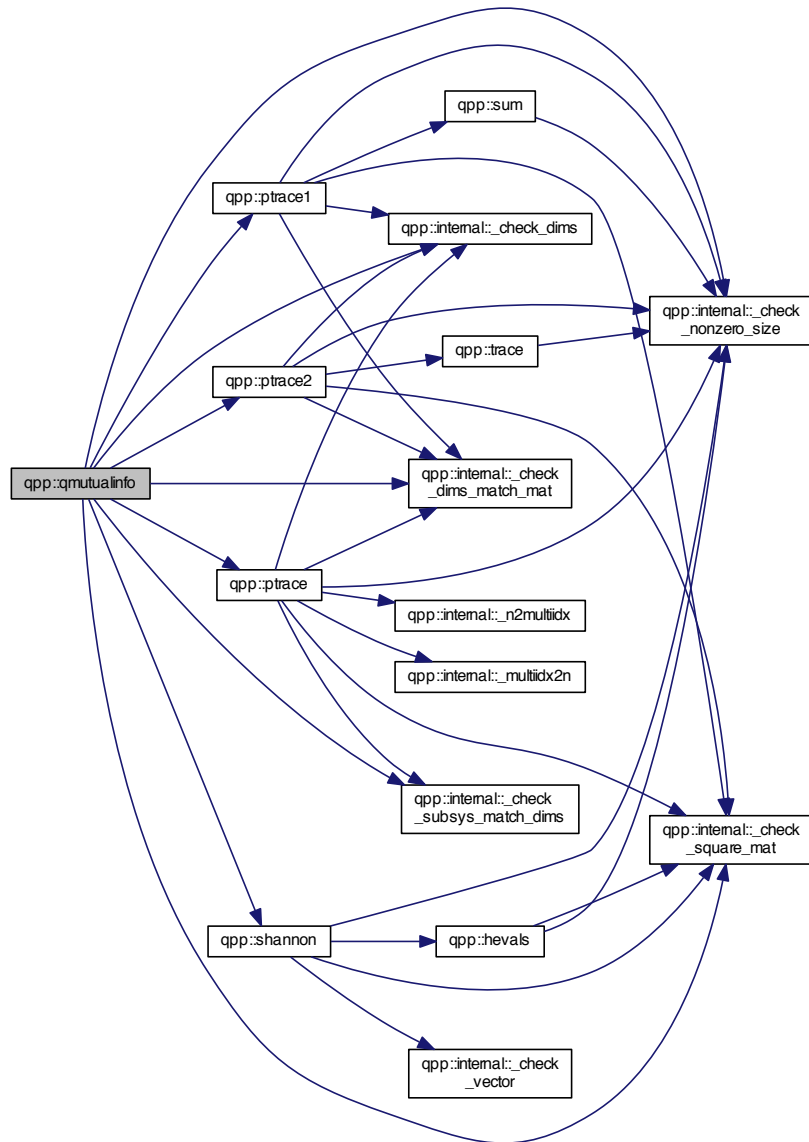
5.1.1.60 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::ptranspose (const Eigen::MatrixBase< Derived> & A, const std::vector< std::size_t> & subsys, const std::vector< std::size_t> & dims)`

Here is the call graph for this function:



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

Here is the call graph for this function:



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

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

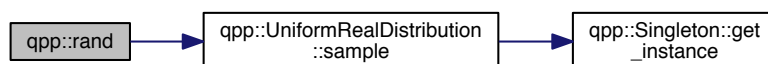
5.1.1.64 `template<> types::cmat qpp::rand (std::size_t rows, std::size_t cols, double a, double b)`

Here is the call graph for this function:



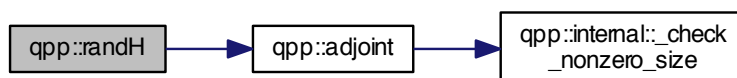
5.1.1.65 `double qpp::rand (double a = 0, double b = 1)`

Here is the call graph for this function:



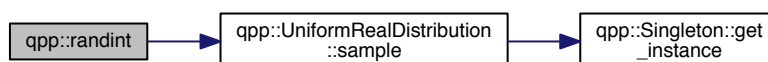
5.1.1.66 `types::cmat qpp::randH (std::size_t D)`

Here is the call graph for this function:



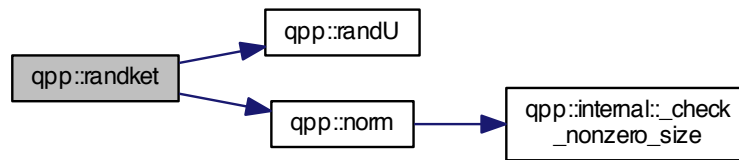
5.1.1.67 `long long qpp::randint (long long a, long long b)`

Here is the call graph for this function:



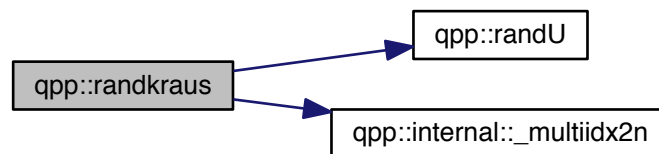
5.1.1.68 `types::ket qpp::randket (std::size_t D)`

Here is the call graph for this function:



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

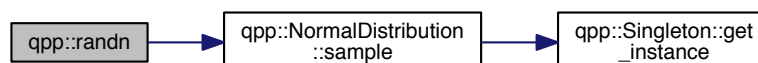
Here is the call graph for this function:



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

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

Here is the call graph for this function:



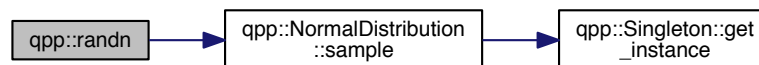
5.1.1.72 `template<> types::cmat qpp::randn (std::size_t rows, std::size_t cols, double mean, double sigma)`

Here is the call graph for this function:



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

Here is the call graph for this function:



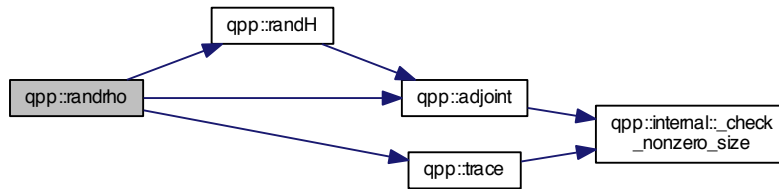
5.1.1.74 `std::vector<std::size_t> qpp::randperm (std::size_t n)`

Here is the call graph for this function:



5.1.1.75 `types::cmat qpp::randrho (std::size_t D)`

Here is the call graph for this function:



5.1.1.76 `types::cmat qpp::randU (std::size_t D)`

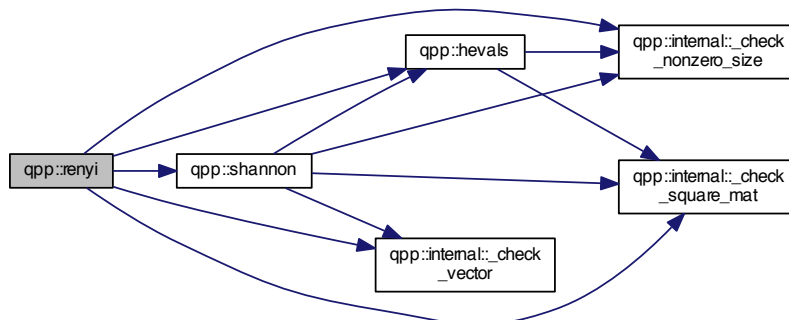
5.1.1.77 `types::cmat qpp::randV (std::size_t Din, std::size_t Dout)`

Here is the call graph for this function:



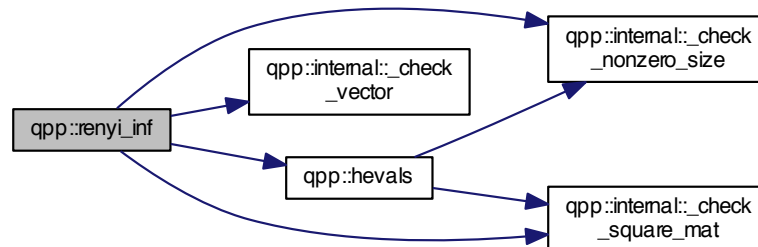
5.1.1.78 `template<typename Derived> double qpp::renyi (const double alpha, const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



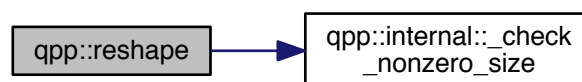
5.1.1.79 `template<typename Derived> double qpp::renyi_inf (const Eigen::MatrixBase< Derived > & A)`

Here is the call graph for this function:



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

Here is the call graph for this function:

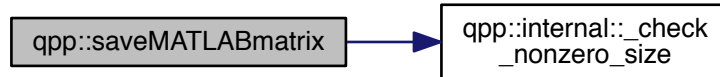


5.1.1.81 `template<typename Derived> void qpp::save (const Eigen::MatrixBase< Derived > & A, const std::string & fname)`

5.1.1.82 `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)`

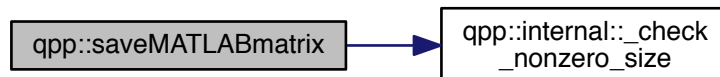
5.1.1.83 `template<> void qpp::saveMATLABmatrix (const Eigen::MatrixBase< types::dmat > & A, const std::string & mat_file, const std::string & var_name, const std::string & mode)`

Here is the call graph for this function:



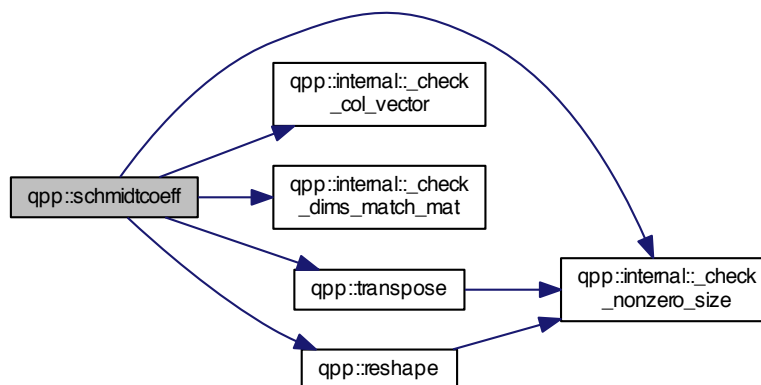
5.1.1.84 `template<> void qpp::saveMATLABmatrix (const Eigen::MatrixBase< typename types::cmat > & A, const std::string & mat_file, const std::string & var_name, const std::string & mode)`

Here is the call graph for this function:



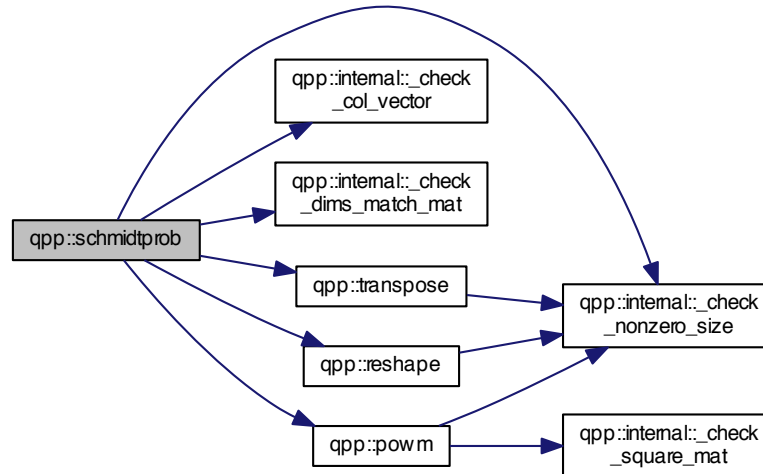
5.1.1.85 `template<typename Derived> types::cmat qpp::schmidtcoeff (const Eigen::MatrixBase< Derived > & A, const std::vector< std::size_t > & dims)`

Here is the call graph for this function:



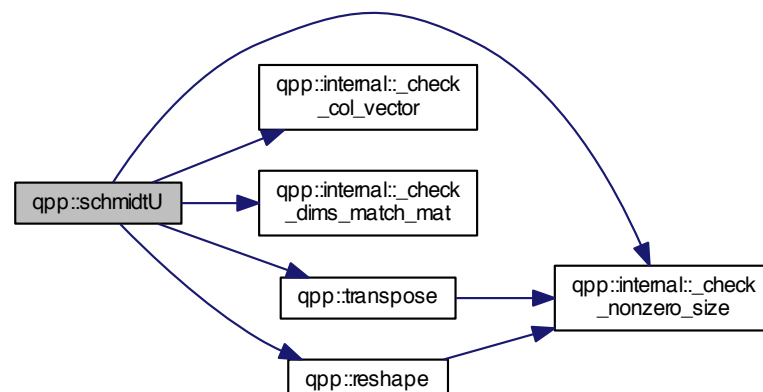
5.1.1.86 `template<typename Derived> types::cmat qpp::schmidtprob (const Eigen::MatrixBase< Derived> & A, const std::vector< std::size_t> & dims)`

Here is the call graph for this function:



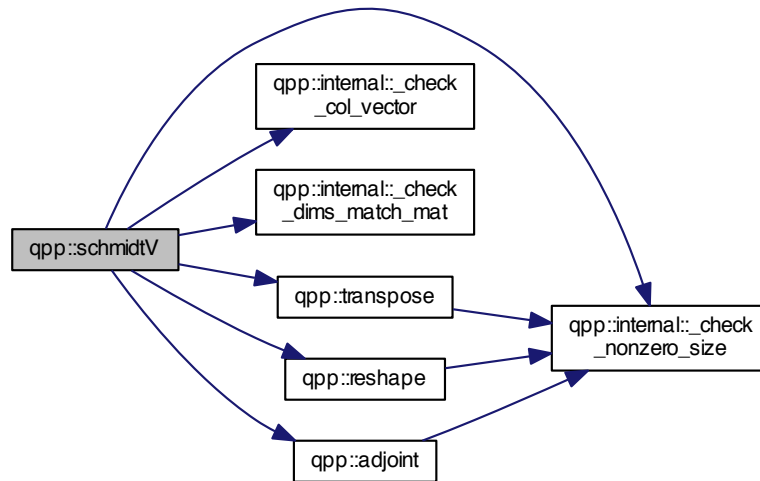
5.1.1.87 `template<typename Derived> types::cmat qpp::schmidtU (const Eigen::MatrixBase< Derived> & A, const std::vector< std::size_t> & dims)`

Here is the call graph for this function:



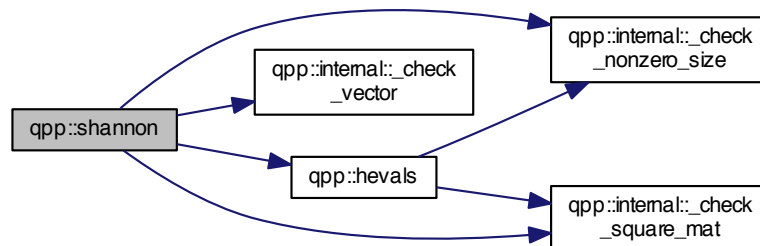
5.1.1.88 `template<typename Derived> types::cmat qpp::schmidtV (const Eigen::MatrixBase< Derived> & A, const std::vector< std::size_t> & dims)`

Here is the call graph for this function:



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

Here is the call graph for this function:



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

Matrix sin.

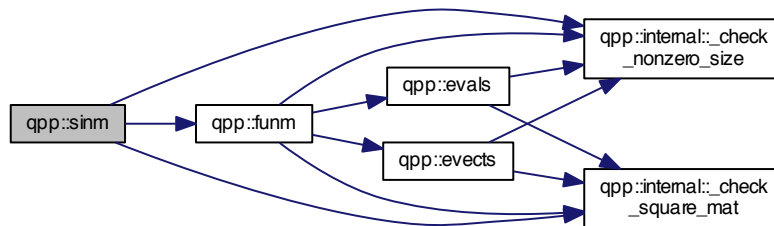
Parameters

A	Eigen expression
-----	------------------

Returns

Matrix sine of A , as a dynamic matrix over the complex field

Here is the call graph for this function:



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

Matrix power.

Defines $A^0 = \text{Identity}$

Uses the spectral decomposition to compute the matrix power

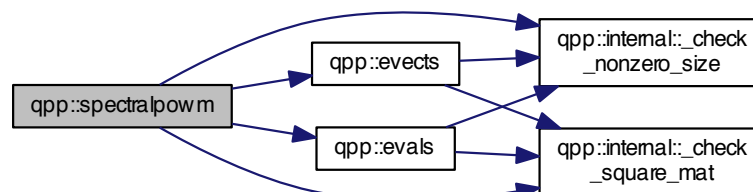
Parameters

A	Eigen expression
z	complex number

Returns

Matrix power of A , as a dynamic matrix over the complex field

Here is the call graph for this function:



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

Matrix square root.

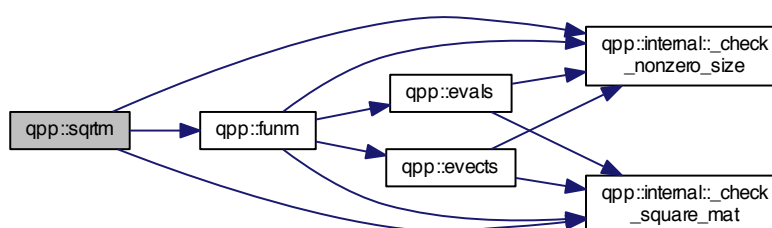
Parameters

<i>A</i>	Eigen expression
----------	------------------

Returns

Matrix square root of *A*, as a dynamic matrix over the complex field

Here is the call graph for this function:



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

Element-wise sum.

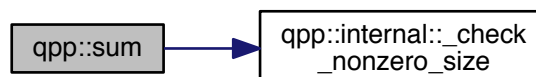
Parameters

<i>A</i>	Eigen expression
----------	------------------

Returns

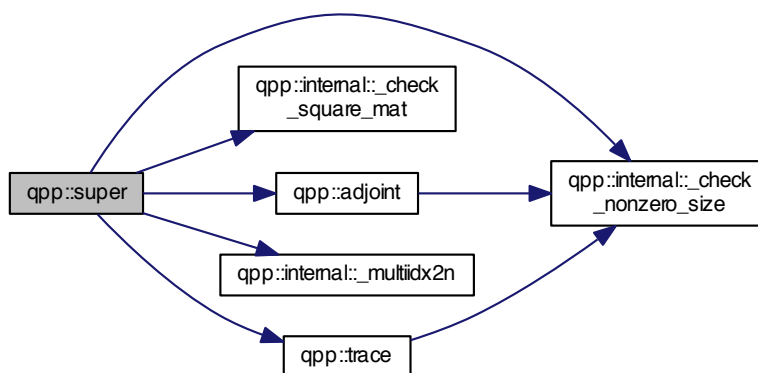
Element-wise sum of *A*, as a dynamic matrix over the same scalar field

Here is the call graph for this function:



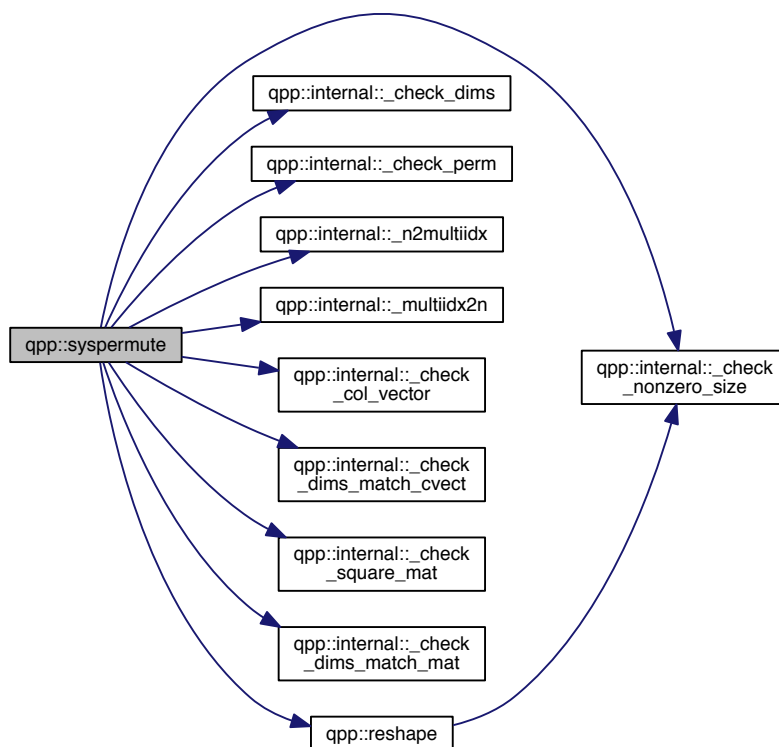
5.1.1.94 `types::cmat qpp::super (const std::vector< types::cmat > & Ks)`

Here is the call graph for this function:



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

Here is the call graph for this function:



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

Trace.

Parameters

A	Eigen expression
---	------------------

Returns

Trace of A, as a dynamic matrix over the same scalar field

Here is the call graph for this function:



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

Transpose.

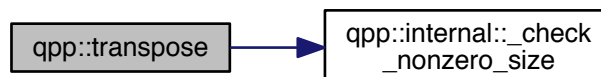
Parameters

<i>A</i>	Eigen expression
----------	------------------

Returns

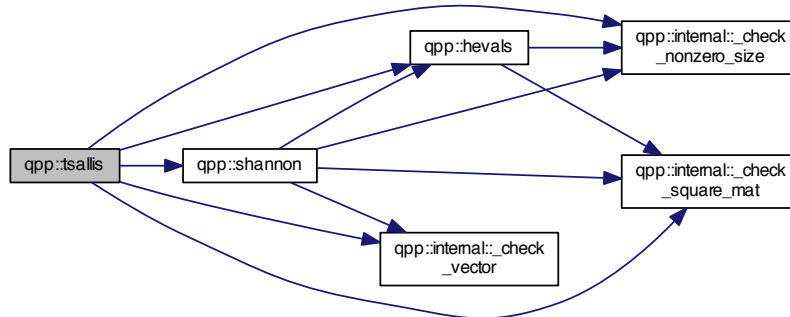
Transpose of A, as a dynamic matrix over the same scalar field

Here is the call graph for this function:



5.1.1.98 `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::get_instance()`

5.1.2.2 `const RandomDevices& qpp::rdevs = RandomDevices::get_instance()`

5.1.2.3 `const States& qpp::st = States::get_instance()`

5.2 qpp::ct Namespace Reference

Functions

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

Variables

- `constexpr double chop = 1e-10`
- `constexpr double eps = 1e-12`
- `constexpr std::size_t maxn = 64`
- `constexpr double pi = 3.141592653589793238462643383279502884`
- `constexpr double ee = 2.718281828459045235360287471352662497`

5.2.1 Function Documentation

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

5.2.2 Variable Documentation

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

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

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

5.2.2.4 constexpr std::size_t qpp::ct::maxn = 64

5.2.2.5 constexpr double qpp::ct::pi = 3.141592653589793238462643383279502884

5.3 qpp::internal Namespace Reference

Functions

- void [_n2multiidx](#) (std::size_t n, std::size_t numdims, const std::size_t *dims, std::size_t *result)
- std::size_t [_multiidx2n](#) (const std::size_t *midx, std::size_t numdims, const std::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< std::size_t > &dims)
- template<typename Derived >
bool [_check_dims_match_mat](#) (const std::vector< std::size_t > &dims, const Eigen::MatrixBase< Derived > &A)
- template<typename Derived >
bool [_check_dims_match_cvect](#) (const std::vector< std::size_t > &dims, const Eigen::MatrixBase< Derived > &V)
- template<typename Derived >
bool [_check_dims_match_rvect](#) (const std::vector< std::size_t > &dims, const Eigen::MatrixBase< Derived > &V)
- bool [_check_eq_dims](#) (const std::vector< std::size_t > &dims, std::size_t dim)
- bool [_check_subsys_match_dims](#) (const std::vector< std::size_t > &subsys, const std::vector< std::size_t > &dims)
- bool [_check_perm](#) (const std::vector< std::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)

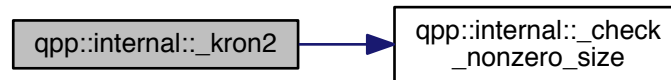
5.3.1.2 bool qpp::internal::_check_dims (const std::vector< std::size_t > & dims)

5.3.1.3 template<typename Derived > bool qpp::internal::_check_dims_match_cvect (const std::vector< std::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< std::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< std::size_t > & dims, const Eigen::MatrixBase< Derived > & V)`
- 5.3.1.6 `bool qpp::internal::_check_eq_dims (const std::vector< std::size_t > & dims, std::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< std::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< std::size_t > & subsys, const std::vector< std::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 `std::size_t qpp::internal::_multiidx2n (const std::size_t * midx, std::size_t numdims, const std::size_t * dims)`
- 5.3.1.15 `void qpp::internal::_n2multiidx (std::size_t n, std::size_t numdims, const std::size_t * dims, std::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)
- `std::size_t sample` ()
- `std::vector< double > probabilities` () const

Protected Attributes

- `std::discrete_distribution`
< std::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]

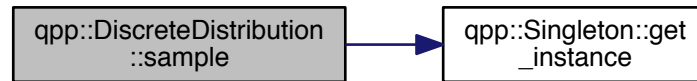
6.1.1.3 `qpp::DiscreteDistribution::DiscreteDistribution (std::vector< double > weights)` [inline]

6.1.2 Member Function Documentation

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

6.1.2.2 `std::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<std::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)
- `template<typename Derived >`
`DiscreteDistributionAbsSquare` (const Eigen::MatrixBase< Derived > &V)
- `std::size_t sample ()`
- `std::vector< double > probabilities () const`

Protected Member Functions

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

Protected Attributes

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

6.2.1 Constructor & Destructor Documentation

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

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

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

6.2.1.4 `template<typename Derived > qpp::DiscreteDistributionAbsSquare::DiscreteDistributionAbsSquare (const Eigen::MatrixBase< Derived > & V) [inline]`

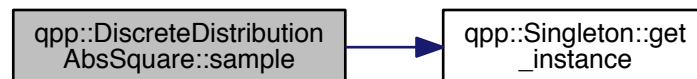
6.2.2 Member Function Documentation

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

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

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

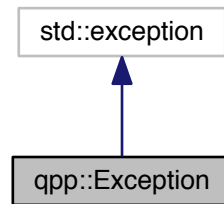
The documentation for this class was generated from the following file:

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

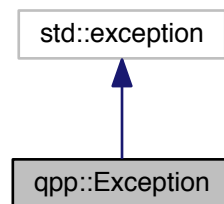
6.3 qpp::Exception Class Reference

```
#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::TYPE_MISMATCH`, `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

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
TYPE_MISMATCH
UNDEFINED_TYPE
CUSTOM_EXCEPTION

6.3.2 Constructor & Destructor Documentation

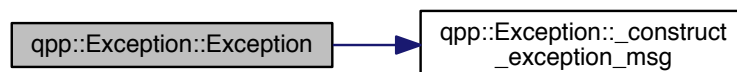
6.3.2.1 `qpp::Exception::Exception (const std::string & where, const Type & type)` `[inline]`

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:



6.3.3 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]`

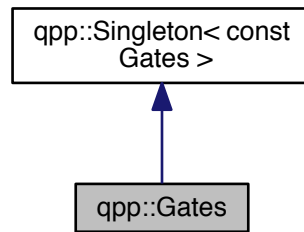
The documentation for this class was generated from the following file:

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

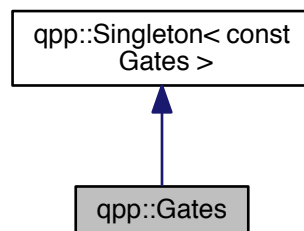
6.4 qpp::Gates Class Reference

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

Inheritance diagram for qpp::Gates:



Collaboration diagram for qpp::Gates:



Public Member Functions

- [types::cmat Rn](#) (double theta, std::vector< double > n) const
- [types::cmat Zd](#) (std::size_t D) const
- [types::cmat Fd](#) (std::size_t D) const
- [types::cmat Xd](#) (std::size_t D) const
- template<typename Derived = Eigen::MatrixXcd>
Derived [ld](#) (std::size_t D) const
- template<typename Derived1 , typename Derived2 >
[types::DynMat](#)< typename
Derived1::Scalar > [applyCTRL](#) (const Eigen::MatrixBase< Derived1 > &state, const Eigen::MatrixBase<
Derived2 > &A, const std::vector< std::size_t > &ctrl, const std::vector< std::size_t > &subsys, std::size_t
n, std::size_t d=2) const
- template<typename Derived1 , typename Derived2 >
[types::DynMat](#)< typename
Derived1::Scalar > [apply](#) (const Eigen::MatrixBase< Derived1 > &state, const Eigen::MatrixBase< De-
rived2 > &A, const std::vector< std::size_t > &subsys, const std::vector< std::size_t > &dims) const
- template<typename Derived >
[types::DynMat](#)< typename
Derived::Scalar > [CTRL](#) (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t > &ctrl,
const std::vector< std::size_t > &subsys, std::size_t n, std::size_t d=2) const

Public Attributes

- [types::cmat Id2](#) { types::cmat::Identity(2, 2) }
- [types::cmat H](#) { types::cmat::Zero(2, 2) }
- [types::cmat X](#) { types::cmat::Zero(2, 2) }
- [types::cmat Y](#) { types::cmat::Zero(2, 2) }
- [types::cmat Z](#) { types::cmat::Zero(2, 2) }
- [types::cmat S](#) { types::cmat::Zero(2, 2) }
- [types::cmat T](#) { types::cmat::Zero(2, 2) }
- [types::cmat CNOTab](#) { types::cmat::Identity(4, 4) }
- [types::cmat CZ](#) { types::cmat::Identity(4, 4) }
- [types::cmat CNOTba](#) { types::cmat::Zero(4, 4) }
- [types::cmat SWAP](#) { types::cmat::Identity(4, 4) }
- [types::cmat TOF](#) { types::cmat::Identity(8, 8) }
- [types::cmat FRED](#) { types::cmat::Identity(8, 8) }

Private Member Functions

- [Gates](#) ()

Friends

- class [Singleton< const Gates >](#)

Additional Inherited Members

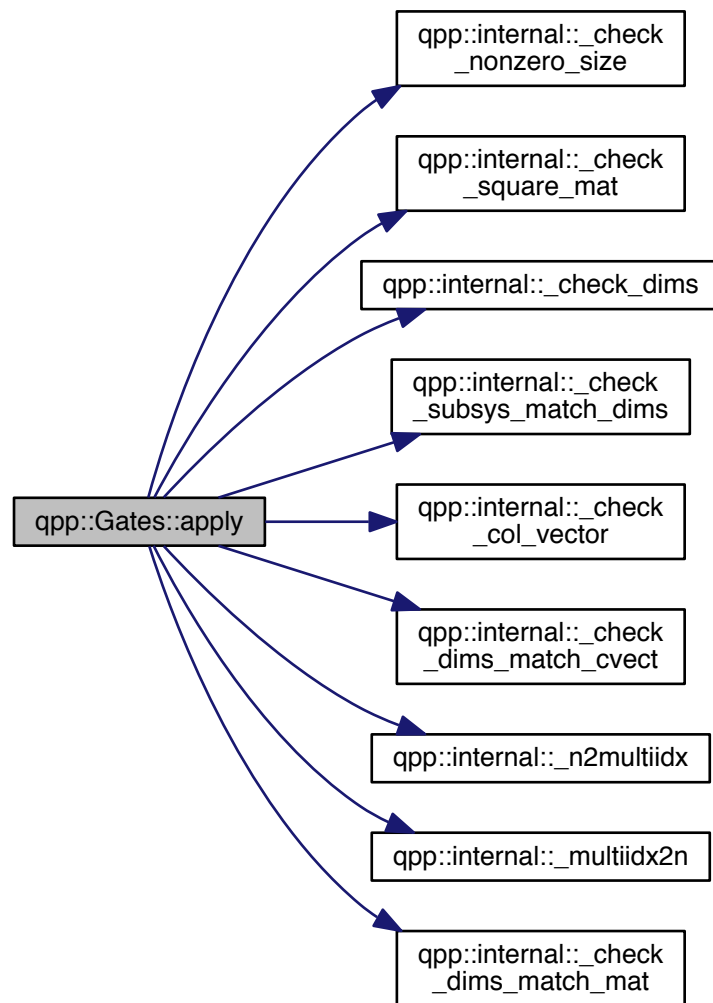
6.4.1 Constructor & Destructor Documentation

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

6.4.2 Member Function Documentation

```
6.4.2.1 template<typename Derived1 , typename Derived2 > types::DynMat<typename Derived1::Scalar>
qpp::Gates::apply ( const Eigen::MatrixBase< Derived1 > & state, const Eigen::MatrixBase< Derived2 > & A, const
std::vector< std::size_t > & subsys, const std::vector< std::size_t > & dims ) const [inline]
```

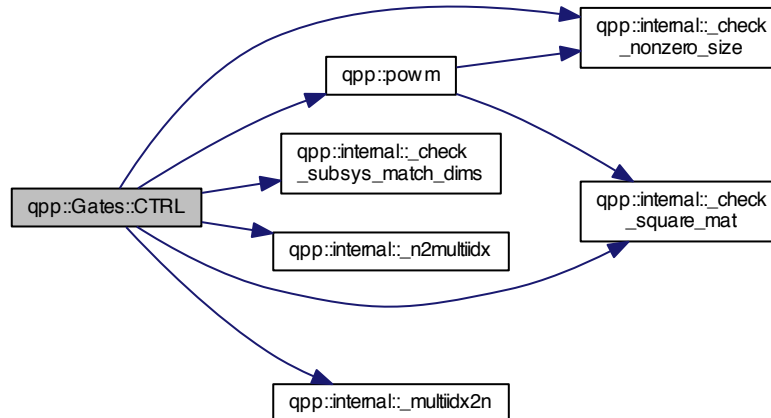
Here is the call graph for this function:



```
6.4.2.2 template<typename Derived1 , typename Derived2 > types::DynMat<typename Derived1::Scalar>
qpp::Gates::applyCTRL ( const Eigen::MatrixBase< Derived1 > & state, const Eigen::MatrixBase< Derived2 > & A,
const std::vector< std::size_t > & ctrl, const std::vector< std::size_t > & subsys, std::size_t n, std::size_t d = 2 )
const [inline]
```

6.4.2.3 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::Gates::CTRL (const Eigen::MatrixBase< Derived > & A, const std::vector< std::size_t > & ctrl, const std::vector< std::size_t > & subsys, std::size_t n, std::size_t d = 2) const [inline]`

Here is the call graph for this function:



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

Here is the call graph for this function:

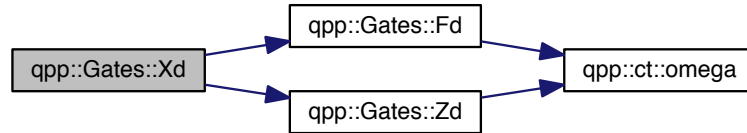


6.4.2.5 `template<typename Derived = Eigen::MatrixXcd> Derived qpp::Gates::Id (std::size_t D) const [inline]`

6.4.2.6 `types::cmat qpp::Gates::Rn (double theta, std::vector< double > n) const [inline]`

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

Here is the call graph for this function:



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

Here is the call graph for this function:



6.4.3 Friends And Related Function Documentation

6.4.3.1 `friend class Singleton< const Gates > [friend]`

6.4.4 Member Data Documentation

6.4.4.1 `types::cmat qpp::Gates::CNOTab { types::cmat::Identity(4, 4) }`

6.4.4.2 `types::cmat qpp::Gates::CNOTba { types::cmat::Zero(4, 4) }`

6.4.4.3 `types::cmat qpp::Gates::CZ { types::cmat::Identity(4, 4) }`

6.4.4.4 `types::cmat qpp::Gates::FRED { types::cmat::Identity(8, 8) }`

6.4.4.5 `types::cmat qpp::Gates::H { types::cmat::Zero(2, 2) }`

6.4.4.6 `types::cmat qpp::Gates::Id2 { types::cmat::Identity(2, 2) }`

6.4.4.7 `types::cmat qpp::Gates::S { types::cmat::Zero(2, 2) }`

6.4.4.8 `types::cmat qpp::Gates::SWAP { types::cmat::Identity(4, 4) }`

6.4.4.9 `types::cmat qpp::Gates::T { types::cmat::Zero(2, 2) }`

6.4.4.10 `types::cmat qpp::Gates::TOF { types::cmat::Identity(8, 8) }`

6.4.4.11 `types::cmat qpp::Gates::X { types::cmat::Zero(2, 2) }`

6.4.4.12 `types::cmat qpp::Gates::Y { types::cmat::Zero(2, 2) }`

6.4.4.13 `types::cmat qpp::Gates::Z { types::cmat::Zero(2, 2) }`

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::get_instance\(\)](#).pz0)
- [std::size_t measure](#) (const [types::cmat](#) &U, bool destructive=false)
- [std::size_t measure](#) (bool destructive=false)
- [types::cmat getRho](#) () const
- [std::size_t getD](#) () const

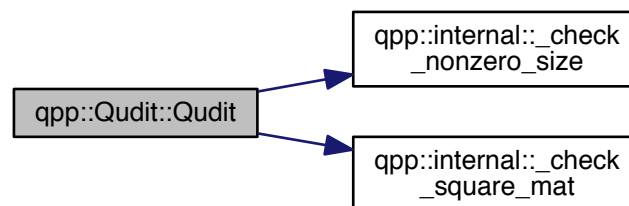
Private Attributes

- [types::cmat_rho](#)
- [std::size_t _D](#)

6.6.1 Constructor & Destructor Documentation

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

Here is the call graph for this function:



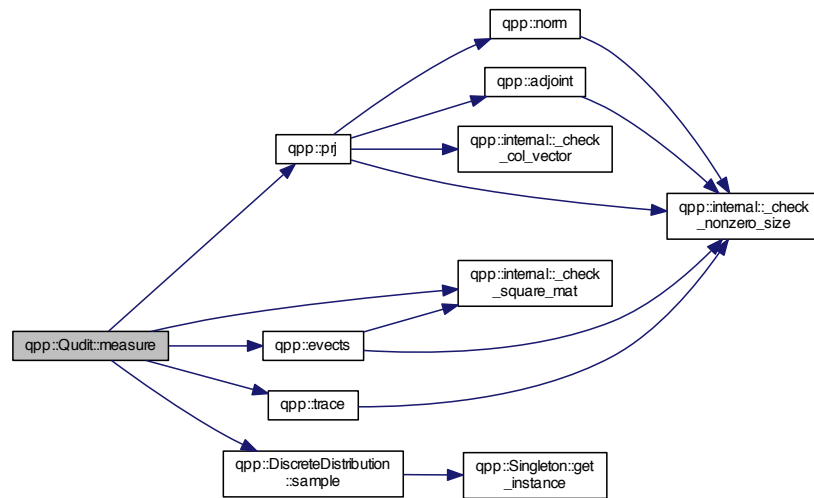
6.6.2 Member Function Documentation

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

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

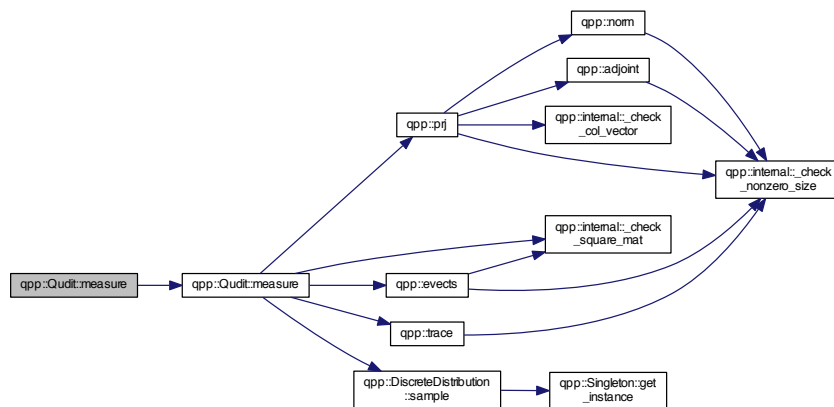
6.6.2.3 `std::size_t qpp::Qudit::measure (const types::cmat & U, bool destructive = false) [inline]`

Here is the call graph for this function:



6.6.2.4 `std::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 `std::size_t qpp::Qudit::_D [private]`

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

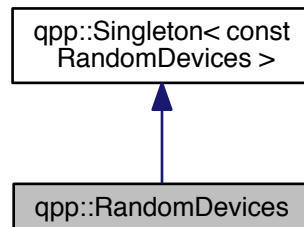
The documentation for this class was generated from the following file:

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

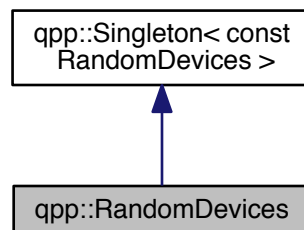
6.7 qpp::RandomDevices Class Reference

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

Inheritance diagram for qpp::RandomDevices:



Collaboration diagram for qpp::RandomDevices:



Public Attributes

- `std::random_device _rd`
- `std::mt19937 _rng`

Private Member Functions

- `RandomDevices ()`

Friends

- class `Singleton< const RandomDevices >`

Additional Inherited Members

6.7.1 Constructor & Destructor Documentation

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

6.7.2 Friends And Related Function Documentation

6.7.2.1 `friend class Singleton< const RandomDevices > [friend]`

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 [mutable]`

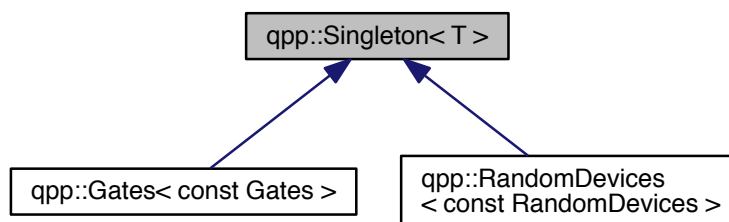
The documentation for this class was generated from the following file:

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

6.8 qpp::Singleton< T > Class Template Reference

```
#include <singleton.h>
```

Inheritance diagram for `qpp::Singleton< T >`:



Static Public Member Functions

- static `T & get_instance ()`

Protected Member Functions

- `Singleton ()=default`
- `virtual ~Singleton ()`
- `Singleton (const Singleton &)=delete`
- `Singleton & operator= (const Singleton &)=delete`

6.8.1 Constructor & Destructor Documentation

6.8.1.1 `template<typename T> qpp::Singleton< T >::Singleton () [protected],[default]`

6.8.1.2 `template<typename T> virtual qpp::Singleton< T >::~~Singleton ()` `[inline]`, `[protected]`, `[virtual]`

6.8.1.3 `template<typename T> qpp::Singleton< T >::~Singleton (const Singleton< T > &)` `[protected]`, `[delete]`

6.8.2 Member Function Documentation

6.8.2.1 `template<typename T> static T& qpp::Singleton< T >::get_instance ()` `[inline]`, `[static]`

6.8.2.2 `template<typename T> Singleton& qpp::Singleton< T >::operator= (const Singleton< T > &)` `[protected]`, `[delete]`

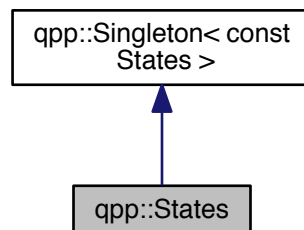
The documentation for this class was generated from the following file:

- `include/classes/singleton.h`

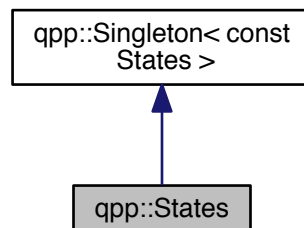
6.9 qpp::States Class Reference

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

Inheritance diagram for qpp::States:



Collaboration diagram for qpp::States:



Public Attributes

- [types::ket x0](#) { [types::ket::Zero\(2\)](#) }
- [types::ket x1](#) { [types::ket::Zero\(2\)](#) }
- [types::ket y0](#) { [types::ket::Zero\(2\)](#) }
- [types::ket y1](#) { [types::ket::Zero\(2\)](#) }
- [types::ket z0](#) { [types::ket::Zero\(2\)](#) }
- [types::ket z1](#) { [types::ket::Zero\(2\)](#) }
- [types::cmat px0](#) { [types::cmat::Zero\(2, 2\)](#) }
- [types::cmat px1](#) { [types::cmat::Zero\(2, 2\)](#) }
- [types::cmat py0](#) { [types::cmat::Zero\(2, 2\)](#) }
- [types::cmat py1](#) { [types::cmat::Zero\(2, 2\)](#) }
- [types::cmat pz0](#) { [types::cmat::Zero\(2, 2\)](#) }
- [types::cmat pz1](#) { [types::cmat::Zero\(2, 2\)](#) }
- [types::ket b00](#) { [types::ket::Zero\(4\)](#) }
- [types::ket b01](#) { [types::ket::Zero\(4\)](#) }
- [types::ket b10](#) { [types::ket::Zero\(4\)](#) }
- [types::ket b11](#) { [types::ket::Zero\(4\)](#) }
- [types::cmat pb00](#) { [types::cmat::Zero\(4, 4\)](#) }
- [types::cmat pb01](#) { [types::cmat::Zero\(4, 4\)](#) }
- [types::cmat pb10](#) { [types::cmat::Zero\(4, 4\)](#) }
- [types::cmat pb11](#) { [types::cmat::Zero\(4, 4\)](#) }
- [types::ket GHZ](#) { [types::ket::Zero\(8\)](#) }
- [types::ket W](#) { [types::ket::Zero\(8\)](#) }
- [types::cmat pGHZ](#) { [types::cmat::Zero\(8, 8\)](#) }
- [types::cmat pW](#) { [types::cmat::Zero\(8, 8\)](#) }

Private Member Functions

- [States](#) ()

Friends

- class [Singleton< const States >](#)

Additional Inherited Members

6.9.1 Constructor & Destructor Documentation

6.9.1.1 [qpp::States::States \(\)](#) [[inline](#)], [[private](#)]

6.9.2 Friends And Related Function Documentation

6.9.2.1 [friend class Singleton< const States >](#) [[friend](#)]

6.9.3 Member Data Documentation

6.9.3.1 [types::ket qpp::States::b00](#) { [types::ket::Zero\(4\)](#) }

6.9.3.2 [types::ket qpp::States::b01](#) { [types::ket::Zero\(4\)](#) }

6.9.3.3 [types::ket qpp::States::b10](#) { [types::ket::Zero\(4\)](#) }


```

6.9.3.4  types::ket qpp::States::b11 { types::ket::Zero(4) }
6.9.3.5  types::ket qpp::States::GHZ { types::ket::Zero(8) }
6.9.3.6  types::cmat qpp::States::pb00 { types::cmat::Zero(4, 4) }
6.9.3.7  types::cmat qpp::States::pb01 { types::cmat::Zero(4, 4) }
6.9.3.8  types::cmat qpp::States::pb10 { types::cmat::Zero(4, 4) }
6.9.3.9  types::cmat qpp::States::pb11 { types::cmat::Zero(4, 4) }
6.9.3.10 types::cmat qpp::States::pGHZ { types::cmat::Zero(8, 8) }
6.9.3.11 types::cmat qpp::States::pW { types::cmat::Zero(8, 8) }
6.9.3.12 types::cmat qpp::States::px0 { types::cmat::Zero(2, 2) }
6.9.3.13 types::cmat qpp::States::px1 { types::cmat::Zero(2, 2) }
6.9.3.14 types::cmat qpp::States::py0 { types::cmat::Zero(2, 2) }
6.9.3.15 types::cmat qpp::States::py1 { types::cmat::Zero(2, 2) }
6.9.3.16 types::cmat qpp::States::pz0 { types::cmat::Zero(2, 2) }
6.9.3.17 types::cmat qpp::States::pz1 { types::cmat::Zero(2, 2) }
6.9.3.18 types::ket qpp::States::W { types::ket::Zero(8) }
6.9.3.19 types::ket qpp::States::x0 { types::ket::Zero(2) }
6.9.3.20 types::ket qpp::States::x1 { types::ket::Zero(2) }
6.9.3.21 types::ket qpp::States::y0 { types::ket::Zero(2) }
6.9.3.22 types::ket qpp::States::y1 { types::ket::Zero(2) }
6.9.3.23 types::ket qpp::States::z0 { types::ket::Zero(2) }
6.9.3.24 types::ket qpp::States::z1 { types::ket::Zero(2) }

```

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

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

6.10 qpp::Timer Class Reference

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

Public Member Functions

- [Timer](#) ()
- void [tic](#) ()

- void [toc](#) ()
- double [seconds](#) () const

Protected Attributes

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

Friends

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

6.10.1 Constructor & Destructor Documentation

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

6.10.2 Member Function Documentation

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

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

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

6.10.3 Friends And Related Function Documentation

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

6.10.4 Member Data Documentation

6.10.4.1 `std::chrono::steady_clock::time_point qpp::Timer::_end` [[protected](#)]

6.10.4.2 `std::chrono::steady_clock::time_point qpp::Timer::_start` [[protected](#)]

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

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

6.11 qpp::UniformIntDistribution Class Reference

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

Public Member Functions

- [UniformIntDistribution](#) (int a=0, int b=1)
- int [sample](#) ()

Protected Attributes

- std::uniform_int_distribution [_d](#)

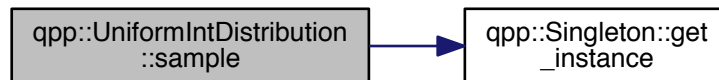
6.11.1 Constructor & Destructor Documentation

6.11.1.1 `qpp::UniformIntDistribution::UniformIntDistribution (int a = 0, int b = 1)` `[inline]`

6.11.2 Member Function Documentation

6.11.2.1 `int qpp::UniformIntDistribution::sample ()` `[inline]`

Here is the call graph for this function:



6.11.3 Member Data Documentation

6.11.3.1 `std::uniform_int_distribution qpp::UniformIntDistribution::_d` `[protected]`

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

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

6.12 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`

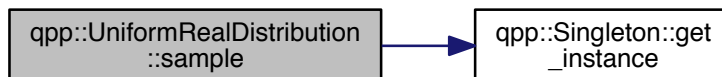
6.12.1 Constructor & Destructor Documentation

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

6.12.2 Member Function Documentation

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

Here is the call graph for this function:



6.12.3 Member Data Documentation

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

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

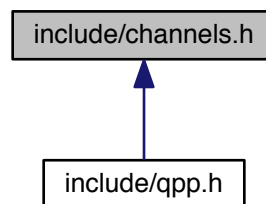
- `include/classes/stat.h`

Chapter 7

File Documentation

7.1 include/channels.h File Reference

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



Namespaces

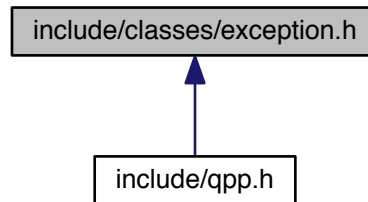
- [qpp](#)

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< std::size_t > &subsys, const std::vector< std::size_t > &dims)`

7.2 include/classes/exception.h File Reference

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



Classes

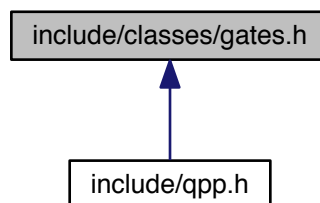
- class [qpp::Exception](#)

Namespaces

- [qpp](#)

7.3 include/classes/gates.h File Reference

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



Classes

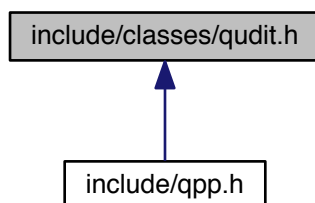
- class [qpp::Gates](#)

Namespaces

- [qpp](#)

7.4 include/classes/qudit.h File Reference

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



Classes

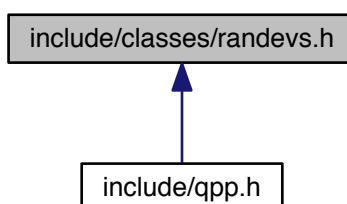
- class [qpp::Qudit](#)

Namespaces

- [qpp](#)

7.5 include/classes/randevs.h File Reference

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



Classes

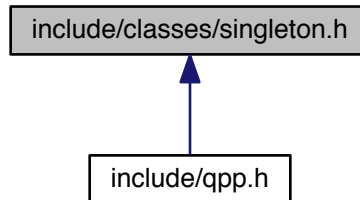
- class [qpp::RandomDevices](#)

Namespaces

- [qpp](#)

7.6 include/classes/singleton.h File Reference

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



Classes

- class [qpp::Singleton< T >](#)

Namespaces

- [qpp](#)

Macros

- [#define CLASS_SINGLETON\(Foo\)](#)
- [#define CLASS_CONST_SINGLETON\(Foo\)](#)

7.6.1 Macro Definition Documentation

7.6.1.1 [#define CLASS_CONST_SINGLETON\(Foo \)](#)

Value:

```

class Foo: public Singleton<const Foo>\
{
    friend class Singleton<const Foo>;
}
  
```

7.6.1.2 [#define CLASS_SINGLETON\(Foo \)](#)

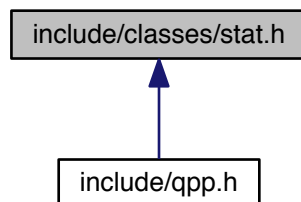
Value:

```

class Foo: public Singleton<Foo>\
{
    friend class Singleton<Foo>;
}
  
```


7.7 include/classes/stat.h File Reference

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



Classes

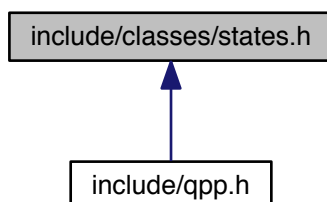
- class [qpp::NormalDistribution](#)
- class [qpp::UniformRealDistribution](#)
- class [qpp::UniformIntDistribution](#)
- class [qpp::DiscreteDistribution](#)
- class [qpp::DiscreteDistributionAbsSquare](#)

Namespaces

- [qpp](#)

7.8 include/classes/states.h File Reference

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



Classes

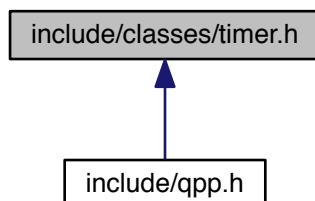
- class [qpp::States](#)

Namespaces

- [qpp](#)

7.9 include/classes/timer.h File Reference

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



Classes

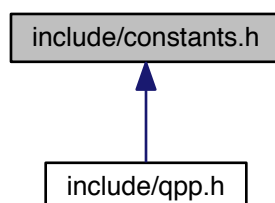
- class [qpp::Timer](#)

Namespaces

- [qpp](#)

7.10 include/constants.h File Reference

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



Namespaces

- [qpp](#)

- [qpp::ct](#)

Functions

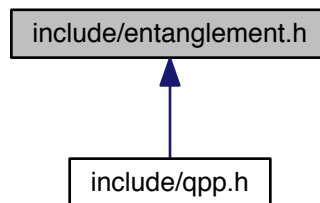
- constexpr std::complex< double > [qpp::operator""_i](#) (unsigned long long int x)
- constexpr std::complex< double > [qpp::operator""_i](#) (long double x)
- std::complex< double > [qpp::ct::omega](#) (std::size_t D)

Variables

- constexpr double [qpp::ct::chop](#) = 1e-10
- constexpr double [qpp::ct::eps](#) = 1e-12
- constexpr std::size_t [qpp::ct::maxn](#) = 64
- constexpr double [qpp::ct::pi](#) = 3.141592653589793238462643383279502884
- constexpr double [qpp::ct::ee](#) = 2.718281828459045235360287471352662497

7.11 include/entanglement.h File Reference

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



Namespaces

- [qpp](#)

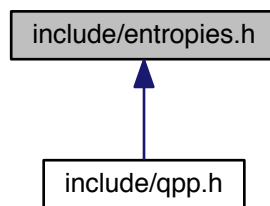
Functions

- template<typename Derived >
types::cmat [qpp::schmidtcoeff](#) (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t > &dims)
- template<typename Derived >
types::cmat [qpp::schmidtU](#) (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t > &dims)
- template<typename Derived >
types::cmat [qpp::schmidtV](#) (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t > &dims)
- template<typename Derived >
types::cmat [qpp::schmidtprob](#) (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t > &dims)
- template<typename Derived >
double [qpp::entanglement](#) (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t > &dims)

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

7.12 include/entropies.h File Reference

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



Namespaces

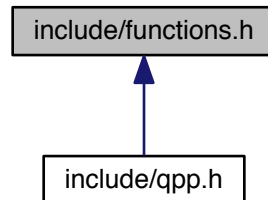
- [qpp](#)

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< std::size_t > &subsys, const std::vector< std::size_t > &dims)`

7.13 include/functions.h File Reference

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)`
Transpose.
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::conjugate (const Eigen::MatrixBase< Derived > &A)`
Complex conjugate.
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::adjoint (const Eigen::MatrixBase< Derived > &A)`
Adjoint.
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::inverse (const Eigen::MatrixBase< Derived > &A)`
Inverse.
- `template<typename Derived >`
`Derived::Scalar qpp::trace (const Eigen::MatrixBase< Derived > &A)`
Trace.
- `template<typename Derived >`
`Derived::Scalar qpp::det (const Eigen::MatrixBase< Derived > &A)`
Determinant.
- `template<typename Derived >`
`Derived::Scalar qpp::logdet (const Eigen::MatrixBase< Derived > &A)`
Logarithm of the determinant.
- `template<typename Derived >`
`Derived::Scalar qpp::sum (const Eigen::MatrixBase< Derived > &A)`
Element-wise sum.

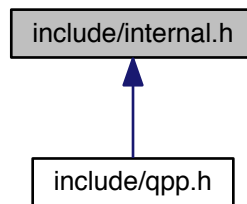
- template<typename Derived >
double [qpp::norm](#) (const Eigen::MatrixBase< Derived > &A)
Trace norm.
- template<typename Derived >
types::cmat [qpp::evals](#) (const Eigen::MatrixBase< Derived > &A)
Eigenvalues.
- template<typename Derived >
types::cmat [qpp::evecs](#) (const Eigen::MatrixBase< Derived > &A)
Eigenvectors.
- template<typename Derived >
types::dmat [qpp::hevals](#) (const Eigen::MatrixBase< Derived > &A)
Hermitian eigenvalues.
- template<typename Derived >
types::cmat [qpp::hevecs](#) (const Eigen::MatrixBase< Derived > &A)
Hermitian eigenvectors.
- template<typename Derived >
types::cmat [qpp::funm](#) (const Eigen::MatrixBase< Derived > &A, types::cplx(*f)(const types::cplx &))
Functional calculus f(A)
- template<typename Derived >
types::cmat [qpp::sqrtm](#) (const Eigen::MatrixBase< Derived > &A)
Matrix square root.
- template<typename Derived >
types::cmat [qpp::absm](#) (const Eigen::MatrixBase< Derived > &A)
Matrix absolut value.
- template<typename Derived >
types::cmat [qpp::expm](#) (const Eigen::MatrixBase< Derived > &A)
Matrix exponential.
- template<typename Derived >
types::cmat [qpp::logm](#) (const Eigen::MatrixBase< Derived > &A)
Matrix logarithm.
- template<typename Derived >
types::cmat [qpp::sinm](#) (const Eigen::MatrixBase< Derived > &A)
Matrix sin.
- template<typename Derived >
types::cmat [qpp::cosm](#) (const Eigen::MatrixBase< Derived > &A)
Matrix cos.
- template<typename Derived >
types::cmat [qpp::spectralpwm](#) (const Eigen::MatrixBase< Derived > &A, const types::cplx z)
Matrix power.
- template<typename Derived >
types::DynMat< typename
Derived::Scalar > [qpp::pwm](#) (const Eigen::MatrixBase< Derived > &A, std::size_t n)
Matrix power.
- template<typename OutputScalar , typename Derived >
types::DynMat< OutputScalar > [qpp::cwise](#) (const Eigen::MatrixBase< Derived > &A, Output←
Scalar(*f)(const typename Derived::Scalar &))
Functor.
- 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, std::size_t n)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::reshape (const Eigen::MatrixBase< Derived > &A, std::size_t rows, std::size_t cols)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::syspermute (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t`
`> &perm, const std::vector< std::size_t > &dims)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::ptrace1 (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t >`
`&dims)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::ptrace2 (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t >`
`&dims)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::ptrace (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t >`
`&subsys, const std::vector< std::size_t > &dims)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::ptranspose (const Eigen::MatrixBase< Derived > &A, const std::vector< std::size_t`
`> &subsys, const std::vector< std::size_t > &dims)`
- `template<typename Derived1 , typename Derived2 >`
`types::DynMat< typename`
`Derived1::Scalar > qpp::comm (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase< De-`
`derived2 > &B)`
- `template<typename Derived1 , typename Derived2 >`
`types::DynMat< typename`
`Derived1::Scalar > qpp::anticomm (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase<`
`Derived2 > &B)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::prj (const Eigen::MatrixBase< Derived > &V)`
- `template<typename Derived >`
`types::DynMat< typename`
`Derived::Scalar > qpp::expandout (const Eigen::MatrixBase< Derived > &A, std::size_t pos, const std::vector<`
`std::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< std::size_t > qpp::n2multiidx (std::size_t n, const std::vector< std::size_t > &dims)`
- `std::size_t qpp::multiidx2n (const std::vector< std::size_t > &midx, const std::vector< std::size_t > &dims)`
- `types::ket qpp::mket (const std::vector< std::size_t > &mask)`
- `types::ket qpp::mket (const std::vector< std::size_t > &mask, const std::vector< std::size_t > &dims)`

- `types::ket qpp::mket` (`const std::vector< std::size_t > &mask, std::size_t d`)
- `std::vector< std::size_t > qpp::invperm` (`const std::vector< std::size_t > &perm`)
- `std::vector< std::size_t > qpp::compperm` (`const std::vector< std::size_t > &perm, const std::vector< std::size_t > &sigma`)

7.14 include/internal.h File Reference

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



Namespaces

- `qpp`
- `qpp::internal`

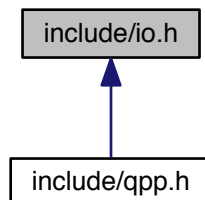
Functions

- `void qpp::internal::_n2multiidx` (`std::size_t n, std::size_t numdims, const std::size_t *dims, std::size_t *result`)
- `std::size_t qpp::internal::_multiidx2n` (`const std::size_t *midx, std::size_t numdims, const std::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< std::size_t > &dims`)
- `template<typename Derived >`
`bool qpp::internal::_check_dims_match_mat` (`const std::vector< std::size_t > &dims, const Eigen::MatrixBase< Derived > &A`)
- `template<typename Derived >`
`bool qpp::internal::_check_dims_match_cvect` (`const std::vector< std::size_t > &dims, const Eigen::MatrixBase< Derived > &V`)
- `template<typename Derived >`
`bool qpp::internal::_check_dims_match_rvect` (`const std::vector< std::size_t > &dims, const Eigen::MatrixBase< Derived > &V`)
- `bool qpp::internal::_check_eq_dims` (`const std::vector< std::size_t > &dims, std::size_t dim`)

- bool [qpp::internal::_check_subsys_match_dims](#) (const std::vector< std::size_t > &subsys, const std::vector< std::size_t > &dims)
- bool [qpp::internal::_check_perm](#) (const std::vector< std::size_t > &perm)
- template<typename Derived1 , typename Derived2 >
types::DynMat< typename
Derived1::Scalar > [qpp::internal::_kron2](#) (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase< 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)

7.15 include/io.h File Reference

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 std::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 std::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)

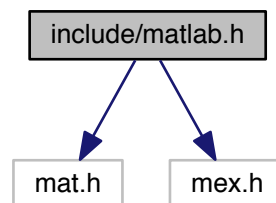
- void [qpp::disp](#) (const types::cplx c, double chop=ct::chop, std::ostream &os=std::cout)
- void [qpp::displn](#) (const types::cplx c, double chop=ct::chop, std::ostream &os=std::cout)
- template<typename Derived >
void [qpp::save](#) (const Eigen::MatrixBase< Derived > &A, const std::string &fname)
- template<typename Derived >
types::DynMat< typename
Derived::Scalar > [qpp::load](#) (const std::string &fname)

7.16 include/matlab.h File Reference

```
#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< 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)

7.17 include/qpp.h File Reference

```
#include <algorithm>
#include <chrono>
#include <cmath>
#include <complex>
#include <cstdlib>
#include <cstring>
#include <exception>
#include <fstream>
#include <functional>
#include <iomanip>
#include <iostream>
#include <iterator>
#include <numeric>
#include <ostream>
#include <random>
#include <stdexcept>
#include <string>
#include <type_traits>
#include <utility>
#include <vector>
#include <Eigen/Dense>
#include <Eigen/SVD>
#include "constants.h"
#include "types.h"
#include "classes/exception.h"
#include "classes/singleton.h"
#include "classes/states.h"
#include "classes/randevs.h"
#include "internal.h"
#include "functions.h"
#include "classes/gates.h"
#include "classes/stat.h"
#include "entropies.h"
#include "entanglement.h"
#include "channels.h"
#include "io.h"
#include "random.h"
#include "classes/qudit.h"
#include "classes/timer.h"
```

Include dependency graph for qpp.h:



Namespaces

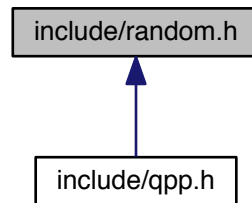
- [qpp](#)

Variables

- const RandomDevices & [qpp::rdevs](#) = RandomDevices::get_instance()
- const Gates & [qpp::gt](#) = Gates::get_instance()
- const States & [qpp::st](#) = States::get_instance()

7.18 include/random.h File Reference

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



Namespaces

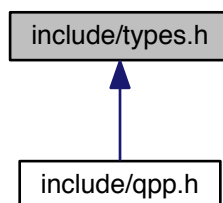
- [qpp](#)

Functions

- `template<typename Derived >`
Derived [qpp::rand](#) (std::size_t rows, std::size_t cols, double a=0, double b=1)
- `template<>`
types::dmat [qpp::rand](#) (std::size_t rows, std::size_t cols, double a, double b)
- `template<>`
types::cmat [qpp::rand](#) (std::size_t rows, std::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](#) (std::size_t rows, std::size_t cols, double mean=0, double sigma=1)
- `template<>`
types::dmat [qpp::randn](#) (std::size_t rows, std::size_t cols, double mean, double sigma)
- `template<>`
types::cmat [qpp::randn](#) (std::size_t rows, std::size_t cols, double mean, double sigma)
- double [qpp::randn](#) (double mean=0, double sigma=1)
- types::cmat [qpp::randU](#) (std::size_t D)
- types::cmat [qpp::randV](#) (std::size_t Din, std::size_t Dout)
- std::vector< types::cmat > [qpp::randkraus](#) (std::size_t n, std::size_t D)
- types::cmat [qpp::randH](#) (std::size_t D)
- types::ket [qpp::randket](#) (std::size_t D)
- types::cmat [qpp::randrho](#) (std::size_t D)
- std::vector< std::size_t > [qpp::randperm](#) (std::size_t n)

7.19 include/types.h File Reference

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 >`

Index

- absm
 - qpp, [14](#)
- adjoint
 - qpp, [14](#)
- anticomm
 - qpp, [15](#)
- CUSTOM_EXCEPTION
 - qpp::Exception, [63](#)
- channel
 - qpp, [15](#)
- choi
 - qpp, [16](#)
- choi2kraus
 - qpp, [16](#)
- comm
 - qpp, [17](#)
- compperm
 - qpp, [17](#)
- conjugate
 - qpp, [18](#)
- cosm
 - qpp, [18](#)
- cwise
 - qpp, [19](#)
- DIMS_INVALID
 - qpp::Exception, [63](#)
- DIMS_MISMATCH_CVECTOR
 - qpp::Exception, [63](#)
- DIMS_MISMATCH_MATRIX
 - qpp::Exception, [63](#)
- DIMS_MISMATCH_RVECTOR
 - qpp::Exception, [63](#)
- DIMS_MISMATCH_VECTOR
 - qpp::Exception, [63](#)
- DIMS_NOT_EQUAL
 - qpp::Exception, [63](#)
- det
 - qpp, [19](#)
- disp
 - qpp, [20](#)
- displn
 - qpp, [20](#), [21](#)
- entanglement
 - qpp, [22](#)
- evals
 - qpp, [22](#)
- evects
 - qpp, [23](#)
- expandout
 - qpp, [23](#)
- expm
 - qpp, [24](#)
- funm
 - qpp, [24](#)
- gconcurrency
 - qpp, [26](#)
- grams
 - qpp, [26](#), [27](#)
- gt
 - qpp, [55](#)
- hevals
 - qpp, [27](#)
- hevects
 - qpp, [28](#)
- inverse
 - qpp, [28](#)
- invperm
 - qpp, [30](#)
- kron
 - qpp, [30](#), [31](#)
- kronpow
 - qpp, [31](#)
- load
 - qpp, [31](#)
- logdet
 - qpp, [32](#)
- logm
 - qpp, [32](#)
- MATRIX_NOT_CVECTOR
 - qpp::Exception, [63](#)
- MATRIX_NOT_RVECTOR
 - qpp::Exception, [63](#)
- MATRIX_NOT_SQUARE
 - qpp::Exception, [63](#)
- MATRIX_NOT_SQUARE_OR_CVECTOR
 - qpp::Exception, [63](#)
- MATRIX_NOT_SQUARE_OR_RVECTOR
 - qpp::Exception, [63](#)
- MATRIX_NOT_SQUARE_OR_VECTOR
 - qpp::Exception, [63](#)
- MATRIX_NOT_VECTOR
 - qpp::Exception, [63](#)

- qpp::Exception, 63
- mket
 - qpp, 33
- multiidx2n
 - qpp, 34
- n2multiidx
 - qpp, 34
- NOT_BIPARTITE
 - qpp::Exception, 63
- NOT_QUBIT_GATE
 - qpp::Exception, 63
- NOT_QUBIT_SUBSYS
 - qpp::Exception, 63
- norm
 - qpp, 34
- OUT_OF_RANGE
 - qpp::Exception, 63
- PERM_INVALID
 - qpp::Exception, 63
- powm
 - qpp, 35
- prj
 - qpp, 35
- ptrace
 - qpp, 36
- ptrace1
 - qpp, 37
- ptrace2
 - qpp, 38
- ptranspose
 - qpp, 38
- qmutualinfo
 - qpp, 39
- qpp, 9
 - absm, 14
 - adjoint, 14
 - anticomm, 15
 - channel, 15
 - choi, 16
 - choi2kraus, 16
 - comm, 17
 - compperm, 17
 - conjugate, 18
 - cosm, 18
 - cwise, 19
 - det, 19
 - disp, 20
 - displn, 20, 21
 - entanglement, 22
 - evals, 22
 - evects, 23
 - expandout, 23
 - expm, 24
 - funm, 24
 - gconcurrency, 26
 - grams, 26, 27
 - gt, 55
 - hevals, 27
 - hevects, 28
 - inverse, 28
 - invperm, 30
 - kron, 30, 31
 - kronpow, 31
 - load, 31
 - logdet, 32
 - logm, 32
 - mket, 33
 - multiidx2n, 34
 - n2multiidx, 34
 - norm, 34
 - powm, 35
 - prj, 35
 - ptrace, 36
 - ptrace1, 37
 - ptrace2, 38
 - ptranspose, 38
 - qmutualinfo, 39
 - rand, 40, 41
 - randint, 41
 - randket, 41
 - randkraus, 42
 - randn, 42, 43
 - randperm, 43
 - randrho, 43
 - rdevs, 55
 - renyi, 44
 - reshape, 45
 - save, 45
 - schmidtcoeff, 46
 - schmidtprob, 47
 - shannon, 48
 - sinm, 48
 - spectralpowm, 49
 - sqrtn, 49
 - st, 55
 - sum, 51
 - super, 51
 - syspermute, 52
 - trace, 53
 - transpose, 54
 - tsallis, 54
- qpp::Exception
 - CUSTOM_EXCEPTION, 63
 - DIMS_INVALID, 63
 - DIMS_MISMATCH_CVECTOR, 63
 - DIMS_MISMATCH_MATRIX, 63
 - DIMS_MISMATCH_RVECTOR, 63
 - DIMS_MISMATCH_VECTOR, 63
 - DIMS_NOT_EQUAL, 63
 - MATRIX_NOT_CVECTOR, 63
 - MATRIX_NOT_RVECTOR, 63
 - MATRIX_NOT_SQUARE, 63
 - MATRIX_NOT_SQUARE_OR_CVECTOR, 63

MATRIX_NOT_SQUARE_OR_RVECTOR, 63
 MATRIX_NOT_SQUARE_OR_VECTOR, 63
 MATRIX_NOT_VECTOR, 63
 NOT_BIPARTITE, 63
 NOT_QUBIT_GATE, 63
 NOT_QUBIT_SUBSYS, 63
 OUT_OF_RANGE, 63
 PERM_INVALID, 63
 SUBSYS_MISMATCH_DIMS, 63
 TYPE_MISMATCH, 63
 UNDEFINED_TYPE, 63
 UNKNOWN_EXCEPTION, 63
 ZERO_SIZE, 63

rand
 qpp, 40, 41
 randint
 qpp, 41
 randket
 qpp, 41
 randkraus
 qpp, 42
 randn
 qpp, 42, 43
 randperm
 qpp, 43
 randrho
 qpp, 43
 rdevs
 qpp, 55
 renyi
 qpp, 44
 reshape
 qpp, 45

SUBSYS_MISMATCH_DIMS
 qpp::Exception, 63

save
 qpp, 45
 schmidtcoeff
 qpp, 46
 schmidtprob
 qpp, 47
 shannon
 qpp, 48
 sinm
 qpp, 48
 spectralpowm
 qpp, 49
 sqrtm
 qpp, 49
 st
 qpp, 55
 sum
 qpp, 51
 super
 qpp, 51
 syspermute
 qpp, 52

TYPE_MISMATCH
 qpp::Exception, 63
 trace
 qpp, 53
 transpose
 qpp, 54
 tsallis
 qpp, 54

UNDEFINED_TYPE
 qpp::Exception, 63
 UNKNOWN_EXCEPTION
 qpp::Exception, 63

ZERO_SIZE
 qpp::Exception, 63