

qpp  
0.1

Generated by Doxygen 1.8.7

Sat Oct 18 2014 19:00:22



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	1
<b>2</b>	<b>Hierarchical Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>Namespace Documentation</b>	<b>9</b>
5.1	qpp Namespace Reference . . . . .	9
5.1.1	Function Documentation . . . . .	13
5.1.1.1	absm . . . . .	14
5.1.1.2	adjoint . . . . .	14
5.1.1.3	anticomm . . . . .	15
5.1.1.4	channel . . . . .	15
5.1.1.5	channel . . . . .	16
5.1.1.6	choi . . . . .	16
5.1.1.7	choi2kraus . . . . .	17
5.1.1.8	comm . . . . .	17
5.1.1.9	compperm . . . . .	18
5.1.1.10	conjugate . . . . .	18
5.1.1.11	cosm . . . . .	18
5.1.1.12	cwise . . . . .	19
5.1.1.13	det . . . . .	19
5.1.1.14	disp . . . . .	19
5.1.1.15	disp . . . . .	19
5.1.1.16	disp . . . . .	19
5.1.1.17	disp . . . . .	19
5.1.1.18	displn . . . . .	20

5.1.1.19	displn	20
5.1.1.20	displn	20
5.1.1.21	displn	21
5.1.1.22	entanglement	21
5.1.1.23	evals	22
5.1.1.24	evects	22
5.1.1.25	expandout	23
5.1.1.26	expm	23
5.1.1.27	funm	23
5.1.1.28	gconcurrency	24
5.1.1.29	grams	25
5.1.1.30	grams	25
5.1.1.31	grams	25
5.1.1.32	hevals	26
5.1.1.33	hevects	26
5.1.1.34	inverse	26
5.1.1.35	invperm	27
5.1.1.36	kron	27
5.1.1.37	kron	27
5.1.1.38	kron	27
5.1.1.39	kron	28
5.1.1.40	kronpow	28
5.1.1.41	load	28
5.1.1.42	loadMATLABmatrix	28
5.1.1.43	loadMATLABmatrix	28
5.1.1.44	loadMATLABmatrix	28
5.1.1.45	logdet	29
5.1.1.46	logm	29
5.1.1.47	mket	29
5.1.1.48	mket	30
5.1.1.49	mket	30
5.1.1.50	multiidx2n	30
5.1.1.51	n2multiidx	30
5.1.1.52	norm	31
5.1.1.53	operator""_i	31
5.1.1.54	operator""_i	31
5.1.1.55	powm	31
5.1.1.56	prj	32
5.1.1.57	ptrace	33
5.1.1.58	ptrace1	34

5.1.1.59	<a href="#">ptrace2</a>	34
5.1.1.60	<a href="#">ptranspose</a>	35
5.1.1.61	<a href="#">qmutualinfo</a>	36
5.1.1.62	<a href="#">rand</a>	36
5.1.1.63	<a href="#">rand</a>	36
5.1.1.64	<a href="#">rand</a>	37
5.1.1.65	<a href="#">rand</a>	37
5.1.1.66	<a href="#">randH</a>	37
5.1.1.67	<a href="#">randint</a>	37
5.1.1.68	<a href="#">randket</a>	38
5.1.1.69	<a href="#">randkraus</a>	38
5.1.1.70	<a href="#">randn</a>	38
5.1.1.71	<a href="#">randn</a>	38
5.1.1.72	<a href="#">randn</a>	39
5.1.1.73	<a href="#">randn</a>	39
5.1.1.74	<a href="#">randperm</a>	39
5.1.1.75	<a href="#">randrho</a>	40
5.1.1.76	<a href="#">randU</a>	40
5.1.1.77	<a href="#">randV</a>	40
5.1.1.78	<a href="#">renyi</a>	40
5.1.1.79	<a href="#">renyi_inf</a>	41
5.1.1.80	<a href="#">reshape</a>	41
5.1.1.81	<a href="#">save</a>	41
5.1.1.82	<a href="#">saveMATLABmatrix</a>	41
5.1.1.83	<a href="#">saveMATLABmatrix</a>	41
5.1.1.84	<a href="#">saveMATLABmatrix</a>	42
5.1.1.85	<a href="#">schmidtcoeff</a>	42
5.1.1.86	<a href="#">schmidtprob</a>	43
5.1.1.87	<a href="#">schmidtU</a>	43
5.1.1.88	<a href="#">schmidtV</a>	44
5.1.1.89	<a href="#">shannon</a>	44
5.1.1.90	<a href="#">sinm</a>	45
5.1.1.91	<a href="#">spectralpowm</a>	45
5.1.1.92	<a href="#">sqrtm</a>	45
5.1.1.93	<a href="#">sum</a>	46
5.1.1.94	<a href="#">super</a>	46
5.1.1.95	<a href="#">syspermute</a>	47
5.1.1.96	<a href="#">trace</a>	47
5.1.1.97	<a href="#">transpose</a>	48
5.1.1.98	<a href="#">tsallis</a>	48

5.1.2	Variable Documentation	48
5.1.2.1	gt	48
5.1.2.2	rdevs	48
5.1.2.3	st	48
5.2	qpp::ct Namespace Reference	48
5.2.1	Function Documentation	49
5.2.1.1	omega	49
5.2.2	Variable Documentation	49
5.2.2.1	chop	49
5.2.2.2	ee	49
5.2.2.3	eps	49
5.2.2.4	maxn	49
5.2.2.5	pi	49
5.3	qpp::internal Namespace Reference	49
5.3.1	Function Documentation	50
5.3.1.1	_check_col_vector	50
5.3.1.2	_check_dims	50
5.3.1.3	_check_dims_match_cvect	50
5.3.1.4	_check_dims_match_mat	50
5.3.1.5	_check_dims_match_rvect	50
5.3.1.6	_check_eq_dims	50
5.3.1.7	_check_nonzero_size	50
5.3.1.8	_check_perm	50
5.3.1.9	_check_row_vector	50
5.3.1.10	_check_square_mat	50
5.3.1.11	_check_subsys_match_dims	50
5.3.1.12	_check_vector	50
5.3.1.13	_kron2	50
5.3.1.14	_multiidx2n	50
5.3.1.15	_n2multiidx	50
5.3.1.16	variadic_vector_emplace	51
5.3.1.17	variadic_vector_emplace	51
5.4	qpp::types Namespace Reference	51
5.4.1	Typedef Documentation	51
5.4.1.1	bra	51
5.4.1.2	cmat	51
5.4.1.3	cplx	51
5.4.1.4	dmat	51
5.4.1.5	DynMat	51
5.4.1.6	ket	51

<b>6</b>	<b>Class Documentation</b>	<b>53</b>
6.1	qpp::DiscreteDistribution Class Reference	53
6.1.1	Constructor & Destructor Documentation	53
6.1.1.1	DiscreteDistribution	53
6.1.1.2	DiscreteDistribution	53
6.1.1.3	DiscreteDistribution	53
6.1.2	Member Function Documentation	53
6.1.2.1	probabilities	53
6.1.2.2	sample	54
6.1.3	Member Data Documentation	54
6.1.3.1	_d	54
6.2	qpp::DiscreteDistributionAbsSquare Class Reference	54
6.2.1	Constructor & Destructor Documentation	55
6.2.1.1	DiscreteDistributionAbsSquare	55
6.2.1.2	DiscreteDistributionAbsSquare	55
6.2.1.3	DiscreteDistributionAbsSquare	55
6.2.1.4	DiscreteDistributionAbsSquare	55
6.2.2	Member Function Documentation	55
6.2.2.1	cplx2weights	55
6.2.2.2	probabilities	55
6.2.2.3	sample	55
6.2.3	Member Data Documentation	55
6.2.3.1	_d	55
6.3	qpp::Exception Class Reference	55
6.3.1	Member Enumeration Documentation	57
6.3.1.1	Type	57
6.3.2	Constructor & Destructor Documentation	58
6.3.2.1	Exception	58
6.3.2.2	Exception	58
6.3.3	Member Function Documentation	58
6.3.3.1	_construct_exception_msg	58
6.3.3.2	what	58
6.3.4	Member Data Documentation	58
6.3.4.1	_custom	58
6.3.4.2	_msg	58
6.3.4.3	_type	58
6.3.4.4	_where	58
6.4	qpp::Gates Class Reference	58
6.4.1	Constructor & Destructor Documentation	60
6.4.1.1	Gates	60

6.4.2	Member Function Documentation	60
6.4.2.1	apply	61
6.4.2.2	applyCTRL	61
6.4.2.3	CTRL	62
6.4.2.4	Fd	62
6.4.2.5	Id	62
6.4.2.6	Rn	62
6.4.2.7	Xd	63
6.4.2.8	Zd	63
6.4.3	Friends And Related Function Documentation	63
6.4.3.1	Singleton< const Gates >	63
6.4.4	Member Data Documentation	63
6.4.4.1	CNOTab	63
6.4.4.2	CNOTba	63
6.4.4.3	CZ	63
6.4.4.4	FRED	63
6.4.4.5	H	63
6.4.4.6	Id2	63
6.4.4.7	S	63
6.4.4.8	SWAP	63
6.4.4.9	T	63
6.4.4.10	TOF	64
6.4.4.11	X	64
6.4.4.12	Y	64
6.4.4.13	Z	64
6.5	qpp::NormalDistribution Class Reference	64
6.5.1	Constructor & Destructor Documentation	64
6.5.1.1	NormalDistribution	64
6.5.2	Member Function Documentation	64
6.5.2.1	sample	64
6.5.3	Member Data Documentation	64
6.5.3.1	_d	64
6.6	qpp::Qudit Class Reference	65
6.6.1	Constructor & Destructor Documentation	65
6.6.1.1	Qudit	65
6.6.2	Member Function Documentation	65
6.6.2.1	getD	65
6.6.2.2	getRho	65
6.6.2.3	measure	66
6.6.2.4	measure	66



6.6.3	Member Data Documentation . . . . .	66
6.6.3.1	_D . . . . .	66
6.6.3.2	_rho . . . . .	66
6.7	qpp::RandomDevices Class Reference . . . . .	67
6.7.1	Constructor & Destructor Documentation . . . . .	68
6.7.1.1	RandomDevices . . . . .	68
6.7.2	Friends And Related Function Documentation . . . . .	68
6.7.2.1	Singleton< const RandomDevices > . . . . .	68
6.7.3	Member Data Documentation . . . . .	68
6.7.3.1	_rd . . . . .	68
6.7.3.2	_rng . . . . .	68
6.8	qpp::Singleton< T > Class Template Reference . . . . .	68
6.8.1	Constructor & Destructor Documentation . . . . .	68
6.8.1.1	Singleton . . . . .	68
6.8.1.2	~Singleton . . . . .	69
6.8.1.3	Singleton . . . . .	69
6.8.2	Member Function Documentation . . . . .	69
6.8.2.1	get_instance . . . . .	69
6.8.2.2	operator= . . . . .	69
6.9	qpp::States Class Reference . . . . .	69
6.9.1	Constructor & Destructor Documentation . . . . .	70
6.9.1.1	States . . . . .	70
6.9.2	Friends And Related Function Documentation . . . . .	70
6.9.2.1	Singleton< const States > . . . . .	70
6.9.3	Member Data Documentation . . . . .	70
6.9.3.1	b00 . . . . .	70
6.9.3.2	b01 . . . . .	70
6.9.3.3	b10 . . . . .	70
6.9.3.4	b11 . . . . .	71
6.9.3.5	GHZ . . . . .	71
6.9.3.6	pb00 . . . . .	71
6.9.3.7	pb01 . . . . .	71
6.9.3.8	pb10 . . . . .	71
6.9.3.9	pb11 . . . . .	71
6.9.3.10	pGHZ . . . . .	71
6.9.3.11	pW . . . . .	71
6.9.3.12	px0 . . . . .	71
6.9.3.13	px1 . . . . .	71
6.9.3.14	py0 . . . . .	71
6.9.3.15	py1 . . . . .	71

6.9.3.16	pz0 . . . . .	71
6.9.3.17	pz1 . . . . .	71
6.9.3.18	W . . . . .	71
6.9.3.19	x0 . . . . .	71
6.9.3.20	x1 . . . . .	71
6.9.3.21	y0 . . . . .	71
6.9.3.22	y1 . . . . .	71
6.9.3.23	z0 . . . . .	71
6.9.3.24	z1 . . . . .	71
6.10	qpp::Timer Class Reference . . . . .	71
6.10.1	Constructor & Destructor Documentation . . . . .	72
6.10.1.1	Timer . . . . .	72
6.10.2	Member Function Documentation . . . . .	72
6.10.2.1	seconds . . . . .	72
6.10.2.2	tic . . . . .	72
6.10.2.3	toc . . . . .	72
6.10.3	Friends And Related Function Documentation . . . . .	72
6.10.3.1	operator<< . . . . .	72
6.10.4	Member Data Documentation . . . . .	72
6.10.4.1	_end . . . . .	72
6.10.4.2	_start . . . . .	72
6.11	qpp::UniformIntDistribution Class Reference . . . . .	72
6.11.1	Constructor & Destructor Documentation . . . . .	73
6.11.1.1	UniformIntDistribution . . . . .	73
6.11.2	Member Function Documentation . . . . .	73
6.11.2.1	sample . . . . .	73
6.11.3	Member Data Documentation . . . . .	73
6.11.3.1	_d . . . . .	73
6.12	qpp::UniformRealDistribution Class Reference . . . . .	73
6.12.1	Constructor & Destructor Documentation . . . . .	73
6.12.1.1	UniformRealDistribution . . . . .	73
6.12.2	Member Function Documentation . . . . .	73
6.12.2.1	sample . . . . .	74
6.12.3	Member Data Documentation . . . . .	74
6.12.3.1	_d . . . . .	74
<b>7</b>	<b>File Documentation</b>	<b>75</b>
7.1	include/channels.h File Reference . . . . .	75
7.2	include/classes/exception.h File Reference . . . . .	76
7.3	include/classes/gates.h File Reference . . . . .	76

7.4	include/classes/qudit.h File Reference . . . . .	77
7.5	include/classes/randevs.h File Reference . . . . .	77
7.6	include/classes/singleton.h File Reference . . . . .	78
7.6.1	Macro Definition Documentation . . . . .	78
7.6.1.1	CLASS_CONST_SINGLETON . . . . .	78
7.6.1.2	CLASS_SINGLETON . . . . .	78
7.7	include/classes/stat.h File Reference . . . . .	79
7.8	include/classes/states.h File Reference . . . . .	79
7.9	include/classes/timer.h File Reference . . . . .	80
7.10	include/constants.h File Reference . . . . .	80
7.11	include/entanglement.h File Reference . . . . .	81
7.12	include/entropies.h File Reference . . . . .	82
7.13	include/functions.h File Reference . . . . .	83
7.14	include/internal.h File Reference . . . . .	85
7.15	include/io.h File Reference . . . . .	86
7.16	include/matlab.h File Reference . . . . .	87
7.17	include/qpp.h File Reference . . . . .	89
7.18	include/random.h File Reference . . . . .	90
7.19	include/types.h File Reference . . . . .	91
	<b>Index</b>	<b>92</b>



# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">qpp</a>	9
<a href="#">qpp::ct</a>	48
<a href="#">qpp::internal</a>	49
<a href="#">qpp::types</a>	51



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

qpp::DiscreteDistribution . . . . .	53
qpp::DiscreteDistributionAbsSquare . . . . .	54
exception	
qpp::Exception . . . . .	55
qpp::NormalDistribution . . . . .	64
qpp::Qudit . . . . .	65
qpp::Singleton< T > . . . . .	68
qpp::Gates . . . . .	58
qpp::RandomDevices . . . . .	67
qpp::Singleton< const Gates > . . . . .	68
qpp::Singleton< const RandomDevices > . . . . .	68
qpp::Singleton< const States > . . . . .	68
qpp::States . . . . .	69
qpp::Timer . . . . .	71
qpp::UniformIntDistribution . . . . .	72
qpp::UniformRealDistribution . . . . .	73





## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">qpp::DiscreteDistribution</a>	53
<a href="#">qpp::DiscreteDistributionAbsSquare</a>	54
<a href="#">qpp::Exception</a>	55
<a href="#">qpp::Gates</a>	58
<a href="#">qpp::NormalDistribution</a>	64
<a href="#">qpp::Qudit</a>	65
<a href="#">qpp::RandomDevices</a>	67
<a href="#">qpp::Singleton&lt; T &gt;</a>	68
<a href="#">qpp::States</a>	69
<a href="#">qpp::Timer</a>	71
<a href="#">qpp::UniformIntDistribution</a>	72
<a href="#">qpp::UniformRealDistribution</a>	73



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

include/channels.h	75
include/constants.h	80
include/entanglement.h	81
include/entropies.h	82
include/functions.h	83
include/internal.h	85
include/io.h	86
include/matlab.h	87
include/qpp.h	89
include/random.h	90
include/types.h	91
include/classes/exception.h	76
include/classes/gates.h	76
include/classes/qudit.h	77
include/classes/randevs.h	77
include/classes/singleton.h	78
include/classes/stat.h	79
include/classes/states.h	79
include/classes/timer.h	80



## Chapter 5

# Namespace Documentation

### 5.1 qpp Namespace Reference

#### Namespaces

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

#### Classes

- class [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)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > conjugate` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > adjoint` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > inverse` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`Derived::Scalar trace` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`Derived::Scalar det` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`Derived::Scalar logdet` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`Derived::Scalar sum` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`double norm` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::cmat evals` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::cmat evecs` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::dmat hevals` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::cmat hevects` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::cmat funm` (const Eigen::MatrixBase< Derived > &A, `types::cplx`(\*f)(const `types::cplx` &))
- `template<typename Derived >`  
`types::cmat sqrtm` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::cmat absm` (const Eigen::MatrixBase< Derived > &A)

- `template<typename Derived >`  
`types::cmat expm` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::cmat logm` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::cmat sinm` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::cmat cosm` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::cmat spectralpowm` (const Eigen::MatrixBase< Derived > &A, const `types::cplx` z)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > powm` (const Eigen::MatrixBase< Derived > &A, std::size\_t n)
- `template<typename OutputScalar , typename Derived >`  
`types::DynMat< OutputScalar > cwise` (const Eigen::MatrixBase< Derived > &A, OutputScalar(\*) (const  
`typename Derived::Scalar &))`
- `template<typename T >`  
`types::DynMat< typename T::Scalar > kron` (const T &head)
- `template<typename T , typename... Args>`  
`types::DynMat< typename T::Scalar > kron` (const T &head, const Args &...tail)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > kron` (const std::vector< Derived > &As)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > kron` (const std::initializer\_list< Derived > &As)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > kronpow` (const Eigen::MatrixBase< Derived > &A, 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 > &sub-  
sys, 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< De-  
rived2 > &B)

- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > prj (const Eigen::MatrixBase< Derived > &V)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > expandout (const Eigen::MatrixBase< Derived > &A, 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)`

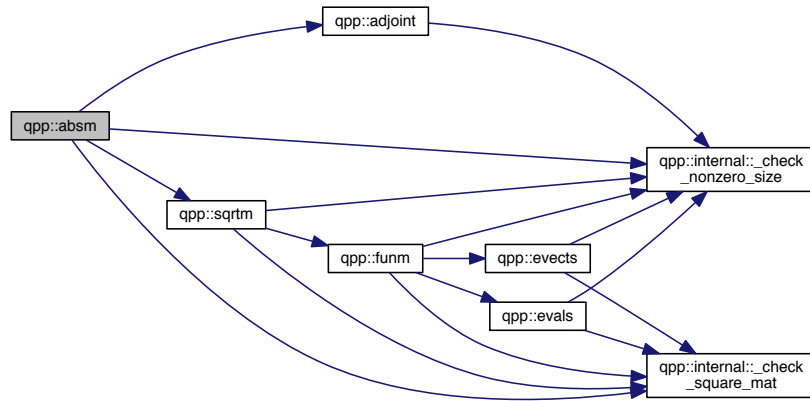
## Variables

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

### 5.1.1 Function Documentation

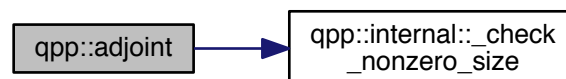
#### 5.1.1.1 `template<typename Derived > types::cmat qpp::absm ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



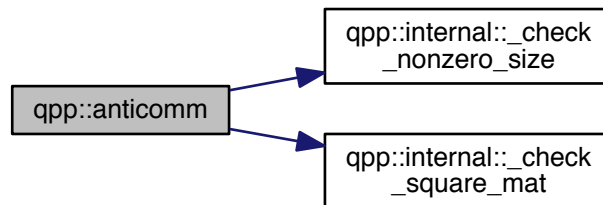
#### 5.1.1.2 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::adjoint ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



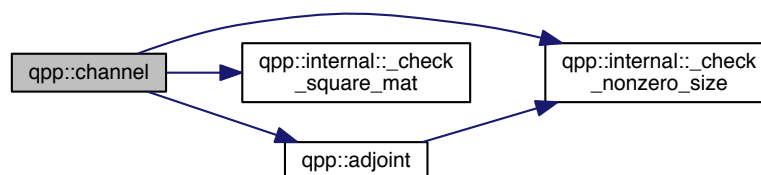
5.1.1.3 `template<typename Derived1 , typename Derived2 > types::DynMat<typename Derived1::Scalar> qpp::anticomm ( const Eigen::MatrixBase< Derived1 > & A, const Eigen::MatrixBase< Derived2 > & B )`

Here is the call graph for this function:



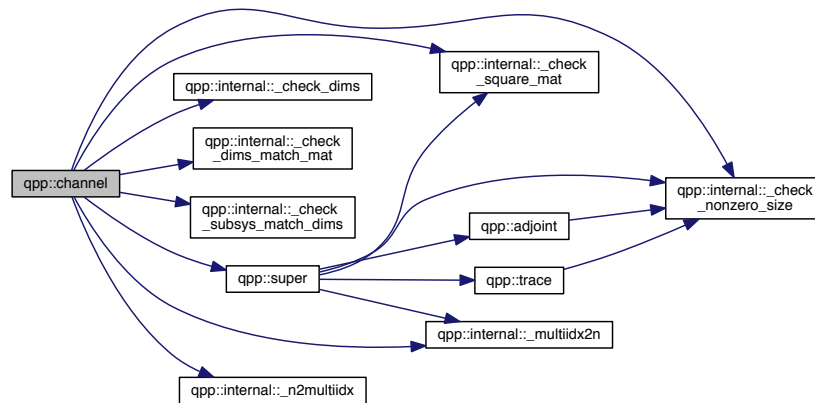
5.1.1.4 `template<typename Derived > types::cmat qpp::channel ( const Eigen::MatrixBase< Derived > & rho, const std::vector< types::cmat > & Ks )`

Here is the call graph for this function:



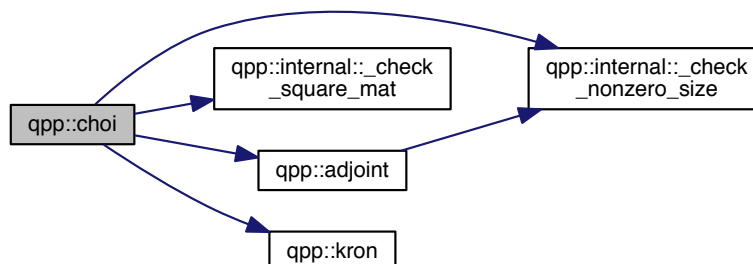
5.1.1.5 `template<typename Derived > types::cmat qpp::channel ( const Eigen::MatrixBase< Derived > & rho, const std::vector< types::cmat > & Ks, const std::vector< std::size_t > & subsys, const std::vector< std::size_t > & dims )`

Here is the call graph for this function:



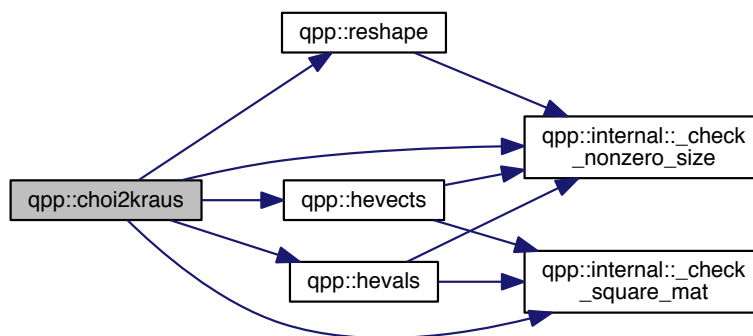
5.1.1.6 `types::cmat qpp::choi ( const std::vector< types::cmat > & Ks )`

Here is the call graph for this function:



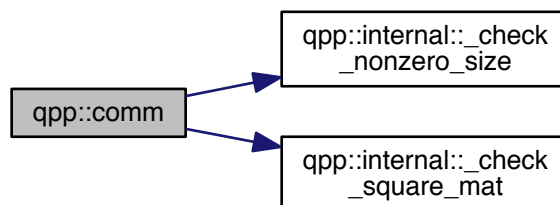
### 5.1.1.7 `std::vector<types::cmat> qpp::choi2kraus ( const types::cmat & A )`

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:



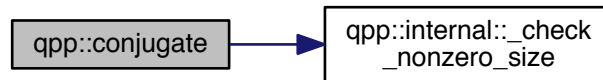
5.1.1.9 `std::vector<std::size_t> qpp::compperm ( const std::vector< std::size_t > & perm, const std::vector< std::size_t > & sigma )`

Here is the call graph for this function:



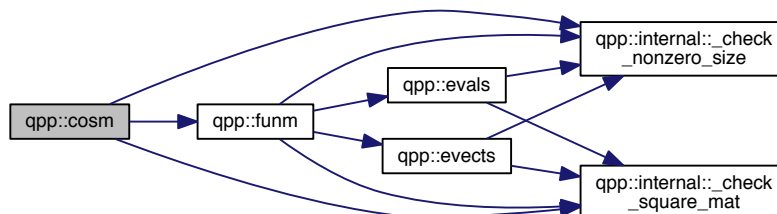
5.1.1.10 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::conjugate ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



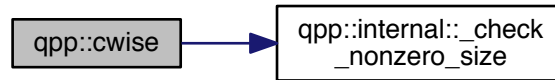
5.1.1.11 `template<typename Derived > types::cmat qpp::cosm ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



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

Here is the call graph for this function:



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

Here is the call graph for this function:



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

5.1.1.15 `template<typename T > void qpp::disp ( const T * x, const 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:





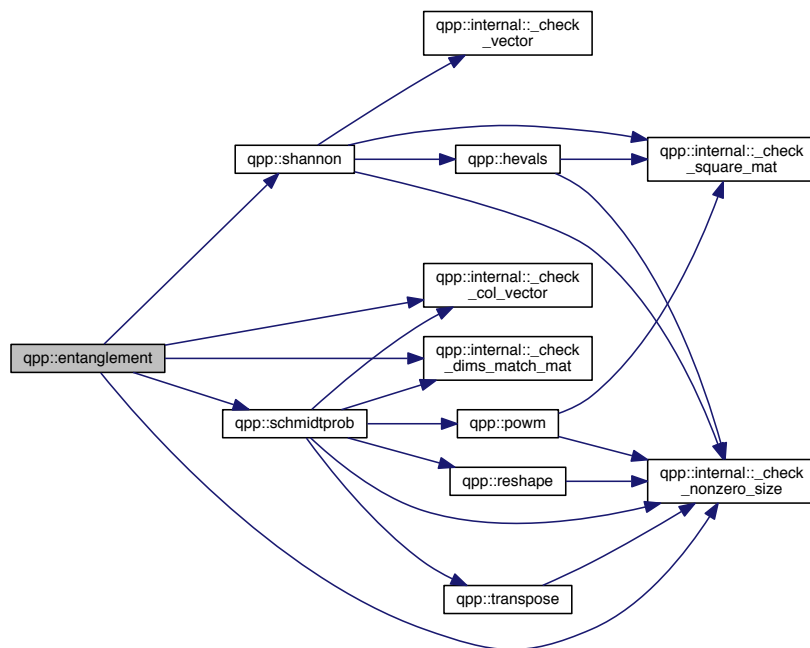
5.1.1.21 `void qpp::displn ( const types::cplx c, double chop = ct:::chop, std::ostream & os = std:::cout )`

Here is the call graph for this function:



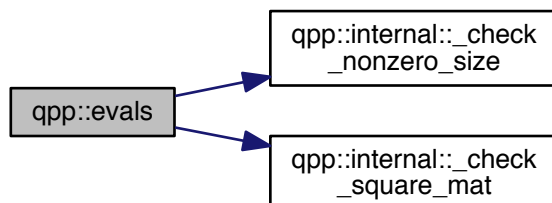
5.1.1.22 `template<typename Derived> double qpp::entanglement ( const Eigen::MatrixBase< Derived> & A, const std::vector< std::size_t> & dims )`

Here is the call graph for this function:



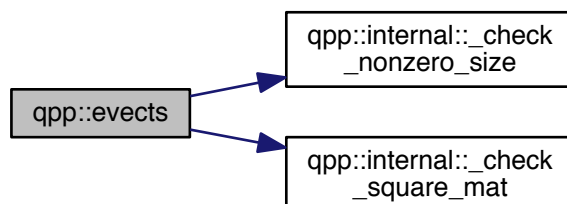
5.1.1.23 `template<typename Derived> types::cmat qpp::evals ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



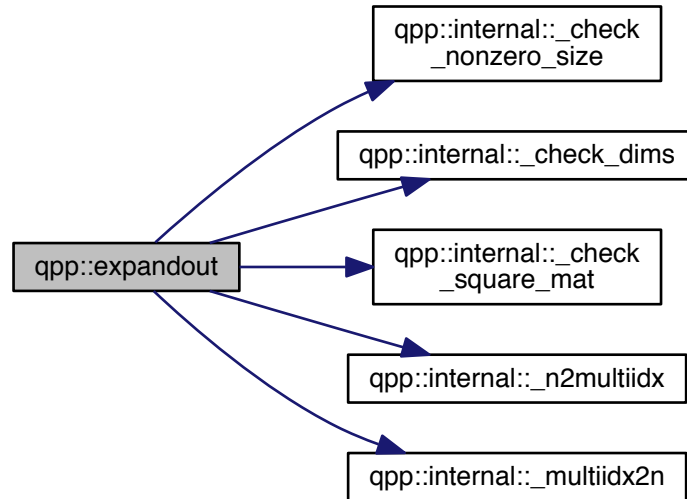
5.1.1.24 `template<typename Derived> types::cmat qpp::evecs ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



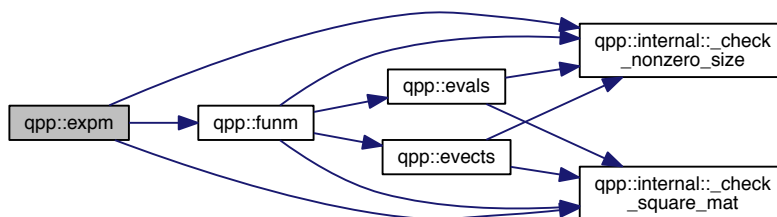
5.1.1.25 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::expandout ( const Eigen::MatrixBase< Derived> & A, std::size_t pos, const std::vector< std::size_t> & dims )`

Here is the call graph for this function:



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

Here is the call graph for this function:



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

Parameters

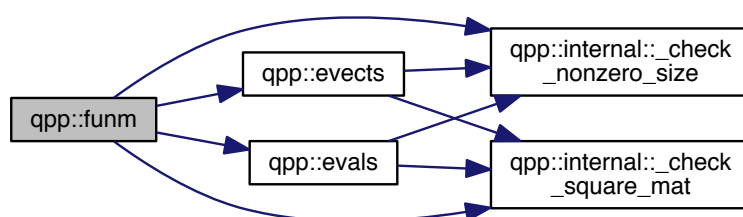
---

$A$	input matrix
$f$	function pointer

Returns

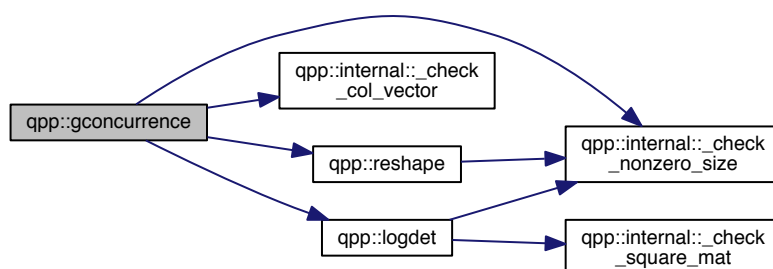
[types::cmat](#)

Here is the call graph for this function:



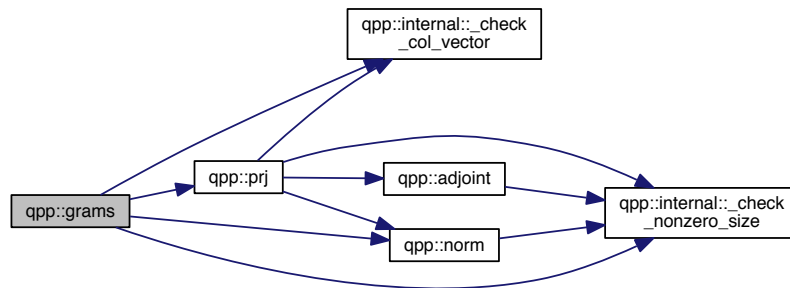
5.1.1.28 `template<typename Derived> double qpp::gconcurrency ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



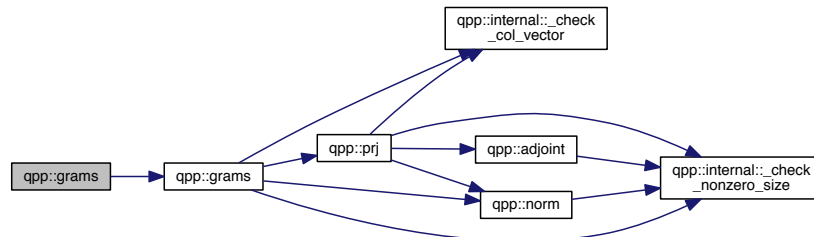
5.1.1.29 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::grams ( const std::vector< Derived > & Vs )`

Here is the call graph for this function:



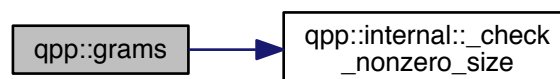
5.1.1.30 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::grams ( const std::initializer_list< Derived > & Vs )`

Here is the call graph for this function:



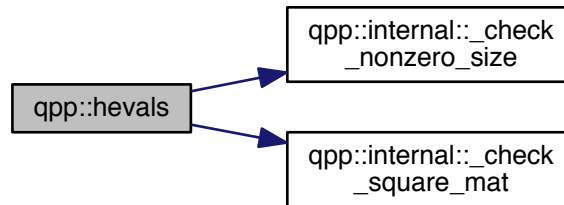
5.1.1.31 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::grams ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



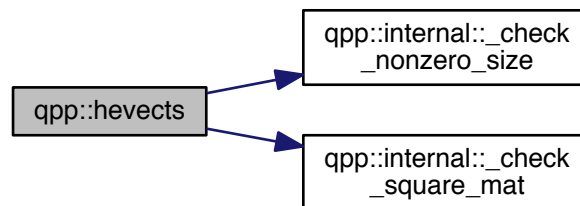
5.1.1.32 `template<typename Derived> types::dmat qpp::hevals ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



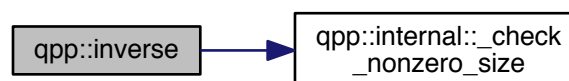
5.1.1.33 `template<typename Derived> types::cmat qpp::hevects ( const Eigen::MatrixBase< Derived > & A )`

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 )`

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 )`

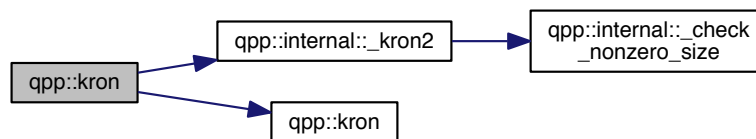
Here is the call graph for this function:



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:



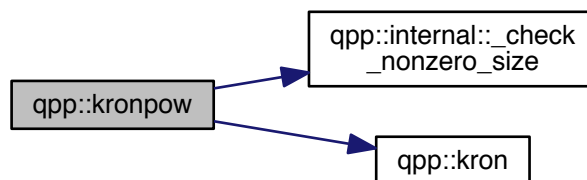
5.1.1.39 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::kron ( const std::initializer_list<Derived > & As )`

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 )`

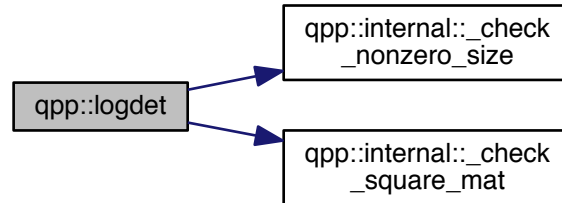
5.1.1.43 `template<> types::dmat qpp::loadMATLABmatrix ( const std::string & mat_file, const std::string & var_name )`

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



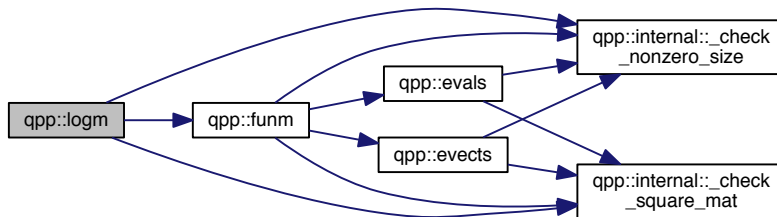
5.1.1.45 `template<typename Derived> Derived::Scalar qpp::logdet ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



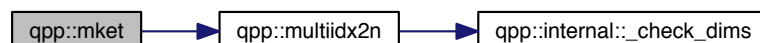
5.1.1.46 `template<typename Derived> types::cmat qpp::logm ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



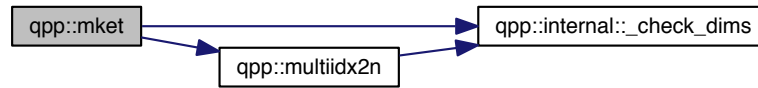
5.1.1.47 `types::ket qpp::mket ( const std::vector< std::size_t> & mask )`

Here is the call graph for this function:



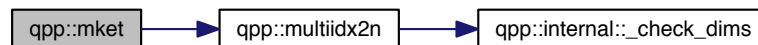
5.1.1.48 `types::ket qpp::mket ( const std::vector< std::size_t > & mask, const std::vector< std::size_t > & dims )`

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:



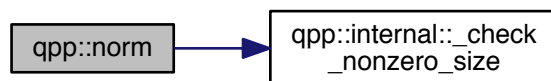
5.1.1.51 `std::vector<std::size_t> qpp::n2multiidx ( std::size_t n, const std::vector< std::size_t > & dims )`

Here is the call graph for this function:



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

Here is the call graph for this function:

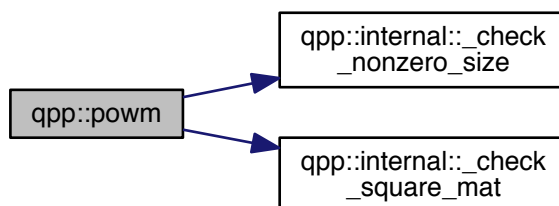


5.1.1.53 `constexpr std::complex<double> qpp::operator""_i ( unsigned long long int x )`

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

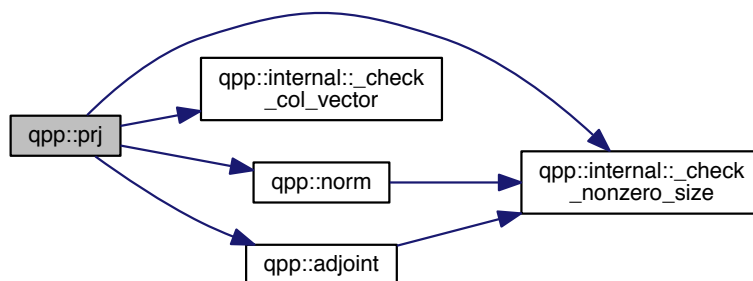
5.1.1.55 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::powm ( const Eigen::MatrixBase< Derived > & A, std::size_t n )`

Here is the call graph for this function:



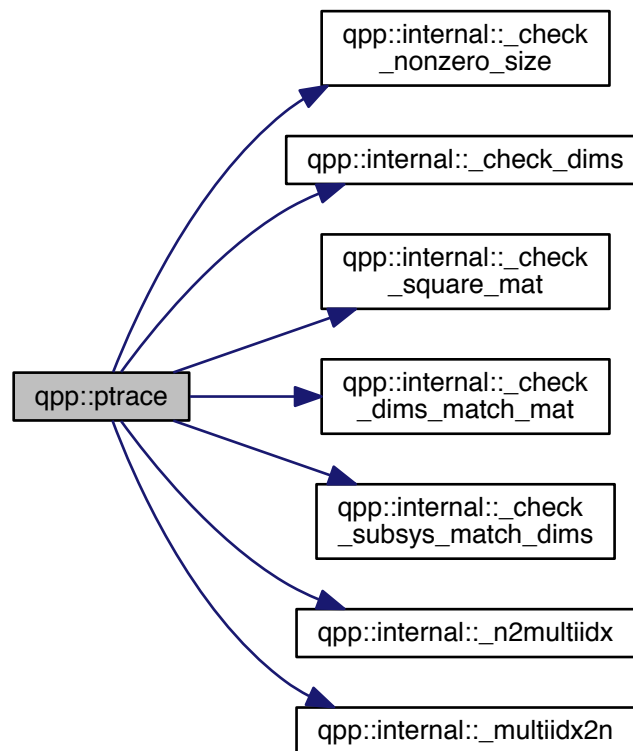
5.1.1.56 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::prj ( const Eigen::MatrixBase<Derived > & V )`

Here is the call graph for this function:



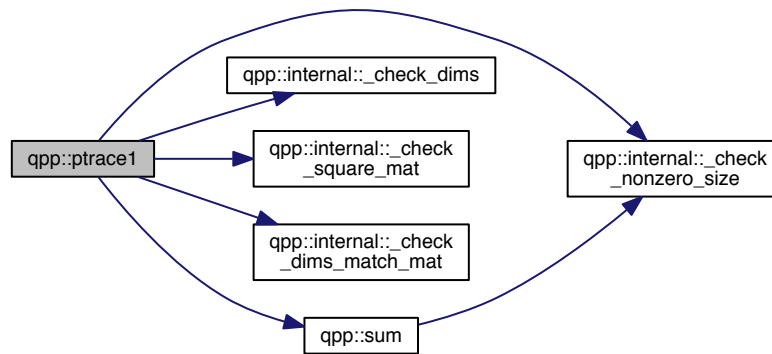
5.1.1.57 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::ptrace ( const Eigen::MatrixBase< Derived > & A, const std::vector< std::size_t > & subsys, const std::vector< std::size_t > & dims )`

Here is the call graph for this function:



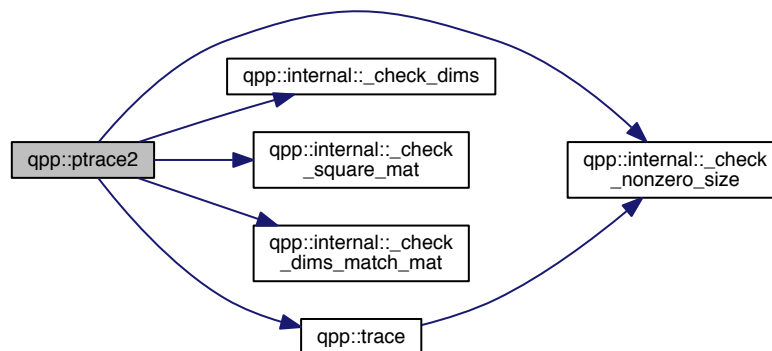
5.1.1.58 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::ptrace1 ( const Eigen::MatrixBase< Derived> & A, const std::vector< std::size_t> & dims )`

Here is the call graph for this function:



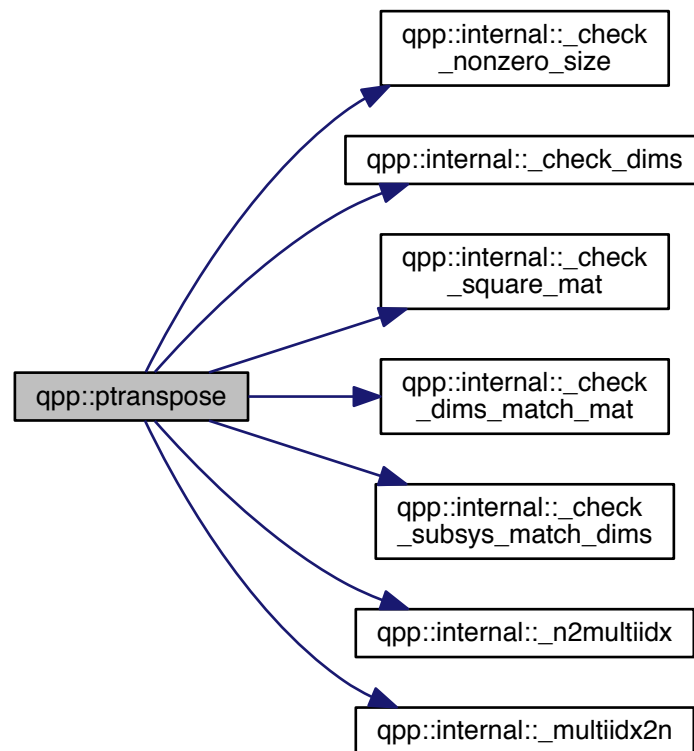
5.1.1.59 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::ptrace2 ( const Eigen::MatrixBase< Derived> & A, const std::vector< std::size_t> & dims )`

Here is the call graph for this function:



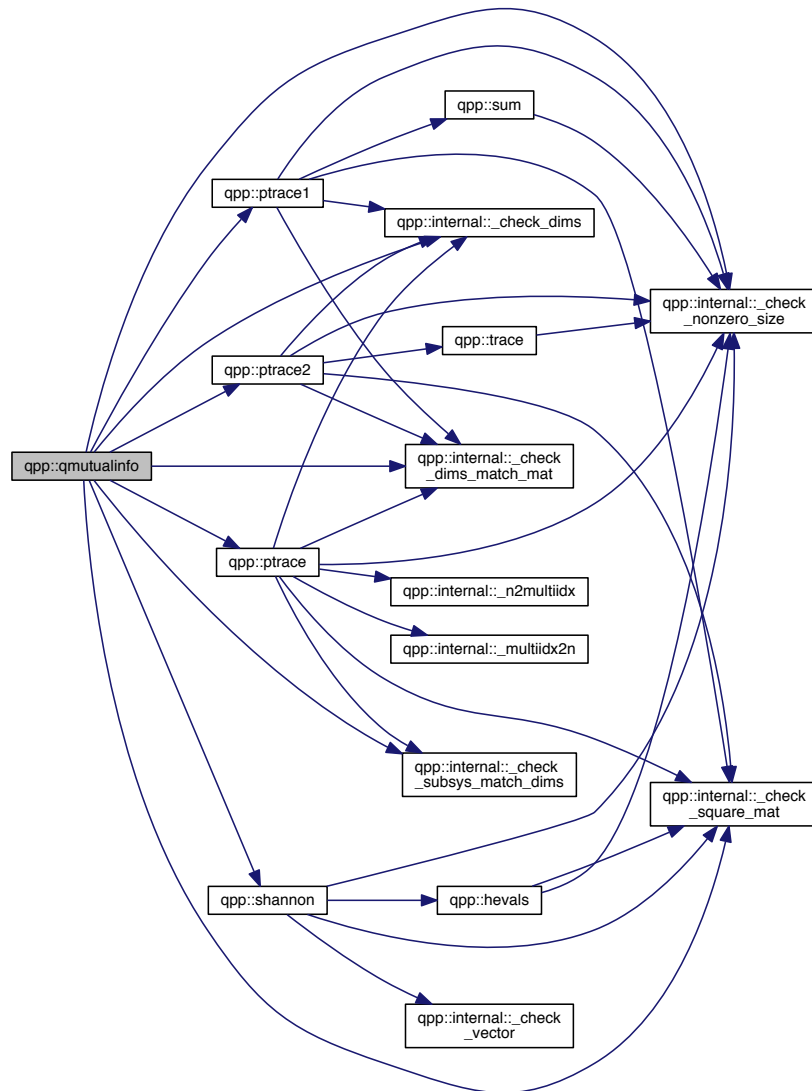
5.1.1.60 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::ptranspose ( const Eigen::MatrixBase< Derived> & A, const std::vector< std::size_t> & subsys, const std::vector< std::size_t> & dims )`

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 )`



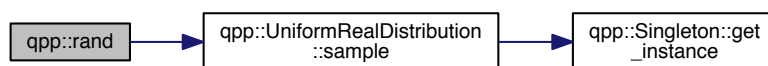
5.1.1.64 `template<> types::cmat qpp::rand ( std::size_t rows, std::size_t cols, double a, double b )`

Here is the call graph for this function:



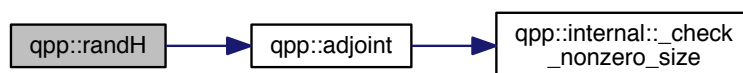
5.1.1.65 `double qpp::rand ( double a = 0, double b = 1 )`

Here is the call graph for this function:



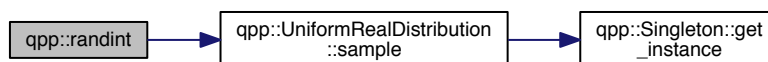
5.1.1.66 `types::cmat qpp::randH ( std::size_t D )`

Here is the call graph for this function:



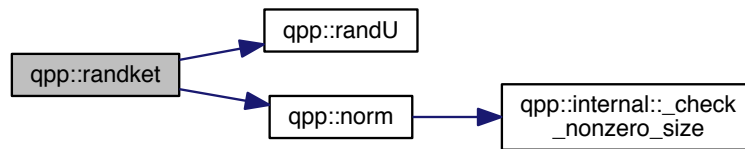
5.1.1.67 `long long qpp::randint ( long long a, long long b )`

Here is the call graph for this function:



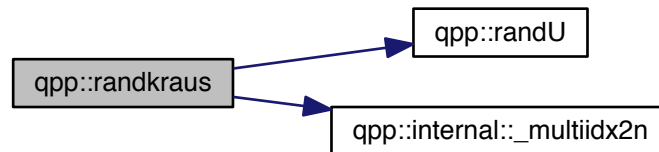
#### 5.1.1.68 `types::ket qpp::randket ( std::size_t D )`

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:



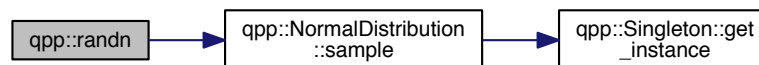
5.1.1.72 `template<> types::cmat qpp::randn ( std::size_t rows, std::size_t cols, double mean, double sigma )`

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:



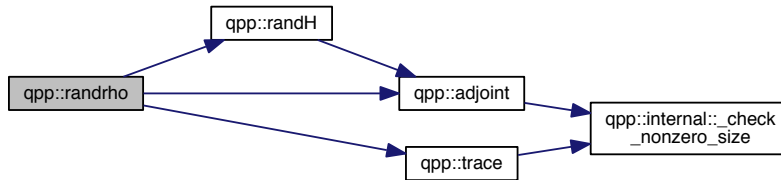
5.1.1.74 `std::vector<std::size_t> qpp::randperm ( std::size_t n )`

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 )`

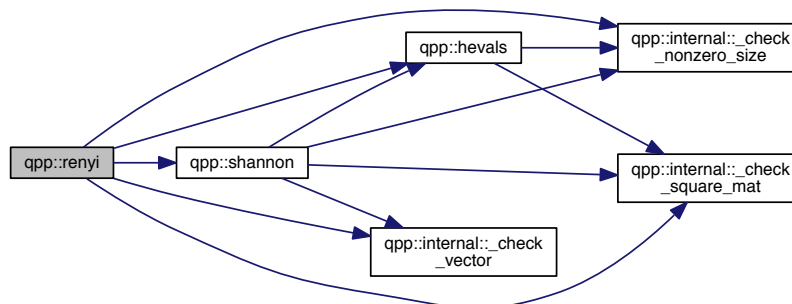
#### 5.1.1.77 `types::cmat qpp::randV ( std::size_t Din, std::size_t Dout )`

Here is the call graph for this function:



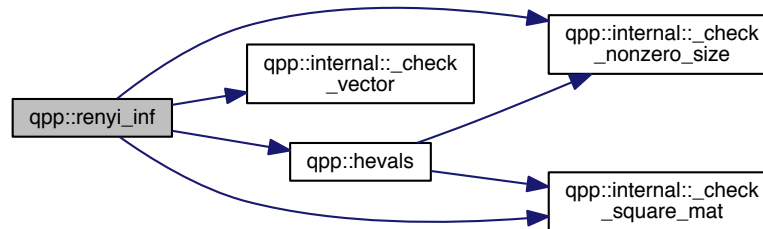
#### 5.1.1.78 `template<typename Derived> double qpp::renyi ( const double alpha, const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



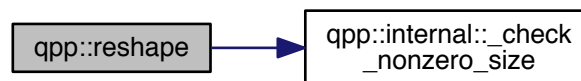
5.1.1.79 `template<typename Derived> double qpp::renyi_inf ( const Eigen::MatrixBase< Derived> & A )`

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:

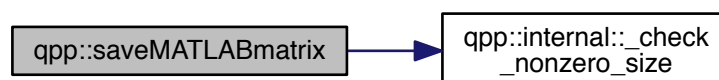


5.1.1.81 `template<typename Derived> void qpp::save ( const Eigen::MatrixBase< Derived> & A, const std::string & fname )`

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 )`

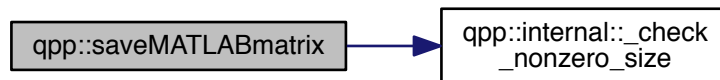
5.1.1.83 `template<> void qpp::saveMATLABmatrix ( const Eigen::MatrixBase< types::dmat> & A, const std::string & mat_file, const std::string & var_name, const std::string & mode )`

Here is the call graph for this function:



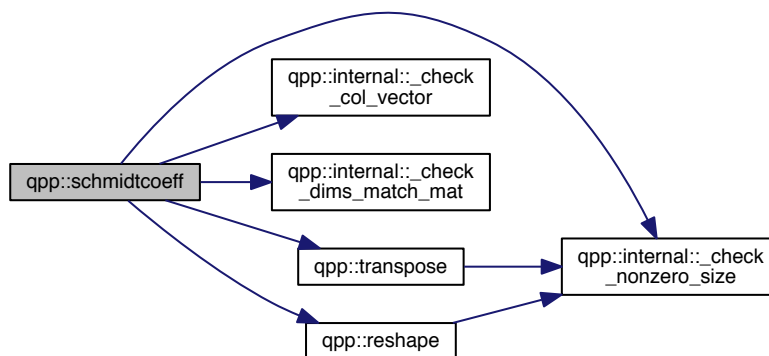
5.1.1.84 `template<> void qpp::saveMATLABmatrix ( const Eigen::MatrixBase< typename types::cmat > & A, const std::string & mat_file, const std::string & var_name, const std::string & mode )`

Here is the call graph for this function:



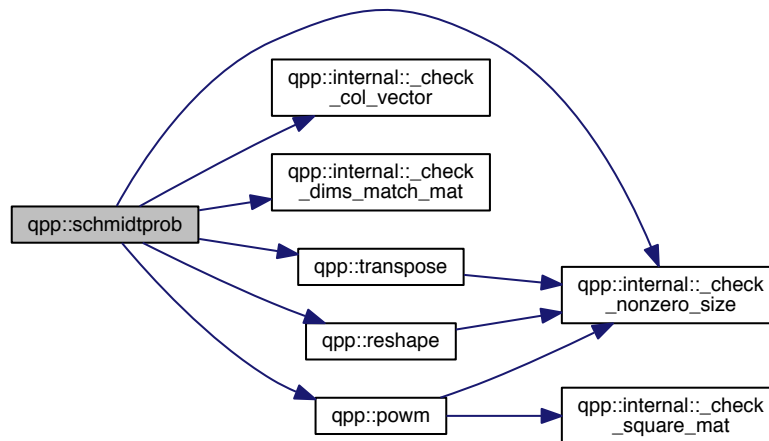
5.1.1.85 `template<typename Derived> types::cmat qpp::schmidtcoeff ( const Eigen::MatrixBase< Derived > & A, const std::vector< std::size_t > & dims )`

Here is the call graph for this function:



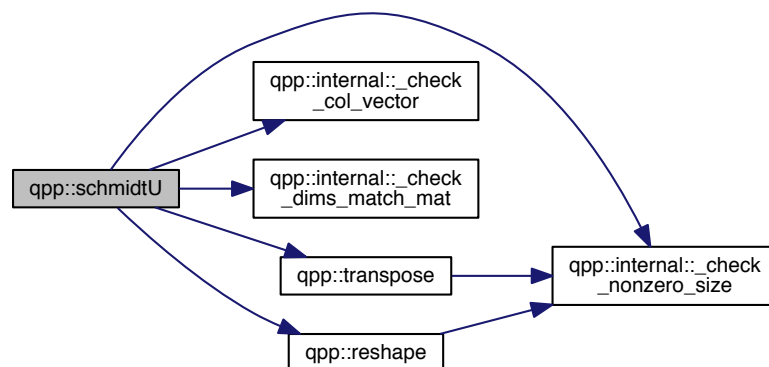
5.1.1.86 `template<typename Derived> types::cmat qpp::schmidtprob ( const Eigen::MatrixBase< Derived> & A, const std::vector< std::size_t> & dims )`

Here is the call graph for this function:



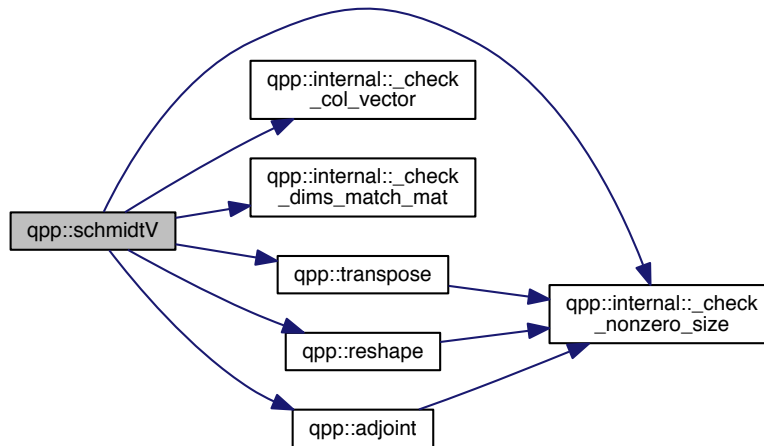
5.1.1.87 `template<typename Derived> types::cmat qpp::schmidtU ( const Eigen::MatrixBase< Derived> & A, const std::vector< std::size_t> & dims )`

Here is the call graph for this function:



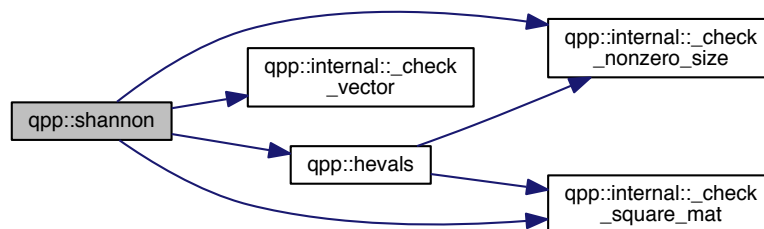
5.1.1.88 `template<typename Derived> types::cmat qpp::schmidtV ( const Eigen::MatrixBase< Derived > & A, const std::vector< std::size_t> & dims )`

Here is the call graph for this function:



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

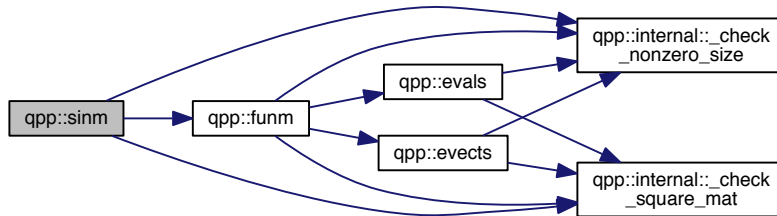
Here is the call graph for this function:





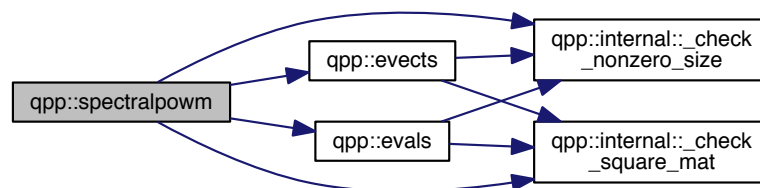
5.1.1.90 `template<typename Derived> types::cmat qpp::sinm ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



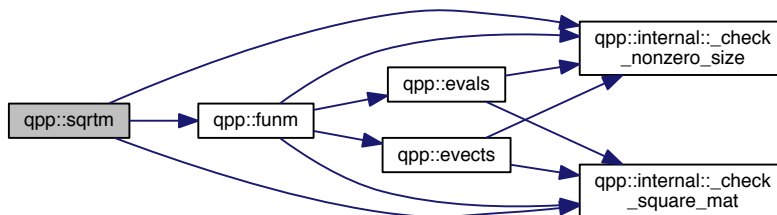
5.1.1.91 `template<typename Derived> types::cmat qpp::spectralpowm ( const Eigen::MatrixBase< Derived > & A, const types::cplx z )`

Here is the call graph for this function:



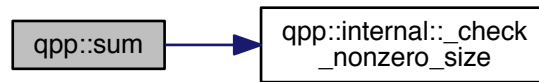
5.1.1.92 `template<typename Derived> types::cmat qpp::sqrtm ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



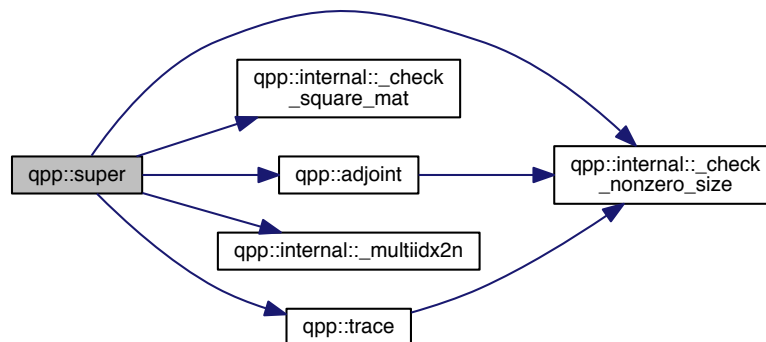
#### 5.1.1.93 `template<typename Derived > Derived::Scalar qpp::sum ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



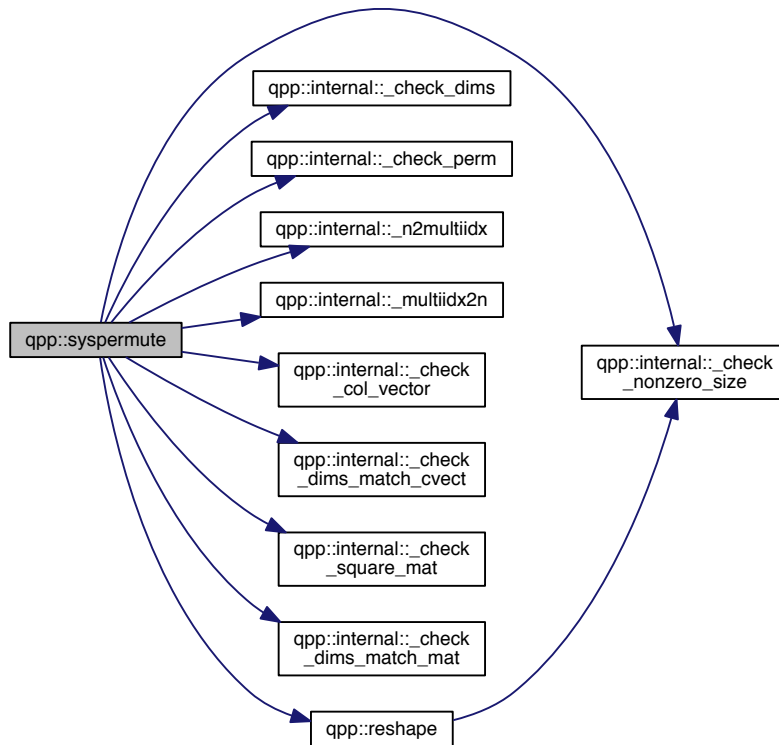
#### 5.1.1.94 `types::cmat qpp::super ( const std::vector< types::cmat > & Ks )`

Here is the call graph for this function:



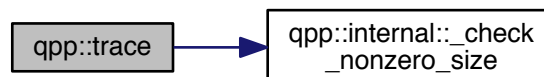
5.1.1.95 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::syspermute ( const Eigen::MatrixBase< Derived> & A, const std::vector< std::size_t> & perm, const std::vector< std::size_t> & dims )`

Here is the call graph for this function:



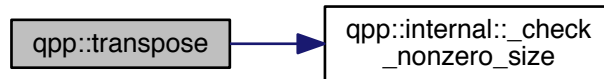
5.1.1.96 `template<typename Derived> Derived::Scalar qpp::trace ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



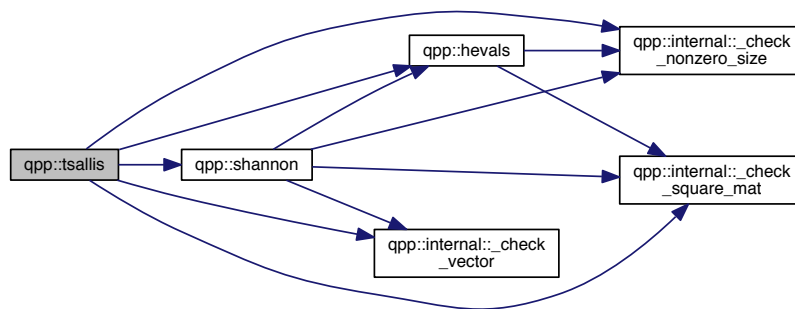
5.1.1.97 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::transpose ( const Eigen::MatrixBase< Derived> & A )`

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 )`

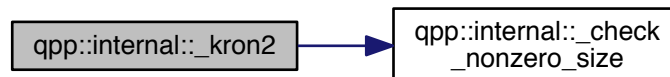
5.3.1.10 `template<typename Derived > bool qpp::internal::_check_square_mat ( const Eigen::MatrixBase< Derived > & A )`

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]`

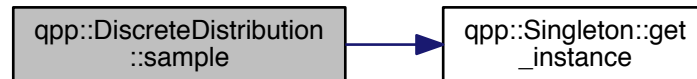
6.1.1.3 `qpp::DiscreteDistribution::DiscreteDistribution ( std::vector< double > weights ) [inline]`

#### 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]`

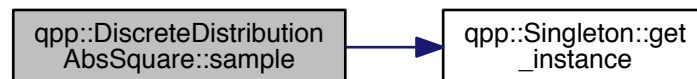
### 6.2.2 Member Function Documentation

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]`

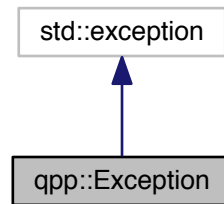
The documentation for this class was generated from the following file:

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

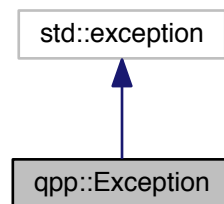
## 6.3 qpp::Exception Class Reference

```
#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

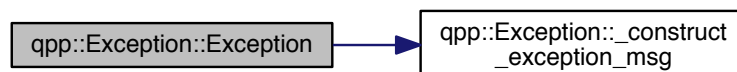
6.3.2.1 `qpp::Exception::Exception ( const std::string & where, const Type & type )` `[inline]`

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]`

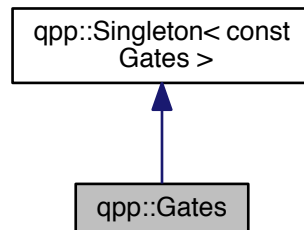
The documentation for this class was generated from the following file:

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

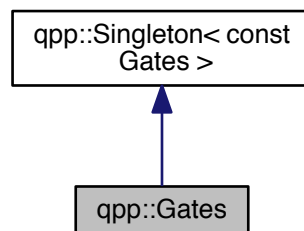
## 6.4 qpp::Gates Class Reference

```
#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 H](#)
- [types::cmat X](#)
- [types::cmat Y](#)
- [types::cmat Z](#)
- [types::cmat S](#)
- [types::cmat T](#)
- [types::cmat CNOTab](#)
- [types::cmat CZ](#)
- [types::cmat CNOTba](#)
- [types::cmat SWAP](#)
- [types::cmat TOF](#)
- [types::cmat FRED](#)

## Private Member Functions

- [Gates](#) ()

## Friends

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

## Additional Inherited Members

### 6.4.1 Constructor & Destructor Documentation

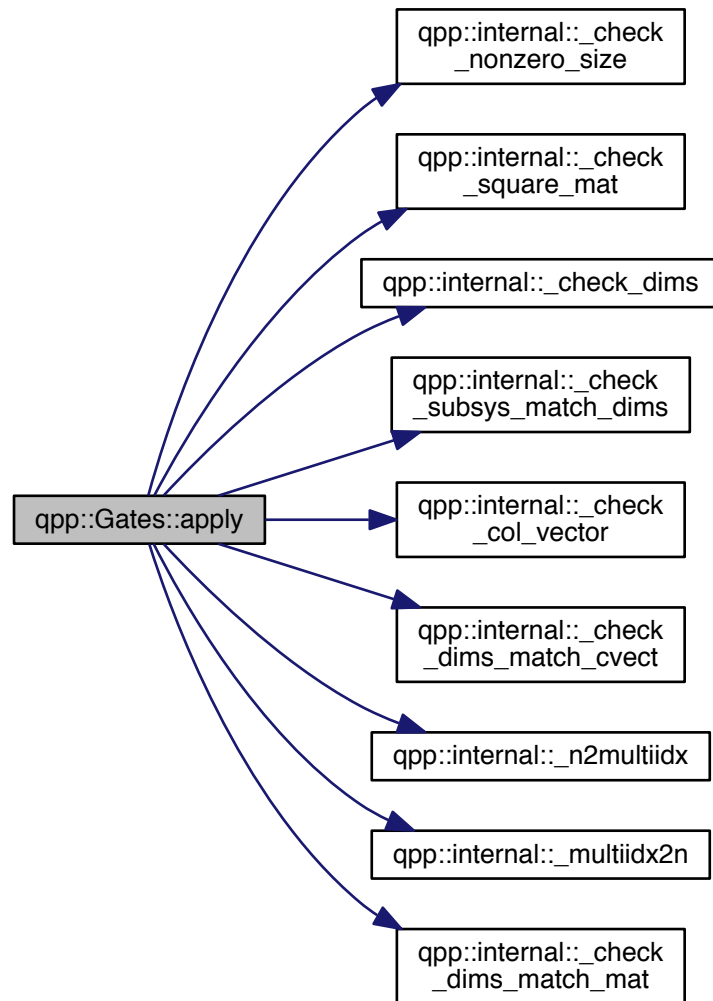
6.4.1.1 `qpp::Gates::Gates ( )` `[inline]`, `[private]`

### 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]
```

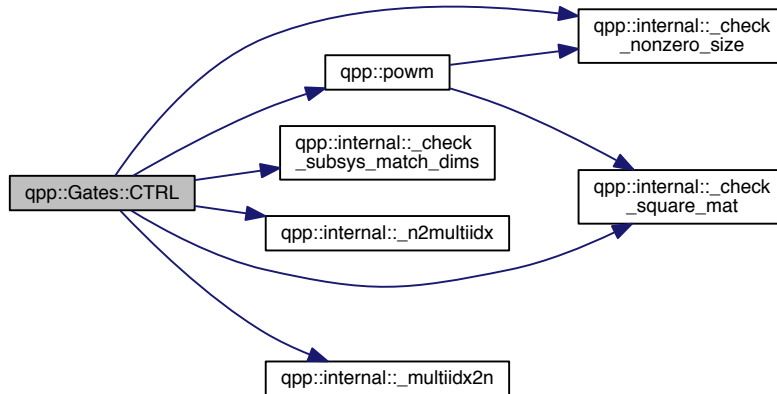
Here is the call graph for this function:



```
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:

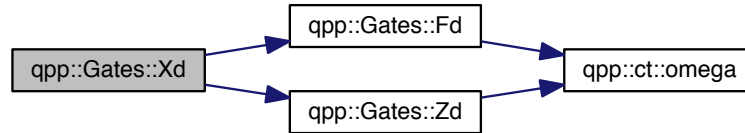


6.4.2.5 `template<typename Derived = Eigen::MatrixXcd> Derived qpp::Gates::Id ( std::size_t D ) const [inline]`

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`

#### 6.4.4.2 `types::cmat qpp::Gates::CNOTba`

#### 6.4.4.3 `types::cmat qpp::Gates::CZ`

#### 6.4.4.4 `types::cmat qpp::Gates::FRED`

#### 6.4.4.5 `types::cmat qpp::Gates::H`

#### 6.4.4.6 `types::cmat qpp::Gates::Id2`

#### 6.4.4.7 `types::cmat qpp::Gates::S`

#### 6.4.4.8 `types::cmat qpp::Gates::SWAP`

#### 6.4.4.9 `types::cmat qpp::Gates::T`

6.4.4.10 `types::cmat qpp::Gates::TOF`

6.4.4.11 `types::cmat qpp::Gates::X`

6.4.4.12 `types::cmat qpp::Gates::Y`

6.4.4.13 `types::cmat qpp::Gates::Z`

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

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

## 6.5 qpp::NormalDistribution Class Reference

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

### Public Member Functions

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

### Protected Attributes

- `std::normal_distribution _d`

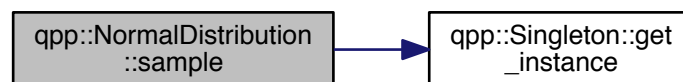
#### 6.5.1 Constructor & Destructor Documentation

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

#### 6.5.2 Member Function Documentation

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

Here is the call graph for this function:



#### 6.5.3 Member Data Documentation

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

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

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

## 6.6 qpp::Qudit Class Reference

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

### Public Member Functions

- [Qudit](#) (const [types::cmat](#) &rho=[States::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

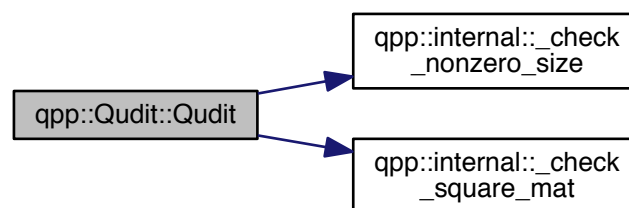
### Private Attributes

- [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:



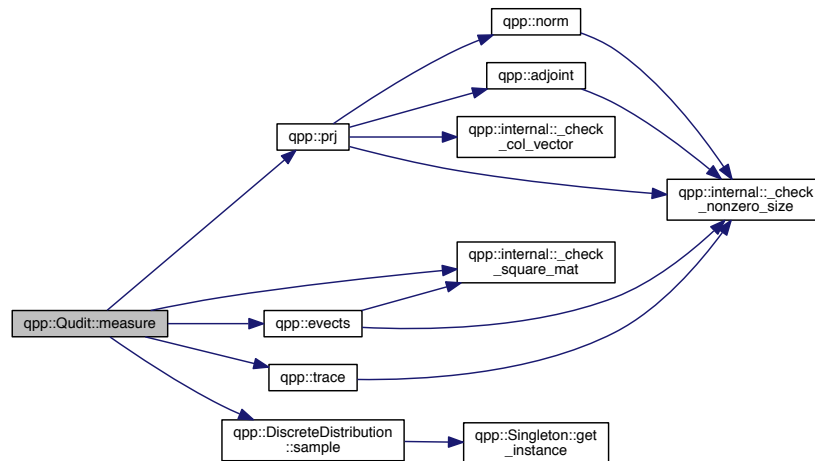
### 6.6.2 Member Function Documentation

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

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

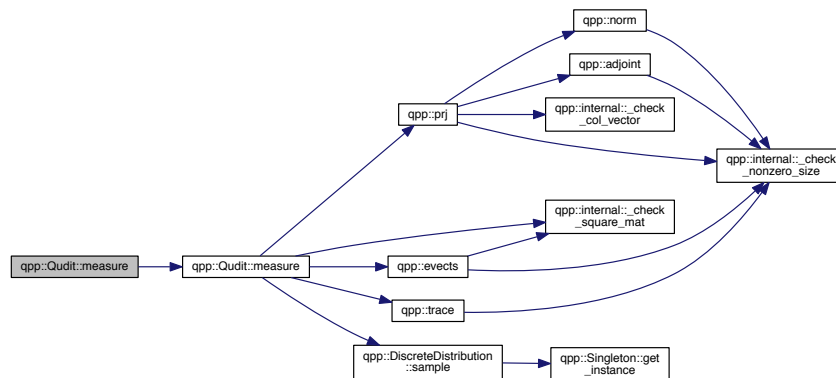
### 6.6.2.3 `std::size_t qpp::Qudit::measure ( const types::cmat & U, bool destructive = false ) [inline]`

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]`

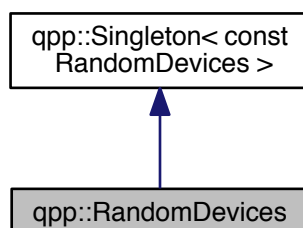
The documentation for this class was generated from the following file:

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

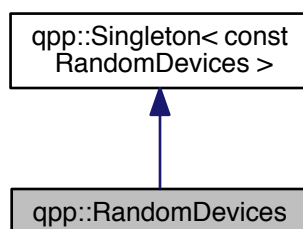
## 6.7 qpp::RandomDevices Class Reference

```
#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]`

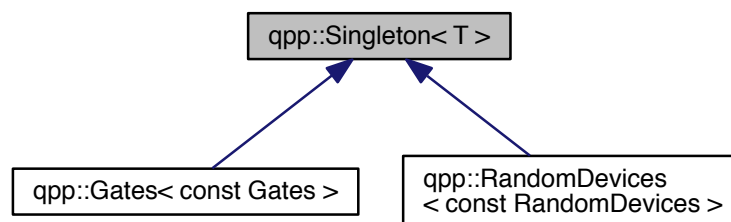
The documentation for this class was generated from the following file:

- [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]`

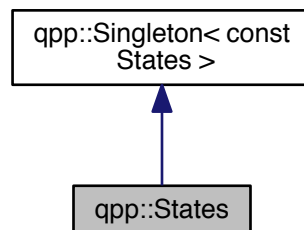
The documentation for this class was generated from the following file:

- `include/classes/singleton.h`

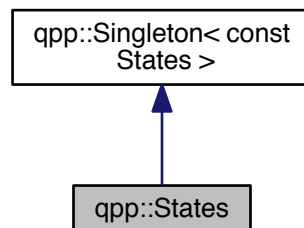
## 6.9 qpp::States Class Reference

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

Inheritance diagram for qpp::States:



Collaboration diagram for qpp::States:



## Public Attributes

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

## Private Member Functions

- [States \(\)](#)

## 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`

6.9.3.2 `types::ket qpp::States::b01`

6.9.3.3 `types::ket qpp::States::b10`

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

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](#)

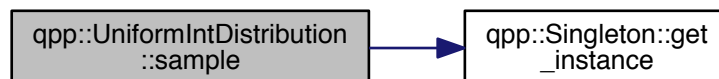
### 6.11.1 Constructor & Destructor Documentation

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`

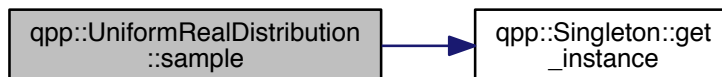
### 6.12.1 Constructor & Destructor Documentation

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:

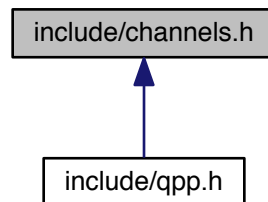
- `include/classes/stat.h`

## Chapter 7

# File Documentation

### 7.1 include/channels.h File Reference

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



### Namespaces

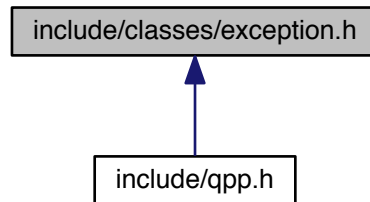
- [qpp](#)

### 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

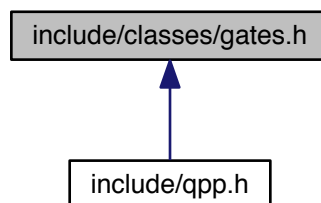
- class [qpp::Exception](#)

### 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](#)

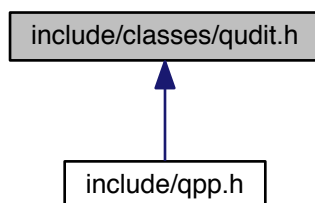
### Namespaces

- [qpp](#)



## 7.4 include/classes/qudit.h File Reference

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



### Classes

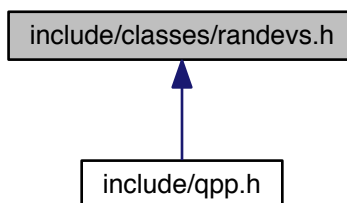
- class [qpp::Qudit](#)

### Namespaces

- [qpp](#)

## 7.5 include/classes/randevs.h File Reference

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



### Classes

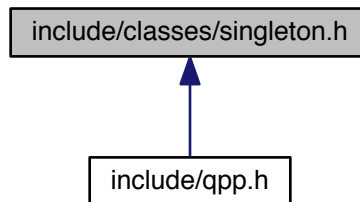
- class [qpp::RandomDevices](#)

### 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>\n{\n    friend class Singleton<const Foo>;
```

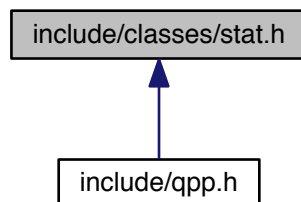
##### 7.6.1.2 `#define CLASS_SINGLETON( Foo )`

###### Value:

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

## 7.7 include/classes/stat.h File Reference

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



### Classes

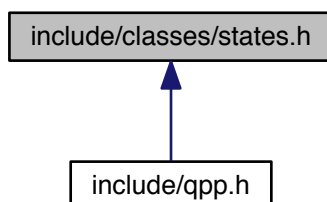
- class [qpp::NormalDistribution](#)
- class [qpp::UniformRealDistribution](#)
- class [qpp::UniformIntDistribution](#)
- class [qpp::DiscreteDistribution](#)
- class [qpp::DiscreteDistributionAbsSquare](#)

### Namespaces

- [qpp](#)

## 7.8 include/classes/states.h File Reference

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



### Classes

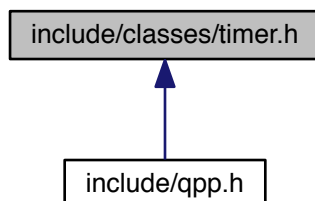
- class [qpp::States](#)

## Namespaces

- [qpp](#)

## 7.9 include/classes/timer.h File Reference

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



## Classes

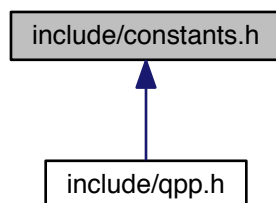
- class [qpp::Timer](#)

## 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

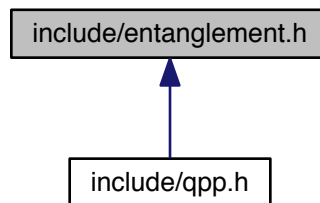
- constexpr std::complex< double > [qpp::operator""\\_i](#) (unsigned long long int x)
- constexpr std::complex< double > [qpp::operator""\\_i](#) (long double x)
- std::complex< double > [qpp::ct::omega](#) (std::size\_t D)

## 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](#)

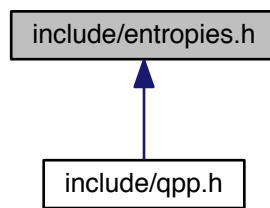
## Functions

- 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

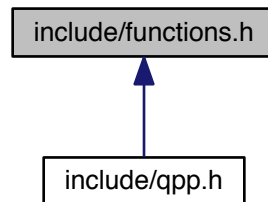
- [qpp](#)

## 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)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::conjugate (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::adjoint (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::inverse (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`Derived::Scalar qpp::trace (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`Derived::Scalar qpp::det (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`Derived::Scalar qpp::logdet (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`Derived::Scalar qpp::sum (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`double qpp::norm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`types::cmat qpp::evals (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`types::cmat qpp::evecs (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`types::dmat qpp::hevals (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`types::cmat qpp::hevecs (const Eigen::MatrixBase< Derived > &A)`

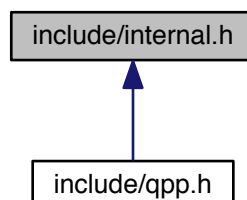
- `template<typename Derived >`  
`types::cmat qpp::funm (const Eigen::MatrixBase< Derived > &A, types::cplx(*f)(const types::cplx &))`
- `template<typename Derived >`  
`types::cmat qpp::sqrtm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`types::cmat qpp::absm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`types::cmat qpp::expm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`types::cmat qpp::logm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`types::cmat qpp::sinm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`types::cmat qpp::cosm (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`types::cmat qpp::spectralpwm (const Eigen::MatrixBase< Derived > &A, const types::cplx z)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::pwm (const Eigen::MatrixBase< Derived > &A, std::size_t n)`
- `template<typename OutputScalar, typename Derived >`  
`types::DynMat< OutputScalar > qpp::cwise (const Eigen::MatrixBase< Derived > &A, OutputScalar(*f)(const typename Derived::Scalar &))`
- `template<typename T >`  
`types::DynMat< typename T::Scalar > qpp::kron (const T &head)`
- `template<typename T, typename... Args>`  
`types::DynMat< typename T::Scalar > qpp::kron (const T &head, const Args &...tail)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::kron (const std::vector< Derived > &As)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::kron (const std::initializer_list< Derived > &As)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::kronpow (const Eigen::MatrixBase< Derived > &A, 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 > &subs, 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 > &subs, 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< 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

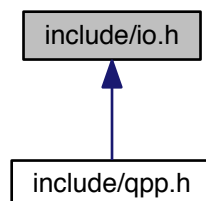
- [qpp](#)
- [qpp::internal](#)

## 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

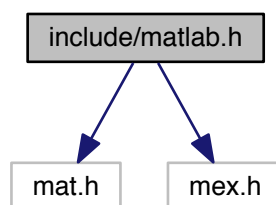
- [qpp](#)

## 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

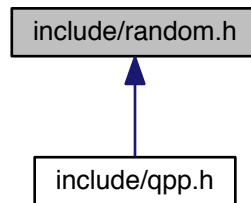
- [qpp](#)

### 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

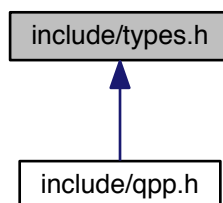
- [qpp](#)

### 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, [13](#)
- adjoint
  - qpp, [14](#)
- anticomm
  - qpp, [14](#)
- CUSTOM\_EXCEPTION
  - qpp::Exception, [57](#)
- channel
  - qpp, [15](#)
- choi
  - qpp, [16](#)
- choi2kraus
  - qpp, [16](#)
- comm
  - qpp, [17](#)
- compperm
  - qpp, [17](#)
- conjugate
  - qpp, [18](#)
- cosm
  - qpp, [18](#)
- cwise
  - qpp, [18](#)
- DIMS\_INVALID
  - qpp::Exception, [57](#)
- DIMS\_MISMATCH\_CVECTOR
  - qpp::Exception, [57](#)
- DIMS\_MISMATCH\_MATRIX
  - qpp::Exception, [57](#)
- DIMS\_MISMATCH\_RVECTOR
  - qpp::Exception, [57](#)
- DIMS\_MISMATCH\_VECTOR
  - qpp::Exception, [57](#)
- DIMS\_NOT\_EQUAL
  - qpp::Exception, [57](#)
- det
  - qpp, [19](#)
- disp
  - qpp, [19](#)
- displn
  - qpp, [19](#), [20](#)
- entanglement
  - qpp, [21](#)
- evals
  - qpp, [21](#)
- evects
  - qpp, [22](#)
- expandout
  - qpp, [22](#)
- expm
  - qpp, [23](#)
- funm
  - qpp, [23](#)
- gconcurrency
  - qpp, [24](#)
- grams
  - qpp, [24](#), [25](#)
- gt
  - qpp, [48](#)
- hevals
  - qpp, [25](#)
- hevects
  - qpp, [26](#)
- inverse
  - qpp, [26](#)
- invperm
  - qpp, [26](#)
- kron
  - qpp, [27](#)
- kronpow
  - qpp, [28](#)
- load
  - qpp, [28](#)
- logdet
  - qpp, [28](#)
- logm
  - qpp, [29](#)
- MATRIX\_NOT\_CVECTOR
  - qpp::Exception, [57](#)
- MATRIX\_NOT\_RVECTOR
  - qpp::Exception, [57](#)
- MATRIX\_NOT\_SQUARE
  - qpp::Exception, [57](#)
- MATRIX\_NOT\_SQUARE\_OR\_CVECTOR
  - qpp::Exception, [57](#)
- MATRIX\_NOT\_SQUARE\_OR\_RVECTOR
  - qpp::Exception, [57](#)
- MATRIX\_NOT\_SQUARE\_OR\_VECTOR
  - qpp::Exception, [57](#)
- MATRIX\_NOT\_VECTOR
  - qpp, [57](#)



- qpp::Exception, 57
- mket
  - qpp, 29, 30
- multiidx2n
  - qpp, 30
- n2multiidx
  - qpp, 30
- NOT\_BIPARTITE
  - qpp::Exception, 57
- NOT\_QUBIT\_GATE
  - qpp::Exception, 57
- NOT\_QUBIT\_SUBSYS
  - qpp::Exception, 57
- norm
  - qpp, 31
- OUT\_OF\_RANGE
  - qpp::Exception, 57
- PERM\_INVALID
  - qpp::Exception, 57
- powm
  - qpp, 31
- prj
  - qpp, 31
- ptrace
  - qpp, 32
- ptrace1
  - qpp, 33
- ptrace2
  - qpp, 34
- ptranspose
  - qpp, 34
- qmutualinfo
  - qpp, 35
- qpp, 9
  - absm, 13
  - adjoint, 14
  - anticomm, 14
  - channel, 15
  - choi, 16
  - choi2kraus, 16
  - comm, 17
  - compperm, 17
  - conjugate, 18
  - cosm, 18
  - cwise, 18
  - det, 19
  - disp, 19
  - displn, 19, 20
  - entanglement, 21
  - evals, 21
  - evects, 22
  - expandout, 22
  - expm, 23
  - funm, 23
  - gconcurrency, 24
  - grams, 24, 25
  - gt, 48
  - hevals, 25
  - hevects, 26
  - inverse, 26
  - invperm, 26
  - kron, 27
  - kronpow, 28
  - load, 28
  - logdet, 28
  - logm, 29
  - mket, 29, 30
  - multiidx2n, 30
  - n2multiidx, 30
  - norm, 31
  - powm, 31
  - prj, 31
  - ptrace, 32
  - ptrace1, 33
  - ptrace2, 34
  - ptranspose, 34
  - qmutualinfo, 35
  - rand, 36, 37
  - randint, 37
  - randket, 37
  - randkraus, 38
  - randn, 38, 39
  - randperm, 39
  - randrho, 39
  - rdevs, 48
  - renyi, 40
  - reshape, 41
  - save, 41
  - schmidtcoeff, 42
  - schmidtprob, 42
  - shannon, 44
  - sinm, 44
  - spectralpowm, 45
  - sqrtn, 45
  - st, 48
  - sum, 45
  - super, 46
  - syspermute, 46
  - trace, 47
  - transpose, 47
  - tsallis, 48
- qpp::Exception
  - CUSTOM\_EXCEPTION, 57
  - DIMS\_INVALID, 57
  - DIMS\_MISMATCH\_CVECTOR, 57
  - DIMS\_MISMATCH\_MATRIX, 57
  - DIMS\_MISMATCH\_RVECTOR, 57
  - DIMS\_MISMATCH\_VECTOR, 57
  - DIMS\_NOT\_EQUAL, 57
  - MATRIX\_NOT\_CVECTOR, 57
  - MATRIX\_NOT\_RVECTOR, 57
  - MATRIX\_NOT\_SQUARE, 57
  - MATRIX\_NOT\_SQUARE\_OR\_CVECTOR, 57

- MATRIX\_NOT\_SQUARE\_OR\_RVECTOR, [57](#)
  - MATRIX\_NOT\_SQUARE\_OR\_VECTOR, [57](#)
  - MATRIX\_NOT\_VECTOR, [57](#)
  - NOT\_BIPARTITE, [57](#)
  - NOT\_QUBIT\_GATE, [57](#)
  - NOT\_QUBIT\_SUBSYS, [57](#)
  - OUT\_OF\_RANGE, [57](#)
  - PERM\_INVALID, [57](#)
  - SUBSYS\_MISMATCH\_DIMS, [57](#)
  - TYPE\_MISMATCH, [57](#)
  - UNDEFINED\_TYPE, [57](#)
  - UNKNOWN\_EXCEPTION, [57](#)
  - ZERO\_SIZE, [57](#)
- rand
- [qpp, 36, 37](#)
- randint
- [qpp, 37](#)
- randket
- [qpp, 37](#)
- randkraus
- [qpp, 38](#)
- randn
- [qpp, 38, 39](#)
- randperm
- [qpp, 39](#)
- randrho
- [qpp, 39](#)
- rdevs
- [qpp, 48](#)
- renyi
- [qpp, 40](#)
- reshape
- [qpp, 41](#)
- SUBSYS\_MISMATCH\_DIMS
- [qpp::Exception, 57](#)
- save
- [qpp, 41](#)
- schmidtcoeff
- [qpp, 42](#)
- schmidtprob
- [qpp, 42](#)
- shannon
- [qpp, 44](#)
- sinm
- [qpp, 44](#)
- spectralpowm
- [qpp, 45](#)
- sqrtn
- [qpp, 45](#)
- st
- [qpp, 48](#)
- sum
- [qpp, 45](#)
- super
- [qpp, 46](#)
- syspermute
- [qpp, 46](#)
- TYPE\_MISMATCH
- [qpp::Exception, 57](#)
- trace
- [qpp, 47](#)
- transpose
- [qpp, 47](#)
- tsallis
- [qpp, 48](#)
- UNDEFINED\_TYPE
- [qpp::Exception, 57](#)
- UNKNOWN\_EXCEPTION
- [qpp::Exception, 57](#)
- ZERO\_SIZE
- [qpp::Exception, 57](#)