

qpp  
0.1

Generated by Doxygen 1.8.5

Tue May 20 2014 19:44:44



# 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 . . . . .	13
5.1.1.2	adjoint . . . . .	14
5.1.1.3	anticomm . . . . .	14
5.1.1.4	channel . . . . .	14
5.1.1.5	channel . . . . .	15
5.1.1.6	choi . . . . .	15
5.1.1.7	choi2kraus . . . . .	16
5.1.1.8	comm . . . . .	16
5.1.1.9	compperm . . . . .	17
5.1.1.10	conjugate . . . . .	17
5.1.1.11	cosm . . . . .	17
5.1.1.12	cwise . . . . .	18
5.1.1.13	det . . . . .	18
5.1.1.14	disp . . . . .	18
5.1.1.15	disp . . . . .	18
5.1.1.16	disp . . . . .	18
5.1.1.17	disp . . . . .	18
5.1.1.18	displn . . . . .	19

5.1.1.19	displn	19
5.1.1.20	displn	19
5.1.1.21	displn	20
5.1.1.22	entanglement	20
5.1.1.23	evals	21
5.1.1.24	evecs	21
5.1.1.25	expandout	22
5.1.1.26	expm	22
5.1.1.27	funm	22
5.1.1.28	gconcurrency	23
5.1.1.29	grams	24
5.1.1.30	grams	24
5.1.1.31	grams	24
5.1.1.32	hevals	25
5.1.1.33	hevecs	25
5.1.1.34	inverse	25
5.1.1.35	invperm	26
5.1.1.36	kron	26
5.1.1.37	kron	26
5.1.1.38	kron	26
5.1.1.39	kron	27
5.1.1.40	kronpow	27
5.1.1.41	load	27
5.1.1.42	loadMATLABmatrix	27
5.1.1.43	loadMATLABmatrix	27
5.1.1.44	loadMATLABmatrix	27
5.1.1.45	logdet	28
5.1.1.46	logm	28
5.1.1.47	mket	28
5.1.1.48	mket	29
5.1.1.49	mket	29
5.1.1.50	multiidx2n	29
5.1.1.51	n2multiidx	30
5.1.1.52	norm	30
5.1.1.53	powm	30
5.1.1.54	prj	31
5.1.1.55	ptrace	32
5.1.1.56	ptrace1	33
5.1.1.57	ptrace2	33
5.1.1.58	ptranspose	34

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

5.1.2.2	<a href="#">rdevs</a>	46
5.1.2.3	<a href="#">st</a>	46
5.2	<a href="#">qpp::ct Namespace Reference</a>	46
5.2.1	<a href="#">Function Documentation</a>	47
5.2.1.1	<a href="#">omega</a>	47
5.2.2	<a href="#">Variable Documentation</a>	47
5.2.2.1	<a href="#">chop</a>	47
5.2.2.2	<a href="#">ee</a>	47
5.2.2.3	<a href="#">eps</a>	47
5.2.2.4	<a href="#">ii</a>	47
5.2.2.5	<a href="#">maxn</a>	47
5.2.2.6	<a href="#">pi</a>	47
5.3	<a href="#">qpp::internal Namespace Reference</a>	47
5.3.1	<a href="#">Function Documentation</a>	48
5.3.1.1	<a href="#">_check_col_vector</a>	48
5.3.1.2	<a href="#">_check_dims</a>	48
5.3.1.3	<a href="#">_check_dims_match_cvect</a>	48
5.3.1.4	<a href="#">_check_dims_match_mat</a>	48
5.3.1.5	<a href="#">_check_dims_match_rvect</a>	48
5.3.1.6	<a href="#">_check_eq_dims</a>	48
5.3.1.7	<a href="#">_check_nonzero_size</a>	48
5.3.1.8	<a href="#">_check_perm</a>	48
5.3.1.9	<a href="#">_check_row_vector</a>	48
5.3.1.10	<a href="#">_check_square_mat</a>	48
5.3.1.11	<a href="#">_check_subsys_match_dims</a>	48
5.3.1.12	<a href="#">_check_vector</a>	48
5.3.1.13	<a href="#">_kron2</a>	48
5.3.1.14	<a href="#">_multiidx2n</a>	49
5.3.1.15	<a href="#">_n2multiidx</a>	49
5.3.1.16	<a href="#">variadic_vector_emplace</a>	49
5.3.1.17	<a href="#">variadic_vector_emplace</a>	49
5.4	<a href="#">qpp::types Namespace Reference</a>	49
5.4.1	<a href="#">Typedef Documentation</a>	49
5.4.1.1	<a href="#">bra</a>	49
5.4.1.2	<a href="#">cmat</a>	49
5.4.1.3	<a href="#">cplx</a>	49
5.4.1.4	<a href="#">dmat</a>	49
5.4.1.5	<a href="#">DynMat</a>	49
5.4.1.6	<a href="#">ket</a>	49

<b>6</b>	<b>Class Documentation</b>	<b>51</b>
6.1	qpp::DiscreteDistribution Class Reference	51
6.1.1	Constructor & Destructor Documentation	51
6.1.1.1	DiscreteDistribution	51
6.1.1.2	DiscreteDistribution	51
6.1.1.3	DiscreteDistribution	51
6.1.2	Member Function Documentation	51
6.1.2.1	probabilities	51
6.1.2.2	sample	52
6.1.3	Member Data Documentation	52
6.1.3.1	_d	52
6.2	qpp::DiscreteDistributionAbsSquare Class Reference	52
6.2.1	Constructor & Destructor Documentation	52
6.2.1.1	DiscreteDistributionAbsSquare	53
6.2.1.2	DiscreteDistributionAbsSquare	53
6.2.1.3	DiscreteDistributionAbsSquare	53
6.2.1.4	DiscreteDistributionAbsSquare	54
6.2.2	Member Function Documentation	54
6.2.2.1	cplx2weights	54
6.2.2.2	probabilities	54
6.2.2.3	sample	54
6.2.3	Member Data Documentation	54
6.2.3.1	_d	54
6.3	qpp::Exception Class Reference	54
6.3.1	Member Enumeration Documentation	56
6.3.1.1	Type	56
6.3.2	Constructor & Destructor Documentation	57
6.3.2.1	Exception	57
6.3.2.2	Exception	57
6.3.2.3	~Exception	57
6.3.3	Member Function Documentation	57
6.3.3.1	_construct_exception_msg	57
6.3.3.2	what	57
6.3.4	Member Data Documentation	57
6.3.4.1	_custom	57
6.3.4.2	_msg	57
6.3.4.3	_type	57
6.3.4.4	_where	57
6.4	qpp::Gates Class Reference	58
6.4.1	Constructor & Destructor Documentation	59

6.4.1.1	<a href="#">~Gates</a>	59
6.4.1.2	<a href="#">Gates</a>	59
6.4.2	<a href="#">Member Function Documentation</a>	59
6.4.2.1	<a href="#">apply</a>	60
6.4.2.2	<a href="#">CTRL</a>	61
6.4.2.3	<a href="#">Fd</a>	61
6.4.2.4	<a href="#">Id</a>	61
6.4.2.5	<a href="#">Rn</a>	61
6.4.2.6	<a href="#">Xd</a>	62
6.4.2.7	<a href="#">Zd</a>	62
6.4.3	<a href="#">Friends And Related Function Documentation</a>	62
6.4.3.1	<a href="#">Singleton&lt; const Gates &gt;</a>	62
6.4.4	<a href="#">Member Data Documentation</a>	62
6.4.4.1	<a href="#">CNOTab</a>	62
6.4.4.2	<a href="#">CNOTba</a>	62
6.4.4.3	<a href="#">CZ</a>	62
6.4.4.4	<a href="#">FRED</a>	62
6.4.4.5	<a href="#">H</a>	62
6.4.4.6	<a href="#">Id2</a>	62
6.4.4.7	<a href="#">S</a>	62
6.4.4.8	<a href="#">SWAP</a>	62
6.4.4.9	<a href="#">T</a>	62
6.4.4.10	<a href="#">TOF</a>	63
6.4.4.11	<a href="#">X</a>	63
6.4.4.12	<a href="#">Y</a>	63
6.4.4.13	<a href="#">Z</a>	63
6.5	<a href="#">qpp::NormalDistribution Class Reference</a>	63
6.5.1	<a href="#">Constructor &amp; Destructor Documentation</a>	63
6.5.1.1	<a href="#">NormalDistribution</a>	63
6.5.2	<a href="#">Member Function Documentation</a>	63
6.5.2.1	<a href="#">sample</a>	63
6.5.3	<a href="#">Member Data Documentation</a>	63
6.5.3.1	<a href="#">_d</a>	63
6.6	<a href="#">qpp::Qudit Class Reference</a>	64
6.6.1	<a href="#">Constructor &amp; Destructor Documentation</a>	64
6.6.1.1	<a href="#">Qudit</a>	64
6.6.1.2	<a href="#">~Qudit</a>	64
6.6.2	<a href="#">Member Function Documentation</a>	64
6.6.2.1	<a href="#">getD</a>	64
6.6.2.2	<a href="#">getRho</a>	64



6.6.2.3	<a href="#">measure</a>	65
6.6.2.4	<a href="#">measure</a>	65
6.6.3	<a href="#">Member Data Documentation</a>	65
6.6.3.1	<a href="#">_D</a>	65
6.6.3.2	<a href="#">_rho</a>	65
6.7	<a href="#">qpp::RandomDevices Class Reference</a>	66
6.7.1	<a href="#">Constructor &amp; Destructor Documentation</a>	67
6.7.1.1	<a href="#">~RandomDevices</a>	67
6.7.1.2	<a href="#">RandomDevices</a>	67
6.7.2	<a href="#">Friends And Related Function Documentation</a>	67
6.7.2.1	<a href="#">Singleton&lt; RandomDevices &gt;</a>	67
6.7.3	<a href="#">Member Data Documentation</a>	67
6.7.3.1	<a href="#">_rd</a>	67
6.7.3.2	<a href="#">_rng</a>	67
6.8	<a href="#">qpp::Singleton&lt; T &gt; Class Template Reference</a>	67
6.8.1	<a href="#">Constructor &amp; Destructor Documentation</a>	68
6.8.1.1	<a href="#">Singleton</a>	68
6.8.1.2	<a href="#">Singleton</a>	68
6.8.2	<a href="#">Member Function Documentation</a>	68
6.8.2.1	<a href="#">getInstance</a>	68
6.8.2.2	<a href="#">operator=</a>	68
6.9	<a href="#">qpp::States Class Reference</a>	68
6.9.1	<a href="#">Constructor &amp; Destructor Documentation</a>	69
6.9.1.1	<a href="#">~States</a>	69
6.9.1.2	<a href="#">States</a>	69
6.9.2	<a href="#">Friends And Related Function Documentation</a>	69
6.9.2.1	<a href="#">Singleton&lt; const States &gt;</a>	69
6.9.3	<a href="#">Member Data Documentation</a>	69
6.9.3.1	<a href="#">b00</a>	69
6.9.3.2	<a href="#">b01</a>	69
6.9.3.3	<a href="#">b10</a>	70
6.9.3.4	<a href="#">b11</a>	70
6.9.3.5	<a href="#">GHZ</a>	70
6.9.3.6	<a href="#">pb00</a>	70
6.9.3.7	<a href="#">pb01</a>	70
6.9.3.8	<a href="#">pb10</a>	70
6.9.3.9	<a href="#">pb11</a>	70
6.9.3.10	<a href="#">pGHZ</a>	70
6.9.3.11	<a href="#">pW</a>	70
6.9.3.12	<a href="#">px0</a>	70

6.9.3.13	px1	70
6.9.3.14	py0	70
6.9.3.15	py1	70
6.9.3.16	pz0	70
6.9.3.17	pz1	70
6.9.3.18	W	70
6.9.3.19	x0	70
6.9.3.20	x1	70
6.9.3.21	y0	70
6.9.3.22	y1	70
6.9.3.23	z0	70
6.9.3.24	z1	70
6.10	qpp::Timer Class Reference	70
6.10.1	Constructor & Destructor Documentation	71
6.10.1.1	Timer	71
6.10.1.2	~Timer	71
6.10.2	Member Function Documentation	71
6.10.2.1	seconds	71
6.10.2.2	tic	71
6.10.2.3	toc	71
6.10.3	Friends And Related Function Documentation	71
6.10.3.1	operator<<	71
6.10.4	Member Data Documentation	71
6.10.4.1	_end	71
6.10.4.2	_start	71
6.11	qpp::UniformRealDistribution Class Reference	71
6.11.1	Constructor & Destructor Documentation	72
6.11.1.1	UniformRealDistribution	72
6.11.2	Member Function Documentation	72
6.11.2.1	sample	72
6.11.3	Member Data Documentation	72
6.11.3.1	_d	72
<b>7</b>	<b>File Documentation</b>	<b>73</b>
7.1	include/channels.h File Reference	73
7.2	include/classes/exception.h File Reference	74
7.3	include/classes/gates.h File Reference	74
7.4	include/classes/qudit.h File Reference	75
7.5	include/classes/randevs.h File Reference	75
7.6	include/classes/singleton.h File Reference	76

7.7	<a href="#">include/classes/stat.h File Reference</a>	76
7.8	<a href="#">include/classes/states.h File Reference</a>	77
7.9	<a href="#">include/classes/timer.h File Reference</a>	77
7.10	<a href="#">include/constants.h File Reference</a>	78
7.11	<a href="#">include/entanglement.h File Reference</a>	79
7.12	<a href="#">include/entropies.h File Reference</a>	80
7.13	<a href="#">include/functions.h File Reference</a>	80
7.14	<a href="#">include/internal.h File Reference</a>	83
7.15	<a href="#">include/io.h File Reference</a>	84
7.16	<a href="#">include/matlab.h File Reference</a>	85
7.17	<a href="#">include/qpp.h File Reference</a>	86
7.18	<a href="#">include/random.h File Reference</a>	87
7.19	<a href="#">include/types.h File Reference</a>	88



# 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>	46
<a href="#">qpp::internal</a>	47
<a href="#">qpp::types</a>	49



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

qpp::DiscreteDistribution . . . . .	51
qpp::DiscreteDistributionAbsSquare . . . . .	52
exception	
qpp::Exception . . . . .	54
qpp::NormalDistribution . . . . .	63
qpp::Qudit . . . . .	64
qpp::Singleton< T > . . . . .	67
qpp::Gates . . . . .	58
qpp::RandomDevices . . . . .	66
qpp::Singleton< const Gates > . . . . .	67
qpp::Singleton< const States > . . . . .	67
qpp::States . . . . .	68
qpp::Singleton< RandomDevices > . . . . .	67
qpp::Timer . . . . .	70
qpp::UniformRealDistribution . . . . .	71





## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">qpp::DiscreteDistribution</a>	51
<a href="#">qpp::DiscreteDistributionAbsSquare</a>	52
<a href="#">qpp::Exception</a>	54
<a href="#">qpp::Gates</a>	58
<a href="#">qpp::NormalDistribution</a>	63
<a href="#">qpp::Qudit</a>	64
<a href="#">qpp::RandomDevices</a>	66
<a href="#">qpp::Singleton&lt; T &gt;</a>	67
<a href="#">qpp::States</a>	68
<a href="#">qpp::Timer</a>	70
<a href="#">qpp::UniformRealDistribution</a>	71



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

include/ <a href="#">channels.h</a>	73
include/ <a href="#">constants.h</a>	78
include/ <a href="#">entanglement.h</a>	79
include/ <a href="#">entropies.h</a>	80
include/ <a href="#">functions.h</a>	80
include/ <a href="#">internal.h</a>	83
include/ <a href="#">io.h</a>	84
include/ <a href="#">matlab.h</a>	85
include/ <a href="#">qpp.h</a>	86
include/ <a href="#">random.h</a>	87
include/ <a href="#">types.h</a>	88
include/classes/ <a href="#">exception.h</a>	74
include/classes/ <a href="#">gates.h</a>	74
include/classes/ <a href="#">qudit.h</a>	75
include/classes/ <a href="#">randevs.h</a>	75
include/classes/ <a href="#">singleton.h</a>	76
include/classes/ <a href="#">stat.h</a>	76
include/classes/ <a href="#">states.h</a>	77
include/classes/ <a href="#">timer.h</a>	77



## Chapter 5

# Namespace Documentation

### 5.1 qpp Namespace Reference

#### Namespaces

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

#### Classes

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

#### Functions

- [types::cmat super](#) (const std::vector< [types::cmat](#) > &Ks)
- [types::cmat choi](#) (const std::vector< [types::cmat](#) > &Ks)
- std::vector< [types::cmat](#) > [choi2kraus](#) (const [types::cmat](#) &A)
- template<typename Derived >  
[types::cmat channel](#) (const Eigen::MatrixBase< Derived > &rho, const std::vector< [types::cmat](#) > &Ks)
- template<typename Derived >  
[types::cmat channel](#) (const Eigen::MatrixBase< Derived > &rho, const std::vector< [types::cmat](#) > &Ks,  
const std::vector< size\_t > &subsys, const std::vector< size\_t > &dims)
- template<typename Derived >  
[types::cmat schmidtcoeff](#) (const Eigen::MatrixBase< Derived > &A, const std::vector< size\_t > &dims)
- template<typename Derived >  
[types::cmat schmidtU](#) (const Eigen::MatrixBase< Derived > &A, const std::vector< size\_t > &dims)
- template<typename Derived >  
[types::cmat schmidtV](#) (const Eigen::MatrixBase< Derived > &A, const std::vector< size\_t > &dims)

- `template<typename Derived >`  
`types::cmat schmidtprob` (const Eigen::MatrixBase< Derived > &A, const std::vector< size\_t > &dims)
- `template<typename Derived >`  
`double entanglement` (const Eigen::MatrixBase< Derived > &A, const std::vector< size\_t > &dims)
- `template<typename Derived >`  
`double gconcurrency` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`double shannon` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`double renyi` (const double alpha, const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`double renyi_inf` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`double tsallis` (const double alpha, const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`double qmutualinfo` (const Eigen::MatrixBase< Derived > &A, const std::vector< size\_t > &subsys, const std::vector< size\_t > &dims)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > transpose` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > conjugate` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > adjoint` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > inverse` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`Derived::Scalar trace` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`Derived::Scalar det` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`Derived::Scalar logdet` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`Derived::Scalar sum` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`double norm` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::cmat evals` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::cmat evecs` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::dmat hevals` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::cmat hevecs` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::cmat funm` (const Eigen::MatrixBase< Derived > &A, `types::cplx`(\*f)(const `types::cplx` &))
- `template<typename Derived >`  
`types::cmat sqrtm` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::cmat absm` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::cmat expm` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::cmat logm` (const Eigen::MatrixBase< Derived > &A)

- `template<typename Derived >`  
`types::cmat sinm` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::cmat cosm` (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
`types::cmat spectralpowm` (const Eigen::MatrixBase< Derived > &A, const `types::cplx` z)
- `template<typename Derived >`  
`types::DynMat< typename`  
Derived::Scalar > `powm` (const Eigen::MatrixBase< Derived > &A, size\_t n)
- `template<typename OutputScalar, typename Derived >`  
`types::DynMat< OutputScalar > cwise` (const Eigen::MatrixBase< Derived > &A, OutputScalar(\*) (const  
typename Derived::Scalar &))
- `template<typename T >`  
`types::DynMat< typename T::Scalar > kron` (const T &head)
- `template<typename T, typename... Args>`  
`types::DynMat< typename T::Scalar > kron` (const T &head, const Args &...tail)
- `template<typename Derived >`  
`types::DynMat< typename`  
Derived::Scalar > `kron` (const std::vector< Derived > &As)
- `template<typename Derived >`  
`types::DynMat< typename`  
Derived::Scalar > `kron` (const std::initializer\_list< Derived > &As)
- `template<typename Derived >`  
`types::DynMat< typename`  
Derived::Scalar > `kronpow` (const Eigen::MatrixBase< Derived > &A, size\_t n)
- `template<typename Derived >`  
`types::DynMat< typename`  
Derived::Scalar > `reshape` (const Eigen::MatrixBase< Derived > &A, size\_t rows, size\_t cols)
- `template<typename Derived >`  
`types::DynMat< typename`  
Derived::Scalar > `syspermute` (const Eigen::MatrixBase< Derived > &A, const std::vector< size\_t > &perm,  
const std::vector< size\_t > &dims)
- `template<typename Derived >`  
`types::DynMat< typename`  
Derived::Scalar > `ptrace1` (const Eigen::MatrixBase< Derived > &A, const std::vector< size\_t > &dims)
- `template<typename Derived >`  
`types::DynMat< typename`  
Derived::Scalar > `ptrace2` (const Eigen::MatrixBase< Derived > &A, const std::vector< size\_t > &dims)
- `template<typename Derived >`  
`types::DynMat< typename`  
Derived::Scalar > `ptrace` (const Eigen::MatrixBase< Derived > &A, const std::vector< size\_t > &subsyst,  
const std::vector< size\_t > &dims)
- `template<typename Derived >`  
`types::DynMat< typename`  
Derived::Scalar > `ptranspose` (const Eigen::MatrixBase< Derived > &A, const std::vector< size\_t > &sub-  
sys, const std::vector< size\_t > &dims)
- `template<typename Derived1, typename Derived2 >`  
`types::DynMat< typename`  
Derived1::Scalar > `comm` (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase< Derived2  
> &B)
- `template<typename Derived1, typename Derived2 >`  
`types::DynMat< typename`  
Derived1::Scalar > `anticomm` (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase< De-  
rived2 > &B)
- `template<typename Derived >`  
`types::DynMat< typename`  
Derived::Scalar > `prj` (const Eigen::MatrixBase< Derived > &V)

- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > expandout` (const Eigen::MatrixBase< Derived > &A, size\_t pos, const std::vector< size\_t > &dims)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > grams` (const std::vector< Derived > &Vs)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > grams` (const std::initializer\_list< Derived > &Vs)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > grams` (const Eigen::MatrixBase< Derived > &A)
- `std::vector< size_t > n2multiidx` (size\_t n, const std::vector< size\_t > &dims)
- `size_t multiidx2n` (const std::vector< size\_t > &midx, const std::vector< size\_t > &dims)
- `types::ket mket` (const std::vector< size\_t > &mask)
- `types::ket mket` (const std::vector< size\_t > &mask, const std::vector< size\_t > &dims)
- `types::ket mket` (const std::vector< size\_t > &mask, size\_t d)
- `std::vector< size_t > invperm` (const std::vector< size\_t > &perm)
- `std::vector< size_t > compperm` (const std::vector< size\_t > &perm, const std::vector< size\_t > &sigma)
- `template<typename T >`  
`void disp` (const T &x, const std::string &separator, const std::string &start="[", const std::string &end="]", std::ostream &os=std::cout)
- `template<typename T >`  
`void displn` (const T &x, const std::string &separator, const std::string &start="[", const std::string &end="]", std::ostream &os=std::cout)
- `template<typename T >`  
`void disp` (const T \*x, const size\_t n, const std::string &separator, const std::string &start="[", const std::string &end="]", std::ostream &os=std::cout)
- `template<typename T >`  
`void displn` (const T \*x, const size\_t n, const std::string &separator, const std::string &start="[", const std::string &end="]", std::ostream &os=std::cout)
- `template<typename Derived >`  
`void disp` (const Eigen::MatrixBase< Derived > &A, double chop=ct::chop, std::ostream &os=std::cout)
- `template<typename Derived >`  
`void displn` (const Eigen::MatrixBase< Derived > &A, double chop=ct::chop, std::ostream &os=std::cout)
- `void disp` (const types::cplx c, double chop=ct::chop, std::ostream &os=std::cout)
- `void displn` (const types::cplx c, double chop=ct::chop, std::ostream &os=std::cout)
- `template<typename Derived >`  
`void save` (const Eigen::MatrixBase< Derived > &A, const std::string &fname)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > load` (const std::string &fname)
- `template<typename Derived >`  
`Derived loadMATLABmatrix` (const std::string &mat\_file, const std::string &var\_name)
- `template<>`  
`types::dmat loadMATLABmatrix` (const std::string &mat\_file, const std::string &var\_name)
- `template<>`  
`types::cmat loadMATLABmatrix` (const std::string &mat\_file, const std::string &var\_name)
- `template<typename Derived >`  
`void saveMATLABmatrix` (const Eigen::MatrixBase< Derived > &A, const std::string &mat\_file, const std::string &var\_name, const std::string &mode)
- `template<>`  
`void saveMATLABmatrix` (const Eigen::MatrixBase< typename types::dmat > &A, const std::string &mat\_file, const std::string &var\_name, const std::string &mode)



- `template<>`  
void `saveMATLABmatrix` (const Eigen::MatrixBase< typename `types::cmat` > &A, const std::string &mat\_file, const std::string &var\_name, const std::string &mode)
- `template<typename Derived >`  
Derived `rand` (size\_t rows, size\_t cols, double a=0, double b=1)
- `template<>`  
`types::dmat rand` (size\_t rows, size\_t cols, double a, double b)
- `template<>`  
`types::cmat rand` (size\_t rows, size\_t cols, double a, double b)
- double `rand` (double a=0, double b=1)
- long long `randint` (long long a, long long b)
- `template<typename Derived >`  
Derived `randn` (size\_t rows, size\_t cols, double mean=0, double sigma=1)
- `template<>`  
`types::dmat randn` (size\_t rows, size\_t cols, double mean, double sigma)
- `template<>`  
`types::cmat randn` (size\_t rows, size\_t cols, double mean, double sigma)
- double `randn` (double mean=0, double sigma=1)
- `types::cmat randU` (size\_t D)
- `types::cmat randV` (size\_t Din, size\_t Dout)
- `std::vector< types::cmat > randkraus` (size\_t n, size\_t D)
- `types::cmat randH` (size\_t D)
- `types::ket randket` (size\_t D)
- `types::cmat randrho` (size\_t D)
- `std::vector< size_t > randperm` (size\_t n)

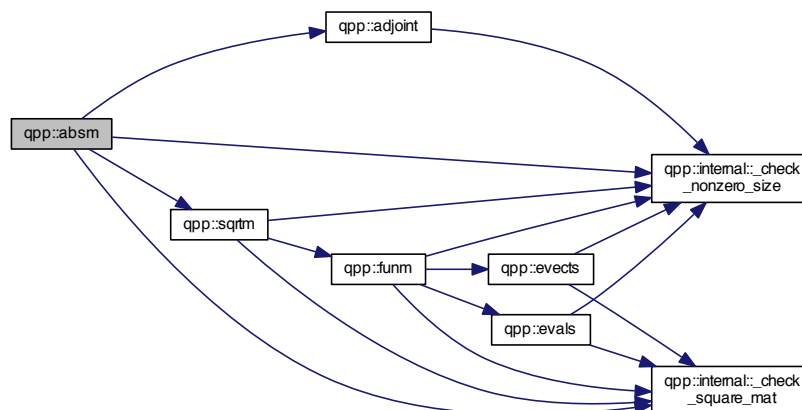
## Variables

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

### 5.1.1 Function Documentation

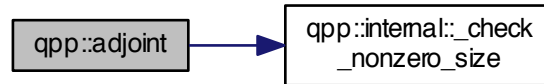
#### 5.1.1.1 `template<typename Derived > types::cmat qpp::absm ( const Eigen::MatrixBase< Derived > &A )`

Here is the call graph for this function:



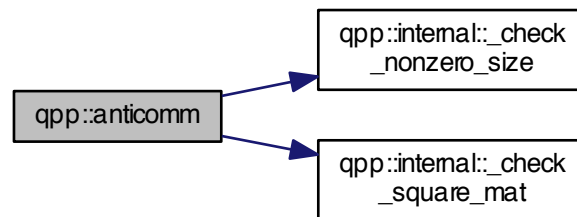
5.1.1.2 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::adjoint ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



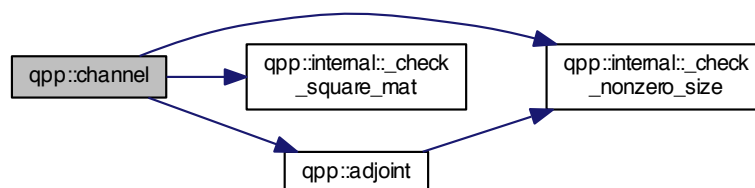
5.1.1.3 `template<typename Derived1 , typename Derived2 > types::DynMat<typename Derived1::Scalar> qpp::anticomm ( const Eigen::MatrixBase< Derived1 > & A, const Eigen::MatrixBase< Derived2 > & B )`

Here is the call graph for this function:



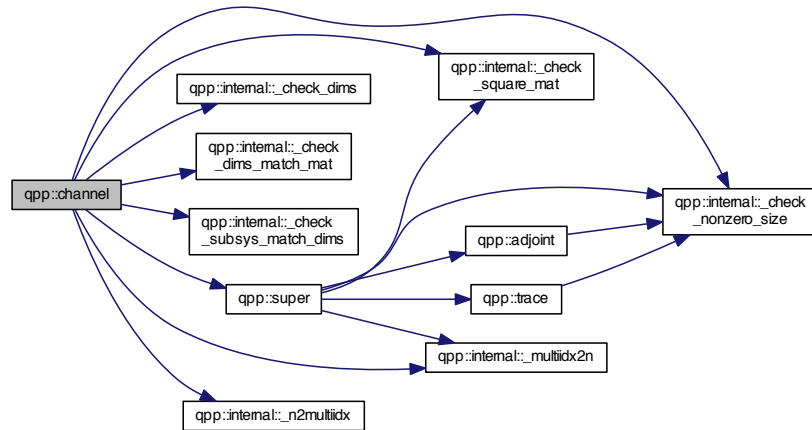
5.1.1.4 `template<typename Derived > types::cmat qpp::channel ( const Eigen::MatrixBase< Derived > & rho, const std::vector< types::cmat > & Ks )`

Here is the call graph for this function:



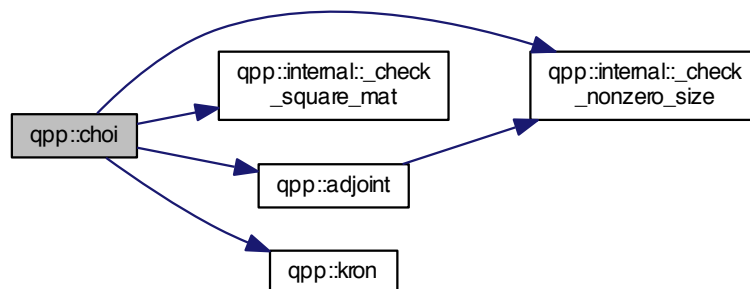
5.1.1.5 `template<typename Derived > types::cmat qpp::channel ( const Eigen::MatrixBase< Derived > & rho, const std::vector< types::cmat > & Ks, const std::vector< size_t > & subsys, const std::vector< size_t > & dims )`

Here is the call graph for this function:



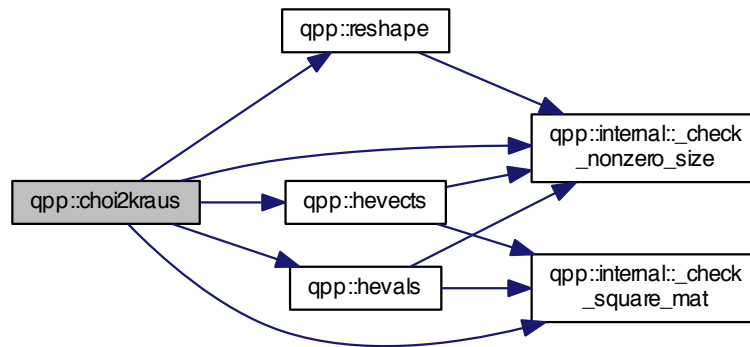
5.1.1.6 `types::cmat qpp::choi ( const std::vector< types::cmat > & Ks )`

Here is the call graph for this function:



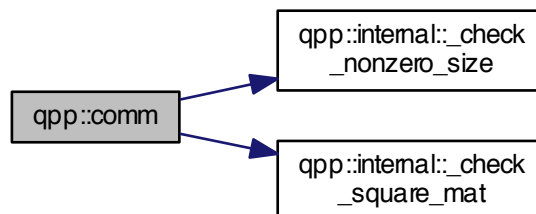
#### 5.1.1.7 `std::vector<types::cmat> qpp::choi2kraus ( const types::cmat & A )`

Here is the call graph for this function:



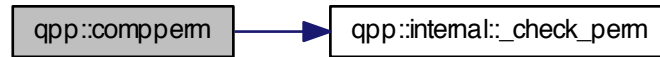
#### 5.1.1.8 `template<typename Derived1 , typename Derived2 > types::DynMat<typename Derived1::Scalar> qpp::comm ( const Eigen::MatrixBase< Derived1 > & A, const Eigen::MatrixBase< Derived2 > & B )`

Here is the call graph for this function:



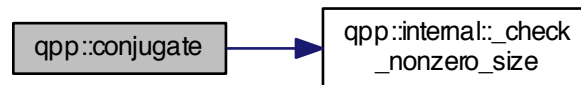
5.1.1.9 `std::vector<size_t> qpp::compperm ( const std::vector< size_t > & perm, const std::vector< size_t > & sigma )`

Here is the call graph for this function:



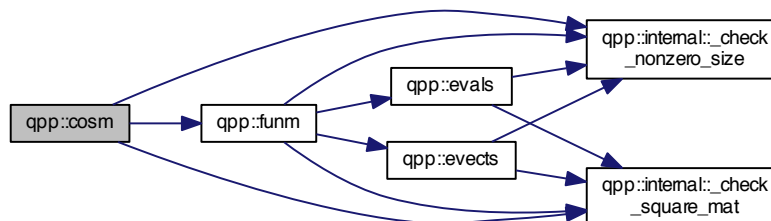
5.1.1.10 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::conjugate ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



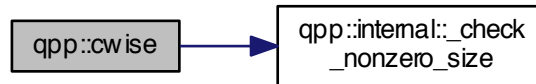
5.1.1.11 `template<typename Derived > types::cmat qpp::cosm ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



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

Here is the call graph for this function:



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

Here is the call graph for this function:



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

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

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

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

Here is the call graph for this function:



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

Here is the call graph for this function:



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

Here is the call graph for this function:



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

Here is the call graph for this function:



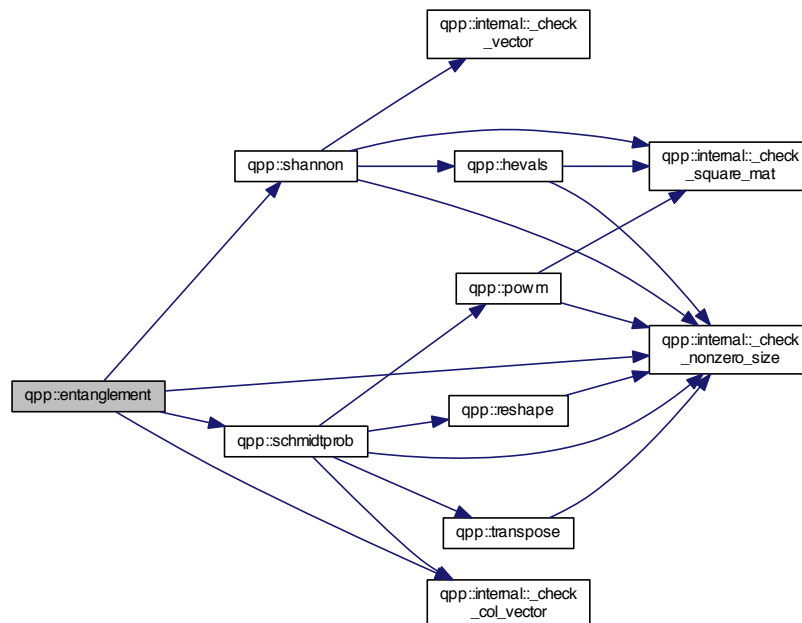
5.1.1.21 `void qpp::displn ( const types::cplx c, double chop = ct : : chop, std::ostream & os = std : : cout )`

Here is the call graph for this function:



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

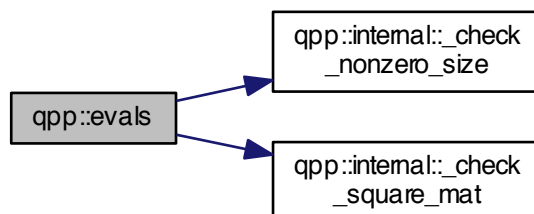
Here is the call graph for this function:





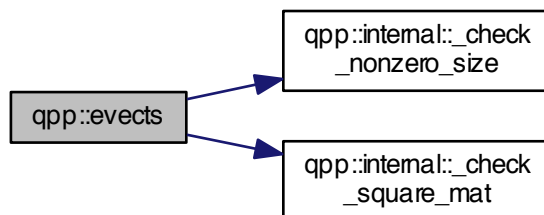
5.1.1.23 `template<typename Derived> types::cmat qpp::evals ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



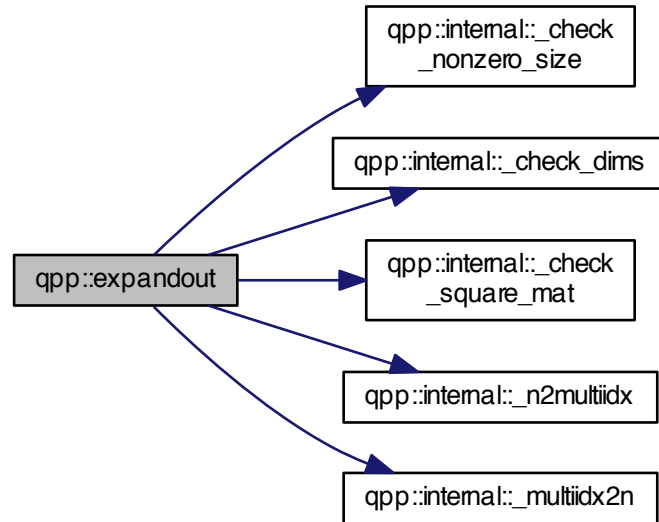
5.1.1.24 `template<typename Derived> types::cmat qpp::evecs ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



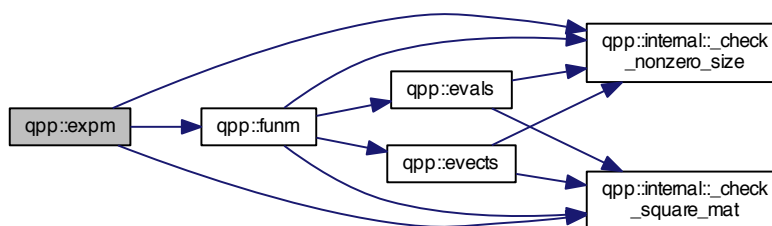
5.1.1.25 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::expandout ( const Eigen::MatrixBase< Derived> & A, size_t pos, const std::vector< size_t> & dims )`

Here is the call graph for this function:



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

Here is the call graph for this function:



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

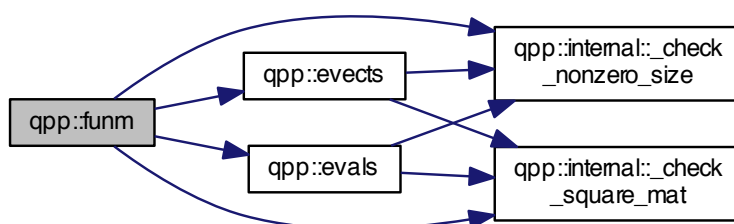
## Parameters

$A$	input matrix
$f$	function pointer

## Returns

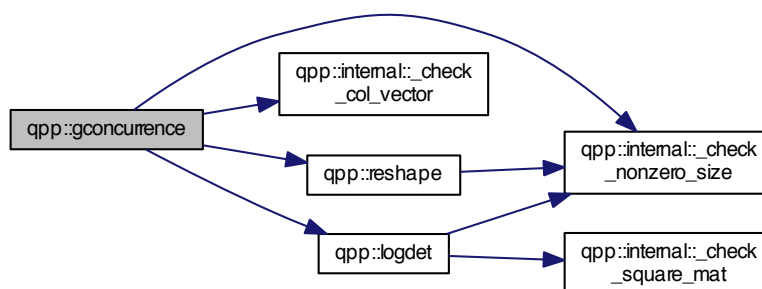
[types::cmat](#)

Here is the call graph for this function:



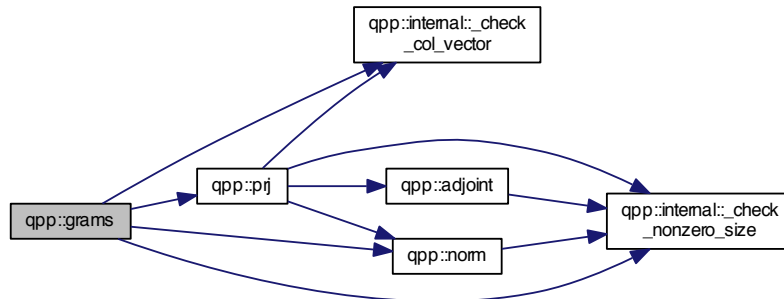
5.1.1.28 `template<typename Derived> double qpp::gconcurrency ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



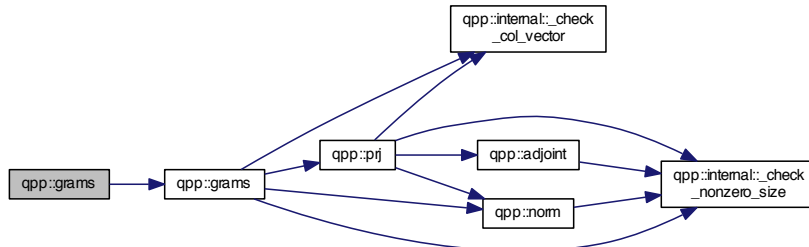
5.1.1.29 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::grams ( const std::vector< Derived > & Vs )`

Here is the call graph for this function:



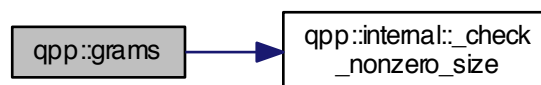
5.1.1.30 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::grams ( const std::initializer_list< Derived > & Vs )`

Here is the call graph for this function:



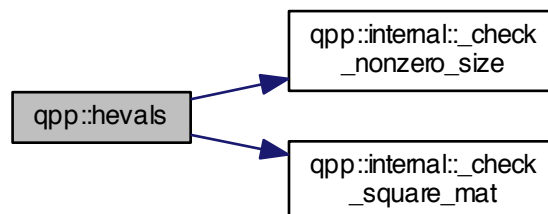
5.1.1.31 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::grams ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



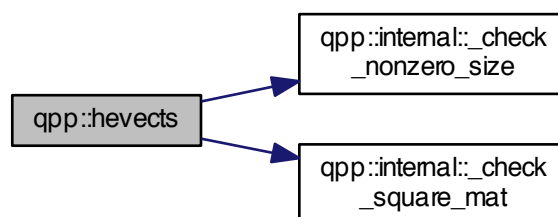
5.1.1.32 `template<typename Derived> types::dmat qpp::hevals ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



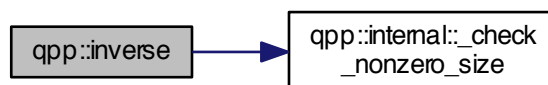
5.1.1.33 `template<typename Derived> types::cmat qpp::hevects ( const Eigen::MatrixBase< Derived > & A )`

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

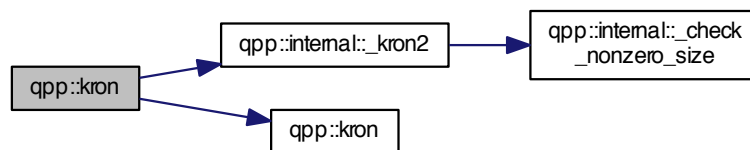
Here is the call graph for this function:



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:



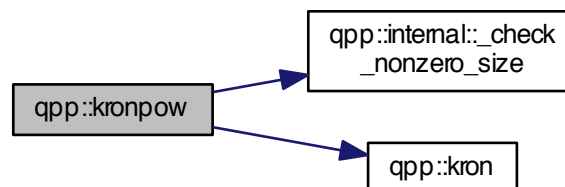
5.1.1.39 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::kron ( const std::initializer_list<Derived> & As )`

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, 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 )`

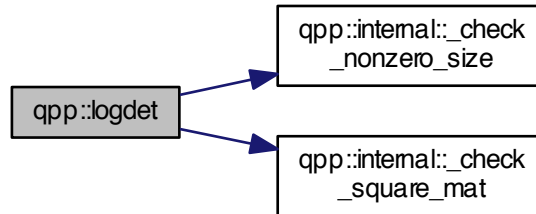
5.1.1.42 `template<typename Derived> Derived qpp::loadMATLABmatrix ( const std::string & mat_file, const std::string & var_name )`

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 )`

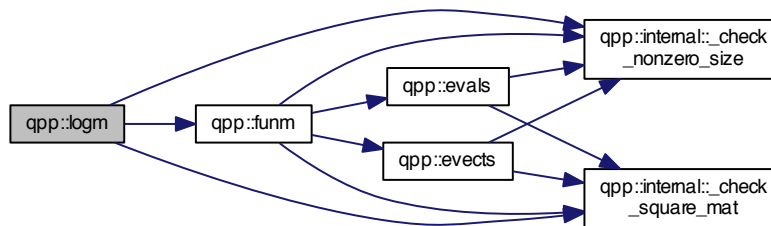
5.1.1.45 `template<typename Derived> Derived::Scalar qpp::logdet ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



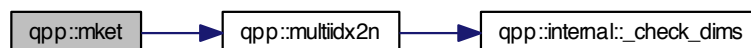
5.1.1.46 `template<typename Derived> types::cmat qpp::logm ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



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

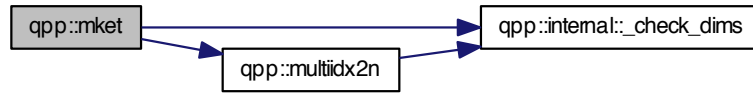
Here is the call graph for this function:





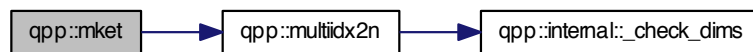
5.1.1.48 `types::ket qpp::mket ( const std::vector< size_t > & mask, const std::vector< size_t > & dims )`

Here is the call graph for this function:



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

Here is the call graph for this function:



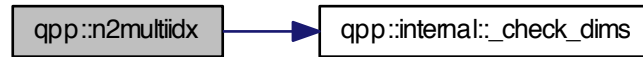
5.1.1.50 `size_t qpp::multidx2n ( const std::vector< size_t > & midx, const std::vector< size_t > & dims )`

Here is the call graph for this function:



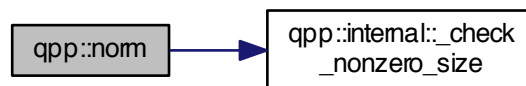
5.1.1.51 `std::vector<size_t> qpp::n2multiidx ( size_t n, const std::vector< size_t > & dims )`

Here is the call graph for this function:



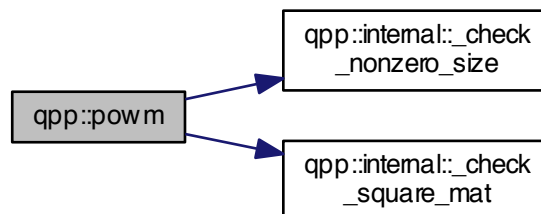
5.1.1.52 `template<typename Derived > double qpp::norm ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



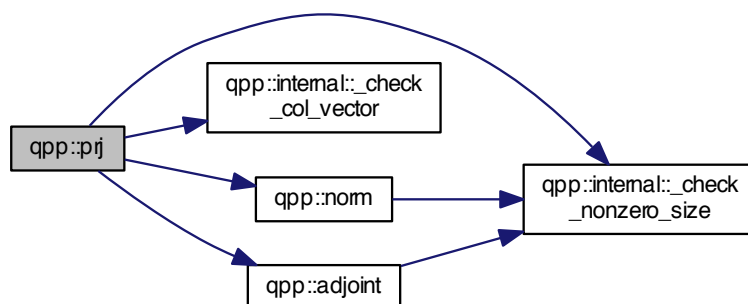
5.1.1.53 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::powm ( const Eigen::MatrixBase< Derived > & A, size_t n )`

Here is the call graph for this function:



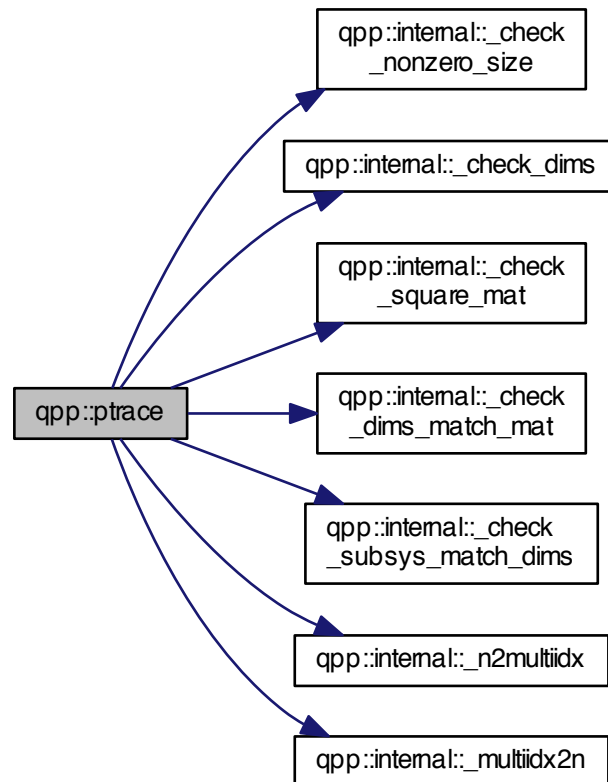
5.1.1.54 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::prj ( const Eigen::MatrixBase<Derived> & V )`

Here is the call graph for this function:



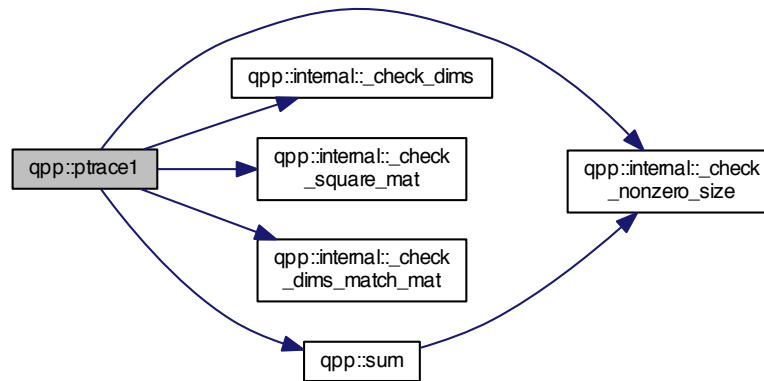
5.1.1.55 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::ptrace ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & subsys, const std::vector< size_t> & dims )`

Here is the call graph for this function:



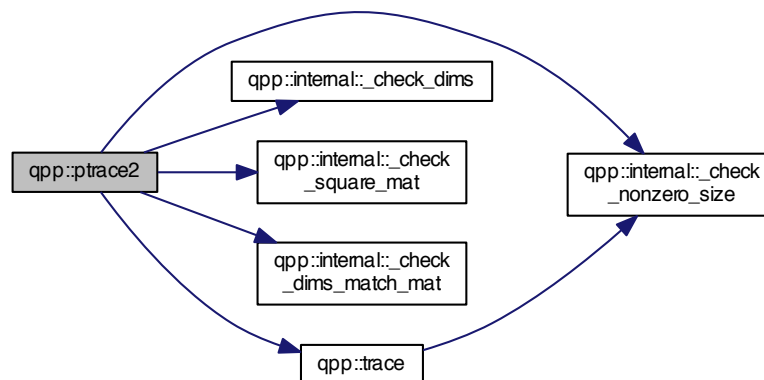
5.1.1.56 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::ptrace1 ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims )`

Here is the call graph for this function:



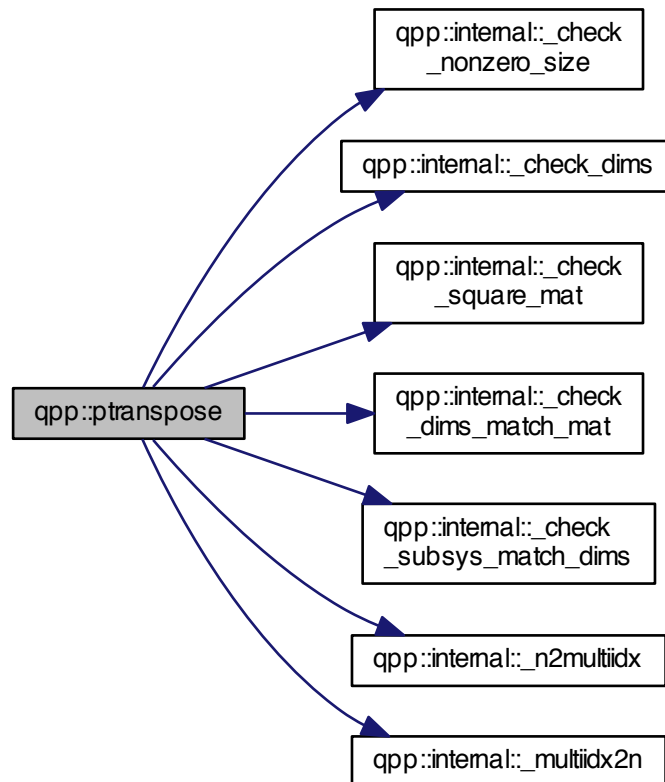
5.1.1.57 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::ptrace2 ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims )`

Here is the call graph for this function:



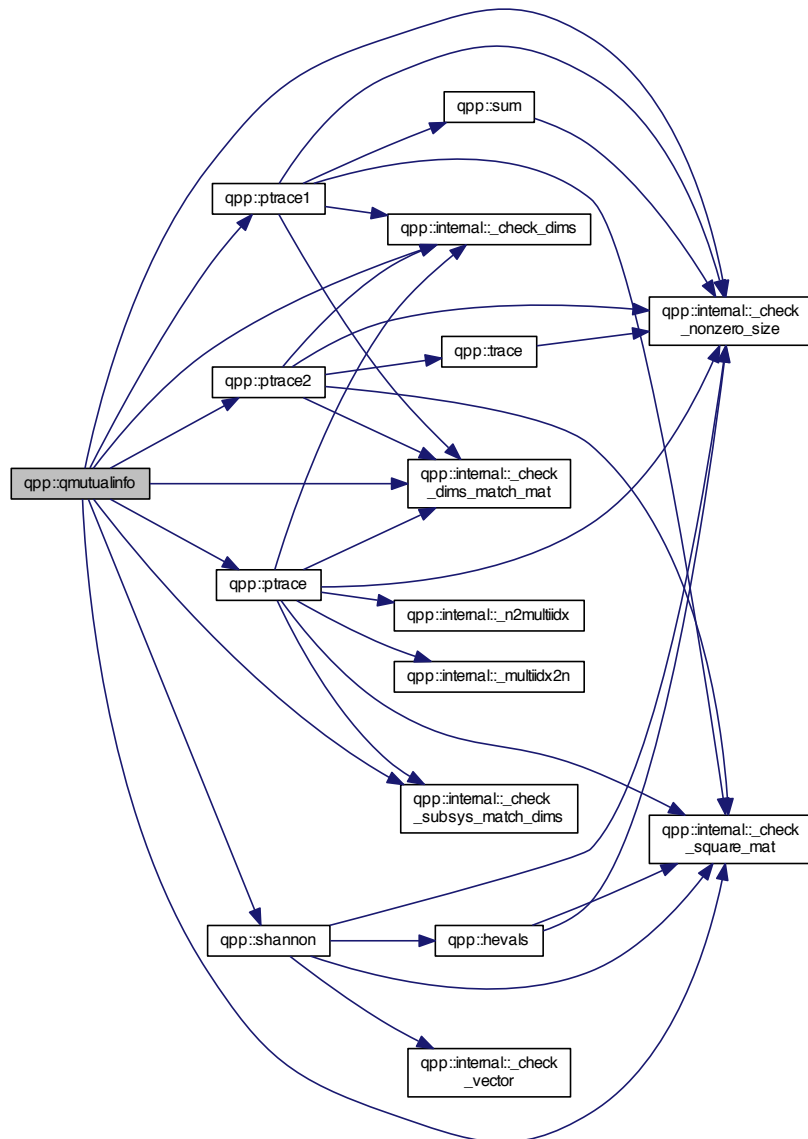
5.1.1.58 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::ptranspose ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & subsys, const std::vector< size_t> & dims )`

Here is the call graph for this function:



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

Here is the call graph for this function:



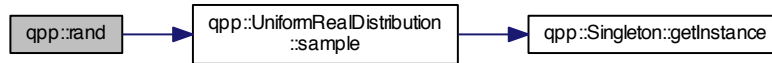
5.1.1.60 `template<typename Derived> Derived qpp::rand ( size_t rows, size_t cols, double a = 0, double b = 1 )`

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

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

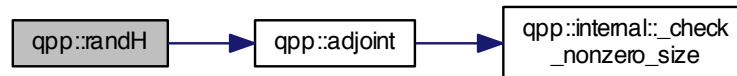
#### 5.1.1.63 `double qpp::rand ( double a = 0, double b = 1 )`

Here is the call graph for this function:



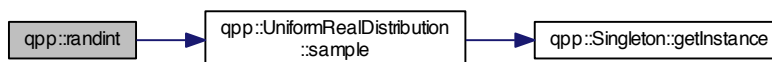
#### 5.1.1.64 `types::cmat qpp::randH ( size_t D )`

Here is the call graph for this function:



#### 5.1.1.65 `long long qpp::randint ( long long a, long long b )`

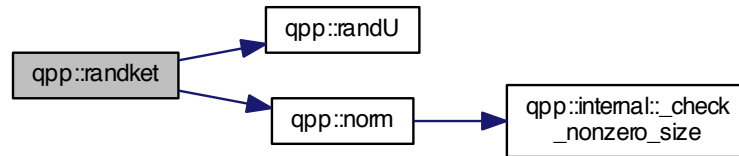
Here is the call graph for this function:



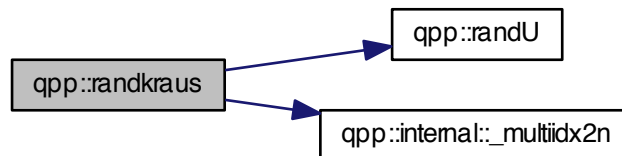


5.1.1.66 `types::ket qpp::randket ( size_t D )`

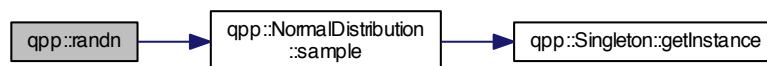
Here is the call graph for this function:

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

Here is the call graph for this function:

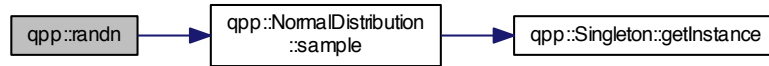
5.1.1.68 `template<typename Derived > Derived qpp::randn ( size_t rows, size_t cols, double mean = 0, double sigma = 1 )`5.1.1.69 `template<> types::dmatrix qpp::randn ( size_t rows, size_t cols, double mean, double sigma )`

Here is the call graph for this function:

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

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

Here is the call graph for this function:



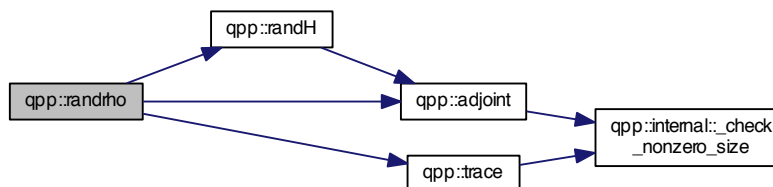
#### 5.1.1.72 `std::vector<size_t> qpp::randperm ( size_t n )`

Here is the call graph for this function:



#### 5.1.1.73 `types::cmat qpp::randrho ( size_t D )`

Here is the call graph for this function:



#### 5.1.1.74 `types::cmat qpp::randU ( size_t D )`

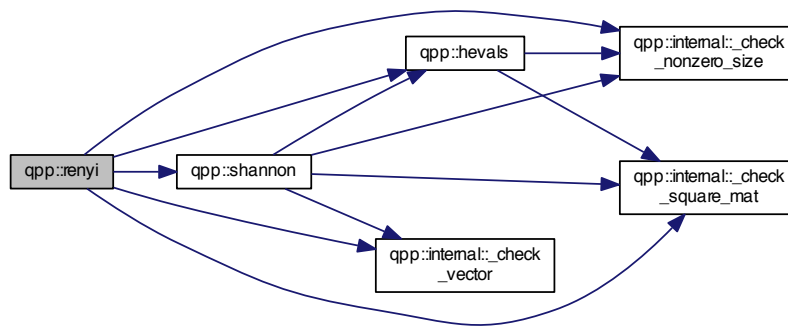
#### 5.1.1.75 `types::cmat qpp::randV ( size_t Din, size_t Dout )`

Here is the call graph for this function:



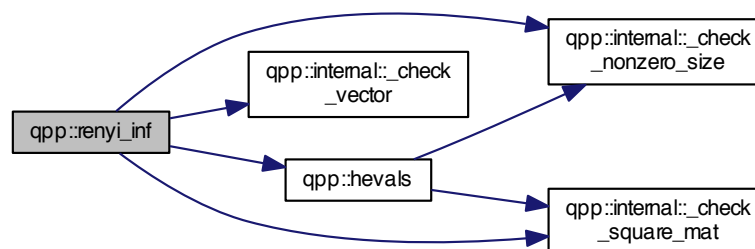
#### 5.1.1.76 `template<typename Derived> double qpp::renyi ( const double alpha, const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



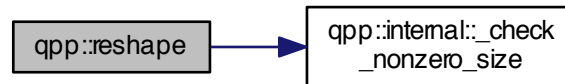
#### 5.1.1.77 `template<typename Derived> double qpp::renyi_inf ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



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

Here is the call graph for this function:

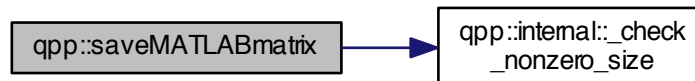


5.1.1.79 `template<typename Derived> void qpp::save ( const Eigen::MatrixBase< Derived> & A, const std::string & fname )`

5.1.1.80 `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 )`

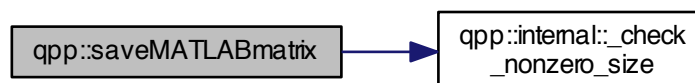
5.1.1.81 `template<> void qpp::saveMATLABmatrix ( const Eigen::MatrixBase< typename types::dmat> & A, const std::string & mat_file, const std::string & var_name, const std::string & mode )`

Here is the call graph for this function:



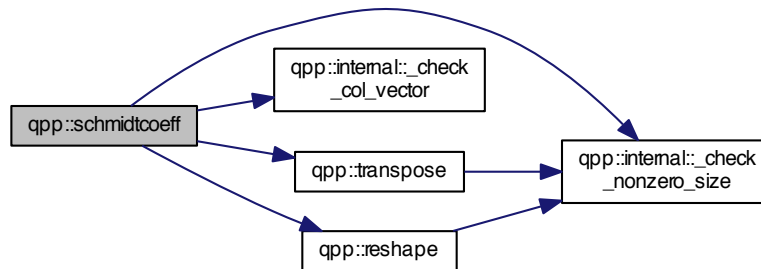
5.1.1.82 `template<> void qpp::saveMATLABmatrix ( const Eigen::MatrixBase< typename types::cmat> & A, const std::string & mat_file, const std::string & var_name, const std::string & mode )`

Here is the call graph for this function:



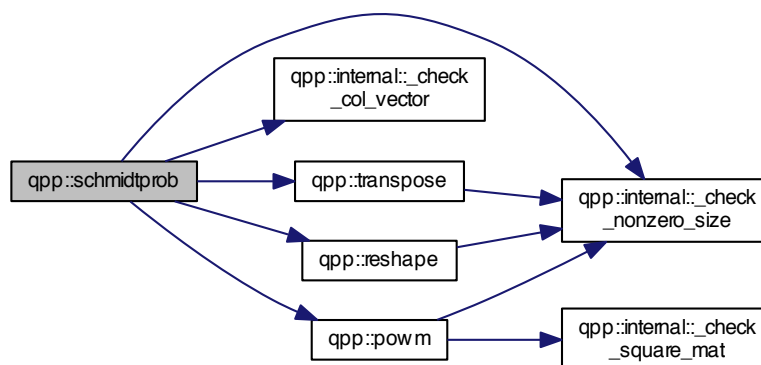
5.1.1.83 `template<typename Derived> types::cmat qpp::schmidtcoeff ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims )`

Here is the call graph for this function:



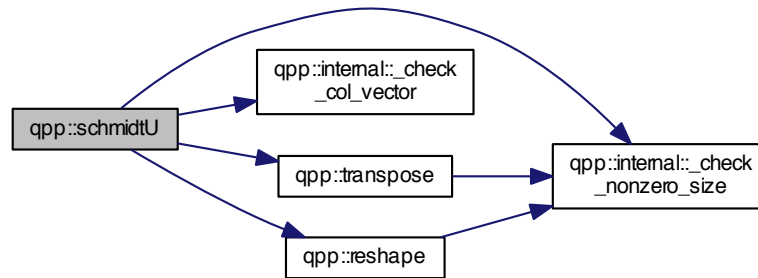
5.1.1.84 `template<typename Derived> types::cmat qpp::schmidtprob ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims )`

Here is the call graph for this function:



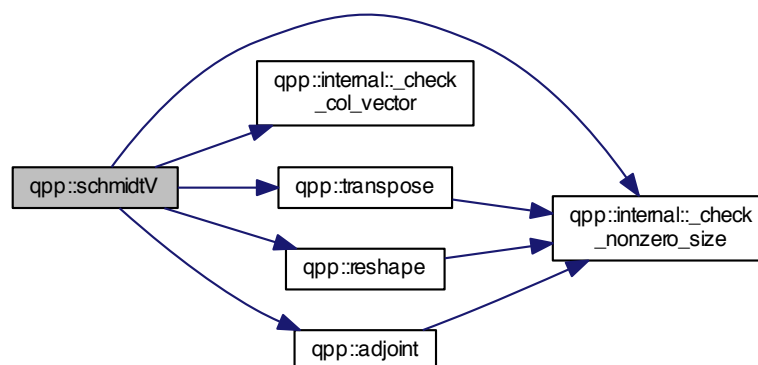
5.1.1.85 `template<typename Derived> types::cmat qpp::schmidtU ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims )`

Here is the call graph for this function:



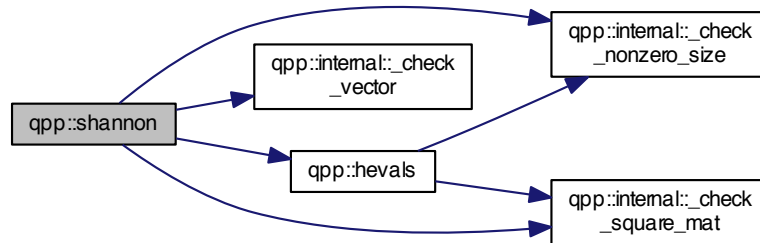
5.1.1.86 `template<typename Derived> types::cmat qpp::schmidtV ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims )`

Here is the call graph for this function:



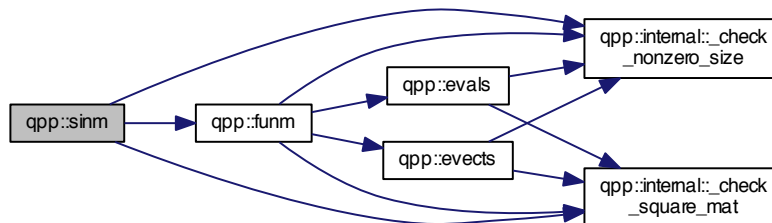
5.1.1.87 `template<typename Derived> double qpp::shannon ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



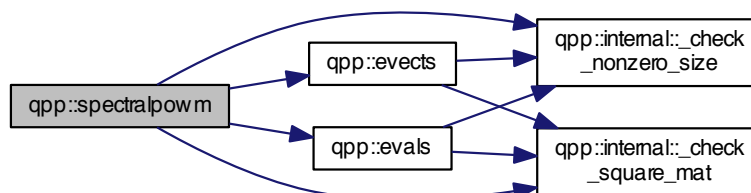
5.1.1.88 `template<typename Derived> types::cmat qpp::sinm ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



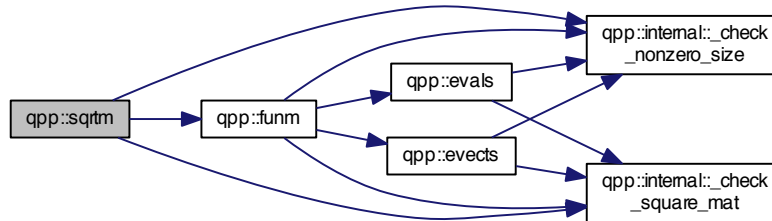
5.1.1.89 `template<typename Derived> types::cmat qpp::spectralpowm ( const Eigen::MatrixBase< Derived > & A, const types::cplx z )`

Here is the call graph for this function:



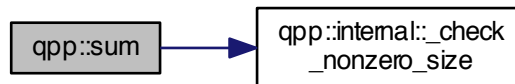
5.1.1.90 `template<typename Derived> types::cmat qpp::sqrtm ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



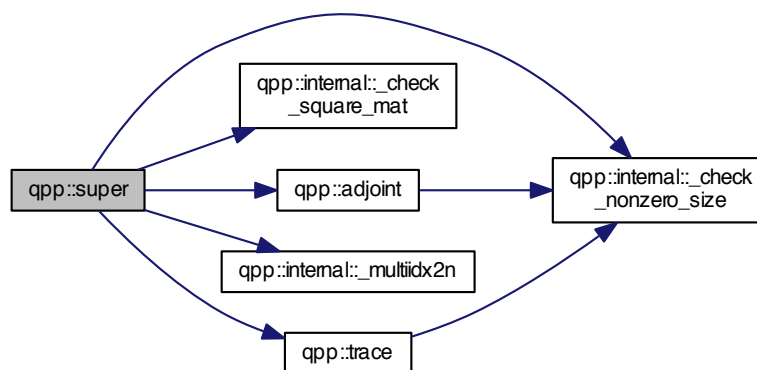
5.1.1.91 `template<typename Derived> Derived::Scalar qpp::sum ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



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

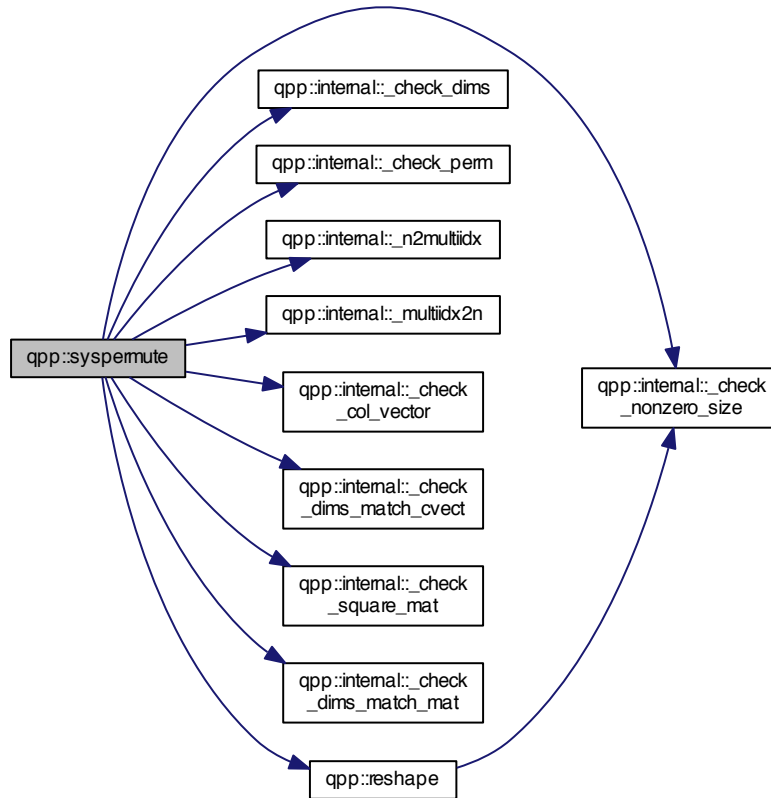
Here is the call graph for this function:





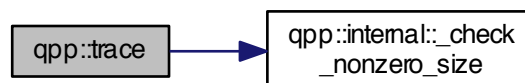
5.1.1.93 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::syspermute ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & perm, const std::vector< size_t> & dims )`

Here is the call graph for this function:



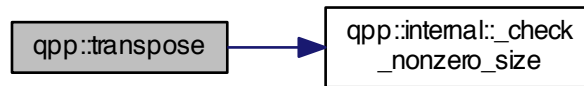
5.1.1.94 `template<typename Derived> Derived::Scalar qpp::trace ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



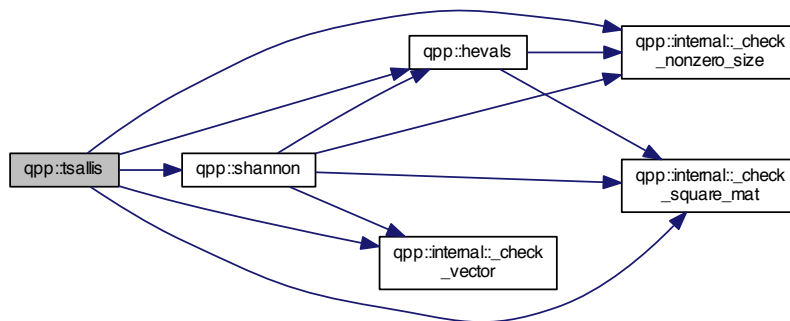
5.1.1.95 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::transpose ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



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

Here is the call graph for this function:



## 5.1.2 Variable Documentation

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

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

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

## 5.2 qpp::ct Namespace Reference

### Functions

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

### Variables

- `const double chop = 1e-10`

- const double [eps](#) = 1e-12
- const size\_t [maxn](#) = 64
- const std::complex< double > [ii](#) = { 0, 1 }
- const double [pi](#) = 3.141592653589793238462643383279502884
- const double [ee](#) = 2.718281828459045235360287471352662497

## 5.2.1 Function Documentation

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

## 5.2.2 Variable Documentation

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

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

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

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

5.2.2.5 `const size_t qpp::ct::maxn = 64`

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

## 5.3 qpp::internal Namespace Reference

### Functions

- void [\\_n2multiidx](#) (size\_t n, size\_t numdims, const size\_t \*dims, size\_t \*result)
- size\_t [\\_multiidx2n](#) (const size\_t \*midx, size\_t numdims, const size\_t \*dims)
- template<typename Derived >  
bool [\\_check\\_square\\_mat](#) (const Eigen::MatrixBase< Derived > &A)
- template<typename Derived >  
bool [\\_check\\_vector](#) (const Eigen::MatrixBase< Derived > &A)
- template<typename Derived >  
bool [\\_check\\_row\\_vector](#) (const Eigen::MatrixBase< Derived > &A)
- template<typename Derived >  
bool [\\_check\\_col\\_vector](#) (const Eigen::MatrixBase< Derived > &A)
- template<typename T >  
bool [\\_check\\_nonzero\\_size](#) (const T &x)
- bool [\\_check\\_dims](#) (const std::vector< size\_t > &dims)
- template<typename Derived >  
bool [\\_check\\_dims\\_match\\_mat](#) (const std::vector< size\_t > &dims, const Eigen::MatrixBase< Derived > &A)
- template<typename Derived >  
bool [\\_check\\_dims\\_match\\_cvect](#) (const std::vector< size\_t > &dims, const Eigen::MatrixBase< Derived > &V)
- template<typename Derived >  
bool [\\_check\\_dims\\_match\\_rvect](#) (const std::vector< size\_t > &dims, const Eigen::MatrixBase< Derived > &V)
- bool [\\_check\\_eq\\_dims](#) (const std::vector< size\_t > &dims, size\_t dim)
- bool [\\_check\\_subsys\\_match\\_dims](#) (const std::vector< size\_t > &subsys, const std::vector< size\_t > &dims)
- bool [\\_check\\_perm](#) (const std::vector< size\_t > &perm)

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

### 5.3.1 Function Documentation

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

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

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

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

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

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

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

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

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

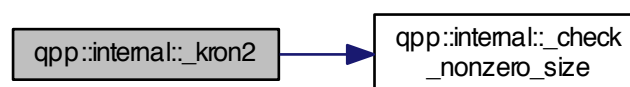
5.3.1.10 `template<typename Derived > bool qpp::internal::_check_square_mat ( const Eigen::MatrixBase< Derived > & A )`

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

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

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

Here is the call graph for this function:



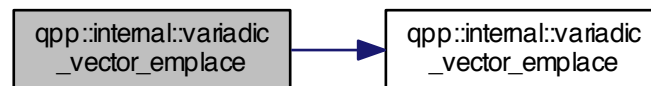
5.3.1.14 `size_t qpp::internal::_multiidx2n ( const size_t * midx, size_t numdims, const size_t * dims )`

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

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

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

Here is the call graph for this function:



## 5.4 qpp::types Namespace Reference

### Typedefs

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

### 5.4.1 Typedef Documentation

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

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

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

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

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

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



## Chapter 6

# Class Documentation

### 6.1 qpp::DiscreteDistribution Class Reference

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

#### Public Member Functions

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

#### Protected Attributes

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

#### 6.1.1 Constructor & Destructor Documentation

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

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

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

#### 6.1.2 Member Function Documentation

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

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

Here is the call graph for this function:



## 6.1.3 Member Data Documentation

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

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

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

## 6.2 `qpp::DiscreteDistributionAbsSquare` Class Reference

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

### Public Member Functions

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

### Protected Member Functions

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

### Protected Attributes

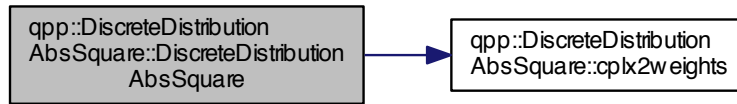
- `std::discrete_distribution`  
`< size_t > \_d`

### 6.2.1 Constructor & Destructor Documentation



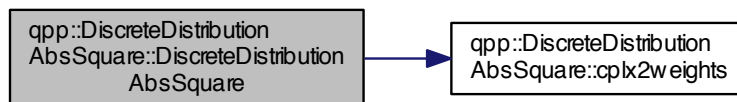
6.2.1.1 `template<typename InputIterator> qpp::DiscreteDistributionAbsSquare::DiscreteDistributionAbsSquare ( InputIterator first, InputIterator last ) [inline]`

Here is the call graph for this function:



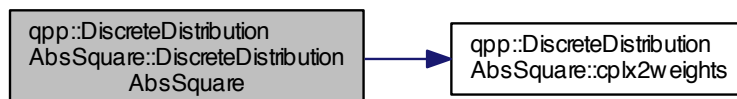
6.2.1.2 `qpp::DiscreteDistributionAbsSquare::DiscreteDistributionAbsSquare ( std::initializer_list< types::cplx > amplitudes ) [inline]`

Here is the call graph for this function:



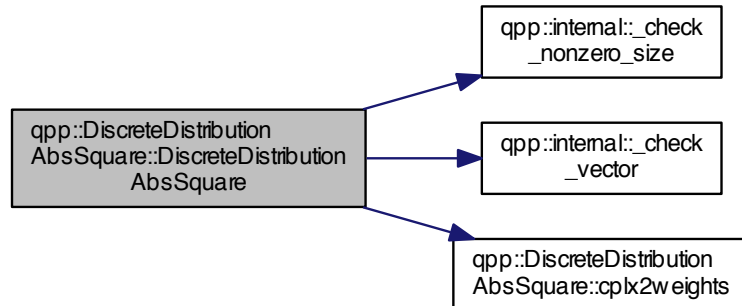
6.2.1.3 `qpp::DiscreteDistributionAbsSquare::DiscreteDistributionAbsSquare ( std::vector< types::cplx > amplitudes ) [inline]`

Here is the call graph for this function:



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

Here is the call graph for this function:



## 6.2.2 Member Function Documentation

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

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

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

Here is the call graph for this function:



## 6.2.3 Member Data Documentation

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

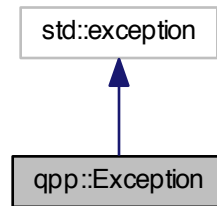
The documentation for this class was generated from the following file:

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

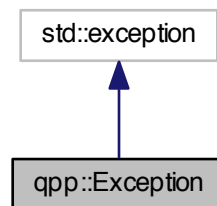
## 6.3 qpp::Exception Class Reference

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

Inheritance diagram for qpp::Exception:



Collaboration diagram for qpp::Exception:



## Public Types

- enum `Type` {  
`Type::UNKNOWN_EXCEPTION = 1`, `Type::ZERO_SIZE`, `Type::MATRIX_NOT_SQUARE`, `Type::MATRIX_NOT_CVECTOR`,  
`Type::MATRIX_NOT_RVECTOR`, `Type::MATRIX_NOT_VECTOR`, `Type::MATRIX_NOT_SQUARE_OR_CVECTOR`, `Type::MATRIX_NOT_SQUARE_OR_RVECTOR`,  
`Type::MATRIX_NOT_SQUARE_OR_VECTOR`, `Type::DIMS_INVALID`, `Type::DIMS_NOT_EQUAL`, `Type::DIMS_MISMATCH_MATRIX`,  
`Type::DIMS_MISMATCH_CVECTOR`, `Type::DIMS_MISMATCH_RVECTOR`, `Type::DIMS_MISMATCH_VECTOR`, `Type::SUBSYS_MISMATCH_DIMS`,  
`Type::PERM_INVALID`, `Type::NOT_QUBIT_GATE`, `Type::NOT_QUBIT_SUBSYS`, `Type::NOT_BIPARTITE`,  
`Type::OUT_OF_RANGE`, `Type::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
- virtual `~Exception` () noexcept

## Private Member Functions

- `std::string _construct_exception_msg ()`

## Private Attributes

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

## 6.3.1 Member Enumeration Documentation

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

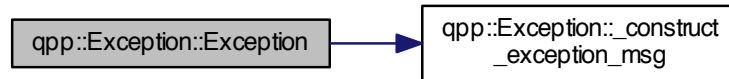
Enumerator

***UNKNOWN\_EXCEPTION***  
***ZERO\_SIZE***  
***MATRIX\_NOT\_SQUARE***  
***MATRIX\_NOT\_CVECTOR***  
***MATRIX\_NOT\_RVECTOR***  
***MATRIX\_NOT\_VECTOR***  
***MATRIX\_NOT\_SQUARE\_OR\_CVECTOR***  
***MATRIX\_NOT\_SQUARE\_OR\_RVECTOR***  
***MATRIX\_NOT\_SQUARE\_OR\_VECTOR***  
***DIMS\_INVALID***  
***DIMS\_NOT\_EQUAL***  
***DIMS\_MISMATCH\_MATRIX***  
***DIMS\_MISMATCH\_CVECTOR***  
***DIMS\_MISMATCH\_RVECTOR***  
***DIMS\_MISMATCH\_VECTOR***  
***SUBSYS\_MISMATCH\_DIMS***  
***PERM\_INVALID***  
***NOT\_QUBIT\_GATE***  
***NOT\_QUBIT\_SUBSYS***  
***NOT\_BIPARTITE***  
***OUT\_OF\_RANGE***  
***TYPE\_MISMATCH***  
***UNDEFINED\_TYPE***  
***CUSTOM\_EXCEPTION***

### 6.3.2 Constructor & Destructor Documentation

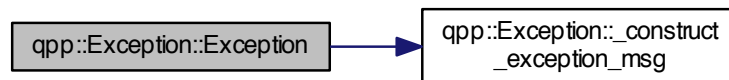
6.3.2.1 `qpp::Exception::Exception ( const std::string & where, const Type & type )` `[inline]`

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:



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

### 6.3.3 Member Function Documentation

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

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

### 6.3.4 Member Data Documentation

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

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

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

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

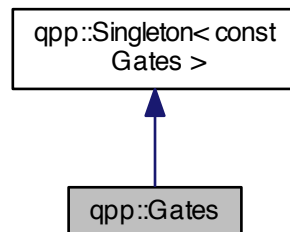
The documentation for this class was generated from the following file:

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

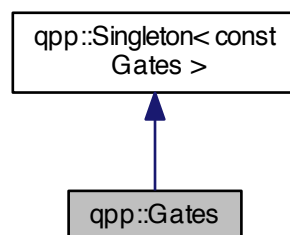
## 6.4 qpp::Gates Class Reference

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

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` (size\_t D) const
- `types::cmat Fd` (size\_t D) const
- `types::cmat Xd` (size\_t D) const
- template<typename Derived = Eigen::MatrixXcd>  
Derived `ld` (size\_t D) const
- template<typename Derived1 , typename Derived2 >  
`types::DynMat`< typename  
Derived1::Scalar > `apply` (const Eigen::MatrixBase< Derived1 > &state, const Eigen::MatrixBase< Derived2 > &A, const std::vector< size\_t > &subsys, const std::vector< size\_t > &dims) const
- template<typename Derived >  
`types::DynMat`< typename  
Derived::Scalar > `CTRL` (const Eigen::MatrixBase< Derived > &A, const std::vector< size\_t > &ctrl, const  
std::vector< size\_t > &subsys, size\_t n, 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 \(\)](#)
- [Gates \(\)](#)

## Friends

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

## Additional Inherited Members

### 6.4.1 Constructor & Destructor Documentation

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

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

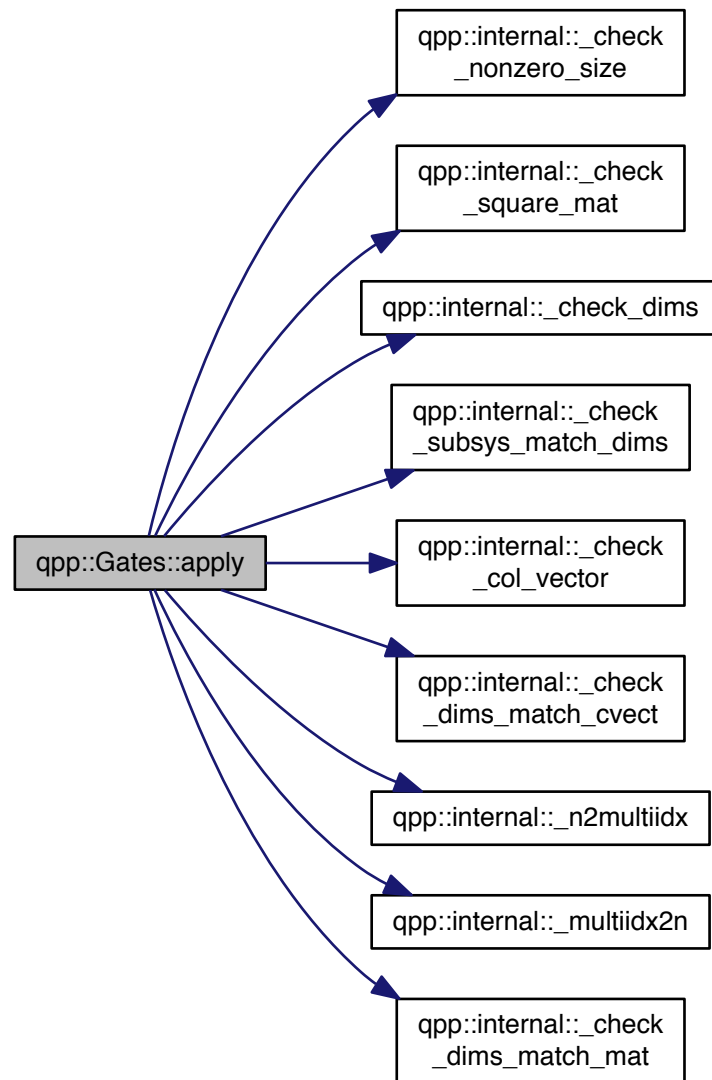
### 6.4.2 Member Function Documentation

```

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< size_t > & subsys, const std::vector< size_t > & dims ) const  [inline]

```

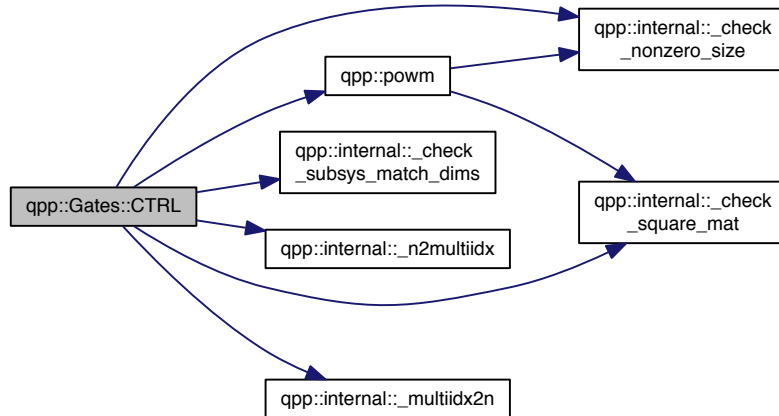
Here is the call graph for this function:





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

Here is the call graph for this function:



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

Here is the call graph for this function:

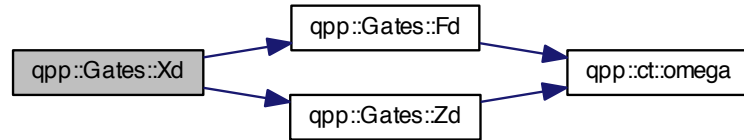


6.4.2.4 `template<typename Derived = Eigen::MatrixXcd> Derived qpp::Gates::Id ( size_t D ) const [inline]`

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

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

Here is the call graph for this function:



#### 6.4.2.7 `types::cmat qpp::Gates::Zd ( 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`

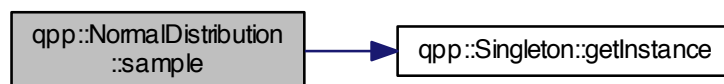
#### 6.5.1 Constructor & Destructor Documentation

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::getInstance\(\)](#).pz0)
- [size\\_t measure](#) (const [types::cmat](#) &U, bool destructive=false)
- [size\\_t measure](#) (bool destructive=false)
- [types::cmat getRho](#) () const
- [size\\_t getD](#) () const
- virtual [~Qudit](#) ()=default

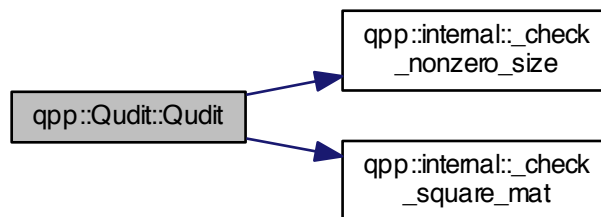
### Private Attributes

- [types::cmat \\_rho](#)
- [size\\_t \\_D](#)

### 6.6.1 Constructor & Destructor Documentation

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

Here is the call graph for this function:



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

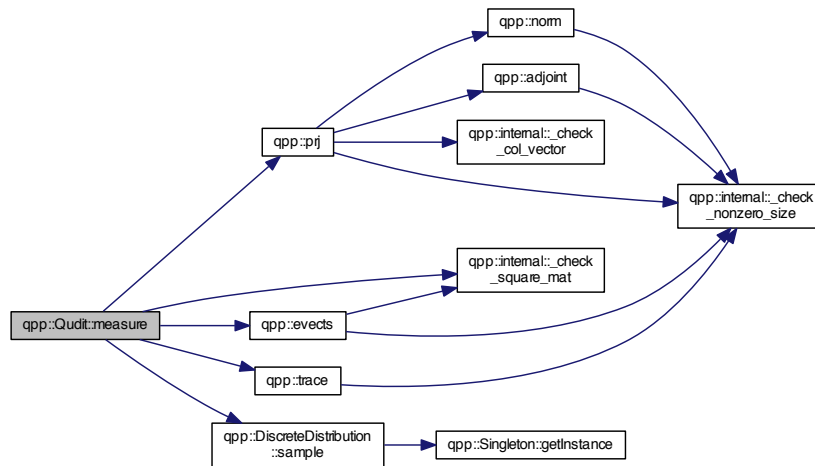
### 6.6.2 Member Function Documentation

6.6.2.1 `size\_t qpp::Qudit::getD ( ) const [inline]`

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

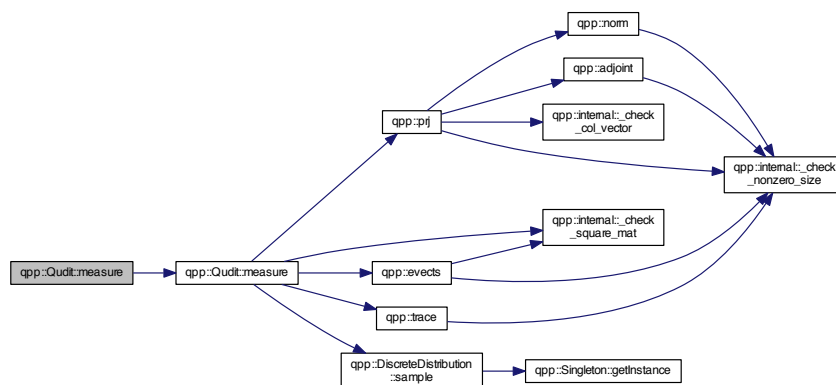
### 6.6.2.3 `size_t qpp::Qudit::measure ( const types::cmat & U, bool destructive = false ) [inline]`

Here is the call graph for this function:



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

Here is the call graph for this function:



## 6.6.3 Member Data Documentation

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

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

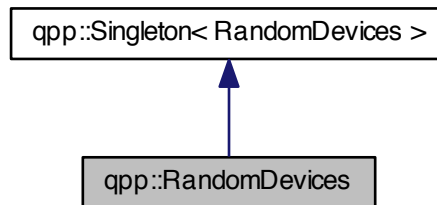
The documentation for this class was generated from the following file:

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

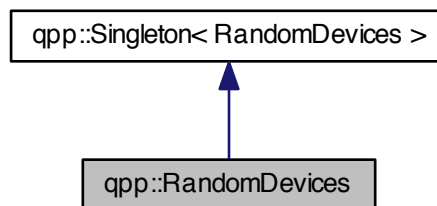
## 6.7 qpp::RandomDevices Class Reference

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

Inheritance diagram for qpp::RandomDevices:



Collaboration diagram for qpp::RandomDevices:



### Public Attributes

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

### Private Member Functions

- `~RandomDevices ()`
- `RandomDevices ()`

### Friends

- class `Singleton< RandomDevices >`

### Additional Inherited Members

### 6.7.1 Constructor & Destructor Documentation

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

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

### 6.7.2 Friends And Related Function Documentation

6.7.2.1 `friend class Singleton< 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`

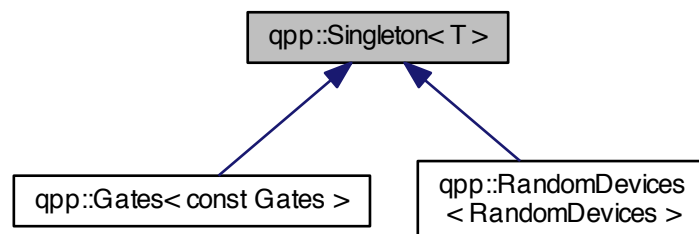
The documentation for this class was generated from the following file:

- `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 & getInstance ()`

### Protected Member Functions

- `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 ( ) [inline], [protected]`

6.8.1.2 `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 >::getInstance ( ) [inline], [static]`

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

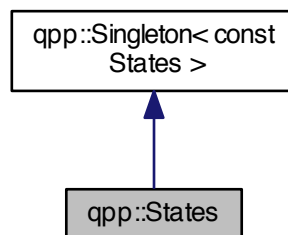
The documentation for this class was generated from the following file:

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

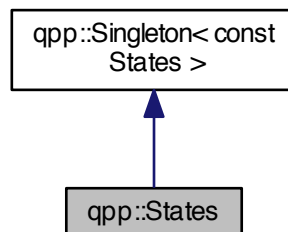
## 6.9 qpp::States Class Reference

```
#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 \(\)](#)
- [States \(\)](#)

## Friends

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

## Additional Inherited Members

### 6.9.1 Constructor & Destructor Documentation

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

6.9.1.2 [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
- virtual [~Timer](#) ()=default

### 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](#) ( ) [\[inline\]](#)

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

## 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::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`

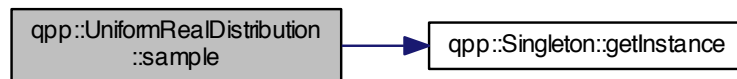
## 6.11.1 Constructor & Destructor Documentation

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

## 6.11.2 Member Function Documentation

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

Here is the call graph for this function:



## 6.11.3 Member Data Documentation

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

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

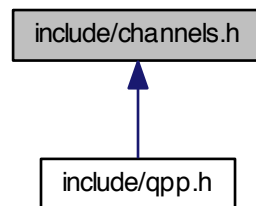
- `include/classes/stat.h`

## Chapter 7

# File Documentation

### 7.1 include/channels.h File Reference

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



### Namespaces

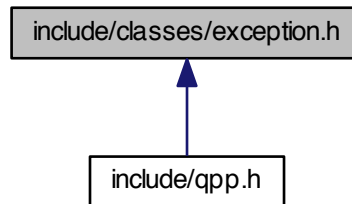
- [qpp](#)

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

## 7.2 include/classes/exception.h File Reference

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



### Classes

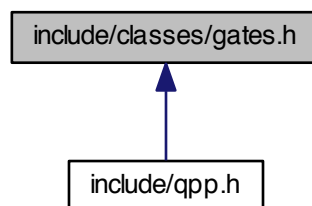
- class [qpp::Exception](#)

### Namespaces

- [qpp](#)

## 7.3 include/classes/gates.h File Reference

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



### Classes

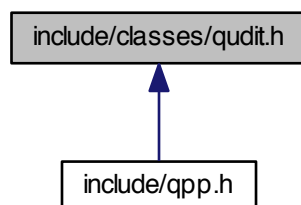
- class [qpp::Gates](#)

### Namespaces

- [qpp](#)

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

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



### Classes

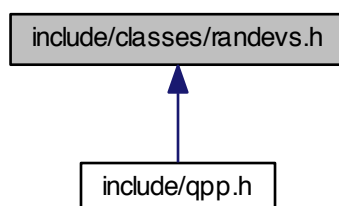
- class [qpp::Qudit](#)

### Namespaces

- [qpp](#)

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

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



### Classes

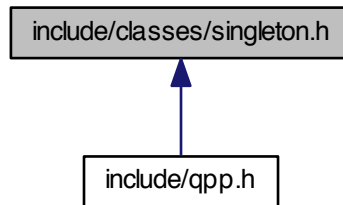
- class [qpp::RandomDevices](#)

### Namespaces

- [qpp](#)

## 7.6 include/classes/singleton.h File Reference

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



### Classes

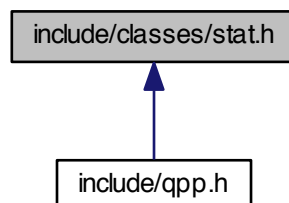
- class [qpp::Singleton< T >](#)

### Namespaces

- [qpp](#)

## 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::DiscreteDistribution](#)
- class [qpp::DiscreteDistributionAbsSquare](#)

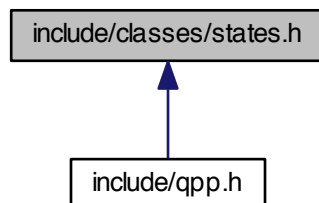


## Namespaces

- [qpp](#)

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

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



## Classes

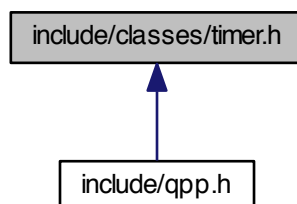
- class [qpp::States](#)

## Namespaces

- [qpp](#)

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

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



## Classes

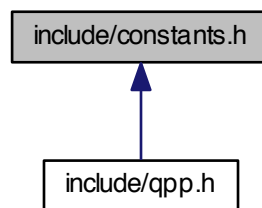
- class [qpp::Timer](#)

## 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

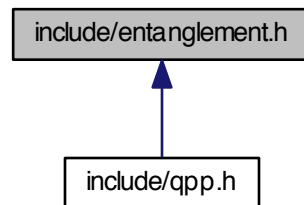
- `std::complex< double > qpp::ct::omega (size_t D)`

## Variables

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

## 7.11 include/entanglement.h File Reference

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



### Namespaces

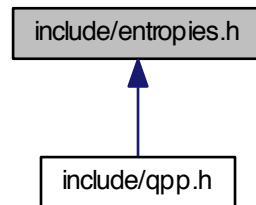
- [qpp](#)

### Functions

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

## 7.12 include/entropies.h File Reference

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



### Namespaces

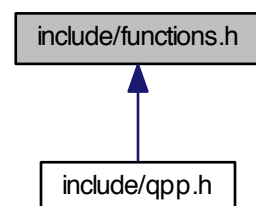
- [qpp](#)

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

## 7.13 include/functions.h File Reference

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



## Namespaces

- [qpp](#)

## Functions

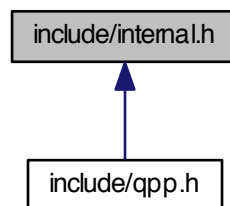
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::transpose (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::conjugate (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived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

- `template<typename OutputScalar , typename Derived >`  
`types::DynMat< OutputScalar > qpp::cwise (const Eigen::MatrixBase< Derived > &A, Output-`  
`Scalar(*)f)(const typename Derived::Scalar &))`
- `template<typename T >`  
`types::DynMat< typename T::Scalar > qpp::kron (const T &head)`
- `template<typename T , typename... Args>`  
`types::DynMat< typename T::Scalar > qpp::kron (const T &head, const Args &...tail)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::kron (const std::vector< Derived > &As)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::kron (const std::initializer_list< Derived > &As)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::kronpow (const Eigen::MatrixBase< Derived > &A, size_t n)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::reshape (const Eigen::MatrixBase< Derived > &A, size_t rows, size_t cols)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::syspermute (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t >`  
`&perm, const std::vector< size_t > &dims)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::ptrace1 (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t >`  
`&dims)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::ptrace2 (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t >`  
`&dims)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::ptrace (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &sub-`  
`sys, const std::vector< size_t > &dims)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::ptranspose (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t >`  
`&subsys, const std::vector< size_t > &dims)`
- `template<typename Derived1 , typename Derived2 >`  
`types::DynMat< typename`  
`Derived1::Scalar > qpp::comm (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase< De-`  
`derived2 > &B)`
- `template<typename Derived1 , typename Derived2 >`  
`types::DynMat< typename`  
`Derived1::Scalar > qpp::anticomm (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase<`  
`Derived2 > &B)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::prj (const Eigen::MatrixBase< Derived > &V)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::expandout (const Eigen::MatrixBase< Derived > &A, size_t pos, const std::vector<`  
`size_t > &dims)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::grams (const std::vector< Derived > &Vs)`

- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::grams (const std::initializer_list< Derived > &Vs)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::grams (const Eigen::MatrixBase< Derived > &A)`
- `std::vector< size_t > qpp::n2multiidx (size_t n, const std::vector< size_t > &dims)`
- `size_t qpp::multiidx2n (const std::vector< size_t > &midx, const std::vector< size_t > &dims)`
- `types::ket qpp::mket (const std::vector< size_t > &mask)`
- `types::ket qpp::mket (const std::vector< size_t > &mask, const std::vector< size_t > &dims)`
- `types::ket qpp::mket (const std::vector< size_t > &mask, size_t d)`
- `std::vector< size_t > qpp::invperm (const std::vector< size_t > &perm)`
- `std::vector< size_t > qpp::compperm (const std::vector< size_t > &perm, const std::vector< size_t > &sigma)`

## 7.14 include/internal.h File Reference

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



### Namespaces

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

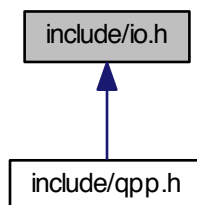
### Functions

- `void qpp::internal::\_n2multiidx (size_t n, size_t numdims, const size_t *dims, size_t *result)`
- `size_t qpp::internal::\_multiidx2n (const size_t *midx, size_t numdims, const size_t *dims)`
- `template<typename Derived >`  
`bool qpp::internal::\_check\_square\_mat (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`bool qpp::internal::\_check\_vector (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`bool qpp::internal::\_check\_row\_vector (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`bool qpp::internal::\_check\_col\_vector (const Eigen::MatrixBase< Derived > &A)`
- `template<typename T >`  
`bool qpp::internal::\_check\_nonzero\_size (const T &x)`
- `bool qpp::internal::\_check\_dims (const std::vector< size_t > &dims)`

- `template<typename Derived >`  
`bool qpp::internal::\_check\_dims\_match\_mat (const std::vector< size_t > &dims, const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`bool qpp::internal::\_check\_dims\_match\_cvect (const std::vector< size_t > &dims, const Eigen::MatrixBase< Derived > &V)`
- `template<typename Derived >`  
`bool qpp::internal::\_check\_dims\_match\_rvect (const std::vector< size_t > &dims, const Eigen::MatrixBase< Derived > &V)`
- `bool qpp::internal::\_check\_eq\_dims (const std::vector< size_t > &dims, size_t dim)`
- `bool qpp::internal::\_check\_subsys\_match\_dims (const std::vector< size_t > &subsys, const std::vector< size_t > &dims)`
- `bool qpp::internal::\_check\_perm (const std::vector< size_t > &perm)`
- `template<typename Derived1 , typename Derived2 >`  
`types::DynMat< typename Derived1::Scalar > qpp::internal::\_kron2 (const Eigen::MatrixBase< Derived1 > &A, const Eigen::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)`



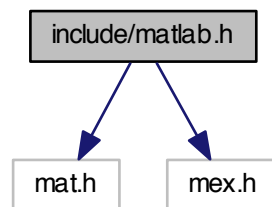
- `template<typename T >`  
`void qpp::disp (const T *x, const size_t n, const std::string &separator, const std::string &start="[" , const std::string &end="]", std::ostream &os=std::cout)`
- `template<typename T >`  
`void qpp::displn (const T *x, const size_t n, const std::string &separator, const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

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

## 7.17 include/qpp.h File Reference

```
#include <algorithm>
#include <cmath>
#include <chrono>
#include <complex>
#include <cstdint>
#include <cstdlib>
#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 "internal.h"
#include "functions.h"
#include "classes/states.h"
#include "classes/gates.h"
#include "classes/randevs.h"
#include "classes/stat.h"
#include "channels.h"
#include "entanglement.h"
#include "entropies.h"
#include "io.h"
#include "random.h"
#include "classes/qudit.h"
#include "classes/timer.h"
Include dependency graph for qpp.h:
```



## Namespaces

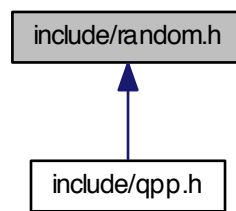
- `qpp`

## Variables

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

## 7.18 include/random.h File Reference

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



## Namespaces

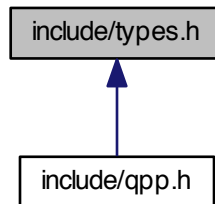
- [qpp](#)

## Functions

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

## 7.19 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 >`