

qpp  
0.1

Generated by Doxygen 1.8.5

Mon Apr 21 2014 19:06:45



# 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	conjugate . . . . .	17
5.1.1.10	cosm . . . . .	17
5.1.1.11	cwise . . . . .	17
5.1.1.12	det . . . . .	18
5.1.1.13	disp . . . . .	18
5.1.1.14	disp . . . . .	18
5.1.1.15	disp . . . . .	18
5.1.1.16	disp . . . . .	18
5.1.1.17	displn . . . . .	18
5.1.1.18	displn . . . . .	19

5.1.1.19	displn	19
5.1.1.20	displn	19
5.1.1.21	entanglement	20
5.1.1.22	evals	20
5.1.1.23	evects	21
5.1.1.24	expandout	21
5.1.1.25	expm	22
5.1.1.26	funm	22
5.1.1.27	gate	23
5.1.1.28	gconcurrence	24
5.1.1.29	grams	24
5.1.1.30	grams	25
5.1.1.31	hevals	25
5.1.1.32	hevects	25
5.1.1.33	kron	26
5.1.1.34	kronlist	26
5.1.1.35	kronpow	26
5.1.1.36	load	26
5.1.1.37	loadMATLABmatrix	26
5.1.1.38	loadMATLABmatrix	27
5.1.1.39	loadMATLABmatrix	27
5.1.1.40	logdet	27
5.1.1.41	logm	27
5.1.1.42	mket	27
5.1.1.43	mket	28
5.1.1.44	mket	28
5.1.1.45	multiidx2n	28
5.1.1.46	n2multiidx	29
5.1.1.47	norm	29
5.1.1.48	powm	29
5.1.1.49	prj	30
5.1.1.50	ptrace	31
5.1.1.51	ptrace1	32
5.1.1.52	ptrace2	32
5.1.1.53	ptranspose	33
5.1.1.54	qmutualinfo	34
5.1.1.55	rand	34
5.1.1.56	rand	34
5.1.1.57	rand	34
5.1.1.58	rand	35

5.1.1.59	<a href="#">randH</a>	35
5.1.1.60	<a href="#">randket</a>	35
5.1.1.61	<a href="#">randkraus</a>	36
5.1.1.62	<a href="#">randn</a>	36
5.1.1.63	<a href="#">randn</a>	36
5.1.1.64	<a href="#">randn</a>	36
5.1.1.65	<a href="#">randn</a>	36
5.1.1.66	<a href="#">randrho</a>	37
5.1.1.67	<a href="#">randU</a>	37
5.1.1.68	<a href="#">randV</a>	37
5.1.1.69	<a href="#">renyi</a>	37
5.1.1.70	<a href="#">renyi_inf</a>	38
5.1.1.71	<a href="#">reshape</a>	38
5.1.1.72	<a href="#">save</a>	38
5.1.1.73	<a href="#">saveMATLABmatrix</a>	38
5.1.1.74	<a href="#">saveMATLABmatrix</a>	39
5.1.1.75	<a href="#">saveMATLABmatrix</a>	39
5.1.1.76	<a href="#">schmidtcoeff</a>	39
5.1.1.77	<a href="#">schmidtprob</a>	40
5.1.1.78	<a href="#">schmidtU</a>	40
5.1.1.79	<a href="#">schmidtV</a>	41
5.1.1.80	<a href="#">shannon</a>	41
5.1.1.81	<a href="#">sinm</a>	42
5.1.1.82	<a href="#">spectralpowm</a>	42
5.1.1.83	<a href="#">sqrtm</a>	42
5.1.1.84	<a href="#">sum</a>	43
5.1.1.85	<a href="#">super</a>	43
5.1.1.86	<a href="#">syspermute</a>	44
5.1.1.87	<a href="#">trace</a>	44
5.1.1.88	<a href="#">transpose</a>	45
5.1.1.89	<a href="#">tsallis</a>	45
5.1.2	<a href="#">Variable Documentation</a>	45
5.1.2.1	<a href="#">gt</a>	45
5.1.2.2	<a href="#">rdevs</a>	45
5.1.2.3	<a href="#">st</a>	45
5.2	<a href="#">qpp::ct Namespace Reference</a>	45
5.2.1	<a href="#">Function Documentation</a>	46
5.2.1.1	<a href="#">omega</a>	46
5.2.2	<a href="#">Variable Documentation</a>	46
5.2.2.1	<a href="#">chop</a>	46

5.2.2.2	ee	46
5.2.2.3	eps	46
5.2.2.4	ii	46
5.2.2.5	maxn	46
5.2.2.6	pi	46
5.3	qpp::internal Namespace Reference	46
5.3.1	Function Documentation	47
5.3.1.1	_check_col_vector	47
5.3.1.2	_check_dims	47
5.3.1.3	_check_dims_match_cvect	47
5.3.1.4	_check_dims_match_mat	47
5.3.1.5	_check_dims_match_rvect	47
5.3.1.6	_check_eq_dims	47
5.3.1.7	_check_nonzero_size	47
5.3.1.8	_check_perm_match_dims	47
5.3.1.9	_check_row_vector	47
5.3.1.10	_check_square_mat	47
5.3.1.11	_check_subsys_match_dims	47
5.3.1.12	_check_vector	47
5.3.1.13	_multiidx2n	47
5.3.1.14	_n2multiidx	47
5.4	qpp::types Namespace Reference	47
5.4.1	Typedef Documentation	47
5.4.1.1	bra	47
5.4.1.2	cmat	47
5.4.1.3	cplx	48
5.4.1.4	dmat	48
5.4.1.5	DynMat	48
5.4.1.6	ket	48
<b>6</b>	<b>Class Documentation</b>	<b>49</b>
6.1	qpp::DiscreteDistribution Class Reference	49
6.1.1	Constructor & Destructor Documentation	49
6.1.1.1	DiscreteDistribution	49
6.1.1.2	DiscreteDistribution	49
6.1.1.3	DiscreteDistribution	49
6.1.2	Member Function Documentation	49
6.1.2.1	probabilities	49
6.1.2.2	sample	50
6.1.3	Member Data Documentation	50

6.1.3.1	<a href="#">_d</a>	50
6.2	<a href="#">qpp::DiscreteDistributionAbsSquare Class Reference</a>	50
6.2.1	<a href="#">Constructor &amp; Destructor Documentation</a>	50
6.2.1.1	<a href="#">DiscreteDistributionAbsSquare</a>	51
6.2.1.2	<a href="#">DiscreteDistributionAbsSquare</a>	51
6.2.1.3	<a href="#">DiscreteDistributionAbsSquare</a>	51
6.2.1.4	<a href="#">DiscreteDistributionAbsSquare</a>	52
6.2.2	<a href="#">Member Function Documentation</a>	52
6.2.2.1	<a href="#">cplx2weights</a>	52
6.2.2.2	<a href="#">probabilities</a>	52
6.2.2.3	<a href="#">sample</a>	52
6.2.3	<a href="#">Member Data Documentation</a>	52
6.2.3.1	<a href="#">_d</a>	52
6.3	<a href="#">qpp::Exception Class Reference</a>	52
6.3.1	<a href="#">Member Enumeration Documentation</a>	54
6.3.1.1	<a href="#">Type</a>	54
6.3.2	<a href="#">Constructor &amp; Destructor Documentation</a>	55
6.3.2.1	<a href="#">Exception</a>	55
6.3.2.2	<a href="#">Exception</a>	55
6.3.2.3	<a href="#">~Exception</a>	55
6.3.3	<a href="#">Member Function Documentation</a>	55
6.3.3.1	<a href="#">_construct_exception_msg</a>	55
6.3.3.2	<a href="#">what</a>	55
6.3.4	<a href="#">Member Data Documentation</a>	55
6.3.4.1	<a href="#">_custom</a>	55
6.3.4.2	<a href="#">_msg</a>	55
6.3.4.3	<a href="#">_type</a>	55
6.3.4.4	<a href="#">_where</a>	55
6.4	<a href="#">qpp::Gates Class Reference</a>	56
6.4.1	<a href="#">Constructor &amp; Destructor Documentation</a>	56
6.4.1.1	<a href="#">Gates</a>	56
6.4.1.2	<a href="#">Gates</a>	56
6.4.1.3	<a href="#">~Gates</a>	56
6.4.2	<a href="#">Member Function Documentation</a>	56
6.4.2.1	<a href="#">CTRL</a>	57
6.4.2.2	<a href="#">Fd</a>	57
6.4.2.3	<a href="#">getInstance</a>	57
6.4.2.4	<a href="#">Id</a>	57
6.4.2.5	<a href="#">operator=</a>	57
6.4.2.6	<a href="#">Rtheta</a>	57

6.4.2.7	Xd	58
6.4.2.8	Zd	58
6.4.3	Member Data Documentation	58
6.4.3.1	CNOTab	58
6.4.3.2	CNOTba	58
6.4.3.3	CZ	58
6.4.3.4	FRED	58
6.4.3.5	H	58
6.4.3.6	Id2	58
6.4.3.7	S	58
6.4.3.8	SWAP	58
6.4.3.9	T	58
6.4.3.10	TOF	58
6.4.3.11	X	58
6.4.3.12	Y	59
6.4.3.13	Z	59
6.5	qpp::NormalDistribution Class Reference	59
6.5.1	Constructor & Destructor Documentation	59
6.5.1.1	NormalDistribution	59
6.5.2	Member Function Documentation	59
6.5.2.1	sample	59
6.5.3	Member Data Documentation	59
6.5.3.1	_d	59
6.6	qpp::Qudit Class Reference	60
6.6.1	Constructor & Destructor Documentation	60
6.6.1.1	Qudit	60
6.6.1.2	~Qudit	60
6.6.2	Member Function Documentation	60
6.6.2.1	getD	60
6.6.2.2	getRho	60
6.6.2.3	measure	61
6.6.2.4	measure	61
6.6.3	Member Data Documentation	61
6.6.3.1	_D	61
6.6.3.2	_rho	61
6.7	qpp::RandomDevices Class Reference	62
6.7.1	Constructor & Destructor Documentation	62
6.7.1.1	RandomDevices	62
6.7.1.2	RandomDevices	62
6.7.1.3	~RandomDevices	62



6.7.2	Member Function Documentation	62
6.7.2.1	getInstance	62
6.7.2.2	operator=	62
6.7.3	Member Data Documentation	62
6.7.3.1	_rd	62
6.7.3.2	_rng	62
6.8	qpp::States Class Reference	62
6.8.1	Constructor & Destructor Documentation	63
6.8.1.1	States	63
6.8.1.2	States	63
6.8.1.3	~States	63
6.8.2	Member Function Documentation	63
6.8.2.1	getInstance	63
6.8.2.2	operator=	64
6.8.3	Member Data Documentation	64
6.8.3.1	b00	64
6.8.3.2	b01	64
6.8.3.3	b10	64
6.8.3.4	b11	64
6.8.3.5	GHZ	64
6.8.3.6	pb00	64
6.8.3.7	pb01	64
6.8.3.8	pb10	64
6.8.3.9	pb11	64
6.8.3.10	pGHZ	64
6.8.3.11	pW	64
6.8.3.12	px0	64
6.8.3.13	px1	64
6.8.3.14	py0	64
6.8.3.15	py1	64
6.8.3.16	pz0	64
6.8.3.17	pz1	64
6.8.3.18	W	64
6.8.3.19	x0	64
6.8.3.20	x1	64
6.8.3.21	y0	64
6.8.3.22	y1	64
6.8.3.23	z0	64
6.8.3.24	z1	64
6.9	qpp::Timer Class Reference	65

6.9.1	Constructor & Destructor Documentation	65
6.9.1.1	Timer	65
6.9.1.2	~Timer	65
6.9.2	Member Function Documentation	65
6.9.2.1	seconds	65
6.9.2.2	tic	65
6.9.2.3	toc	65
6.9.3	Friends And Related Function Documentation	65
6.9.3.1	operator<<	65
6.9.4	Member Data Documentation	65
6.9.4.1	_end	65
6.9.4.2	_start	65
6.10	qpp::UniformRealDistribution Class Reference	65
6.10.1	Constructor & Destructor Documentation	66
6.10.1.1	UniformRealDistribution	66
6.10.2	Member Function Documentation	66
6.10.2.1	sample	66
6.10.3	Member Data Documentation	66
6.10.3.1	_d	66
<b>7</b>	<b>File Documentation</b>	<b>67</b>
7.1	include/channels.h File Reference	67
7.2	include/classes/exception.h File Reference	68
7.3	include/classes/gates.h File Reference	69
7.4	include/classes/qudit.h File Reference	70
7.5	include/classes/randevs.h File Reference	71
7.6	include/classes/stat.h File Reference	72
7.7	include/classes/states.h File Reference	74
7.8	include/classes/timer.h File Reference	75
7.9	include/constants.h File Reference	75
7.10	include/entanglement.h File Reference	77
7.11	include/entropies.h File Reference	78
7.12	include/functions.h File Reference	79
7.13	include/internal.h File Reference	82
7.14	include/io.h File Reference	84
7.15	include/matlab.h File Reference	85
7.16	include/qpp.h File Reference	86
7.17	include/random.h File Reference	87
7.18	include/types.h File Reference	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>	45
<a href="#">qpp::internal</a>	46
<a href="#">qpp::types</a>	47



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

qpp::DiscreteDistribution . . . . .	49
qpp::DiscreteDistributionAbsSquare . . . . .	50
exception	
qpp::Exception . . . . .	52
qpp::Gates . . . . .	56
qpp::NormalDistribution . . . . .	59
qpp::Qudit . . . . .	60
qpp::RandomDevices . . . . .	62
qpp::States . . . . .	62
qpp::Timer . . . . .	65
qpp::UniformRealDistribution . . . . .	65



## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">qpp::DiscreteDistribution</a>	49
<a href="#">qpp::DiscreteDistributionAbsSquare</a>	50
<a href="#">qpp::Exception</a>	52
<a href="#">qpp::Gates</a>	56
<a href="#">qpp::NormalDistribution</a>	59
<a href="#">qpp::Qudit</a>	60
<a href="#">qpp::RandomDevices</a>	62
<a href="#">qpp::States</a>	62
<a href="#">qpp::Timer</a>	65
<a href="#">qpp::UniformRealDistribution</a>	65





## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

include/channels.h	67
include/constants.h	75
include/entanglement.h	77
include/entropies.h	78
include/functions.h	79
include/internal.h	82
include/io.h	84
include/matlab.h	85
include/qpp.h	86
include/random.h	87
include/types.h	88
include/classes/exception.h	68
include/classes/gates.h	69
include/classes/qudit.h	70
include/classes/randevs.h	71
include/classes/stat.h	72
include/classes/states.h	74
include/classes/timer.h	75



## Chapter 5

# Namespace Documentation

### 5.1 qpp Namespace Reference

#### Namespaces

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

#### Classes

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

#### Functions

- `template<typename Derived >`  
`types::cmat channel` (const Eigen::MatrixBase< Derived > &rho, const std::vector< [types::cmat](#) > &Ks)
- `types::cmat super` (const std::vector< [types::cmat](#) > &Ks)
- `types::cmat choi` (const std::vector< [types::cmat](#) > &Ks)
- `std::vector< types::cmat > choi2kraus` (const [types::cmat](#) &A)
- `template<typename Derived >`  
`types::cmat 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 Derived1 , typename Derived2 >`  
`types::DynMat< typename`  
`Derived1::Scalar > gate` (const Eigen::MatrixBase< Derived2 > &state, const Eigen::MatrixBase< Derived1 > &A, 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 >`  
`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(\*f)(const  
typename Derived::Scalar &))
- `template<typename Derived1, typename Derived2 >`  
`types::DynMat< typename`  
`Derived1::Scalar > kron` (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase< Derived2 >  
&B)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > kronlist` (const std::vector< `types::DynMat< typename` Derived::Scalar > > &As)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > kronpow` (const Eigen::MatrixBase< Derived > &A, size\_t n)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > reshape` (const Eigen::MatrixBase< Derived > &A, size\_t rows, size\_t cols)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > syspermute` (const Eigen::MatrixBase< Derived > &A, const std::vector< size\_t > &perm,  
const std::vector< size\_t > &dims)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > 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 > &subsys,  
const std::vector< size\_t > &dims)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > ptranspose` (const Eigen::MatrixBase< Derived > &A, const std::vector< size\_t > &sub-  
sys, const std::vector< size\_t > &dims)
- `template<typename Derived1, typename Derived2 >`  
`types::DynMat< typename`  
`Derived1::Scalar > comm` (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase< Derived2  
> &B)
- `template<typename Derived1, typename Derived2 >`  
`types::DynMat< typename`  
`Derived1::Scalar > anticomm` (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase< De-  
rived2 > &B)
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > 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< types::DynMat< typename Derived::Scalar > > &Vs)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > grams (const Eigen::MatrixBase< Derived > &A)`
- `std::vector< size_t > n2multiidx (size_t n, const std::vector< size_t > &dims)`
- `size_t multiidx2n (const std::vector< size_t > &midx, const std::vector< size_t > &dims)`
- `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)`
- `template<typename T >`  
`void disp (const T &x, const std::string &separator, const std::string &start="[" , const std::string &end=""]",`  
`std::ostream &os=std::cout)`
- `template<typename T >`  
`void displn (const T &x, const std::string &separator, const std::string &start="[" , const std::string &end=""]",`  
`std::ostream &os=std::cout)`
- `template<typename T >`  
`void disp (const T *x, const size_t n, const std::string &separator, const std::string &start="[" , const std::string`  
`&end=""]", std::ostream &os=std::cout)`
- `template<typename T >`  
`void displn (const T *x, const size_t n, const std::string &separator, const std::string &start="[" , const std-`  
`::string &end=""]", std::ostream &os=std::cout)`
- `template<typename Derived >`  
`void disp (const Eigen::MatrixBase< Derived > &A, double chop=ct::chop, std::ostream &os=std::cout)`
- `template<typename Derived >`  
`void displn (const Eigen::MatrixBase< Derived > &A, double chop=ct::chop, std::ostream &os=std::cout)`
- `void disp (const types::cplx c, double chop=ct::chop, std::ostream &os=std::cout)`
- `void displn (const types::cplx c, double chop=ct::chop, std::ostream &os=std::cout)`
- `template<typename Derived >`  
`void save (const Eigen::MatrixBase< Derived > &A, const std::string &fname)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > load (const std::string &fname)`
- `template<typename Derived >`  
`Derived loadMATLABmatrix (const std::string &mat_file, const std::string &var_name)`
- `template<>`  
`types::dmat loadMATLABmatrix (const std::string &mat_file, const std::string &var_name)`
- `template<>`  
`types::cmat loadMATLABmatrix (const std::string &mat_file, const std::string &var_name)`
- `template<typename Derived >`  
`void saveMATLABmatrix (const Eigen::MatrixBase< Derived > &A, const std::string &mat_file, const std-`  
`::string &var_name, const std::string &mode)`
- `template<>`  
`void saveMATLABmatrix (const Eigen::MatrixBase< typename types::dmat > &A, const std::string &mat_file,`  
`const std::string &var_name, const std::string &mode)`
- `template<>`  
`void saveMATLABmatrix (const Eigen::MatrixBase< typename types::cmat > &A, const std::string &mat_file,`  
`const std::string &var_name, const std::string &mode)`
- `template<typename Derived >`  
`Derived rand (size_t rows, size_t cols, double a=0, double b=1)`
- `template<>`  
`types::dmat rand (size_t rows, size_t cols, double a, double b)`
- `template<>`  
`types::cmat rand (size_t rows, size_t cols, double a, double b)`
- `double rand (double a=0, double b=1)`

- `template<typename Derived >`  
Derived `randn` (size\_t rows, size\_t cols, double mean=0, double sigma=1)
- `template<>`  
`types::dmat randn` (size\_t rows, size\_t cols, double mean, double sigma)
- `template<>`  
`types::cmat randn` (size\_t rows, size\_t cols, double mean, double sigma)
- `double randn` (double mean=0, double sigma=1)
- `types::cmat randU` (size\_t D)
- `types::cmat randV` (size\_t Din, size\_t Dout)
- `std::vector< types::cmat > randkraus` (size\_t n, size\_t D)
- `types::cmat randH` (size\_t D)
- `types::ket randket` (size\_t D)
- `types::cmat randrho` (size\_t D)

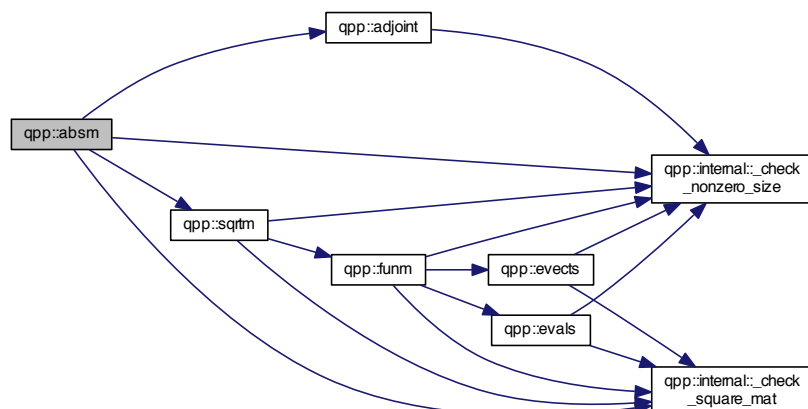
## Variables

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

### 5.1.1 Function Documentation

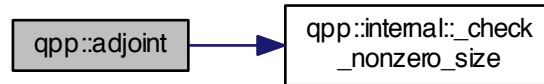
#### 5.1.1.1 `template<typename Derived > types::cmat qpp::absm ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



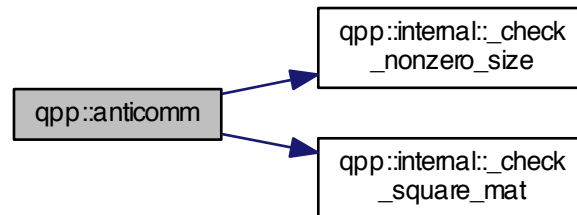
5.1.1.2 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::adjoint ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



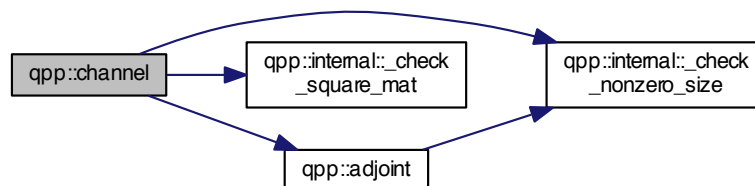
5.1.1.3 `template<typename Derived1 , typename Derived2 > types::DynMat<typename Derived1::Scalar> qpp::anticomm ( const Eigen::MatrixBase< Derived1 > & A, const Eigen::MatrixBase< Derived2 > & B )`

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 )`

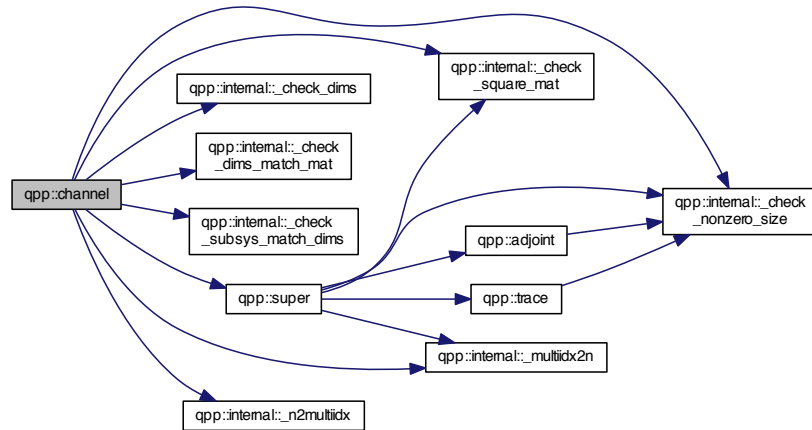
Here is the call graph for this function:





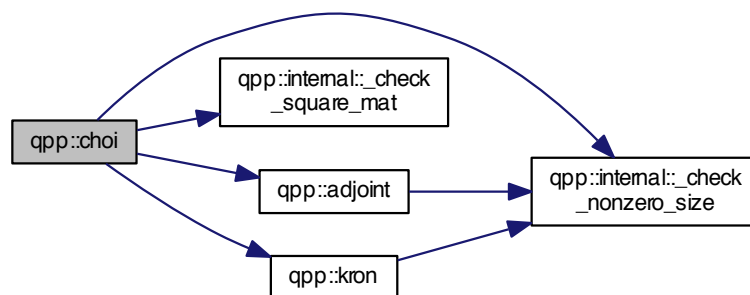
5.1.1.5 `template<typename Derived > types::cmat qpp::channel ( const Eigen::MatrixBase< Derived > & rho, const std::vector< types::cmat > & Ks, const std::vector< size_t > & subsys, const std::vector< size_t > & dims )`

Here is the call graph for this function:



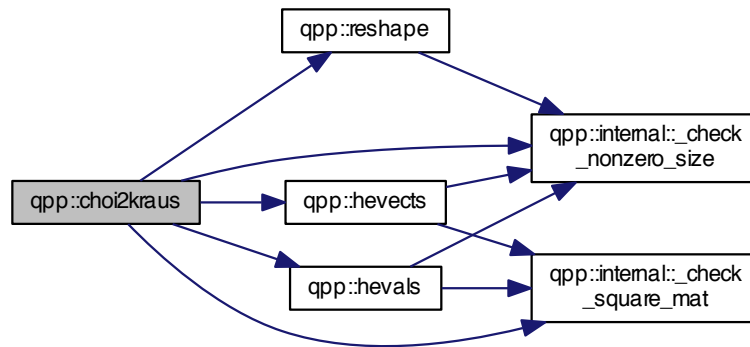
5.1.1.6 `types::cmat qpp::choi ( const std::vector< types::cmat > & Ks )`

Here is the call graph for this function:



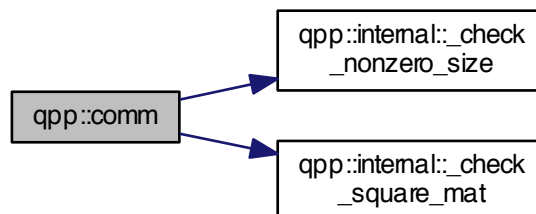
#### 5.1.1.7 `std::vector<types::cmat> qpp::choi2kraus ( const types::cmat & A )`

Here is the call graph for this function:



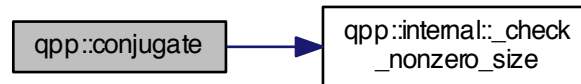
#### 5.1.1.8 `template<typename Derived1 , typename Derived2 > types::DynMat<typename Derived1::Scalar> qpp::comm ( const Eigen::MatrixBase< Derived1 > & A, const Eigen::MatrixBase< Derived2 > & B )`

Here is the call graph for this function:



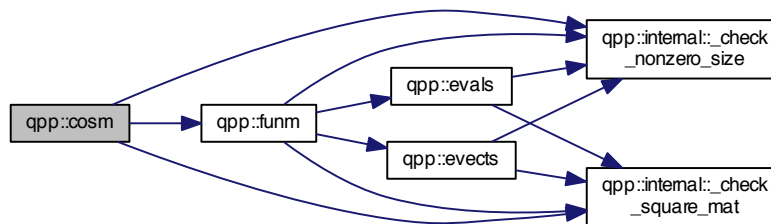
5.1.1.9 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::conjugate ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



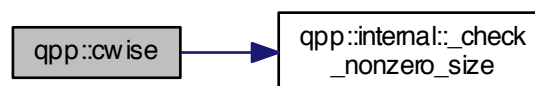
5.1.1.10 `template<typename Derived > types::cmat qpp::cosm ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



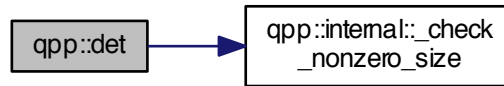
5.1.1.11 `template<typename OutputScalar , typename Derived > types::DynMat<OutputScalar> qpp::cwise ( const Eigen::MatrixBase< Derived > & A, OutputScalar*)(const typename Derived::Scalar &) f )`

Here is the call graph for this function:



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

Here is the call graph for this function:



5.1.1.13 `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 )`

5.1.1.14 `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 )`

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

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

Here is the call graph for this function:



5.1.1.17 `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:



5.1.1.18 `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:



5.1.1.19 `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:



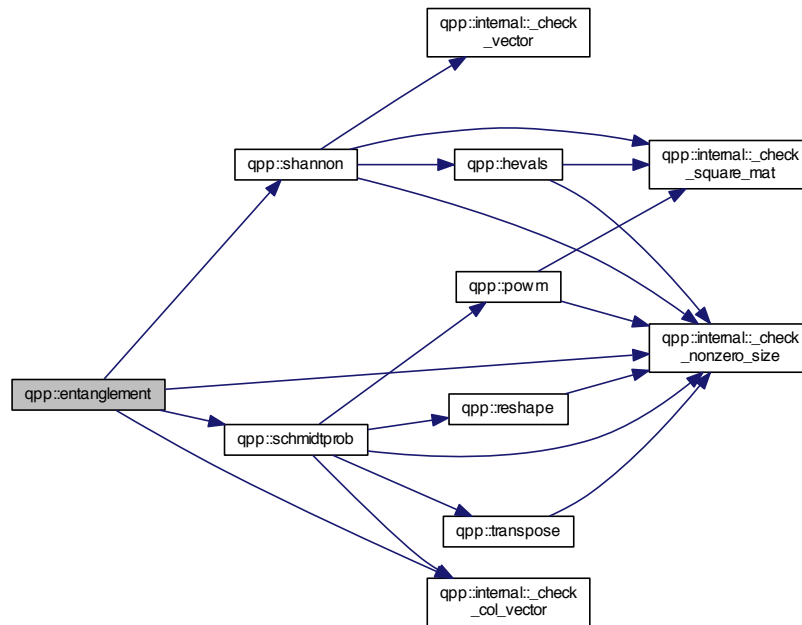
5.1.1.20 `void qpp::displn ( const types::cplx c, double chop = ct::chop, std::ostream & os = std::cout )`

Here is the call graph for this function:



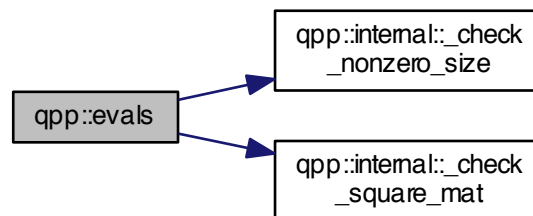
5.1.1.21 `template<typename Derived> double qpp::entanglement ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims )`

Here is the call graph for this function:



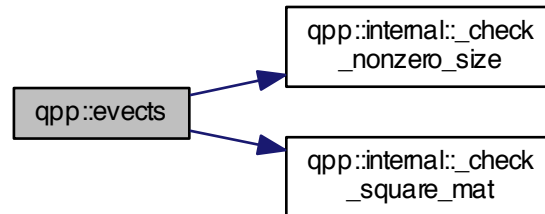
5.1.1.22 `template<typename Derived> types::cmat qpp::evals ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



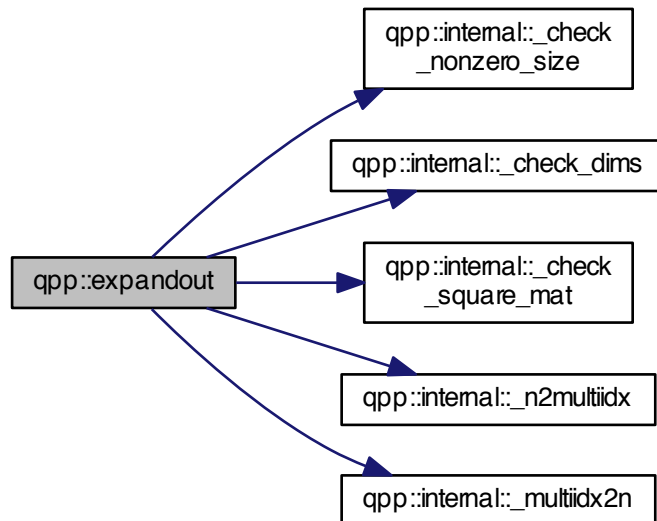
5.1.1.23 `template<typename Derived> types::cmat qpp::evects ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



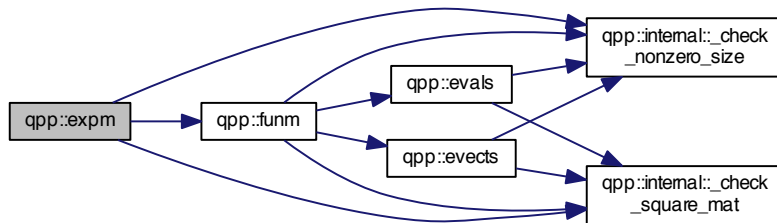
5.1.1.24 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::expandout ( const Eigen::MatrixBase< Derived> & A, size_t pos, const std::vector< size_t> & dims )`

Here is the call graph for this function:



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

Here is the call graph for this function:



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

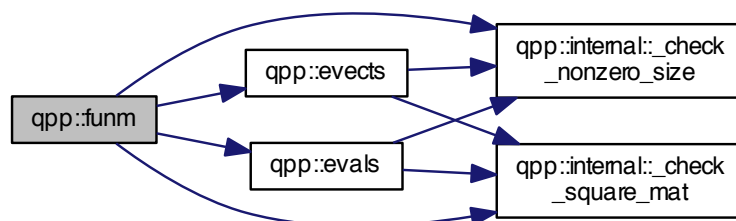
##### Parameters

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

##### Returns

`types::cmat`

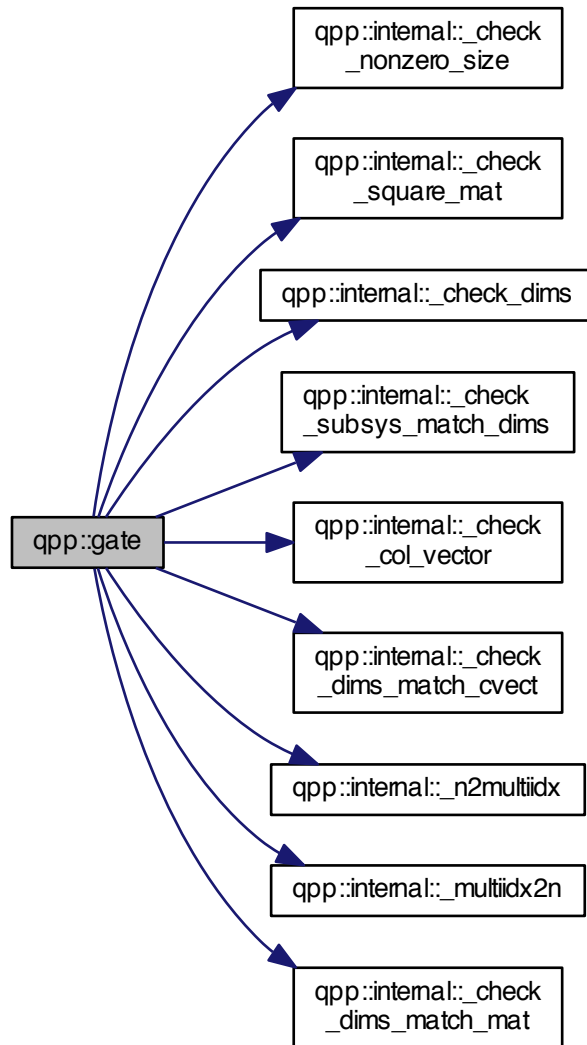
Here is the call graph for this function:





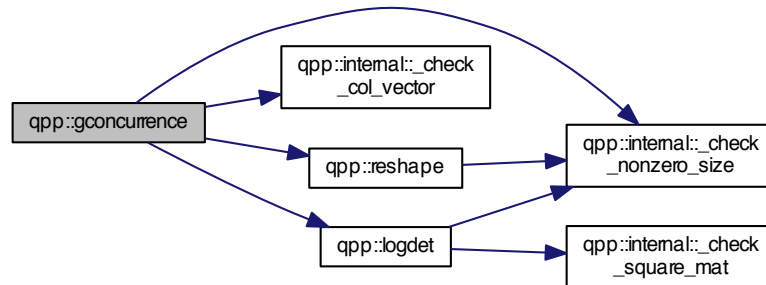
5.1.1.27 `template<typename Derived1 , typename Derived2 > types::DynMat<typename Derived1::Scalar> qpp::gate ( const Eigen::MatrixBase< Derived2 > & state, const Eigen::MatrixBase< Derived1 > & A, const std::vector< size_t > & subsys, const std::vector< size_t > & dims )`

Here is the call graph for this function:



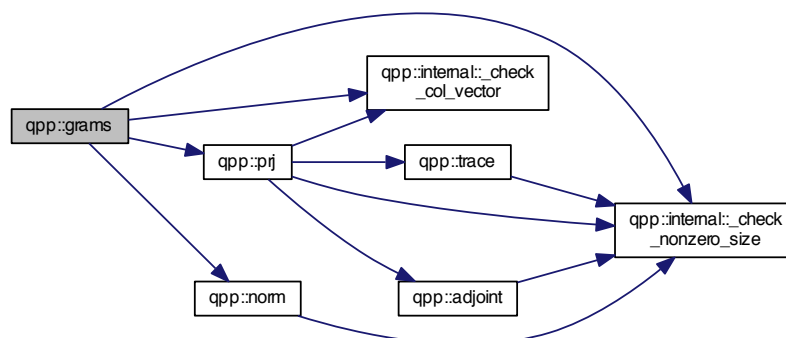
5.1.1.28 `template<typename Derived > double qpp::gconcurrency ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



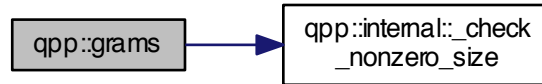
5.1.1.29 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::grams ( const std::vector< types::DynMat< typename Derived::Scalar > > & Vs )`

Here is the call graph for this function:



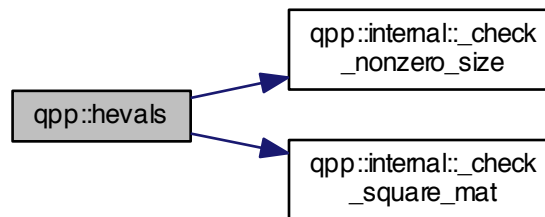
5.1.1.30 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::grams ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



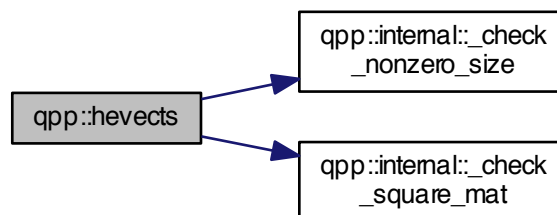
5.1.1.31 `template<typename Derived> types::dmat qpp::hevals ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



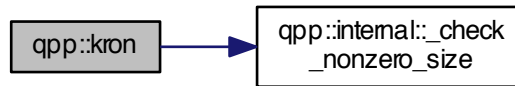
5.1.1.32 `template<typename Derived> types::cmat qpp::hevects ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



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

Here is the call graph for this function:



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

Here is the call graph for this function:



5.1.1.35 `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:



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

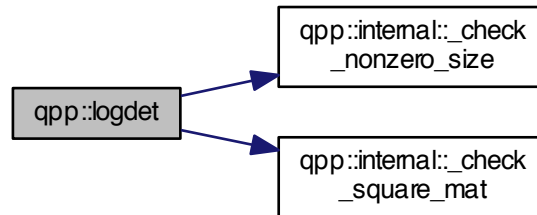
5.1.1.37 `template<typename Derived > Derived qpp::loadMATLABmatrix ( const std::string & mat_file, const std::string & var_name )`

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

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

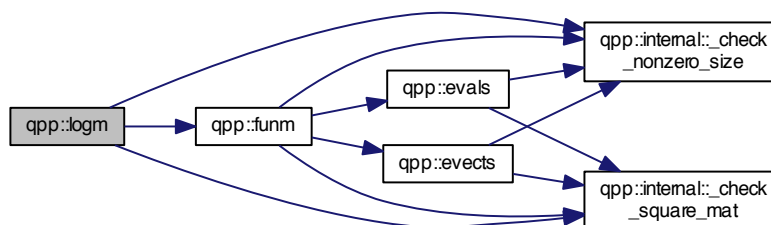
5.1.1.40 `template<typename Derived> Derived::Scalar qpp::logdet ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



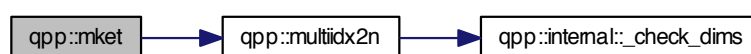
5.1.1.41 `template<typename Derived> types::cmat qpp::logm ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



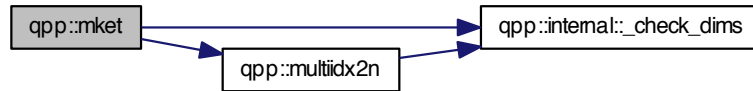
5.1.1.42 `types::ket qpp::mket ( const std::vector< size_t > & mask )`

Here is the call graph for this function:



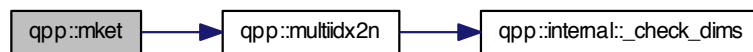
#### 5.1.1.43 `types::ket qpp::mket ( const std::vector< size_t > & mask, const std::vector< size_t > & dims )`

Here is the call graph for this function:



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

Here is the call graph for this function:



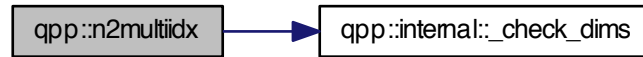
#### 5.1.1.45 `size_t qpp::multiidx2n ( const std::vector< size_t > & midx, const std::vector< size_t > & dims )`

Here is the call graph for this function:



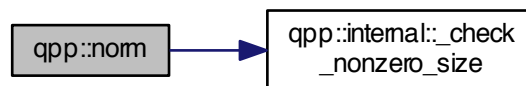
5.1.1.46 `std::vector<size_t> qpp::n2multiidx ( size_t n, const std::vector< size_t > & dims )`

Here is the call graph for this function:



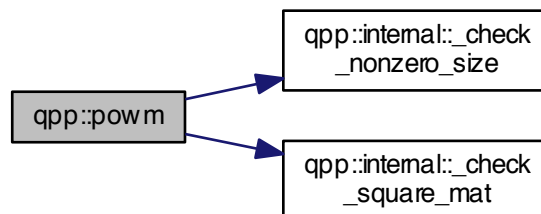
5.1.1.47 `template<typename Derived > double qpp::norm ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



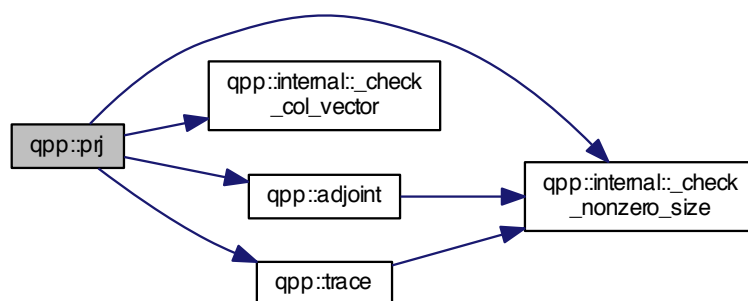
5.1.1.48 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::powm ( const Eigen::MatrixBase< Derived > & A, size_t n )`

Here is the call graph for this function:



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

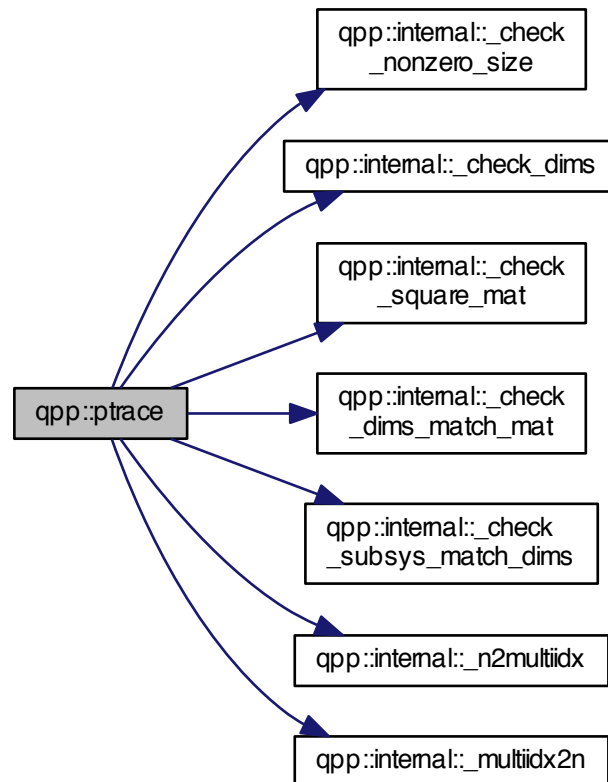
Here is the call graph for this function:





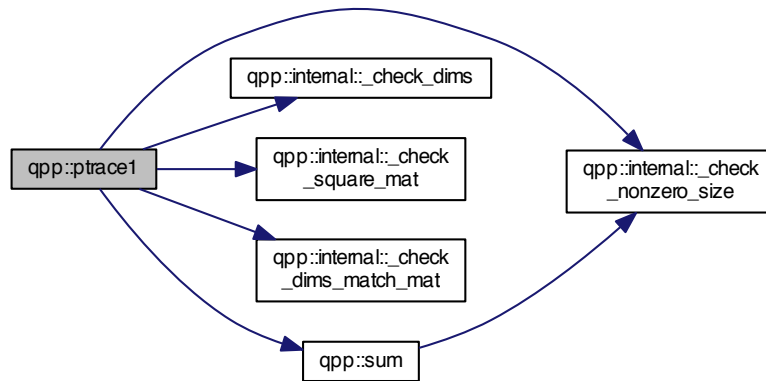
5.1.1.50 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::ptrace ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & subsys, const std::vector< size_t> & dims )`

Here is the call graph for this function:



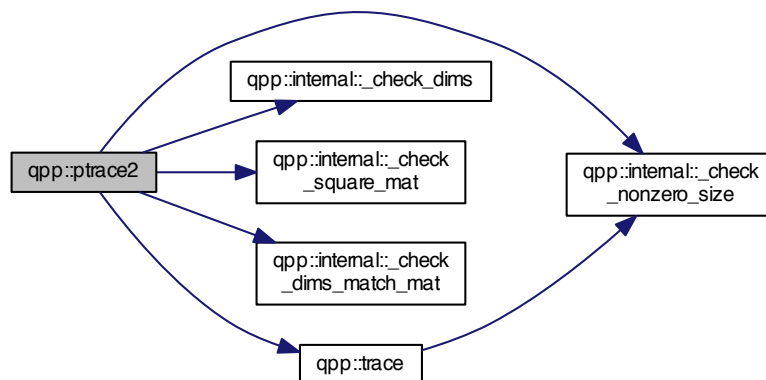
5.1.1.51 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::ptrace1 ( const Eigen::MatrixBase< Derived > & A, const std::vector< size_t > & dims )`

Here is the call graph for this function:



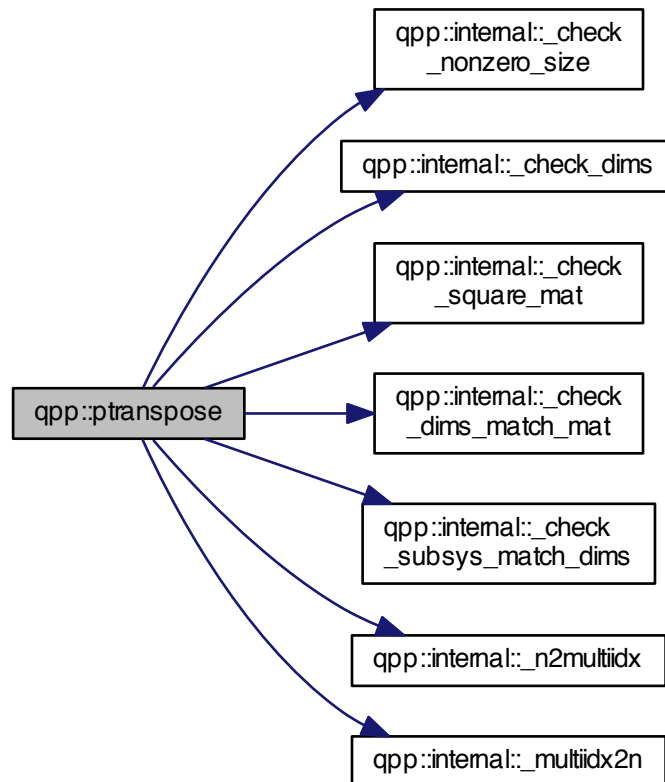
5.1.1.52 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::ptrace2 ( const Eigen::MatrixBase< Derived > & A, const std::vector< size_t > & dims )`

Here is the call graph for this function:



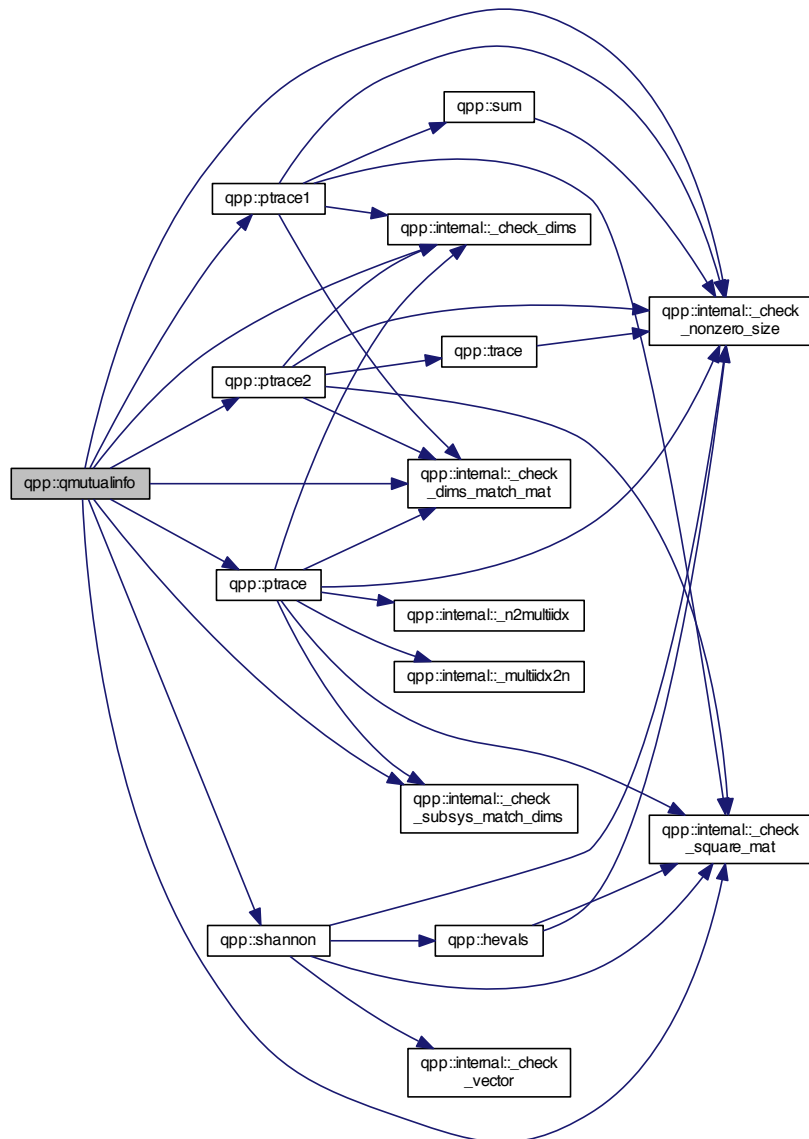
5.1.1.53 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::ptranspose ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & subsys, const std::vector< size_t> & dims )`

Here is the call graph for this function:



5.1.1.54 `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:



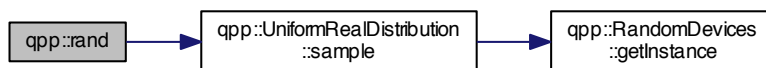
5.1.1.55 `template<typename Derived > Derived qpp::rand ( size_t rows, size_t cols, double a = 0, double b = 1 )`

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

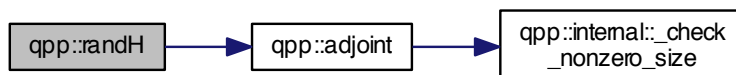
5.1.1.57 `template<> types::cmat qpp::rand ( size_t rows, size_t cols, double a, double b )`

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

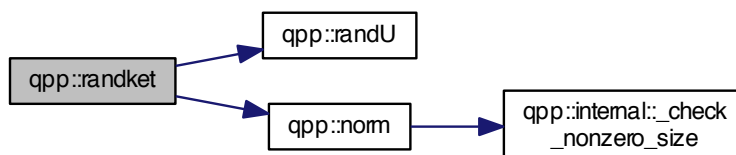
Here is the call graph for this function:

**5.1.1.59** `types::cmat qpp::randH ( size_t D )`

Here is the call graph for this function:

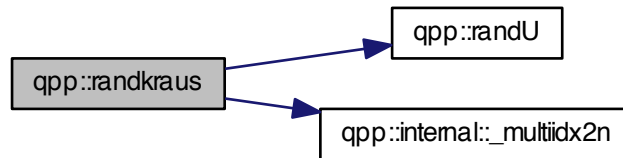
**5.1.1.60** `types::ket qpp::randket ( size_t D )`

Here is the call graph for this function:



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

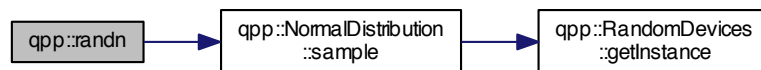
Here is the call graph for this function:



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

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

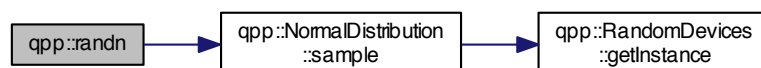
Here is the call graph for this function:



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

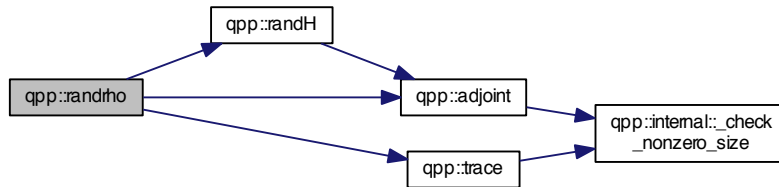
5.1.1.65 `double qpp::randn ( double mean = 0, double sigma = 1 )`

Here is the call graph for this function:



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

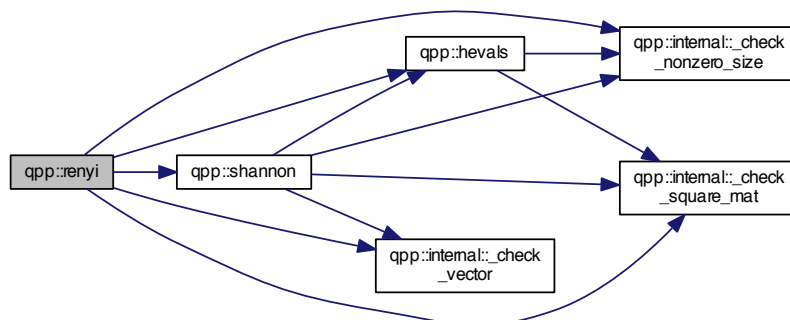
Here is the call graph for this function:

5.1.1.67 `types::cmat qpp::randU ( size_t D )`5.1.1.68 `types::cmat qpp::randV ( size_t Din, size_t Dout )`

Here is the call graph for this function:

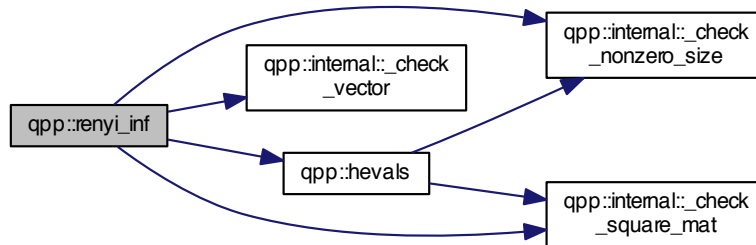
5.1.1.69 `template<typename Derived> double qpp::renyi ( const double alpha, const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



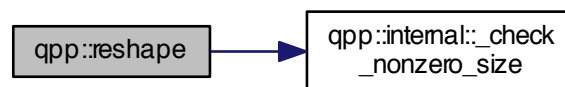
5.1.1.70 `template<typename Derived > double qpp::renyi_inf ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



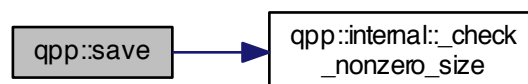
5.1.1.71 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::reshape ( const Eigen::MatrixBase< Derived > & A, size_t rows, size_t cols )`

Here is the call graph for this function:



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

Here is the call graph for this function:

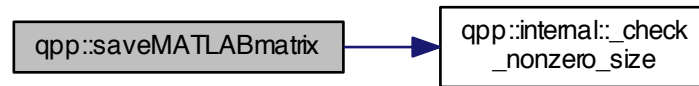


5.1.1.73 `template<typename Derived > void qpp::saveMATLABmatrix ( const Eigen::MatrixBase< Derived > & A, const std::string & mat_file, const std::string & var_name, const std::string & mode )`



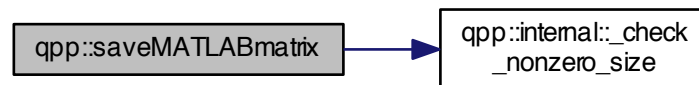
5.1.1.74 `template<> void qpp::saveMATLABmatrix ( const Eigen::MatrixBase< typename types::dmat > & A, const std::string & mat_file, const std::string & var_name, const std::string & mode )`

Here is the call graph for this function:



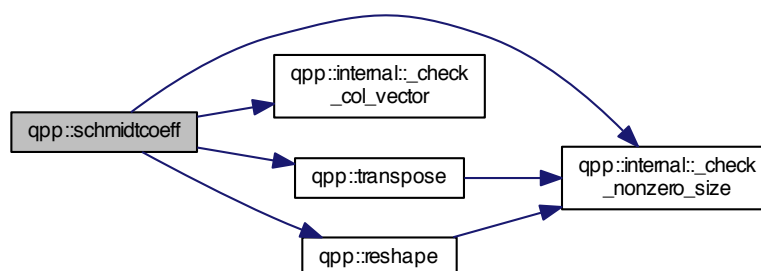
5.1.1.75 `template<> void qpp::saveMATLABmatrix ( const Eigen::MatrixBase< typename types::cmat > & A, const std::string & mat_file, const std::string & var_name, const std::string & mode )`

Here is the call graph for this function:



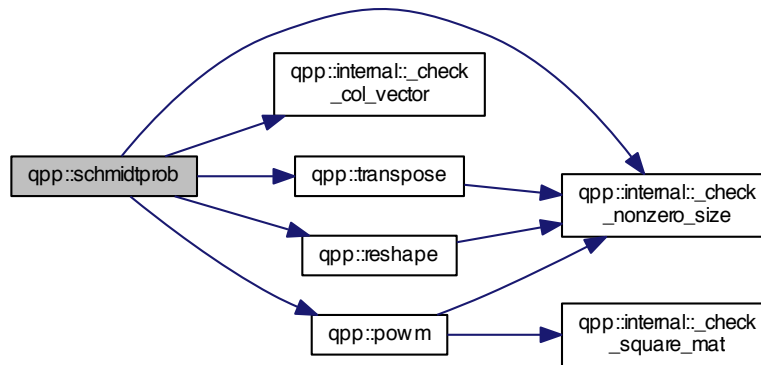
5.1.1.76 `template<typename Derived> types::cmat qpp::schmidtcoeff ( const Eigen::MatrixBase< Derived > & A, const std::vector< size_t > & dims )`

Here is the call graph for this function:



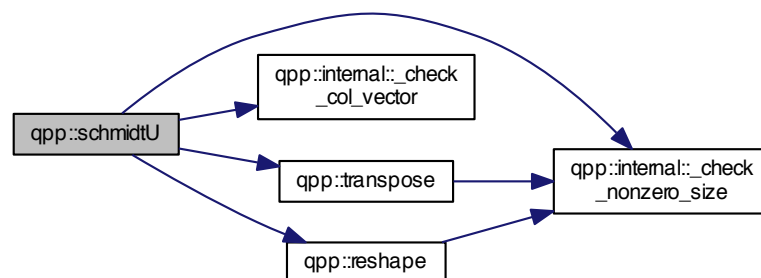
5.1.1.77 `template<typename Derived> types::cmat qpp::schmidtprob ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims )`

Here is the call graph for this function:



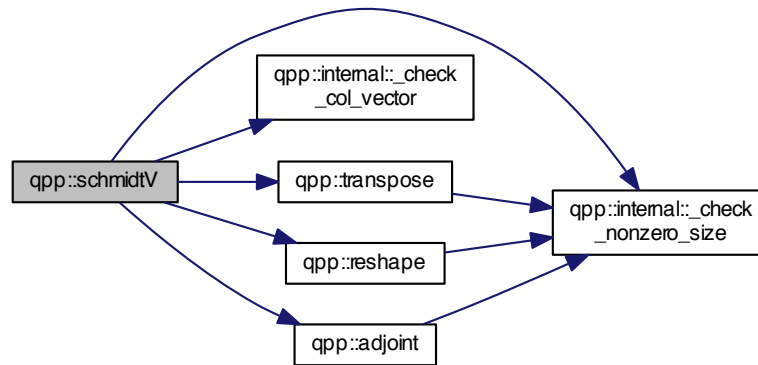
5.1.1.78 `template<typename Derived> types::cmat qpp::schmidtU ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims )`

Here is the call graph for this function:



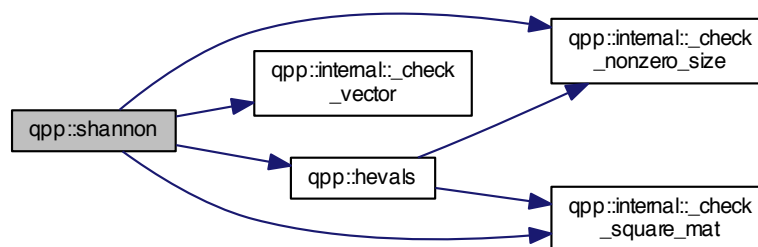
5.1.1.79 `template<typename Derived> types::cmat qpp::schmidtV ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims )`

Here is the call graph for this function:



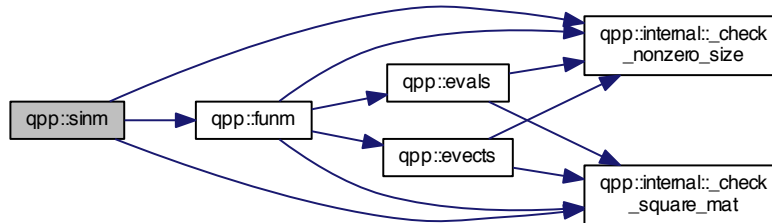
5.1.1.80 `template<typename Derived> double qpp::shannon ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



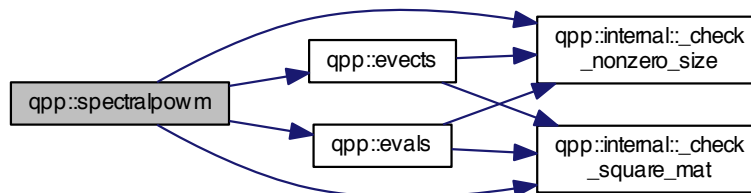
5.1.1.81 `template<typename Derived> types::cmat qpp::sinm ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



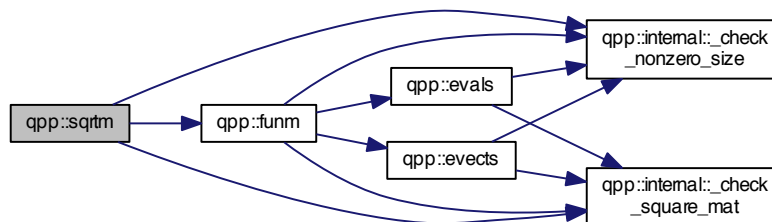
5.1.1.82 `template<typename Derived> types::cmat qpp::spectralpowm ( const Eigen::MatrixBase< Derived> & A, const types::cplx z )`

Here is the call graph for this function:



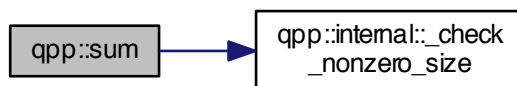
5.1.1.83 `template<typename Derived> types::cmat qpp::sqrtm ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



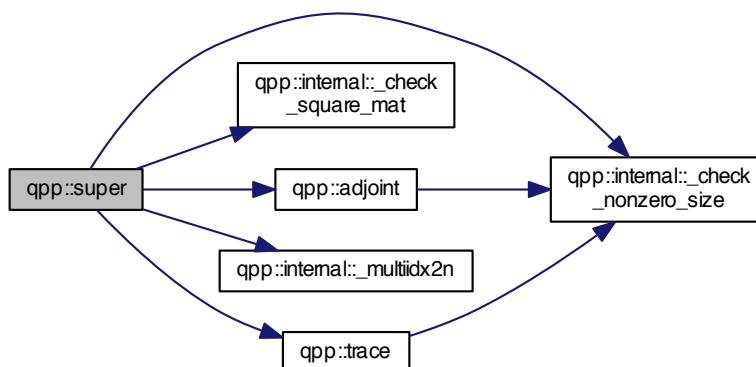
5.1.1.84 `template<typename Derived > Derived::Scalar qpp::sum ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



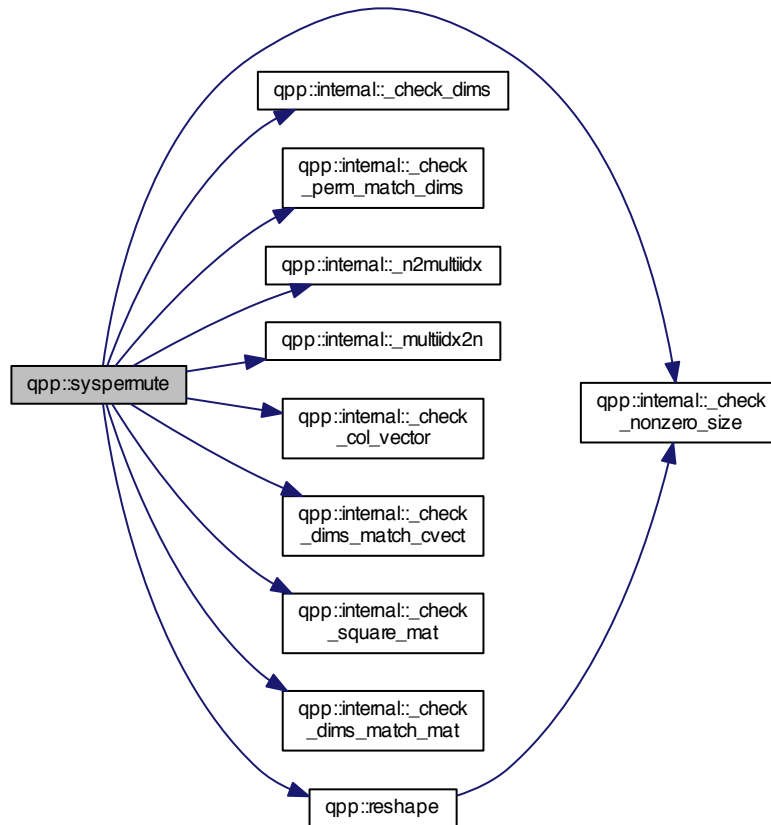
5.1.1.85 `types::cmat qpp::super ( const std::vector< types::cmat > & Ks )`

Here is the call graph for this function:



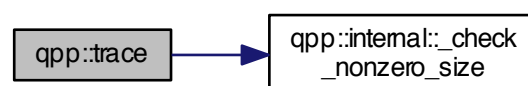
5.1.1.86 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::syspermute ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & perm, const std::vector< size_t> & dims )`

Here is the call graph for this function:



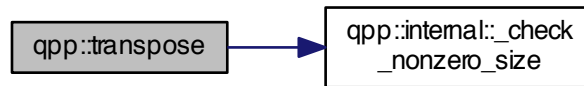
5.1.1.87 `template<typename Derived> Derived::Scalar qpp::trace ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



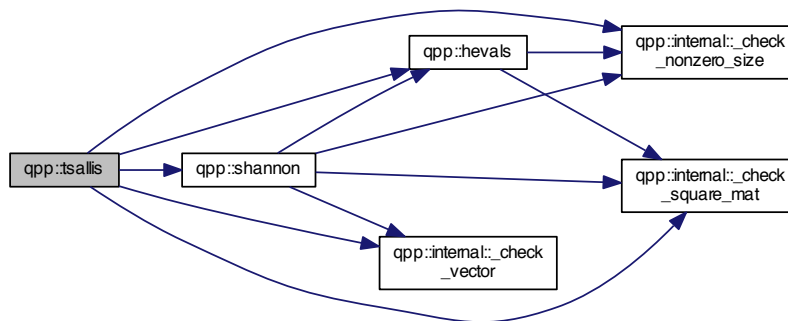
5.1.1.88 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::transpose ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



5.1.1.89 `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\\_match\\_dims](#) (const std::vector< size\_t > &perm, const std::vector< size\_t > &dims)



### 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_match_dims ( const std::vector< size_t > & perm, const std::vector< size_t > & dims )`

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

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

5.3.1.11 `bool qpp::internal::_check_subsys_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 `size_t qpp::internal::_multiidx2n ( const size_t * midx, size_t numdims, const size_t * dims )`

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

## 5.4 qpp::types Namespace Reference

### Typedefs

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

### 5.4.1 Typedef Documentation

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

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

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

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

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

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

## Chapter 6

# Class Documentation

### 6.1 qpp::DiscreteDistribution Class Reference

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

#### Public Member Functions

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

#### Protected Attributes

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

#### 6.1.1 Constructor & Destructor Documentation

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

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

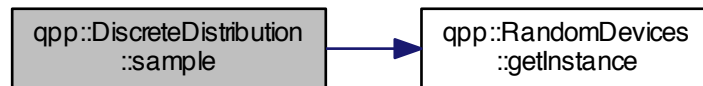
6.1.1.3 `qpp::DiscreteDistribution::DiscreteDistribution ( std::vector< double > weights )` [inline]

#### 6.1.2 Member Function Documentation

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

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

Here is the call graph for this function:



## 6.1.3 Member Data Documentation

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

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

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

## 6.2 `qpp::DiscreteDistributionAbsSquare` Class Reference

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

### Public Member Functions

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

### Protected Member Functions

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

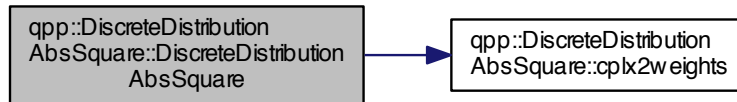
### Protected Attributes

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

## 6.2.1 Constructor & Destructor Documentation

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

Here is the call graph for this function:



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

Here is the call graph for this function:



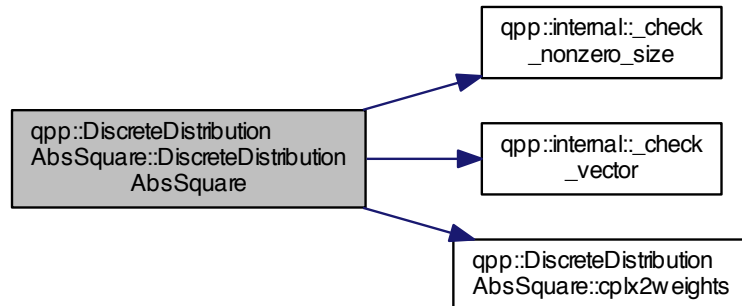
6.2.1.3 `qpp::DiscreteDistributionAbsSquare::DiscreteDistributionAbsSquare ( std::vector< types::cplx > amplitudes ) [inline]`

Here is the call graph for this function:



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

Here is the call graph for this function:



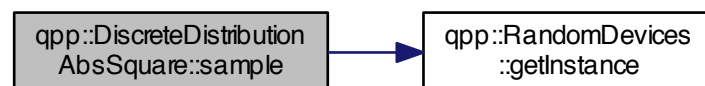
## 6.2.2 Member Function Documentation

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

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

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

Here is the call graph for this function:



## 6.2.3 Member Data Documentation

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

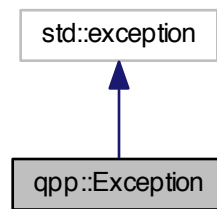
The documentation for this class was generated from the following file:

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

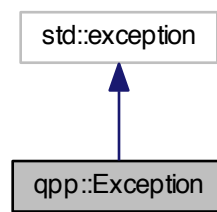
## 6.3 qpp::Exception Class Reference

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

Inheritance diagram for qpp::Exception:



Collaboration diagram for qpp::Exception:



## Public Types

- enum `Type` {  
`Type::UNKNOWN_EXCEPTION = 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_MISMATCH_DIMS`, `Type::NOT_QUBIT_GATE`, `Type::NOT_QUBIT_SUBSYS`, `Type::NOT_BIPARTITE`,  
`Type::OUT_OF_RANGE`, `Type::UNDEFINED_TYPE`, `Type::TYPE_MISMATCH`, `Type::CUSTOM_EXCEPTION` }

## Public Member Functions

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

## Private Member Functions

- `std::string _construct_exception_msg ()`

## Private Attributes

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

## 6.3.1 Member Enumeration Documentation

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

Enumerator

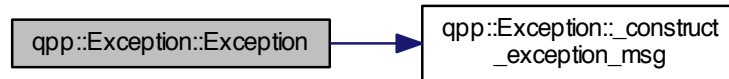
***UNKNOWN\_EXCEPTION***  
***ZERO\_SIZE***  
***MATRIX\_NOT\_SQUARE***  
***MATRIX\_NOT\_CVECTOR***  
***MATRIX\_NOT\_RVECTOR***  
***MATRIX\_NOT\_VECTOR***  
***MATRIX\_NOT\_SQUARE\_OR\_CVECTOR***  
***MATRIX\_NOT\_SQUARE\_OR\_RVECTOR***  
***MATRIX\_NOT\_SQUARE\_OR\_VECTOR***  
***DIMS\_INVALID***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



### 6.3.2 Constructor & Destructor Documentation

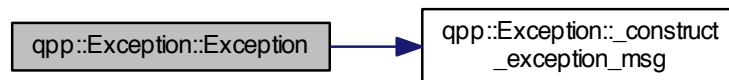
6.3.2.1 `qpp::Exception::Exception ( const std::string & where, const Type & type )` `[inline]`

Here is the call graph for this function:



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

Here is the call graph for this function:



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

### 6.3.3 Member Function Documentation

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

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

### 6.3.4 Member Data Documentation

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

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

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

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

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

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

## 6.4 qpp::Gates Class Reference

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

### Public Member Functions

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

### Static Public Member Functions

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

### Public Attributes

- [types::cmat Id2](#)
- [types::cmat H](#)
- [types::cmat X](#)
- [types::cmat Y](#)
- [types::cmat Z](#)
- [types::cmat S](#)
- [types::cmat T](#)
- [types::cmat CNOTab](#)
- [types::cmat CZ](#)
- [types::cmat CNOTba](#)
- [types::cmat SWAP](#)
- [types::cmat TOF](#)
- [types::cmat FRED](#)

### Private Member Functions

- [Gates](#) ()

#### 6.4.1 Constructor & Destructor Documentation

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

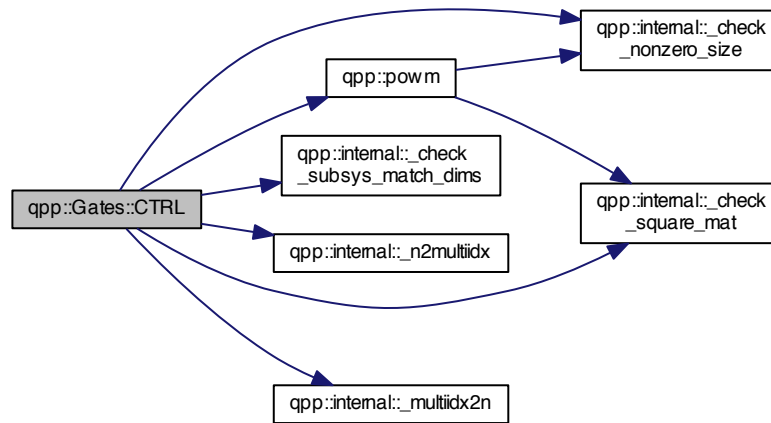
6.4.1.2 `qpp::Gates::Gates ( const Gates & )` [delete]

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

#### 6.4.2 Member Function Documentation

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

Here is the call graph for this function:



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

Here is the call graph for this function:



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

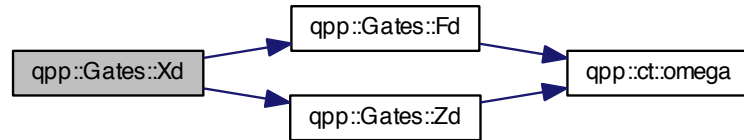
6.4.2.4 `types::cmat qpp::Gates::Id ( size_t D ) const` [inline]

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

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

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

Here is the call graph for this function:



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

Here is the call graph for this function:



### 6.4.3 Member Data Documentation

#### 6.4.3.1 `types::cmat qpp::Gates::CNOTab`

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

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

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

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

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

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

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

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

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

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

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

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

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

- `include/classes/gates.h`

## 6.5 qpp::NormalDistribution Class Reference

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

### Public Member Functions

- `NormalDistribution` (double *mean*=0, double *sigma*=1)
- double `sample` ()

### Protected Attributes

- `std::normal_distribution _d`

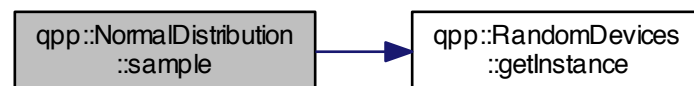
### 6.5.1 Constructor & Destructor Documentation

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

### 6.5.2 Member Function Documentation

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

Here is the call graph for this function:



### 6.5.3 Member Data Documentation

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

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

- `include/classes/stat.h`

## 6.6 qpp::Qudit Class Reference

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

### Public Member Functions

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

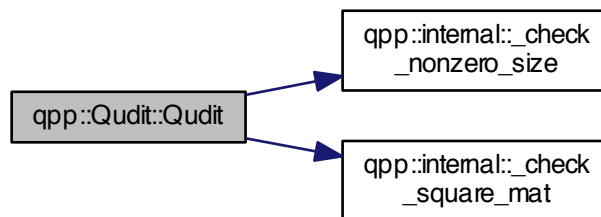
### Private Attributes

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

### 6.6.1 Constructor & Destructor Documentation

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

Here is the call graph for this function:



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

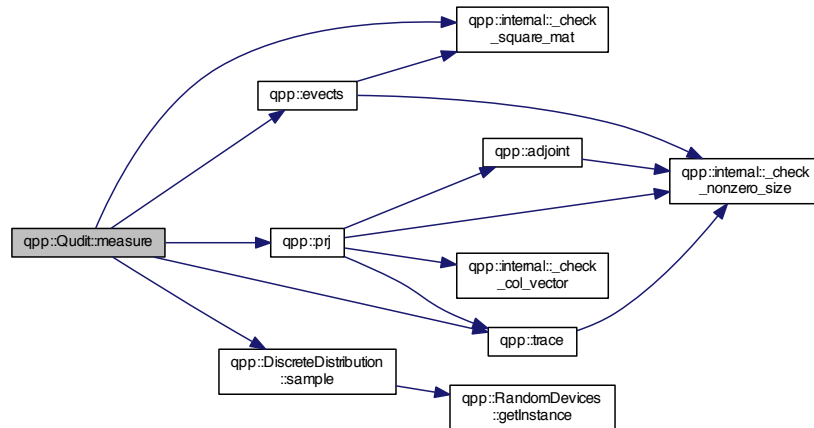
### 6.6.2 Member Function Documentation

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

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

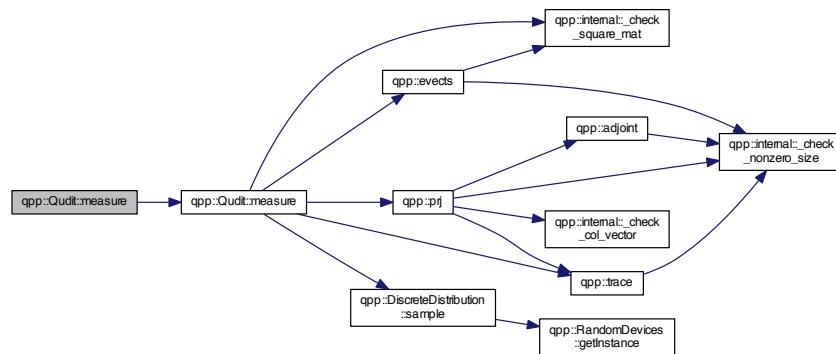
### 6.6.2.3 `size_t qpp::Qudit::measure ( const types::cmat & U, bool destructive = false ) [inline]`

Here is the call graph for this function:



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

Here is the call graph for this function:



## 6.6.3 Member Data Documentation

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

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

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

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

## 6.7 qpp::RandomDevices Class Reference

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

### Public Member Functions

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

### Static Public Member Functions

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

### Public Attributes

- std::random\_device [\\_rd](#)
- std::mt19937 [\\_rng](#)

### Private Member Functions

- [RandomDevices](#) ()

### 6.7.1 Constructor & Destructor Documentation

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

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

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

### 6.7.2 Member Function Documentation

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

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

### 6.7.3 Member Data Documentation

6.7.3.1 std::random\_device [qpp::RandomDevices::\\_rd](#)

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

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

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

## 6.8 qpp::States Class Reference

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



## Public Member Functions

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

## Static Public Member Functions

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

## Public Attributes

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

## Private Member Functions

- [States](#) ()

### 6.8.1 Constructor & Destructor Documentation

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

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

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

### 6.8.2 Member Function Documentation

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

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

### 6.8.3 Member Data Documentation

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## 6.9 qpp::Timer Class Reference

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

### Public Member Functions

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

### Protected Attributes

- std::chrono::high\_resolution\_clock::time\_point [\\_start](#)
- std::chrono::high\_resolution\_clock::time\_point [\\_end](#)

### Friends

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

#### 6.9.1 Constructor & Destructor Documentation

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

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

#### 6.9.2 Member Function Documentation

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

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

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

#### 6.9.3 Friends And Related Function Documentation

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

#### 6.9.4 Member Data Documentation

6.9.4.1 [std::chrono::high\\_resolution\\_clock::time\\_point qpp::Timer::\\_end](#) [\[protected\]](#)

6.9.4.2 [std::chrono::high\\_resolution\\_clock::time\\_point qpp::Timer::\\_start](#) [\[protected\]](#)

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

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

## 6.10 qpp::UniformRealDistribution Class Reference

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

## Public Member Functions

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

## Protected Attributes

- std::uniform\_real\_distribution [\\_d](#)

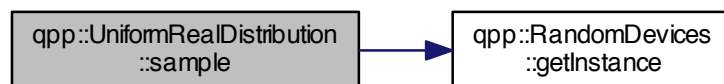
### 6.10.1 Constructor & Destructor Documentation

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

### 6.10.2 Member Function Documentation

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

Here is the call graph for this function:



### 6.10.3 Member Data Documentation

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

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

- `include/classes/stat.h`

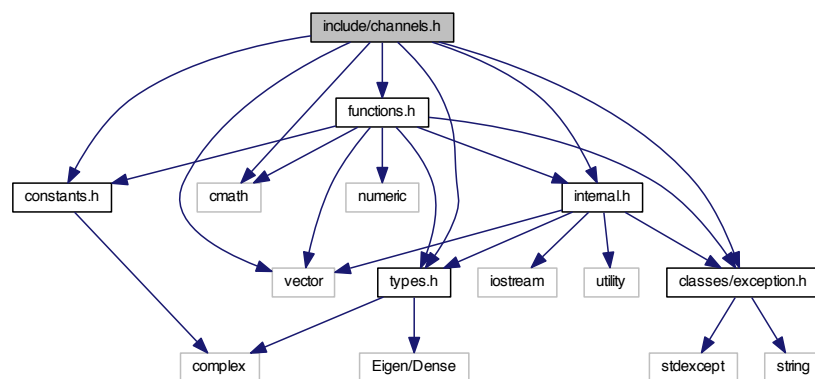
## Chapter 7

# File Documentation

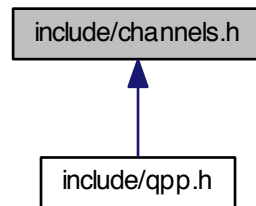
### 7.1 include/channels.h File Reference

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

Include dependency graph for channels.h:



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



## Namespaces

- [qpp](#)

## Functions

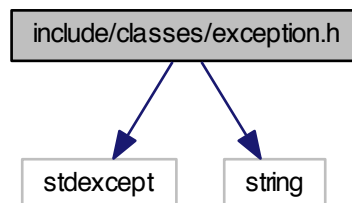
- `template<typename Derived >`  
`types::cmat qpp::channel (const Eigen::MatrixBase< Derived > &rho, const std::vector< types::cmat > &Ks)`
- `types::cmat qpp::super (const std::vector< types::cmat > &Ks)`
- `types::cmat qpp::choi (const std::vector< types::cmat > &Ks)`
- `std::vector< types::cmat >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

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

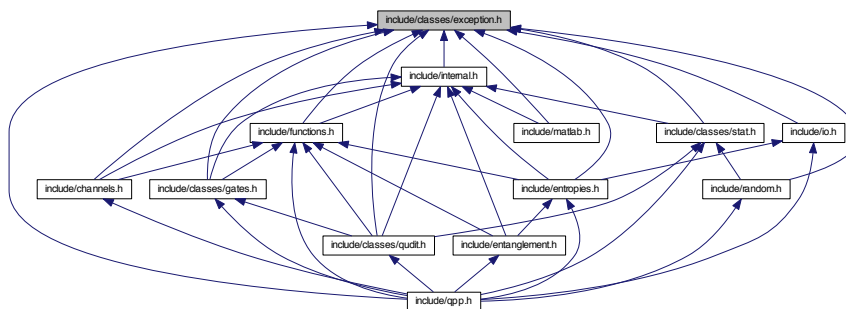
```
#include <stdexcept>
```

```
#include <string>
```

Include dependency graph for exception.h:



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



## Classes

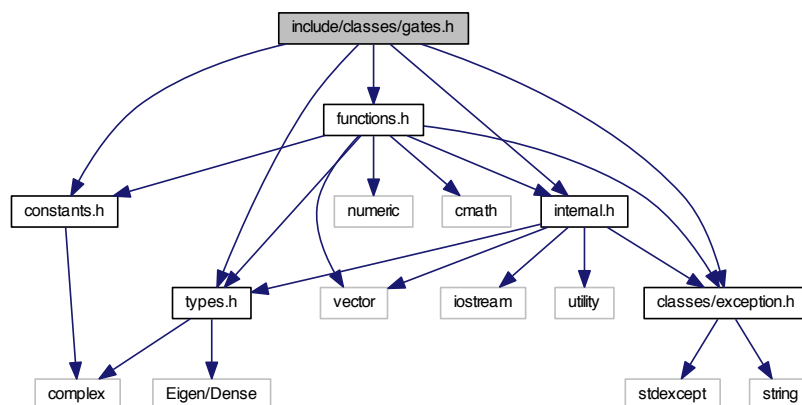
- class `qpp::Exception`

## Namespaces

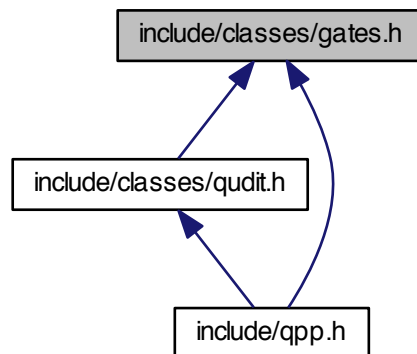
- `qpp`

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

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



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



## Classes

- class [qpp::Gates](#)

## Namespaces

- [qpp](#)

## Functions

- `template<typename Derived1 , typename Derived2 >`  
`types::DynMat< typename`  
`Derived1::Scalar > qpp::gate (const Eigen::MatrixBase< Derived2 > &state, const Eigen::MatrixBase< De-`  
`derived1 > &A, const std::vector< size_t > &subsys, const std::vector< size_t > &dims)`

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

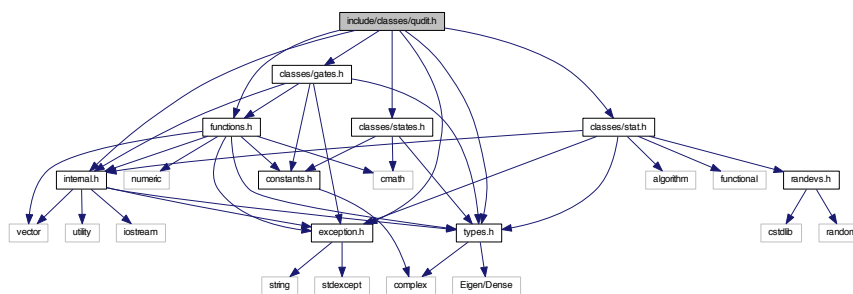
```

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

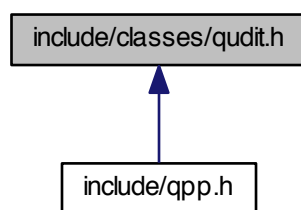
```



Include dependency graph for qudit.h:



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



## Classes

- class [qpp::Qudit](#)

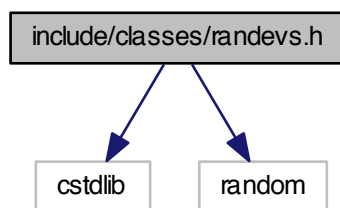
## Namespaces

- [qpp](#)

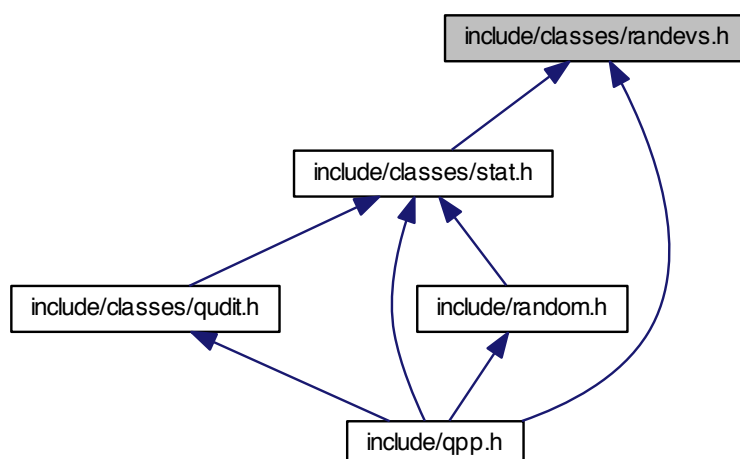
## 7.5 include/classes/randevs.h File Reference

```
#include <cstdlib>
#include <random>
```

Include dependency graph for randevs.h:



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



## Classes

- class [qpp::RandomDevices](#)

## Namespaces

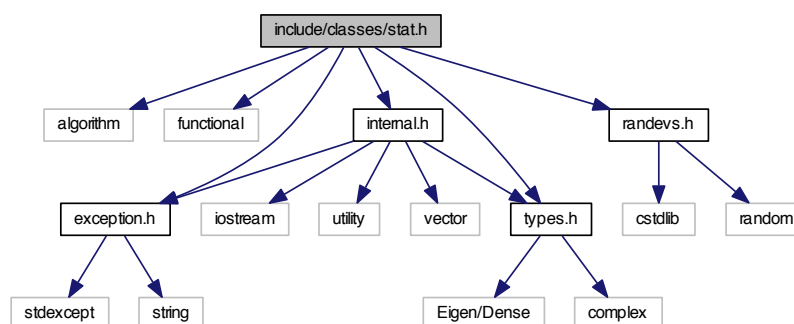
- [qpp](#)

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

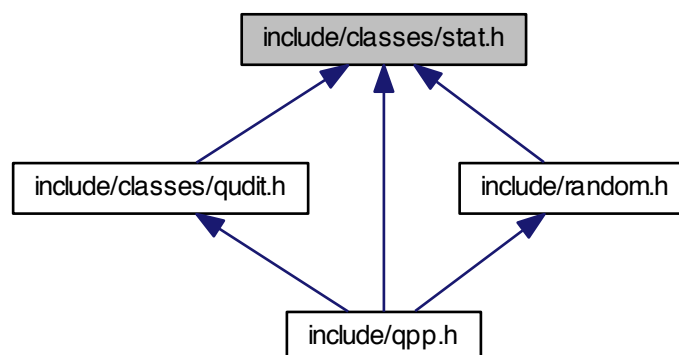
```
#include <algorithm>
```

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

Include dependency graph for stat.h:



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



## Classes

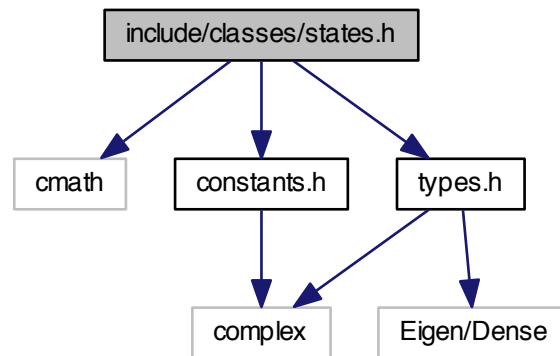
- class [qpp::NormalDistribution](#)
- class [qpp::UniformRealDistribution](#)
- class [qpp::DiscreteDistribution](#)
- class [qpp::DiscreteDistributionAbsSquare](#)

## Namespaces

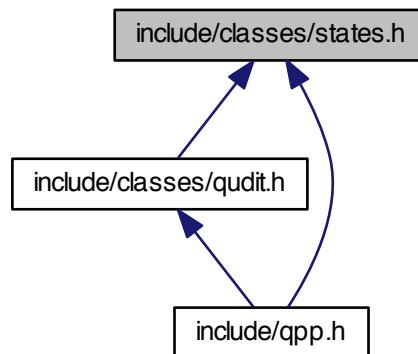
- [qpp](#)

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

```
#include <cmath>
#include "constants.h"
#include "types.h"
Include dependency graph for states.h:
```



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



### Classes

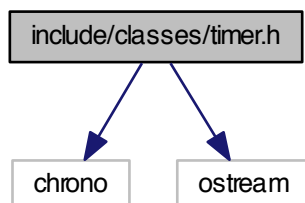
- class [qpp::States](#)

### Namespaces

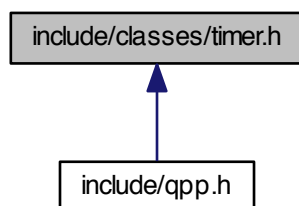
- [qpp](#)

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

```
#include <chrono>
#include <ostream>
Include dependency graph for timer.h:
```



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



### Classes

- class `qpp::Timer`

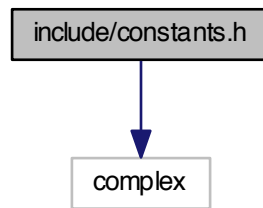
### Namespaces

- `qpp`

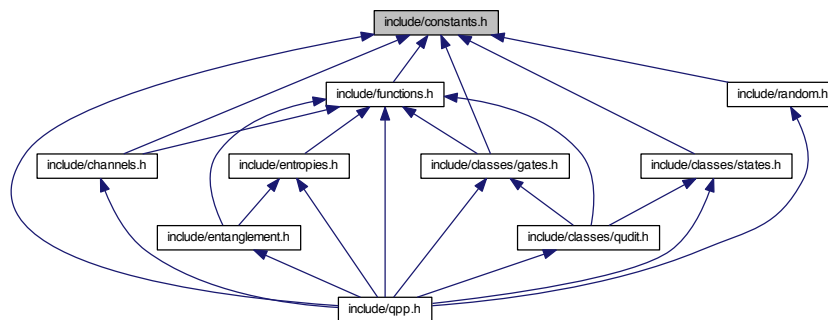
## 7.9 include/constants.h File Reference

```
#include <complex>
```

Include dependency graph for constants.h:



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



## Namespaces

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

## Functions

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

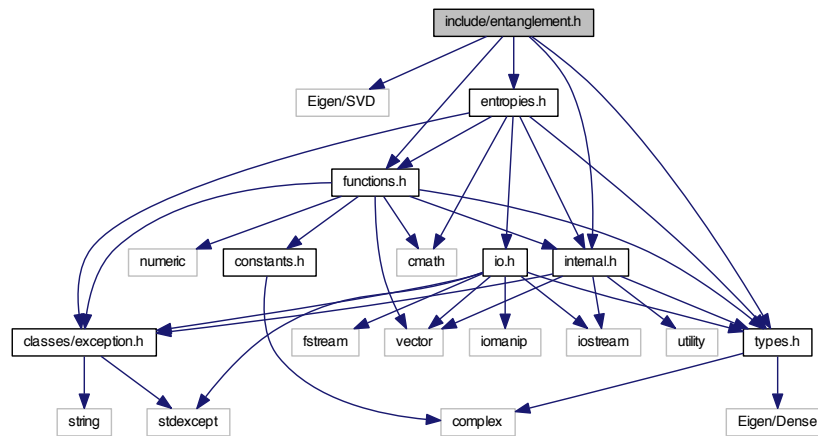
## Variables

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

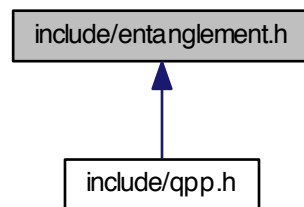
## 7.10 include/entanglement.h File Reference

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

Include dependency graph for entanglement.h:



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



### Namespaces

- [qpp](#)

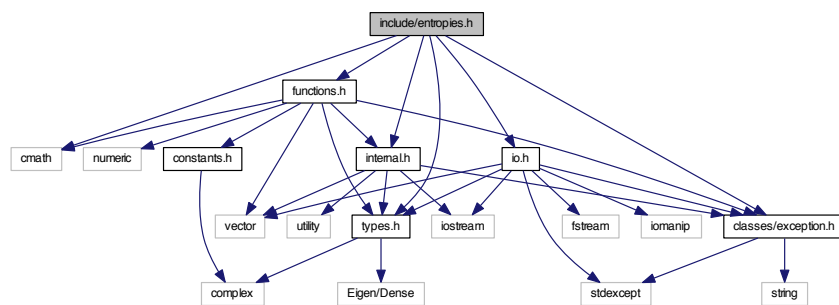
### Functions

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

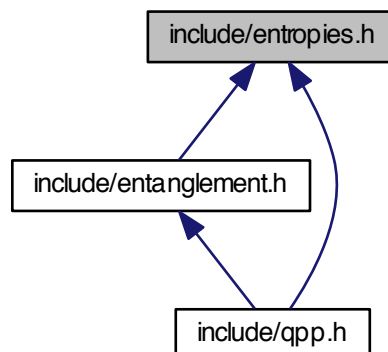
- `template<typename Derived >`  
`types::cmat qpp::schmidtV (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &dims)`
- `template<typename Derived >`  
`types::cmat qpp::schmidtprob (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &dims)`
- `template<typename Derived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

## 7.11 include/entropies.h File Reference

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



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





## Namespaces

- [qpp](#)

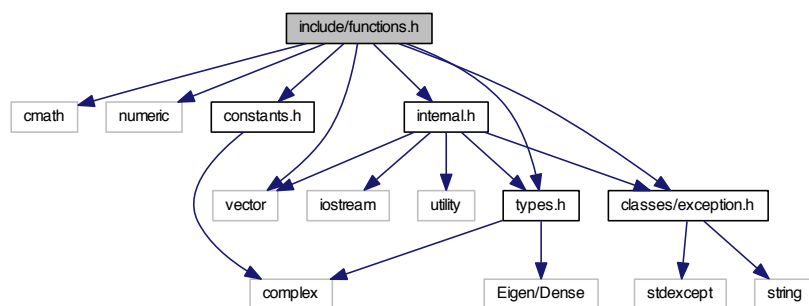
## Functions

- `template<typename Derived >`  
double [qpp::shannon](#) (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
double [qpp::renyi](#) (const double alpha, const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
double [qpp::renyi\\_inf](#) (const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
double [qpp::tsallis](#) (const double alpha, const Eigen::MatrixBase< Derived > &A)
- `template<typename Derived >`  
double [qpp::qmutualinfo](#) (const Eigen::MatrixBase< Derived > &A, const std::vector< size\_t > &subsys, const std::vector< size\_t > &dims)

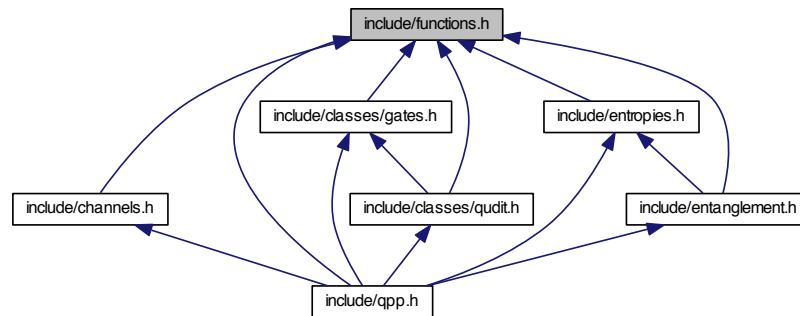
## 7.12 include/functions.h File Reference

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

Include dependency graph for functions.h:



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



## Namespaces

- [qpp](#)

## Functions

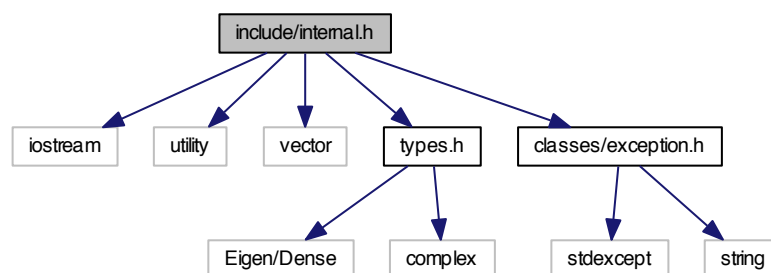
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::transpose (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::conjugate (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::adjoint (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`Derived::Scalar qpp::trace (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`Derived::Scalar qpp::det (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`Derived::Scalar qpp::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 Derived1 , typename Derived2 >`  
`types::DynMat< typename`  
`Derived1::Scalar > qpp::kron (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase< De-`  
`rived2 > &B)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::kronlist (const std::vector< types::DynMat< typename Derived::Scalar > > &As)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::kronpow (const Eigen::MatrixBase< Derived > &A, size_t n)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::reshape (const Eigen::MatrixBase< Derived > &A, size_t rows, size_t cols)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::syspermute (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t >`  
`&perm, const std::vector< size_t > &dims)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::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-`  
`rived2 > &B)`

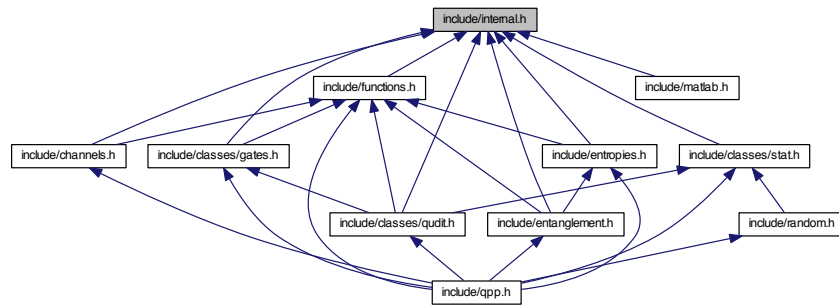
- `template<typename Derived1 , typename Derived2 >`  
`types::DynMat< typename`  
`Derived1::Scalar > qpp::anticomm (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase<`  
`Derived2 > &B)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::prj (const Eigen::MatrixBase< 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< types::DynMat< typename Derived::Scalar > > &Vs)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::grams (const Eigen::MatrixBase< Derived > &A)`
- `std::vector< size_t > qpp::n2multiidx (size_t n, const std::vector< size_t > &dims)`
- `size_t qpp::multiidx2n (const std::vector< size_t > &midx, const std::vector< size_t > &dims)`
- `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)`

## 7.13 include/internal.h File Reference

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



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



## Namespaces

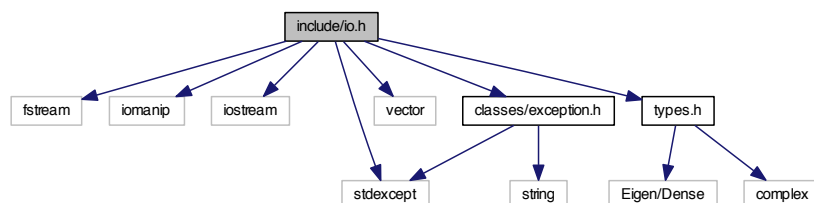
- [qpp](#)
- [qpp::internal](#)

## Functions

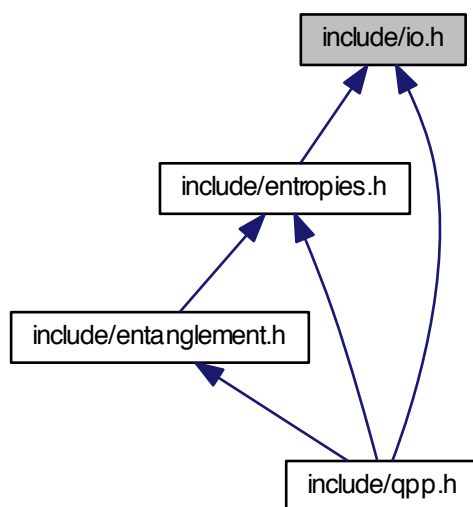
- void [qpp::internal::\\_n2multiidx](#) (size\_t n, size\_t numdims, const size\_t \*dims, size\_t \*result)
- size\_t [qpp::internal::\\_multiidx2n](#) (const size\_t \*midx, size\_t numdims, const size\_t \*dims)
- template<typename Derived >  
bool [qpp::internal::\\_check\\_square\\_mat](#) (const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

## 7.14 include/io.h File Reference

```
#include <fstream>
#include <iomanip>
#include <iostream>
#include <stdexcept>
#include <vector>
#include "types.h"
#include "classes/exception.h"
Include dependency graph for io.h:
```



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



### Namespaces

- [qpp](#)

### Functions

- `template<typename T>`

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

- `template<typename T >`  

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

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

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

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

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

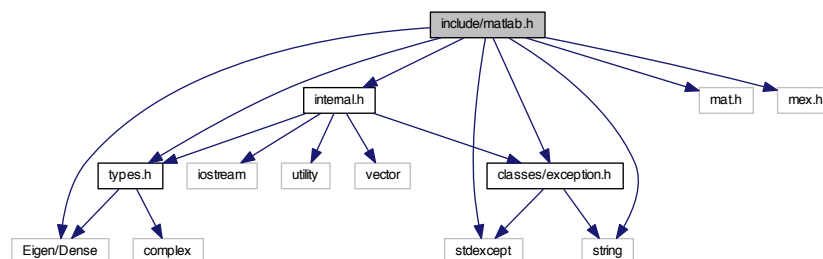
```
void qpp::save (const Eigen::MatrixBase< Derived > &A, const std::string &fname)
```
- `template<typename Derived >`  

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

## 7.15 include/matlab.h File Reference

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

Include dependency graph for matlab.h:



## Namespaces

- [qpp](#)

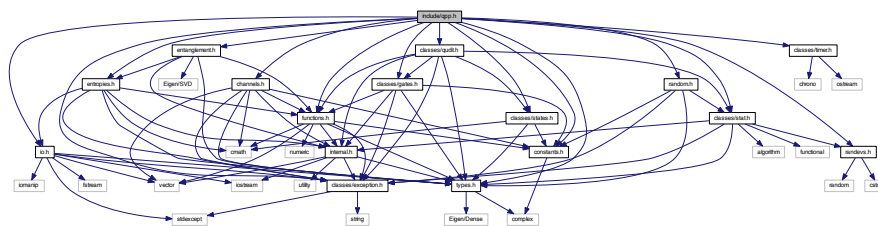
## Functions

- template<typename Derived >  
Derived [qpp::loadMATLABmatrix](#) (const std::string &mat\_file, const std::string &var\_name)
- template<>  
types::dmat [qpp::loadMATLABmatrix](#) (const std::string &mat\_file, const std::string &var\_name)
- template<>  
types::cmat [qpp::loadMATLABmatrix](#) (const std::string &mat\_file, const std::string &var\_name)
- template<typename Derived >  
void [qpp::saveMATLABmatrix](#) (const Eigen::MatrixBase< Derived > &A, const std::string &mat\_file, const std::string &var\_name, const std::string &mode)
- template<>  
void [qpp::saveMATLABmatrix](#) (const Eigen::MatrixBase< typename types::dmat > &A, const std::string &mat\_file, const std::string &var\_name, const std::string &mode)
- template<>  
void [qpp::saveMATLABmatrix](#) (const Eigen::MatrixBase< typename types::cmat > &A, const std::string &mat\_file, const std::string &var\_name, const std::string &mode)

## 7.16 include/qpp.h File Reference

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

Include dependency graph for qpp.h:



## Namespaces

- [qpp](#)

## Variables

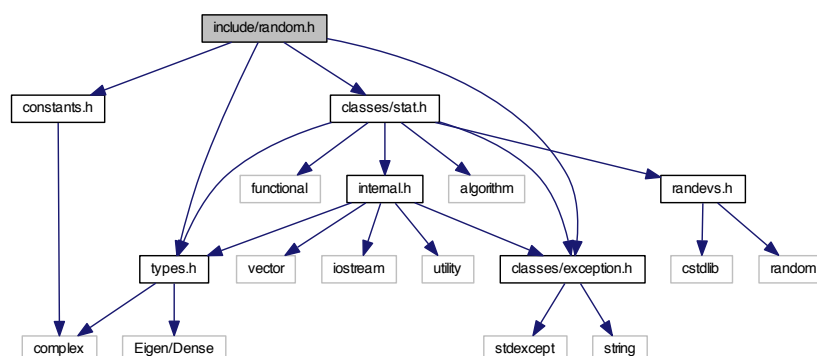
- RandomDevices & [qpp::rdevs](#) = RandomDevices::getInstance()



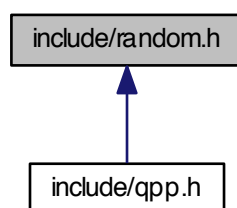
- const Gates & [qpp::gt](#) = Gates::getInstance()
- const States & [qpp::st](#) = States::getInstance()

## 7.17 include/random.h File Reference

```
#include "constants.h"
#include "types.h"
#include "classes/exception.h"
#include "classes/stat.h"
Include dependency graph for random.h:
```



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



### Namespaces

- [qpp](#)

### Functions

- template<typename Derived >  
Derived [qpp::rand](#) (size\_t rows, size\_t cols, double a=0, double b=1)
- template<>  
types::dmat [qpp::rand](#) (size\_t rows, size\_t cols, double a, double b)

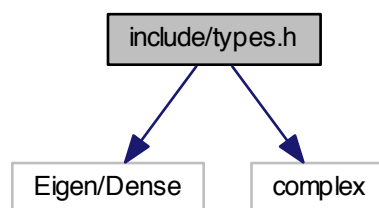
- `template<>`  
`types::cmat qpp::rand (size_t rows, size_t cols, double a, double b)`
- `double qpp::rand (double a=0, double b=1)`
- `template<typename Derived >`  
`Derived qpp::randn (size_t rows, size_t cols, double mean=0, double sigma=1)`
- `template<>`  
`types::dmat qpp::randn (size_t rows,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

## 7.18 include/types.h File Reference

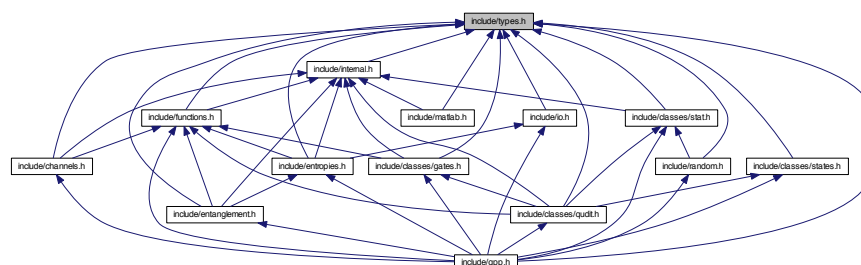
```
#include <Eigen/Dense>
```

```
#include <complex>
```

Include dependency graph for types.h:



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



## Namespaces

- [qpp](#)

- [qpp::types](#)

## Typedefs

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