

qpp  
0.1

Generated by Doxygen 1.8.5

Wed Apr 16 2014 23:32:52



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

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

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

5.2.2.5	maxn	45
5.2.2.6	pi	45
5.3	qpp::internal Namespace Reference	45
5.3.1	Function Documentation	46
5.3.1.1	_check_col_vector	46
5.3.1.2	_check_dims	46
5.3.1.3	_check_dims_match_cvect	46
5.3.1.4	_check_dims_match_mat	46
5.3.1.5	_check_dims_match_rvect	46
5.3.1.6	_check_eq_dims	46
5.3.1.7	_check_nonzero_size	46
5.3.1.8	_check_perm_match_dims	46
5.3.1.9	_check_row_vector	46
5.3.1.10	_check_square_mat	46
5.3.1.11	_check_subsys_match_dims	46
5.3.1.12	_check_vector	46
5.3.1.13	_multiidx2n	46
5.3.1.14	_n2multiidx	46
5.4	qpp::types Namespace Reference	46
5.4.1	Typedef Documentation	46
5.4.1.1	bra	46
5.4.1.2	cmat	46
5.4.1.3	cplx	47
5.4.1.4	dmat	47
5.4.1.5	DynMat	47
5.4.1.6	ket	47
<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	_d	50
6.2	qpp::DiscreteDistributionAbsSquare Class Reference	50
6.2.1	Constructor & Destructor Documentation	50

6.2.1.1	DiscreteDistributionAbsSquare	51
6.2.1.2	DiscreteDistributionAbsSquare	51
6.2.1.3	DiscreteDistributionAbsSquare	51
6.2.1.4	DiscreteDistributionAbsSquare	52
6.2.2	Member Function Documentation	52
6.2.2.1	cplx2weights	52
6.2.2.2	probabilities	52
6.2.2.3	sample	52
6.2.3	Member Data Documentation	52
6.2.3.1	_d	52
6.3	qpp::Exception Class Reference	52
6.3.1	Member Enumeration Documentation	54
6.3.1.1	Type	54
6.3.2	Constructor & Destructor Documentation	55
6.3.2.1	Exception	55
6.3.2.2	Exception	55
6.3.2.3	~Exception	55
6.3.3	Member Function Documentation	55
6.3.3.1	_construct_exception_msg	55
6.3.3.2	what	55
6.3.4	Member Data Documentation	55
6.3.4.1	_custom	55
6.3.4.2	_msg	55
6.3.4.3	_type	55
6.3.4.4	_where	55
6.4	qpp::Gates Class Reference	56
6.4.1	Constructor & Destructor Documentation	57
6.4.1.1	Gates	57
6.4.1.2	Gates	57
6.4.1.3	~Gates	57
6.4.2	Member Function Documentation	57
6.4.2.1	CTRL	57
6.4.2.2	Fd	58
6.4.2.3	getInstance	58
6.4.2.4	Id	58
6.4.2.5	operator=	58
6.4.2.6	Rtheta	58
6.4.2.7	Xd	58
6.4.2.8	Zd	58
6.4.3	Member Data Documentation	59

6.4.3.1	b00	59
6.4.3.2	b01	59
6.4.3.3	b10	59
6.4.3.4	b11	59
6.4.3.5	CNOTab	59
6.4.3.6	CNOTba	59
6.4.3.7	CS	59
6.4.3.8	CZ	59
6.4.3.9	FRED	59
6.4.3.10	GHZ	59
6.4.3.11	H	59
6.4.3.12	Id2	59
6.4.3.13	pb00	59
6.4.3.14	pb01	59
6.4.3.15	pb10	59
6.4.3.16	pb11	59
6.4.3.17	pGHZ	59
6.4.3.18	pW	59
6.4.3.19	px0	59
6.4.3.20	px1	59
6.4.3.21	py0	59
6.4.3.22	py1	59
6.4.3.23	pz0	59
6.4.3.24	pz1	59
6.4.3.25	S	59
6.4.3.26	SWAP	59
6.4.3.27	T	59
6.4.3.28	TOF	60
6.4.3.29	W	60
6.4.3.30	X	60
6.4.3.31	x0	60
6.4.3.32	x1	60
6.4.3.33	Y	60
6.4.3.34	y0	60
6.4.3.35	y1	60
6.4.3.36	Z	60
6.4.3.37	z0	60
6.4.3.38	z1	60
6.5	qpp::NormalDistribution Class Reference	60
6.5.1	Constructor & Destructor Documentation	60



6.5.1.1	NormalDistribution	60
6.5.2	Member Function Documentation	60
6.5.2.1	sample	61
6.5.3	Member Data Documentation	61
6.5.3.1	_d	61
6.6	qpp::Qudit Class Reference	61
6.6.1	Constructor & Destructor Documentation	61
6.6.1.1	Qudit	62
6.6.1.2	~Qudit	62
6.6.2	Member Function Documentation	62
6.6.2.1	getD	62
6.6.2.2	getRho	62
6.6.2.3	measure	62
6.6.2.4	measure	63
6.6.3	Member Data Documentation	63
6.6.3.1	_D	63
6.6.3.2	_rho	63
6.7	qpp::RandomDevices Class Reference	63
6.7.1	Constructor & Destructor Documentation	64
6.7.1.1	RandomDevices	64
6.7.1.2	RandomDevices	64
6.7.1.3	~RandomDevices	64
6.7.2	Member Function Documentation	64
6.7.2.1	getInstance	64
6.7.2.2	operator=	64
6.7.3	Member Data Documentation	64
6.7.3.1	_rd	64
6.7.3.2	_rng	64
6.8	qpp::Timer Class Reference	64
6.8.1	Constructor & Destructor Documentation	64
6.8.1.1	Timer	64
6.8.1.2	~Timer	64
6.8.2	Member Function Documentation	65
6.8.2.1	seconds	65
6.8.2.2	tic	65
6.8.2.3	toc	65
6.8.3	Friends And Related Function Documentation	65
6.8.3.1	operator<<	65
6.8.4	Member Data Documentation	65
6.8.4.1	_end	65

6.8.4.2	<code>_start</code>	65
6.9	<code>qpp::UniformRealDistribution</code> Class Reference	65
6.9.1	Constructor & Destructor Documentation	65
6.9.1.1	<code>UniformRealDistribution</code>	65
6.9.2	Member Function Documentation	65
6.9.2.1	<code>sample</code>	66
6.9.3	Member Data Documentation	66
6.9.3.1	<code>_d</code>	66
<b>7</b>	<b>File Documentation</b>	<b>67</b>
7.1	<code>include/channels.h</code> File Reference	67
7.2	<code>include/classes/exception.h</code> File Reference	68
7.3	<code>include/classes/gates.h</code> File Reference	69
7.4	<code>include/classes/qudit.h</code> File Reference	70
7.5	<code>include/classes/randevs.h</code> File Reference	71
7.6	<code>include/classes/stat.h</code> File Reference	72
7.7	<code>include/classes/timer.h</code> File Reference	73
7.8	<code>include/constants.h</code> File Reference	74
7.9	<code>include/entanglement.h</code> File Reference	76
7.10	<code>include/entropies.h</code> File Reference	77
7.11	<code>include/functions.h</code> File Reference	78
7.12	<code>include/internal.h</code> File Reference	81
7.13	<code>include/io.h</code> File Reference	82
7.14	<code>include/matlab.h</code> File Reference	84
7.15	<code>include/qpp.h</code> File Reference	84
7.16	<code>include/random.h</code> File Reference	85
7.17	<code>include/types.h</code> File Reference	87

# 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>	44
<a href="#">qpp::internal</a>	45
<a href="#">qpp::types</a>	46



## 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 . . . . .	60
qpp::Qudit . . . . .	61
qpp::RandomDevices . . . . .	63
qpp::Timer . . . . .	64
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>	60
<a href="#">qpp::Qudit</a>	61
<a href="#">qpp::RandomDevices</a>	63
<a href="#">qpp::Timer</a>	64
<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	74
include/entanglement.h	76
include/entropies.h	77
include/functions.h	78
include/internal.h	81
include/io.h	82
include/matlab.h	84
include/qpp.h	84
include/random.h	85
include/types.h	87
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/timer.h	73



## 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 [Timer](#)

#### Functions

- [types::cmat channel](#) (const [types::cmat](#) &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 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 > &subs, 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 > fun (const Eigen::MatrixBase< Derived > &A, OutputScalar(*f)(const type-`  
`name 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-`  
`derived2 > &B)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > prj (const Eigen::MatrixBase< Derived > &V)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > expandout (const Eigen::MatrixBase< Derived > &A, 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)

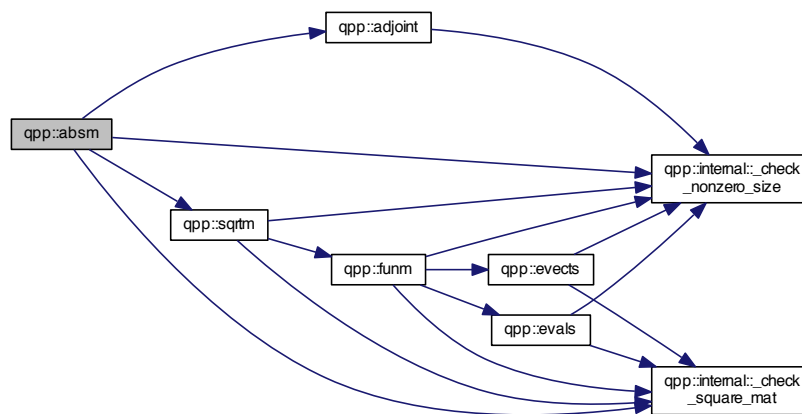
## Variables

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

### 5.1.1 Function Documentation

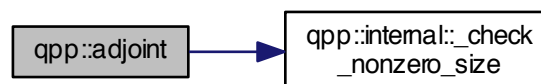
5.1.1.1 `template<typename Derived > types::cmat qpp::absm ( const Eigen::MatrixBase< Derived > & A )`

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:



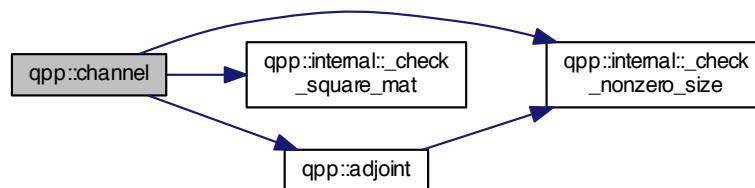
5.1.1.3 `template<typename Derived1 , typename Derived2 > types::DynMat<typename Derived1::Scalar> qpp::anticomm ( const Eigen::MatrixBase< Derived1 > & A, const Eigen::MatrixBase< Derived2 > & B )`

Here is the call graph for this function:



5.1.1.4 `types::cmat qpp::channel ( const types::cmat & rho, const std::vector< types::cmat > & Ks )`

Here is the call graph for this function:





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

Here is the call graph for this function:



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

Here is the call graph for this function:



5.1.1.7 `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:



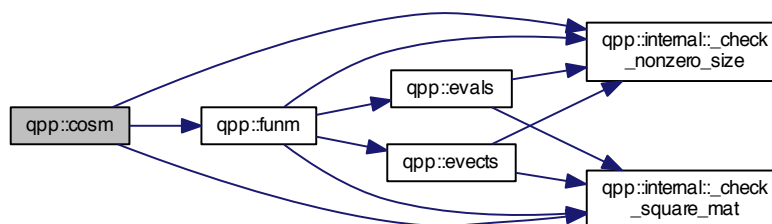
5.1.1.8 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::conjugate ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



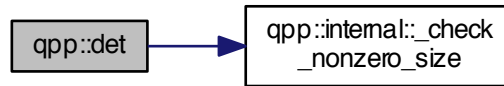
5.1.1.9 `template<typename Derived > types::cmat qpp::cosm ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



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

Here is the call graph for this function:



5.1.1.11 `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 )`

5.1.1.12 `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 )`

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

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

Here is the call graph for this function:



5.1.1.15 `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:



5.1.1.16 `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:



5.1.1.17 `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:



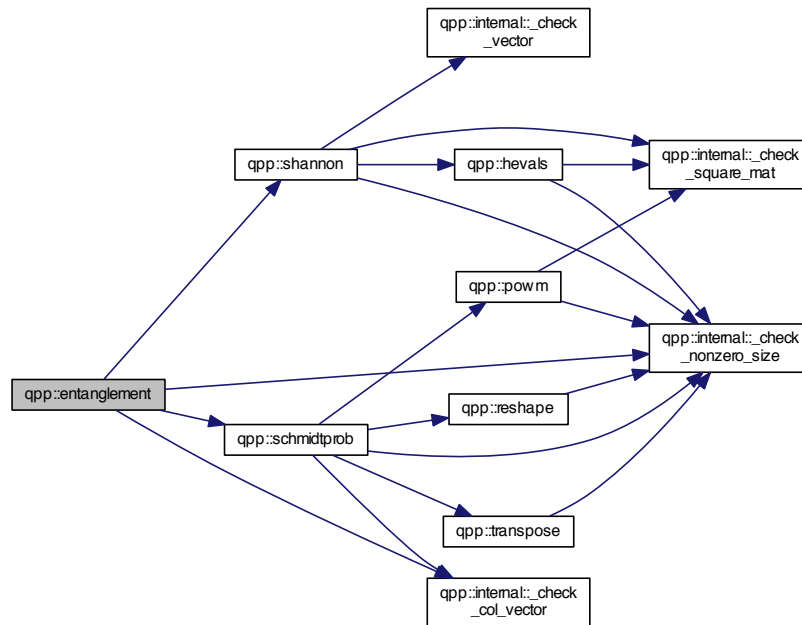
5.1.1.18 `void qpp::displn ( const types::cplx c, double chop = ct::chop, std::ostream & os = std::cout )`

Here is the call graph for this function:



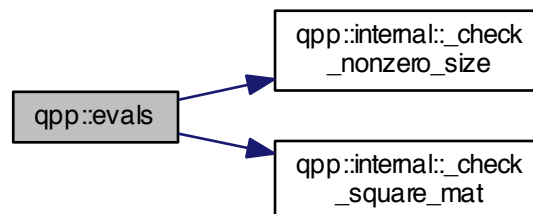
5.1.1.19 `template<typename Derived> double qpp::entanglement ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims )`

Here is the call graph for this function:



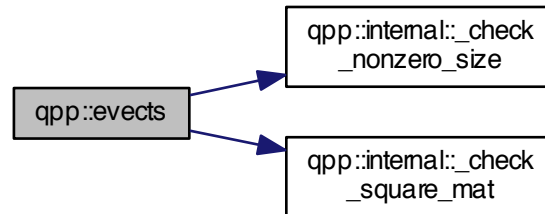
5.1.1.20 `template<typename Derived> types::cmat qpp::evals ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



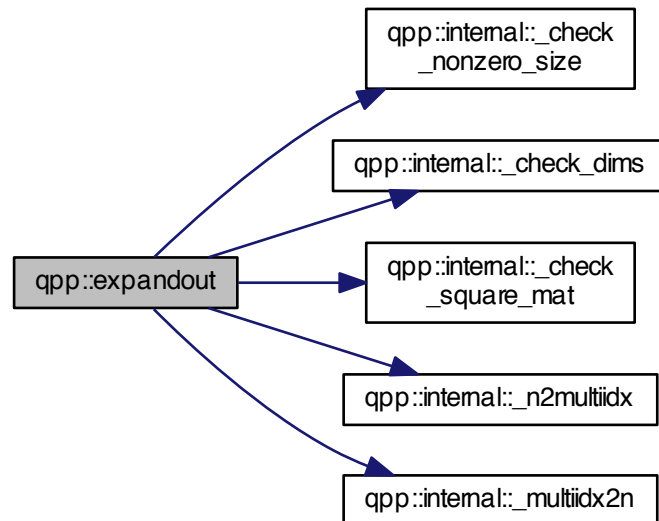
5.1.1.21 `template<typename Derived> types::cmat qpp::evecs ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



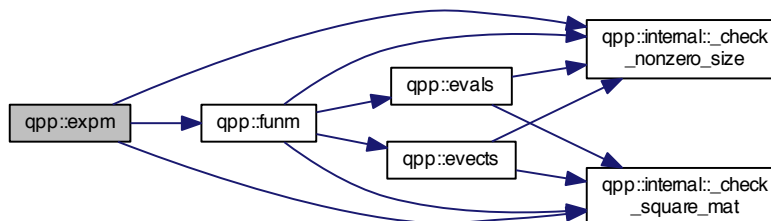
5.1.1.22 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::expandout ( const Eigen::MatrixBase< Derived> & A, size_t pos, const std::vector< size_t> & dims )`

Here is the call graph for this function:



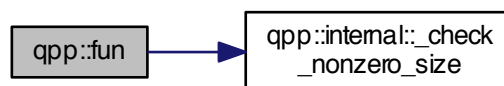
5.1.1.23 `template<typename Derived > types::cmat qpp::expm ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



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

Here is the call graph for this function:



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

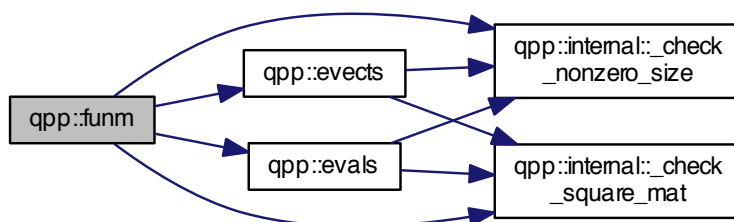
#### Parameters

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

Returns

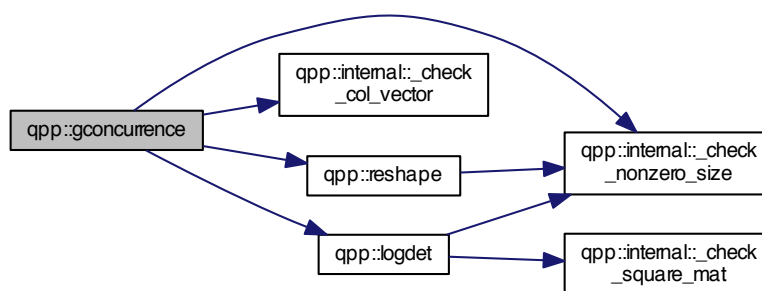
[types::cmat](#)

Here is the call graph for this function:



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

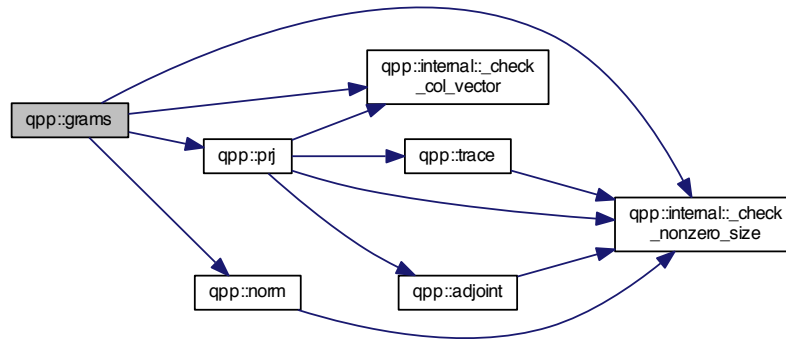
Here is the call graph for this function:





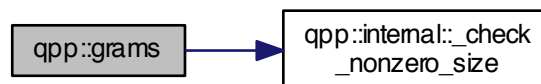
5.1.1.27 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::grams ( const std::vector<types::DynMat<typename Derived::Scalar>> & Vs )`

Here is the call graph for this function:



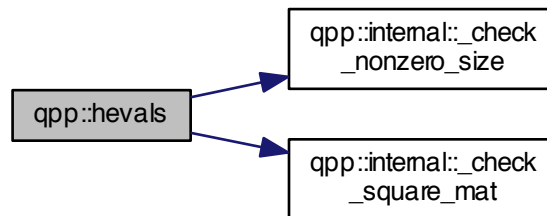
5.1.1.28 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::grams ( const Eigen::MatrixBase<Derived> & A )`

Here is the call graph for this function:



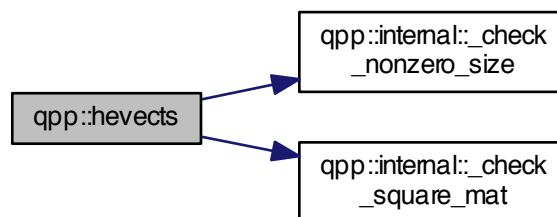
5.1.1.29 `template<typename Derived> types::dmat qpp::hevals ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



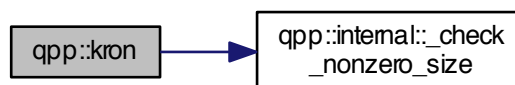
5.1.1.30 `template<typename Derived> types::cmat qpp::hevects ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



5.1.1.31 `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:



5.1.1.32 `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:



5.1.1.33 `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:



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

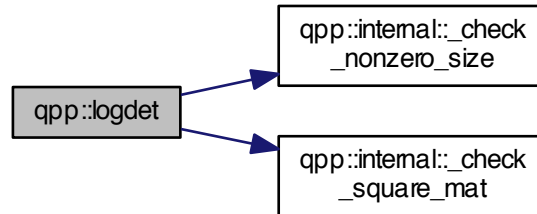
5.1.1.35 `template<typename Derived> Derived qpp::loadMATLABmatrix ( const std::string & mat_file, const std::string & var_name )`

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

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

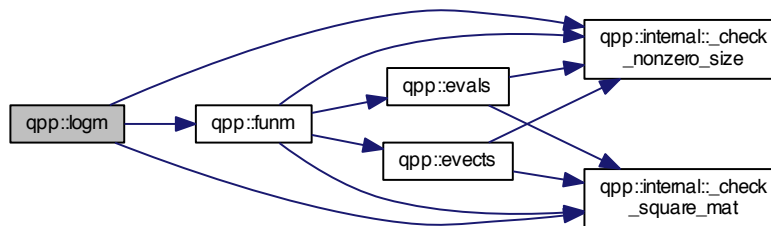
5.1.1.38 `template<typename Derived> Derived::Scalar qpp::logdet ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



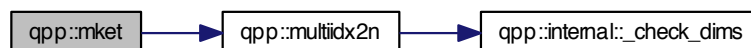
5.1.1.39 `template<typename Derived> types::cmat qpp::logm ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



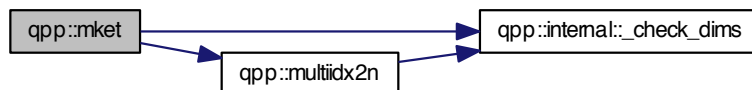
5.1.1.40 `types::ket qpp::mket ( const std::vector< size_t> & mask )`

Here is the call graph for this function:



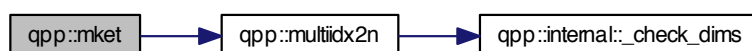
#### 5.1.1.41 `types::ket qpp::mket ( const std::vector< size_t > & mask, const std::vector< size_t > & dims )`

Here is the call graph for this function:



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

Here is the call graph for this function:



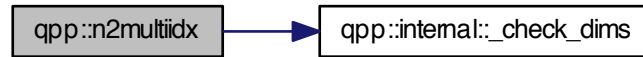
#### 5.1.1.43 `size_t qpp::multidx2n ( const std::vector< size_t > & midx, const std::vector< size_t > & dims )`

Here is the call graph for this function:



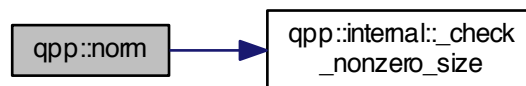
5.1.1.44 `std::vector<size_t> qpp::n2multiidx ( size_t n, const std::vector< size_t > & dims )`

Here is the call graph for this function:



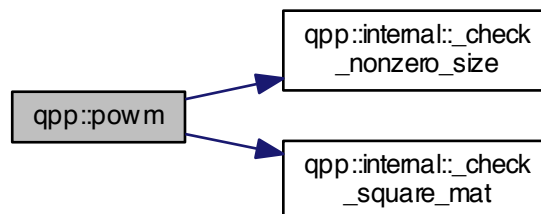
5.1.1.45 `template<typename Derived > double qpp::norm ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



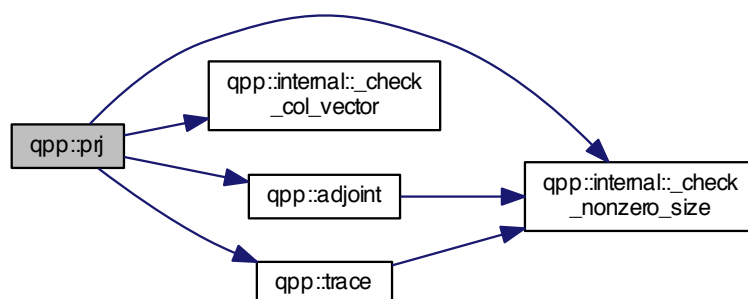
5.1.1.46 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::powm ( const Eigen::MatrixBase< Derived > & A, size_t n )`

Here is the call graph for this function:



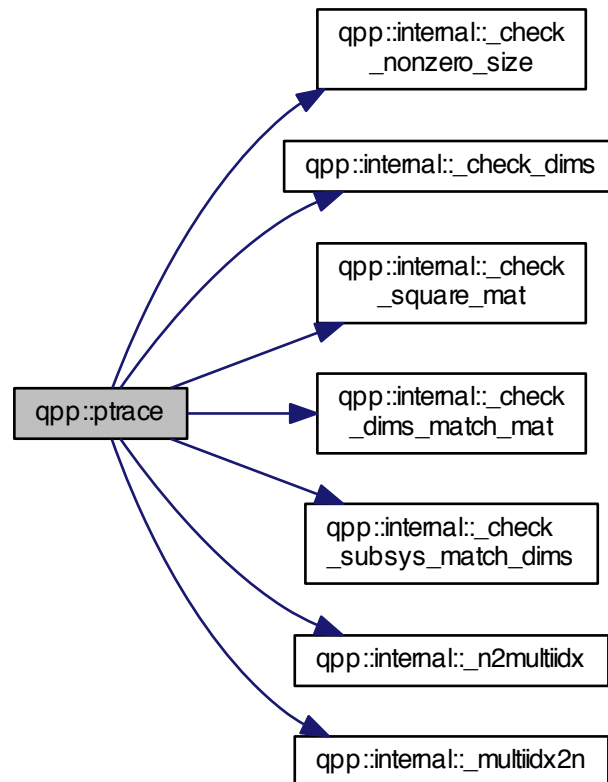
5.1.1.47 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::prj ( const Eigen::MatrixBase<Derived> & V )`

Here is the call graph for this function:



5.1.1.48 `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 )`

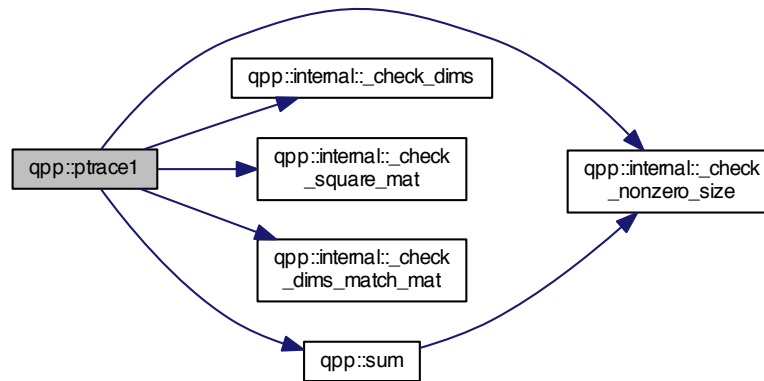
Here is the call graph for this function:





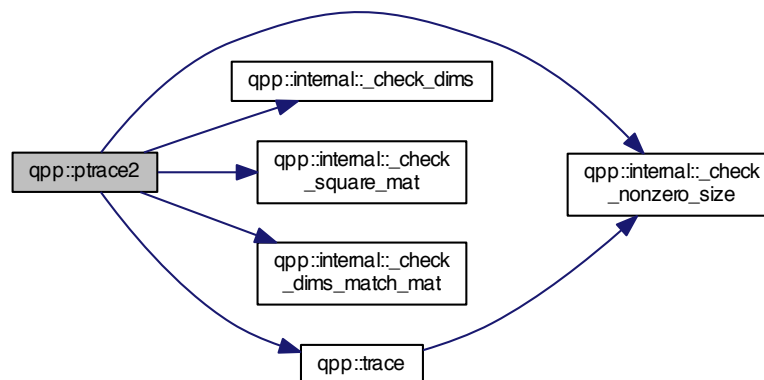
5.1.1.49 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::ptrace1 ( const Eigen::MatrixBase< Derived > & A, const std::vector< size_t > & dims )`

Here is the call graph for this function:



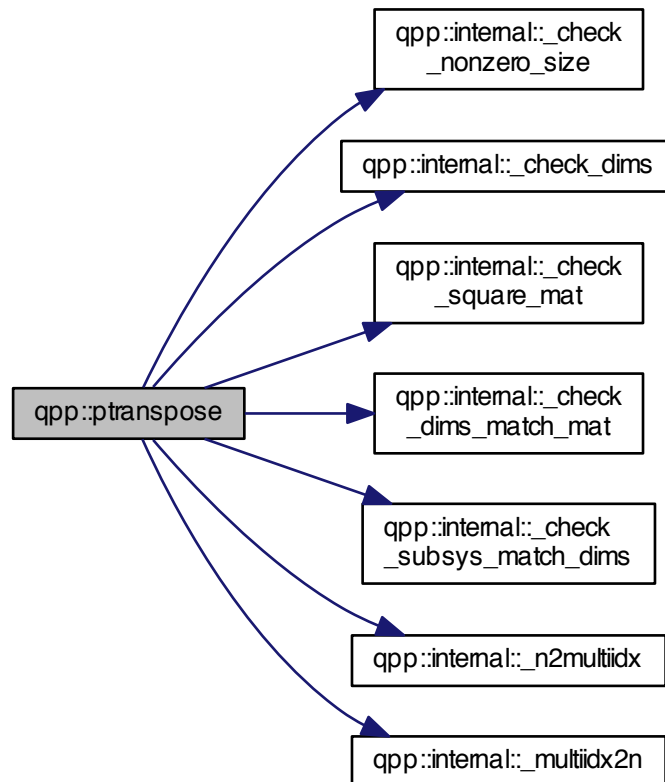
5.1.1.50 `template<typename Derived > types::DynMat<typename Derived::Scalar> qpp::ptrace2 ( const Eigen::MatrixBase< Derived > & A, const std::vector< size_t > & dims )`

Here is the call graph for this function:



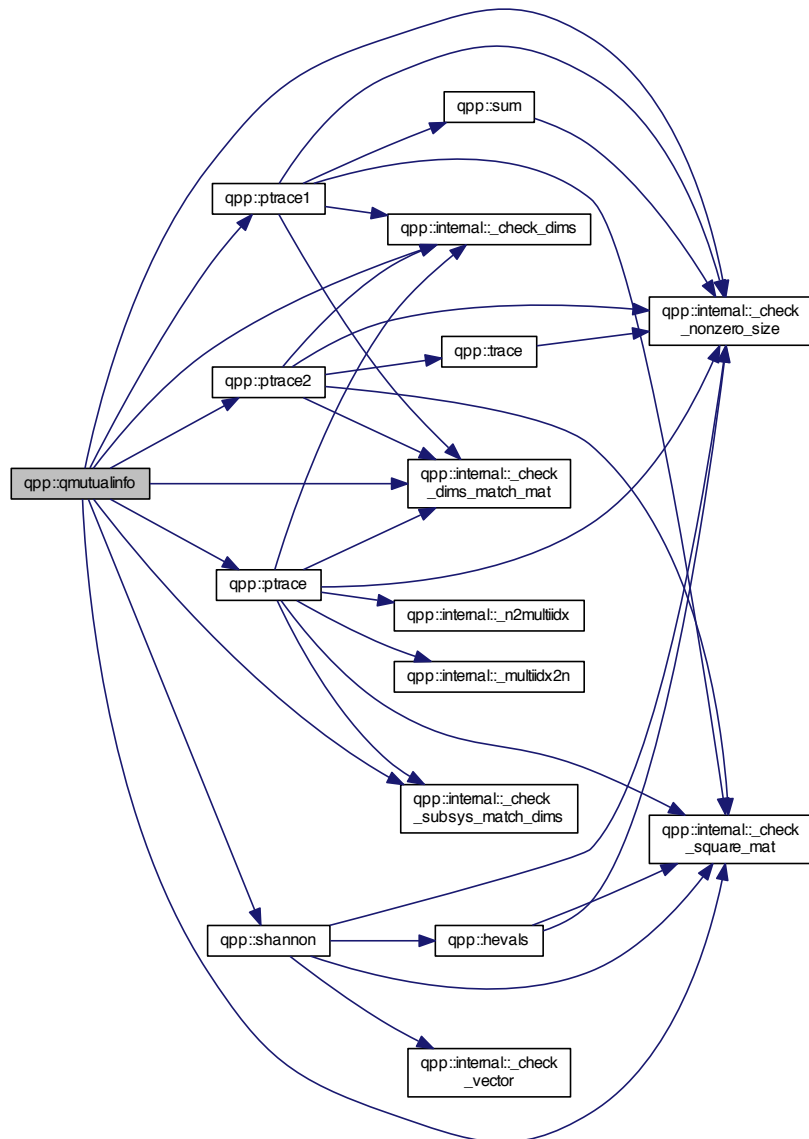
5.1.1.51 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::ptranspose ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & subsys, const std::vector< size_t> & dims )`

Here is the call graph for this function:



5.1.1.52 `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:



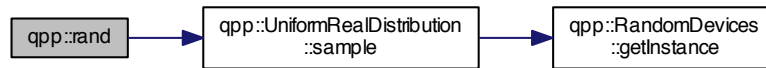
5.1.1.53 `template<typename Derived > Derived qpp::rand ( size_t rows, size_t cols, double a = 0, double b = 1 )`

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

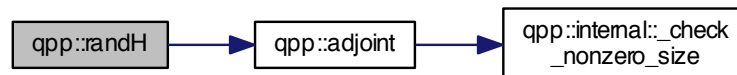
5.1.1.55 `template<> types::cmat qpp::rand ( size_t rows, size_t cols, double a, double b )`

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

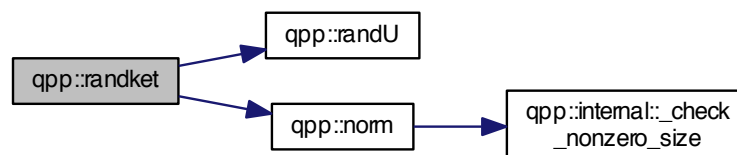
Here is the call graph for this function:

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

Here is the call graph for this function:

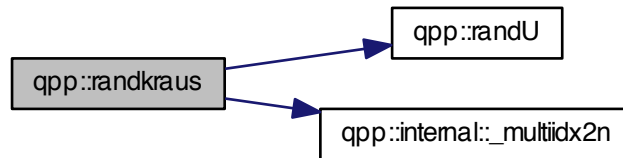
**5.1.1.58** `types::ket qpp::randket ( size_t D )`

Here is the call graph for this function:



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

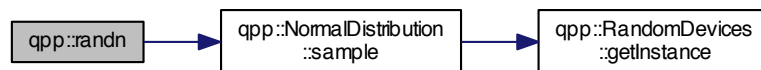
Here is the call graph for this function:



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

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

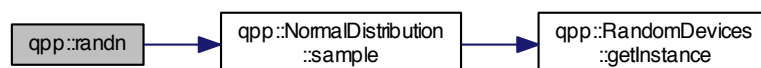
Here is the call graph for this function:



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

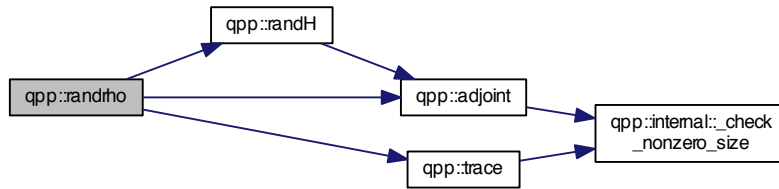
5.1.1.63 `double qpp::randn ( double mean = 0, double sigma = 1 )`

Here is the call graph for this function:



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

Here is the call graph for this function:



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

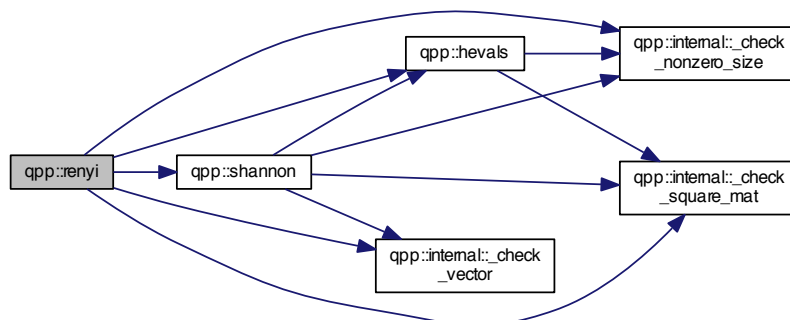
#### 5.1.1.66 `types::cmat qpp::randV ( size_t Din, size_t Dout )`

Here is the call graph for this function:



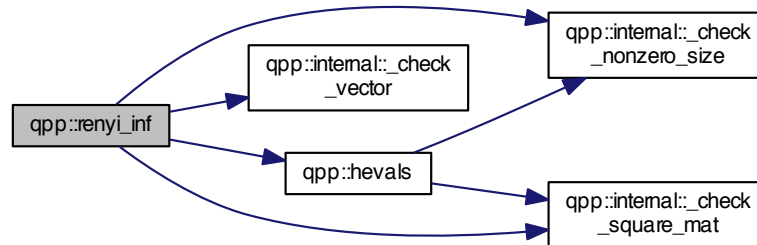
#### 5.1.1.67 `template<typename Derived> double qpp::renyi ( const double alpha, const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



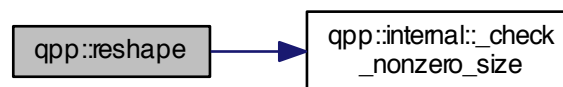
5.1.1.68 `template<typename Derived> double qpp::renyi_inf ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



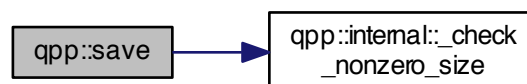
5.1.1.69 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::reshape ( const Eigen::MatrixBase< Derived > & A, size_t rows, size_t cols )`

Here is the call graph for this function:



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

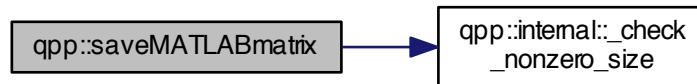
Here is the call graph for this function:



5.1.1.71 `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 )`

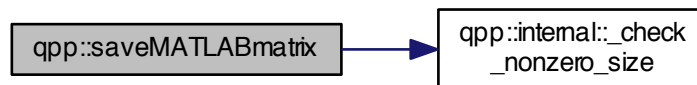
5.1.1.72 `template<> void qpp::saveMATLABmatrix ( const Eigen::MatrixBase< typename types::dmat > & A, const std::string & mat_file, const std::string & var_name, const std::string & mode )`

Here is the call graph for this function:



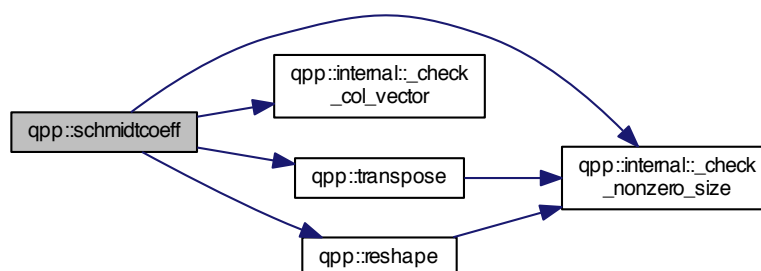
5.1.1.73 `template<> void qpp::saveMATLABmatrix ( const Eigen::MatrixBase< typename types::cmat > & A, const std::string & mat_file, const std::string & var_name, const std::string & mode )`

Here is the call graph for this function:



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

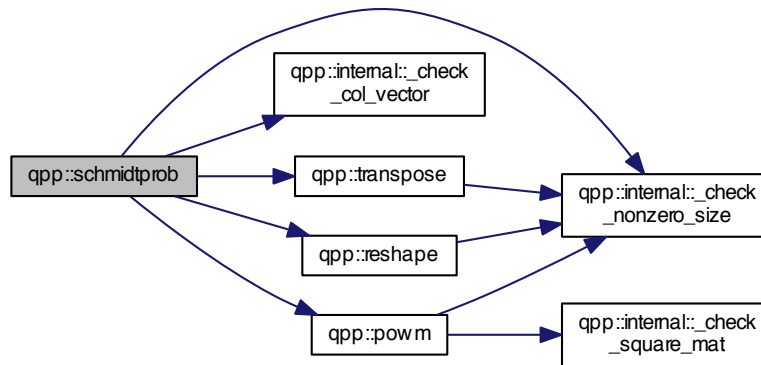
Here is the call graph for this function:





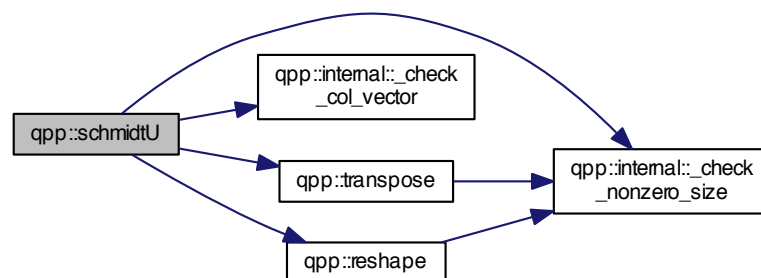
5.1.1.75 `template<typename Derived> types::cmat qpp::schmidtprob ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims )`

Here is the call graph for this function:



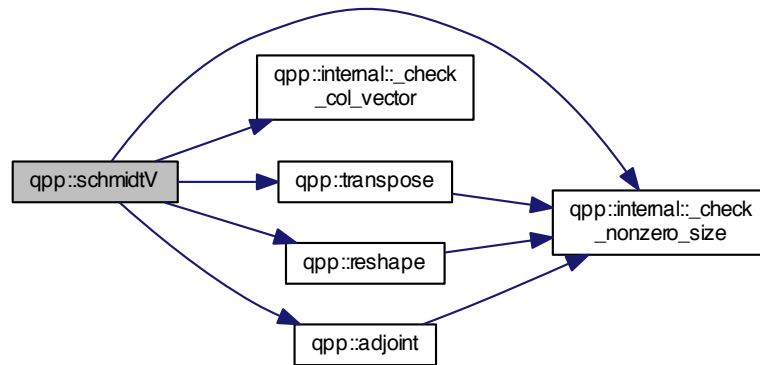
5.1.1.76 `template<typename Derived> types::cmat qpp::schmidtU ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims )`

Here is the call graph for this function:



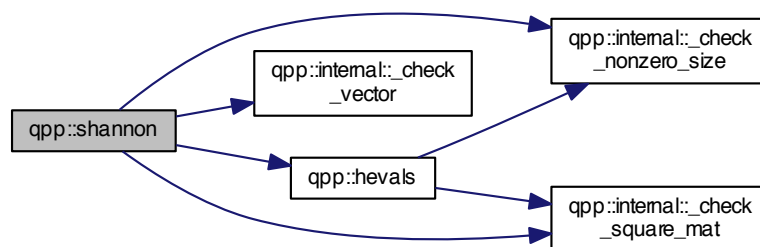
5.1.1.77 `template<typename Derived> types::cmat qpp::schmidtV ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & dims )`

Here is the call graph for this function:



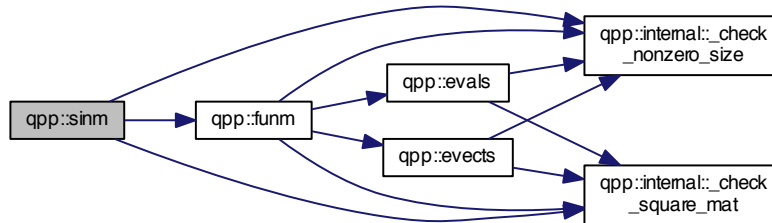
5.1.1.78 `template<typename Derived> double qpp::shannon ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



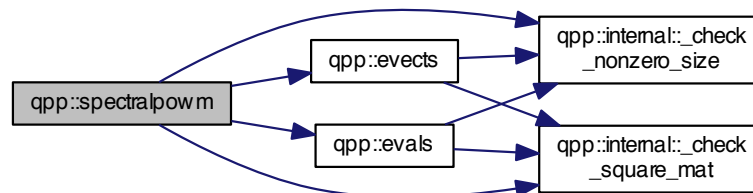
5.1.1.79 `template<typename Derived> types::cmat qpp::sinm ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



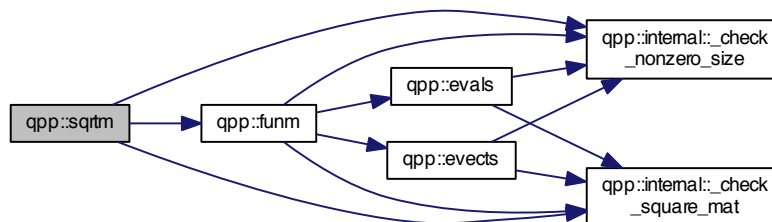
5.1.1.80 `template<typename Derived> types::cmat qpp::spectralpowm ( const Eigen::MatrixBase< Derived > & A, const types::cplx z )`

Here is the call graph for this function:



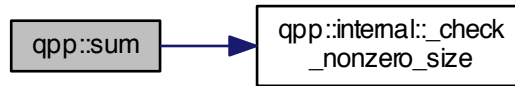
5.1.1.81 `template<typename Derived> types::cmat qpp::sqrtm ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



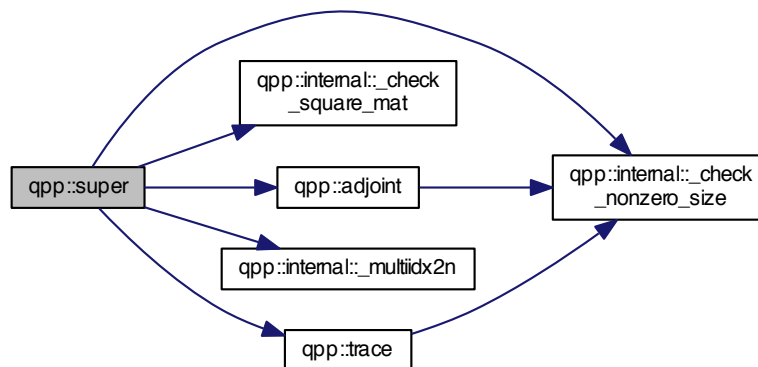
5.1.1.82 `template<typename Derived > Derived::Scalar qpp::sum ( const Eigen::MatrixBase< Derived > & A )`

Here is the call graph for this function:



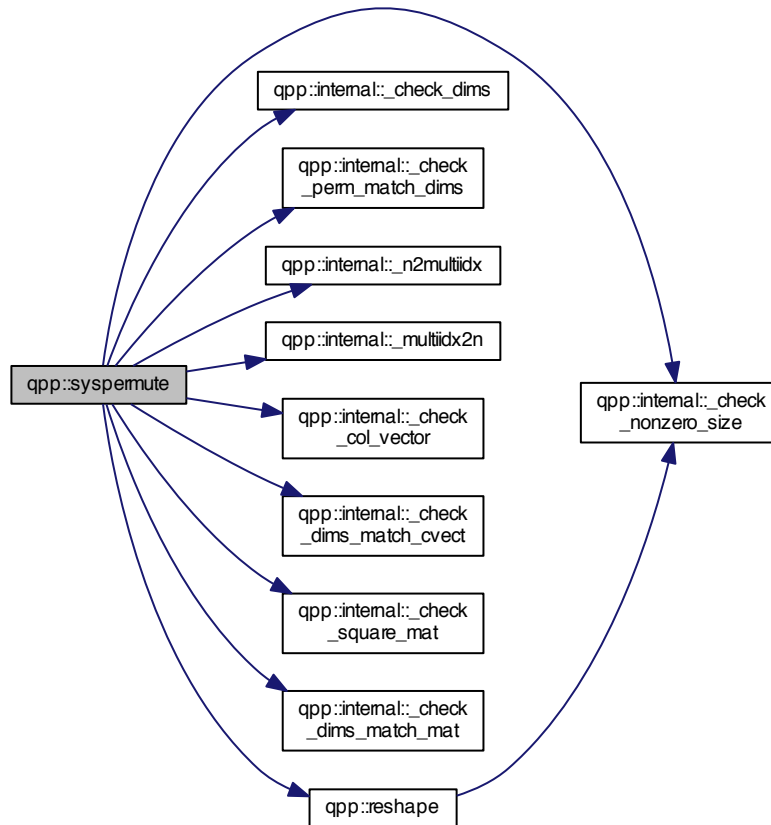
5.1.1.83 `types::cmat qpp::super ( const std::vector< types::cmat > & Ks )`

Here is the call graph for this function:



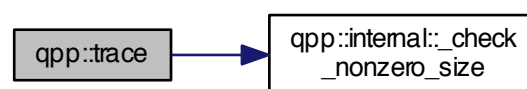
5.1.1.84 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::syspermute ( const Eigen::MatrixBase< Derived> & A, const std::vector< size_t> & perm, const std::vector< size_t> & dims )`

Here is the call graph for this function:



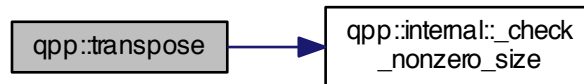
5.1.1.85 `template<typename Derived> Derived::Scalar qpp::trace ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



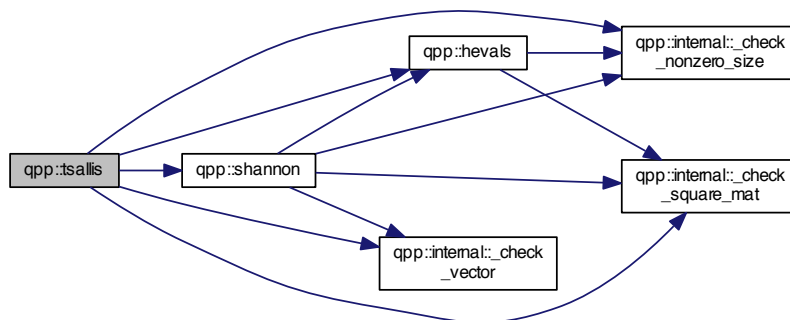
5.1.1.86 `template<typename Derived> types::DynMat<typename Derived::Scalar> qpp::transpose ( const Eigen::MatrixBase< Derived> & A )`

Here is the call graph for this function:



5.1.1.87 `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()`

## 5.2 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]

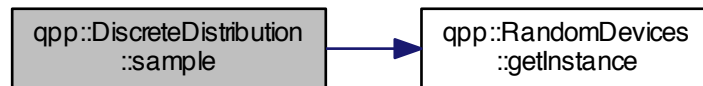
6.1.1.3 `qpp::DiscreteDistribution::DiscreteDistribution ( std::vector< double > weights )` [inline]

#### 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)

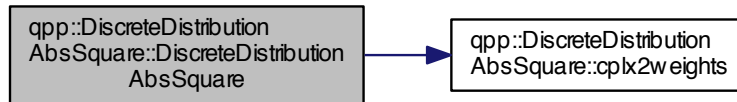
### Protected Attributes

- `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:



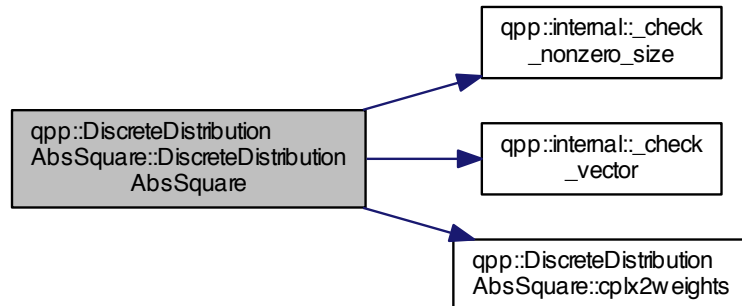
6.2.1.3 `qpp::DiscreteDistributionAbsSquare::DiscreteDistributionAbsSquare ( std::vector< types::cplx > amplitudes ) [inline]`

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:



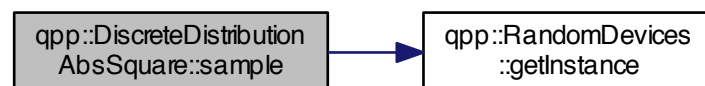
## 6.2.2 Member Function Documentation

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]`

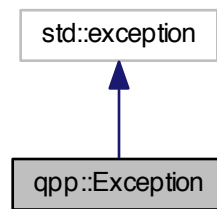
The documentation for this class was generated from the following file:

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

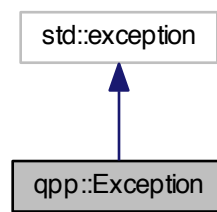
## 6.3 qpp::Exception Class Reference

```
#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

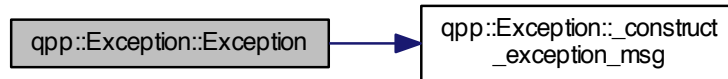
***UNKNOWN\_EXCEPTION***  
***ZERO\_SIZE***  
***MATRIX\_NOT\_SQUARE***  
***MATRIX\_NOT\_CVECTOR***  
***MATRIX\_NOT\_RVECTOR***  
***MATRIX\_NOT\_VECTOR***  
***MATRIX\_NOT\_SQUARE\_OR\_CVECTOR***  
***MATRIX\_NOT\_SQUARE\_OR\_RVECTOR***  
***MATRIX\_NOT\_SQUARE\_OR\_VECTOR***  
***DIMS\_INVALID***  
***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

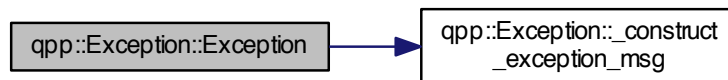
6.3.2.1 `qpp::Exception::Exception ( const std::string & where, const Type & type )` `[inline]`

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

- [Gates \(\)](#)

### 6.4.1 Constructor & Destructor Documentation

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

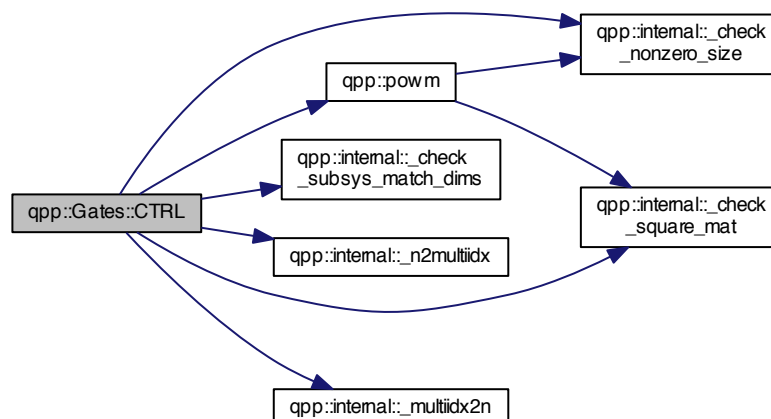
6.4.1.2 `qpp::Gates::Gates ( const Gates & )` `[delete]`

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 > & gate, 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]`

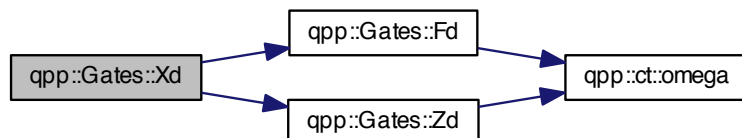
6.4.2.4 `types::cmat qpp::Gates::ld ( size_t D ) const` `[inline]`

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::ket qpp::Gates::b00`

6.4.3.2 `types::ket qpp::Gates::b01`

6.4.3.3 `types::ket qpp::Gates::b10`

6.4.3.4 `types::ket qpp::Gates::b11`

6.4.3.5 `types::cmat qpp::Gates::CNOTab`

6.4.3.6 `types::cmat qpp::Gates::CNOTba`

6.4.3.7 `types::cmat qpp::Gates::CS`

6.4.3.8 `types::cmat qpp::Gates::CZ`

6.4.3.9 `types::cmat qpp::Gates::FRED`

6.4.3.10 `types::ket qpp::Gates::GHZ`

6.4.3.11 `types::cmat qpp::Gates::H`

6.4.3.12 `types::cmat qpp::Gates::Id2`

6.4.3.13 `types::cmat qpp::Gates::pb00`

6.4.3.14 `types::cmat qpp::Gates::pb01`

6.4.3.15 `types::cmat qpp::Gates::pb10`

6.4.3.16 `types::cmat qpp::Gates::pb11`

6.4.3.17 `types::cmat qpp::Gates::pGHZ`

6.4.3.18 `types::cmat qpp::Gates::pW`

6.4.3.19 `types::cmat qpp::Gates::px0`

6.4.3.20 `types::cmat qpp::Gates::px1`

6.4.3.21 `types::cmat qpp::Gates::py0`

6.4.3.22 `types::cmat qpp::Gates::py1`

6.4.3.23 `types::cmat qpp::Gates::pz0`

6.4.3.24 `types::cmat qpp::Gates::pz1`

6.4.3.25 `types::cmat qpp::Gates::S`

6.4.3.26 `types::cmat qpp::Gates::SWAP`

6.4.3.27 `types::cmat qpp::Gates::T`

6.4.3.28 `types::cmat qpp::Gates::TOF`

6.4.3.29 `types::ket qpp::Gates::W`

6.4.3.30 `types::cmat qpp::Gates::X`

6.4.3.31 `types::ket qpp::Gates::x0`

6.4.3.32 `types::ket qpp::Gates::x1`

6.4.3.33 `types::cmat qpp::Gates::Y`

6.4.3.34 `types::ket qpp::Gates::y0`

6.4.3.35 `types::ket qpp::Gates::y1`

6.4.3.36 `types::cmat qpp::Gates::Z`

6.4.3.37 `types::ket qpp::Gates::z0`

6.4.3.38 `types::ket qpp::Gates::z1`

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`

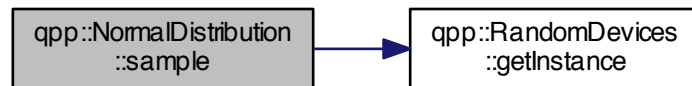
### 6.5.1 Constructor & Destructor Documentation

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=[Gates::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

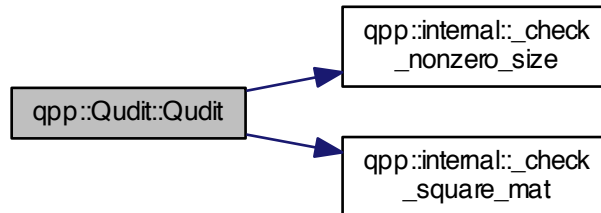
### Private Attributes

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

#### 6.6.1 Constructor & Destructor Documentation

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

Here is the call graph for this function:



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

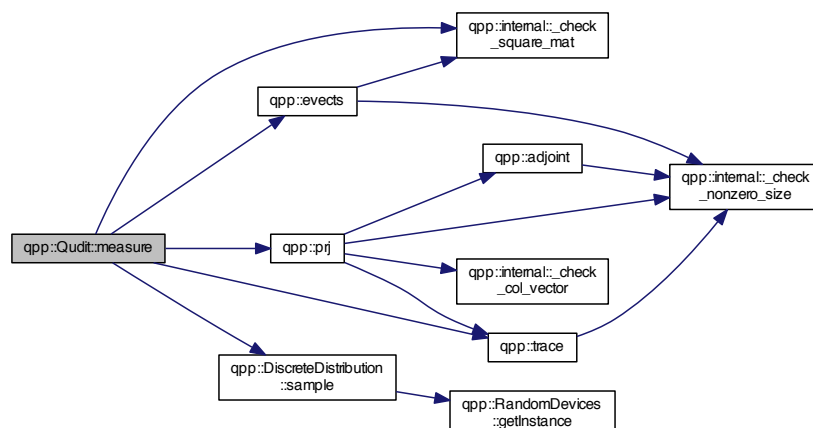
## 6.6.2 Member Function Documentation

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]`

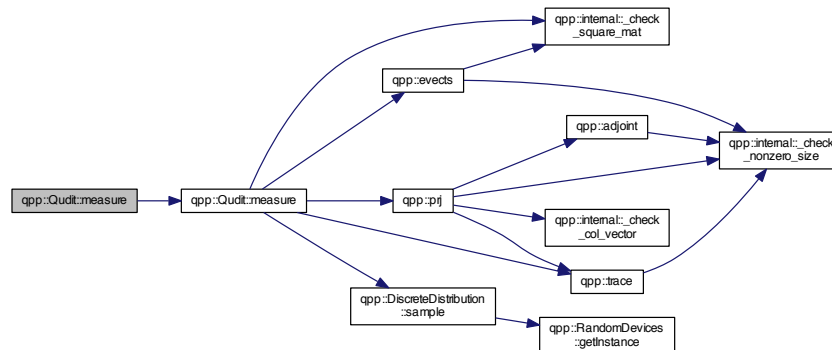
Here is the call graph for this function:





#### 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::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)`

### 6.8.1 Constructor & Destructor Documentation

6.8.1.1 `qpp::Timer::Timer ( ) [inline]`

6.8.1.2 `virtual qpp::Timer::~~Timer ( ) [virtual],[default]`

## 6.8.2 Member Function Documentation

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

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

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

## 6.8.3 Friends And Related Function Documentation

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

## 6.8.4 Member Data Documentation

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

6.8.4.2 `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](#)

# 6.9 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`

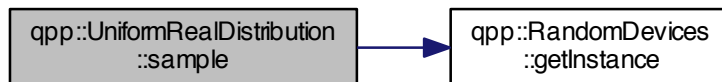
## 6.9.1 Constructor & Destructor Documentation

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

## 6.9.2 Member Function Documentation

#### 6.9.2.1 `double qpp::UniformRealDistribution::sample ( ) [inline]`

Here is the call graph for this function:



### 6.9.3 Member Data Documentation

#### 6.9.3.1 `std::uniform_real_distribution qpp::UniformRealDistribution::_d [protected]`

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

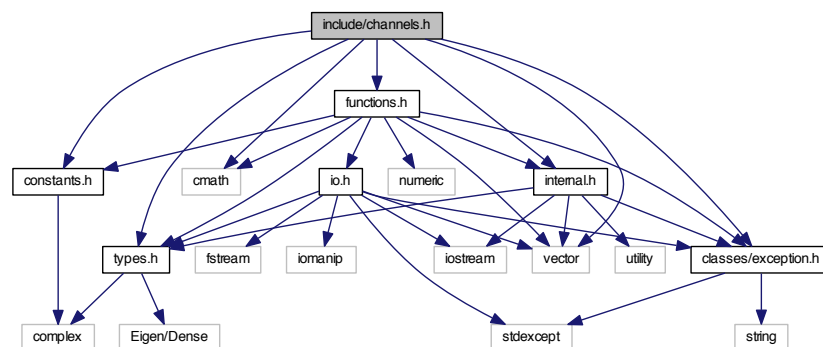
- `include/classes/stat.h`

## Chapter 7

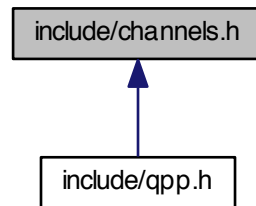
# File Documentation

### 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

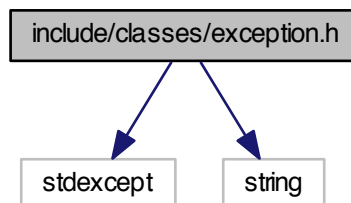
- `types::cmat` [qpp::channel](#) (`const types::cmat &rho`, `const std::vector< types::cmat > &Ks`)
- `types::cmat` [qpp::super](#) (`const std::vector< types::cmat > &Ks`)
- `types::cmat` [qpp::choi](#) (`const std::vector< types::cmat > &Ks`)
- `std::vector< types::cmat >` [qpp::choi2kraus](#) (`const types::cmat &A`)

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

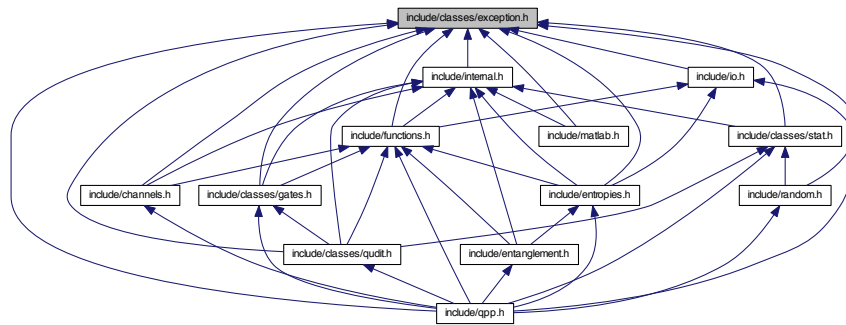
```
#include <stdexcept>
```

```
#include <string>
```

Include dependency graph for exception.h:



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



## Classes

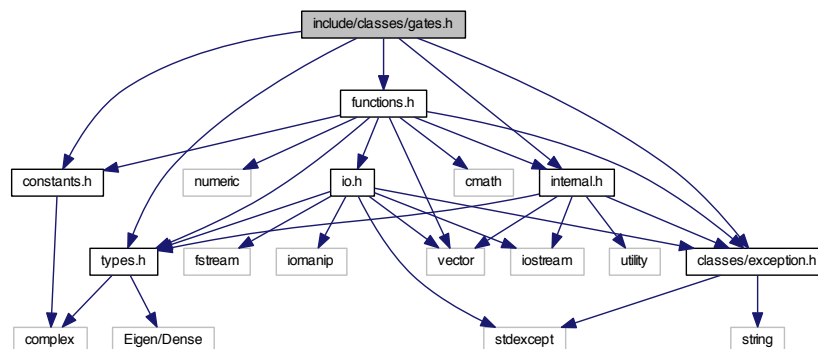
- class `qpp::Exception`

## Namespaces

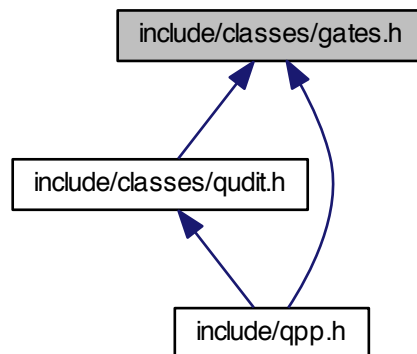
- `qpp`

## 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](#)

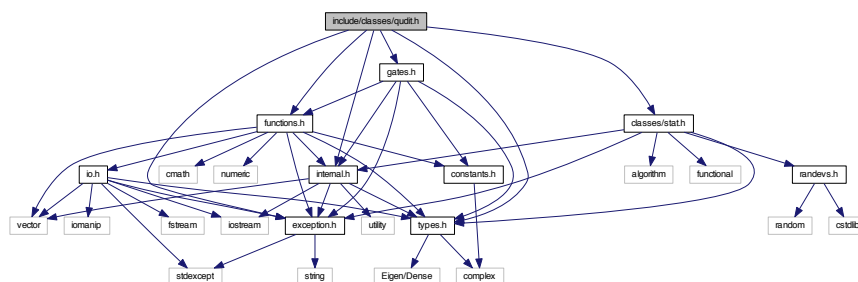
## Namespaces

- [qpp](#)

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

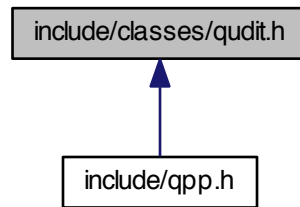
```
#include "exception.h"
#include "functions.h"
#include "gates.h"
#include "internal.h"
#include "types.h"
#include "classes/stat.h"
```

Include dependency graph for qudit.h:





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



### Classes

- class [qpp::Qudit](#)

### Namespaces

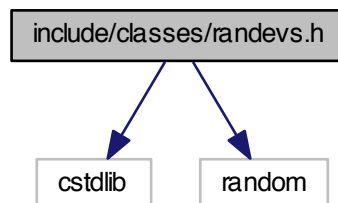
- [qpp](#)

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

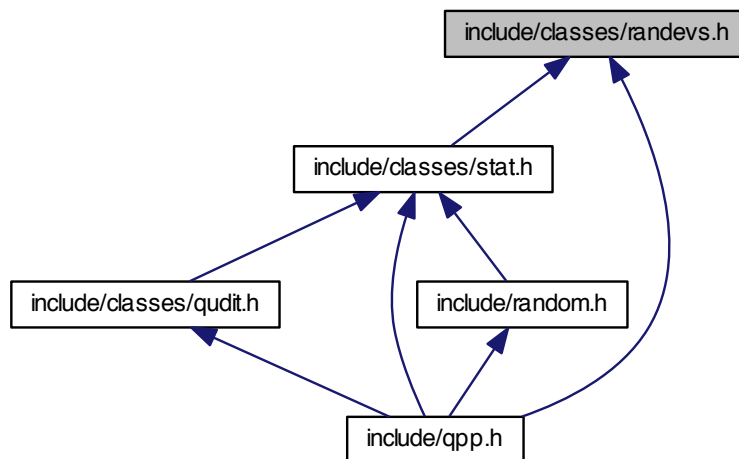
```
#include <cstdlib>
```

```
#include <random>
```

Include dependency graph for randevs.h:



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



## Classes

- class [qpp::RandomDevices](#)

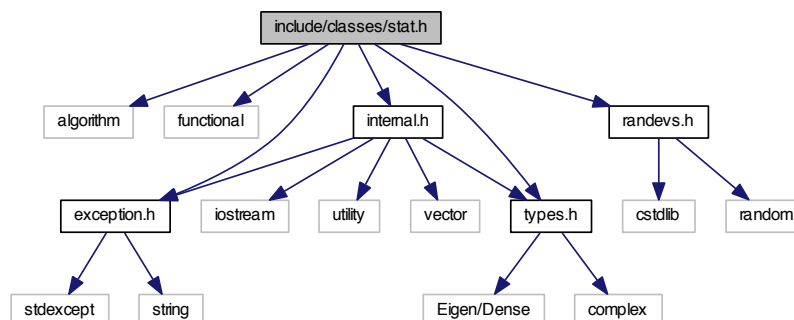
## Namespaces

- [qpp](#)

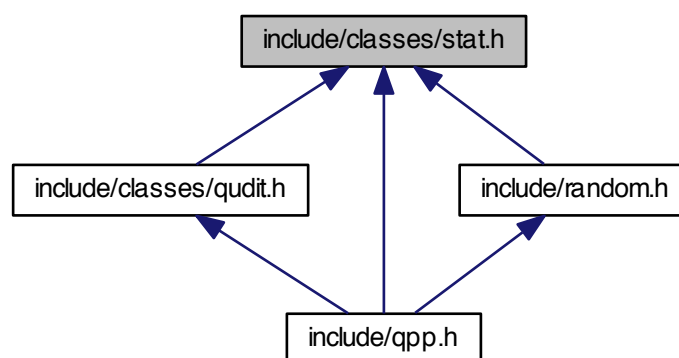
## 7.6 include/classes/stat.h File Reference

```
#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

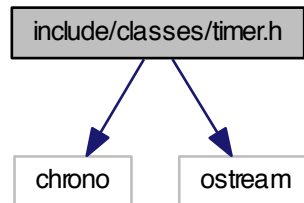
- `qpp`

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

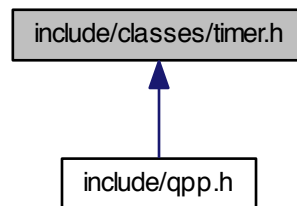
```
#include <chrono>
```

```
#include <ostream>
```

Include dependency graph for timer.h:



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



## Classes

- class [qpp::Timer](#)

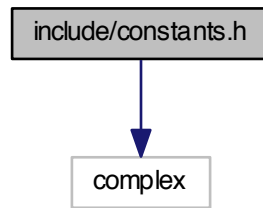
## Namespaces

- [qpp](#)

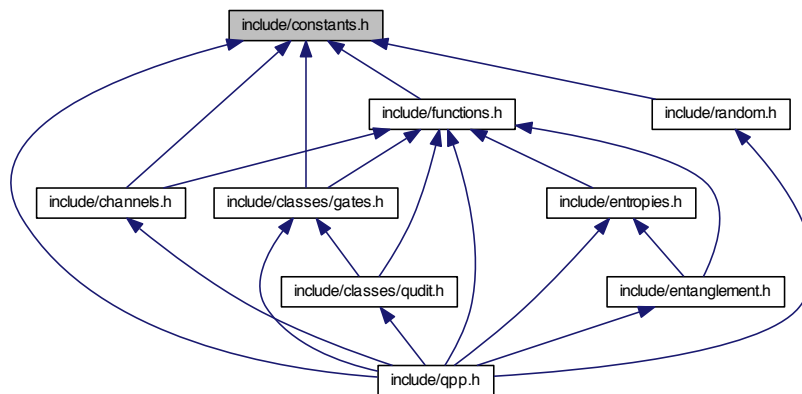
## 7.8 include/constants.h File Reference

```
#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)`

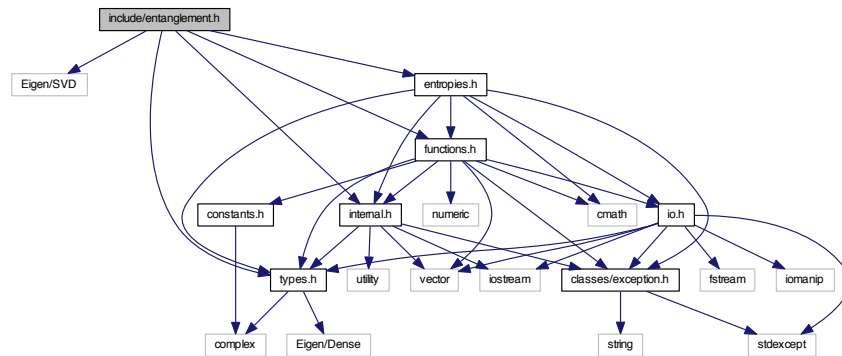
## Variables

- `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`

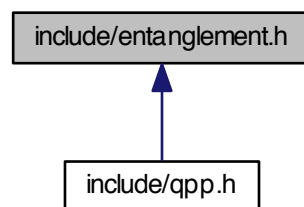
## 7.9 include/entanglement.h File Reference

```
#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

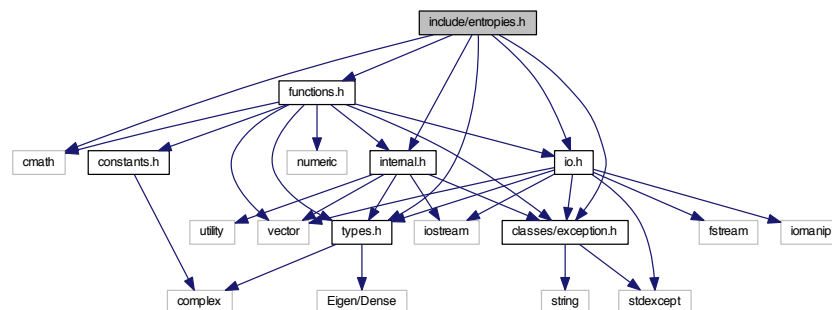
- `template<typename Derived >`  
`types::cmat qpp::schmidtcoeff (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &dims)`
- `template<typename Derived >`  
`types::cmat qpp::schmidtU (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &dims)`
- `template<typename Derived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

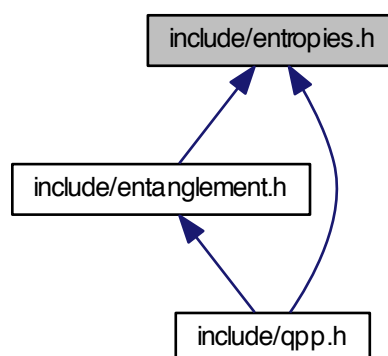
## 7.10 include/entropies.h File Reference

```
#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](#)

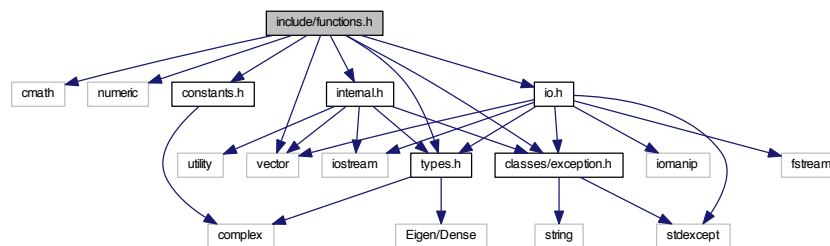
## Functions

- 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)

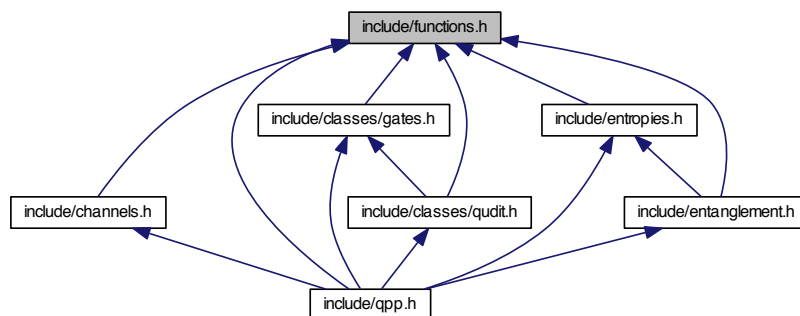
## 7.11 include/functions.h File Reference

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

Include dependency graph for functions.h:



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





## Namespaces

- [qpp](#)

## Functions

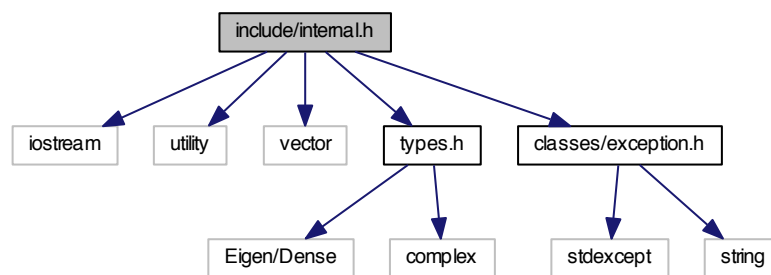
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::transpose (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::conjugate (const Eigen::MatrixBase< Derived > &A)`
- `template<typename Derived >`  
`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::fun (const Eigen::MatrixBase< Derived > &A, OutputScalar(*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< Derived2 > &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 > &subsys, const std::vector< size_t > &dims)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::ptranspose (const Eigen::MatrixBase< Derived > &A, const std::vector< size_t > &subsys, const std::vector< size_t > &dims)`
- `template<typename Derived1 , typename Derived2 >`  
`types::DynMat< typename`  
`Derived1::Scalar > qpp::comm (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase< Derived2 > &B)`
- `template<typename Derived1 , typename Derived2 >`  
`types::DynMat< typename`  
`Derived1::Scalar > qpp::anticomm (const Eigen::MatrixBase< Derived1 > &A, const Eigen::MatrixBase< Derived2 > &B)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::prj (const Eigen::MatrixBase< Derived > &V)`
- `template<typename Derived >`  
`types::DynMat< typename`  
`Derived::Scalar > qpp::expandout (const Eigen::MatrixBase< Derived > &A, size_t pos, const std::vector< size_t > &dims)`
- `template<typename 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)`

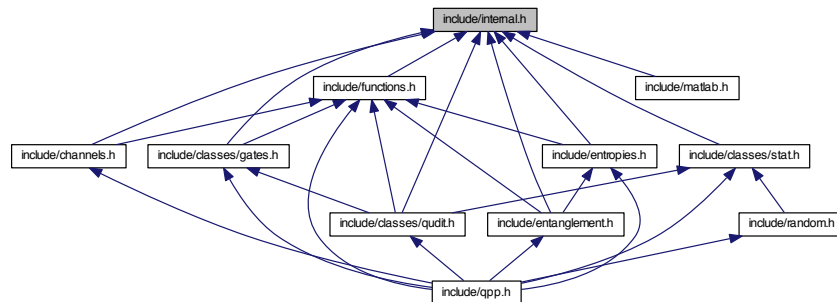
- `types::ket` [qpp::mket](#) (`const std::vector< size_t > &mask`, `const std::vector< size_t > &dims`)
- `types::ket` [qpp::mket](#) (`const std::vector< size_t > &mask`, `size_t d`)

## 7.12 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

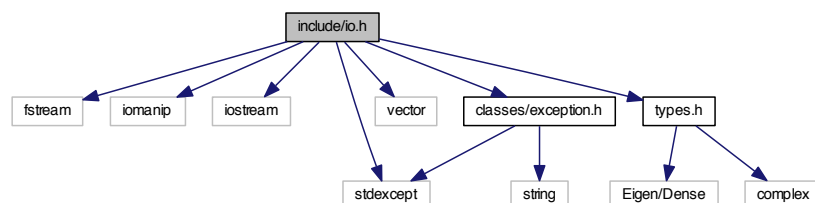
- `void` [qpp::internal::\\_n2multiidx](#) (`size_t n`, `size_t numdims`, `const size_t *dims`, `size_t *result`)
- `size_t` [qpp::internal::\\_multiidx2n](#) (`const size_t *midx`, `size_t numdims`, `const size_t *dims`)
- `template<typename Derived >`  
`bool` [qpp::internal::\\_check\\_square\\_mat](#) (`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)`

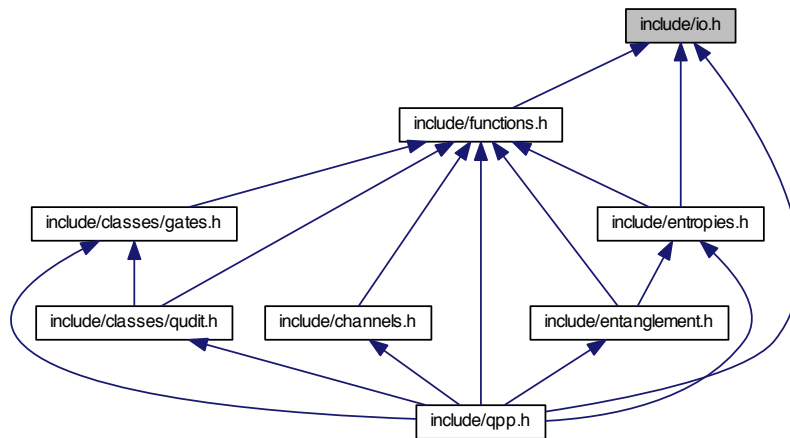
## 7.13 include/io.h File Reference

```
#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](#)

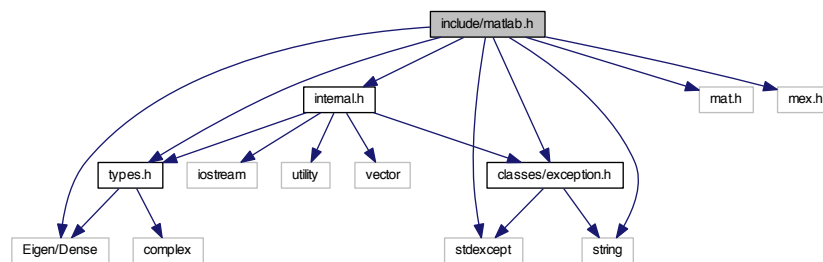
## Functions

- `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)`

## 7.14 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)`

## 7.15 include/qpp.h File Reference

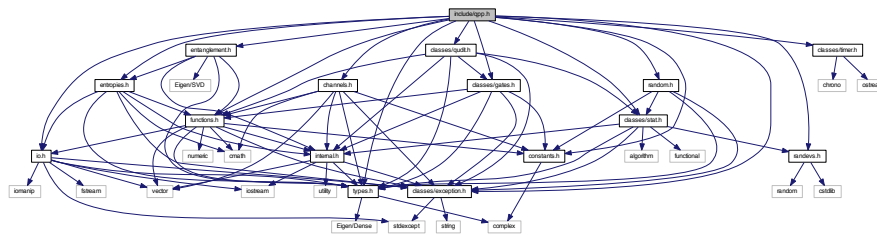
```
#include "channels.h"
```

```

#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/timer.h"

```

Include dependency graph for qpp.h:



## Namespaces

- [qpp](#)

## Variables

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

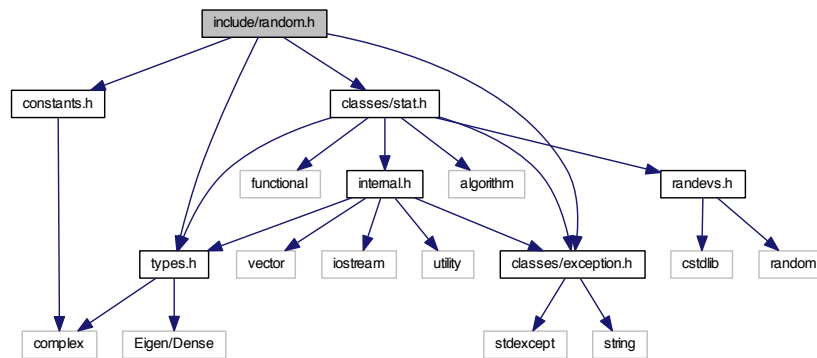
## 7.16 include/random.h File Reference

```

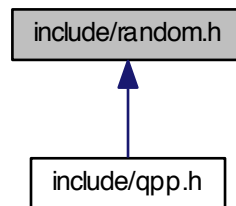
#include "constants.h"
#include "types.h"
#include "classes/exception.h"
#include "classes/stat.h"

```

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)



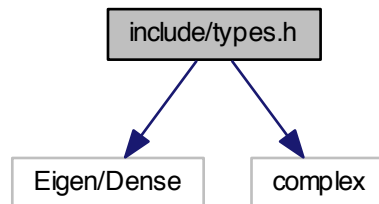
- `types::cmat` [qpp::randU](#) (size\_t D)
- `types::cmat` [qpp::randV](#) (size\_t Din, size\_t Dout)
- `std::vector< types::cmat >` [qpp::randkraus](#) (size\_t n, size\_t D)
- `types::cmat` [qpp::randH](#) (size\_t D)
- `types::ket` [qpp::randket](#) (size\_t D)
- `types::cmat` [qpp::randrho](#)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

## 7.17 include/types.h File Reference

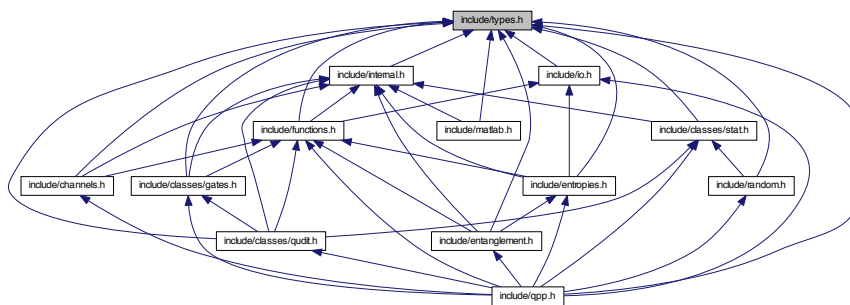
```
#include <Eigen/Dense>
```

```
#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 >