

qpp  
0.1

Generated by Doxygen 1.8.5

Sun Mar 30 2014 15:02:59



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Namespace Documentation</b>	<b>7</b>
4.1	qpp Namespace Reference . . . . .	7
4.1.1	Function Documentation . . . . .	9
4.1.1.1	_init . . . . .	9
4.1.1.2	abs . . . . .	9
4.1.1.3	absm . . . . .	10
4.1.1.4	adjoint . . . . .	10
4.1.1.5	conjugate . . . . .	10
4.1.1.6	cosm . . . . .	10
4.1.1.7	disp . . . . .	10
4.1.1.8	disp . . . . .	11
4.1.1.9	disp . . . . .	11
4.1.1.10	displn . . . . .	11
4.1.1.11	displn . . . . .	12
4.1.1.12	evals . . . . .	12
4.1.1.13	evects . . . . .	12
4.1.1.14	expm . . . . .	12
4.1.1.15	fun . . . . .	12
4.1.1.16	funm . . . . .	12
4.1.1.17	kron . . . . .	13
4.1.1.18	kron_list . . . . .	13
4.1.1.19	kron_pow . . . . .	13
4.1.1.20	load . . . . .	14

4.1.1.21	loadMATLAB	14
4.1.1.22	loadMATLAB	14
4.1.1.23	loadMATLAB	14
4.1.1.24	logm	14
4.1.1.25	norm	14
4.1.1.26	powm	15
4.1.1.27	powm_int	15
4.1.1.28	ptrace	16
4.1.1.29	ptrace2	16
4.1.1.30	ptranspose	17
4.1.1.31	rand	17
4.1.1.32	rand	17
4.1.1.33	rand_unitary	18
4.1.1.34	randn	18
4.1.1.35	randn	18
4.1.1.36	renyi	19
4.1.1.37	renyi_inf	19
4.1.1.38	reshape	19
4.1.1.39	save	19
4.1.1.40	saveMATLAB	19
4.1.1.41	saveMATLAB	19
4.1.1.42	shannon	20
4.1.1.43	sinm	20
4.1.1.44	sqrtn	21
4.1.1.45	syspermute	21
4.1.1.46	trace	21
4.1.1.47	transpose	21
4.2	qpp::ct Namespace Reference	21
4.2.1	Function Documentation	22
4.2.1.1	ii	22
4.2.1.2	omega	22
4.2.2	Variable Documentation	22
4.2.2.1	chop	22
4.2.2.2	ee	22
4.2.2.3	pi	22
4.3	qpp::gt Namespace Reference	22
4.3.1	Function Documentation	23
4.3.1.1	_init_gates	23
4.3.1.2	CU	23
4.3.1.3	CUd	23

4.3.1.4	<a href="#">Fd</a>	24
4.3.1.5	<a href="#">Rtheta</a>	24
4.3.1.6	<a href="#">TOF</a>	24
4.3.1.7	<a href="#">Xd</a>	24
4.3.1.8	<a href="#">Zd</a>	25
4.3.2	<a href="#">Variable Documentation</a>	25
4.3.2.1	<a href="#">CNOT</a>	25
4.3.2.2	<a href="#">CP</a>	25
4.3.2.3	<a href="#">H</a>	25
4.3.2.4	<a href="#">Id2</a>	25
4.3.2.5	<a href="#">S</a>	25
4.3.2.6	<a href="#">T</a>	25
4.3.2.7	<a href="#">TOF</a>	25
4.3.2.8	<a href="#">X</a>	25
4.3.2.9	<a href="#">Y</a>	25
4.3.2.10	<a href="#">Z</a>	25
4.4	<a href="#">qpp::internal Namespace Reference</a>	25
4.4.1	<a href="#">Function Documentation</a>	26
4.4.1.1	<a href="#">_check_dims</a>	26
4.4.1.2	<a href="#">_check_dims_match_mat</a>	26
4.4.1.3	<a href="#">_check_eq_dims</a>	26
4.4.1.4	<a href="#">_check_perm</a>	26
4.4.1.5	<a href="#">_check_square_mat</a>	26
4.4.1.6	<a href="#">_check_subsys</a>	26
4.4.1.7	<a href="#">_disp_container</a>	26
4.4.1.8	<a href="#">_multiidx2n</a>	26
4.4.1.9	<a href="#">_n2multiidx</a>	26
4.4.1.10	<a href="#">_ptranspose_worker</a>	26
4.4.1.11	<a href="#">_syspermute_worker</a>	27
4.5	<a href="#">qpp::stat Namespace Reference</a>	27
4.5.1	<a href="#">Variable Documentation</a>	27
4.5.1.1	<a href="#">_rd</a>	27
4.5.1.2	<a href="#">_rng</a>	27
4.6	<a href="#">qpp::types Namespace Reference</a>	27
4.6.1	<a href="#">Typedef Documentation</a>	28
4.6.1.1	<a href="#">cmat</a>	28
4.6.1.2	<a href="#">cplx</a>	28
4.6.1.3	<a href="#">dmat</a>	28
4.6.1.4	<a href="#">EigenExpression</a>	28
4.6.1.5	<a href="#">fmat</a>	28

4.6.1.6	imat	28
4.6.1.7	TemplatedEigenMatrix	28
<b>5</b>	<b>Class Documentation</b>	<b>29</b>
5.1	qpp::stat::DiscreteDistribution Class Reference	29
5.1.1	Constructor & Destructor Documentation	29
5.1.1.1	DiscreteDistribution	29
5.1.1.2	DiscreteDistribution	29
5.1.1.3	DiscreteDistribution	29
5.1.2	Member Function Documentation	29
5.1.2.1	sample	29
5.1.3	Member Data Documentation	29
5.1.3.1	_d	29
5.2	qpp::stat::DiscreteDistributionFromComplex Class Reference	30
5.2.1	Constructor & Destructor Documentation	30
5.2.1.1	DiscreteDistributionFromComplex	30
5.2.1.2	DiscreteDistributionFromComplex	30
5.2.1.3	DiscreteDistributionFromComplex	30
5.2.2	Member Function Documentation	30
5.2.2.1	sample	30
5.2.3	Member Data Documentation	30
5.2.3.1	_d	30
5.3	qpp::stat::NormalDistribution Class Reference	31
5.3.1	Constructor & Destructor Documentation	31
5.3.1.1	NormalDistribution	31
5.3.2	Member Function Documentation	31
5.3.2.1	sample	31
5.3.3	Member Data Documentation	31
5.3.3.1	_d	31
5.4	qpp::stat::UniformRealDistribution Class Reference	31
5.4.1	Constructor & Destructor Documentation	31
5.4.1.1	UniformRealDistribution	31
5.4.2	Member Function Documentation	31
5.4.2.1	sample	31
5.4.3	Member Data Documentation	32
5.4.3.1	_d	32
<b>6</b>	<b>File Documentation</b>	<b>33</b>
6.1	include/constants.h File Reference	33
6.2	include/entropy.h File Reference	34
6.3	include/functional.h File Reference	36

6.4	include/gates.h File Reference . . . . .	38
6.5	include/internal.h File Reference . . . . .	39
6.6	include/io.h File Reference . . . . .	41
6.7	include/matlab.h File Reference . . . . .	43
6.8	include/qpp.h File Reference . . . . .	43
6.9	include/random.h File Reference . . . . .	44
6.10	include/stat.h File Reference . . . . .	46
6.11	include/types.h File Reference . . . . .	47
6.12	include/util.h File Reference . . . . .	48
6.13	src/main.cpp File Reference . . . . .	50
	6.13.1 Function Documentation . . . . .	51
	6.13.1.1 main . . . . .	51
6.14	src/qpp.cpp File Reference . . . . .	51





# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">qpp</a>	7
<a href="#">qpp::ct</a>	21
<a href="#">qpp::gt</a>	22
<a href="#">qpp::internal</a>	25
<a href="#">qpp::stat</a>	27
<a href="#">qpp::types</a>	27



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">qpp::stat::DiscreteDistribution</a>	29
<a href="#">qpp::stat::DiscreteDistributionFromComplex</a>	30
<a href="#">qpp::stat::NormalDistribution</a>	31
<a href="#">qpp::stat::UniformRealDistribution</a>	31



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

<a href="#">include/constants.h</a>	33
<a href="#">include/entropy.h</a>	34
<a href="#">include/functional.h</a>	36
<a href="#">include/gates.h</a>	38
<a href="#">include/internal.h</a>	39
<a href="#">include/io.h</a>	41
<a href="#">include/matlab.h</a>	43
<a href="#">include/qpp.h</a>	43
<a href="#">include/random.h</a>	44
<a href="#">include/stat.h</a>	46
<a href="#">include/types.h</a>	47
<a href="#">include/util.h</a>	48
<a href="#">src/main.cpp</a>	50
<a href="#">src/qpp.cpp</a>	51



## Chapter 4

# Namespace Documentation

### 4.1 qpp Namespace Reference

#### Namespaces

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

#### Functions

- `template<typename MatrixType >`  
`double shannon (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`double renyi (const double alpha, const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`double renyi_inf (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::cmat funm (const types::EigenExpression< MatrixType > &A, types::cplx(*f)(const types::cplx &))`
- `template<typename MatrixType >`  
`types::cmat absm (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::cmat expm (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::cmat logm (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::cmat sqrtm (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::cmat sinm (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::cmat cosm (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::cmat powm (const types::EigenExpression< MatrixType > &A, const types::cplx z)`
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > powm\_int (const types::EigenExpression< MatrixType > &A, size_t n)`
- `template<typename MatrixType >`  
`void disp (const types::EigenExpression< MatrixType > &A, unsigned int precision=4, double chop=ct::chop, std::ostream &os=std::cout)`

- `template<>`  
`void disp (const types::EigenExpression< types::cmat > &A, unsigned int precision, double chop, std::ostream &os)`
- `template<typename MatrixType >`  
`void displn (const types::EigenExpression< MatrixType > &A, unsigned int precision=4, double chop=ct::chop, std::ostream &os=std::cout)`
- `void disp (const types::cplx c, unsigned int precision=4, double chop=ct::chop, std::ostream &os=std::cout)`
- `void displn (const types::cplx c, unsigned int precision=4, double chop=ct::chop, std::ostream &os=std::cout)`
- `template<typename MatrixType >`  
`void save (const types::EigenExpression< MatrixType > &A, const std::string &fname)`
- `template<typename MatrixType >`  
`MatrixType load (const std::string &fname)`
- `template<typename MatrixType >`  
`MatrixType loadMATLAB (const std::string &mat_file, const std::string &var_name)`
- `template<>`  
`types::dmat loadMATLAB (const std::string &mat_file, const std::string &var_name)`
- `template<>`  
`types::cmat loadMATLAB (const std::string &mat_file, const std::string &var_name)`
- `template<typename MatrixType >`  
`void saveMATLAB (const types::EigenExpression< MatrixType > &A, const std::string &mat_file, const std::string &var_name, const std::string &mode)`
- `template<>`  
`void saveMATLAB (const types::EigenExpression< types::cmat > &A, const std::string &mat_file, const std::string &var_name, const std::string &mode)`
- `int _init ()`
- `types::dmat rand (size_t rows, size_t cols)`
- `types::dmat rand (size_t rows)`
- `types::dmat randn (size_t rows, size_t cols)`
- `types::dmat randn (size_t rows)`
- `types::cmat rand_unitary (size_t size)`
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix  
< MatrixType > transpose (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix  
< MatrixType > conjugate (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix  
< MatrixType > adjoint (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`MatrixType::Scalar trace (const types::EigenExpression< MatrixType > &A)`
- `template<typename FunctionInputType , typename FunctionOutputType , typename MatrixInputType >`  
`Eigen::Matrix  
< FunctionOutputType,  
Eigen::Dynamic, Eigen::Dynamic > fun (const types::EigenExpression< MatrixInputType > &A, Function-  
OutputType(*f)(const FunctionInputType &))`
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix  
< MatrixType > abs (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`double norm (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::cmat evals (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::cmat evecs (const types::EigenExpression< MatrixType > &A)`

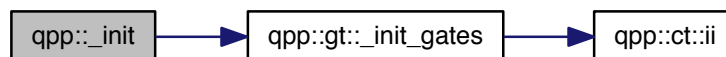


- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > kron` (const `types::EigenExpression< MatrixType > &A`, const `types::EigenExpression< MatrixType > &B`)
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > kron_list` (const `std::vector< MatrixType > &list`)
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > kron_pow` (const `types::TemplatedEigenMatrix< MatrixType > &A`, `size_t n`)
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > reshape` (const `types::EigenExpression< MatrixType > &A`, `size_t rows`, `size_t cols`)
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > syspermute` (const `types::EigenExpression< MatrixType > &A`, const `std::vector< size_t > perm`, const `std::vector< size_t > &dims`)
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > ptrace2` (const `types::EigenExpression< MatrixType > &A`, const `std::vector< size_t > dims`)
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > ptrace` (const `types::EigenExpression< MatrixType > &A`, const `std::vector< size_t > &subsys`, const `std::vector< size_t > &dims`)
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > ptranspose` (const `types::EigenExpression< MatrixType > &A`, const `std::vector< size_t > &subsys`, const `std::vector< size_t > &dims`)

### 4.1.1 Function Documentation

#### 4.1.1.1 `int qpp::_init ( )`

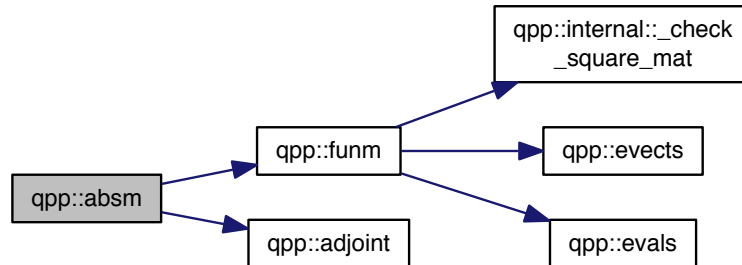
Here is the call graph for this function:



#### 4.1.1.2 `template<typename MatrixType > types::TemplatedEigenMatrix<MatrixType> qpp::abs ( const types::EigenExpression< MatrixType > & A )`

4.1.1.3 `template<typename MatrixType > types::cmat qpp::absm ( const types::EigenExpression< MatrixType > & A )`

Here is the call graph for this function:

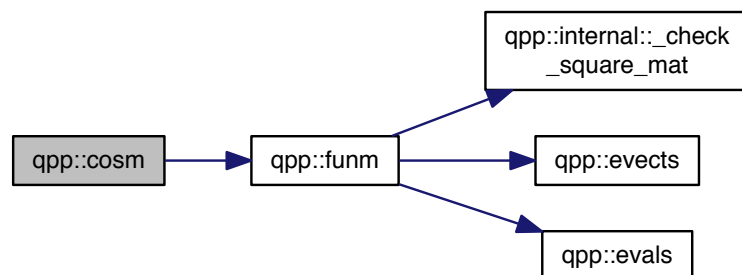


4.1.1.4 `template<typename MatrixType > types::TemplatedEigenMatrix<MatrixType> qpp::adjoint ( const types::EigenExpression< MatrixType > & A )`

4.1.1.5 `template<typename MatrixType > types::TemplatedEigenMatrix<MatrixType> qpp::conjugate ( const types::EigenExpression< MatrixType > & A )`

4.1.1.6 `template<typename MatrixType > types::cmat qpp::cosm ( const types::EigenExpression< MatrixType > & A )`

Here is the call graph for this function:



4.1.1.7 `template<typename MatrixType > void qpp::disp ( const types::EigenExpression< MatrixType > & A, unsigned int precision = 4, double chop = ct::chop, std::ostream & os = std::cout )`

4.1.1.8 `template<> void qpp::disp ( const types::EigenExpression< types::cmat > & A, unsigned int precision, double chop, std::ostream & os ) [inline]`

Here is the call graph for this function:



4.1.1.9 `void qpp::disp ( const types::cplx c, unsigned int precision = 4, double chop = ct : : chop, std::ostream & os = std::cout ) [inline]`

Here is the call graph for this function:



4.1.1.10 `template<typename MatrixType> void qpp::displn ( const types::EigenExpression< MatrixType > & A, unsigned int precision = 4, double chop = ct : : chop, std::ostream & os = std::cout )`

Here is the call graph for this function:



4.1.1.11 `void qpp::displn ( const types::cplx c, unsigned int precision = 4, double chop = ct::chop, std::ostream & os = std::cout ) [inline]`

Here is the call graph for this function:

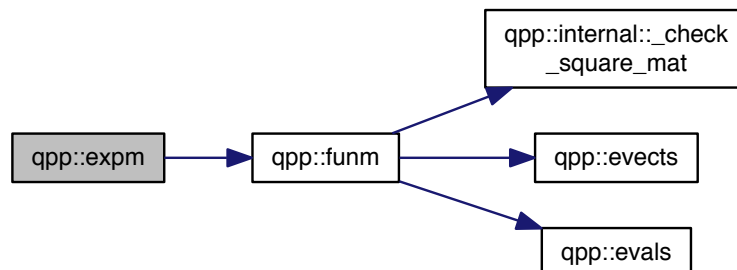


4.1.1.12 `template<typename MatrixType> types::cmat qpp::evals ( const types::EigenExpression< MatrixType> & A )`

4.1.1.13 `template<typename MatrixType> types::cmat qpp::evects ( const types::EigenExpression< MatrixType> & A )`

4.1.1.14 `template<typename MatrixType> types::cmat qpp::expm ( const types::EigenExpression< MatrixType> & A )`

Here is the call graph for this function:



4.1.1.15 `template<typename FunctionInputType, typename FunctionOutputType, typename MatrixInputType> Eigen::Matrix<FunctionOutputType, Eigen::Dynamic, Eigen::Dynamic> qpp::fun ( const types::EigenExpression< MatrixInputType> & A, FunctionOutputType(*) (const FunctionInputType &) f )`

4.1.1.16 `template<typename MatrixType> types::cmat qpp::funm ( const types::EigenExpression< MatrixType> & A, types::cplx(*) (const types::cplx &) f )`

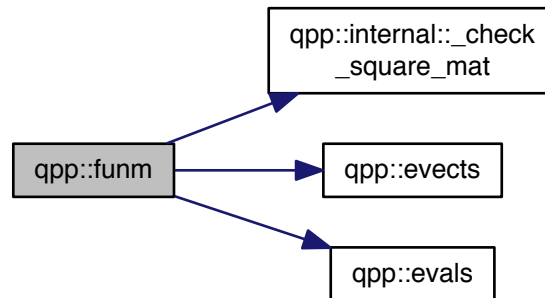
#### Parameters

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

Returns

[types::cmat](#)

Here is the call graph for this function:



4.1.1.17 `template<typename MatrixType> types::TemplatedEigenMatrix<MatrixType> qpp::kron ( const types::EigenExpression< MatrixType> & A, const types::EigenExpression< MatrixType> & B )`

4.1.1.18 `template<typename MatrixType> types::TemplatedEigenMatrix<MatrixType> qpp::kron_list ( const std::vector< MatrixType> & list )`

Here is the call graph for this function:



4.1.1.19 `template<typename MatrixType> types::TemplatedEigenMatrix<MatrixType> qpp::kron_pow ( const types::TemplatedEigenMatrix< MatrixType> & A, size_t n )`

Here is the call graph for this function:



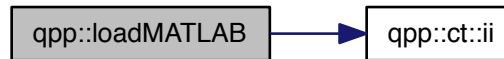
4.1.1.20 `template<typename MatrixType > MatrixType qpp::load ( const std::string & fname )`

4.1.1.21 `template<typename MatrixType > MatrixType qpp::loadMATLAB ( const std::string & mat_file, const std::string & var_name )`

4.1.1.22 `template<> types::dmat qpp::loadMATLAB ( const std::string & mat_file, const std::string & var_name )`  
[inline]

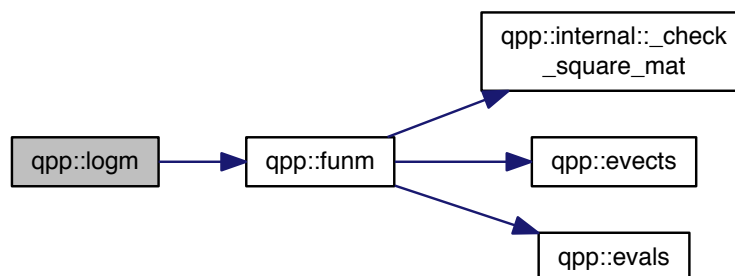
4.1.1.23 `template<> types::cmat qpp::loadMATLAB ( const std::string & mat_file, const std::string & var_name )`  
[inline]

Here is the call graph for this function:



4.1.1.24 `template<typename MatrixType > types::cmat qpp::logm ( const types::EigenExpression< MatrixType > & A )`

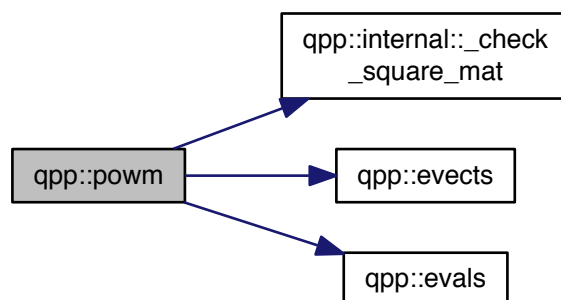
Here is the call graph for this function:



4.1.1.25 `template<typename MatrixType > double qpp::norm ( const types::EigenExpression< MatrixType > & A )`

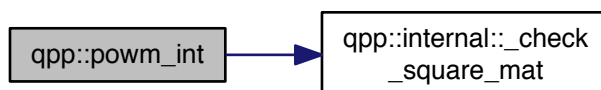
4.1.1.26 `template<typename MatrixType> types::cmat qpp::powm ( const types::EigenExpression< MatrixType> & A,  
const types::cplx z )`

Here is the call graph for this function:



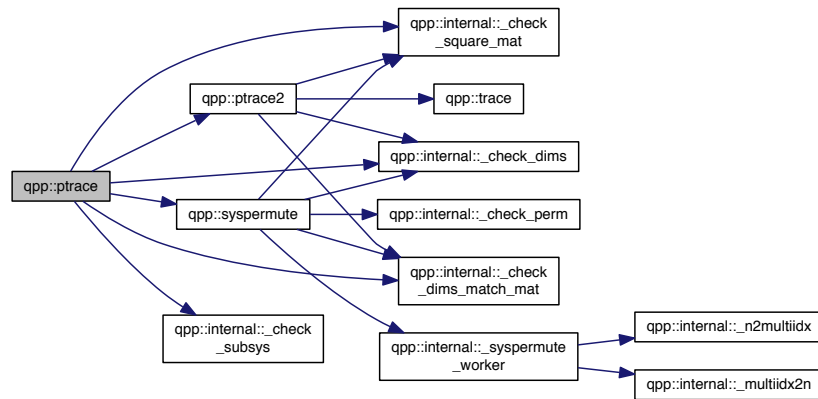
4.1.1.27 `template<typename MatrixType> types::TemplatedEigenMatrix<MatrixType> qpp::powm_int ( const  
types::EigenExpression< MatrixType> & A, size_t n )`

Here is the call graph for this function:



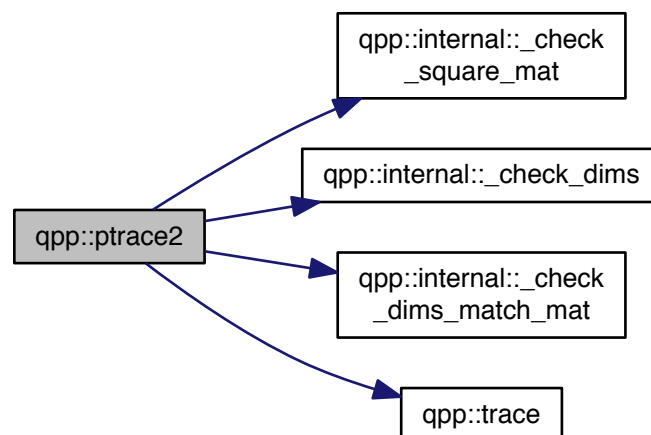
4.1.1.28 `template<typename MatrixType> types::TemplatedEigenMatrix<MatrixType> qpp::ptrace ( const types::EigenExpression< MatrixType> & A, const std::vector< size_t> & subsys, const std::vector< size_t> & dims )`

Here is the call graph for this function:



4.1.1.29 `template<typename MatrixType> types::TemplatedEigenMatrix<MatrixType> qpp::ptrace2 ( const types::EigenExpression< MatrixType> & A, const std::vector< size_t> dims )`

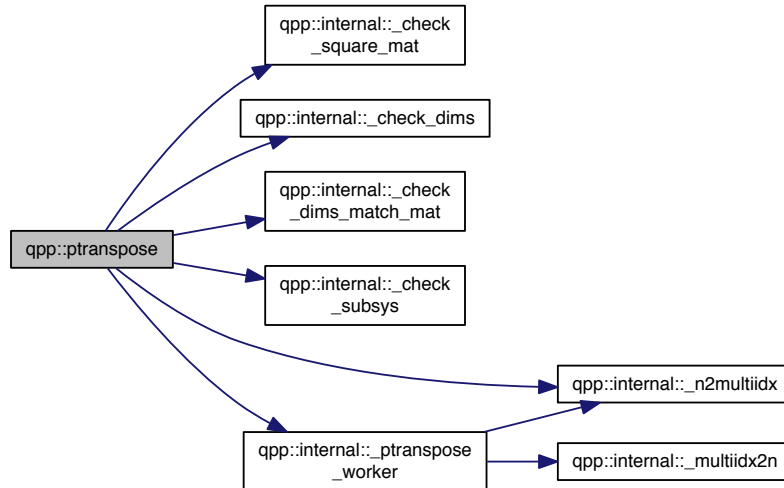
Here is the call graph for this function:





4.1.1.30 `template<typename MatrixType> types::TemplatedEigenMatrix<MatrixType> qpp::ptranspose ( const types::EigenExpression< MatrixType> & A, const std::vector< size_t> & subsys, const std::vector< size_t> & dims )`

Here is the call graph for this function:



4.1.1.31 `types::dmat qpp::rand ( size_t rows, size_t cols ) [inline]`

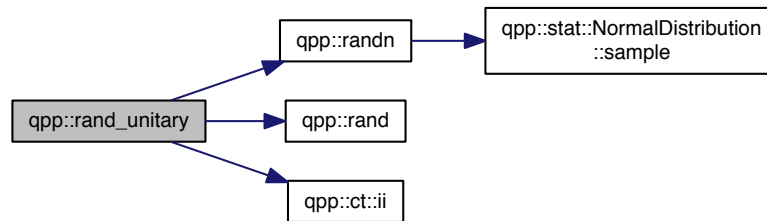
4.1.1.32 `types::dmat qpp::rand ( size_t rows ) [inline]`

Here is the call graph for this function:



#### 4.1.1.33 `types::cmat qpp::rand_unitary ( size_t size ) [inline]`

Here is the call graph for this function:



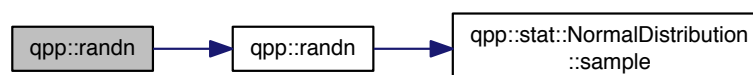
#### 4.1.1.34 `types::dmat qpp::randn ( size_t rows, size_t cols ) [inline]`

Here is the call graph for this function:



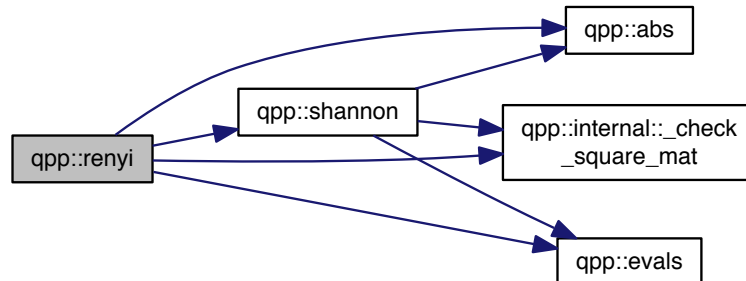
#### 4.1.1.35 `types::dmat qpp::randn ( size_t rows ) [inline]`

Here is the call graph for this function:



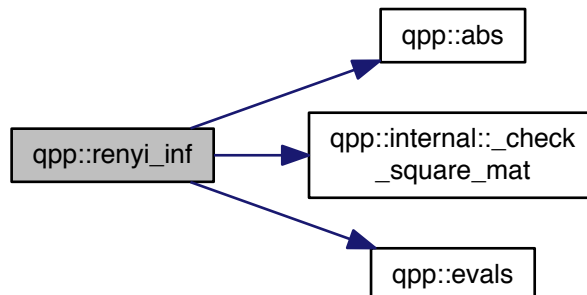
4.1.1.36 `template<typename MatrixType > double qpp::renyi ( const double alpha, const types::EigenExpression< MatrixType > & A )`

Here is the call graph for this function:



4.1.1.37 `template<typename MatrixType > double qpp::renyi_inf ( const types::EigenExpression< MatrixType > & A )`

Here is the call graph for this function:



4.1.1.38 `template<typename MatrixType > types::TemplatedEigenMatrix<MatrixType> qpp::reshape ( const types::EigenExpression< MatrixType > & A, size_t rows, size_t cols )`

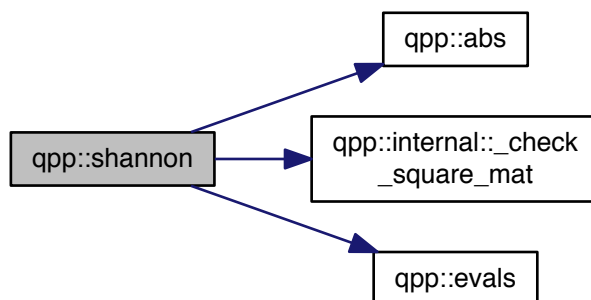
4.1.1.39 `template<typename MatrixType > void qpp::save ( const types::EigenExpression< MatrixType > & A, const std::string & fname )`

4.1.1.40 `template<typename MatrixType > void qpp::saveMATLAB ( const types::EigenExpression< MatrixType > & A, const std::string & mat_file, const std::string & var_name, const std::string & mode )`

4.1.1.41 `template<> void qpp::saveMATLAB ( const types::EigenExpression< types::cmat > & A, const std::string & mat_file, const std::string & var_name, const std::string & mode ) [inline]`

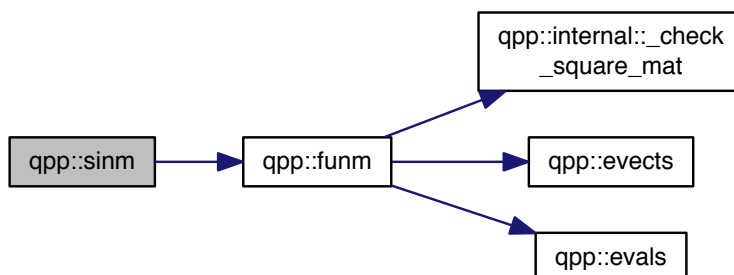
4.1.1.42 `template<typename MatrixType > double qpp::shannon ( const types::EigenExpression< MatrixType > & A )`

Here is the call graph for this function:



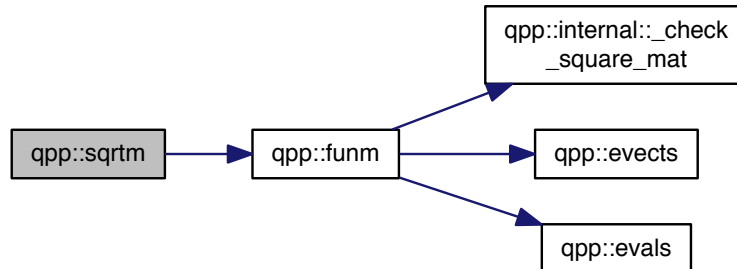
4.1.1.43 `template<typename MatrixType > types::cmat qpp::sinm ( const types::EigenExpression< MatrixType > & A )`

Here is the call graph for this function:



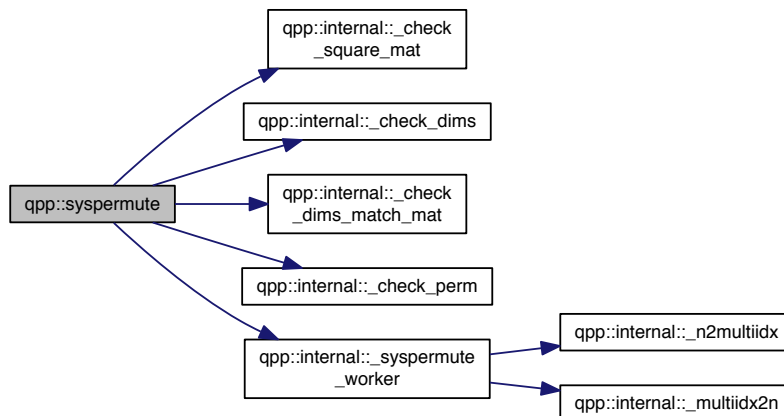
4.1.1.44 `template<typename MatrixType> types::cmat qpp::sqrtm ( const types::EigenExpression< MatrixType> & A )`

Here is the call graph for this function:



4.1.1.45 `template<typename MatrixType> types::TemplatedEigenMatrix<MatrixType> qpp::syspermute ( const types::EigenExpression< MatrixType> & A, const std::vector< size_t> perm, const std::vector< size_t> & dims )`

Here is the call graph for this function:



4.1.1.46 `template<typename MatrixType> MatrixType::Scalar qpp::trace ( const types::EigenExpression< MatrixType> & A )`

4.1.1.47 `template<typename MatrixType> types::TemplatedEigenMatrix<MatrixType> qpp::transpose ( const types::EigenExpression< MatrixType> & A )`

## 4.2 qpp::ct Namespace Reference

## Functions

- const `types::cplx ii` (0, 1)
- `types::cplx omega` (size\_t D)

## Variables

- const double `chop` = 1e-10
- const double `pi` = 3.141592653589793238462643383279502884
- const double `ee` = 2.718281828459045235360287471352662497

### 4.2.1 Function Documentation

4.2.1.1 `const types::cplx qpp::ct::ii ( 0 , 1 )`

4.2.1.2 `types::cplx qpp::ct::omega ( size_t D )` `[inline]`

Here is the call graph for this function:



### 4.2.2 Variable Documentation

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

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

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

## 4.3 qpp::gt Namespace Reference

### Functions

- void `_init_gates` ()
- `types::cmat Rtheta` (double theta)
- `types::cmat CU` (const `types::cmat` &U)
- `types::cmat Zd` (size\_t D)
- `types::cmat Fd` (size\_t D)
- `types::cmat Xd` (size\_t D)
- `types::cmat CUd` (const `types::cmat` &U)
- `types::cmat TOF` (8, 8)

## Variables

- [types::cmat H](#)
- [types::cmat Id2](#)
- [types::cmat X](#)
- [types::cmat Y](#)
- [types::cmat Z](#)
- [types::cmat S](#)
- [types::cmat T](#)
- [types::cmat CNOT](#)
- [types::cmat CP](#)
- [types::cmat TOF](#)

## 4.3.1 Function Documentation

## 4.3.1.1 void qpp::gt::\_init\_gates ( ) [inline]

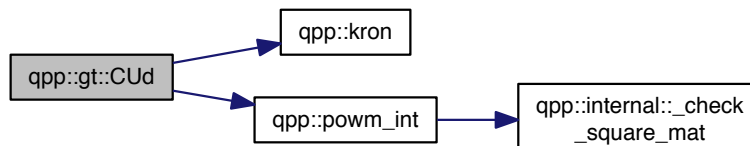
Here is the call graph for this function:



## 4.3.1.2 types::cmat qpp::gt::CU ( const types::cmat &amp; U ) [inline]

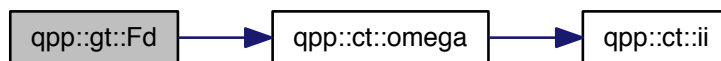
## 4.3.1.3 types::cmat qpp::gt::CUd ( const types::cmat &amp; U ) [inline]

Here is the call graph for this function:



#### 4.3.1.4 `types::cmat qpp::gt::Fd ( size_t D ) [inline]`

Here is the call graph for this function:



#### 4.3.1.5 `types::cmat qpp::gt::Rtheta ( double theta ) [inline]`

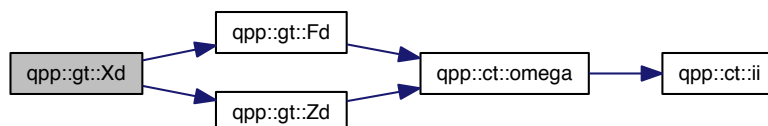
Here is the call graph for this function:



#### 4.3.1.6 `types::cmat qpp::gt::TOF ( 8, 8 )`

#### 4.3.1.7 `types::cmat qpp::gt::Xd ( size_t D ) [inline]`

Here is the call graph for this function:





4.3.1.8 `types::cmat qpp::gt::Zd ( size_t D ) [inline]`

Here is the call graph for this function:



## 4.3.2 Variable Documentation

4.3.2.1 `types::cmat qpp::gt::CNOT`4.3.2.2 `types::cmat qpp::gt::CP`4.3.2.3 `types::cmat qpp::gt::H`4.3.2.4 `types::cmat qpp::gt::Id2`4.3.2.5 `types::cmat qpp::gt::S`4.3.2.6 `types::cmat qpp::gt::T`4.3.2.7 `types::cmat qpp::gt::TOF`4.3.2.8 `types::cmat qpp::gt::X`4.3.2.9 `types::cmat qpp::gt::Y`4.3.2.10 `types::cmat qpp::gt::Z`4.4 `qpp::internal` Namespace Reference

## Functions

- `template<typename T >`  
`void _disp_container (const T &x)`
- `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 MatrixType >`  
`bool _check_square_mat (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`bool _check_dims_match_mat (const std::vector< size_t > &dims, const types::EigenExpression< MatrixType > &A)`
- `bool _check_dims (const std::vector< size_t > &dims)`
- `bool _check_eq_dims (const std::vector< size_t > &dims, size_t dim)`
- `bool _check_subsys (const std::vector< size_t > &subsys, const std::vector< size_t > &dims)`
- `bool _check_perm (const std::vector< size_t > &perm, const std::vector< size_t > &dims)`

- `template<typename MatrixType >`  
`void __syspermute_worker (size_t numdims, const size_t *cdims, const size_t *cperm, size_t i, size_t j, size_t &iperm, size_t &jperm, const types::EigenExpression< MatrixType > &A, types::EigenExpression< MatrixType > &result)`
- `template<typename MatrixType >`  
`void __ptranspose_worker (const size_t *midxrow, size_t numdims, size_t numsubsys, const size_t *cdims, const size_t *csubsys, size_t i, size_t j, size_t &iperm, size_t &jperm, const types::EigenExpression< MatrixType > &A, types::EigenExpression< MatrixType > &result)`

#### 4.4.1 Function Documentation

4.4.1.1 `bool qpp::internal::_check_dims ( const std::vector< size_t > & dims ) [inline]`

4.4.1.2 `template<typename MatrixType > bool qpp::internal::_check_dims_match_mat ( const std::vector< size_t > & dims, const types::EigenExpression< MatrixType > & A )`

4.4.1.3 `bool qpp::internal::_check_eq_dims ( const std::vector< size_t > & dims, size_t dim ) [inline]`

4.4.1.4 `bool qpp::internal::_check_perm ( const std::vector< size_t > & perm, const std::vector< size_t > & dims ) [inline]`

4.4.1.5 `template<typename MatrixType > bool qpp::internal::_check_square_mat ( const types::EigenExpression< MatrixType > & A )`

4.4.1.6 `bool qpp::internal::_check_subsys ( const std::vector< size_t > & subsys, const std::vector< size_t > & dims ) [inline]`

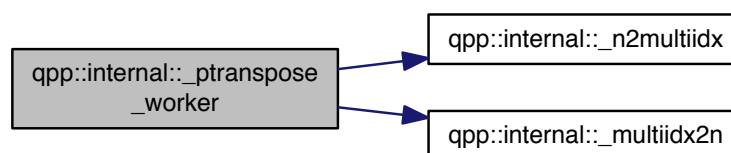
4.4.1.7 `template<typename T > void qpp::internal::_disp_container ( const T & x )`

4.4.1.8 `size_t qpp::internal::_multiidx2n ( const size_t * midx, size_t numdims, const size_t * dims ) [inline]`

4.4.1.9 `void qpp::internal::_n2multiidx ( size_t n, size_t numdims, const size_t * dims, size_t * result ) [inline]`

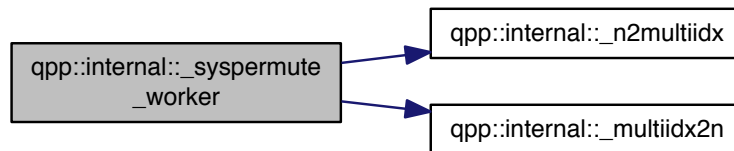
4.4.1.10 `template<typename MatrixType > void qpp::internal::_ptranspose_worker ( const size_t * midxrow, size_t numdims, size_t numsubsys, const size_t * cdims, const size_t * csubsys, size_t i, size_t j, size_t & iperm, size_t & jperm, const types::EigenExpression< MatrixType > & A, types::EigenExpression< MatrixType > & result ) [inline]`

Here is the call graph for this function:



4.4.1.11 `template<typename MatrixType > void qpp::internal::_syspermute_worker ( size_t numdims, const size_t * cdims, const size_t * cperm, size_t i, size_t j, size_t & iperm, size_t & jperm, const types::EigenExpression< MatrixType > & A, types::EigenExpression< MatrixType > & result ) [inline]`

Here is the call graph for this function:



## 4.5 qpp::stat Namespace Reference

### Classes

- class [NormalDistribution](#)
- class [UniformRealDistribution](#)
- class [DiscreteDistribution](#)
- class [DiscreteDistributionFromComplex](#)

### Variables

- `std::random_device` [\\_rd](#)
- `std::mt19937` [\\_rng](#)

### 4.5.1 Variable Documentation

4.5.1.1 `std::random_device` [qpp::stat::\\_rd](#)

4.5.1.2 `std::mt19937` [qpp::stat::\\_rng](#)

## 4.6 qpp::types Namespace Reference

### Typedefs

- `typedef std::complex< double >` [cplx](#)
- `typedef Eigen::MatrixXcd` [cmat](#)
- `typedef Eigen::MatrixXd` [dmat](#)
- `typedef Eigen::MatrixXf` [fmat](#)
- `typedef Eigen::MatrixXi` [imat](#)
- `template<typename Derived >`  
using [EigenExpression](#) = `Eigen::MatrixBase< Derived >`
- `template<typename MatrixType >`  
using [TemplatedEigenMatrix](#) = `Eigen::Matrix< typename MatrixType::Scalar, Eigen::Dynamic, Eigen::Dynamic >`

#### 4.6.1 Typedef Documentation

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

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

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

4.6.1.4 `template<typename Derived > using qpp::types::EigenExpression = typedef Eigen::MatrixBase<Derived>`

4.6.1.5 `typedef Eigen::MatrixXf qpp::types::fmat`

4.6.1.6 `typedef Eigen::MatrixXi qpp::types::imat`

4.6.1.7 `template<typename MatrixType > using qpp::types::TemplatedEigenMatrix = typedef Eigen::Matrix<typename MatrixType::Scalar, Eigen::Dynamic, Eigen::Dynamic>`

## Chapter 5

# Class Documentation

### 5.1 qpp::stat::DiscreteDistribution Class Reference

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

#### Public Member Functions

- [DiscreteDistribution](#) (std::initializer\_list< double > weights)
- template<typename InputIterator >  
[DiscreteDistribution](#) (InputIterator first, InputIterator last)
- [DiscreteDistribution](#) (std::vector< double > weights)
- size\_t [sample](#) ()

#### Public Attributes

- std::discrete\_distribution  
< size\_t > [\\_d](#)

#### 5.1.1 Constructor & Destructor Documentation

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

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

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

#### 5.1.2 Member Function Documentation

5.1.2.1 `size_t qpp::stat::DiscreteDistribution::sample ( ) [inline]`

#### 5.1.3 Member Data Documentation

5.1.3.1 `std::discrete_distribution<size_t> qpp::stat::DiscreteDistribution::_d`

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

- include/[stat.h](#)

## 5.2 qpp::stat::DiscreteDistributionFromComplex Class Reference

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

### Public Member Functions

- [DiscreteDistributionFromComplex](#) (std::initializer\_list< [types::cplx](#) > amplitudes)
- template<typename InputIterator >  
  [DiscreteDistributionFromComplex](#) (InputIterator first, InputIterator last)
- [DiscreteDistributionFromComplex](#) (std::vector< double > weights)
- size\_t [sample](#) ()

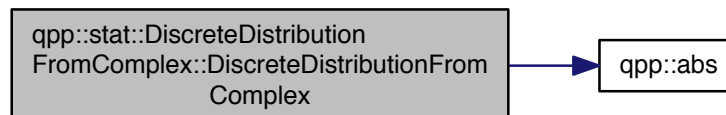
### Public Attributes

- std::discrete\_distribution  
  < size\_t > [\\_d](#)

### 5.2.1 Constructor & Destructor Documentation

5.2.1.1 `qpp::stat::DiscreteDistributionFromComplex::DiscreteDistributionFromComplex ( std::initializer_list< types::cplx > amplitudes ) [inline]`

Here is the call graph for this function:



5.2.1.2 `template<typename InputIterator > qpp::stat::DiscreteDistributionFromComplex::DiscreteDistributionFromComplex ( InputIterator first, InputIterator last ) [inline]`

5.2.1.3 `qpp::stat::DiscreteDistributionFromComplex::DiscreteDistributionFromComplex ( std::vector< double > weights ) [inline]`

### 5.2.2 Member Function Documentation

5.2.2.1 `size_t qpp::stat::DiscreteDistributionFromComplex::sample ( ) [inline]`

### 5.2.3 Member Data Documentation

5.2.3.1 `std::discrete_distribution<size_t> qpp::stat::DiscreteDistributionFromComplex::_d`

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

- include/[stat.h](#)

## 5.3 qpp::stat::NormalDistribution Class Reference

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

### Public Member Functions

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

### Public Attributes

- std::normal\_distribution [\\_d](#)

#### 5.3.1 Constructor & Destructor Documentation

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

#### 5.3.2 Member Function Documentation

5.3.2.1 `double qpp::stat::NormalDistribution::sample ( )` [inline]

#### 5.3.3 Member Data Documentation

5.3.3.1 `std::normal_distribution qpp::stat::NormalDistribution::_d`

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

- include/[stat.h](#)

## 5.4 qpp::stat::UniformRealDistribution Class Reference

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

### Public Member Functions

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

### Public Attributes

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

#### 5.4.1 Constructor & Destructor Documentation

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

#### 5.4.2 Member Function Documentation

5.4.2.1 `double qpp::stat::UniformRealDistribution::sample ( )` [inline]

### 5.4.3 Member Data Documentation

#### 5.4.3.1 `std::uniform_real_distribution qpp::stat::UniformRealDistribution::_d`

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

- `include/stat.h`



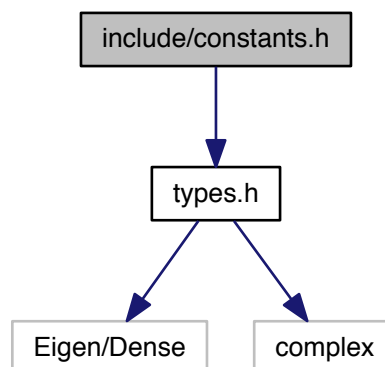
## Chapter 6

# File Documentation

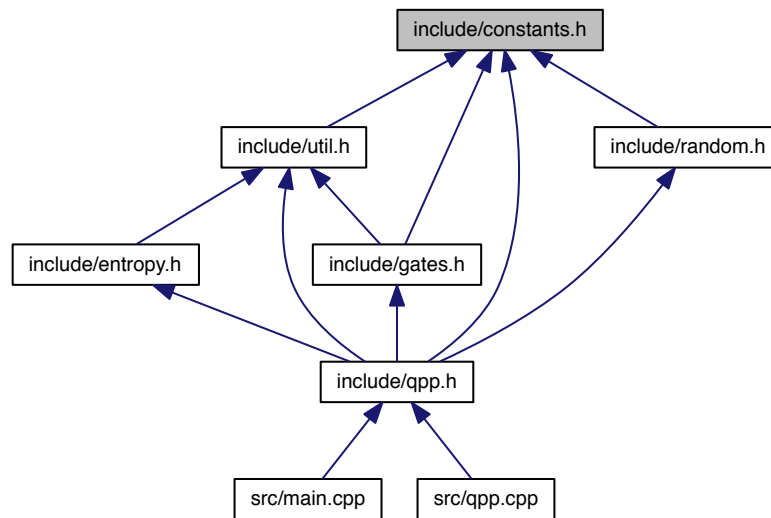
### 6.1 include/constants.h File Reference

```
#include "types.h"
```

Include dependency graph for constants.h:



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



## Namespaces

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

## Functions

- `const types::cplx qpp::ct::ii (0, 1)`
- `types::cplx qpp::ct::omega (size_t D)`

## Variables

- `const double qpp::ct::chop = 1e-10`
- `const double qpp::ct::pi = 3.141592653589793238462643383279502884`
- `const double qpp::ct::ee = 2.718281828459045235360287471352662497`

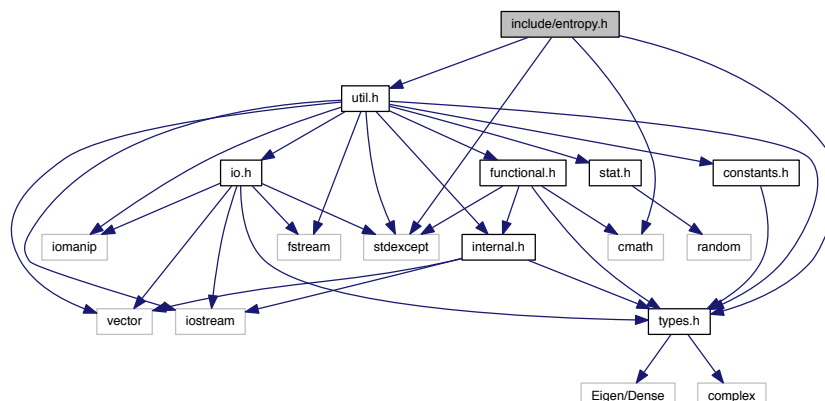
## 6.2 include/entropy.h File Reference

```

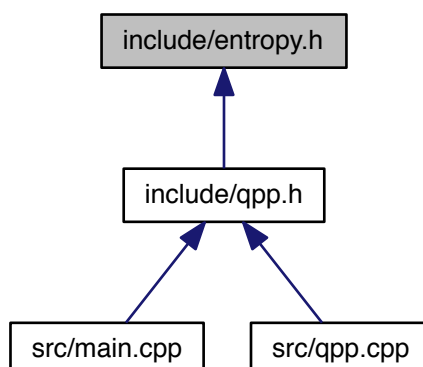
#include <cmath>
#include <stdexcept>
#include "types.h"
#include "util.h"

```

Include dependency graph for entropy.h:



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



## Namespaces

- [qpp](#)

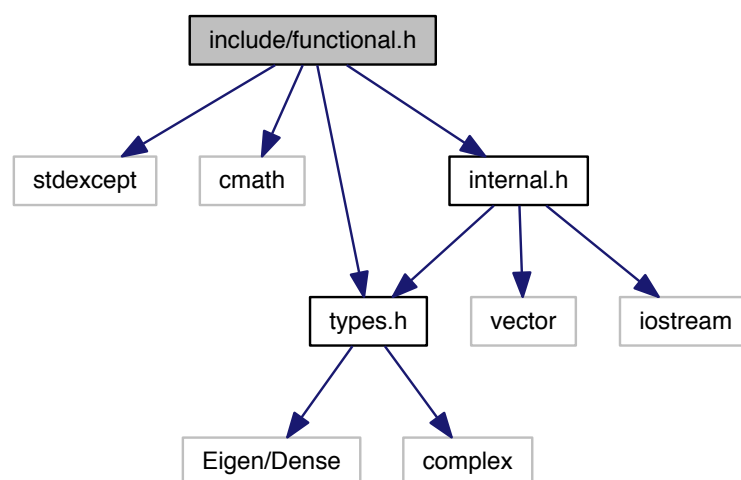
## Functions

- `template<typename MatrixType >`  
`double qpp::shannon (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`double qpp::renyi (const double alpha, const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`double qpp::renyi\_inf (const types::EigenExpression< MatrixType > &A)`

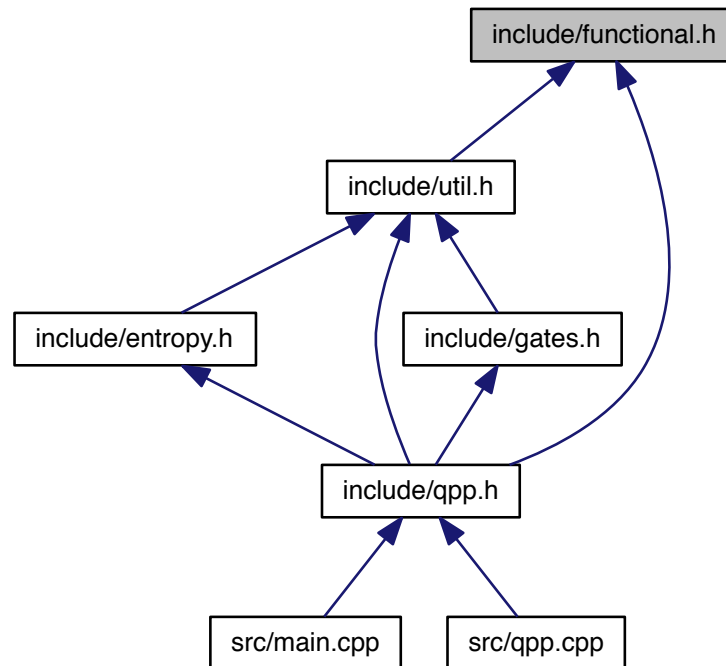
### 6.3 include/functional.h File Reference

```
#include <stdexcept>
#include <cmath>
#include "types.h"
#include "internal.h"
```

Include dependency graph for functional.h:



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



## Namespaces

- [qpp](#)

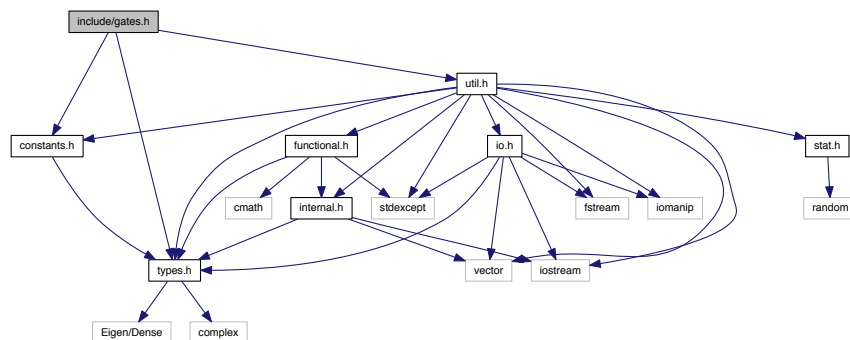
## Functions

- `template<typename MatrixType >`  
`types::cmat qpp::fun (const types::EigenExpression< MatrixType > &A, types::cplx(*f)(const types::cplx &))`
- `template<typename MatrixType >`  
`types::cmat qpp::absm (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::cmat qpp::expm (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::cmat qpp::logm (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::cmat qpp::sqrtm (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::cmat qpp::sinm (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::cmat qpp::cosm (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::cmat qpp::powm (const types::EigenExpression< MatrixType > &A, const types::cplx z)`

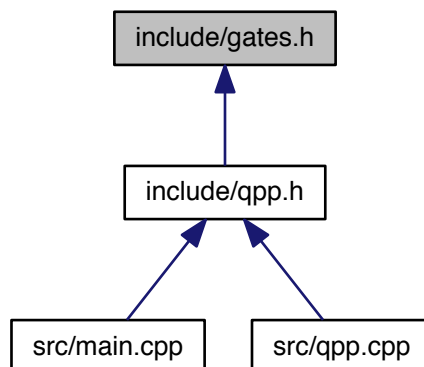
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > qpp::powm\_int (const types::EigenExpression< MatrixType > &A, size_t n)`

## 6.4 include/gates.h File Reference

```
#include "types.h"
#include "constants.h"
#include "util.h"
Include dependency graph for gates.h:
```



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



## Namespaces

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

## Functions

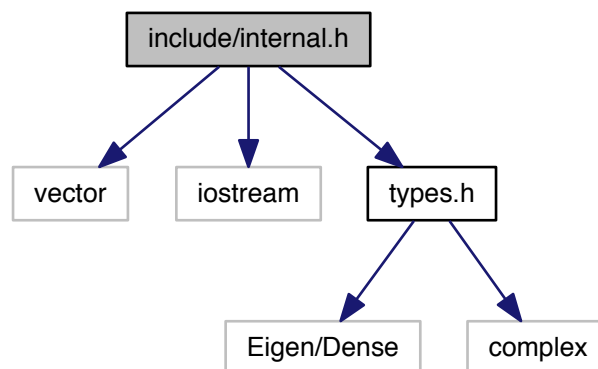
- void `qpp::gt::_init_gates` ()
- `types::cmat` `qpp::gt::Rtheta` (double theta)
- `types::cmat` `qpp::gt::CU` (const `types::cmat` &U)
- `types::cmat` `qpp::gt::Zd` (size\_t D)
- `types::cmat` `qpp::gt::Fd` (size\_t D)
- `types::cmat` `qpp::gt::Xd` (size\_t D)
- `types::cmat` `qpp::gt::CUd` (const `types::cmat` &U)

## Variables

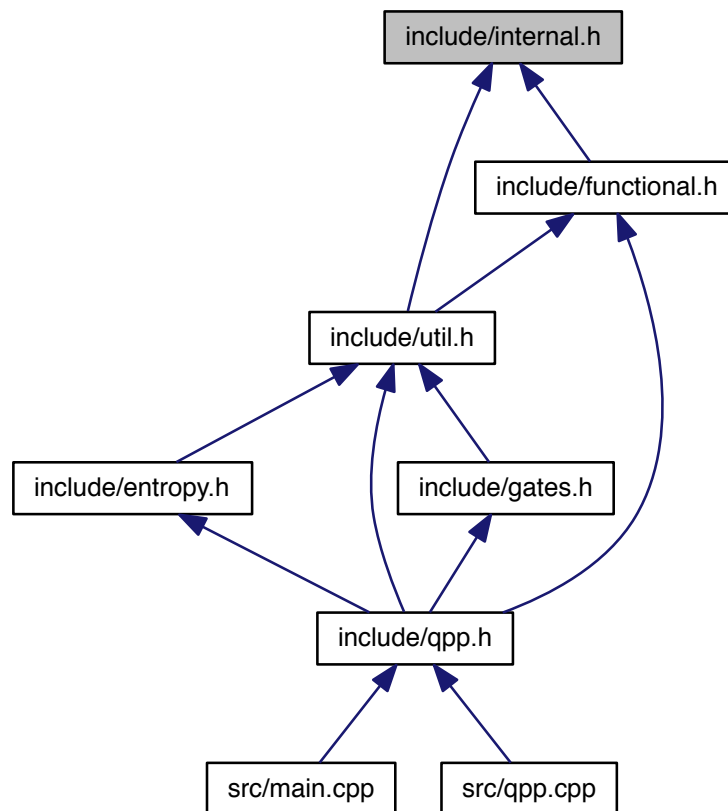
- `types::cmat` `qpp::gt::H`
- `types::cmat` `qpp::gt::Id2`
- `types::cmat` `qpp::gt::X`
- `types::cmat` `qpp::gt::Y`
- `types::cmat` `qpp::gt::Z`
- `types::cmat` `qpp::gt::S`
- `types::cmat` `qpp::gt::T`
- `types::cmat` `qpp::gt::CNOT`
- `types::cmat` `qpp::gt::CP`
- `types::cmat` `qpp::gt::TOF`

## 6.5 include/internal.h File Reference

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



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



## Namespaces

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

## Functions

- `template<typename T >`  
void [qpp::internal::\\_disp\\_container](#) (const T &x)
- 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 MatrixType >`  
bool [qpp::internal::\\_check\\_square\\_mat](#) (const types::EigenExpression< MatrixType > &A)
- `template<typename MatrixType >`  
bool [qpp::internal::\\_check\\_dims\\_match\\_mat](#) (const std::vector< size\_t > &dims, const types::EigenExpression< MatrixType > &A)
- bool [qpp::internal::\\_check\\_dims](#) (const std::vector< size\_t > &dims)
- bool [qpp::internal::\\_check\\_eq\\_dims](#) (const std::vector< size\_t > &dims, size\_t dim)
- bool [qpp::internal::\\_check\\_subsys](#) (const std::vector< size\_t > &subsys, const std::vector< size\_t > &dims)
- bool [qpp::internal::\\_check\\_perm](#) (const std::vector< size\_t > &perm, const std::vector< size\_t > &dims)

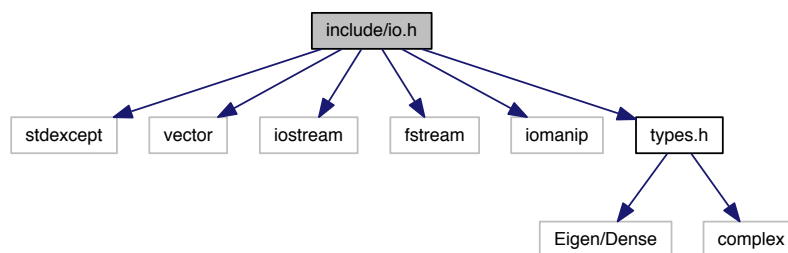


- `template<typename MatrixType >`  
`void qpp::internal::\_syspermute\_worker (size_t numdims, const size_t *cdims, const size_t *cperm, size_t i, size_t j, size_t &iperm, size_t &jperm, const types::EigenExpression< MatrixType > &A, types::EigenExpression< MatrixType > &result)`
- `template<typename MatrixType >`  
`void qpp::internal::\_ptranspose\_worker (const size_t *midxrow, size_t numdims, size_t numsubsys, const size_t *cdims, const size_t *csubsys, size_t i, size_t j, size_t &iperm, size_t &jperm, const types::EigenExpression< MatrixType > &A, types::EigenExpression< MatrixType > &result)`

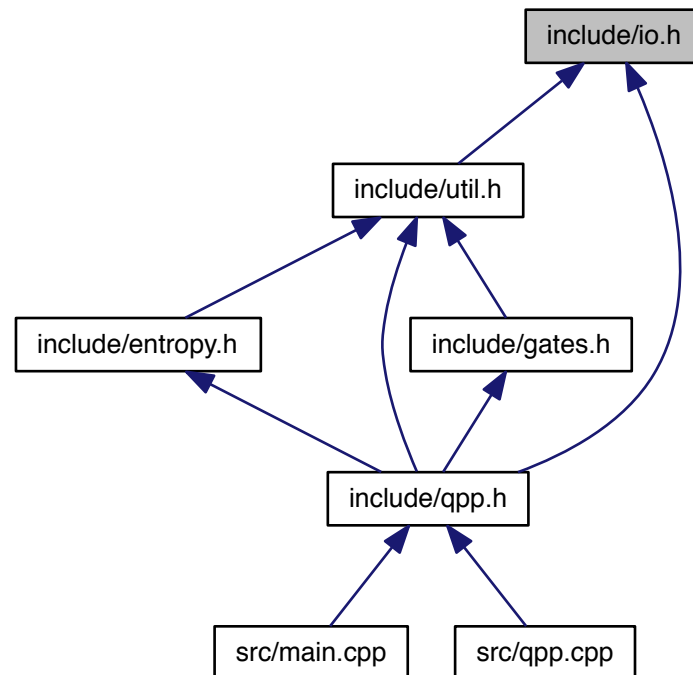
## 6.6 include/io.h File Reference

```
#include <stdexcept>
#include <vector>
#include <iostream>
#include <fstream>
#include <iomanip>
#include "types.h"
```

Include dependency graph for io.h:



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



## Namespaces

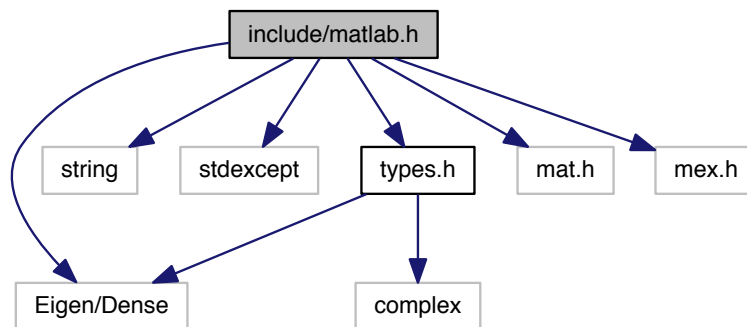
- [qpp](#)

## Functions

- `template<typename MatrixType >`  
`void qpp::disp (const types::EigenExpression< MatrixType > &A, unsigned int precision=4, double chop=ct::chop, std::ostream &os=std::cout)`
- `template<>`  
`void qpp::disp (const types::EigenExpression< types::cmat > &A, unsigned int precision, double chop, std::ostream &os)`
- `template<typename MatrixType >`  
`void qpp::displn (const types::EigenExpression< MatrixType > &A, unsigned int precision=4, double chop=ct::chop, std::ostream &os=std::cout)`
- `void qpp::disp (const types::cplx c, unsigned int precision=4, double chop=ct::chop, std::ostream &os=std::cout)`
- `void qpp::displn (const types::cplx c, unsigned int precision=4, double chop=ct::chop, std::ostream &os=std::cout)`
- `template<typename MatrixType >`  
`void qpp::save (const types::EigenExpression< MatrixType > &A, const std::string &fname)`
- `template<typename MatrixType >`  
`MatrixType qpp::load (const std::string &fname)`

## 6.7 include/matlab.h File Reference

```
#include <Eigen/Dense>
#include <string>
#include <stdexcept>
#include "types.h"
#include "mat.h"
#include "mex.h"
Include dependency graph for matlab.h:
```



### Namespaces

- [qpp](#)

### Functions

- `template<typename MatrixType >`  
`MatrixType qpp::loadMATLAB (const std::string &mat_file, const std::string &var_name)`
- `template<>`  
`types::dmat qpp::loadMATLAB (const std::string &mat_file, const std::string &var_name)`
- `template<>`  
`types::cmat qpp::loadMATLAB (const std::string &mat_file, const std::string &var_name)`
- `template<typename MatrixType >`  
`void qpp::saveMATLAB (const types::EigenExpression< MatrixType > &A, const std::string &mat_file, const std::string &var_name, const std::string &mode)`
- `template<>`  
`void qpp::saveMATLAB (const types::EigenExpression< types::cmat > &A, const std::string &mat_file, const std::string &var_name, const std::string &mode)`

## 6.8 include/qpp.h File Reference

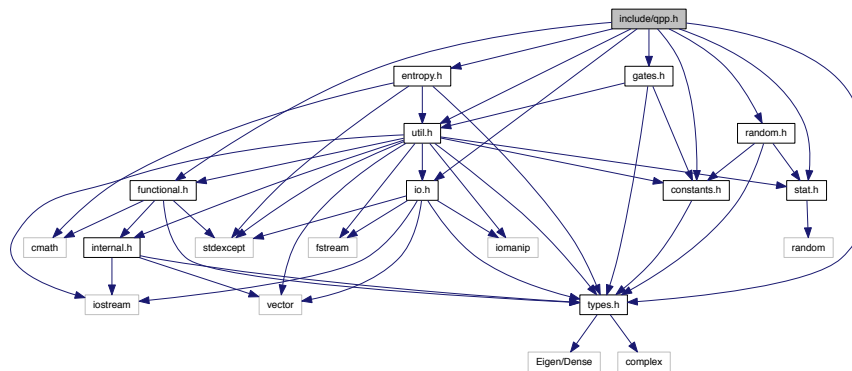
```
#include "types.h"
```

```

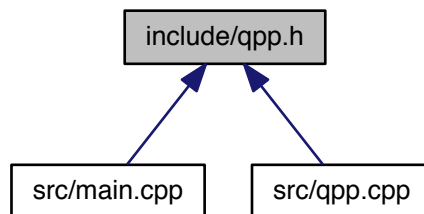
#include "util.h"
#include "constants.h"
#include "gates.h"
#include "stat.h"
#include "functional.h"
#include "random.h"
#include "entropy.h"
#include "io.h"

```

Include dependency graph for qpp.h:



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



## Namespaces

- `qpp`

## Functions

- `int qpp::_init ()`

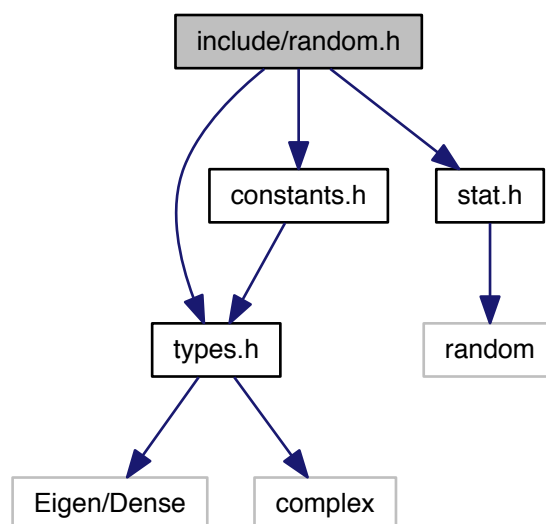
## 6.9 include/random.h File Reference

```

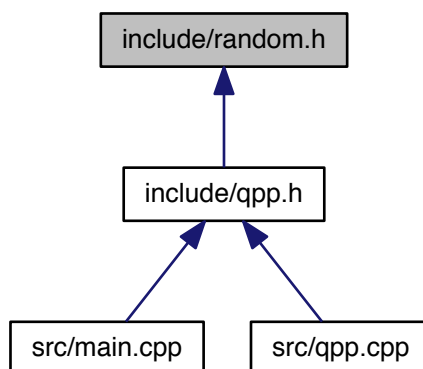
#include "types.h"

```

```
#include "stat.h"  
#include "constants.h"  
Include dependency graph for random.h:
```



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



## Namespaces

- [qpp](#)

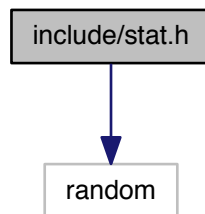
## Functions

- `types::dmat qpp::rand` (`size_t` rows, `size_t` cols)
- `types::dmat qpp::rand` (`size_t` rows)
- `types::dmat qpp::randn` (`size_t` rows, `size_t` cols)
- `types::dmat qpp::randn` (`size_t` rows)
- `types::cmat qpp::rand_unitary` (`size_t` size)

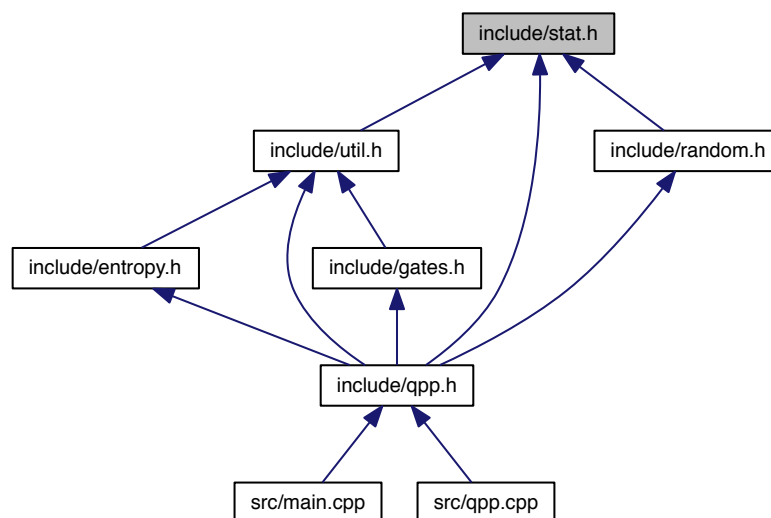
## 6.10 include/stat.h File Reference

```
#include <random>
```

Include dependency graph for stat.h:



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



## Classes

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

## Namespaces

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

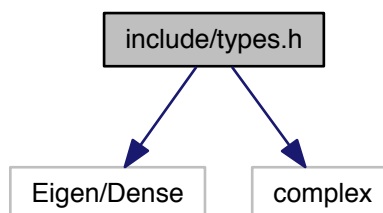
## Variables

- `std::random_device` [qpp::stat::\\_rd](#)
- `std::mt19937` [qpp::stat::\\_rng](#)

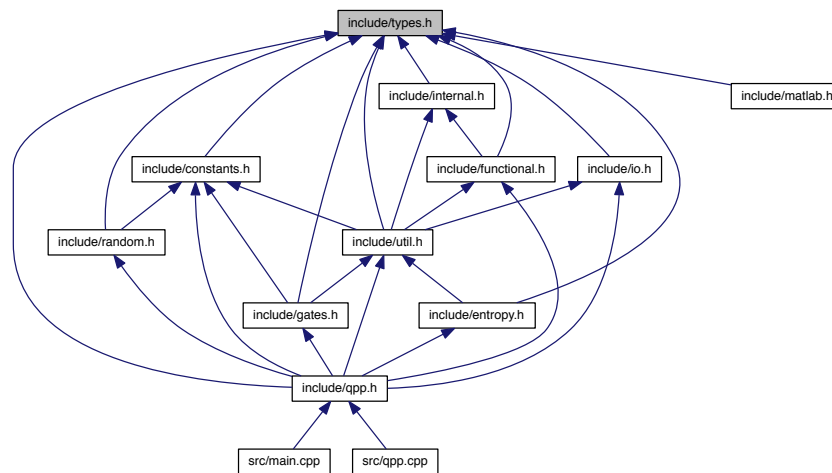
## 6.11 include/types.h File Reference

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

Include dependency graph for types.h:



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



## Namespaces

- [qp](#)
- [qp::types](#)

## Typedefs

- `typedef std::complex< double > qp::types::cplx`
- `typedef Eigen::MatrixXcd qp::types::cmat`
- `typedef Eigen::MatrixXd qp::types::dmat`
- `typedef Eigen::MatrixXf qp::types::fmat`
- `typedef Eigen::MatrixXi qp::types::imat`
- `template<typename Derived >`  
`using qp::types::EigenExpression = Eigen::MatrixBase< Derived >`
- `template<typename MatrixType >`  
`using qp::types::TemplatedEigenMatrix = Eigen::Matrix< typename MatrixType::Scalar, Eigen::Dynamic,`  
`Eigen::Dynamic >`

## 6.12 include/util.h File Reference

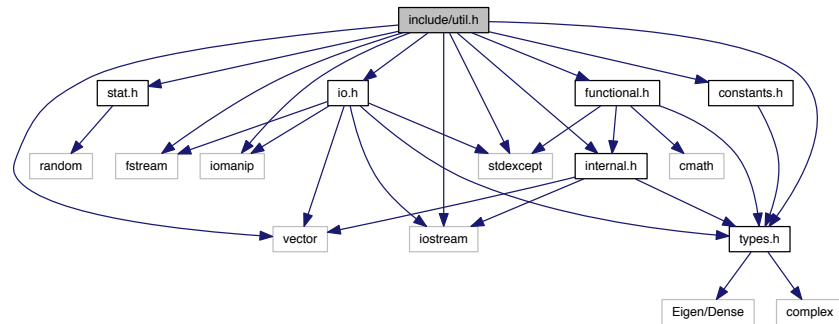
```

#include <stdexcept>
#include <vector>
#include <iostream>
#include <fstream>
#include <iomanip>
#include "types.h"
#include "constants.h"
#include "internal.h"
#include "stat.h"
#include "io.h"
#include "functional.h"

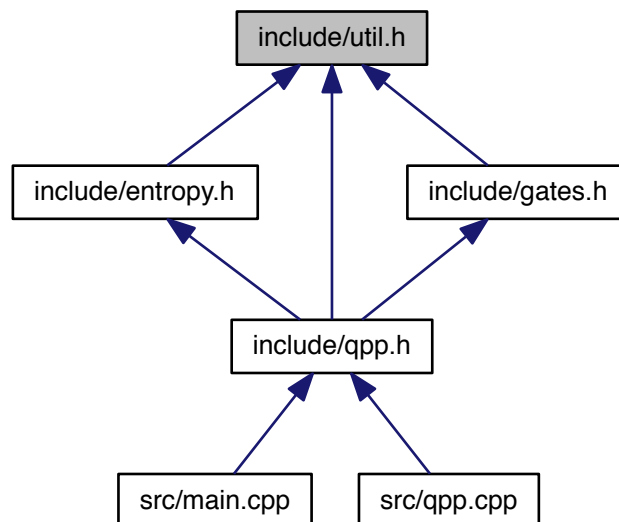
```



Include dependency graph for util.h:



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



## Namespaces

- [qpp](#)

## Functions

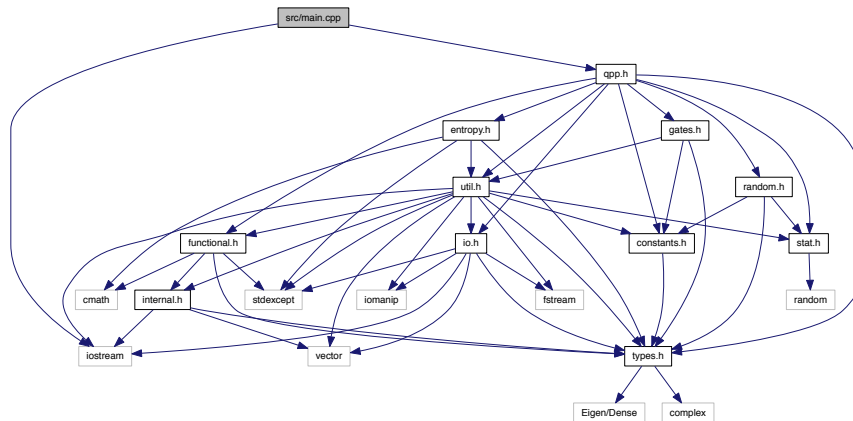
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > qpp::transpose (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > qpp::conjugate (const types::EigenExpression< MatrixType > &A)`

- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > qpp::adjoint (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`MatrixType::Scalar qpp::trace (const types::EigenExpression< MatrixType > &A)`
- `template<typename FunctionInputType , typename FunctionOutputType , typename MatrixInputType >`  
`Eigen::Matrix`  
`< FunctionOutputType,`  
`Eigen::Dynamic, Eigen::Dynamic > qpp::fun (const types::EigenExpression< MatrixInputType > &A,`  
`FunctionOutputType(*f)(const FunctionInputType &))`
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > qpp::abs (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`double qpp::norm (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::cmat qpp::evals (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::cmat qpp::evecs (const types::EigenExpression< MatrixType > &A)`
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > qpp::kron (const types::EigenExpression< MatrixType > &A, const types::Eigen-`  
`Expression< MatrixType > &B)`
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > qpp::kron\_list (const std::vector< MatrixType > &list)`
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > qpp::kron\_pow (const types::TemplatedEigenMatrix< MatrixType > &A, size_t n)`
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > qpp::reshape (const types::EigenExpression< MatrixType > &A, size_t rows, size_t cols)`
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > qpp::syspermute (const types::EigenExpression< MatrixType > &A, const std::vector< size-`  
`_t > perm, const std::vector< size_t > &dims)`
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > qpp::ptrace2 (const types::EigenExpression< MatrixType > &A, const std::vector< size_t >`  
`dims)`
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > qpp::ptrace (const types::EigenExpression< MatrixType > &A, const std::vector< size_t >`  
`&subsys, const std::vector< size_t > &dims)`
- `template<typename MatrixType >`  
`types::TemplatedEigenMatrix`  
`< MatrixType > qpp::ptranspose (const types::EigenExpression< MatrixType > &A, const std::vector< size-`  
`_t > &subsys, const std::vector< size_t > &dims)`

## 6.13 src/main.cpp File Reference

```
#include <iostream>
#include "qpp.h"
```

Include dependency graph for main.cpp:



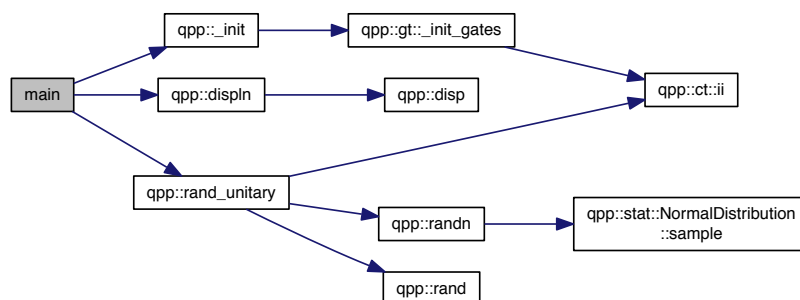
## Functions

- int [main](#) ()

### 6.13.1 Function Documentation

#### 6.13.1.1 int main ( )

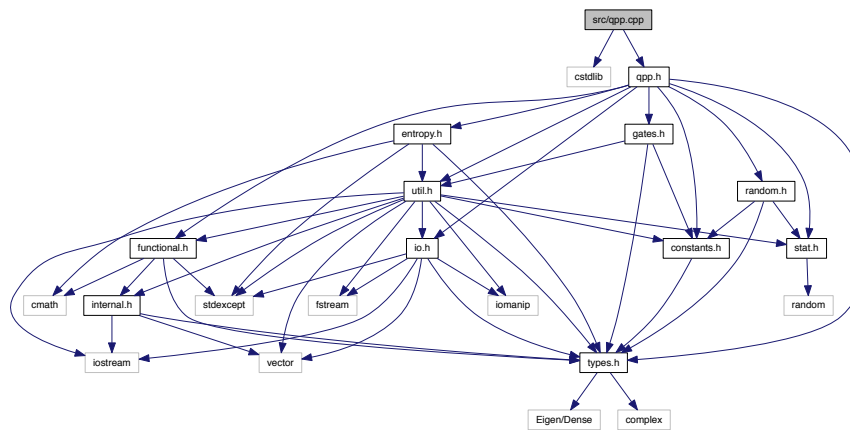
Here is the call graph for this function:



## 6.14 src/qpp.cpp File Reference

```
#include <cstdlib>
#include "qpp.h"
```

Include dependency graph for qpp.cpp:



## Namespaces

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

## Functions

- `types::cmat` [qpp::gt::TOF](#) (8, 8)
- `int` [qpp::\\_init](#) ()