

# C++ for Estimating Integer Least Square Problems

Generated by Doxygen 1.8.17



<b>1 Namespace Index</b>	<b>1</b>
1.1 Namespace List	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 File Index</b>	<b>5</b>
3.1 File List	5
<b>4 Namespace Documentation</b>	<b>7</b>
4.1 cils Namespace Reference	7
4.1.1 Detailed Description	7
4.1.2 Function Documentation	7
4.1.2.1 matrix_vector_mult() [1/2]	8
4.1.2.2 matrix_vector_mult() [2/2]	8
4.2 cils::program_def Namespace Reference	8
4.2.1 Typedef Documentation	9
4.2.1.1 index	9
4.2.1.2 scalar	9
4.2.2 Function Documentation	9
4.2.2.1 d_s()	9
4.2.2.2 indicator()	10
4.2.2.3 init_guess()	10
4.2.2.4 init_program_def()	10
4.2.2.5 permutation()	10
4.2.3 Variable Documentation	10
4.2.3.1 block_size	10
4.2.3.2 chunk	11
4.2.3.3 chunk_size	11
4.2.3.4 coeff	11
4.2.3.5 is_constrained	11
4.2.3.6 is_local	11
4.2.3.7 is_matlab	11
4.2.3.8 is_nc	12
4.2.3.9 is_qr	12
4.2.3.10 max_iter	12
4.2.3.11 max_proc	12
4.2.3.12 max_search	12
4.2.3.13 max_thre	12
4.2.3.14 min_proc	13
4.2.3.15 mode	13
4.2.3.16 num_trials	13
4.2.3.17 offset	13

4.2.3.18 plot_itr	13
4.2.3.19 q	13
4.2.3.20 qam	14
4.2.3.21 schedule	14
4.2.3.22 search_iter	14
4.2.3.23 SNR	14
4.2.3.24 spilt_size	14
4.2.3.25 stop	14
4.2.3.26 verbose	15
4.3 helper Namespace Reference	15
4.3.1 Function Documentation	16
4.3.1.1 b_mtimes()	16
4.3.1.2 b_rand()	16
4.3.1.3 display_matrix()	16
4.3.1.4 display_vector()	17
4.3.1.5 eye()	17
4.3.1.6 find_bit_error_rate()	17
4.3.1.7 find_residual()	18
4.3.1.8 if_all_x_true()	19
4.3.1.9 inv()	19
4.3.1.10 length_nonzeros()	19
4.3.1.11 mtimes() [1/2]	20
4.3.1.12 mtimes() [2/2]	20
4.3.1.13 mtimes_AP()	20
4.3.1.14 mtimes_Axy()	21
4.3.1.15 mtimes_col()	21
4.3.1.16 mtimes_v()	21
4.3.1.17 norm()	22
4.3.1.18 planerot()	22
4.3.1.19 randperm()	23
4.3.1.20 unique_vector()	23
4.4 plot_helper Namespace Reference	23
4.4.1 Function Documentation	24
4.4.1.1 plot_first_block()	24
4.4.1.2 plot_runtime()	24
4.4.1.3 plot_runtime_III()	25
4.4.1.4 plot_runtime_ud()	25
4.4.2 Variable Documentation	26
4.4.2.1 a	26
4.4.2.2 ber	26
4.4.2.3 is_constrained	26
4.4.2.4 k	26

4.4.2.5 l_max	26
4.4.2.6 m	27
4.4.2.7 max_iter	27
4.4.2.8 max_proc	27
4.4.2.9 min_proc	27
4.4.2.10 n	27
4.4.2.11 proc_num	27
4.4.2.12 res	28
4.4.2.13 SNR	28
4.4.2.14 spu	28
4.4.2.15 tim	28
4.4.2.16 title1	28
4.4.2.17 title3	28
4.5 py_qr Namespace Reference	29
4.5.1 Function Documentation	29
4.5.1.1 check_mem()	29
4.5.2 Variable Documentation	29
4.5.2.1 block_mem	29
4.5.2.2 max_mem	29
4.5.2.3 total	30
4.5.2.4 used	30
4.5.2.5 x	30
<b>5 Class Documentation</b>	<b>31</b>
5.1 cils::cils< scalar, index, m, n > Class Template Reference	31
5.1.1 Detailed Description	32
5.1.2 Constructor & Destructor Documentation	32
5.1.2.1 cils()	32
5.1.2.2 ~cils()	33
5.1.3 Member Function Documentation	33
5.1.3.1 cils_babai_search_omp()	33
5.1.3.2 cils_babai_search_serial()	33
5.1.3.3 cils_back_solve()	34
5.1.3.4 cils_back_solve_omp()	34
5.1.3.5 cils_block_search_omp()	35
5.1.3.6 cils_block_search_serial()	35
5.1.3.7 cils_block_search_serial_CPUTEST()	36
5.1.3.8 cils_grad_proj()	36
5.1.3.9 cils_partition_deficient()	37
5.1.3.10 cils_qrp_serial()	37
5.1.3.11 cils_scp_block_babai_omp()	37
5.1.3.12 cils_scp_block_babai_serial()	38

5.1.3.13 cils_scp_block_optimal_mpi()	38
5.1.3.14 cils_scp_block_optimal_omp()	38
5.1.3.15 cils_scp_block_optimal_serial()	38
5.1.3.16 cils_scp_block_suboptimal_omp()	39
5.1.3.17 cils_scp_block_suboptimal_serial()	39
5.1.3.18 cils_sic_serial()	39
5.1.3.19 cils_sic_subopt()	40
5.1.3.20 init()	40
5.1.3.21 init_R()	40
5.1.3.22 init_ud()	40
5.1.3.23 init_y()	40
5.1.4 Member Data Documentation	41
5.1.4.1 A	41
5.1.4.2 H	41
5.1.4.3 init_res	41
5.1.4.4 I	41
5.1.4.5 lower	41
5.1.4.6 matlabPtr	42
5.1.4.7 P	42
5.1.4.8 qam	42
5.1.4.9 R_A	42
5.1.4.10 sigma	42
5.1.4.11 snr	42
5.1.4.12 tolerance	43
5.1.4.13 u	43
5.1.4.14 upper	43
5.1.4.15 v_a	43
5.1.4.16 v_q	43
5.1.4.17 x_r	43
5.1.4.18 x_t	44
5.1.4.19 y_a	44
5.1.4.20 Z	44
5.2 cils::cils_reduction< scalar, index > Class Template Reference	44
5.2.1 Detailed Description	45
5.2.2 Constructor & Destructor Documentation	45
5.2.2.1 cils_reduction()	45
5.2.3 Member Function Documentation	45
5.2.3.1 cils_eml_qr()	45
5.2.3.2 cils_LLL_omp()	46
5.2.3.3 cils_LLL_qr_omp()	46
5.2.3.4 cils_LLL_qr_reduction()	46
5.2.3.5 cils_LLL_qr_serial()	46

5.2.3.6 cils_LLL_reduction()	47
5.2.3.7 cils_LLL_serial()	47
5.2.3.8 cils_obils_matlab()	47
5.2.3.9 cils_obils_reduction()	47
5.2.3.10 cils_obils_reduction_matlab()	48
5.2.3.11 cils_qr_omp()	48
5.2.3.12 cils_qr_py()	48
5.2.3.13 cils_qr_py_helper()	48
5.2.3.14 cils_qr_serial()	48
5.2.3.15 cils_qr_serial_col()	49
5.2.4 Member Data Documentation	49
5.2.4.1 A	49
5.2.4.2 G	49
5.2.4.3 P	50
5.2.4.4 p	50
5.2.4.5 Q	50
5.2.4.6 R_Q	50
5.2.4.7 R_R	50
5.2.4.8 y_a	50
5.2.4.9 y_q	51
5.2.4.10 y_r	51
5.2.4.11 Z	51
5.3 cils::cils_search< scalar, index > Class Template Reference	51
5.3.1 Detailed Description	52
5.3.2 Constructor & Destructor Documentation	52
5.3.2.1 cils_search()	52
5.3.3 Member Function Documentation	52
5.3.3.1 bound()	52
5.3.3.2 ils_search()	52
5.3.3.3 ils_search_omp()	53
5.3.3.4 obils_search()	53
5.3.3.5 obils_search2()	53
5.3.3.6 obils_search_matlab()	53
5.3.3.7 obils_search_omp()	54
5.3.3.8 ubils_search()	54
5.4 cils::returnType< scalar, index > Struct Template Reference	54
5.4.1 Detailed Description	54
5.4.2 Member Data Documentation	55
5.4.2.1 info	55
5.4.2.2 run_time	55
5.4.2.3 x	55

<b>6 File Documentation</b>	<b>57</b>
6.1 src/example/cils_standard_test.cpp File Reference	57
6.2 src/example/cils_underdetermined_test.cpp File Reference	57
6.2.1 Function Documentation	57
6.2.1.1 block_optimal_test()	58
6.2.1.2 plot_run()	58
6.3 src/example/plot_helper.py File Reference	58
6.4 src/example/py_qr.py File Reference	59
6.5 src/include/cils.h File Reference	59
6.5.1 Detailed Description	60
6.6 src/include/config.h File Reference	60
6.7 src/include/helper.h File Reference	61
6.7.1 Detailed Description	62
6.7.2 Variable Documentation	63
6.7.2.1 ZERO	63
6.8 src/source/cils.cpp File Reference	63
6.9 src/source/cils_babai_search.cpp File Reference	63
6.9.1 Detailed Description	64
6.10 src/source/cils_block_search.cpp File Reference	64
6.10.1 Detailed Description	64
6.11 src/source/cils_block_search_GPU.cpp File Reference	65
6.11.1 Macro Definition Documentation	65
6.11.1.1 NUM_GLOBAL_WITEMS	65
6.11.2 Function Documentation	65
6.11.2.1 compareResults()	65
6.11.2.2 run()	66
6.11.2.3 timeAddVectorsCPU()	66
6.11.2.4 warmup()	66
6.12 src/source/cils_ils_search.cpp File Reference	66
6.12.1 Detailed Description	67
6.13 src/source/cils_init_point.cpp File Reference	67
6.13.1 Detailed Description	67
6.14 src/source/cils_partition_deficient.cpp File Reference	67
6.15 src/source/cils_reduction.cpp File Reference	68
6.16 src/source/cils_sic_opt.cpp File Reference	68
6.16.1 Detailed Description	68
<b>Index</b>	<b>69</b>



# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">cils</a>	7
<a href="#">cils::program_def</a>	8
<a href="#">helper</a>	15
<a href="#">plot_helper</a>	23
<a href="#">py_qr</a>	29



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">cils::cils&lt; scalar, index, m, n &gt;</a>	<a href="#">31</a>
<a href="#">cils::cils_reduction&lt; scalar, index &gt;</a>	<a href="#">44</a>
<a href="#">cils::cils_search&lt; scalar, index &gt;</a>	<a href="#">51</a>
<a href="#">cils::returnType&lt; scalar, index &gt;</a>	<a href="#">54</a>



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

src/example/cils_standard_test.cpp . . . . .	57
src/example/cils_underdetermined_test.cpp . . . . .	57
src/example/plot_helper.py . . . . .	58
src/example/py_qr.py . . . . .	59
src/include/cils.h	
Computation of integer least square problem . . . . .	59
src/include/config.h . . . . .	60
src/include/helper.h	
Computation of integer least square problem . . . . .	61
src/source/cils.cpp . . . . .	63
src/source/cils_babai_search.cpp	
Computation of integer least square problem by constrained non-block Babai Estimator . . . . .	63
src/source/cils_block_search.cpp	
Computation of Block Babai Algorithm . . . . .	64
src/source/cils_block_search_GPU.cpp . . . . .	65
src/source/cils_ils_search.cpp	
Computation of SS_search Algorithm . . . . .	66
src/source/cils_init_point.cpp	
Computation of indexeger least square problem by constrained non-block Babai Estimator . . . . .	67
src/source/cils_partition_deficient.cpp . . . . .	67
src/source/cils_reduction.cpp . . . . .	68
src/source/cils_sic_opt.cpp	
Computation of indexeger least square problem by constrained non-block Babai Estimator . . . . .	68



## Chapter 4

# Namespace Documentation

### 4.1 cils Namespace Reference

#### Namespaces

- [program\\_def](#)

#### Classes

- class [cils](#)
- class [cils\\_reduction](#)
- class [cils\\_search](#)
- struct [returnType](#)

#### Functions

- `template<typename scalar , typename index , index m, index n>`  
`void matrix\_vector\_mult (const array< scalar, m *n > &Z, vector< scalar > *x)`
- `template<typename scalar , typename index , index m, index n>`  
`void matrix\_vector\_mult (const array< scalar, m *n > &Z, const vector< scalar > &x, vector< scalar > &c)`

#### 4.1.1 Detailed Description

namespace of cils

#### 4.1.2 Function Documentation

#### 4.1.2.1 `matrix_vector_mult()` [1/2]

```
template<typename scalar , typename index , index m, index n>
void cils::matrix_vector_mult (
    const array< scalar, m *n > & Z,
    const vector< scalar > & x,
    vector< scalar > & c ) [inline]
```

Definition at line 250 of file cils.cpp.

#### 4.1.2.2 `matrix_vector_mult()` [2/2]

```
template<typename scalar , typename index , index m, index n>
void cils::matrix_vector_mult (
    const array< scalar, m *n > & Z,
    vector< scalar > * x ) [inline]
```

Definition at line 234 of file cils.cpp.

## 4.2 `cils::program_def` Namespace Reference

### Typedefs

- typedef int [index](#)
- typedef double [scalar](#)

### Functions

- `std::vector< index > d_s` (`N/block_size+spilt_size - 1, block_size`)
- `std::vector< index > indicator` (`2 *q, 0`)
- `vector< vector< scalar > > permutation` (`search_iter+3`)
- void `init_program_def` (`int argc, char *argv[]`)
- template<typename scalar , typename index , index m, index n>  
void `init_guess` (`index init_value, vector< scalar > *z_B, scalar *x_R`)

### Variables

- `index qam` = 1
- `index SNR` = 15
- `index max_iter` = 100
- `index search_iter` = 1e5
- `index stop` = 3
- `index schedule` = 2
- `index chunk_size` = 1
- `index block_size` = 2
- `index spilt_size` = 2
- `index offset` = 2
- `index is_constrained` = true



- `index is_nc` = false
- `index is_matlab` = false
- `index is_qr` = false
- `index mode` = 1
- `index num_trials` = 10
- `index is_local` = 1
- `index max_search` = 400000
- `index min_proc` = 2
- `index plot_itr` = 1
- `scalar coeff` = 17.5
- `index max_proc` = 10
- `index max_thre` = 400000
- `auto q` = `static_cast<index>(std::ceil((scalar) N / (scalar) M))`
- `index verbose` = false
- `index chunk` = 1

## 4.2.1 Typedef Documentation

### 4.2.1.1 `index`

```
typedef int cils::program_def::index
```

Definition at line 17 of file config.h.

### 4.2.1.2 `scalar`

```
typedef double cils::program_def::scalar
```

Definition at line 18 of file config.h.

## 4.2.2 Function Documentation

### 4.2.2.1 `d_s()`

```
std::vector<index> cils::program_def::d_s (
    N/block_size+spilt_size - 1,
    block_size )
```

#### 4.2.2.2 indicator()

```
std::vector<index> cils::program_def::indicator (
    2 * q,
    0 )
```

#### 4.2.2.3 init\_guess()

```
template<typename scalar , typename index , index m, index n>
void cils::program_def::init_guess (
    index init_value,
    vector< scalar > * z_B,
    scalar * x_R )
```

Definition at line 152 of file config.h.

#### 4.2.2.4 init\_program\_def()

```
void cils::program_def::init_program_def (
    int argc,
    char * argv[] )
```

Definition at line 82 of file config.h.

#### 4.2.2.5 permutation()

```
vector<vector<scalar> > cils::program_def::permutation (
    search_iter+ 3 )
```

### 4.2.3 Variable Documentation

#### 4.2.3.1 block\_size

```
index cils::program_def::block_size = 2
```

Definition at line 32 of file config.h.

#### 4.2.3.2 chunk

`index` `cils::program_def::chunk = 1`

Definition at line 50 of file config.h.

#### 4.2.3.3 chunk\_size

`index` `cils::program_def::chunk_size = 1`

Definition at line 31 of file config.h.

#### 4.2.3.4 coeff

`scalar` `cils::program_def::coeff = 17.5`

Definition at line 45 of file config.h.

#### 4.2.3.5 is\_constrained

`index` `cils::program_def::is_constrained = true`

Definition at line 35 of file config.h.

#### 4.2.3.6 is\_local

`index` `cils::program_def::is_local = 1`

Definition at line 41 of file config.h.

#### 4.2.3.7 is\_matlab

`index` `cils::program_def::is_matlab = false`

Definition at line 37 of file config.h.

#### 4.2.3.8 is\_nc

`index` cils::program\_def::is\_nc = false

Definition at line 36 of file config.h.

#### 4.2.3.9 is\_qr

`index` cils::program\_def::is\_qr = false

Definition at line 38 of file config.h.

#### 4.2.3.10 max\_iter

`index` cils::program\_def::max\_iter = 100

Definition at line 27 of file config.h.

#### 4.2.3.11 max\_proc

`index` cils::program\_def::max\_proc = 10

Definition at line 46 of file config.h.

#### 4.2.3.12 max\_search

`index` cils::program\_def::max\_search = 400000

Definition at line 42 of file config.h.

#### 4.2.3.13 max\_thre

`index` cils::program\_def::max\_thre = 400000

Definition at line 47 of file config.h.

#### 4.2.3.14 min\_proc

```
index cils::program_def::min_proc = 2
```

Definition at line 43 of file config.h.

#### 4.2.3.15 mode

```
index cils::program_def::mode = 1
```

Definition at line 39 of file config.h.

#### 4.2.3.16 num\_trials

```
index cils::program_def::num_trials = 10
```

Definition at line 40 of file config.h.

#### 4.2.3.17 offset

```
index cils::program_def::offset = 2
```

Definition at line 34 of file config.h.

#### 4.2.3.18 plot\_itr

```
index cils::program_def::plot_itr = 1
```

Definition at line 44 of file config.h.

#### 4.2.3.19 q

```
auto cils::program_def::q = static_cast<index>(std::ceil((scalar) N / (scalar) M))
```

Definition at line 48 of file config.h.

#### 4.2.3.20 qam

`index` `cils::program_def::qam = 1`

`omp_sched_static = 0x1, omp_sched_dynamic = 0x2, omp_sched_guided = 0x3, omp_sched_auto = 0x4,`

Definition at line 25 of file `config.h`.

#### 4.2.3.21 schedule

`index` `cils::program_def::schedule = 2`

Definition at line 30 of file `config.h`.

#### 4.2.3.22 search\_iter

`index` `cils::program_def::search_iter = 1e5`

Definition at line 28 of file `config.h`.

#### 4.2.3.23 SNR

`index` `cils::program_def::SNR = 15`

Definition at line 26 of file `config.h`.

#### 4.2.3.24 spilt\_size

`index` `cils::program_def::spilt_size = 2`

Definition at line 33 of file `config.h`.

#### 4.2.3.25 stop

`index` `cils::program_def::stop = 3`

Definition at line 29 of file `config.h`.

## 4.2.3.26 verbose

```
index cils::program_def::verbose = false
```

Definition at line 49 of file config.h.

## 4.3 helper Namespace Reference

### Functions

- void [b\\_rand](#) (const int n, double \*r)
- void [randperm](#) (int n, double \*p)
- template<typename scalar , typename index >  
void [eye](#) (index n, scalar \*A)
- template<typename scalar , typename index , index m, index n, index mb>  
void [b\\_mtimes](#) (const array< scalar, m \*n > &A\_C, const array< scalar, n \*mb > &B, array< scalar, m \*mb > &C)
- template<typename scalar , typename index , index m, index n>  
void [mtimes](#) (const scalar A\_C[4], const array< scalar, m \*n > &B, array< scalar, 2 \*n > &C)
- template<typename scalar , typename index , index m, index n>  
void [mtimes](#) (const array< scalar, m \*m > &Q, const array< scalar, m \*n > &R, array< scalar, m \*n > &A\_t)
- template<typename scalar , typename index >  
void [mtimes\\_v](#) (index m, index n, const vector< scalar > &Q, const vector< scalar > &R, vector< scalar > &A\_t)
- template<typename scalar , typename index >  
void [mtimes\\_col](#) (index m, index n, const vector< scalar > &Q, const vector< scalar > &R, vector< scalar > &A\_t)
- template<typename scalar , typename index >  
void [mtimes\\_AP](#) (index m, index n, const scalar \*A, const scalar \*P, scalar \*A\_t)
- template<typename scalar , typename index >  
void [inv](#) (const index K, const index N, const vector< scalar > &x, vector< scalar > &y)
- template<typename scalar , typename index >  
void [mtimes\\_Axy](#) (const index m, const index n, const scalar \*A, const scalar \*x, scalar \*y)
- template<typename scalar , typename index , index n>  
index [length\\_nonzeros](#) (const scalar \*x, const scalar \*y)
- template<typename scalar , typename index >  
void [planerot](#) (scalar x[2], scalar G[4])
- template<typename scalar , typename index >  
scalar [norm](#) (const index n, const scalar \*v)
- template<typename scalar , typename index >  
scalar [find\\_bit\\_error\\_rate](#) (const index n, const scalar \*x\_b, const scalar \*x\_t, const index k)
- template<typename scalar , typename index >  
void [display\\_matrix](#) (index m, index n, const scalar \*x, const string &name)
- template<typename scalar , typename index >  
void [display\\_vector](#) (const index n, const scalar \*x, const string &name)
- template<typename index >  
bool [if\\_all\\_x\\_true](#) (const vector< bool > &x)
- template<typename scalar , typename index >  
void [unique\\_vector](#) (const vector< scalar > &a, vector< scalar > &b)
- template<typename scalar , typename index >  
scalar [find\\_residual](#) (const index m, const index n, const scalar \*A, const scalar \*x, const scalar \*y)

### 4.3.1 Function Documentation

#### 4.3.1.1 `b_mtimes()`

```
template<typename scalar , typename index , index m, index n, index mb>
void helper::b_mtimes (
    const array< scalar, m *n > & A_C,
    const array< scalar, n *mb > & B,
    array< scalar, m *mb > & C )
```

Definition at line 226 of file helper.h.

#### 4.3.1.2 `b_rand()`

```
void helper::b_rand (
    const int n,
    double * r )
```

Definition at line 31 of file helper.h.

#### 4.3.1.3 `display_matrix()`

```
template<typename scalar , typename index >
void helper::display_matrix (
    index m,
    index n,
    const scalar * x,
    const string & name )
```

Simple function for displaying a m-by-n matrix with name

##### Template Parameters

<i>scalar</i>	: real number type
<i>index</i>	: integer type

##### Parameters

<i>m</i>	: integer scalar, size of the matrix
<i>n</i>	: integer scalar, size of the matrix
<i>x</i>	: matrix, in pointer
<i>name</i>	display name of the matrix



Definition at line 634 of file helper.h.

#### 4.3.1.4 display\_vector()

```
template<typename scalar , typename index >
void helper::display_vector (
    const index n,
    const scalar * x,
    const string & name )
```

Simple function for displaying the a vector with name

##### Template Parameters

<i>scalar</i>	: real number type
<i>index</i>	: integer type

##### Parameters

<i>n</i>	: integer scalar, size of the vector
<i>x</i>	: vector, in pointer
<i>name</i>	display name of the vector

Definition at line 654 of file helper.h.

#### 4.3.1.5 eye()

```
template<typename scalar , typename index >
void helper::eye (
    index n,
    scalar * A )
```

Definition at line 216 of file helper.h.

#### 4.3.1.6 find\_bit\_error\_rate()

```
template<typename scalar , typename index >
scalar helper::find_bit_error_rate (
    const index n,
    const scalar * x_b,
    const scalar * x_t,
    const index k )
```

Find BER with given two vectors

## Template Parameters

<i>scalar</i>	: real number type
<i>index</i>	: integer type

## Parameters

<i>n</i>	: integer scalar, size of the vector
$x_{\leftrightarrow}$ <i>_b</i>	: input vector 1
$x_{\leftrightarrow}$ <i>_t</i>	: input vector 2
<i>k</i>	: log <sub>4</sub> (qam)

## Returns

Definition at line 597 of file helper.h.

## 4.3.1.7 find\_residual()

```
template<typename scalar , typename index >
scalar helper::find_residual (
    const index m,
    const index n,
    const scalar * A,
    const scalar * x,
    const scalar * y ) [inline]
```

Return the result of  $\|y-A*x\|$ .

## Template Parameters

<i>scalar</i>	: real number type
<i>index</i>	: integer type

## Parameters

<i>m</i>	: integer scalar, size of the matrix
<i>n</i>	: integer scalar, size of the matrix
<i>A</i>	: matrix, m-by-n in pointer
<i>x</i>	: vector, n-by-1 in pointer
<i>y</i>	: vector, m-by-1 in pointer, storing result.

**Returns**

residual : l2 norm

Definition at line 842 of file helper.h.

**4.3.1.8 if\_all\_x\_true()**

```
template<typename index >
bool helper::if_all_x_true (
    const vector< bool > & x )
```

Determine whether all values of x are true by lambda expression.

**Template Parameters**

<i>index</i>	: integer type : integer required
--------------	-----------------------------------

**Parameters**

<i>x</i>	: Testing vector
----------	------------------

**Returns**

true/false

Definition at line 671 of file helper.h.

**4.3.1.9 inv()**

```
template<typename scalar , typename index >
void helper::inv (
    const index K,
    const index N,
    const vector< scalar > & x,
    vector< scalar > & y )
```

Definition at line 297 of file helper.h.

**4.3.1.10 length\_nonzeros()**

```
template<typename scalar , typename index , index n>
index helper::length_nonzeros (
    const scalar * x,
    const scalar * y )
```

**Template Parameters**

<i>scalar</i>	
<i>index</i>	
<i>n</i>	

**Parameters**

<i>x</i>	
<i>y</i>	

**Returns**

Definition at line 469 of file helper.h.

**4.3.1.11 mtimes() [1/2]**

```
template<typename scalar , typename index , index m, index n>
void helper::mtimes (
    const array< scalar, m *m > & Q,
    const array< scalar, m *n > & R,
    array< scalar, m *n > & A_t )
```

Definition at line 253 of file helper.h.

**4.3.1.12 mtimes() [2/2]**

```
template<typename scalar , typename index , index m, index n>
void helper::mtimes (
    const scalar A_C[4],
    const array< scalar, m *n > & B,
    array< scalar, 2 *n > & C )
```

Definition at line 243 of file helper.h.

**4.3.1.13 mtimes\_AP()**

```
template<typename scalar , typename index >
void helper::mtimes_AP (
    index m,
    index n,
    const scalar * A,
    const scalar * P,
    scalar * A_t )
```

Definition at line 286 of file helper.h.

**4.3.1.14 mtimes\_Axy()**

```
template<typename scalar , typename index >
void helper::mtimes_Axy (
    const index m,
    const index n,
    const scalar * A,
    const scalar * x,
    scalar * y )
```

Matrix-vector multiplication:  $Ax=y$ , where  $A$  is m-by-n matrix

**Template Parameters**

<i>scalar</i>	: real number type
<i>index</i>	: integer type

**Parameters**

<i>m</i>	: integer scalar, size of the matrix
<i>n</i>	: integer scalar, size of the matrix
<i>A</i>	: matrix, m-by-n in pointer
<i>x</i>	: vector, n-by-1 in pointer
<i>y</i>	: vector, m-by-1 in pointer, storing result.

Definition at line 447 of file helper.h.

**4.3.1.15 mtimes\_col()**

```
template<typename scalar , typename index >
void helper::mtimes_col (
    index m,
    index n,
    const vector< scalar > & Q,
    const vector< scalar > & R,
    vector< scalar > & A_t )
```

Definition at line 275 of file helper.h.

**4.3.1.16 mtimes\_v()**

```
template<typename scalar , typename index >
void helper::mtimes_v (
    index m,
    index n,
    const vector< scalar > & Q,
    const vector< scalar > & R,
    vector< scalar > & A_t )
```

Definition at line 264 of file helper.h.

**4.3.1.17 norm()**

```
template<typename scalar , typename index >
scalar helper::norm (
    const index n,
    const scalar * v )
```

The Euclidean norm of vector v. This norm is also called the 2-norm, vector magnitude, or Euclidean length.

**Template Parameters**

<i>scalar</i>	: real number type
<i>index</i>	: integer type

**Parameters**

<i>n</i>	: the size of the vector
<i>v</i>	: input vector

**Returns**

Definition at line 560 of file helper.h.

**4.3.1.18 planerot()**

```
template<typename scalar , typename index >
void helper::planerot (
    scalar x[2],
    scalar G[4] )
```

Givens plane rotation

**Template Parameters**

<i>scalar</i>	: real number type
<i>index</i>	: integer type

**Parameters**

<i>x</i>	: A 2-component column vector.
<i>G</i>	: A 2-by-2 orthogonal matrix G so that $y = G*x$ has $y(2) = 0$ .

Definition at line 514 of file helper.h.

#### 4.3.1.19 randperm()

```
void helper::randperm (
    int n,
    double * p )
```

Definition at line 126 of file helper.h.

#### 4.3.1.20 unique\_vector()

```
template<typename scalar , typename index >
void helper::unique_vector (
    const vector< scalar > & a,
    vector< scalar > & b )
```

Returns the same data as in a, but with no repetitions. b is in sorted order.

##### Template Parameters

<i>scalar</i>	: real number type
<i>index</i>	: integer type

##### Parameters

<i>a</i>	: input vector to be processed
<i>b</i>	: output vector to store the results

Definition at line 693 of file helper.h.

## 4.4 plot\_helper Namespace Reference

### Functions

- def [plot\\_runtime\\_ud](#) (n, SNR, k, l\_max, max\_iter, res, ber, tim, proc\_num, spu, max\_proc, min\_proc, is\_constrained, m)
- def [plot\\_runtime\\_III](#) (n, qr\_l, i, max\_proc, min\_proc, qrT, III, III\_qr, qr\_spu, III\_spu, III\_qr\_spu, qIII\_spu)
- def [plot\\_runtime](#) (n, SNR, k, l\_max, block\_size, max\_iter, is\_qr, res, ber, tim, itr, ser\_tim, d\_s, proc\_num, spu, time, qr\_l, max\_proc, min\_proc, qrT, III, III\_qr, qr\_spu, III\_spu, III\_qr\_spu, qIII\_spu, tpu, is\_constrained, m)
- def [plot\\_first\\_block](#) (n, SNR, k, block\_size, ser\_tim, is\_qr, d\_s)

### Variables

- int [n](#) = 30
- int [SNR](#) = 35
- string [title3](#) = 'underdetermined'
- int [max\\_iter](#) = 1

- string `title1` = 'Box-constrained'
- `a` = `np.load(f'../cmake-build-release/{n}_report_plot_{SNR}_{title3}_{int(max_iter / 100)}_{title1}.npz')`
- `m` = `a['m']`
- `k` = `a['k']`
- `l_max` = `a['l_max']`
- `res` = `a['res']`
- `ber` = `a['ber']`
- `tim` = `a['tim']`
- `spu` = `a['spu']`
- `proc_num` = `a['proc_num']`
- `max_proc` = `a['max_proc']`
- `min_proc` = `a['min_proc']`
- `is_constrained` = `a['is_constrained']`

## 4.4.1 Function Documentation

### 4.4.1.1 `plot_first_block()`

```
def plot_helper.plot_first_block (
    n,
    SNR,
    k,
    block_size,
    ser_tim,
    is_qr,
    d_s )
```

Definition at line 411 of file `plot_helper.py`.

### 4.4.1.2 `plot_runtime()`

```
def plot_helper.plot_runtime (
    n,
    SNR,
    k,
    l_max,
    block_size,
    max_iter,
    is_qr,
    res,
    ber,
    tim,
    itr,
    ser_tim,
    d_s,
    proc_num,
    spu,
    time,
```



```
    qr_l,  
    max_proc,  
    min_proc,  
    qrT,  
    lll,  
    lll_qr,  
    qr_spu,  
    lll_spu,  
    lll_qr_spu,  
    qlll_spu,  
    tpu,  
    is_constrained,  
    m )
```

Definition at line 179 of file plot\_helper.py.

#### 4.4.1.3 plot\_runtime\_lll()

```
def plot_helper.plot_runtime_lll (  
    n,  
    qr_l,  
    i,  
    max_proc,  
    min_proc,  
    qrT,  
    lll,  
    lll_qr,  
    qr_spu,  
    lll_spu,  
    lll_qr_spu,  
    qlll_spu )
```

Definition at line 119 of file plot\_helper.py.

#### 4.4.1.4 plot\_runtime\_ud()

```
def plot_helper.plot_runtime_ud (  
    n,  
    SNR,  
    k,  
    l_max,  
    max_iter,  
    res,  
    ber,  
    tim,  
    proc_num,  
    spu,  
    max_proc,  
    min_proc,  
    is_constrained,  
    m )
```

Definition at line 9 of file plot\_helper.py.

## 4.4.2 Variable Documentation

### 4.4.2.1 a

```
plot_helper.a = np.load(f'../../cmake-build-release/{n}_report_plot_{SNR}_{title3}_{int(max_iter / 100)}_{title1}.npz')
```

Definition at line 468 of file plot\_helper.py.

### 4.4.2.2 ber

```
plot_helper.ber = a['ber']
```

Definition at line 476 of file plot\_helper.py.

### 4.4.2.3 is\_constrained

```
plot_helper.is_constrained = a['is_constrained']
```

Definition at line 482 of file plot\_helper.py.

### 4.4.2.4 k

```
plot_helper.k = a['k']
```

Definition at line 472 of file plot\_helper.py.

### 4.4.2.5 l\_max

```
plot_helper.l_max = a['l_max']
```

Definition at line 473 of file plot\_helper.py.

#### 4.4.2.6 m

```
plot_helper.m = a['m']
```

Definition at line 471 of file plot\_helper.py.

#### 4.4.2.7 max\_iter

```
plot_helper.max_iter = 1
```

Definition at line 466 of file plot\_helper.py.

#### 4.4.2.8 max\_proc

```
plot_helper.max_proc = a['max_proc']
```

Definition at line 480 of file plot\_helper.py.

#### 4.4.2.9 min\_proc

```
plot_helper.min_proc = a['min_proc']
```

Definition at line 481 of file plot\_helper.py.

#### 4.4.2.10 n

```
plot_helper.n = 30
```

Definition at line 463 of file plot\_helper.py.

#### 4.4.2.11 proc\_num

```
plot_helper.proc_num = a['proc_num']
```

Definition at line 479 of file plot\_helper.py.

#### 4.4.2.12 res

```
plot_helper.res = a['res']
```

Definition at line 475 of file plot\_helper.py.

#### 4.4.2.13 SNR

```
int plot_helper.SNR = 35
```

Definition at line 464 of file plot\_helper.py.

#### 4.4.2.14 spu

```
plot_helper.spu = a['spu']
```

Definition at line 478 of file plot\_helper.py.

#### 4.4.2.15 tim

```
plot_helper.tim = a['tim']
```

Definition at line 477 of file plot\_helper.py.

#### 4.4.2.16 title1

```
string plot_helper.title1 = 'Box-constrained'
```

Definition at line 467 of file plot\_helper.py.

#### 4.4.2.17 title3

```
string plot_helper.title3 = 'underdetermined'
```

Definition at line 465 of file plot\_helper.py.

## 4.5 py\_qr Namespace Reference

### Functions

- def `check_mem` ()

### Variables

- `total` = `int(total)`
- `used` = `int(used)`
- `max_mem` = `int(total * 0.8)`
- `block_mem` = `max_mem - used`
- `x` = `torch.rand((256,1024,block_mem)).cuda()`

### 4.5.1 Function Documentation

#### 4.5.1.1 `check_mem()`

```
def py_qr.check_mem ( )
```

Definition at line 246 of file `py_qr.py`.

### 4.5.2 Variable Documentation

#### 4.5.2.1 `block_mem`

```
py_qr.block_mem = max_mem - used
```

Definition at line 260 of file `py_qr.py`.

#### 4.5.2.2 `max_mem`

```
py_qr.max_mem = int(total * 0.8)
```

Definition at line 259 of file `py_qr.py`.

#### 4.5.2.3 total

```
py_qr.total = int(total)
```

Definition at line 254 of file py\_qr.py.

#### 4.5.2.4 used

```
py_qr.used = int(used)
```

Definition at line 254 of file py\_qr.py.

#### 4.5.2.5 x

```
py_qr.x = torch.rand((256,1024,block_mem)).cuda()
```

Definition at line 262 of file py\_qr.py.

## Chapter 5

# Class Documentation

### 5.1 cils::cils< scalar, index, m, n > Class Template Reference

```
#include <cils.h>
```

#### Public Member Functions

- [cils](#) (index [qam](#), index [snr](#))
- [~cils](#) ()
- void [init](#) (index rank)
- void [init\\_ud](#) ()
- void [init\\_y](#) ()
- void [init\\_R](#) ()
- [returnType](#)< scalar, index > [cils\\_back\\_solve](#) (array< scalar, n > &z\_B)
- [returnType](#)< scalar, index > [cils\\_babai\\_search\\_serial](#) (vector< scalar > \*z\_B)
- [returnType](#)< scalar, index > [cils\\_block\\_search\\_serial\\_CPUTEST](#) (const scalar \*R\_R, const scalar y\_r, const vector< index > \*d, vector< scalar > \*z\_B)
- [returnType](#)< scalar, index > [cils\\_babai\\_search\\_omp](#) (const index n\_proc, const index nswp, vector< scalar > \*z\_B)
- [returnType](#)< scalar, index > [cils\\_back\\_solve\\_omp](#) (const index n\_proc, const index nswp, vector< scalar > \*z\_B)
- [returnType](#)< scalar, index > [cils\\_block\\_search\\_omp](#) (const index n\_proc, const index nswp, const index [init](#), const scalar \*y\_r, const vector< index > \*d, vector< scalar > \*z\_B)
- [returnType](#)< scalar, index > [cils\\_sic\\_serial](#) (vector< scalar > &x)
- [returnType](#)< scalar, index > [cils\\_qrp\\_serial](#) (vector< scalar > &x)
- [returnType](#)< scalar, index > [cils\\_grad\\_proj](#) (vector< scalar > &x, const index max\_iter)
- [returnType](#)< scalar, index > [cils\\_sic\\_subopt](#) (vector< scalar > &z, array< scalar, m > &v\_cur, array< scalar, m \* n > A\_t, scalar v\_norm\_cur, scalar [tolerance](#), index method)
- [returnType](#)< scalar, index > [cils\\_scp\\_block\\_optimal\\_serial](#) (vector< scalar > &x\_cur, scalar v\_norm\_cur, index mode)
- [returnType](#)< scalar, index > [cils\\_scp\\_block\\_suboptimal\\_serial](#) (vector< scalar > &x\_cur, scalar v\_norm\_cur, index mode)
- [returnType](#)< scalar, index > [cils\\_scp\\_block\\_suboptimal\\_omp](#) (vector< scalar > &x\_cur, scalar v\_norm\_cur, index n\_proc, index mode)
- [returnType](#)< scalar, index > [cils\\_scp\\_block\\_babai\\_serial](#) (vector< scalar > &x\_cur, scalar v\_norm\_cur, index mode)
- [returnType](#)< scalar, index > [cils\\_scp\\_block\\_babai\\_omp](#) (vector< scalar > &x\_cur, scalar v\_norm\_cur, index n\_proc, index mode)

- `returnType`< scalar, index > `cils_partition_deficient` (scalar \*z\_B, scalar \*Q\_tilde, scalar \*R\_tilde, scalar \*H\_A, scalar \*Piv\_cum)
- `returnType`< scalar, index > `cils_block_search_serial` (const index `init`, const scalar \*R\_R, const scalar \*y\_r, const vector< index > \*d, vector< scalar > \*z\_B)
- `returnType`< scalar, index > `cils_scp_block_optimal_omp` (vector< scalar > &x\_cur, scalar v\_norm\_cur, index n\_proc, index mode)
- `returnType`< scalar, index > `cils_scp_block_optimal_mpi` (vector< scalar > &x\_cur, scalar \*v\_norm\_cur, index size, index rank)

## Public Attributes

- index `qam`
- index `snr`
- index `upper`
- index `lower`
- scalar `init_res`
- scalar `sigma`
- scalar `tolerance`
- array< scalar, m \*(n+1)/2 > `R_A`
- array< scalar, m \*n > `A`
- array< scalar, m \*n > `H`
- array< scalar, n \*n > `Z`
- array< scalar, n \*n > `P`
- array< scalar, n > `x_r`
- array< scalar, n > `x_t`
- array< scalar, n > `l`
- array< scalar, n > `u`
- array< scalar, m > `y_a`
- array< scalar, m > `v_a`
- array< scalar, m > `v_q`
- `std::unique_ptr`< `matlab::engine::MATLABEngine` > `matlabPtr`

### 5.1.1 Detailed Description

```
template<typename scalar, typename index, index m, index n>
class cils::cils< scalar, index, m, n >
```

Definition at line 64 of file `cils.h`.

### 5.1.2 Constructor & Destructor Documentation

#### 5.1.2.1 `cils()`

```
template<typename scalar , typename index , index m, index n>
cils::cils< scalar, index, m, n >::cils (
    index qam,
    index snr ) [inline]
```

Definition at line 79 of file `cils.h`.



### 5.1.2.2 `~cils()`

```
template<typename scalar , typename index , index m, index n>
cils::cils< scalar, index, m, n >::~~cils ( ) [inline]
```

Definition at line 107 of file `cils.h`.

## 5.1.3 Member Function Documentation

### 5.1.3.1 `cils_babai_search_omp()`

```
template<typename scalar , typename index , index m, index n>
returnType< scalar, index > cils::cils< scalar, index, m, n >::cils_babai_search_omp (
    const index n_proc,
    const index nswp,
    vector< scalar > * z_B )
```

Parallel Babai solver

Parameters

<i>n_proc</i>	number of Processors/Threads
<i>nswp</i>	maximum number of iterations
<i>z_B</i>	estimation of the true parameter

Returns

Definition at line 23 of file `cils_babai_search.cpp`.

### 5.1.3.2 `cils_babai_search_serial()`

```
template<typename scalar , typename index , index m, index n>
returnType< scalar, index > cils::cils< scalar, index, m, n >::cils_babai_search_serial (
    vector< scalar > * z_B )
```

Serial Babai solver

Parameters

<i>z_B</i>	
<i>is_constrained</i>	

## Returns

Definition at line 85 of file `cils_babai_search.cpp`.

### 5.1.3.3 `cils_back_solve()`

```
template<typename scalar , typename index , index m, index n>
returnType< scalar, index > cils::cils< scalar, index, m, n >::cils_back_solve (
    array< scalar, n > & z_B )
```

Parallel version of QR-factorization using modified Gram-Schmidt algorithm, row-oriented

## Parameters

<i>eval</i>	
<i>verbose</i>	
<i>n_proc</i>	

## Returns

Usage Caution: If LLL reduction is applied, please do permutation after getting the result.

## Parameters

<i>n_proc</i>	
<i>nswp</i>	
<i>update</i>	
<i>z_B</i>	
<i>z_B</i> ← <i>_p</i>	

## Returns

Definition at line 112 of file `cils_babai_search.cpp`.

### 5.1.3.4 `cils_back_solve_omp()`

```
template<typename scalar , typename index , index m, index n>
returnType< scalar, index > cils::cils< scalar, index, m, n >::cils_back_solve_omp (
    const index n_proc,
    const index nswp,
    vector< scalar > * z_B )
```

Parallel Babai solver

## Parameters

<i>n_proc</i>	number of Processors/Threads
<i>nswp</i>	maximum number of iterations
<i>z_B</i>	estimation of the true parameter

## Returns

Definition at line 138 of file cils\_babai\_search.cpp.

## 5.1.3.5 cils\_block\_search\_omp()

```
template<typename scalar , typename index , index m, index n>
returnType< scalar, index > cils::cils< scalar, index, m, n >::cils_block_search_omp (
    const index n_proc,
    const index nswp,
    const index init,
    const scalar * y_r,
    const vector< index > * d,
    vector< scalar > * z_B )
```

Parallel version of Block Babai solver

## Parameters

<i>n_proc</i>	
<i>nswp</i>	
<i>stop</i>	
<i>init</i>	
<i>d</i>	
<i>z_B</i>	

## Returns

R\_S\_1[i] &&

Definition at line 167 of file cils\_block\_search.cpp.

## 5.1.3.6 cils\_block\_search\_serial()

```
template<typename scalar , typename index , index m, index n>
returnType< scalar, index > cils::cils< scalar, index, m, n >::cils_block_search_serial (
```

```

const index init,
const scalar * R_R,
const scalar * y_r,
const vector< index > * d,
vector< scalar > * z_B )

```

Definition at line 27 of file `cils_block_search.cpp`.

#### 5.1.3.7 `cils_block_search_serial_CPU`TEST()

```

template<typename scalar , typename index , index m, index n>
returnType< scalar, index > cils::cils< scalar, index, m, n >::cils_block_search_serial_CPU↵
TEST (
    const scalar * R_R,
    const scalar y_r,
    const vector< index > * d,
    vector< scalar > * z_B )

```

Serial Babai solver

##### Parameters

<i>z_B</i>	
<i>is_constrained</i>	

##### Returns

Definition at line 96 of file `cils_block_search.cpp`.

#### 5.1.3.8 `cils_grad_proj`()

```

template<typename scalar , typename index , index m, index n>
returnType< scalar, index > cils::cils< scalar, index, m, n >::cils_grad_proj (
    vector< scalar > & x,
    const index max_iter )

```

Definition at line 258 of file `cils_init_point.cpp`.

### 5.1.3.9 cils\_partition\_deficient()

```
template<typename scalar , typename index , index m, index n>
returnType< scalar, index > cils::cils< scalar, index, m, n >::cils_partition_deficient (
    scalar * z_B,
    scalar * Q_tilde,
    scalar * R_tilde,
    scalar * H_A,
    scalar * Piv_cum )
```

Corresponds to Algorithm 5 (Partition Strategy) in Report 10 [H\_A, P, z, Q\_tilde, R\_tilde, indicator] = partition\_H(A, z\_B, m, n) permutes and partitions H\_A so that the submatrices H\_i are full-column rank

Inputs: A - m-by-n real matrix z\_B - n-dimensional integer vector m - integer scalar n - integer scalar

Outputs: P - n-by-n real matrix, permutation such that H\_A\*P=A z - n-dimensional integer vector (z\_B permuted to correspond to H\_A) Q\_tilde - m-by-n real matrix (Q factors) R\_tilde - m-by-n real matrix (R factors) indicator - 2-by-q integer matrix (indicates submatrices of H\_A)

#### Parameters

<i>z_B</i>	
<i>Q_tilde</i>	
<i>R_tilde</i>	
<i>H_A</i>	
<i>Piv_cum</i>	

#### Returns

Definition at line 7 of file cils\_partition\_deficient.cpp.

### 5.1.3.10 cils\_qrp\_serial()

```
template<typename scalar , typename index , index m, index n>
returnType< scalar, index > cils::cils< scalar, index, m, n >::cils_qrp_serial (
    vector< scalar > & x )
```

Definition at line 116 of file cils\_init\_point.cpp.

### 5.1.3.11 cils\_scp\_block\_babai\_omp()

```
template<typename scalar , typename index , index m, index n>
returnType< scalar, index > cils::cils< scalar, index, m, n >::cils_scp_block_babai_omp (
    vector< scalar > & x_cur,
    scalar v_norm_cur,
    index n_proc,
    index mode )
```

Definition at line 926 of file cils\_block\_search.cpp.

#### 5.1.3.12 cils\_scp\_block\_babai\_serial()

```
template<typename scalar , typename index , index m, index n>
returnType< scalar, index > cils::cils< scalar, index, m, n >::cils_scp_block_babai_serial (
    vector< scalar > & x_cur,
    scalar v_norm_cur,
    index mode )
```

Definition at line 731 of file cils\_block\_search.cpp.

#### 5.1.3.13 cils\_scp\_block\_optimal\_mpi()

```
template<typename scalar , typename index , index m, index n>
returnType< scalar, index > cils::cils< scalar, index, m, n >::cils_scp_block_optimal_mpi (
    vector< scalar > & x_cur,
    scalar * v_norm_cur,
    index size,
    index rank )
```

Definition at line 1490 of file cils\_block\_search.cpp.

#### 5.1.3.14 cils\_scp\_block\_optimal\_omp()

```
template<typename scalar , typename index , index m, index n>
returnType< scalar, index > cils::cils< scalar, index, m, n >::cils_scp_block_optimal_omp (
    vector< scalar > & x_cur,
    scalar v_norm_cur,
    index n_proc,
    index mode )
```

Definition at line 1125 of file cils\_block\_search.cpp.

#### 5.1.3.15 cils\_scp\_block\_optimal\_serial()

```
template<typename scalar , typename index , index m, index n>
returnType< scalar, index > cils::cils< scalar, index, m, n >::cils_scp_block_optimal_serial
(
    vector< scalar > & x_cur,
    scalar v_norm_cur,
    index mode )
```

Applies the SCP-Block Optimal method to obtain a sub-optimal solution

##### Template Parameters

<i>scalar</i>	
<i>index</i>	
<i>m</i>	- integer scalar
<i>n</i>	- integer scalar

## Parameters

<i>x_cur</i>	- n-dimensional integer vector for the sub-optimal solution
<i>v_norm_cur</i>	- real scalar for the norm of the residual vector
<i>max_Babai</i>	- integer scalar, maximum number of calls to block_opt
<i>stopping</i>	- 1-by-3 boolean vector, indicates stopping criterion used

## Returns

{}

Definition at line 349 of file cils\_block\_search.cpp.

## 5.1.3.16 cils\_scp\_block\_suboptimal\_omp()

```
template<typename scalar , typename index , index m, index n>
returnType< scalar, index > cils::cils< scalar, index, m, n >::cils_scp_block_suboptimal_omp
(
    vector< scalar > & x_cur,
    scalar v_norm_cur,
    index n_proc,
    index mode )
```

Definition at line 1288 of file cils\_block\_search.cpp.

## 5.1.3.17 cils\_scp\_block\_suboptimal\_serial()

```
template<typename scalar , typename index , index m, index n>
returnType< scalar, index > cils::cils< scalar, index, m, n >::cils_scp_block_suboptimal_←
serial (
    vector< scalar > & x_cur,
    scalar v_norm_cur,
    index mode )
```

Definition at line 541 of file cils\_block\_search.cpp.

## 5.1.3.18 cils\_sic\_serial()

```
template<typename scalar , typename index , index m, index n>
returnType< scalar, index > cils::cils< scalar, index, m, n >::cils_sic_serial (
    vector< scalar > & x )
```

Definition at line 23 of file cils\_init\_point.cpp.

#### 5.1.3.19 cils\_sic\_subopt()

```
template<typename scalar , typename index , index m, index n>
returnType< scalar, index > cils::cils< scalar, index, m, n >::cils_sic_subopt (
    vector< scalar > & z,
    array< scalar, m > & v_cur,
    array< scalar, m *n > A_t,
    scalar v_norm_cur,
    scalar tolerance,
    index method )
```

Definition at line 22 of file cils\_sic\_opt.cpp.

#### 5.1.3.20 init()

```
template<typename scalar , typename index , index m, index n>
void cils::cils< scalar, index, m, n >::init (
    index rank )
```

Initialize the problem either reading from files (.csv or .nc) or generating the problem

Definition at line 9 of file cils.cpp.

#### 5.1.3.21 init\_R()

```
template<typename scalar , typename index , index m, index n>
void cils::cils< scalar, index, m, n >::init_R ( )
```

#### 5.1.3.22 init\_ud()

```
template<typename scalar , typename index , index m, index n>
void cils::cils< scalar, index, m, n >::init_ud
```

Definition at line 108 of file cils.cpp.

#### 5.1.3.23 init\_y()

```
template<typename scalar , typename index , index m, index n>
void cils::cils< scalar, index, m, n >::init_y ( )
```

Only invoke is function when it is not reading from files and after completed qr!



### 5.1.4 Member Data Documentation

#### 5.1.4.1 `A`

```
template<typename scalar , typename index , index m, index n>  
array<scalar, m * n> cils::cils< scalar, index, m, n >::A
```

Definition at line 71 of file `cils.h`.

#### 5.1.4.2 `H`

```
template<typename scalar , typename index , index m, index n>  
array<scalar, m * n> cils::cils< scalar, index, m, n >::H
```

Definition at line 71 of file `cils.h`.

#### 5.1.4.3 `init_res`

```
template<typename scalar , typename index , index m, index n>  
scalar cils::cils< scalar, index, m, n >::init_res
```

Definition at line 68 of file `cils.h`.

#### 5.1.4.4 `l`

```
template<typename scalar , typename index , index m, index n>  
array<scalar, n> cils::cils< scalar, index, m, n >::l
```

Definition at line 74 of file `cils.h`.

#### 5.1.4.5 `lower`

```
template<typename scalar , typename index , index m, index n>  
index cils::cils< scalar, index, m, n >::lower
```

Definition at line 67 of file `cils.h`.

#### 5.1.4.6 matlabPtr

```
template<typename scalar , typename index , index m, index n>  
std::unique_ptr<matlab::engine::MATLABEngine> cils::cils< scalar, index, m, n >::matlabPtr
```

Definition at line 77 of file cils.h.

#### 5.1.4.7 P

```
template<typename scalar , typename index , index m, index n>  
array<scalar, n * n> cils::cils< scalar, index, m, n >::P
```

Definition at line 72 of file cils.h.

#### 5.1.4.8 qam

```
template<typename scalar , typename index , index m, index n>  
index cils::cils< scalar, index, m, n >::qam
```

Definition at line 67 of file cils.h.

#### 5.1.4.9 R\_A

```
template<typename scalar , typename index , index m, index n>  
array<scalar, m * (n + 1) / 2> cils::cils< scalar, index, m, n >::R_A
```

Definition at line 69 of file cils.h.

#### 5.1.4.10 sigma

```
template<typename scalar , typename index , index m, index n>  
scalar cils::cils< scalar, index, m, n >::sigma
```

Definition at line 68 of file cils.h.

#### 5.1.4.11 snr

```
template<typename scalar , typename index , index m, index n>  
index cils::cils< scalar, index, m, n >::snr
```

Definition at line 67 of file cils.h.

#### 5.1.4.12 tolerance

```
template<typename scalar , typename index , index m, index n>  
scalar cils::cils< scalar, index, m, n >::tolerance
```

Definition at line 68 of file cils.h.

#### 5.1.4.13 u

```
template<typename scalar , typename index , index m, index n>  
array<scalar, n> cils::cils< scalar, index, m, n >::u
```

Definition at line 74 of file cils.h.

#### 5.1.4.14 upper

```
template<typename scalar , typename index , index m, index n>  
index cils::cils< scalar, index, m, n >::upper
```

Definition at line 67 of file cils.h.

#### 5.1.4.15 v\_a

```
template<typename scalar , typename index , index m, index n>  
array<scalar, m> cils::cils< scalar, index, m, n >::v_a
```

Definition at line 75 of file cils.h.

#### 5.1.4.16 v\_q

```
template<typename scalar , typename index , index m, index n>  
array<scalar, m> cils::cils< scalar, index, m, n >::v_q
```

Definition at line 75 of file cils.h.

#### 5.1.4.17 x\_r

```
template<typename scalar , typename index , index m, index n>  
array<scalar, n> cils::cils< scalar, index, m, n >::x_r
```

Definition at line 74 of file cils.h.

#### 5.1.4.18 x\_t

```
template<typename scalar , typename index , index m, index n>
array<scalar, n> cils::cils< scalar, index, m, n >::x_t
```

Definition at line 74 of file cils.h.

#### 5.1.4.19 y\_a

```
template<typename scalar , typename index , index m, index n>
array<scalar, m> cils::cils< scalar, index, m, n >::y_a
```

Definition at line 75 of file cils.h.

#### 5.1.4.20 Z

```
template<typename scalar , typename index , index m, index n>
array<scalar, n * n> cils::cils< scalar, index, m, n >::Z
```

Definition at line 72 of file cils.h.

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

- [src/include/cils.h](#)
- [src/source/cils.cpp](#)
- [src/source/cils\\_babai\\_search.cpp](#)
- [src/source/cils\\_block\\_search.cpp](#)
- [src/source/cils\\_init\\_point.cpp](#)
- [src/source/cils\\_partition\\_deficient.cpp](#)
- [src/source/cils\\_sic\\_opt.cpp](#)

## 5.2 cils::cils\_reduction< scalar, index > Class Template Reference

### Public Member Functions

- [cils\\_reduction](#) (index m, index n, index lower, index upper, bool eval, bool verbose)
- [returnType](#)< scalar, index > [cils\\_eml\\_qr](#) (const scalar \*B)
- [returnType](#)< scalar, index > [cils\\_qr\\_serial](#) (const scalar \*B, const scalar \*y)
- [returnType](#)< scalar, index > [cils\\_qr\\_serial\\_col](#) (const scalar \*B, const scalar \*y)
- [returnType](#)< scalar, index > [cils\\_obils\\_matlab](#) (const vector< scalar > &B, const vector< scalar > &y)
- [returnType](#)< scalar, index > [cils\\_obils\\_reduction](#) (const vector< scalar > &B, const vector< scalar > &y)
- [returnType](#)< scalar, index > [cils\\_obils\\_reduction\\_matlab](#) (const vector< scalar > &B, const vector< scalar > &y)
- [returnType](#)< scalar, index > [cils\\_qr\\_omp](#) (const index n\_proc)
- [returnType](#)< scalar, index > [cils\\_LLL\\_qr\\_reduction](#) (const index n\_proc)
- [returnType](#)< scalar, index > [cils\\_LLL\\_reduction](#) (const index n\_proc)
- [returnType](#)< scalar, index > [cils\\_LLL\\_serial](#) ()
- scalar [cils\\_LLL\\_omp](#) (const index n\_proc)
- [returnType](#)< scalar, index > [cils\\_LLL\\_qr\\_serial](#) ()
- scalar [cils\\_LLL\\_qr\\_omp](#) (const index n\_proc)
- [returnType](#)< scalar, index > [cils\\_qr\\_py](#) (const index eval, const index qr\_eval)
- long int [cils\\_qr\\_py\\_helper](#) ()

## Public Attributes

- vector< scalar > [A](#)
- vector< scalar > [R\\_Q](#)
- vector< scalar > [R\\_R](#)
- vector< scalar > [Q](#)
- vector< scalar > [G](#)
- vector< scalar > [P](#)
- vector< scalar > [Z](#)
- vector< scalar > [p](#)
- vector< scalar > [y\\_a](#)
- vector< scalar > [y\\_q](#)
- vector< scalar > [y\\_r](#)

### 5.2.1 Detailed Description

```
template<typename scalar, typename index>
class cils::cils_reduction< scalar, index >
```

Definition at line 9 of file cils\_reduction.cpp.

### 5.2.2 Constructor & Destructor Documentation

#### 5.2.2.1 cils\_reduction()

```
template<typename scalar , typename index >
cils::cils_reduction< scalar, index >::cils_reduction (
    index m,
    index n,
    index lower,
    index upper,
    bool eval,
    bool verbose ) [inline]
```

Definition at line 652 of file cils\_reduction.cpp.

### 5.2.3 Member Function Documentation

#### 5.2.3.1 cils\_eml\_qr()

```
template<typename scalar , typename index >
returnType<scalar, index> cils::cils_reduction< scalar, index >::cils_eml_qr (
    const scalar * B ) [inline]
```

Serial version of QR-factorization with column pivoting Results are stored in the class object.

## Parameters

$B$	: m-by-n input matrix
-----	-----------------------

Definition at line 675 of file cils\_reduction.cpp.

### 5.2.3.2 cils\_LLL\_omp()

```
template<typename scalar , typename index >
scalar cils::cils_reduction< scalar, index >::cils_LLL_omp (
    const index n_proc ) [inline]
```

Definition at line 1831 of file cils\_reduction.cpp.

### 5.2.3.3 cils\_LLL\_qr\_omp()

```
template<typename scalar , typename index >
scalar cils::cils_reduction< scalar, index >::cils_LLL_qr_omp (
    const index n_proc ) [inline]
```

Definition at line 2398 of file cils\_reduction.cpp.

### 5.2.3.4 cils\_LLL\_qr\_reduction()

```
template<typename scalar , typename index >
returnType<scalar, index> cils::cils_reduction< scalar, index >::cils_LLL_qr_reduction (
    const index n_proc ) [inline]
```

Definition at line 1504 of file cils\_reduction.cpp.

### 5.2.3.5 cils\_LLL\_qr\_serial()

```
template<typename scalar , typename index >
returnType<scalar, index> cils::cils_reduction< scalar, index >::cils_LLL_qr_serial ( ) [inline]
```

Definition at line 2068 of file cils\_reduction.cpp.

### 5.2.3.6 cils\_LLL\_reduction()

```
template<typename scalar , typename index >
returnType<scalar, index> cils::cils_reduction< scalar, index >::cils_LLL_reduction (
    const index n_proc ) [inline]
```

Definition at line 1553 of file cils\_reduction.cpp.

### 5.2.3.7 cils\_LLL\_serial()

```
template<typename scalar , typename index >
returnType<scalar, index> cils::cils_reduction< scalar, index >::cils_LLL_serial ( ) [inline]
```

Definition at line 1604 of file cils\_reduction.cpp.

### 5.2.3.8 cils\_obils\_matlab()

```
template<typename scalar , typename index >
returnType<scalar, index> cils::cils_reduction< scalar, index >::cils_obils_matlab (
    const vector< scalar > & B,
    const vector< scalar > & y ) [inline]
```

Matlab Caller of obils. Results are stored in the class object.

#### Parameters

<i>B</i>	: m-by-n input matrix
<i>y</i>	: m-by-1 input right hand vector

Definition at line 930 of file cils\_reduction.cpp.

### 5.2.3.9 cils\_obils\_reduction()

```
template<typename scalar , typename index >
returnType<scalar, index> cils::cils_reduction< scalar, index >::cils_obils_reduction (
    const vector< scalar > & B,
    const vector< scalar > & y ) [inline]
```

Definition at line 984 of file cils\_reduction.cpp.

**5.2.3.10 cils\_obils\_reduction\_matlab()**

```
template<typename scalar , typename index >
returnType<scalar, index> cils::cils_reduction< scalar, index >::cils_obils_reduction_matlab
(
    const vector< scalar > & B,
    const vector< scalar > & y ) [inline]
```

Definition at line 1303 of file cils\_reduction.cpp.

**5.2.3.11 cils\_qr\_omp()**

```
template<typename scalar , typename index >
returnType<scalar, index> cils::cils_reduction< scalar, index >::cils_qr_omp (
    const index n_proc ) [inline]
```

Definition at line 1351 of file cils\_reduction.cpp.

**5.2.3.12 cils\_qr\_py()**

```
template<typename scalar , typename index >
returnType<scalar, index> cils::cils_reduction< scalar, index >::cils_qr_py (
    const index eval,
    const index qr_eval ) [inline]
```

Definition at line 2821 of file cils\_reduction.cpp.

**5.2.3.13 cils\_qr\_py\_helper()**

```
template<typename scalar , typename index >
long int cils::cils_reduction< scalar, index >::cils_qr_py_helper ( ) [inline]
```

Definition at line 2835 of file cils\_reduction.cpp.

**5.2.3.14 cils\_qr\_serial()**

```
template<typename scalar , typename index >
returnType<scalar, index> cils::cils_reduction< scalar, index >::cils_qr_serial (
    const scalar * B,
    const scalar * y ) [inline]
```

Serial version of FULL QR-factorization using modified Gram-Schmidt algorithm, row-oriented Results are stored in the class object.



**Parameters**

$B$	: m-by-n input matrix
$y$	: m-by-1 input right hand vector

Definition at line 768 of file cils\_reduction.cpp.

**5.2.3.15 cils\_qr\_serial\_col()**

```
template<typename scalar , typename index >
returnType<scalar, index> cils::cils_reduction< scalar, index >::cils_qr_serial_col (
    const scalar * B,
    const scalar * y ) [inline]
```

Serial version of REDUCED QR-factorization using modified Gram-Schmidt algorithm, col-oriented Results are stored in the class object.

**Parameters**

$B$	: m-by-n input matrix
$y$	: m-by-1 input right hand vector

Definition at line 838 of file cils\_reduction.cpp.

**5.2.4 Member Data Documentation****5.2.4.1 A**

```
template<typename scalar , typename index >
vector<scalar> cils::cils_reduction< scalar, index >::A
```

Definition at line 647 of file cils\_reduction.cpp.

**5.2.4.2 G**

```
template<typename scalar , typename index >
vector<scalar> cils::cils_reduction< scalar, index >::G
```

Definition at line 647 of file cils\_reduction.cpp.

#### 5.2.4.3 P

```
template<typename scalar , typename index >  
vector<scalar> cils::cils_reduction< scalar, index >::P
```

Definition at line 647 of file cils\_reduction.cpp.

#### 5.2.4.4 p

```
template<typename scalar , typename index >  
vector<scalar> cils::cils_reduction< scalar, index >::p
```

Definition at line 648 of file cils\_reduction.cpp.

#### 5.2.4.5 Q

```
template<typename scalar , typename index >  
vector<scalar> cils::cils_reduction< scalar, index >::Q
```

Definition at line 647 of file cils\_reduction.cpp.

#### 5.2.4.6 R\_Q

```
template<typename scalar , typename index >  
vector<scalar> cils::cils_reduction< scalar, index >::R_Q
```

Definition at line 647 of file cils\_reduction.cpp.

#### 5.2.4.7 R\_R

```
template<typename scalar , typename index >  
vector<scalar> cils::cils_reduction< scalar, index >::R_R
```

Definition at line 647 of file cils\_reduction.cpp.

#### 5.2.4.8 y\_a

```
template<typename scalar , typename index >  
vector<scalar> cils::cils_reduction< scalar, index >::y_a
```

Definition at line 649 of file cils\_reduction.cpp.

#### 5.2.4.9 y\_q

```
template<typename scalar , typename index >
vector<scalar> cils::cils_reduction< scalar, index >::y_q
```

Definition at line 649 of file cils\_reduction.cpp.

#### 5.2.4.10 y\_r

```
template<typename scalar , typename index >
vector<scalar> cils::cils_reduction< scalar, index >::y_r
```

Definition at line 649 of file cils\_reduction.cpp.

#### 5.2.4.11 Z

```
template<typename scalar , typename index >
vector<scalar> cils::cils_reduction< scalar, index >::Z
```

Definition at line 648 of file cils\_reduction.cpp.

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

- [src/source/cils\\_reduction.cpp](#)

## 5.3 cils::cils\_search< scalar, index > Class Template Reference

### Public Member Functions

- [cils\\_search](#) (index m, index n, index qam)
- void [obils\\_search\\_matlab](#) (const index n\_dx\_q\_0, const index n\_dx\_q\_1, const bool check, vector< scalar > &R\_R, vector< scalar > &y\_B, vector< scalar > &z\_x)
- bool [obils\\_search2](#) (const vector< scalar > &R, const vector< scalar > &y, vector< scalar > &zhat)
- bool [obils\\_search](#) (const index n\_dx\_q\_0, const index n\_dx\_q\_1, const bool check, const scalar \*R\_R, const scalar \*y\_B, vector< scalar > &z\_x)
- bool [obils\\_search\\_omp](#) (const index n\_dx\_q\_0, const index n\_dx\_q\_1, const index i, const index check, const scalar \*R\_A, const scalar \*y\_B, scalar \*z\_x)
- bool [ils\\_search](#) (const index n\_dx\_q\_0, const index n\_dx\_q\_1, const bool check, const scalar \*R\_R, const scalar \*y\_B, vector< scalar > \*z\_x)
- bool [ils\\_search\\_omp](#) (const index n\_dx\_q\_0, const index n\_dx\_q\_1, const index i, const index check, const scalar \*R\_A, const scalar \*y\_B, scalar \*z\_x)
- bool [ubils\\_search](#) (const index n\_dx\_q\_0, const index n\_dx\_q\_1, const bool check, scalar beta, const scalar \*R\_R, const scalar \*y\_B, vector< scalar > &z\_x)
- void [bound](#) (double c\_k, const scalar \*R\_R, double beta, const scalar \*prsd, double k, double \*l\_k, double \*u\_k)

### 5.3.1 Detailed Description

```
template<typename scalar, typename index>
class cils::cils_search< scalar, index >
```

Definition at line 22 of file cils\_ils\_search.cpp.

### 5.3.2 Constructor & Destructor Documentation

#### 5.3.2.1 cils\_search()

```
template<typename scalar , typename index >
cils::cils_search< scalar, index >::cils_search (
    index m,
    index n,
    index qam ) [inline]
```

Definition at line 80 of file cils\_ils\_search.cpp.

### 5.3.3 Member Function Documentation

#### 5.3.3.1 bound()

```
template<typename scalar , typename index >
void cils::cils_search< scalar, index >::bound (
    double c_k,
    const scalar * R_R,
    double beta,
    const scalar * prsd,
    double k,
    double * l_k,
    double * u_k ) [inline]
```

Definition at line 1113 of file cils\_ils\_search.cpp.

#### 5.3.3.2 ils\_search()

```
template<typename scalar , typename index >
bool cils::cils_search< scalar, index >::ils_search (
    const index n_dx_q_0,
    const index n_dx_q_1,
    const bool check,
    const scalar * R_R,
    const scalar * y_B,
    vector< scalar > * z_x ) [inline]
```

Definition at line 593 of file cils\_ils\_search.cpp.

### 5.3.3.3 ils\_search\_omp()

```
template<typename scalar , typename index >
bool cils::cils_search< scalar, index >::ils_search_omp (
    const index n_dx_q_0,
    const index n_dx_q_1,
    const index i,
    const index check,
    const scalar * R_A,
    const scalar * y_B,
    scalar * z_x ) [inline]
```

Definition at line 663 of file cils\_ils\_search.cpp.

### 5.3.3.4 obils\_search()

```
template<typename scalar , typename index >
bool cils::cils_search< scalar, index >::obils_search (
    const index n_dx_q_0,
    const index n_dx_q_1,
    const bool check,
    const scalar * R_R,
    const scalar * y_B,
    vector< scalar > & z_x ) [inline]
```

Definition at line 372 of file cils\_ils\_search.cpp.

### 5.3.3.5 obils\_search2()

```
template<typename scalar , typename index >
bool cils::cils_search< scalar, index >::obils_search2 (
    const vector< scalar > & R,
    const vector< scalar > & y,
    vector< scalar > & zhat ) [inline]
```

Definition at line 133 of file cils\_ils\_search.cpp.

### 5.3.3.6 obils\_search\_matlab()

```
template<typename scalar , typename index >
void cils::cils_search< scalar, index >::obils_search_matlab (
    const index n_dx_q_0,
    const index n_dx_q_1,
    const bool check,
    vector< scalar > & R_R,
    vector< scalar > & y_B,
    vector< scalar > & z_x ) [inline]
```

Definition at line 101 of file cils\_ils\_search.cpp.

### 5.3.3.7 obils\_search\_omp()

```
template<typename scalar , typename index >
bool cils::cils_search< scalar, index >::obils_search_omp (
    const index n_dx_q_0,
    const index n_dx_q_1,
    const index i,
    const index check,
    const scalar * R_A,
    const scalar * y_B,
    scalar * z_x ) [inline]
```

Definition at line 483 of file cils\_ils\_search.cpp.

### 5.3.3.8 ubils\_search()

```
template<typename scalar , typename index >
bool cils::cils_search< scalar, index >::ubils_search (
    const index n_dx_q_0,
    const index n_dx_q_1,
    const bool check,
    scalar beta,
    const scalar * R_R,
    const scalar * y_B,
    vector< scalar > & z_x ) [inline]
```

Definition at line 739 of file cils\_ils\_search.cpp.

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

- [src/source/cils\\_ils\\_search.cpp](#)

## 5.4 cils::returnType< scalar, index > Struct Template Reference

```
#include <cils.h>
```

### Public Attributes

- vector< scalar > [x](#)
- scalar [run\\_time](#)
- scalar [info](#)

### 5.4.1 Detailed Description

```
template<typename scalar, typename index>
struct cils::returnType< scalar, index >
```

Return scalar pointer array along with the size.

## Template Parameters

<i>scalar</i>	
<i>index</i>	

Definition at line 55 of file `cils.h`.

## 5.4.2 Member Data Documentation

### 5.4.2.1 info

```
template<typename scalar , typename index >  
scalar cils::returnType< scalar, index >::info
```

Definition at line 58 of file `cils.h`.

### 5.4.2.2 run\_time

```
template<typename scalar , typename index >  
scalar cils::returnType< scalar, index >::run_time
```

Definition at line 57 of file `cils.h`.

### 5.4.2.3 x

```
template<typename scalar , typename index >  
vector<scalar> cils::returnType< scalar, index >::x
```

Definition at line 56 of file `cils.h`.

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

- `src/include/cils.h`





## Chapter 6

# File Documentation

### 6.1 src/example/cils\_standard\_test.cpp File Reference

```
#include "../source/cils.cpp"
#include "../source/cils_ils_search.cpp"
#include "../source/cils_block_search.cpp"
#include "../source/cils_babai_search.cpp"
#include "../source/cils_reduction.cpp"
#include "../source/cils_init_point.cpp"
#include <ctime>
```

Include dependency graph for cils\_standard\_test.cpp:

### 6.2 src/example/cils\_underdetermined\_test.cpp File Reference

```
#include "../source/cils.cpp"
#include "../source/cils_block_search.cpp"
#include "../source/cils_init_point.cpp"
#include "../source/cils_sic_opt.cpp"
#include "../source/cils_partition_deficient.cpp"
#include <ctime>
```

Include dependency graph for cils\_underdetermined\_test.cpp:

## Functions

- template<typename scalar , typename index , index m, index n>  
void [block\\_optimal\\_test](#) (int size, int rank)
- template<typename scalar , typename index , index m, index n>  
long [plot\\_run](#) (int size, int rank)

#### 6.2.1 Function Documentation

### 6.2.1.1 block\_optimal\_test()

```
template<typename scalar , typename index , index m, index n>
void block_optimal_test (
    int size,
    int rank )
```

Definition at line 92 of file cils\_underdetermined\_test.cpp.

### 6.2.1.2 plot\_run()

```
template<typename scalar , typename index , index m, index n>
long plot_run (
    int size,
    int rank )
```

Definition at line 253 of file cils\_underdetermined\_test.cpp.

## 6.3 src/example/plot\_helper.py File Reference

### Namespaces

- [plot\\_helper](#)

### Functions

- def [plot\\_helper.plot\\_runtime\\_ud](#) (n, SNR, k, l\_max, max\_iter, res, ber, tim, proc\_num, spu, max\_proc, min\_proc, is\_constrained, m)
- def [plot\\_helper.plot\\_runtime\\_III](#) (n, qr\_l, i, max\_proc, min\_proc, qrT, III, III\_qr, qr\_spu, III\_spu, III\_qr\_spu, qIII\_spu)
- def [plot\\_helper.plot\\_runtime](#) (n, SNR, k, l\_max, block\_size, max\_iter, is\_qr, res, ber, tim, itr, ser\_tim, d\_s, proc\_num, spu, time, qr\_l, max\_proc, min\_proc, qrT, III, III\_qr, qr\_spu, III\_spu, III\_qr\_spu, qIII\_spu, tpu, is\_constrained, m)
- def [plot\\_helper.plot\\_first\\_block](#) (n, SNR, k, block\_size, ser\_tim, is\_qr, d\_s)

### Variables

- int [plot\\_helper.n](#) = 30
- int [plot\\_helper.SNR](#) = 35
- string [plot\\_helper.title3](#) = 'underdetermined'
- int [plot\\_helper.max\\_iter](#) = 1
- string [plot\\_helper.title1](#) = 'Box-constrained'
- [plot\\_helper.a](#) = np.load(f'../cmake-build-release/{n}\_report\_plot\_{SNR}\_{title3}\_{int(max\_iter / 100)}\_{title1}.npz')
- [plot\\_helper.m](#) = a['m']
- [plot\\_helper.k](#) = a['k']
- [plot\\_helper.l\\_max](#) = a['l\_max']
- [plot\\_helper.res](#) = a['res']
- [plot\\_helper.ber](#) = a['ber']
- [plot\\_helper.tim](#) = a['tim']
- [plot\\_helper.spu](#) = a['spu']
- [plot\\_helper.proc\\_num](#) = a['proc\_num']
- [plot\\_helper.max\\_proc](#) = a['max\_proc']
- [plot\\_helper.min\\_proc](#) = a['min\_proc']
- [plot\\_helper.is\\_constrained](#) = a['is\_constrained']

## 6.4 src/example/py\_qr.py File Reference

### Namespaces

- [py\\_qr](#)

### Functions

- [def py\\_qr.check\\_mem \(\)](#)

### Variables

- [py\\_qr.total](#) = int(total)
- [py\\_qr.used](#) = int(used)
- [py\\_qr.max\\_mem](#) = int(total \* 0.8)
- [py\\_qr.block\\_mem](#) = max\_mem - used
- [py\\_qr.x](#) = torch.rand((256,1024,block\_mem)).cuda()

## 6.5 src/include/cils.h File Reference

Computation of integer least square problem.

```
#include <iostream>
#include <omp.h>
#include <cstdio>
#include <cstdlib>
#include <fstream>
#include <string>
#include <vector>
#include <random>
#include <ctime>
#include <iomanip>
#include <algorithm>
#include <netcdf.h>
#include <bitset>
#include <cmath>
#include "config.h"
#include "MatlabDataArray.hpp"
#include "MatlabEngine.hpp"
#include <numeric>
#include "mpi.h"
```

Include dependency graph for cils.h: This graph shows which files directly or indirectly include this file:

### Classes

- struct [cils::returnType< scalar, index >](#)
- class [cils::cils< scalar, index, m, n >](#)

## Namespaces

- [cils](#)

### 6.5.1 Detailed Description

Computation of integer least square problem.

#### Author

Shilei Lin This file is part of CILS. CILS is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

CILS is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with CILS. If not, see <http://www.gnu.org/licenses/>.

## 6.6 src/include/config.h File Reference

```
#include <vector>
#include <string>
#include <cmath>
#include <iostream>
#include <climits>
#include "helper.h"
```

Include dependency graph for config.h: This graph shows which files directly or indirectly include this file:

## Namespaces

- [cils](#)
- [cils::program\\_def](#)

## Typedefs

- typedef int [cils::program\\_def::index](#)
- typedef double [cils::program\\_def::scalar](#)

## Functions

- std::vector< index > [cils::program\\_def::d\\_s](#) (N/block\_size+spilt\_size - 1, block\_size)
- std::vector< index > [cils::program\\_def::indicator](#) (2 \*q, 0)
- vector< vector< scalar > > [cils::program\\_def::permutation](#) (search\_iter+3)
- void [cils::program\\_def::init\\_program\\_def](#) (int argc, char \*argv[])
- template<typename scalar, typename index, index m, index n>  
void [cils::program\\_def::init\\_guess](#) (index init\_value, vector< scalar > \*z\_B, scalar \*x\_R)

## Variables

- index `cils::program_def::qam` = 1
- index `cils::program_def::SNR` = 15
- index `cils::program_def::max_iter` = 100
- index `cils::program_def::search_iter` = 1e5
- index `cils::program_def::stop` = 3
- index `cils::program_def::schedule` = 2
- index `cils::program_def::chunk_size` = 1
- index `cils::program_def::block_size` = 2
- index `cils::program_def::spilt_size` = 2
- index `cils::program_def::offset` = 2
- index `cils::program_def::is_constrained` = true
- index `cils::program_def::is_nc` = false
- index `cils::program_def::is_matlab` = false
- index `cils::program_def::is_qr` = false
- index `cils::program_def::mode` = 1
- index `cils::program_def::num_trials` = 10
- index `cils::program_def::is_local` = 1
- index `cils::program_def::max_search` = 400000
- index `cils::program_def::min_proc` = 2
- index `cils::program_def::plot_itr` = 1
- scalar `cils::program_def::coeff` = 17.5
- index `cils::program_def::max_proc` = 10
- index `cils::program_def::max_thre` = 400000
- auto `cils::program_def::q` = `static_cast<index>(std::ceil((scalar) N / (scalar) M))`
- index `cils::program_def::verbose` = false
- index `cils::program_def::chunk` = 1

## 6.7 src/include/helper.h File Reference

Computation of integer least square problem.

```
#include <cmath>
#include <cstring>
```

Include dependency graph for helper.h: This graph shows which files directly or indirectly include this file:

## Namespaces

- `helper`

## Functions

- void [helper::b\\_rand](#) (const int n, double \*r)
- void [helper::randperm](#) (int n, double \*p)
- template<typename scalar , typename index >  
void [helper::eye](#) (index n, scalar \*A)
- template<typename scalar , typename index , index m, index n, index mb>  
void [helper::b\\_mtimes](#) (const array< scalar, m \*n > &A\_C, const array< scalar, n \*mb > &B, array< scalar, m \*mb > &C)
- template<typename scalar , typename index , index m, index n>  
void [helper::mtimes](#) (const scalar A\_C[4], const array< scalar, m \*n > &B, array< scalar, 2 \*n > &C)
- template<typename scalar , typename index , index m, index n>  
void [helper::mtimes](#) (const array< scalar, m \*m > &Q, const array< scalar, m \*n > &R, array< scalar, m \*n > &A\_t)
- template<typename scalar , typename index >  
void [helper::mtimes\\_v](#) (index m, index n, const vector< scalar > &Q, const vector< scalar > &R, vector< scalar > &A\_t)
- template<typename scalar , typename index >  
void [helper::mtimes\\_col](#) (index m, index n, const vector< scalar > &Q, const vector< scalar > &R, vector< scalar > &A\_t)
- template<typename scalar , typename index >  
void [helper::mtimes\\_AP](#) (index m, index n, const scalar \*A, const scalar \*P, scalar \*A\_t)
- template<typename scalar , typename index >  
void [helper::inv](#) (const index K, const index N, const vector< scalar > &x, vector< scalar > &y)
- template<typename scalar , typename index >  
void [helper::mtimes\\_Axy](#) (const index m, const index n, const scalar \*A, const scalar \*x, scalar \*y)
- template<typename scalar , typename index , index n>  
index [helper::length\\_nonzeros](#) (const scalar \*x, const scalar \*y)
- template<typename scalar , typename index >  
void [helper::planerot](#) (scalar x[2], scalar G[4])
- template<typename scalar , typename index >  
scalar [helper::norm](#) (const index n, const scalar \*v)
- template<typename scalar , typename index >  
scalar [helper::find\\_bit\\_error\\_rate](#) (const index n, const scalar \*x\_b, const scalar \*x\_t, const index k)
- template<typename scalar , typename index >  
void [helper::display\\_matrix](#) (index m, index n, const scalar \*x, const string &name)
- template<typename scalar , typename index >  
void [helper::display\\_vector](#) (const index n, const scalar \*x, const string &name)
- template<typename index >  
bool [helper::if\\_all\\_x\\_true](#) (const vector< bool > &x)
- template<typename scalar , typename index >  
void [helper::unique\\_vector](#) (const vector< scalar > &a, vector< scalar > &b)
- template<typename scalar , typename index >  
scalar [helper::find\\_residual](#) (const index m, const index n, const scalar \*A, const scalar \*x, const scalar \*y)

## Variables

- const double [ZERO](#) = 3.3121686421112381E-170

### 6.7.1 Detailed Description

Computation of integer least square problem.

**Author**

Shilei Lin This file is part of CILS. CILS is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

CILS is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with CILS. If not, see <http://www.gnu.org/licenses/>.

**6.7.2 Variable Documentation****6.7.2.1 ZERO**

```
const double ZERO = 3.3121686421112381E-170
```

Definition at line 27 of file helper.h.

**6.8 src/source/cils.cpp File Reference**

```
#include "../include/cils.h"
```

Include dependency graph for cils.cpp: This graph shows which files directly or indirectly include this file:

**Namespaces**

- [cils](#)

**Functions**

- `template<typename scalar , typename index , index m, index n>  
void cils::matrix\_vector\_mult (const array< scalar, m *n > &Z, vector< scalar > *x)`
- `template<typename scalar , typename index , index m, index n>  
void cils::matrix\_vector\_mult (const array< scalar, m *n > &Z, const vector< scalar > &x, vector< scalar > &c)`

**6.9 src/source/cils\_babai\_search.cpp File Reference**

Computation of integer least square problem by constrained non-block Babai Estimator.

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

## Namespaces

- [cils](#)

### 6.9.1 Detailed Description

Computation of integer least square problem by constrained non-block Babai Estimator.

#### Author

Shilei Lin This file is part of CILS. CILS is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

CILS is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with CILS. If not, see <http://www.gnu.org/licenses/>.

## 6.10 src/source/cils\_block\_search.cpp File Reference

Computation of Block Babai Algorithm.

```
#include "cils_reduction.cpp"
#include "cils_ils_search.cpp"
#include <mpi.h>
```

Include dependency graph for cils\_block\_search.cpp: This graph shows which files directly or indirectly include this file:

## Namespaces

- [cils](#)

### 6.10.1 Detailed Description

Computation of Block Babai Algorithm.

#### Author

Shilei Lin This file is part of CILS. CILS is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

CILS is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR P\_A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with CILS. If not, see <http://www.gnu.org/licenses/>.



## 6.11 src/source/cils\_block\_search\_GPU.cpp File Reference

```
#include <OpenCL/cl.hpp>
```

Include dependency graph for cils\_block\_search\_GPU.cpp:

### Macros

- `#define NUM_GLOBAL_WITEMS 1024`

### Functions

- void `compareResults` (double CPUtime, double GPUtime, int trial)
- double `timeAddVectorsCPU` (int n, int k)
- void `warmup` (cl::Context &context, cl::CommandQueue &queue, cl::Kernel &add, int A[], int B[], int n)
- int `run` (int argc, char \*argv[])

### 6.11.1 Macro Definition Documentation

#### 6.11.1.1 NUM\_GLOBAL\_WITEMS

```
#define NUM_GLOBAL_WITEMS 1024
```

Definition at line 6 of file cils\_block\_search\_GPU.cpp.

### 6.11.2 Function Documentation

#### 6.11.2.1 compareResults()

```
void compareResults (  
    double CPUtime,  
    double GPUtime,  
    int trial )
```

Definition at line 8 of file cils\_block\_search\_GPU.cpp.

### 6.11.2.2 run()

```
int run (
    int argc,
    char * argv[] )
```

Definition at line 68 of file `cils_block_search_GPU.cpp`.

### 6.11.2.3 timeAddVectorsCPU()

```
double timeAddVectorsCPU (
    int n,
    int k )
```

Definition at line 21 of file `cils_block_search_GPU.cpp`.

### 6.11.2.4 warmup()

```
void warmup (
    cl::Context & context,
    cl::CommandQueue & queue,
    cl::Kernel & add,
    int A[],
    int B[],
    int n )
```

Definition at line 44 of file `cils_block_search_GPU.cpp`.

## 6.12 src/source/cils\_ils\_search.cpp File Reference

Computation of SS\_search Algorithm.

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

### Classes

- class [cils::cils\\_search< scalar, index >](#)

### Namespaces

- [cils](#)

### 6.12.1 Detailed Description

Computation of SS\_search Algorithm.

#### Author

Shilei Lin This file is part of CILS. CILS is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

CILS is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with CILS. If not, see <http://www.gnu.org/licenses/>.

## 6.13 src/source/cils\_init\_point.cpp File Reference

Computation of indexeger least square problem by constrained non-blocl Babai Estimator.

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

### Namespaces

- [cils](#)

### 6.13.1 Detailed Description

Computation of indexeger least square problem by constrained non-blocl Babai Estimator.

#### Author

Shilei Lin This file is part of CILS. CILS is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

CILS is distributed in the hope that it will be useful, but WITAOUT ANY WARRANTY; without even the implied warranty of MERCA\_tNTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with CILS. If not, see <http://www.gnu.org/licenses/>.

## 6.14 src/source/cils\_partition\_deficient.cpp File Reference

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

## Namespaces

- [cils](#)

## 6.15 src/source/cils\_reduction.cpp File Reference

```
#include <cstring>
#include <Python.h>
#include <numpy/arrayobject.h>
```

Include dependency graph for cils\_reduction.cpp: This graph shows which files directly or indirectly include this file:

## Classes

- class [cils::cils\\_reduction< scalar, index >](#)

## Namespaces

- [cils](#)

## 6.16 src/source/cils\_sic\_opt.cpp File Reference

Computation of indexeger least square problem by constrained non-blocl Babai Estimator.

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

## Namespaces

- [cils](#)

### 6.16.1 Detailed Description

Computation of indexeger least square problem by constrained non-blocl Babai Estimator.

#### Author

Shilei Lin This file is part of CILS. CILS is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

CILS is distributed in the hope that it will be useful, but WITAOUT ANY WARRANTY; without even the implied warranty of MERCA\_tNTABILITY or FITNESS FOR A\_t PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with CILS. If not, see <http://www.gnu.org/licenses/>.

# Index

- ~cils
  - cils::cils< scalar, index, m, n >, [32](#)
- A
  - cils::cils< scalar, index, m, n >, [41](#)
  - cils::cils\_reduction< scalar, index >, [49](#)
- a
  - plot\_helper, [26](#)
- b\_mtimes
  - helper, [16](#)
- b\_rand
  - helper, [16](#)
- ber
  - plot\_helper, [26](#)
- block\_mem
  - py\_qr, [29](#)
- block\_optimal\_test
  - cils\_underdetermined\_test.cpp, [57](#)
- block\_size
  - cils::program\_def, [10](#)
- bound
  - cils::cils\_search< scalar, index >, [52](#)
- check\_mem
  - py\_qr, [29](#)
- chunk
  - cils::program\_def, [10](#)
- chunk\_size
  - cils::program\_def, [11](#)
- cils, [7](#)
  - cils::cils< scalar, index, m, n >, [32](#)
  - matrix\_vector\_mult, [7, 8](#)
- cils::cils< scalar, index, m, n >, [31](#)
  - ~cils, [32](#)
  - A, [41](#)
  - cils, [32](#)
  - cils\_babai\_search\_omp, [33](#)
  - cils\_babai\_search\_serial, [33](#)
  - cils\_back\_solve, [34](#)
  - cils\_back\_solve\_omp, [34](#)
  - cils\_block\_search\_omp, [35](#)
  - cils\_block\_search\_serial, [35](#)
  - cils\_block\_search\_serial\_CPUEST, [36](#)
  - cils\_grad\_proj, [36](#)
  - cils\_partition\_deficient, [36](#)
  - cils\_qrp\_serial, [37](#)
  - cils\_scp\_block\_babai\_omp, [37](#)
  - cils\_scp\_block\_babai\_serial, [37](#)
  - cils\_scp\_block\_optimal\_mpi, [38](#)
  - cils\_scp\_block\_optimal\_omp, [38](#)
  - cils\_scp\_block\_optimal\_serial, [38](#)
  - cils\_scp\_block\_suboptimal\_omp, [39](#)
  - cils\_scp\_block\_suboptimal\_serial, [39](#)
  - cils\_sic\_serial, [39](#)
  - cils\_sic\_subopt, [39](#)
  - H, [41](#)
  - init, [40](#)
  - init\_R, [40](#)
  - init\_res, [41](#)
  - init\_ud, [40](#)
  - init\_y, [40](#)
  - I, [41](#)
  - lower, [41](#)
  - matlabPtr, [41](#)
  - P, [42](#)
  - qam, [42](#)
  - R\_A, [42](#)
  - sigma, [42](#)
  - snr, [42](#)
  - tolerance, [42](#)
  - u, [43](#)
  - upper, [43](#)
  - v\_a, [43](#)
  - v\_q, [43](#)
  - x\_r, [43](#)
  - x\_t, [43](#)
  - y\_a, [44](#)
  - Z, [44](#)
- cils::cils\_reduction< scalar, index >, [44](#)
  - A, [49](#)
  - cils\_eml\_qr, [45](#)
  - cils\_LLL\_omp, [46](#)
  - cils\_LLL\_qr\_omp, [46](#)
  - cils\_LLL\_qr\_reduction, [46](#)
  - cils\_LLL\_qr\_serial, [46](#)
  - cils\_LLL\_reduction, [46](#)
  - cils\_LLL\_serial, [47](#)
  - cils\_obils\_matlab, [47](#)
  - cils\_obils\_reduction, [47](#)
  - cils\_obils\_reduction\_matlab, [47](#)
  - cils\_qr\_omp, [48](#)
  - cils\_qr\_py, [48](#)
  - cils\_qr\_py\_helper, [48](#)
  - cils\_qr\_serial, [48](#)
  - cils\_qr\_serial\_col, [49](#)
  - cils\_reduction, [45](#)
  - G, [49](#)
  - P, [49](#)

- p, 50
- Q, 50
- R\_Q, 50
- R\_R, 50
- y\_a, 50
- y\_q, 50
- y\_r, 51
- Z, 51
- cils::cils\_search< scalar, index >, 51
  - bound, 52
  - cils\_search, 52
  - ils\_search, 52
  - ils\_search\_omp, 52
  - obils\_search, 53
  - obils\_search2, 53
  - obils\_search\_matlab, 53
  - obils\_search\_omp, 53
  - ubils\_search, 54
- cils::program\_def, 8
  - block\_size, 10
  - chunk, 10
  - chunk\_size, 11
  - coeff, 11
  - d\_s, 9
  - index, 9
  - indicator, 9
  - init\_guess, 10
  - init\_program\_def, 10
  - is\_constrained, 11
  - is\_local, 11
  - is\_matlab, 11
  - is\_nc, 11
  - is\_qr, 12
  - max\_iter, 12
  - max\_proc, 12
  - max\_search, 12
  - max\_thre, 12
  - min\_proc, 12
  - mode, 13
  - num\_trials, 13
  - offset, 13
  - permutation, 10
  - plot\_itr, 13
  - q, 13
  - qam, 13
  - scalar, 9
  - schedule, 14
  - search\_iter, 14
  - SNR, 14
  - spilt\_size, 14
  - stop, 14
  - verbose, 14
- cils::returnType< scalar, index >, 54
  - info, 55
  - run\_time, 55
  - x, 55
- cils\_babai\_search\_omp
  - cils::cils< scalar, index, m, n >, 33
- cils\_babai\_search\_serial
  - cils::cils< scalar, index, m, n >, 33
- cils\_back\_solve
  - cils::cils< scalar, index, m, n >, 34
- cils\_back\_solve\_omp
  - cils::cils< scalar, index, m, n >, 34
- cils\_block\_search\_GPU.cpp
  - compareResults, 65
  - NUM\_GLOBAL\_WITEMS, 65
  - run, 65
  - timeAddVectorsCPU, 66
  - warmup, 66
- cils\_block\_search\_omp
  - cils::cils< scalar, index, m, n >, 35
- cils\_block\_search\_serial
  - cils::cils< scalar, index, m, n >, 35
- cils\_block\_search\_serial\_CPTEST
  - cils::cils< scalar, index, m, n >, 36
- cils\_eml\_qr
  - cils::cils\_reduction< scalar, index >, 45
- cils\_grad\_proj
  - cils::cils< scalar, index, m, n >, 36
- cils\_LLL\_omp
  - cils::cils\_reduction< scalar, index >, 46
- cils\_LLL\_qr\_omp
  - cils::cils\_reduction< scalar, index >, 46
- cils\_LLL\_qr\_reduction
  - cils::cils\_reduction< scalar, index >, 46
- cils\_LLL\_qr\_serial
  - cils::cils\_reduction< scalar, index >, 46
- cils\_LLL\_reduction
  - cils::cils\_reduction< scalar, index >, 46
- cils\_LLL\_serial
  - cils::cils\_reduction< scalar, index >, 47
- cils\_obils\_matlab
  - cils::cils\_reduction< scalar, index >, 47
- cils\_obils\_reduction
  - cils::cils\_reduction< scalar, index >, 47
- cils\_obils\_reduction\_matlab
  - cils::cils\_reduction< scalar, index >, 47
- cils\_partition\_deficient
  - cils::cils< scalar, index, m, n >, 36
- cils\_qr\_omp
  - cils::cils\_reduction< scalar, index >, 48
- cils\_qr\_py
  - cils::cils\_reduction< scalar, index >, 48
- cils\_qr\_py\_helper
  - cils::cils\_reduction< scalar, index >, 48
- cils\_qr\_serial
  - cils::cils\_reduction< scalar, index >, 48
- cils\_qr\_serial\_col
  - cils::cils\_reduction< scalar, index >, 49
- cils\_qrp\_serial
  - cils::cils< scalar, index, m, n >, 37
- cils\_reduction
  - cils::cils\_reduction< scalar, index >, 45
- cils\_scp\_block\_babai\_omp
  - cils::cils< scalar, index, m, n >, 37

- cils\_scp\_block\_babai\_serial
  - cils::cils< scalar, index, m, n >, 37
- cils\_scp\_block\_optimal\_mpi
  - cils::cils< scalar, index, m, n >, 38
- cils\_scp\_block\_optimal\_omp
  - cils::cils< scalar, index, m, n >, 38
- cils\_scp\_block\_optimal\_serial
  - cils::cils< scalar, index, m, n >, 38
- cils\_scp\_block\_suboptimal\_omp
  - cils::cils< scalar, index, m, n >, 39
- cils\_scp\_block\_suboptimal\_serial
  - cils::cils< scalar, index, m, n >, 39
- cils\_search
  - cils::cils\_search< scalar, index >, 52
- cils\_sic\_serial
  - cils::cils< scalar, index, m, n >, 39
- cils\_sic\_subopt
  - cils::cils< scalar, index, m, n >, 39
- cils\_underdetermined\_test.cpp
  - block\_optimal\_test, 57
  - plot\_run, 58
- coeff
  - cils::program\_def, 11
- compareResults
  - cils\_block\_search\_GPU.cpp, 65
- d\_s
  - cils::program\_def, 9
- display\_matrix
  - helper, 16
- display\_vector
  - helper, 17
- eye
  - helper, 17
- find\_bit\_error\_rate
  - helper, 17
- find\_residual
  - helper, 18
- G
  - cils::cils\_reduction< scalar, index >, 49
- H
  - cils::cils< scalar, index, m, n >, 41
- helper, 15
  - b\_mtimes, 16
  - b\_rand, 16
  - display\_matrix, 16
  - display\_vector, 17
  - eye, 17
  - find\_bit\_error\_rate, 17
  - find\_residual, 18
  - if\_all\_x\_true, 19
  - inv, 19
  - length\_nonzeros, 19
  - mtimes, 20
  - mtimes\_AP, 20
  - mtimes\_Axy, 20
  - mtimes\_col, 21
  - mtimes\_v, 21
  - norm, 21
  - planerot, 22
  - randperm, 22
  - unique\_vector, 23
- helper.h
  - ZERO, 63
- if\_all\_x\_true
  - helper, 19
- ils\_search
  - cils::cils\_search< scalar, index >, 52
- ils\_search\_omp
  - cils::cils\_search< scalar, index >, 52
- index
  - cils::program\_def, 9
- indicator
  - cils::program\_def, 9
- info
  - cils::returnType< scalar, index >, 55
- init
  - cils::cils< scalar, index, m, n >, 40
- init\_guess
  - cils::program\_def, 10
- init\_program\_def
  - cils::program\_def, 10
- init\_R
  - cils::cils< scalar, index, m, n >, 40
- init\_res
  - cils::cils< scalar, index, m, n >, 41
- init\_ud
  - cils::cils< scalar, index, m, n >, 40
- init\_y
  - cils::cils< scalar, index, m, n >, 40
- inv
  - helper, 19
- is\_constrained
  - cils::program\_def, 11
  - plot\_helper, 26
- is\_local
  - cils::program\_def, 11
- is\_matlab
  - cils::program\_def, 11
- is\_nc
  - cils::program\_def, 11
- is\_qr
  - cils::program\_def, 12
- k
  - plot\_helper, 26
- l
  - cils::cils< scalar, index, m, n >, 41
- l\_max
  - plot\_helper, 26
- length\_nonzeros
  - helper, 19

- lower
  - cils::cils< scalar, index, m, n >, [41](#)
- m
  - plot\_helper, [26](#)
- matlabPtr
  - cils::cils< scalar, index, m, n >, [41](#)
- matrix\_vector\_mult
  - cils, [7](#), [8](#)
- max\_iter
  - cils::program\_def, [12](#)
  - plot\_helper, [27](#)
- max\_mem
  - py\_qr, [29](#)
- max\_proc
  - cils::program\_def, [12](#)
  - plot\_helper, [27](#)
- max\_search
  - cils::program\_def, [12](#)
- max\_thre
  - cils::program\_def, [12](#)
- min\_proc
  - cils::program\_def, [12](#)
  - plot\_helper, [27](#)
- mode
  - cils::program\_def, [13](#)
- mtimes
  - helper, [20](#)
- mtimes\_AP
  - helper, [20](#)
- mtimes\_Axy
  - helper, [20](#)
- mtimes\_col
  - helper, [21](#)
- mtimes\_v
  - helper, [21](#)
- n
  - plot\_helper, [27](#)
- norm
  - helper, [21](#)
- NUM\_GLOBAL\_WITEMS
  - cils\_block\_search\_GPU.cpp, [65](#)
- num\_trials
  - cils::program\_def, [13](#)
- obils\_search
  - cils::cils\_search< scalar, index >, [53](#)
- obils\_search2
  - cils::cils\_search< scalar, index >, [53](#)
- obils\_search\_matlab
  - cils::cils\_search< scalar, index >, [53](#)
- obils\_search\_omp
  - cils::cils\_search< scalar, index >, [53](#)
- offset
  - cils::program\_def, [13](#)
- P
  - cils::cils< scalar, index, m, n >, [42](#)
- cils::cils\_reduction< scalar, index >, [49](#)
- p
  - cils::cils\_reduction< scalar, index >, [50](#)
- permutation
  - cils::program\_def, [10](#)
- planerot
  - helper, [22](#)
- plot\_first\_block
  - plot\_helper, [24](#)
- plot\_helper, [23](#)
  - a, [26](#)
  - ber, [26](#)
  - is\_constrained, [26](#)
  - k, [26](#)
  - l\_max, [26](#)
  - m, [26](#)
  - max\_iter, [27](#)
  - max\_proc, [27](#)
  - min\_proc, [27](#)
  - n, [27](#)
  - plot\_first\_block, [24](#)
  - plot\_runtime, [24](#)
  - plot\_runtime\_III, [25](#)
  - plot\_runtime\_ud, [25](#)
  - proc\_num, [27](#)
  - res, [27](#)
  - SNR, [28](#)
  - spu, [28](#)
  - tim, [28](#)
  - title1, [28](#)
  - title3, [28](#)
- plot\_itr
  - cils::program\_def, [13](#)
- plot\_run
  - cils\_underdetermined\_test.cpp, [58](#)
- plot\_runtime
  - plot\_helper, [24](#)
- plot\_runtime\_III
  - plot\_helper, [25](#)
- plot\_runtime\_ud
  - plot\_helper, [25](#)
- proc\_num
  - plot\_helper, [27](#)
- py\_qr, [29](#)
  - block\_mem, [29](#)
  - check\_mem, [29](#)
  - max\_mem, [29](#)
  - total, [29](#)
  - used, [30](#)
  - x, [30](#)
- Q
  - cils::cils\_reduction< scalar, index >, [50](#)
- q
  - cils::program\_def, [13](#)
- qam
  - cils::cils< scalar, index, m, n >, [42](#)
  - cils::program\_def, [13](#)



- R\_A
  - cils::cils< scalar, index, m, n >, [42](#)
- R\_Q
  - cils::cils\_reduction< scalar, index >, [50](#)
- R\_R
  - cils::cils\_reduction< scalar, index >, [50](#)
- randperm
  - helper, [22](#)
- res
  - plot\_helper, [27](#)
- run
  - cils\_block\_search\_GPU.cpp, [65](#)
- run\_time
  - cils::returnType< scalar, index >, [55](#)
- scalar
  - cils::program\_def, [9](#)
- schedule
  - cils::program\_def, [14](#)
- search\_iter
  - cils::program\_def, [14](#)
- sigma
  - cils::cils< scalar, index, m, n >, [42](#)
- SNR
  - cils::program\_def, [14](#)
  - plot\_helper, [28](#)
- snr
  - cils::cils< scalar, index, m, n >, [42](#)
- spilt\_size
  - cils::program\_def, [14](#)
- spu
  - plot\_helper, [28](#)
- src/example/cils\_standard\_test.cpp, [57](#)
- src/example/cils\_underdetermined\_test.cpp, [57](#)
- src/example/plot\_helper.py, [58](#)
- src/example/py\_qr.py, [59](#)
- src/include/cils.h, [59](#)
- src/include/config.h, [60](#)
- src/include/helper.h, [61](#)
- src/source/cils.cpp, [63](#)
- src/source/cils\_babai\_search.cpp, [63](#)
- src/source/cils\_block\_search.cpp, [64](#)
- src/source/cils\_block\_search\_GPU.cpp, [65](#)
- src/source/cils\_ils\_search.cpp, [66](#)
- src/source/cils\_init\_point.cpp, [67](#)
- src/source/cils\_partition\_deficient.cpp, [67](#)
- src/source/cils\_reduction.cpp, [68](#)
- src/source/cils\_sic\_opt.cpp, [68](#)
- stop
  - cils::program\_def, [14](#)
- tim
  - plot\_helper, [28](#)
- timeAddVectorsCPU
  - cils\_block\_search\_GPU.cpp, [66](#)
- title1
  - plot\_helper, [28](#)
- title3
  - plot\_helper, [28](#)
- tolerance
  - cils::cils< scalar, index, m, n >, [42](#)
- total
  - py\_qr, [29](#)
- u
  - cils::cils< scalar, index, m, n >, [43](#)
- ubils\_search
  - cils::cils\_search< scalar, index >, [54](#)
- unique\_vector
  - helper, [23](#)
- upper
  - cils::cils< scalar, index, m, n >, [43](#)
- used
  - py\_qr, [30](#)
- v\_a
  - cils::cils< scalar, index, m, n >, [43](#)
- v\_q
  - cils::cils< scalar, index, m, n >, [43](#)
- verbose
  - cils::program\_def, [14](#)
- warmup
  - cils\_block\_search\_GPU.cpp, [66](#)
- x
  - cils::returnType< scalar, index >, [55](#)
  - py\_qr, [30](#)
- x\_r
  - cils::cils< scalar, index, m, n >, [43](#)
- x\_t
  - cils::cils< scalar, index, m, n >, [43](#)
- y\_a
  - cils::cils< scalar, index, m, n >, [44](#)
  - cils::cils\_reduction< scalar, index >, [50](#)
- y\_q
  - cils::cils\_reduction< scalar, index >, [50](#)
- y\_r
  - cils::cils\_reduction< scalar, index >, [51](#)
- Z
  - cils::cils< scalar, index, m, n >, [44](#)
  - cils::cils\_reduction< scalar, index >, [51](#)
- ZERO
  - helper.h, [63](#)