
ilupp Documentation

Release 1.0.2

Clemens Hofreither

Oct 01, 2021

Contents:

1	Preconditioners for symmetric and positive definite matrices	3
2	Preconditioners for general matrices	5
3	Stand-alone factorization functions	7
4	Solving linear systems	9
5	ILU++ parameters	11
6	Indices and tables	17
	Python Module Index	19
	Index	21

The `ilupp` package provides various incomplete LU and Cholesky factorization routines for sparse matrices. It is implemented in C++ based on Jan Mayer's ILU++ package and comes with convenient Python bindings which use Scipy sparse matrices.

Note: This documentation describes only the Python bindings. The C++ interface is essentially equivalent and should be easy to figure out from the header files.

All preconditioner classes derive from the following base class:

class `ilupp._BaseWrapper` (*dtype, shape*)

Wrapper base class which supports methods and properties common to all preconditioners.

Implements the `scipy.sparse.linalg.LinearOperator` protocol, which means that it has a `.shape` property and can be applied to a vector using `.dot()` or simply the multiplication operator `*`.

To apply the preconditioner to a vector in place, avoiding a copy, use the `apply()` method. To apply the transposed preconditioner, use `apply_trans()` or obtain the transposed operator with `.T`.

apply (*x*)

Apply the preconditioner to the vector *x* in-place.

apply_trans (*x*)

Apply the transposed preconditioner to the vector *x* in-place.

factors ()

Return all matrix factors (usually (L,U) or just (L,)) as a list of sparse matrices.

total_nnz

The total number of nonzeros stored in the factor matrices of the preconditioner.

Preconditioners for symmetric and positive definite matrices

class `ilupp.IChol0Preconditioner` (*A*)

An IChol(0) preconditioner (no fill-in, same sparsity pattern as *A*) for a symmetric positive definite matrix.

Parameters *A* – a symmetric sparse matrix in CSR or CSC format

class `ilupp.ICholTPreconditioner` (*A*, *add_fill_in*=0, *threshold*=0.0)

An incomplete Cholesky preconditioner with user-specifiable additional fill-in and threshold. With *threshold*=0, this is identical to the method described in (Lin, Moré 1999).

Parameters

- *A* – a symmetric sparse matrix in CSR or CSC format
- **add_fill_in** – the number of additional nonzeros to allow per column. By default (0), the factorization keeps the number (but not necessarily the positions) of the nonzeros identical to the original matrix.
- **threshold** – entries with a relative magnitude less than this are dropped. By default (0.0), dropping is only performed based on the number of nonzeros.

Preconditioners for general matrices

class `ilupp.ILU0Preconditioner` (*A*)

An ILU(0) preconditioner (no fill-in, same sparsity pattern as *A*).

Parameters *A* – a sparse matrix in CSR or CSC format

class `ilupp.ILUTPreconditioner` (*A, fill_in=100, threshold=0.1*)

An ILUT (incomplete LU with thresholding) preconditioner.

Parameters

- *A* – a sparse matrix in CSR or CSC format
- *fill_in* – the number of nonzeros to allow per row of L/U
- *threshold* – entries with relative magnitude less than this are dropped

class `ilupp.ILUTPPreconditioner` (*A, fill_in=100, threshold=0.1, piv_tol=0.1, mem_factor=10.0*)

An ILUTP (incomplete LU with thresholding and column pivoting) preconditioner.

Parameters

- *A* – a sparse matrix in CSR or CSC format
- *fill_in* – the number of nonzeros to allow per row of L/U
- *threshold* – entries with relative magnitude less than this are dropped
- *piv_tol* – pivoting tolerance; 0=only pivot when 0 encountered, 1=always pivot to the largest entry, inbetween: pivot depending on relative magnitude

permutations ()

Return a pair (L,R) of permutation arrays to be applied from the left or right due to pivoting.

class `ilupp.ILUCPreconditioner` (*A, fill_in=100, threshold=0.1*)

An ILUC (Crout ILU) preconditioner. Similar to ILUT, but tends to be faster for matrices with symmetric structure. See (Li, Saad, Chow 2003).

Parameters

- *A* – a sparse matrix in CSR or CSC format

- **fill_in** – the number of nonzeros to allow per column/row of L/U
- **threshold** – entries with relative magnitude less than this are dropped

class `ilupp.ILUCPPreconditioner` (*A, fill_in=100, threshold=0.1, piv_tol=0.1, mem_factor=10.0*)

An ILUCP (ILUC with pivoting) preconditioner. See (Mayer 2005).

Parameters

- **A** – a sparse matrix in CSR or CSC format
- **fill_in** – the number of nonzeros to allow per column/row of L/U
- **threshold** – entries with relative magnitude less than this are dropped
- **piv_tol** – pivoting tolerance; 0=only pivot when 0 encountered, 1=always pivot to the largest entry, inbetween: pivot depending on relative magnitude

permutations ()

Return a pair (L,R) of permutation arrays to be applied from the left or right due to pivoting.

class `ilupp.ILUppPreconditioner` (*A, threshold=1.0, fill_in=None, params=None*)

A multilevel ILU++ preconditioner.

Parameters

- **A** – a sparse matrix in CSR or CSC format
- **fill_in** – the fill_in parameter for the ILU++ preconditioner
- **threshold** – the threshold parameter for ILU++; entries with relative magnitude less than this are dropped
- **params** – an instance of `iluplusplus_precond_parameter`; if passed, overrides fill_in and threshold

Stand-alone factorization functions

If you do not want a preconditioner object that you can apply to a vector, you can instead use these factorization functions which directly return the matrix factors of the chosen factorization. The parameters have the same meaning as for the corresponding preconditioner classes above.

Note that if you already have a preconditioner object, you can obtain the matrix factors using the `ilupp._BaseWrapper.factors()` method.

`ilupp.ichol0(A)`

Compute the L factor of an incomplete Cholesky decomposition without fill-in for the symmetric matrix A.

`ilupp.icholt(A, add_fill_in=0, threshold=0.0)`

Compute the L factor of an incomplete Cholesky decomposition with thresholding for the symmetric matrix A.

`ilupp.ilu0(A)`

Compute the (L,U) factors of an incomplete LU decomposition without fill-in.

`ilupp.ilut(A, fill_in=100, threshold=0.1)`

Compute the (L,U) factors of an incomplete LU decomposition with thresholding.

`ilupp.iluc(A, fill_in=100, threshold=0.1)`

Compute the (L,U) factors of an incomplete Crout LU decomposition with thresholding.

Solving linear systems

The `solve()` function provides a convenient interface for setting up a preconditioner and then solving the linear system using a Krylov subspace method.

```
ilupp.solve(A, b, rtol=0.0001, atol=0.0001, max_iter=500, threshold=0.1, fill_in=None, params=None, info=False)
```

Solve the linear system $Ax=b$ using a multilevel ILU++ preconditioner and BiCGStab.

Parameters

- **A** – a sparse matrix in CSR or CSC format
- **b** – the right-hand side vector
- **rtol** – target relative reduction in the residual
- **atol** – target absolute magnitude of the residual
- **max_iter** – maximum number of iterations
- **threshold** – the threshold parameter for ILU++; entries with relative magnitude less than this are dropped
- **fill_in** – the fill_in parameter for the ILU++ preconditioner
- **params** – an instance of `iluplusplus_precond_parameter`; if passed, overrides fill_in and threshold
- **info** – if True, a tuple (nr_of_iterations, achieved_relative_reduction, residual_magnitude) is returned along the solution

Returns a vector containing the solution x

ILU++ parameters

The class `iluplusplus_precond_parameter` provides the ability to tune advanced parameters for the multi-level ILU++ preconditioner. Some comments on these options can be found in [parameters.h](#). Further details are given in the [original publication](#).

The most important members are `iluplusplus_precond_parameter.threshold`, `iluplusplus_precond_parameter.fill_in`, and `iluplusplus_precond_parameter.piv_tol`, which have essentially the same meaning as in the functions above. Various sets of default parameters can be chosen by the `iluplusplus_precond_parameter.default_configuration()` function; see [the readme document](#) for some comments on these.

Furthermore, `iluplusplus_precond_parameter.PREPROCESSING` is an instance of `preprocessing_sequence` (see below) and can be used to choose various methods of reordering the matrix before factorization.

For example:

```
param = ilupp.iluplusplus_precond_parameter()
param.default_configuration(10)
param.PREPROCESSING.set_MAX_WEIGHTED_MATCHING_ORDERING_PQ()
```

will choose default configuration 10 and preprocess the matrix with a max weighted matching ordering followed by a PQ ordering.

class `ilupp.iluplusplus_precond_parameter`

BANDWIDTH_MULTIPLIER

BANDWIDTH_OFFSET

BEGIN_TOTAL_PIV

COMBINE_FACTOR

DROP_TYPE_L

DROP_TYPE_U

EXTERNAL_FINAL_ROW
EXT_MIN_ELIM_FACTOR
FINAL_ROW_CRIT
FINAL_THRESHOLD
GLOBAL_COMMENT
INIT_WEIGHTS_LU
MAX_FILLIN_IS_INF
MAX_LEVELS
MEMORY_MAX_LEVELS
MEM_FACTOR
MIN_ELIM_FACTOR
MIN_ML_SIZE
MIN_PIVOT
MIN_SIZE_ZERO_SCHUR
MIN_WEIGHT
MOVE_LEVEL_FACTOR
MOVE_LEVEL_THRESHOLD
NEUTRAL_ELEMENT
PERMUTE_ROWS
POST_FACT_THRESHOLD
PQ_ALGORITHM
PQ_THRESHOLD
PRECON_PARAMETER
PREPROCESSING
REQUIRE_ZERO_SCHUR
REQ_ZERO_SCHUR_SIZE
ROW_U_MAX
SCALE_WEIGHT_INVDIAG
SCALE_WGT_MAXINVDIAG
SCHUR_COMPLEMENT
SIZE_TABLE_POS_WEIGHTS
SMALL_PIVOT_TERMINATES
SUM_DROPPING
THRESHOLD_SHIFT_SCHUR
THRESHOLD_ZERO_SCHUR
TOTAL_PIV

USE_ERR_PROP_DROPPING
USE_ERR_PROP_DROPPING2
USE_FINAL_THRESHOLD
USE_INVERSE_DROPPING
USE_MAX_AS_MOVE
USE_PIVOT_DROPPING
USE_POS_COMPRESS
USE_STANDARD_DROPPING
USE_STANDARD_DROPPING2
USE_THRES_ZERO_SCHUR
USE_WEIGHTED_DROPPING
USE_WEIGHTED_DROPPING2
VARIABLE_MEM
VARY_THRESHOLD_FACTOR
WEIGHTED_DROPPING
WEIGHT_ERR_PROP_DROP
WEIGHT_ERR_PROP_DROP2
WEIGHT_INVERSE_DROP
WEIGHT_PIVOT_DROP
WEIGHT_STANDARD_DROP
WEIGHT_STANDARD_DROP2
WEIGHT_TABLE_TYPE
WEIGHT_WEIGHTED_DROP
default_configuration (self: ilupp.ilupp.iluplusplus_precond_parameter, arg0: int) → None
fill_in
piv_tol
threshold
use_only_error_propagation_dropping1 (self: ilupp.ilupp.iluplusplus_precond_parameter) → None
use_only_error_propagation_dropping2 (self: ilupp.ilupp.iluplusplus_precond_parameter) → None
use_only_inverse_dropping (self: ilupp.ilupp.iluplusplus_precond_parameter) → None
use_only_pivot_dropping (self: ilupp.ilupp.iluplusplus_precond_parameter) → None
use_only_standard_dropping1 (self: ilupp.ilupp.iluplusplus_precond_parameter) → None
use_only_standard_dropping2 (self: ilupp.ilupp.iluplusplus_precond_parameter) → None
use_only_weighted_dropping1 (self: ilupp.ilupp.iluplusplus_precond_parameter) → None
use_only_weighted_dropping2 (self: ilupp.ilupp.iluplusplus_precond_parameter) → None

```
class ilupp.preprocessing_sequence
```

```
    set_MAX_WEIGHTED_MATCHING_ORDERING (self: ilupp.ilupp.preprocessing_sequence) → None
    set_MAX_WEIGHTED_MATCHING_ORDERING_DD_MOV_COR_IM (self:
                                                         ilupp.ilupp.preprocessing_sequence)
                                                         → None
    set_MAX_WEIGHTED_MATCHING_ORDERING_MOVE_CORNER (self:
                                                         ilupp.ilupp.preprocessing_sequence)
                                                         → None
    set_MAX_WEIGHTED_MATCHING_ORDERING_MOVE_CORNER_IM (self:
                                                         ilupp.ilupp.preprocessing_sequence)
                                                         → None
    set_MAX_WEIGHTED_MATCHING_ORDERING_PQ (self: ilupp.ilupp.preprocessing_sequence) →
                                              None
    set_MAX_WEIGHTED_MATCHING_ORDERING_SP_MOVE_CORNER (self:
                                                         ilupp.ilupp.preprocessing_sequence)
                                                         → None
    set_MAX_WEIGHTED_MATCHING_ORDERING_SP_MOVE_CORNER_IM (self:
                                                            ilupp.ilupp.preprocessing_sequence)
                                                            → None
    set_MAX_WEIGHTED_MATCHING_ORDERING_SYMB_MOVE_CORNER (self:
                                                            ilupp.ilupp.preprocessing_sequence)
                                                            → None
    set_MAX_WEIGHTED_MATCHING_ORDERING_SYMB_MOVE_CORNER_IM (self:
                                                            ilupp.ilupp.preprocessing_sequence)
                                                            → None
    set_MAX_WEIGHTED_MATCHING_ORDERING_SYM_PQ (self: ilupp.ilupp.preprocessing_sequence)
                                                  → None
    set_MAX_WEIGHTED_MATCHING_ORDERING_UNIT_DIAG (self: ilupp.ilupp.preprocessing_sequence)
                                                    → None
    set_MAX_WEIGHTED_MATCHING_ORDERING_UNIT_DIAG_DD_MOV_COR_IM (self:
                                                                    ilupp.ilupp.preprocessing_sequence)
                                                                    → None
    set_MAX_WEIGHTED_MATCHING_ORDERING_WGT2_MOV_COR (self:
                                                         ilupp.ilupp.preprocessing_sequence)
                                                         → None
    set_MAX_WEIGHTED_MATCHING_ORDERING_WGT2_MOV_COR_IM (self:
                                                            ilupp.ilupp.preprocessing_sequence)
                                                            → None
    set_MAX_WEIGHTED_MATCHING_ORDERING_WGT_MOV_COR (self:
                                                         ilupp.ilupp.preprocessing_sequence)
                                                         → None
    set_MAX_WEIGHTED_MATCHING_ORDERING_WGT_MOV_COR_IM (self:
                                                            ilupp.ilupp.preprocessing_sequence)
                                                            → None
    set_NORM_MAX_WEIGHTED_MATCHING_ORDERING (self: ilupp.ilupp.preprocessing_sequence)
                                              → None
    set_NORM_MAX_WEIGHTED_MATCHING_ORDERING_DD_MOV_COR_IM (self:
                                                            ilupp.ilupp.preprocessing_sequence)
                                                            → None
```

```

set_NORM_MAX_WEIGHTED_MATCHING_ORDERING_MOVE_CORNER (self:
                                                    ilupp._ilupp.preprocessing_sequence)
                                                    → None
set_NORM_MAX_WEIGHTED_MATCHING_ORDERING_MOVE_CORNER_IM (self:
                                                         ilupp._ilupp.preprocessing_sequence)
                                                         → None
set_NORM_MAX_WEIGHTED_MATCHING_ORDERING_PQ (self: ilupp._ilupp.preprocessing_sequence)
                                                    → None
set_NORM_MAX_WEIGHTED_MATCHING_ORDERING_SP_MOVE_CORNER (self:
                                                         ilupp._ilupp.preprocessing_sequence)
                                                         → None
set_NORM_MAX_WEIGHTED_MATCHING_ORDERING_SP_MOVE_CORNER_IM (self:
                                                            ilupp._ilupp.preprocessing_sequence)
                                                            → None
set_NORM_MAX_WEIGHTED_MATCHING_ORDERING_SYMB_MOVE_CORNER (self:
                                                            ilupp._ilupp.preprocessing_sequence)
                                                            → None
set_NORM_MAX_WEIGHTED_MATCHING_ORDERING_SYMB_MOVE_CORNER_IM (self:
                                                                ilupp._ilupp.preprocessing_sequence)
                                                                → None
set_NORM_MAX_WEIGHTED_MATCHING_ORDERING_SYM_PQ (self:
                                                  ilupp._ilupp.preprocessing_sequence)
                                                  → None
set_NORM_MAX_WEIGHTED_MATCHING_ORDERING_WGT2_MOV_COR (self:
                                                         ilupp._ilupp.preprocessing_sequence)
                                                         → None
set_NORM_MAX_WEIGHTED_MATCHING_ORDERING_WGT2_MOV_COR_IM (self:
                                                            ilupp._ilupp.preprocessing_sequence)
                                                            → None
set_NORM_MAX_WEIGHTED_MATCHING_ORDERING_WGT_MOV_COR (self:
                                                         ilupp._ilupp.preprocessing_sequence)
                                                         → None
set_NORM_MAX_WEIGHTED_MATCHING_ORDERING_WGT_MOV_COR_IM (self:
                                                            ilupp._ilupp.preprocessing_sequence)
                                                            → None
set_PQ (self: ilupp._ilupp.preprocessing_sequence) → None
set_SPARSE_FIRST (self: ilupp._ilupp.preprocessing_sequence) → None
set_SPARSE_FIRST_MAX_WEIGHTED_MATCHING_ORDERING (self:
                                                    ilupp._ilupp.preprocessing_sequence)
                                                    → None
set_SPARSE_FIRST_MAX_WEIGHTED_MATCHING_ORDERING_DD_MOV_COR_IM (self:
                                                                    ilupp._ilupp.preprocessing_sequence)
                                                                    → None
set_SPARSE_FIRST_MAX_WEIGHTED_MATCHING_ORDERING_UNIT_DIAG (self:
                                                             ilupp._ilupp.preprocessing_sequence)
                                                             → None
set_SPARSE_FIRST_MAX_WEIGHTED_MATCHING_ORDERING_UNIT_DIAG_DD_MOV_COR_IM (self:
                                                                                ilupp._ilupp.preprocessing
                                                                                →
                                                                                None

```

set_none (*self*: *ilupp._ilupp.preprocessing_sequence*) → None

set_normalize (*self*: *ilupp._ilupp.preprocessing_sequence*) → None

to_names (*self*: *ilupp._ilupp.preprocessing_sequence*) → list

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`

i

ilupp, [1](#)

Symbols

`_BaseWrapper` (class in `ilupp`), 1

A

`apply()` (`ilupp._BaseWrapper` method), 1

`apply_trans()` (`ilupp._BaseWrapper` method), 1

B

`BANDWIDTH_MULTIPLIER`
(`ilupp.iluplusplus_precond_parameter` attribute), 11

`BANDWIDTH_OFFSET` (`ilupp.iluplusplus_precond_parameter` attribute), 11

`BEGIN_TOTAL_PIV` (`ilupp.iluplusplus_precond_parameter` attribute), 11

C

`COMBINE_FACTOR` (`ilupp.iluplusplus_precond_parameter` attribute), 11

D

`default_configuration()`
(`ilupp.iluplusplus_precond_parameter` method), 13

`DROP_TYPE_L` (`ilupp.iluplusplus_precond_parameter` attribute), 11

`DROP_TYPE_U` (`ilupp.iluplusplus_precond_parameter` attribute), 11

E

`EXT_MIN_ELIM_FACTOR`
(`ilupp.iluplusplus_precond_parameter` attribute), 12

`EXTERNAL_FINAL_ROW`
(`ilupp.iluplusplus_precond_parameter` attribute), 11

F

`factors()` (`ilupp._BaseWrapper` method), 1

`fill_in` (`ilupp.iluplusplus_precond_parameter` attribute), 13

`FINAL_ROW_CRIT` (`ilupp.iluplusplus_precond_parameter` attribute), 12

`FINAL_THRESHOLD` (`ilupp.iluplusplus_precond_parameter` attribute), 12

G

`GLOBAL_COMMENT` (`ilupp.iluplusplus_precond_parameter` attribute), 12

I

`ichol0()` (in module `ilupp`), 7

`IChol0Preconditioner` (class in `ilupp`), 3

`icholt()` (in module `ilupp`), 7

`ICholtPreconditioner` (class in `ilupp`), 3

`ilu0()` (in module `ilupp`), 7

`ILU0Preconditioner` (class in `ilupp`), 5

`iluc()` (in module `ilupp`), 7

`ILUCPPreconditioner` (class in `ilupp`), 6

`ILUCPreconditioner` (class in `ilupp`), 5

`iluplusplus_precond_parameter` (class in `ilupp`), 11

`ilupp` (module), 1

`ILUpPreconditioner` (class in `ilupp`), 6

`ilut()` (in module `ilupp`), 7

`ILUTPPreconditioner` (class in `ilupp`), 5

`ILUTPreconditioner` (class in `ilupp`), 5

`INIT_WEIGHTS_LU` (`ilupp.iluplusplus_precond_parameter` attribute), 12

M

`MAX_FILLIN_IS_INF`
(`ilupp.iluplusplus_precond_parameter` attribute), 12

`MAX_LEVELS` (`ilupp.iluplusplus_precond_parameter` attribute), 12

`MEM_FACTOR` (`ilupp.iluplusplus_precond_parameter` attribute), 12

MEMORY_MAX_LEVELS
(*ilupp.iluplusplus_precond_parameter* attribute), 12

MIN_ELIM_FACTOR (*ilupp.iluplusplus_precond_parameter* attribute), 12

MIN_ML_SIZE (*ilupp.iluplusplus_precond_parameter* attribute), 12

MIN_PIVOT (*ilupp.iluplusplus_precond_parameter* attribute), 12

MIN_SIZE_ZERO_SCHUR
(*ilupp.iluplusplus_precond_parameter* attribute), 12

MIN_WEIGHT (*ilupp.iluplusplus_precond_parameter* attribute), 12

MOVE_LEVEL_FACTOR
(*ilupp.iluplusplus_precond_parameter* attribute), 12

MOVE_LEVEL_THRESHOLD
(*ilupp.iluplusplus_precond_parameter* attribute), 12

N

NEUTRAL_ELEMENT (*ilupp.iluplusplus_precond_parameter* attribute), 12

P

permutations() (*ilupp.ILUCPPreconditioner* method), 6

permutations() (*ilupp.ILUTPPreconditioner* method), 5

PERMUTE_ROWS (*ilupp.iluplusplus_precond_parameter* attribute), 12

piv_tol (*ilupp.iluplusplus_precond_parameter* attribute), 13

POST_FACT_THRESHOLD
(*ilupp.iluplusplus_precond_parameter* attribute), 12

PQ_ALGORITHM (*ilupp.iluplusplus_precond_parameter* attribute), 12

PQ_THRESHOLD (*ilupp.iluplusplus_precond_parameter* attribute), 12

PRECON_PARAMETER (*ilupp.iluplusplus_precond_parameter* attribute), 12

PREPROCESSING (*ilupp.iluplusplus_precond_parameter* attribute), 12

preprocessing_sequence (class in *ilupp*), 13

R

REQ_ZERO_SCHUR_SIZE
(*ilupp.iluplusplus_precond_parameter* attribute), 12

REQUIRE_ZERO_SCHUR
(*ilupp.iluplusplus_precond_parameter* attribute), 12

ROW_U_MAX (*ilupp.iluplusplus_precond_parameter* attribute), 12

S

SCALE_WEIGHT_INVDIAG
(*ilupp.iluplusplus_precond_parameter* attribute), 12

SCALE_WGT_MAXINVDIAG
(*ilupp.iluplusplus_precond_parameter* attribute), 12

SCHUR_COMPLEMENT (*ilupp.iluplusplus_precond_parameter* attribute), 12

set_MAX_WEIGHTED_MATCHING_ORDERING()
(*ilupp.preprocessing_sequence* method), 14

set_MAX_WEIGHTED_MATCHING_ORDERING_DD_MOV_COR_IM()
(*ilupp.preprocessing_sequence* method), 14

set_MAX_WEIGHTED_MATCHING_ORDERING_MOVE_CORNER()
(*ilupp.preprocessing_sequence* method), 14

set_MAX_WEIGHTED_MATCHING_ORDERING_MOVE_CORNER_IM()
(*ilupp.preprocessing_sequence* method), 14

set_MAX_WEIGHTED_MATCHING_ORDERING_PQ()
(*ilupp.preprocessing_sequence* method), 14

set_MAX_WEIGHTED_MATCHING_ORDERING_SP_MOVE_CORNER()
(*ilupp.preprocessing_sequence* method), 14

set_MAX_WEIGHTED_MATCHING_ORDERING_SP_MOVE_CORNER_I()
(*ilupp.preprocessing_sequence* method), 14

set_MAX_WEIGHTED_MATCHING_ORDERING_SYM_PQ()
(*ilupp.preprocessing_sequence* method), 14

set_MAX_WEIGHTED_MATCHING_ORDERING_SYMB_MOVE_CORNER()
(*ilupp.preprocessing_sequence* method), 14

set_MAX_WEIGHTED_MATCHING_ORDERING_SYMB_MOVE_CORNER_I()
(*ilupp.preprocessing_sequence* method), 14

set_MAX_WEIGHTED_MATCHING_ORDERING_UNIT_DIAG()
(*ilupp.preprocessing_sequence* method), 14

set_MAX_WEIGHTED_MATCHING_ORDERING_UNIT_DIAG_DD_MOV()
(*ilupp.preprocessing_sequence* method), 14

set_MAX_WEIGHTED_MATCHING_ORDERING_WGT2_MOV_COR()
(*ilupp.preprocessing_sequence* method), 14

set_MAX_WEIGHTED_MATCHING_ORDERING_WGT2_MOV_COR_IM()
(*ilupp.preprocessing_sequence* method), 14

set_MAX_WEIGHTED_MATCHING_ORDERING_WGT_MOV_COR()
(*ilupp.preprocessing_sequence* method), 14

set_MAX_WEIGHTED_MATCHING_ORDERING_WGT_MOV_COR_IM()
(*ilupp.preprocessing_sequence* method), 14

set_none() (*ilupp.preprocessing_sequence* method), 15

set_NORM_MAX_WEIGHTED_MATCHING_ORDERING()
(*ilupp.preprocessing_sequence* method), 14

set_NORM_MAX_WEIGHTED_MATCHING_ORDERING_DD_MOV_COR()
(*ilupp.preprocessing_sequence* method), 14

set_NORM_MAX_WEIGHTED_MATCHING_ORDERING_MOVE_CORNER()
(*ilupp.preprocessing_sequence* method), 14

set_NORM_MAX_WEIGHTED_MATCHING_ORDERING_MOVE_CORNER_I()
(*ilupp.preprocessing_sequence* method), 15

*(ilupp.iluplusplus_precond_parameter
attribute), 13*
USE_THRES_ZERO_SCHUR
*(ilupp.iluplusplus_precond_parameter
attribute), 13*
USE_WEIGHTED_DROPPING
*(ilupp.iluplusplus_precond_parameter
attribute), 13*
USE_WEIGHTED_DROPPING2
*(ilupp.iluplusplus_precond_parameter
attribute), 13*

V

VARIABLE_MEM *(ilupp.iluplusplus_precond_parameter
attribute), 13*
VARY_THRESHOLD_FACTOR
*(ilupp.iluplusplus_precond_parameter
attribute), 13*

W

WEIGHT_ERR_PROP_DROP
*(ilupp.iluplusplus_precond_parameter
attribute), 13*
WEIGHT_ERR_PROP_DROP2
*(ilupp.iluplusplus_precond_parameter
attribute), 13*
WEIGHT_INVERSE_DROP
*(ilupp.iluplusplus_precond_parameter
attribute), 13*
WEIGHT_PIVOT_DROP
*(ilupp.iluplusplus_precond_parameter
attribute), 13*
WEIGHT_STANDARD_DROP
*(ilupp.iluplusplus_precond_parameter
attribute), 13*
WEIGHT_STANDARD_DROP2
*(ilupp.iluplusplus_precond_parameter
attribute), 13*
WEIGHT_TABLE_TYPE
*(ilupp.iluplusplus_precond_parameter
attribute), 13*
WEIGHT_WEIGHTED_DROP
*(ilupp.iluplusplus_precond_parameter
attribute), 13*
WEIGHTED_DROPPING
*(ilupp.iluplusplus_precond_parameter
attribute), 13*