# SparseMatrix

Generated by Doxygen 1.8.11

Tue Oct 6 2015 00:07:54

# Contents

# Chapter 1

# Hierarchical Index

## 1.1 Class Hierarchy

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

# Chapter 2

# Class Index

## 2.1 Class List

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

# Chapter 3

# Class Documentation

## 3.1 CSIR< T > Class Template Reference

CSIR - Compressed Sparse (lower triangle) Row.

`#include <csir.h>`

### Public Member Functions

- CSIR (T **mtrx, T eval, int size)

  *Creates an instance of CSIR sparse matrix.*
- ~CSIR ()

  *Deletes an instance of CSR sparse matrix.*
- CSIR (CSIR &other)

  *Copies the data of CSIR sparse matrix from other CSIR matrix.*
- CSIR & operator= (CSIR &other)

  *Assignes an instance of CSIR sparse matrix with other CSIR matrix.*
- T * adiag () const

  *Gets adiag - an array of diagonal elements.*
- T * altr () const

  *Gets altr - an array of nonempty elements of lower triangular matrix.*
- T * autr () const

  *Gets autr - an array of nonempty elements of upper triangular matrix.*
- int * iptr () const

  *Gets the iptr - an array of position in which the corresponding rows appear in altr for the first time.*
- int * jptr () const

  *Gets column-indices of the corresponding altr elements.*
- int size () const

  *Gets size of matrix (number of rows)*
- int size_of_altr () const

  *Gets size of altr - number of nonempty elements in lower triangular matrix.*
- std::vector< T > operator* (std::vector< T > &vec)

  *Multiplies CSIR matrix by vector.*

### 3.1.1 Detailed Description

**template<typename T>**
**class CSIR< T >**

CSIR - Compressed Sparse (lower triangle) Row.

Sparse matrix format for asymmetric matrices with symmetric portraits. A.k.a. Skyline format.

Matrix is stored in following vectors:

- adiag - diagonal elements
- altr - nonempty elements of lower triangular matrix
- autr - nonempty elements of upper triangular matrix
- iptr - i-th element holds the position in which the i-th row appears in altr for the first time
- jptr - column-indices of the corresponding altr elements

**Template Parameters**

| | |
|---|---|
| *T* | - Type of data stored in matrix. |

### 3.1.2 Constructor & Destructor Documentation

#### 3.1.2.1 template<typename T > CSIR< T >::CSIR ( T ** *mtrx*, T *eval*, int *size* ) [inline]

Creates an instance of CSIR sparse matrix.

Parses a given plain symmetricmatrix into a CSIR format. Note that all the diagonal elements are treated as nonempty.

**Parameters**

| | |
|---|---|
| *mtrx* | Plain symmetric matrix |
| *eval* | Empty value |
| *size* | Size of matrix (number of rows) |

#### 3.1.2.2 template<typename T > CSIR< T >::CSIR ( CSIR< T > & *other* ) [inline]

Copies the data of CSIR sparse matrix from other CSIR matrix.

**Parameters**

| | |
|---|---|
| *other* | Reference to other CSIR matrix |

### 3.1.3 Member Function Documentation

#### 3.1.3.1 template<typename T > T* CSIR< T >::adiag ( ) const [inline]

Gets adiag - an array of diagonal elements.

**Returns**

adiag array

#### 3.1.3.2 template<typename T > T* CSIR< T >::altr ( ) const [inline]

Gets altr - an array of nonempty elements of lower triangular matrix.

**Returns**

altr array

#### 3.1.3.3 template<typename T > T* CSIR< T >::autr ( ) const [inline]

Gets autr - an array of nonempty elements of upper triangular matrix.

**Returns**

autr array

#### 3.1.3.4 template<typename T > int* CSIR< T >::iptr ( ) const [inline]

Gets the iptr - an array of position in which the corresponding rows appear in altr for the first time.

**Returns**

iptr array

#### 3.1.3.5 template<typename T > int* CSIR< T >::jptr ( ) const [inline]

Gets column-indices of the corresponding altr elements.

**Returns**

jptr array

#### 3.1.3.6 template<typename T > std::vector<T> CSIR< T >::operator* ( std::vector< T > & *vec* ) [inline]

Multiplies CSIR matrix by vector.

Note that for large sparse matrices this multiplication will be extremely efficient.

**Parameters**

| | |
|---|---|
| *vec* | Given vector |

**Returns**

Result of multiplication

#### 3.1.3.7 template<typename T > CSIR& CSIR< T >::operator= ( CSIR< T > & *other* ) [inline]

Assignes an instance of CSIR sparse matrix with other CSIR matrix.

Copies all the data from other matrix

**Parameters**

| | |
|---|---|
| *other* | Reference to other CSIR matrix |

#### 3.1.3.8 template<typename T > int CSIR< T >::size ( ) const [inline]

Gets size of matrix (number of rows)

**Returns**

Size of matrix

#### 3.1.3.9 template<typename T > int CSIR< T >::size_of_altr ( ) const [inline]

Gets size of altr - number of nonempty elements in lower triangular matrix.

**Returns**

Size of altr

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

- src/csir.h

## 3.2 CSR< T > Class Template Reference

CSR - Compressed Sparse Row.

`#include <csr.h>`

### Public Member Functions

- CSR (T **mtrx, T eval, int rows, int cols=0)
  *Creates an instance of CSR sparse matrix.*
- ~CSR ()
  *Deletes an instance of CSR sparse matrix.*
- CSR (CSR &other)
  *Copies the data of CSR sparse matrix from other CSR matrix.*
- CSR & operator= (CSR &other)
  *Assignes an instance of CSR sparse matrix with other CSR matrix.*
- T * aelem () const
  *Gets aelem - an array of nonempty elements of matrix.*
- int * iptr () const
  *Gets the iptr - an array of position in which the corresponding rows appear in aelem for the first time.*
- int * jptr () const
  *Gets column-indices of the corresponding aelem elements.*

- int rows () const

  *Gets the number of rows in matrix.*
- int cols () const

  *Gets the number of columns in matrix.*
- int size_of_aelem () const

  *Gets size of aelem - number of nonempty elements in matrix.*
- std::vector< T > operator* (std::vector< T > &vec)

  *Multiplies CSR matrix by vector.*

### 3.2.1 Detailed Description

**template<typename T>**
**class CSR< T >**

CSR - Compressed Sparse Row.

Sparse matrix format for asymmetric matrices with asymmetric portraits.

Matrix is stored in following vectors:

- aelem - nonempty elements of matrix
- iptr - i-th element holds the position in which the i-th row appears in aelem for the first time
- jptr - column-indices of the corresponding aelem elements

**Template Parameters**

| | |
|---|---|
| *T* | Type of data stored in matrix. |

### 3.2.2 Constructor & Destructor Documentation

#### 3.2.2.1 template<typename T > CSR< T >::CSR ( T ** *mtrx*, T *eval*, int *rows*, int *cols* = 0 ) [inline]

Creates an instance of CSR sparse matrix.

Parses a given plain matrix into a CSR format

**Parameters**

| | |
|---|---|
| *mtrx* | Plain matrix |
| *eval* | Empty value |
| *rows* | Number of rows in matrix |
| *cols* | Number of columns in matrix |

#### 3.2.2.2 template<typename T > CSR< T >::CSR ( CSR< T > & *other* ) [inline]

Copies the data of CSR sparse matrix from other CSR matrix.

**Parameters**

| | |
|---|---|
| *other* | Reference to other CSR matrix |

### 3.2.3 Member Function Documentation

#### 3.2.3.1 template<typename T > T* CSR< T >::aelem ( ) const [inline]

Gets aelem - an array of nonempty elements of matrix.

**Returns**

aelem array

#### 3.2.3.2 template<typename T > int CSR< T >::cols ( ) const [inline]

Gets the number of columns in matrix.

**Returns**

Number of columns

#### 3.2.3.3 template<typename T > int* CSR< T >::iptr ( ) const [inline]

Gets the iptr - an array of position in which the corresponding rows appear in aelem for the first time.

**Returns**

iptr array

#### 3.2.3.4 template<typename T > int* CSR< T >::jptr ( ) const [inline]

Gets column-indices of the corresponding aelem elements.

**Returns**

jptr array

#### 3.2.3.5 template<typename T > std::vector<T> CSR< T >::operator* ( std::vector< T > & *vec* ) [inline]

Multiplies CSR matrix by vector.

Note that for large sparse matrices this multiplication will be extremely efficient.

**Parameters**

| | |
|---|---|
| *vec* | Given vector |

**Returns**

Result of multiplication

#### 3.2.3.6 template<typename T > CSR& CSR< T >::operator= ( CSR< T > & *other* ) [inline]

Assignes an instance of CSR sparse matrix with other CSR matrix.

Copies all the data from other matrix

**Parameters**

| | |
|---|---|
| *other* | Reference to other CSR matrix |

#### 3.2.3.7 template<typename T > int CSR< T >::rows ( ) const [inline]

Gets the number of rows in matrix.

**Returns**

Number of rows

#### 3.2.3.8 template<typename T > int CSR< T >::size_of_aelem ( ) const [inline]

Gets size of aelem - number of nonempty elements in matrix.

**Returns**

Size of aelem

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

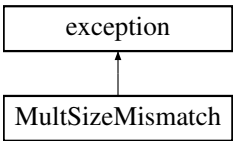
- src/csr.h

## 3.3 MultSizeMismatch Class Reference

Exception that is thrown when trying to multiply the CSR or CSIR matrix on vector of the invalid size.

`#include <exception.h>`

Inheritance diagram for MultSizeMismatch:




### Public Member Functions

- const char * **what** () const throw ()

### 3.3.1 Detailed Description

Exception that is thrown when trying to multiply the CSR or CSIR matrix on vector of the invalid size.

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

- src/exception.h