# Linear Algebra Library

1.0

Generated by Doxygen 1.8.18

**1 Namespace Index** 1

1.1 Namespace List . . . 1

**2 Class Index** 3

2.1 Class List . . . 3

**3 File Index** 5

3.1 File List . . . 5

**4 Namespace Documentation** 7

4.1 linalg Namespace Reference . . . 7

4.1.1 Function Documentation . . . 7

4.1.1.1 operator∗() . . . 7

**5 Class Documentation** 9

5.1 linalg::matrix< T > Class Template Reference . . . 9

5.1.1 Detailed Description . . . 10

5.1.2 Constructor & Destructor Documentation . . . 10

5.1.2.1 matrix() [1/2] . . . 10

5.1.2.2 matrix() [2/2] . . . 10

5.1.3 Member Function Documentation . . . 10

5.1.3.1 at() . . . 10

5.1.3.2 check_range() . . . 11

5.1.3.3 multiply() [1/2] . . . 11

5.1.3.4 multiply() [2/2] . . . 11

5.1.3.5 no_of_cols() . . . 12

5.1.3.6 no_of_rows() . . . 12

5.1.3.7 operator∗() [1/2] . . . 12

5.1.3.8 operator∗() [2/2] . . . 12

5.1.3.9 operator=() . . . 13

5.1.3.10 print() . . . 13

5.1.3.11 transpose() . . . 13

5.1.4 Friends And Related Function Documentation . . . 13

5.1.4.1 operator∗ . . . 13

5.1.5 Member Data Documentation . . . 14

5.1.5.1 cols . . . 14

5.1.5.2 data . . . 14

5.1.5.3 rows . . . 14

**6 File Documentation** 15

6.1 src/example.cpp File Reference . . . 15

6.1.1 Function Documentation . . . 15

6.1.1.1 main() . . . 15

6.2 src/matrix.h File Reference . . . 16

6.2.1 Detailed Description . . . . . 17

**Index** **19**

# Chapter 1

# Namespace Index

## 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

linalg . . . . . 7

# Chapter 2

# Class Index

## 2.1 Class List

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

linalg::matrix< T >
Class for Matrix multiplication and transpose . . . . . 9

# Chapter 3

# File Index

## 3.1 File List

Here is a list of all files with brief descriptions:

src/example.cpp . . . . . 15
src/matrix.h
Linear Algebra library . . . . . 16

# Chapter 4

# Namespace Documentation

## 4.1 linalg Namespace Reference

### Classes

- class matrix

  *Class for Matrix multiplication and transpose.*

### Functions

- template<typename T >
  matrix< T > operator* (T scalar, matrix< T > mat)

### 4.1.1 Function Documentation

#### 4.1.1.1 operator*()

```
template<typename T >
matrix<T> linalg::operator* (
            T scalar,
            matrix< T > mat )
```

# Chapter 5

# Class Documentation

## 5.1 linalg::matrix< T > Class Template Reference

Class for Matrix multiplication and transpose.

`#include <matrix.h>`

### Public Member Functions

- matrix (size_t n_rows, size_t n_cols)
- matrix (size_t n_rows, size_t n_cols, T *input_data)
- T & at (size_t i, size_t j)

  *Access Matrix Elements.*
- matrix< T > transpose ()

  *Returns Transpose of matrix.*
- matrix< T > operator* (const T scalar)

  *Matrix and Scalar Multiplication.*
- matrix< T > operator* (matrix< T > mat)

  *Matrix and Matrix Multiplication.*
- matrix< T > multiply (const T scalar)

  *Matrix and Scalar Multiplication.*
- matrix< T > multiply (matrix< T > mat)

  *Matrix and Matrix Multiplication.*
- matrix< T > & operator= (const matrix< T > &m)

  *Assignment operator overloaded to do deep copy.*
- void check_range (size_t i, size_t j)

  *Check Indices are withing bounds.*
- size_t no_of_rows ()

  *Returns number of rows.*
- size_t no_of_cols ()

  *Returns number of columns.*
- void print ()

  *Prints all elements of this matrix.*

### Private Attributes

- boost::shared_ptr< T[ ]> data

  *Matrix elements as a array.*
- size_t rows

  *Number of Rows.*
- size_t cols

  *Number of Columns.*

### Friends

- template<T >
  matrix< T > operator* (T scalar, const matrix< T > &mat)

  *Matrix and Scalar Multiplication.*

## 5.1.1 Detailed Description

**template<typename T>**
**class linalg::matrix< T >**

Class for Matrix multiplication and transpose.

## 5.1.2 Constructor & Destructor Documentation

#### 5.1.2.1 matrix() [1/2]

```
template<typename T >
linalg::matrix< T >::matrix (
            size_t n_rows,
            size_t n_cols )
```

#### 5.1.2.2 matrix() [2/2]

```
template<typename T >
linalg::matrix< T >::matrix (
            size_t n_rows,
            size_t n_cols,
            T * input_data )
```

## 5.1.3 Member Function Documentation

#### 5.1.3.1 at()

```
template<typename T >
T & linalg::matrix< T >::at (
            size_t i,
            size_t j )
```

Access Matrix Elements.

**Parameters**

| | |
|---|---|
| *i* | Row Index. |
| *j* | Column Index. |

#### 5.1.3.2 check_range()

```
template<typename T >
void linalg::matrix< T >::check_range (
            size_t i,
            size_t j )
```

Check Indices are withing bounds.

**Parameters**

| | |
|---|---|
| *i* | Row Index. |
| *k* | Column Index. |

#### 5.1.3.3 multiply() [1/2]

```
template<typename T >
matrix< T > linalg::matrix< T >::multiply (
            const T scalar )
```

Matrix and Scalar Multiplication.

**Parameters**

| | |
|---|---|
| *scalar* | Scalar value to be multiplied with the matrix. |

#### 5.1.3.4 multiply() [2/2]

```
template<typename T >
matrix< T > linalg::matrix< T >::multiply (
            matrix< T > mat )
```

Matrix and Matrix Multiplication.

**Parameters**

| | |
|---|---|
| *mat* | Matrix to be multiplied. |

#### 5.1.3.5 no_of_cols()

```
template<typename T >
size_t linalg::matrix< T >::no_of_cols
```

Returns number of columns.

#### 5.1.3.6 no_of_rows()

```
template<typename T >
size_t linalg::matrix< T >::no_of_rows
```

Returns number of rows.

#### 5.1.3.7 operator*() [1/2]

```
template<typename T >
matrix< T > linalg::matrix< T >::operator* (
            const T scalar )
```

Matrix and Scalar Multiplication.

**Parameters**

| | |
|---|---|
| *scalar* | Scalar value to be multiplied with the matrix. |

#### 5.1.3.8 operator*() [2/2]

```
template<typename T >
matrix< T > linalg::matrix< T >::operator* (
            matrix< T > mat )
```

Matrix and Matrix Multiplication.

**Parameters**

| | |
|---|---|
| *mat* | Matrix to be multiplied. |

#### 5.1.3.9 operator=()

```
template<typename T >
matrix< T > & linalg::matrix< T >::operator= (
            const matrix< T > & m )
```

Assignment operator overloaded to do deep copy.

#### 5.1.3.10 print()

```
template<typename T >
void linalg::matrix< T >::print
```

Prints all elements of this matrix.

#### 5.1.3.11 transpose()

```
template<typename T >
matrix< T > linalg::matrix< T >::transpose
```

Returns Transpose of matrix.

### 5.1.4 Friends And Related Function Documentation

#### 5.1.4.1 operator*

```
template<typename T >
template<T >
matrix<T> operator* (
            T scalar,
            const matrix< T > & mat )  [friend]
```

Matrix and Scalar Multiplication.

**Parameters**

| | |
|---|---|
| *mat* | Matrix to be multiplied. |
| *scalar* | Scalar value to be multiplied with the matrix. |

### 5.1.5 Member Data Documentation

#### 5.1.5.1 cols

```
template<typename T >
size_t linalg::matrix< T >::cols  [private]
```

Number of Columns.

#### 5.1.5.2 data

```
template<typename T >
boost::shared_ptr<T[]> linalg::matrix< T >::data  [private]
```

Matrix elements as a array.

#### 5.1.5.3 rows

```
template<typename T >
size_t linalg::matrix< T >::rows  [private]
```

Number of Rows.

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

- src/matrix.h

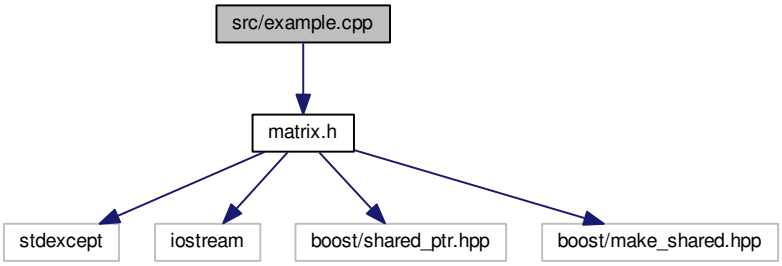
# Chapter 6

# File Documentation

## 6.1 src/example.cpp File Reference

```
#include "matrix.h"
```

Include dependency graph for example.cpp:




### Functions

- int main ()

### 6.1.1 Function Documentation

#### 6.1.1.1 main()

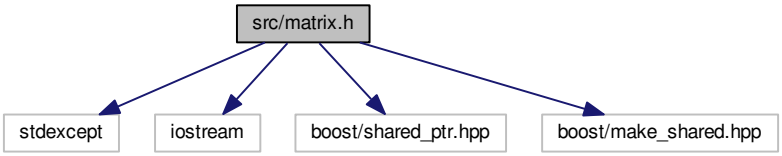
```
int main ( )
```

## 6.2 src/matrix.h File Reference

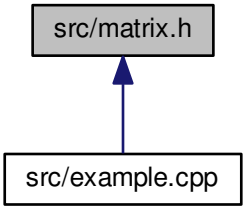
Linear Algebra library.

```
#include <stdexcept>
#include <iostream>
#include <boost/shared_ptr.hpp>
#include <boost/make_shared.hpp>
```

Include dependency graph for matrix.h:




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




### Classes

- class linalg::matrix< T >

  *Class for Matrix multiplication and transpose.*

### Namespaces

- linalg

### Functions

- template<typename T >
  matrix< T > linalg::operator* (T scalar, matrix< T > mat)

### 6.2.1 Detailed Description

Linear Algebra library.

**Author**

Veera Ragav

# Index


at
    linalg::matrix< T >, 10

check_range
    linalg::matrix< T >, 11
cols
    linalg::matrix< T >, 14

data
    linalg::matrix< T >, 14

example.cpp
    main, 15

linalg, 7
    operator*, 7
linalg::matrix< T >, 9
    at, 10
    check_range, 11
    cols, 14
    data, 14
    matrix, 10
    multiply, 11
    no_of_cols, 12
    no_of_rows, 12
    operator*, 12, 13
    operator=, 12
    print, 13
    rows, 14
    transpose, 13

main
    example.cpp, 15
matrix
    linalg::matrix< T >, 10
multiply
    linalg::matrix< T >, 11

no_of_cols
    linalg::matrix< T >, 12
no_of_rows
    linalg::matrix< T >, 12

operator*
    linalg, 7
    linalg::matrix< T >, 12, 13
operator=
    linalg::matrix< T >, 12

print
    linalg::matrix< T >, 13

rows
    linalg::matrix< T >, 14

src/example.cpp, 15
src/matrix.h, 16

transpose
    linalg::matrix< T >, 13