MatrixLib

Generated by Doxygen 1.9.1

1 Class Index 1

1.1 Class List . . . 1

2 Class Documentation 3

2.1 MatrixLib::Matrix< _Scalar, _RowCount, _ColCount > Class Template Reference . . . 3

2.1.1 Detailed Description . . . 4

2.1.2 Constructor & Destructor Documentation . . . 4

2.1.2.1 Matrix() [1/4] . . . 4

2.1.2.2 Matrix() [2/4] . . . 5

2.1.2.3 Matrix() [3/4] . . . 5

2.1.2.4 Matrix() [4/4] . . . 5

2.1.2.5 ~Matrix() . . . 5

2.1.3 Member Function Documentation . . . 6

2.1.3.1 operator"!=() . . . 6

2.1.3.2 operator()() [1/2] . . . 6

2.1.3.3 operator()() [2/2] . . . 7

2.1.3.4 operator*=() . . . 7

2.1.3.5 operator+=() . . . 8

2.1.3.6 operator-=() . . . 8

2.1.3.7 operator=() [1/2] . . . 9

2.1.3.8 operator=() [2/2] . . . 9

2.1.3.9 operator==() . . . 10

2.1.3.10 operator[]() [1/2] . . . 10

2.1.3.11 operator[]() [2/2] . . . 10

2.1.4 Friends And Related Function Documentation . . . 11

2.1.4.1 operator* . . . 11

2.1.4.2 operator<< . . . 12

2.1.4.3 to_string . . . 12

Index 15

Chapter 1

Class Index

1.1 Class List

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

MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >
A class representing a matrix of arbitrary size 3

Chapter 2

Class Documentation

2.1 MatrixLib::Matrix< _Scalar, _RowCount, _ColCount > Class Template Reference

A class representing a matrix of arbitrary size.

`#include <matrixLib.hpp>`

Public Member Functions

- constexpr Matrix ()
  Default constructor that initializes all elements to zero.
- constexpr Matrix (std::initializer_list< std::initializer_list< _Scalar >> list)
  Constructor that initializes the matrix from an initializer list.
- constexpr Matrix (const std::array< std::array< _Scalar, _ColCount >, _RowCount > &data)
  Constructor that initializes the matrix from a std::array of std::arrays.
- constexpr Matrix (const Matrix &other)
  Copy constructor.
- constexpr Matrix (Matrix &&other)
  Move constructor.
- constexpr Matrix & operator= (const Matrix &other)
  Assigns the contents of another matrix to this matrix using the copy assignment operator.
- constexpr Matrix & operator= (Matrix &&other)
  Assigns the contents of another matrix to this matrix using the move assignment operator.
- ~Matrix ()=default
  Destroys the matrix and releases any allocated resources.
- constexpr const std::array< _Scalar, _ColCount > & operator[ ] (size_t index) const
- constexpr const _Scalar & operator() (size_t indexOuter, size_t indexInner) const
- constexpr std::array< _Scalar, _ColCount > & operator[ ] (size_t index)
- constexpr _Scalar & operator() (size_t indexOuter, size_t indexInner)
- constexpr bool operator== (const Matrix &other) const
  Check if the matrix is equal to another matrix.
- constexpr bool operator!= (const Matrix &other) const
  Check if the matrix is not equal to another matrix.
- constexpr Matrix & operator+= (const Matrix &other)
- constexpr Matrix & operator-= (const Matrix &other)
- template<typename _NumericScalar >
  constexpr Matrix & operator*= (const _NumericScalar &val)

Friends

- template<typename T , size_t M, size_t N, size_t _OtherRowCount, size_t _OtherColCount>
  constexpr friend Matrix< T, M, _OtherColCount > operator* (const Matrix< T, M, N > &lhs, const Matrix< T, _OtherRowCount, _OtherColCount > &rhs)
- template<typename T , size_t M, size_t N>
  constexpr friend std::ostream & operator<< (std::ostream &os, Matrix< T, M, N > const &toPrint)

  Overload of the stream output operator for the Matrix class.
- template<typename T , size_t M, size_t N>
  constexpr friend std::string to_string (const Matrix< T, M, N > &toPrint)

  Converts the matrix to a string representation.

2.1.1 Detailed Description

template<typename _Scalar, size_t _RowCount, size_t _ColCount>
class MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >

A class representing a matrix of arbitrary size.

Template Parameters

_Scalar	The scalar type of the matrix elements. Must be a numeric type.
_RowCount	The number of rows in the matrix.
_ColCount	The number of columns in the matrix.

2.1.2 Constructor & Destructor Documentation

2.1.2.1 Matrix() [1/4]

```
template<typename _Scalar , size_t _RowCount, size_t _ColCount>
constexpr MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >::Matrix (
            std::initializer_list< std::initializer_list< _Scalar >> list )  [inline], [constexpr]
```

Constructor that initializes the matrix from an initializer list.

Parameters

list	The initializer list of lists of _Scalar values. The outer list must have _RowCount elements, and each inner list must have _ColCount elements.

Exceptions

std::invalid_argument	if the dimensions of the initializer list do not match the dimensions of the matrix.

2.1.2.2 Matrix() [2/4]

```
template<typename _Scalar , size_t _RowCount, size_t _ColCount>
constexpr MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >::Matrix (
            const std::array< std::array< _Scalar, _ColCount >, _RowCount > & data )  [inline],
[constexpr]
```

Constructor that initializes the matrix from a std::array of std::arrays.

Parameters

data	The std::array of std::arrays representing the matrix elements.

2.1.2.3 Matrix() [3/4]

```
template<typename _Scalar , size_t _RowCount, size_t _ColCount>
constexpr MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >::Matrix (
            const Matrix< _Scalar, _RowCount, _ColCount > & other )  [inline], [constexpr]
```

Copy constructor.

Parameters

other	The Matrix object to copy from.

2.1.2.4 Matrix() [4/4]

```
template<typename _Scalar , size_t _RowCount, size_t _ColCount>
constexpr MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >::Matrix (
            Matrix< _Scalar, _RowCount, _ColCount > && other )  [inline], [constexpr]
```

Move constructor.

Parameters

other	The Matrix object to move from.

2.1.2.5 ~Matrix()

```
template<typename _Scalar , size_t _RowCount, size_t _ColCount>
MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >::~Matrix ( )  [default]
```

Destroys the matrix and releases any allocated resources.

Template Parameters

_Scalar	The scalar type of the matrix.
_RowCount	The number of rows in the matrix.
_ColCount	The number of columns in the matrix.

Remarks

This destructor is implicitly declared as a defaulted function.

2.1.3 Member Function Documentation

2.1.3.1 operator"!=()

```
template<typename _Scalar , size_t _RowCount, size_t _ColCount>
constexpr bool MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >::operator!= (
            const Matrix< _Scalar, _RowCount, _ColCount > & other ) const  [inline], [constexpr]
```

Check if the matrix is not equal to another matrix.

Parameters

other	The matrix to compare against.

Returns

True if the matrices are not equal, false otherwise.

2.1.3.2 operator()() [1/2]

```
template<typename _Scalar , size_t _RowCount, size_t _ColCount>
constexpr _Scalar& MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >::operator() (
            size_t indexOuter,
            size_t indexInner )  [inline], [constexpr]
```

Access an element in the matrix using the function call operator.

Parameters

indexOuter	The row index of the element to access.
indexInner	The column index of the element to access.

Returns

A reference to the element at the specified row and column.

2.1.3.3 operator()() [2/2]

```
template<typename _Scalar , size_t _RowCount, size_t _ColCount>
constexpr const _Scalar& MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >::operator() (
            size_t indexOuter,
            size_t indexInner ) const  [inline], [constexpr]
```

Access an element in the matrix using the function call operator.

Parameters

indexOuter	The row index of the element to access.
indexInner	The column index of the element to access.

Returns

A constant reference to the element at the specified row and column.

2.1.3.4 operator*=()

```
template<typename _Scalar , size_t _RowCount, size_t _ColCount>
template<typename _NumericScalar >
constexpr Matrix& MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >::operator*= (
            const _NumericScalar & val )  [inline], [constexpr]
```

Multiply this matrix by a scalar value.

Template Parameters

_NumericScalar	The type of the scalar value to multiply by.

Parameters

val	The scalar value to multiply by.

Returns

A reference to this matrix after the multiplication.

2.1.3.5 operator+=()

```
template<typename _Scalar , size_t _RowCount, size_t _ColCount>
constexpr Matrix& MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >::operator+= (
            const Matrix< _Scalar, _RowCount, _ColCount > & other )  [inline], [constexpr]
```

Add another matrix to this matrix element-wise.

Precondition

The number of columns and rows in both matrices must be equivalent

Postcondition

The resulting matrix will have the same column and row count

Parameters

other	The matrix to add to this matrix.

Returns

A reference to this matrix after the addition.

2.1.3.6 operator-=()

```
template<typename _Scalar , size_t _RowCount, size_t _ColCount>
constexpr Matrix& MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >::operator-= (
            const Matrix< _Scalar, _RowCount, _ColCount > & other )  [inline], [constexpr]
```

Subtract another matrix to this matrix element-wise.

Precondition

The number of columns and rows in both matrices must be equivalent

Postcondition

The resulting matrix will have the same column and row count

Parameters

other	The matrix to subtract from this matrix.

Returns

A reference to this matrix after the addition.

2.1.3.7 operator=() [1/2]

```
template<typename _Scalar , size_t _RowCount, size_t _ColCount>
constexpr Matrix& MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >::operator= (
            const Matrix< _Scalar, _RowCount, _ColCount > & other )  [inline], [constexpr]
```

Assigns the contents of another matrix to this matrix using the copy assignment operator.

Template Parameters

_Scalar	The scalar type of the matrix.
_RowCount	The number of rows in the matrix.
_ColCount	The number of columns in the matrix.

Parameters

other	The matrix to copy from.

Returns

A reference to this matrix.

2.1.3.8 operator=() [2/2]

```
template<typename _Scalar , size_t _RowCount, size_t _ColCount>
constexpr Matrix& MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >::operator= (
            Matrix< _Scalar, _RowCount, _ColCount > && other )  [inline], [constexpr]
```

Assigns the contents of another matrix to this matrix using the move assignment operator.

Template Parameters

_Scalar	The scalar type of the matrix.
_RowCount	The number of rows in the matrix.
_ColCount	The number of columns in the matrix.

Parameters

other	The matrix to move from.

Returns

A reference to this matrix.

2.1.3.9 operator==()

```
template<typename _Scalar , size_t _RowCount, size_t _ColCount>
constexpr bool MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >::operator== (
            const Matrix< _Scalar, _RowCount, _ColCount > & other ) const  [inline], [constexpr]
```

Check if the matrix is equal to another matrix.

Parameters

other	The matrix to compare against.

Returns

True if the matrices are equal, false otherwise.

2.1.3.10 operator[]() [1/2]

```
template<typename _Scalar , size_t _RowCount, size_t _ColCount>
constexpr std::array<_Scalar, _ColCount>& MatrixLib::Matrix< _Scalar, _RowCount, _ColCount
>::operator[] (
            size_t index )  [inline], [constexpr]
```

Access an element in the matrix using the subscript operator.

Parameters

index	The row index of the element to access.

Returns

A reference to the array of elements in the specified row.

2.1.3.11 operator[]() [2/2]

```
template<typename _Scalar , size_t _RowCount, size_t _ColCount>
constexpr const std::array<_Scalar, _ColCount>& MatrixLib::Matrix< _Scalar, _RowCount, _Col↩
Count >::operator[] (
            size_t index ) const  [inline], [constexpr]
```

Access an element in the matrix using the subscript operator.

Parameters

index	The row index of the element to access.

Returns

A constant reference to the array of elements in the specified row.

2.1.4 Friends And Related Function Documentation

2.1.4.1 operator*

```
template<typename _Scalar , size_t _RowCount, size_t _ColCount>
template<typename T , size_t M, size_t N, size_t _OtherRowCount, size_t _OtherColCount>
constexpr friend Matrix<T, M, _OtherColCount> operator* (
            const Matrix< T, M, N > & lhs,
            const Matrix< T, _OtherRowCount, _OtherColCount > & rhs )  [friend]
```

Multiply two matrices together.

Template Parameters

T	The type of the elements in both matrices
M	The number of rows in the left-hand matrix.
N	The number of columns in the left-hand matrix and the number of rows in the right-hand matrix.
_OtherRowCount	The number of rows in the right-hand matrix.
_OtherColCount	The number of columns in the right-hand matrix.

Parameters

lhs	The left-hand matrix.
rhs	The right-hand matrix.

Precondition

The number of columns in the left matrix must be equal to the number of rows in the right matrix.

Postcondition

The resulting matrix has the same number of rows as the left matrix and the same number of columns as the right matrix.

Returns

The matrix resulting from the multiplication.

2.1.4.2 operator<<

```
template<typename _Scalar , size_t _RowCount, size_t _ColCount>
template<typename T , size_t M, size_t N>
constexpr friend std::ostream& operator<< (
            std::ostream & os,
            Matrix< T, M, N > const & toPrint )  [friend]
```

Overload of the stream output operator for the Matrix class.

Template Parameters

_Scalar	The scalar type of the matrix.
_RowCount	The number of rows in the matrix.
_ColCount	The number of columns in the matrix.

Parameters

os	The output stream to write to.
toPrint	The matrix to write to the stream.

Returns

std::ostream& The output stream after the matrix has been written.

This function overloads the stream output operator to allow easy printing of Matrix objects. The matrix is printed in row-major order, with each row printed on a separate line.

2.1.4.3 to_string

```
template<typename _Scalar , size_t _RowCount, size_t _ColCount>
template<typename T , size_t M, size_t N>
constexpr friend std::string to_string (
            const Matrix< T, M, N > & toPrint )  [friend]
```

Converts the matrix to a string representation.

Template Parameters

_Scalar	The scalar type of the matrix.
_RowCount	The number of rows in the matrix.
_ColCount	The number of columns in the matrix.

Parameters

toPrint	The matrix to convert to a string.

Returns

std::string A string representation of the matrix.

This function converts the matrix to a string representation that can be used for logging or other purposes. The matrix is printed in row-major order, with each row separated by a newline character.

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

- /home/runner/work/matrixlib/matrixlib/include/matrixLib.hpp

Index

~Matrix
MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >, 5

Matrix
MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >, 4, 5
MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >, 3
~Matrix, 5
Matrix, 4, 5
operator!=, 6
operator<<, 11
operator*, 11
operator*=, 7
operator(), 6, 7
operator+=, 7
operator-=, 8
operator=, 9
operator==, 10
operator[], 10
to_string, 12

operator!=
MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >, 6
operator<<
MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >, 11
operator*
MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >, 11
operator*=
MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >, 7
operator()
MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >, 6, 7
operator+=
MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >, 7
operator-=
MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >, 8
operator=
MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >, 9
operator==
MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >, 10
operator[]
MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >, 10

to_string
MatrixLib::Matrix< _Scalar, _RowCount, _ColCount >, 12