GPML

0.1

Generated by Doxygen 1.8.13

Contents

1	Clas	s Index			1
	1.1	Class I	_ist		1
2	File	Index			3
	2.1	File Lis	st		3
3	Clas	s Docu	mentation		5
	3.1	math::I	Matrix< N	> Class Template Reference	5
		3.1.1	Detailed	Description	5
		3.1.2	Construc	tor & Destructor Documentation	6
			3.1.2.1	Matrix() [1/3]	6
			3.1.2.2	Matrix() [2/3]	6
			3.1.2.3	Matrix() [3/3]	6
			3.1.2.4	~Matrix()	7
		3.1.3	Member	Function Documentation	7
			3.1.3.1	at()	7
			3.1.3.2	cols()	7
			3.1.3.3	operator*=() [1/2]	8
			3.1.3.4	operator*=() [2/2]	8
			3.1.3.5	operator+=()	8
			3.1.3.6	operator-=()	9
			3.1.3.7	operator/=()	9
			3.1.3.8	operator=()	10
			3.1.3.9	rows()	10
			3.1.3.10	set()	10
			3.1.3.11	shape()	11
			3 1 3 12	Size()	11

ii CONTENTS

4	File	Docum	entation	13
	4.1	/home/	/daniel/dev/cpp/math/include/Matrix.hpp File Reference	13
		4.1.1	Detailed Description	14
		4.1.2	Typedef Documentation	14
			4.1.2.1 dMatrix	14
			4.1.2.2 fMatrix	14
			4.1.2.3 iMatrix	14
		4.1.3	Function Documentation	15
			4.1.3.1 operator*() [1/3]	15
			4.1.3.2 operator*() [2/3]	15
			4.1.3.3 operator*() [3/3]	15
			4.1.3.4 operator+()	16
			4.1.3.5 operator-()	16
			4.1.3.6 operator/()	17
	4.2	/home/	/daniel/dev/cpp/math/include/typedefs.h File Reference	18
		4.2.1	Detailed Description	18
		4.2.2	Typedef Documentation	18
			4.2.2.1 uint	18
Inc	dex			19

Chapter 1

Class Index

4	4	Class	Lict
	- 1	22RL.)	I IST

Here are the classes, structs, unions	and interfaces with brief descriptions:	
math::Matrix< N >		

2 Class Index

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

/home/daniel/dev/cpp/math/include/Matrix.hpp								 					13
/home/daniel/dev/cpp/math/include/typedefs.h								 					18

File Index

Chapter 3

Class Documentation

3.1 math::Matrix < N > Class Template Reference

```
Matrix generic class.
```

```
#include <Matrix.hpp>
```

Public Member Functions

- Matrix (uint size, N fill)
- Matrix (uint rows, uint cols, N fill)
- Matrix (const Matrix &m)
- N at (uint r, uint c) const
- void set (uint r, uint c, N val)
- std::pair < uint, uint > shape () const
- uint rows () const
- uint cols () const
- uint size () const
- Matrix & operator= (const Matrix &m)
- Matrix & operator+= (const Matrix &m)
- Matrix & operator-= (const Matrix &m)
- Matrix & operator*= (const N &scal)
- Matrix & operator*= (const Matrix &m)
- Matrix & operator/= (const N &scal)
- ∼Matrix ()

3.1.1 Detailed Description

```
template<typename N> class math::Matrix< N>
```

Matrix generic class.

Author

Daniel Nichols

Date

October 2018

6 Class Documentation

3.1.2 Constructor & Destructor Documentation

Square matrix constructor. Creates a size*size matrix with every value set to fill.

Parameters

size	- size of the rows and cols of the matrix
fill	- default value for every entry

Matrix constructor. Creates a rows*cols matrix with every value set to fill.

Parameters

	rows	- number of rows			
	cols - number of cols				
Ī	fill	- default value for every entry			

uint cols,
N fill)

```
3.1.2.3 Matrix() [3/3]
```

Copy constructor. Copies matrix m into new matrix.

Parameters

m - matrix to be	copied
------------------	--------

3.1.2.4 \sim Matrix()

```
template<typename N > math::Matrix< N >::\sim Matrix ( )
```

Destructor. Deletes the matrix internally

3.1.3 Member Function Documentation

3.1.3.1 at()

```
template<typename N >
N math::Matrix< N >::at (
    uint r,
    uint c) const
```

Get element at r, c of the matrix 0-indexed.

Parameters

r	- row of return element				
С	- column of return element				

Exceptions

```
invalid_argument | thrown if r<0 or r>=rows() or c<0 or c>=cols()
```

3.1.3.2 cols()

```
template<typename N>
uint math::Matrix< N >::cols ( ) const [inline]
```

Get the number of columns in the matrix.

Returns

the number of columns in the matrix

8 Class Documentation

```
3.1.3.3 operator*=() [1/2]
```

Adds Multiplies this by scaler scal

Parameters

```
scal - scaler to multiply this by
```

Returns

a pointer to m after multiplication

```
3.1.3.4 operator*=() [2/2]
```

Performs matrix multiplication between this and m. This operation will throw an exception if cols!=m.rows(). It will also reshape this to that rows() does not change and cols() becomes m.cols().

Parameters

```
m - matrix to multiply by this.
```

Returns

a pointer to this after multiplication.

Exceptions

```
invalid_argument | if cols()!=m.rows() matrix multiplication is undefined
```

3.1.3.5 operator+=()

Adds matrix m to this element-wise

Parameters

```
m - matrix to add to this. rows and cols must be equivalent.
```

Returns

a pointer to this after addition

Exceptions

```
invalid_argument | thrown if rows()!=m.rows() or cols()!=m.cols()
```

3.1.3.6 operator-=()

Subtracts matrix m from this element-wise

Parameters

```
m - matrix to subtract from this. rows and cols must be equivalent.
```

Returns

a pointer to this after subtraction

Exceptions

```
invalid_argument | thrown if rows()!=m.rows() or cols()!=m.cols()
```

3.1.3.7 operator/=()

Divides this by scalar scal element-wise. Does not check for scal==0 as division for class N might have non-standard definition.

Parameters

```
scal - scalar to divide this by
```

Generated by Doxygen

10 Class Documentation

Returns

a pointer to m after division

3.1.3.8 operator=()

Copies m into this. Performs an element-wise copy. Ignores self-copy.

Parameters

```
m - matrix to copy into this
```

Returns

pointer to this after copy

3.1.3.9 rows()

```
template<typename N>
uint math::Matrix< N >::rows ( ) const [inline]
```

Get the number of rows in the matrix.

Returns

the number of rows in the matrix

3.1.3.10 set()

Set element at r, c of the matrix 0-indexed.

Parameters

r	- row of element set
С	- column of element set
val	- value to set element at r,c

Exceptions

```
invalid_argument | thrown if r<0 or r>=rows() or c<0 or c>=cols()
```

3.1.3.11 shape()

```
template<typename N>
std::pair<uint, uint> math::Matrix< N >::shape ( ) const [inline]
```

Get the shape or (rows, cols). This is equivalent to std::make_pair(rows(), cols());

Returns

an STL pair containing the row count and column count

3.1.3.12 size()

```
template<typename N>
uint math::Matrix< N >::size ( ) const [inline]
```

Get the size of the matrix (size() == rows() * cols())

Returns

the size of the matrix

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

/home/daniel/dev/cpp/math/include/Matrix.hpp

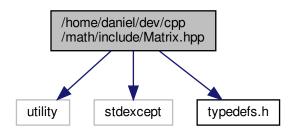
12 Class Documentation

Chapter 4

File Documentation

4.1 /home/daniel/dev/cpp/math/include/Matrix.hpp File Reference

```
#include <utility>
#include <stdexcept>
#include "typedefs.h"
Include dependency graph for Matrix.hpp:
```



Classes

class math::Matrix < N >
 Matrix generic class.

Typedefs

- typedef Matrix < int > math::iMatrix
- typedef Matrix< float > math::fMatrix
- typedef Matrix < double > math::dMatrix

14 **File Documentation**

Functions

```
• template<typename N >
      Matrix < N > math::operator + (Matrix < N > lhs, const Matrix < N > &rhs)
    • template<typename N >
      Matrix< N > math::operator- (Matrix< N > lhs, const Matrix< N > &rhs)
    • template<typename N >
      Matrix < N > math::operator* (Matrix < N > lhs, const N &rhs)
    • template<typename N >
      Matrix< N > math::operator∗ (const N &lhs, Matrix< N > rhs)
    • template<typename N >
      Matrix < N > math::operator/ (Matrix < N > lhs, const N &rhs)
    template<typename N >
      Matrix < N > math::operator* (Matrix < N > lhs, const Matrix < N > &rhs)
4.1.1
       Detailed Description
```

Contains Matrix class definition and implementation.

Author

Daniel Nichols

Date

October 2018

4.1.2 Typedef Documentation

4.1.2.1 dMatrix

```
typedef Matrix<double> math::dMatrix
```

double precision matrix

4.1.2.2 fMatrix

```
typedef Matrix<float> math::fMatrix
```

float precision matrix

4.1.2.3 iMatrix

```
typedef Matrix<int> math::iMatrix
```

integer matrix

4.1.3 Function Documentation

const N & rhs)

Multiplies Ihs and scalar rhs. Copies Ihs and multiplies by scalar rhs

Parameters

lhs	- left hand side matrix
rhs	- right hand side scalar

Returns

a new matrix with elements multiplication of lhs and rhs

4.1.3.2 operator*() [2/3]

Multiplies scalar lhs and matrix rhs. Copies rhs and multiplies by scalar lhs

Parameters

lhs	- left hand side scalar	
rhs	- right hand side matrix	

Returns

a new matrix with elements multiplication of lhs and rhs

4.1.3.3 operator*() [3/3]

```
template<typename N >
Matrix<N> math::operator* (
```

16 File Documentation

```
\label{eq:matrix} \begin{array}{ll} \text{Matrix} < \text{N} > lhs, \\ \text{const Matrix} < \text{N} > \text{\& } rhs \end{array} )
```

Performs matrix multiplication of rhs and lhs

Parameters

lhs	- left hand side matrix	
rhs	- right hand side matrix	

Returns

a new matrix resulting from matrix multiplication. Result will have shape lhs.rows(), rhs.cols().

Exceptions

```
invalid_argument | if lhs.cols() != rhs.rows()
```

4.1.3.4 operator+()

Adds lhs and rhs matrices element-wise. Copies lhs and add rhs to it.

Parameters

lhs	- left hand side matrix of addition
rhs	- right hand side matrix of addition

Returns

a new matrix with elements from element-wise addition of lhs and rhs

Exceptions

```
invalid_argument | if lhs and rhs do not have same shape
```

4.1.3.5 operator-()

```
template<typename N > Matrix< N > math::operator- (
```

```
\label{eq:matrix} \begin{array}{ll} \text{Matrix} < \text{N} > lhs, \\ \text{const Matrix} < \text{N} > \text{\& } rhs \end{array} )
```

Subtracts lhs and rhs matrices element-wise. Copies lhs and subtract rhs from it.

Parameters

lhs	- left hand side matrix of subtraction
rhs	- right hand side matrix of subtraction

Returns

a new matrix with elements from element-wise subtraction of lhs and rhs

Exceptions

	invalid_argument	if lhs and rhs do not have same shape
--	------------------	---------------------------------------

4.1.3.6 operator/()

Divides lhs matrix by scalar rhs. Copies lhs and divides by scalar rhs. Does not check if rhs is zero due to unknown type of \mathbb{N} .

Parameters

lhs	- left hand side matrix - right hand side scalar	
rhs		

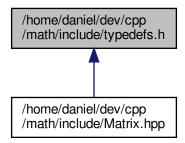
18 File Documentation

Returns

a new matrix with elements division of lhs and rhs

4.2 /home/daniel/dev/cpp/math/include/typedefs.h File Reference

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



Typedefs

• typedef unsigned int math::uint

4.2.1 Detailed Description

Defines utilility types for math library.

Author

Daniel Nichols

Date

October 2018

4.2.2 Typedef Documentation

4.2.2.1 uint

typedef unsigned int math::uint

shorthand for unsigned int type

Index

operator+

/home/daniel/dev/cpp/math/include/Matrix.hpp, 13 /home/daniel/dev/cpp/math/include/typedefs.h, 18 ~Matrix		Matrix.hpp, 16 operator+= math::Matrix, 8		
	math::Matrix, 7	ope	rator- Matrix.hpp, 16	
at		ope	rator-=	
	math::Matrix, 7	•	math::Matrix, 9	
		ope	rator/	
cols	math::Matrix, 7		Matrix.hpp, 17	
	maiiwaiix, 7	ope	rator/=	
dMat	rix		math::Matrix, 9	
	Matrix.hpp, 14	ope	rator= math::Matrix, 10	
fMatr	rix			
	Matrix.hpp, 14	row		
			math::Matrix, 10	
iMatr		set		
	Matrix.hpp, 14		math::Matrix, 10	
math	::Matrix	sha		
	\sim Matrix, 7		math::Matrix, 11	
	at, 7	size		
	cols, 7		math::Matrix, 11	
	Matrix, 6	t vo c	dofo b	
	operator*=, 7, 8	туре	edefs.h uint, 18	
	operator+=, 8		uiiii, 10	
	operator-=, 9	uint		
	operator/=, 9		typedefs.h, 18	
	operator=, 10		,	
	rows, 10			
	set, 10			
	shape, 11			
	size, 11			
	::Matrix< N >, 5			
Matri				
	math::Matrix, 6 ix.hpp			
	dMatrix, 14			
	fMatrix, 14			
	iMatrix, 14			
	operator*, 15			
	operator+, 16			
	operator-, 16			
	operator/, 17			
opera				
	Matrix.hpp, 15			
•	ator*= math::Matrix, 7, 8			
	manimization, 7, 0			