mcpp

Generated by Doxygen 1.9.2

1 Table of contets 1

2 Namespace Index 3

2.1 Namespace List . . . 3

3 Class Index 5

3.1 Class List . . . 5

4 File Index 7

4.1 File List . . . 7

5 Namespace Documentation 9

5.1 var Namespace Reference . . . 9

5.2 vec Namespace Reference . . . 9

5.2.1 Function Documentation . . . 10

5.2.1.1 acos() . . . 10

5.2.1.2 acosh() . . . 10

5.2.1.3 acot() . . . 11

5.2.1.4 acoth() . . . 11

5.2.1.5 acsc() . . . 11

5.2.1.6 acsch() . . . 11

5.2.1.7 asec() . . . 11

5.2.1.8 asech() . . . 11

5.2.1.9 asin() . . . 12

5.2.1.10 asinh() . . . 12

5.2.1.11 atan() . . . 12

5.2.1.12 atanh() . . . 12

5.2.1.13 cos() . . . 12

5.2.1.14 cosh() . . . 12

5.2.1.15 cot() . . . 13

5.2.1.16 coth() . . . 13

5.2.1.17 cross() . . . 13

5.2.1.18 csc() . . . 13

5.2.1.19 csch() . . . 13

5.2.1.20 dot() . . . 13

5.2.1.21 exp() . . . 14

5.2.1.22 pow() `[1/2]` . . . 14

5.2.1.23 pow() `[2/2]` . . . 14

5.2.1.24 sec() . . . 14

5.2.1.25 sech() . . . 14

5.2.1.26 sin() . . . 14

5.2.1.27 sinh() . . . 15

5.2.1.28 tan() . . . 15

5.2.1.29 tanh() . . . 15

6 Class Documentation **17**

6.1 var::matrix< S > Class Template Reference . . . 17

6.1.1 Detailed Description . . . 20

6.1.2 Constructor & Destructor Documentation . . . 20

6.1.2.1 matrix() [1/3] . . . 20

6.1.2.2 matrix() [2/3] . . . 20

6.1.2.3 matrix() [3/3] . . . 21

6.1.3 Member Function Documentation . . . 21

6.1.3.1 col() . . . 21

6.1.3.2 col_op() . . . 21

6.1.3.3 det() . . . 21

6.1.3.4 insert_col() [1/2] . . . 21

6.1.3.5 insert_col() [2/2] . . . 22

6.1.3.6 insert_row() [1/2] . . . 22

6.1.3.7 insert_row() [2/2] . . . 22

6.1.3.8 inv() . . . 23

6.1.3.9 is_identity() . . . 23

6.1.3.10 is_square() . . . 23

6.1.3.11 operator*() . . . 23

6.1.3.12 operator+() . . . 24

6.1.3.13 operator-() . . . 24

6.1.3.14 operator/() . . . 24

6.1.3.15 operator[]() . . . 24

6.1.3.16 resize() . . . 24

6.1.3.17 row() . . . 25

6.1.3.18 row_op() . . . 25

6.1.3.19 rref() . . . 25

6.1.3.20 size() . . . 25

6.1.3.21 sort_col() . . . 26

6.1.3.22 sort_cols() . . . 26

6.1.3.23 sort_row() . . . 26

6.1.3.24 sort_rows() . . . 26

6.1.3.25 sum() . . . 26

6.1.3.26 T() . . . 27

6.1.3.27 turn_to() . . . 27

6.1.4 Friends And Related Function Documentation . . . 27

6.1.4.1 operator<< . . . 27

6.1.5 Member Data Documentation . . . 28

6.1.5.1 _col . . . 28

6.1.5.2 _row . . . 28

6.1.5.3 data . . . 28

6.2 var::matrix< S >::Row Class Reference . . . 29

6.2.1 Constructor & Destructor Documentation 30
6.2.1.1 Row() 30
6.2.2 Member Function Documentation 30
6.2.2.1 operator[]() 30
6.2.3 Member Data Documentation 30
6.2.3.1 _a 30
6.2.3.2 _i 30

7 File Documentation **31**
7.1 includes/mcpp.hpp File Reference 31
7.2 includes/misc/constants.hpp File Reference 31
7.2.1 Typedef Documentation 32
7.2.1.1 table 32
7.3 includes/plot/figure.hpp File Reference 33
7.3.1 Function Documentation 34
7.3.1.1 check_param() 34
7.3.1.2 figure() 34
7.3.1.3 hline() 34
7.3.1.4 plot() 35
7.3.1.5 point() 35
7.3.1.6 vline() 35
7.3.2 Variable Documentation 36
7.3.2.1 BLACK 36
7.3.2.2 BLUE 36
7.3.2.3 GREEN 36
7.3.2.4 RED 36
7.3.2.5 WHITE 36
7.4 includes/variables/graph.hpp File Reference 37
7.5 includes/variables/matrix.hpp File Reference 37
7.6 includes/vectors/vec.hpp File Reference 38
7.7 includes/vectors/vector.hpp File Reference 40
7.7.1 Function Documentation 41
7.7.1.1 operator%() 41
7.7.1.2 operator*() [1/3] 42
7.7.1.3 operator*() [2/3] 42
7.7.1.4 operator*() [3/3] 43
7.7.1.5 operator+() [1/3] 43
7.7.1.6 operator+() [2/3] 44
7.7.1.7 operator+() [3/3] 44
7.7.1.8 operator-() [1/3] 45
7.7.1.9 operator-() [2/3] 45
7.7.1.10 operator-() [3/3] 46

7.7.1.11 operator/() [1/3] 46
7.7.1.12 operator/() [2/3] 47
7.7.1.13 operator/() [3/3] 47
7.7.1.14 operator<<() 48
7.8 README.md File Reference 48

Index **49**

Chapter 1

Table of contets

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

var 9
vec 9

Chapter 3

Class Index

3.1 Class List

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

var::matrix< S >
Class for 2d matrix of objects with matrix properties 17
var::matrix< S >::Row 29

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

includes/mcpp.hpp 31
includes/misc/constants.hpp 31
includes/plot/figure.hpp 33
includes/variables/graph.hpp 37
includes/variables/matrix.hpp 37
includes/vectors/vec.hpp 38
includes/vectors/vector.hpp 40

Chapter 5

Namespace Documentation

5.1 var Namespace Reference

Classes

- class matrix

  Class for 2d matrix of objects with matrix properties.

5.2 vec Namespace Reference

Functions

- template<typename S >
  std::vector< S > dot (std::vector< S > a, std::vector< S > b)
- template<typename S >
  std::vector< S > cross (std::vector< S > a, std::vector< S > b)
- template<typename S >
  std::vector< S > pow (std::vector< S > a, S b)
- template<typename S >
  std::vector< S > pow (std::vector< S > a, std::vector< S > b)
- template<typename S >
  std::vector< S > sin (std::vector< S > a)
- template<typename S >
  std::vector< S > cos (std::vector< S > a)
- template<typename S >
  std::vector< S > tan (std::vector< S > a)
- template<typename S >
  std::vector< S > sec (std::vector< S > a)
- template<typename S >
  std::vector< S > csc (std::vector< S > a)
- template<typename S >
  std::vector< S > cot (std::vector< S > a)
- template<typename S >
  std::vector< S > asin (std::vector< S > a)
- template<typename S >
  std::vector< S > acos (std::vector< S > a)

- template<typename S >
  std::vector< S > atan (std::vector< S > a)
- template<typename S >
  std::vector< S > asec (std::vector< S > a)
- template<typename S >
  std::vector< S > acsc (std::vector< S > a)
- template<typename S >
  std::vector< S > acot (std::vector< S > a)
- template<typename S >
  std::vector< S > sinh (std::vector< S > a)
- template<typename S >
  std::vector< S > cosh (std::vector< S > a)
- template<typename S >
  std::vector< S > tanh (std::vector< S > a)
- template<typename S >
  std::vector< S > sech (std::vector< S > a)
- template<typename S >
  std::vector< S > csch (std::vector< S > a)
- template<typename S >
  std::vector< S > coth (std::vector< S > a)
- template<typename S >
  std::vector< S > asinh (std::vector< S > a)
- template<typename S >
  std::vector< S > acosh (std::vector< S > a)
- template<typename S >
  std::vector< S > atanh (std::vector< S > a)
- template<typename S >
  std::vector< S > asech (std::vector< S > a)
- template<typename S >
  std::vector< S > acsch (std::vector< S > a)
- template<typename S >
  std::vector< S > acoth (std::vector< S > a)
- template<typename S >
  std::vector< S > exp (std::vector< S > a)

5.2.1 Function Documentation

5.2.1.1 acos()

```
template<typename S >
std::vector<S> vec::acos (
            std::vector< S > a )
```

5.2.1.2 acosh()

```
template<typename S >
std::vector<S> vec::acosh (
            std::vector< S > a )
```

5.2.1.3 acot()

```
template<typename S >
std::vector<S> vec::acot (
            std::vector< S > a )
```

5.2.1.4 acoth()

```
template<typename S >
std::vector<S> vec::acoth (
            std::vector< S > a )
```

5.2.1.5 acsc()

```
template<typename S >
std::vector<S> vec::acsc (
            std::vector< S > a )
```

5.2.1.6 acsch()

```
template<typename S >
std::vector<S> vec::acsch (
            std::vector< S > a )
```

5.2.1.7 asec()

```
template<typename S >
std::vector<S> vec::asec (
            std::vector< S > a )
```

5.2.1.8 asech()

```
template<typename S >
std::vector<S> vec::asech (
            std::vector< S > a )
```

5.2.1.9 asin()

```
template<typename S >
std::vector<S> vec::asin (
             std::vector< S > a )
```

5.2.1.10 asinh()

```
template<typename S >
std::vector<S> vec::asinh (
             std::vector< S > a )
```

5.2.1.11 atan()

```
template<typename S >
std::vector<S> vec::atan (
             std::vector< S > a )
```

5.2.1.12 atanh()

```
template<typename S >
std::vector<S> vec::atanh (
             std::vector< S > a )
```

5.2.1.13 cos()

```
template<typename S >
std::vector<S> vec::cos (
             std::vector< S > a )
```

5.2.1.14 cosh()

```
template<typename S >
std::vector<S> vec::cosh (
             std::vector< S > a )
```

5.2.1.15 cot()

```
template<typename S >
std::vector<S> vec::cot (
            std::vector< S > a )
```

5.2.1.16 coth()

```
template<typename S >
std::vector<S> vec::coth (
            std::vector< S > a )
```

5.2.1.17 cross()

```
template<typename S >
std::vector<S> vec::cross (
            std::vector< S > a,
            std::vector< S > b )
```

5.2.1.18 csc()

```
template<typename S >
std::vector<S> vec::csc (
            std::vector< S > a )
```

5.2.1.19 csch()

```
template<typename S >
std::vector<S> vec::csch (
            std::vector< S > a )
```

5.2.1.20 dot()

```
template<typename S >
std::vector<S> vec::dot (
            std::vector< S > a,
            std::vector< S > b )
```

5.2.1.21 exp()

```
template<typename S >
std::vector<S> vec::exp (
             std::vector< S > a )
```

5.2.1.22 pow() [1/2]

```
template<typename S >
std::vector<S> vec::pow (
             std::vector< S > a,
             S b )
```

5.2.1.23 pow() [2/2]

```
template<typename S >
std::vector<S> vec::pow (
             std::vector< S > a,
             std::vector< S > b )
```

5.2.1.24 sec()

```
template<typename S >
std::vector<S> vec::sec (
             std::vector< S > a )
```

5.2.1.25 sech()

```
template<typename S >
std::vector<S> vec::sech (
             std::vector< S > a )
```

5.2.1.26 sin()

```
template<typename S >
std::vector<S> vec::sin (
             std::vector< S > a )
```

5.2.1.27 sinh()

```
template<typename S >
std::vector<S> vec::sinh (
            std::vector< S > a )
```

5.2.1.28 tan()

```
template<typename S >
std::vector<S> vec::tan (
            std::vector< S > a )
```

5.2.1.29 tanh()

```
template<typename S >
std::vector<S> vec::tanh (
            std::vector< S > a )
```

Chapter 6

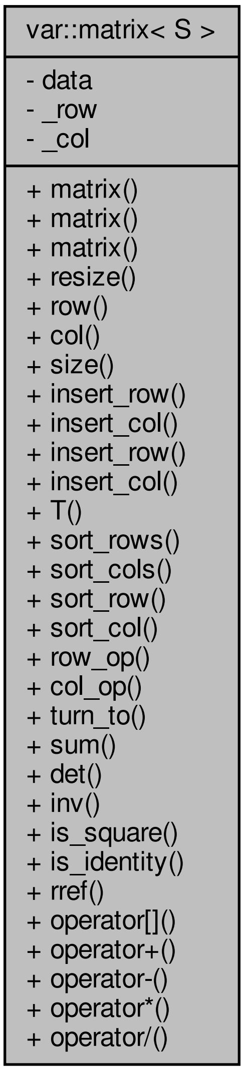
Class Documentation

6.1 var::matrix< S > Class Template Reference

Class for 2d matrix of objects with matrix properties.

```
#include "matrix.hpp"
```

Collaboration diagram for var::matrix< S >:



Classes

- class Row

Public Member Functions

- matrix (int r, int c)

  Construct a new matrix object.
- matrix ()

  Default construct a new matrix object.
- matrix (std::initializer_list< std::initializer_list< S >> a)

- void resize (int r, int c)

  resizes the matrix
- int row ()

  returns the number of rows
- int col ()

  returns the number of columns
- int size ()

  returns total number of elements
- void insert_row (const std::vector< S > &a)

  insterts row at the end
- void insert_col (const std::vector< S > &a)

  insterts column at the end
- void insert_row (int i, const std::vector< S > &a)

  inserts row at specefic index
- void insert_col (int j, const std::vector< S > &a)

  inserts column at specefic index
- void T ()

  mutates data to into transpose
- void sort_rows (int d=1)

  sorts all rows
- void sort_cols (int d=1)

  sorts all columns
- void sort_row (int i, int d=1)
- void sort_col (int j, int d=1)
- template< typename LAMBDA >
  void row_op (int i, LAMBDA f)
- template< typename LAMBDA >
  void col_op (int j, LAMBDA f)
- void turn_to (S n)

  converts all elements to n
- S sum ()

  sum of all elements
- S det ()

  returns the determinant
- S inv ()

  returns the inverse
- bool is_square ()

  checks if matrix is square
- bool is_identity ()

  checks if matrix is an identity matrix
- S rref ()

  rref form of matrix
- Row operator[ ] (int i)
- matrix operator+ (matrix const &other)
- matrix operator- (matrix const &other)
- matrix operator* (matrix const &other)
- matrix operator/ (matrix const &other)

Private Attributes

- table< S > data
- int _row
- int _col

Friends

- std::ostream & operator<< (std::ostream &out, matrix const &other)

    print method for the class var::matrix<int> m; cout << m;

6.1.1 Detailed Description

template<typename S>
class var::matrix< S >

Class for 2d matrix of objects with matrix properties.

Template Parameters

S	can be of any type

6.1.2 Constructor & Destructor Documentation

6.1.2.1 matrix() [1/3]

```
template<typename S >
var::matrix< S >::matrix (
              int r,
              int c )  [inline]
```

Construct a new matrix object.

Parameters

r	number of rows
c	number of colums

6.1.2.2 matrix() [2/3]

```
template<typename S >
var::matrix< S >::matrix ( )  [inline]
```

Default construct a new matrix object.

6.1.2.3 matrix() [3/3]

```
template<typename S >
var::matrix< S >::matrix (
             std::initializer_list< std::initializer_list< S >> a )  [inline]
```

6.1.3 Member Function Documentation

6.1.3.1 col()

```
template<typename S >
int var::matrix< S >::col ( )  [inline]
```

returns the number of columns

Returns

int

6.1.3.2 col_op()

```
template<typename S >
template<typename LAMBDA >
void var::matrix< S >::col_op (
             int j,
             LAMBDA f )  [inline]
```

6.1.3.3 det()

```
template<typename S >
S var::matrix< S >::det ( )  [inline]
```

returns the determinant

Returns

S

6.1.3.4 insert_col() [1/2]

```
template<typename S >
void var::matrix< S >::insert_col (
             const std::vector< S > & a )  [inline]
```

insterts column at the end

Parameters

a	

6.1.3.5 insert_col() [2/2]

```
template<typename S >
void var::matrix< S >::insert_col (
             int j,
             const std::vector< S > & a )  [inline]
```

inserts column at specefic index

Parameters

j	
a	

6.1.3.6 insert_row() [1/2]

```
template<typename S >
void var::matrix< S >::insert_row (
             const std::vector< S > & a )  [inline]
```

insterts row at the end

Parameters

a	

6.1.3.7 insert_row() [2/2]

```
template<typename S >
void var::matrix< S >::insert_row (
             int i,
             const std::vector< S > & a )  [inline]
```

inserts row at specefic index

Parameters

i	
a	

6.1.3.8 inv()

```
template<typename S >
S var::matrix< S >::inv ( )  [inline]
```

returns the inverse

Returns

S

6.1.3.9 is_identity()

```
template<typename S >
bool var::matrix< S >::is_identity ( )  [inline]
```

checks if matrix is an identity matrix

Returns

true

false

6.1.3.10 is_square()

```
template<typename S >
bool var::matrix< S >::is_square ( )  [inline]
```

checks if matrix is square

Returns

true

false

6.1.3.11 operator*()

```
template<typename S >
matrix var::matrix< S >::operator* (
             matrix< S > const & other )  [inline]
```

6.1.3.12 operator+()

```
template<typename S >
matrix var::matrix< S >::operator+ (
             matrix< S > const & other )  [inline]
```

6.1.3.13 operator-()

```
template<typename S >
matrix var::matrix< S >::operator- (
             matrix< S > const & other )  [inline]
```

6.1.3.14 operator/()

```
template<typename S >
matrix var::matrix< S >::operator/ (
             matrix< S > const & other )  [inline]
```

6.1.3.15 operator[]()

```
template<typename S >
Row var::matrix< S >::operator[] (
             int i )  [inline]
```

6.1.3.16 resize()

```
template<typename S >
void var::matrix< S >::resize (
             int r,
             int c )  [inline]
```

resizes the matrix

Parameters

r	number of rows
c	number of colums

6.1.3.17 row()

```
template<typename S >
int var::matrix< S >::row ( )  [inline]
```

returns the number of rows

Returns

int

6.1.3.18 row_op()

```
template<typename S >
template<typename LAMBDA >
void var::matrix< S >::row_op (
             int i,
             LAMBDA f )  [inline]
```

6.1.3.19 rref()

```
template<typename S >
S var::matrix< S >::rref ( )  [inline]
```

rref form of matrix

Returns

S

6.1.3.20 size()

```
template<typename S >
int var::matrix< S >::size ( )  [inline]
```

returns total number of elements

Returns

int

6.1.3.21 sort_col()

```
template<typename S >
void var::matrix< S >::sort_col (
             int j,
             int d = 1 )  [inline]
```

6.1.3.22 sort_cols()

```
template<typename S >
void var::matrix< S >::sort_cols (
             int d = 1 )  [inline]
```

sorts all columns

Parameters

d	d = 1 is accending order -> sort_cols() d = 0 is decending order -> sort_cols(0)

6.1.3.23 sort_row()

```
template<typename S >
void var::matrix< S >::sort_row (
             int i,
             int d = 1 )  [inline]
```

6.1.3.24 sort_rows()

```
template<typename S >
void var::matrix< S >::sort_rows (
             int d = 1 )  [inline]
```

sorts all rows

Parameters

d	d = 1 is accending order -> sort_rows() d = 0 is decending order -> sort_rows(0)

6.1.3.25 sum()

```
template<typename S >
```

```
S var::matrix< S >::sum ( )  [inline]
```

sum of all elements

Returns

S

6.1.3.26 T()

```
template<typename S >
void var::matrix< S >::T ( )  [inline]
```

mutates data to into transpose

6.1.3.27 turn_to()

```
template<typename S >
void var::matrix< S >::turn_to (
             S n )  [inline]
```

converts all elements to n

Parameters

n	

6.1.4 Friends And Related Function Documentation

6.1.4.1 operator<<

```
template<typename S >
std::ostream& operator<< (
             std::ostream & out,
             matrix< S > const & other )  [friend]
```

print method for the class var::matrix<int> m; cout << m;

Template Parameters

S	

Parameters

out	
other	

Returns

std::ostream&

6.1.5 Member Data Documentation

6.1.5.1 _col

```
template<typename S >
int var::matrix< S >::_col  [private]
```

6.1.5.2 _row

```
template<typename S >
int var::matrix< S >::_row  [private]
```

6.1.5.3 data

```
template<typename S >
table<S> var::matrix< S >::data  [private]
```

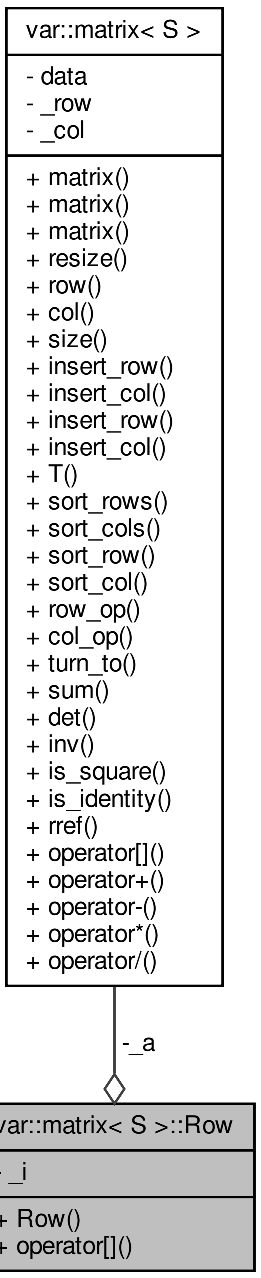
The documentation for this class was generated from the following file:

- includes/variables/matrix.hpp

6.2 var::matrix< S >::Row Class Reference

```
#include "matrix.hpp"
```

Collaboration diagram for var::matrix< S >::Row:



Public Member Functions

- Row (matrix &a, int i)
- S & operator[ ] (int j)

Private Attributes

- matrix & _a
- int _i

6.2.1 Constructor & Destructor Documentation

6.2.1.1 Row()

```
template<typename S >
var::matrix< S >::Row::Row (
             matrix & a,
             int i )  [inline]
```

6.2.2 Member Function Documentation

6.2.2.1 operator[]()

```
template<typename S >
S& var::matrix< S >::Row::operator[] (
             int j )  [inline]
```

6.2.3 Member Data Documentation

6.2.3.1 _a

```
template<typename S >
matrix& var::matrix< S >::Row::_a  [private]
```

6.2.3.2 _i

```
template<typename S >
int var::matrix< S >::Row::_i  [private]
```

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

- includes/variables/matrix.hpp

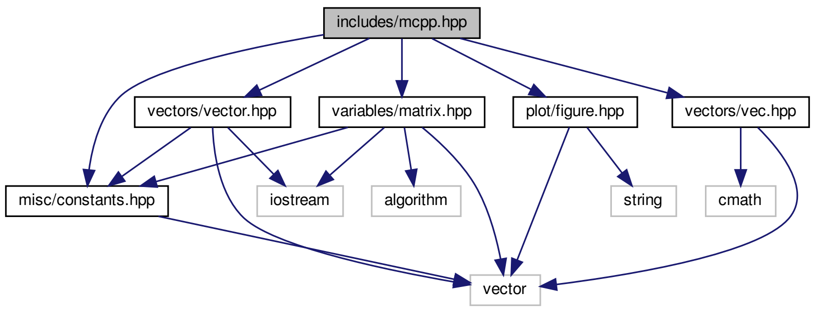
Chapter 7

File Documentation

7.1 includes/mcpp.hpp File Reference

```
#include "misc/constants.hpp"
#include "variables/matrix.hpp"
#include "vectors/vector.hpp"
#include "vectors/vec.hpp"
#include "plot/figure.hpp"
```

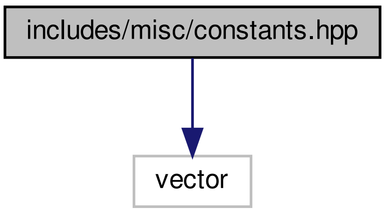
Include dependency graph for mcpp.hpp:



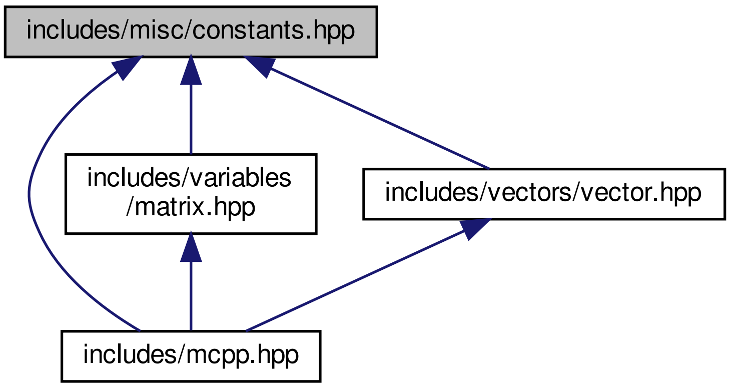
7.2 includes/misc/constants.hpp File Reference

```
#include <vector>
```

Include dependency graph for constants.hpp:



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



Typedefs

- template<typename T >
  using table = std::vector< std::vector< T > >

7.2.1 Typedef Documentation

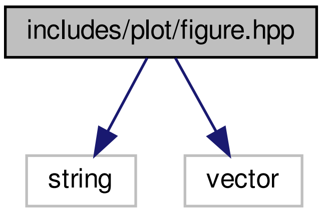
7.2.1.1 table

```
template<typename T >
using table = std::vector<std::vector<T> >
```

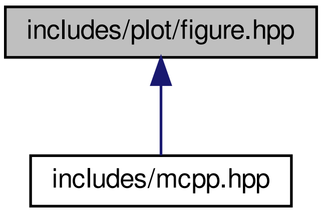
7.3 includes/plot/figure.hpp File Reference

```
#include <string>
#include <vector>
```

Include dependency graph for figure.hpp:



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



Functions

- void check_param (int exp, int act)

    checks parameter for the functions
- void figure (std::string title="Window", int size=400, std::vector< int > background=WHITE)

    initializes figure
- void plot (std::vector< float > x, std::vector< float > y, std::vector< float > xrange={-5, 5}, std::vector< float > yrange={-5, 5}, std::vector< int > draw=BLACK, std::string legend="")

    plots values
- void point (float x, float y, std::vector< float > xrange={-5, 5}, std::vector< float > yrange={-5, 5}, std::vector< int > draw=BLACK, std::string legend="")
- void hline (float x, float y, std::vector< float > xrange={-5, 5}, std::vector< float > yrange={-5, 5}, std::vector< int > draw=BLACK, std::string legend="")
- void vline (float x, float y, std::vector< float > xrange={-5, 5}, std::vector< float > yrange={-5, 5}, std::string legend="", std::vector< int > draw=BLACK)

Variables

- const std::vector< int > RED = {247, 55, 49}

  RGB values for colors.
- const std::vector< int > BLACK = {0, 0, 0}
- const std::vector< int > BLUE = {36, 114, 200}
- const std::vector< int > GREEN = {53, 200, 36}
- const std::vector< int > WHITE = {255, 255, 255}

7.3.1 Function Documentation

7.3.1.1 check_param()

```
void check_param (
             int exp,
             int act )
```

checks parameter for the functions

Parameters

exp	
act	

7.3.1.2 figure()

```
void figure (
             std::string title = "Window",
             int size = 400,
             std::vector< int > background = WHITE )
```

initializes figure

Parameters

title	
size	
background	

7.3.1.3 hline()

```
void hline (
             float x,
             float y,
             std::vector< float > xrange = {-5, 5},
             std::vector< float > yrange = {-5, 5},
             std::vector< int > draw = BLACK,
             std::string legend = "" )
```

7.3.1.4 plot()

```
void plot (
            std::vector< float > x,
            std::vector< float > y,
            std::vector< float > xrange = {-5, 5},
            std::vector< float > yrange = {-5, 5},
            std::vector< int > draw = BLACK,
            std::string legend = "" )
```

plots values

Parameters

x	
y	
xrange	
yrange	
draw	
legend	

7.3.1.5 point()

```
void point (
            float x,
            float y,
            std::vector< float > xrange = {-5, 5},
            std::vector< float > yrange = {-5, 5},
            std::vector< int > draw = BLACK,
            std::string legend = "" )
```

7.3.1.6 vline()

```
void vline (
            float x,
            float y,
            std::vector< float > xrange = {-5, 5},
            std::vector< float > yrange = {-5, 5},
            std::string legend = "",
            std::vector< int > draw = BLACK )
```

7.3.2 Variable Documentation

7.3.2.1 BLACK

```
const std::vector<int> BLACK = {0, 0, 0}  [extern]
```

7.3.2.2 BLUE

```
const std::vector<int> BLUE = {36, 114, 200}  [extern]
```

7.3.2.3 GREEN

```
const std::vector<int> GREEN = {53, 200, 36}  [extern]
```

7.3.2.4 RED

```
const std::vector<int> RED = {247, 55, 49}  [extern]
```

RGB values for colors.

7.3.2.5 WHITE

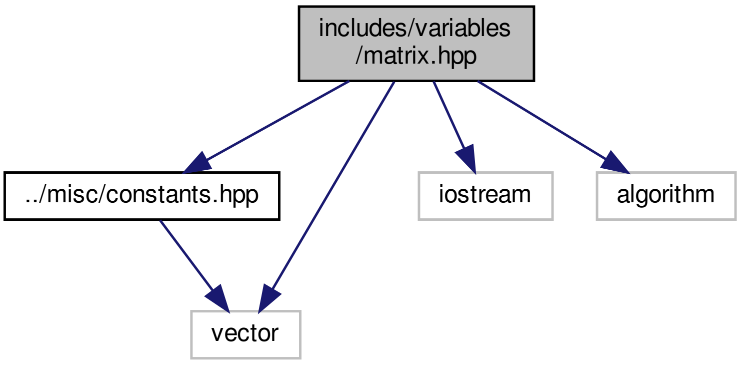
```
const std::vector<int> WHITE = {255, 255, 255}  [extern]
```

7.4 includes/variables/graph.hpp File Reference

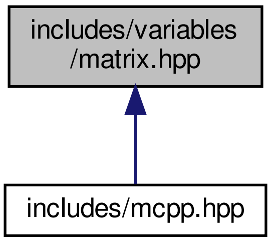
7.5 includes/variables/matrix.hpp File Reference

```
#include "../misc/constants.hpp"
#include <iostream>
#include <vector>
#include <algorithm>
```

Include dependency graph for matrix.hpp:



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



Classes

- class var::matrix< S >

  Class for 2d matrix of objects with matrix properties.
- class var::matrix< S >::Row

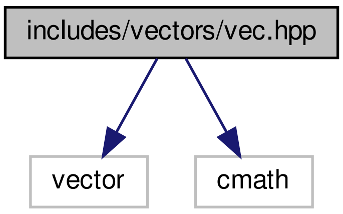
Namespaces

- namespace var

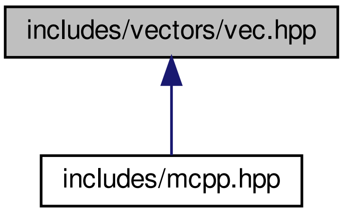
7.6 includes/vectors/vec.hpp File Reference

```
#include <vector>
#include <cmath>
```

Include dependency graph for vec.hpp:



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



Namespaces

- namespace vec

Functions

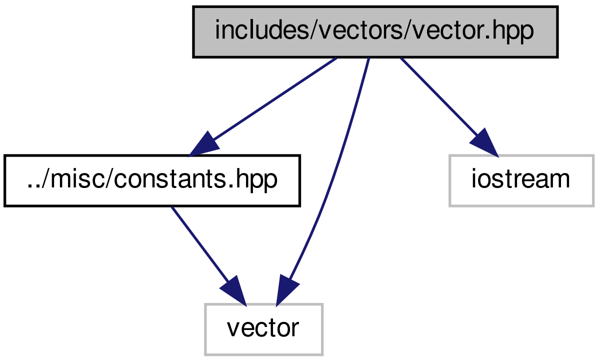
- template<typename S >
  std::vector< S > vec::dot (std::vector< S > a, std::vector< S > b)
- template<typename S >
  std::vector< S > vec::cross (std::vector< S > a, std::vector< S > b)

- template<typename S >
  std::vector< S > vec::pow (std::vector< S > a, S b)
- template<typename S >
  std::vector< S > vec::pow (std::vector< S > a, std::vector< S > b)
- template<typename S >
  std::vector< S > vec::sin (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::cos (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::tan (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::sec (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::csc (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::cot (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::asin (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::acos (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::atan (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::asec (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::acsc (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::acot (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::sinh (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::cosh (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::tanh (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::sech (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::csch (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::coth (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::asinh (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::acosh (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::atanh (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::asech (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::acsch (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::acoth (std::vector< S > a)
- template<typename S >
  std::vector< S > vec::exp (std::vector< S > a)

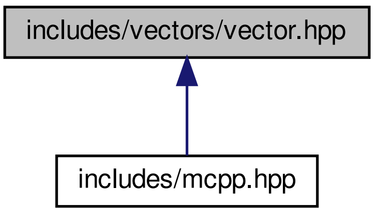
7.7 includes/vectors/vector.hpp File Reference

```
#include "../misc/constants.hpp"
#include <vector>
#include <iostream>
```

Include dependency graph for vector.hpp:



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



Functions

- template<typename S >
  std::vector< S > operator* (std::vector< S > first, std::vector< S > second)

  *vector*vector*

- template<typename S >
  std::vector< S > operator* (S s, std::vector< S > v)

  *var*vector*

- template<typename S >
  std::vector< S > operator* (std::vector< S > v, S s)

vector∗var

- template<typename S >
  std::vector< S > operator+ (std::vector< S > first, std::vector< S > second)

  vector+vector
- template<typename S >
  std::vector< S > operator+ (S s, std::vector< S > v)

  var+vector
- template<typename S >
  std::vector< S > operator+ (std::vector< S > v, S s)

  vector+var
- template<typename S >
  std::vector< S > operator- (std::vector< S > first, std::vector< S > second)

  vector-vector
- template<typename S >
  std::vector< S > operator- (S s, std::vector< S > v)

  var-vector
- template<typename S >
  std::vector< S > operator- (std::vector< S > v, S s)

  vector-var
- template<typename S >
  std::vector< S > operator/ (std::vector< S > first, std::vector< S > second)

  vector/vector
- template<typename S >
  std::vector< S > operator/ (S s, std::vector< S > v)

  var/vector
- template<typename S >
  std::vector< S > operator/ (std::vector< S > v, S s)

  vector/var
- template<typename S >
  std::ostream & operator<< (std::ostream &out, std::vector< S > &other)

  cout << vector
- template<typename S >
  std::vector< S > operator% (std::vector< S > v, S s)

  vectorvar

7.7.1 Function Documentation

7.7.1.1 operator%()

```
template<typename S >
std::vector<S> operator% (
             std::vector< S > v,
             S s )
```

vectorvar

Template Parameters

S	

Parameters

v	
s	

Returns

std::vector<S>

7.7.1.2 operator*() [1/3]

```
template<typename S >
std::vector<S> operator* (
            S s,
            std::vector< S > v )
```

var*vector

Template Parameters

S	

Parameters

s	
v	

Returns

std::vector<S>

7.7.1.3 operator*() [2/3]

```
template<typename S >
std::vector<S> operator* (
            std::vector< S > first,
            std::vector< S > second )
```

vector*vector

Template Parameters

S	

Parameters

first	
second	

Returns

std::vector<S>

7.7.1.4 operator∗() `[3/3]`

```
template<typename S >
std::vector<S> operator* (
            std::vector< S > v,
            S s )
```

vector∗var

Template Parameters

S	

Parameters

v	
s	

Returns

std::vector<S>

7.7.1.5 operator+() `[1/3]`

```
template<typename S >
std::vector<S> operator+ (
            S s,
            std::vector< S > v )
```

var+vector

Template Parameters

S	

Parameters

s	
v	

Returns

std::vector<S>

7.7.1.6 operator+() [2/3]

```
template<typename S >
std::vector<S> operator+ (
            std::vector< S > first,
            std::vector< S > second )
```

vector+vector

Template Parameters

S	

Parameters

first	
second	

Returns

std::vector<S>

7.7.1.7 operator+() [3/3]

```
template<typename S >
std::vector<S> operator+ (
            std::vector< S > v,
            S s )
```

vector+var

Template Parameters

S	

Parameters

v	
s	

Returns

std::vector<S>

7.7.1.8 operator-() [1/3]

```
template<typename S >
std::vector<S> operator- (
            S s,
            std::vector< S > v )
```

var-vector

Template Parameters

S	

Parameters

s	
v	

Returns

std::vector<S>

7.7.1.9 operator-() [2/3]

```
template<typename S >
std::vector<S> operator- (
            std::vector< S > first,
            std::vector< S > second )
```

vector-vector

Template Parameters

S	

Parameters

first	
second	

Returns

std::vector<S>

7.7.1.10 operator-() [3/3]

```
template<typename S >
std::vector<S> operator- (
            std::vector< S > v,
            S s )
```

vector-var

Template Parameters

S	

Parameters

v	
s	

Returns

std::vector<S>

7.7.1.11 operator/() [1/3]

```
template<typename S >
std::vector<S> operator/ (
            S s,
            std::vector< S > v )
```

var/vector

Template Parameters

S	

Parameters

s	
v	

Returns

std::vector<S>

7.7.1.12 operator/() [2/3]

```
template<typename S >
std::vector<S> operator/ (
            std::vector< S > first,
            std::vector< S > second )
```

vector/vector

Template Parameters

S	

Parameters

first	
second	

Returns

std::vector<S>

7.7.1.13 operator/() [3/3]

```
template<typename S >
std::vector<S> operator/ (
            std::vector< S > v,
            S s )
```

vector/var

Template Parameters

S	

Parameters

v	
s	

Returns

std::vector<S>

7.7.1.14 operator<<()

```
template<typename S >
std::ostream& operator<< (
            std::ostream & out,
            std::vector< S > & other )
```

cout << vector

Template Parameters

S	

Parameters

out	
other	

Returns

std::ostream&

7.8 README.md File Reference

Index

_a
    var::matrix< S >::Row, 30
_col
    var::matrix< S >, 28
_i
    var::matrix< S >::Row, 30
_row
    var::matrix< S >, 28

acos
    vec, 10
acosh
    vec, 10
acot
    vec, 10
acoth
    vec, 11
acsc
    vec, 11
acsch
    vec, 11
asec
    vec, 11
asech
    vec, 11
asin
    vec, 11
asinh
    vec, 12
atan
    vec, 12
atanh
    vec, 12

BLACK
    figure.hpp, 36
BLUE
    figure.hpp, 36

check_param
    figure.hpp, 34
col
    var::matrix< S >, 21
col_op
    var::matrix< S >, 21
constants.hpp
    table, 32
cos
    vec, 12
cosh
    vec, 12
cot
    vec, 12
coth
    vec, 13
cross
    vec, 13
csc
    vec, 13
csch
    vec, 13

data
    var::matrix< S >, 28
det
    var::matrix< S >, 21
dot
    vec, 13

exp
    vec, 13

figure
    figure.hpp, 34
figure.hpp
    BLACK, 36
    BLUE, 36
    check_param, 34
    figure, 34
    GREEN, 36
    hline, 34
    plot, 35
    point, 35
    RED, 36
    vline, 35
    WHITE, 36

GREEN
    figure.hpp, 36

hline
    figure.hpp, 34

includes/mcpp.hpp, 31
includes/misc/constants.hpp, 31
includes/plot/figure.hpp, 33
includes/variables/graph.hpp, 37
includes/variables/matrix.hpp, 37
includes/vectors/vec.hpp, 38
includes/vectors/vector.hpp, 40
insert_col

var::matrix< S >, 21, 22
insert_row
    var::matrix< S >, 22
inv
    var::matrix< S >, 23
is_identity
    var::matrix< S >, 23
is_square
    var::matrix< S >, 23

matrix
    var::matrix< S >, 20

operator<<
    var::matrix< S >, 27
    vector.hpp, 48
operator∗
    var::matrix< S >, 23
    vector.hpp, 42, 43
operator+
    var::matrix< S >, 23
    vector.hpp, 43, 44
operator-
    var::matrix< S >, 24
    vector.hpp, 45, 46
operator/
    var::matrix< S >, 24
    vector.hpp, 46, 47
operator%
    vector.hpp, 41
operator[]
    var::matrix< S >, 24
    var::matrix< S >::Row, 30

plot
    figure.hpp, 35
point
    figure.hpp, 35
pow
    vec, 14

README.md, 48
RED
    figure.hpp, 36
resize
    var::matrix< S >, 24
Row
    var::matrix< S >::Row, 30
row
    var::matrix< S >, 24
row_op
    var::matrix< S >, 25
rref
    var::matrix< S >, 25

sec
    vec, 14
sech
    vec, 14
sin
    vec, 14
sinh
    vec, 14
size
    var::matrix< S >, 25
sort_col
    var::matrix< S >, 25
sort_cols
    var::matrix< S >, 26
sort_row
    var::matrix< S >, 26
sort_rows
    var::matrix< S >, 26
sum
    var::matrix< S >, 26

T
    var::matrix< S >, 27
table
    constants.hpp, 32
tan
    vec, 15
tanh
    vec, 15
turn_to
    var::matrix< S >, 27

var, 9
var::matrix< S >, 17
    _col, 28
    _row, 28
    col, 21
    col_op, 21
    data, 28
    det, 21
    insert_col, 21, 22
    insert_row, 22
    inv, 23
    is_identity, 23
    is_square, 23
    matrix, 20
    operator<<, 27
    operator∗, 23
    operator+, 23
    operator-, 24
    operator/, 24
    operator[], 24
    resize, 24
    row, 24
    row_op, 25
    rref, 25
    size, 25
    sort_col, 25
    sort_cols, 26
    sort_row, 26
    sort_rows, 26
    sum, 26
    T, 27

turn_to, 27
var::matrix< S >::Row, 29
_a, 30
_i, 30
operator[], 30
Row, 30
vec, 9
acos, 10
acosh, 10
acot, 10
acoth, 11
acsc, 11
acsch, 11
asec, 11
asech, 11
asin, 11
asinh, 12
atan, 12
atanh, 12
cos, 12
cosh, 12
cot, 12
coth, 13
cross, 13
csc, 13
csch, 13
dot, 13
exp, 13
pow, 14
sec, 14
sech, 14
sin, 14
sinh, 14
tan, 15
tanh, 15
vector.hpp
operator<<, 48
operator∗, 42, 43
operator+, 43, 44
operator-, 45, 46
operator/, 46, 47
operator%, 41
vline
figure.hpp, 35

WHITE
figure.hpp, 36