prog-en-c

Généré par Doxygen 1.8.5

Mercredi Janvier 25 2023 14 :44 :14

Table des matières

Chapitre 1

Index des structures de données

1.1 Structures de données

Liste des structures de données avec une brève description :

Chapitre 2

Index des fichiers

2.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

Chapitre 3

Documentation des structures de données

3.1 Référence de la structure matrix_t

Definition of vector_t type.

`#include <matrix.h>`

Champs de données

- unsigned row_size
- unsigned col_size
- double * values

3.1.1 Description détaillée

Definition of vector_t type.

3.1.2 Documentation des champs

3.1.2.1 unsigned matrix_t : :col_size

the column number of the matrix

3.1.2.2 unsigned matrix_t : :row_size

the row number of the matrix

3.1.2.3 double* matrix_t : :values

a pointer to the array of entries

La documentation de cette structure a été générée à partir du fichier suivant :
- /net/r/jroyet/Documents/S6/PG120/include/matrix.h

3.2 Référence de la structure vector_t

Definition of vector_t type.

`#include <vector.h>`

Champs de données

– unsigned size
– double ∗ values

3.2.1 Description détaillée

Definition of vector_t type.

3.2.2 Documentation des champs

3.2.2.1 unsigned vector_t : :size

the size of the vector

3.2.2.2 double∗ vector_t : :values

a pointer to the array of entries

La documentation de cette structure a été générée à partir du fichier suivant :
– /net/r/jroyet/Documents/S6/PG120/include/vector.h

Chapitre 4

Documentation des fichiers

4.1 Référence du fichier /net/r/jroyet/Documents/S6/PG120/include/matrix.h

A small matrix library.

`#include <vector.h>`

Structures de données

- struct matrix_t
  Definition of vector_t type.

Fonctions

- matrix_t make_matrix (unsigned r, unsigned c)
  Matrix factory.
- void free_matrix (matrix_t m)
  Matrix factory.
- double m_get (matrix_t m, unsigned i, unsigned j)
  Matrix factory.
- void m_set (matrix_t m, unsigned i, unsigned j, double x)
  Matrix factory.
- void m_add (matrix_t m, unsigned i, unsigned j, double x)
  Matrix factory.
- void peek_row (const matrix_t *m, unsigned i, vector_t *v)
- vector_t apply (matrix_t m, vector_t v)
- void m_print (matrix_t m)
  Matrix factory.
- matrix_t make_rotation_matrix (double alpha)
  Matrix factory.

4.1.1 Description détaillée

A small matrix library. This library provide a few functions for the computation with matrixs.

4.1.2 Documentation des fonctions

4.1.2.1 vector_t apply (matrix_t *m,* vector_t *v*)

4.1.2.2 void free_matrix (matrix_t *m*)

Matrix factory.

This function delete in memory the dynamic array of the matrix stucture ;

Paramètres

|>p0.15|p0.805|

m, : the matrix you want to free, must not be null.

4.1.2.3 void m_add (matrix_t *m,* unsigned *i,* unsigned *j,* double *x*)

Matrix factory.

This function insert your value x at the index i,j of the matrix ;

Paramètres

|>p0.15|p0.805|

m,i,j,x, : the vector v you want to add value and i the index and x double you want to set.

4.1.2.4 double m_get (matrix_t *m,* unsigned *i,* unsigned *j*)

Matrix factory.

This function return you the value contained in at the i index of his value array ;

Paramètres

|>p0.15|p0.805|

m,i,j, : the matrix m you want to get value (not null) and i the index of row (not null), j the index of column (not null).

Renvoie

a double, corresponding to the value contained at i,j index in the array.

4.1.2.5 void m_print (matrix_t *m*)

Matrix factory.

This function print the matrix.

Paramètres

|>p0.15|p0.805|

m, : matrix you want to display.

4.1.2.6 void m_set (matrix_t *m,* unsigned *i,* unsigned *j,* double *x*)

Matrix factory.

This function set at the value indexed i,j, the value x you passed in parameter ;

Paramètres

|>p0.15|p0.805|

m,i,j,x, : the vector v you want to set value (not null) and i the index (not null) and x double you want to set.

4.1.2.7 matrix_t make_matrix (unsigned *r,* unsigned *c*)

Matrix factory.

This function builds a null matrix of size s ; the array is dynamically allocated and must be freed with the function free_matrix

Paramètres

>p0.15	p0.805
r,c, : number of row (r) and columns (c) must be >0

Renvoie

a matrix of size s initialized to 0

Voir également

free_matrix

4.1.2.8 matrix_t make_rotation_matrix (double *alpha*)

Matrix factory.

This function return your matrix rotated by alpha degree ;

Paramètres

>p0.15	p0.805
alpha, : degree you want to rotate your matrix.

Renvoie

your matrix.

4.1.2.9 void peek_row (const matrix_t * *m,* unsigned *i,* vector_t * *v*)

4.2 Référence du fichier /net/r/jroyet/Documents/S6/PG120/include/vector.h

A small vector library.

Structures de données

- struct vector_t
  Definition of vector_t type.

Fonctions

- vector_t make_vector (unsigned s)
  Vector factory.
- void free_vector (vector_t v)
  Vector factory.
- double v_get (vector_t v, unsigned i)
  Vector factory.
- void v_set (vector_t v, unsigned i, double x)
  Vector factory.
- void v_add (vector_t v, unsigned i, double x)
  Vector factory.

- double scalar_prod (vector_t v1, vector_t v2)
  Vector factory.
- void v_print_h (vector_t v)
  Vector factory.
- void v_print (vector_t v)
  Vector factory.

4.2.1 Description détaillée

A small vector library. This library provide a few functions for the computation with vectors.

4.2.2 Documentation des fonctions

4.2.2.1 void free_vector (vector_t *v*)

Vector factory.

This function delete in memory the dynamic array of the vector stucture ;

Paramètres

>p0.15	p0.805
v, :	the vector you want to free, must not be null.

4.2.2.2 vector_t make_vector (unsigned *s*)

Vector factory.

This function builds a null vector of size s ; the array is dynamically allocated and must be freed with the function free_vector

Paramètres

>p0.15	p0.805
s, :	the size of the vector must be >0

Renvoie

a vector of size s initialized to 0

Voir également

free_vector

4.2.2.3 double scalar_prod (vector_t *v1,* vector_t *v2*)

Vector factory.

This function do the scalar product of two vectors ;

Paramètres

>p0.15	p0.805
v1,v2, :	two vector you want to multiply (not null)

Renvoie

the product of scalar product.

4.2.2.4 void v_add (vector_t *v,* unsigned *i,* double *x*)

Vector factory.

This function insert your value v at the index i of the vector ;

Paramètres

|>p0.15|p0.805|

v,i,x, : the vector v you want to add value and i the index and x double you want to set.

4.2.2.5 double v_get (vector_t *v,* unsigned *i*)

Vector factory.

This function return you the value contained in at the i index of his value array ;

Paramètres

|>p0.15|p0.805|

v,i, : the vector v you want to get value (not null) and i the index (not null).

Renvoie

a double, corresponding to the value contained at i index in the array.

4.2.2.6 void v_print (vector_t *v*)

Vector factory.

This function print the vector into a vertical array.

Paramètres

|>p0.15|p0.805|

v, : vector you want to display.

4.2.2.7 void v_print_h (vector_t *v*)

Vector factory.

This function print the vector into an horizontal array.

Paramètres

|>p0.15|p0.805|

v, : vector you want to display.

4.2.2.8 void v_set (vector_t *v,* unsigned *i,* double *x*)

Vector factory.

This function set at the value indexed i, the value x you passed in parameter ;

Paramètres

|>p0.15|p0.805|

v,i,x, : the vector v you want to set value (not null) and i the index (not null) and x double you want to set.