# FOS++

0.1

Generated by Doxygen 1.8.11

# Contents

# Chapter 1

# Class Index

## 1.1 Class List

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

# Chapter 2

# File Index

## 2.1 File List

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

# Chapter 3

# Class Documentation

## 3.1 FOS< T > Class Template Reference

The main FOS algorithim.

```
#include <fos.h>
```

### Public Member Functions

- **FOS** (Eigen::Matrix< T, Eigen::Dynamic, Eigen::Dynamic > x, Eigen::Matrix< T, Eigen::Dynamic, 1 > y)
- void **Algorithm** ()

### 3.1.1 Detailed Description

**template<typename T>**
**class FOS< T >**

The main FOS algorithim.

Definition at line 23 of file fos.h.

# Chapter 4

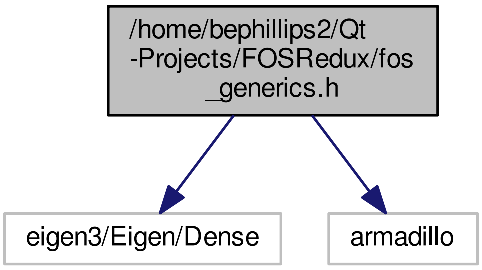
# File Documentation

## 4.1 /home/bephillips2/Qt-Projects/FOSRedux/fos_generics.h File Reference

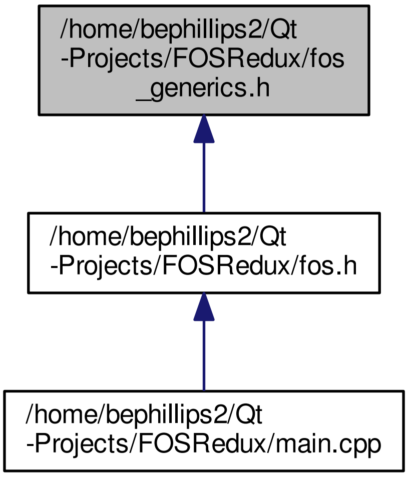
Generic linear algebra functions.

```
#include <eigen3/Eigen/Dense>
#include <armadillo>
```

Include dependency graph for fos_generics.h:



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


/home/bephillips2/Qt -Projects/FOSRedux/fos _generics.h
/home/bephillips2/Qt -Projects/FOSRedux/fos.h
/home/bephillips2/Qt -Projects/FOSRedux/main.cpp

## Functions

- template<typename T >
  T CSV2Eigen (std::string file_path)

  *Read a .csv file into an Eigen Matrix.*
- void **removeRow** (Eigen::MatrixXd &matrix, unsigned int rowToRemove)
- void **removeColumn** (Eigen::MatrixXd &matrix, unsigned int colToRemove)
- template<typename T >
  T StdDev (const Eigen::Matrix< T, Eigen::Dynamic, Eigen::Dynamic > &mat)

  *Compute the standard deviation of a matrix.*
- template<typename T >
  void Normalize (Eigen::Matrix< T, Eigen::Dynamic, Eigen::Dynamic > &mat)

  *Set the mean of a matrix to 0 and the standard deviation to 1.*
- template<typename T >
  void Normalize (Eigen::Matrix< T, Eigen::Dynamic, 1 > &mat)

  *Set the mean of a vector to 0 and the standard deviation to 1.*

### 4.1.1 Detailed Description

Generic linear algebra functions.

### 4.1.2 Function Documentation

#### 4.1.2.1 template<typename T > T CSV2Eigen ( std::string *file_path* )

Read a .csv file into an Eigen Matrix.

Files must -not- have header information of any kind (e.g. row/col labels etc. ) Rows are determined by line breakers, columns are determined by comma-delimiter.

**Parameters**

| | |
|---|---|
| *file_path* | The (hard) path to the data file. |

**Returns**

An Eigen matrix with rows/cols determined by data file.

Definition at line 37 of file fos_generics.h.

#### 4.1.2.2 template< typename T > void Normalize ( Eigen::Matrix< T, Eigen::Dynamic, Eigen::Dynamic > & *mat* )

Set the mean of a matrix to 0 and the standard deviation to 1.

Note this function is done in place, that is the input matrix is modified.

**Parameters**

| | |
|---|---|
| *mat* | An n x m matrix to be normalized. |

Definition at line 105 of file fos_generics.h.

#### 4.1.2.3 template< typename T > void Normalize ( Eigen::Matrix< T, Eigen::Dynamic, 1 > & *mat* )

Set the mean of a vector to 0 and the standard deviation to 1.

Note this function is done in place.

**Parameters**

| | |
|---|---|
| *mat* | An n x 1 vector to be normalized. |

Definition at line 123 of file fos_generics.h.

#### 4.1.2.4 template< typename T > T StdDev ( const Eigen::Matrix< T, Eigen::Dynamic, Eigen::Dynamic > & *mat* )

Compute the standard deviation of a matrix.

**Parameters**

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

**Returns**

Standard deviation of the matrix

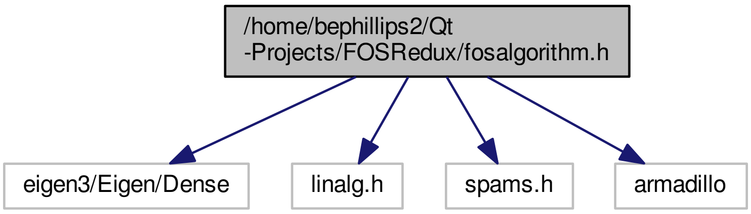
Definition at line 88 of file fos_generics.h.

## 4.2 /home/bephillips2/Qt-Projects/FOSRedux/fosalgorithm.h File Reference

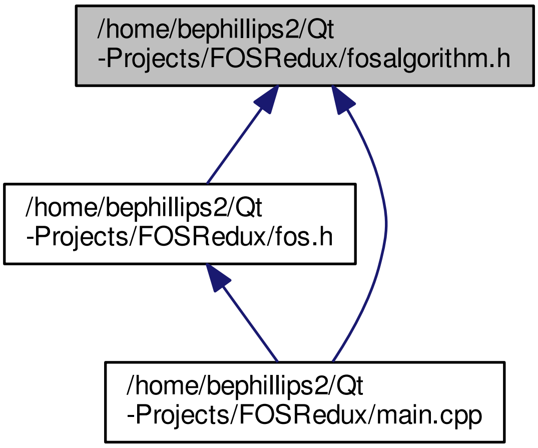
Functions that provide an interface between Eigen and Spams linear algebra libraries.

```
#include <eigen3/Eigen/Dense>
#include "linalg.h"
#include "spams.h"
#include <armadillo>
```

Include dependency graph for fosalgorithm.h:



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



### Functions

- template<typename T , uint m, uint n>
  Eigen::Matrix< T, m, n > Spams2EigenMat (const Matrix< T > ∗spams_mat)

  *Convert a const- Spams Matrix to an Eigen::Matrix.*

- template<typename T , uint m, uint n>
  Eigen::Matrix< T, m, n > Spams2EigenMat (Matrix< T > *spams_mat)

  *Convert a Spams Matrix to an Eigen::Matrix whose rows and cols are known at compile time.*
- template<typename T >
  Eigen::Matrix< T, Eigen::Dynamic, Eigen::Dynamic > Spams2EigenMat (Matrix< T > *spams_mat)

  *Convert a Spams Matrix to an Eigen::Matrix whose rows and cols are assigned at run time.*
- template<typename T , uint m, uint n>
  Matrix< T > * Eigen2SpamsMat (const Eigen::Matrix< T, n, m > &eigen_mat)

  *Get a spams Matrix from an Eigen::Matrix whose dimensions are know at compile time.*
- template<typename T >
  Matrix< T > * Eigen2SpamsMat (const Eigen::Matrix< T, Eigen::Dynamic, Eigen::Dynamic > &eigen_mat)

  *Get a spams Matrix from an Eigen::Matrix whose dimensions are determined at run time.*
- template<typename T , uint m, uint n>
  AbstractMatrixB< T > **Eigen2SpamsAbstractMatB** (const Eigen::Matrix< T, n, m > &eigen_mat)
- char * str_to_c_ptr (std::string &str)

  *Translate a std::string into a pointer to a char array.*
- template<typename T >
  Matrix< T > * internal::FistaFlat (Matrix< T > *Y, Matrix< T > *X, Matrix< T > *Omega_0, const T lambda_1)

  *Performed _fistaFlat on Spams objects, returning parameters useful for the FOS algorithim.*
- template<typename T >
  Eigen::Matrix< T, Eigen::Dynamic, Eigen::Dynamic > FistaFlat (Eigen::Matrix< T, Eigen::Dynamic, Eigen↵::Dynamic > Y, Eigen::Matrix< T, Eigen::Dynamic, Eigen::Dynamic > X, Eigen::Matrix< T, Eigen::Dynamic, Eigen::Dynamic > Omega_0, const T lambda_1)

  *Performed fistaFlat on Eigen objects, returning parameters useful for the FOS algorithim.*

### 4.2.1 Detailed Description

Functions that provide an interface between Eigen and Spams linear algebra libraries.

### 4.2.2 Function Documentation

#### 4.2.2.1 template<typename T , uint m, uint n> Matrix<T>* Eigen2SpamsMat ( const Eigen::Matrix< T, n, m > & *eigen_mat* )

Get a spams Matrix from an Eigen::Matrix whose dimensions are know at compile time.

**Parameters**

| | |
|---|---|
| *eigen_mat* | The Eigen::Matrix to be copied. |

**Returns**

A new Spams Matrix ( in pointer form ).

Definition at line 86 of file fosalgorithm.h.

#### 4.2.2.2 template<typename T > Matrix<T>* Eigen2SpamsMat ( const Eigen::Matrix< T, Eigen::Dynamic, Eigen::Dynamic > & *eigen_mat* )

Get a spams Matrix from an Eigen::Matrix whose dimensions are determined at run time.

**Parameters**

| | |
|---|---|
| *eigen_mat* | The Eigen::Matrix to be copied. |

**Returns**

A new Spams Matrix ( in pointer form ).

Definition at line 113 of file fosalgorithm.h.

#### 4.2.2.3 template$<$typename T $>$ Matrix$<$T$>*$ internal::FistaFlat ( Matrix$<$ T $>$ $*$ *Y,* Matrix$<$ T $>$ $*$ *X,* Matrix$<$ T $>$ $*$ *Omega_0,* const T *lambda_1* )

Performed _fistaFlat on Spams objects, returning parameters useful for the FOS algorithim.

**Parameters**

| | |
|---|---|
| *Y* | A n x 1 vector |
| *X* | An n x m desgin matrix |
| *Omega*↩<br>*_0* | An n x 1 vector of initial guesses ( probably ) |
| *lambda*↩<br>*_1* | Regularization parameter |

**Returns**

Omega, a 1 x n matrix

Definition at line 179 of file fosalgorithm.h.

#### 4.2.2.4 template$<$typename T $>$ Eigen::Matrix$<$ T, Eigen::Dynamic, Eigen::Dynamic$>$ FistaFlat ( Eigen::Matrix$<$ T, Eigen::Dynamic, Eigen::Dynamic $>$ *Y,* Eigen::Matrix$<$ T, Eigen::Dynamic, Eigen::Dynamic $>$ *X,* Eigen::Matrix$<$ T, Eigen::Dynamic, Eigen::Dynamic $>$ *Omega_0,* const T *lambda_1* )

Performed fistaFlat on Eigen objects, returning parameters useful for the FOS algorithim.

**Parameters**

| | |
|---|---|
| *Y* | A n x 1 vector |
| *X* | An n x m desgin matrix |
| *Omega*↩<br>*_0* | An n x 1 vector of initial guesses |
| *lambda*↩<br>*_1* | Regularization parameter |

**Returns**

Omega, a 1 x n matrix

Definition at line 278 of file fosalgorithm.h.

#### 4.2.2.5 template$<$typename T , uint m, uint n$>$ Eigen::Matrix$<$ T, m, n $>$ Spams2EigenMat ( const Matrix$<$ T $>$ $*$ *spams_mat* )

Convert a const- Spams Matrix to an Eigen::Matrix.

**Parameters**

| | |
|---|---|
| *spams_mat* | Spams Matrix pointer to be translated. |

**Returns**

A new Eigen::Matrix with dimensions determined by the Spams Matrix.

Definition at line 35 of file fosalgorithm.h.

#### 4.2.2.6 template$<$typename T , uint m, uint n$>$ Eigen::Matrix$<$ T, m, n $>$ Spams2EigenMat ( Matrix$<$ T $>$ $*$ *spams_mat* )

Convert a Spams Matrix to an Eigen::Matrix whose rows and cols are known at compile time.

**Parameters**

| | |
|---|---|
| *spams_mat* | Spams Matrix pointer to be translated. |

**Returns**

A new Eigen::Matrix with dimensions determined by the Spams Matrix.

Definition at line 50 of file fosalgorithm.h.

#### 4.2.2.7 template$<$typename T $>$ Eigen::Matrix$<$ T, Eigen::Dynamic, Eigen::Dynamic $>$ Spams2EigenMat ( Matrix$<$ T $>$ $*$ *spams_mat* )

Convert a Spams Matrix to an Eigen::Matrix whose rows and cols are assigned at run time.

**Parameters**

| | |
|---|---|
| *spams_mat* | Spams Matrix pointer to be translated. |

**Returns**

A new Eigen::Matrix with dimensions determined by the Spams Matrix.

Definition at line 67 of file fosalgorithm.h.

#### 4.2.2.8 char∗ str_to_c_ptr ( std::string & *str* )

Translate a std::string into a pointer to a char array.

Used with Spams 'print' functions.

**Parameters**

| | |
|---|---|
| *str* | String to be transformed |

**Returns**

char∗ populated with data in str and null terminator, Note that the char∗ will need to be deleted later

Definition at line 147 of file fosalgorithm.h.

# Index