Lanczos

Generated by Doxygen 1.5.7

Wed Sep 8 16:30:01 2010

Contents

1 Lanczos Itertive Method

The Lanczos Iterative method is an alternative to the Dimer method. It has been included in eOn. To compile Lanczos two extra libraries are required GSL (GNU Scientific Library) and Blitz++. These can be downloaded at: http://www.gnu.org/software/gsl/ and http://www.oonumerics.org/blitz/download/ (or check out at

```
export CVSROOT=:pserver:anonymous@blitz.cvs.sourceforge.net:/cvsroot/blitz
cvs login
cvs -z3 checkout blitz </cc>
```

). Because of the inconvenience, Lanczos is not compiled by default. To include it, the client must be with the environment variable WITH_LANCZOS defined. In bash:

```
export WITH_LANCZOS=1
```

2 Module Index

2.1 Modules

Here is a list of all modules:

3 Namespace Index

3.1 Namespace List

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

4 Class Index

4.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

5 Class Index

5.1 Class List

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

6 File Index

6.1 File List

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

7 Module Documentation

7.1 Wrappers between blitz and gsl.

Functions

- blitz::Array< double, 1 > gradient_scanning::wrapToBlitz (gsl_vector ∗const vgsl)

  Wrap gsl_vector into blitz::Array.

- blitz::Array< double, 2 > gradient_scanning::wrapToBlitz (gsl_matrix ∗const mgsl)

  Wrap gsl_matrix into blitz::Array.

- gsl_vector gradient_scanning::wrapToGsl (blitz::Array< double, 1 > &vblitz)

Wrap blitz::Array into gsl_vector.

- gsl_matrix gradient_scanning::wrapToGsl (blitz::Array< double, 2 > &mblitz)

  Wrap blitz::Array into gsl_matrix.

- gsl_vector const gradient_scanning::fastWrapToGsl (blitz::Array< double, 1 > const &vblitz)
- gsl_matrix const gradient_scanning::fastWrapToGsl (blitz::Array< double, 2 > const &mblitz)

7.1.1 Detailed Description

Make gsl_vector and gsl_matrix visible to blitz and vice versa.

7.1.2 Fast Wrapping for const Array.

When wrapping, the memory in blitz::Array is shared with the gsl_vector (or gsl_matrix). Though marked as constant it is still possible change the data through pointer gsl_vector.data. Doing this would affect the blitz::Array declared as constant. It is an error. A safer solution is to copy the blitz beforehand and use the standard wrapToBlitz function.

7.1.3 Function Documentation

7.1.3.1 gsl_matrix const gradient_scanning::fastWrapToGsl (blitz::Array< double, 2 > const & *mblitz*)

Wrap blitz::Array into gsl_matrix.

See also:

Fast Wrapping for const Array.

7.1.3.2 gsl_vector const gradient_scanning::fastWrapToGsl (blitz::Array< double, 1 > const & *vblitz*)

Wrap blitz::Array into gsl_vector.

See also:

Fast Wrapping for const Array.

8 Namespace Documentation

8.1 gradient_scanning Namespace Reference

ConjugateGradient, SaddlePoint searches.

Namespaces

- namespace random

  random number generator.

Classes

- class Lanczos

  Compute the lowest eigenvalue and eigenvector.

- class GradientObject

  To compute gradient. To compute gradient from a function (GradientFunction), or from a member function (GradientTemplate).

- class GradientTemplate

  When gradient is computed by member function.

Typedefs

- typedef void(∗ GradientFunction)(blitz::Array< double, 1 > &coordinates, blitz::Array< double, 1 > &gradient)

Functions

- blitz::Array< double, 1 > wrapToBlitz (gsl_vector ∗const vgsl)

  Wrap gsl_vector into blitz::Array.

- blitz::Array< double, 2 > wrapToBlitz (gsl_matrix ∗const mgsl)

  Wrap gsl_matrix into blitz::Array.

- gsl_vector wrapToGsl (blitz::Array< double, 1 > &vblitz)

  Wrap blitz::Array into gsl_vector.

- gsl_matrix wrapToGsl (blitz::Array< double, 2 > &mblitz)

  Wrap blitz::Array into gsl_matrix.

- gsl_vector const fastWrapToGsl (blitz::Array< double, 1 > const &vblitz)
- gsl_matrix const fastWrapToGsl (blitz::Array< double, 2 > const &mblitz)

8.1.1 Detailed Description

ConjugateGradient, SaddlePoint searches.

Tools to scan a surface. The namespace contains a minimiser: class ConjugateGradient, and a class to search for saddle points: SaddlePoint. The namespace contains also other classes and functions needed by the two latter.

Class Lanczos performs the Lanczos iterative method and is used by SaddlePoint. Files gradient_-scanning_tools.h and gradient_scanning_tools.cpp contains basic tools used by all major classes. In particular, it contains types (GradientFunction) and classes (GradientObject ...) used to manage the functionss computing the gradient.

Tools in this namespace requires library blitz++ which can be downloaded at `http://www.oonumerics.org/blitz` . Class Lanczos and SaddlePoint requires library GSL (GNU Scientific Library) to diagonalise matrices. GSL can be downloaded at `http://www.gnu.org/software/gsl` .

See also:

ConjugateGradient, SaddlePoint.

Note:

Requires library blitz++ and GSL.

8.1.2 Typedef Documentation

8.1.2.1 typedef void($*$ gradient_scanning::GradientFunction)(blitz::Array$<$ double, 1 $>$ &coordinates, blitz::Array$<$ double, 1 $>$ &gradient)

Function to compute the gradient. The user enters an array of *coordinates*. The function returns an array containing the *gradient*. The parameter *coordinates* is not constant as it may be changed by the function to apply constraints (bond length constraints, periodic boundaries, etc...)

Parameters:

$\leftrightarrow$ ***coordinates*** Coordinates. Return coordinates with applied constraints is any.

$\rightarrow$ ***gradient*** Gradient at *coordinates*.

8.2 gradient_scanning::random Namespace Reference

random number generator.

Functions

- unsigned int seedWithTime ()

  Seed generator.

Variables

- ranlib::Normal$<$ double $>$ normal

  Gaussian distribution of mean zero and variance 1.

- ranlib::Uniform$<$ double $>$ uniform

  Uniform distribution in interval [0;1].

8.2.1 Detailed Description

random number generator.

The class is just a wrapper for random number generators defined in Blitz++ (`http://www.oonumerics.org/blitz/`) library

See also:

random/uniform.h and random/normal.h

9 Class Documentation

9.1 gradient_scanning::GradientObject Class Reference

To compute gradient. To compute gradient from a function (GradientFunction), or from a member function (GradientTemplate).

Inherited by gradient_scanning::GradientTemplate< T >.

Public Member Functions

- virtual void compute (blitz::Array< double, 1 > &coordinates, blitz::Array< double, 1 > &gradient)

9.1.1 Detailed Description

To compute gradient. To compute gradient from a function (GradientFunction), or from a member function (GradientTemplate).

See also:

GradientFunction, GradientTemplate

9.1.2 Member Function Documentation

9.1.2.1 void GradientObject::compute (blitz::Array< double, 1 > & *coordinates*, blitz::Array< double, 1 > & *gradient*) `[virtual]`

Compute gradient.

Reimplemented in gradient_scanning::GradientTemplate< T >.

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

- /Users/berthet/eon/client/Lanczos/tools.hpp
- /Users/berthet/eon/client/Lanczos/tools.cpp

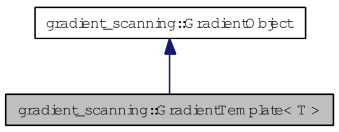
9.2 gradient_scanning::GradientTemplate< T > Class Template Reference

When gradient is computed by member function.

`#include <tools.hpp>`

Inherits gradient_scanning::GradientObject.

Collaboration diagram for gradient_scanning::GradientTemplate< T >:


gradient_scanning::GradientObject
gradient_scanning::GradientTemplate< T >

Public Member Functions

- void compute (blitz::Array< double, 1 > &coordinates, blitz::Array< double, 1 > &gradient)

9.2.1 Detailed Description

template<class T> class gradient_scanning::GradientTemplate< T >

When gradient is computed by member function.

9.2.2 Member Function Documentation

9.2.2.1 template<class T > void gradient_scanning::GradientTemplate< T >::compute (blitz::Array< double, 1 > & *coordinates*, blitz::Array< double, 1 > & *gradient*) `[inline, virtual]`

Compute gradient.

Reimplemented from gradient_scanning::GradientObject.

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

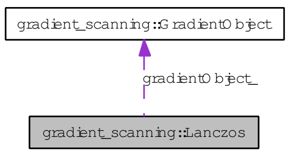
- /Users/berthet/eon/client/Lanczos/tools.hpp

9.3 gradient_scanning::Lanczos Class Reference

Compute the lowest eigenvalue and eigenvector.

Inherited by Lanczos `[private]`.

Collaboration diagram for gradient_scanning::Lanczos:



Public Types

- enum Status {
  NOTHING = 0, CONVERGED = 1, EXCEED_ITERATION = 2, FULL_MATRIX = 4,
  UNCOMPLETE = 8, RUNNING = 32, STARTING = 64 }
- enum Initial { RANDOM, GRADIENT, USER, PREVIOUS }

Public Member Functions

- void clear ()

  Delete data about past Lanczos Iterative Methods performed.

- double convergence () const

Convergence reached by last Lanczos.

- double eigenvalue () const

  Lowest eigenvalue for last call to minimumMode().

- blitz::Array< double, 1 > eigenvector () const

  Lowest mode's eigenvector for last call to minimumMode().

- int iteration () const

  Number of iterations used by last Lanczos.

- Status minimumMode (GradientFunction function, blitz::Array< double, 1 > const &coordinates, double &eigenvalue, blitz::Array< double, 1 > &eigenvector, blitz::Array< double, 1 > &gradient)
- Status minimumMode (GradientObject &object, blitz::Array< double, 1 > const &coordinates, double &eigenvalue, blitz::Array< double, 1 > &eigenvector, blitz::Array< double, 1 > &gradient)
- bool getReport () const

  Tell if a final report is to be printed.

- void setReport (bool turnon)

  Tell if a final report is to be printed.

- Status status () const
- Lanczos const & operator= (Lanczos const &lanczos)

  - double getConvergenceLimit () const

  - double getFiniteDifference () const

  - Initial getInitial () const

  - int getIterationLimit () const

  - Status getWarnWhen () const

9.3.1 Detailed Description

Compute the lowest eigenvalue and eigenvector.

9.3.2 Hessian Minimum Mode

Compute the hessian lowest eigenvalue from function computing gradient by using finite differences.

See also:

minimumMode(), gradient_scanning::GradientFunction.

9.3.3 Reference

Comparison of methods fo finding the saddle points without knowledge of the final sate. R.A. Olsen, G.J. Kroes et al. J. Chem. Physics, (2004), Vol. **121**, n 20.

9.3.4 Member Enumeration Documentation

9.3.4.1 enum gradient_scanning::Lanczos::Initial

Specify how to start the Lanczos iterative algorithm.

See also:

setInitial() and minimumMode()

Enumerator:

RANDOM Start with a random vector and perform at leat two iterations.

GRADIENT Start with gradient vector and perform at leat two iterations.

USER Use *eigenvalue* and *eigenvector* provided by user.

PREVIOUS Use eigenvalue and eigenvector returned by last call to minimumMode(). If not available default to RANDOM.

9.3.4.2 enum gradient_scanning::Lanczos::Status

Status of the Lanczos process. May represent the state of the process both while running or when stopped.

Note:

Status UNCOMPLETE is not necessarily an error.

See also:

status()

Enumerator:

NOTHING Before anything is run.

CONVERGED Lowest eigenvalue has converged (see setTolerance()).

EXCEED_ITERATION Maximum iterations reached (see setIterationLimit()).

FULL_MATRIX The full Matrix has been computed.

UNCOMPLETE Could not complete iterative process. This happens when the initial vector is normal to some eigenvectors.

RUNNING Building the lanczos matrix.

STARTING First call the gradient function.

9.3.5 Member Function Documentation

9.3.5.1 double Lanczos::getConvergenceLimit () const

Parameter controling when the iteration process stops. The iteration process stops when $|\lambda_i - \lambda_{i-1}| < tolerance\_$ where λ is the smallest eigenvalue.

See also:

getMaximumIterations() and setMaximumIterations()

9.3.5.2 double Lanczos::getFiniteDifference () const

Step to calculate derivatives. To calculate the second derivatives of the gradient, a finite difference approximation is used: $\mathbf{g}' = \frac{\mathbf{g}(\mathbf{x}+\Delta\mathbf{x})-\mathbf{g}(\mathbf{x})}{|\Delta\mathbf{x}|}$. The functions getFiniteDifference() and setFiniteDifference() allows to read and set the step $|\Delta\mathbf{x}|$.

9.3.5.3 Lanczos::Initial Lanczos::getInitial () const

Tell how to initialise the Lanczos iterative algorithm. Lanczos needs to initial vector to start the iterative process. The convergence of the process is tested by comparing the lowest eigenvalues found by two successive iterations. Initial controls how the initial vector is chosen and to what value is compared the first eigenvalue.

9.3.5.4 int Lanczos::getIterationLimit () const

Maximum number of iterations.

9.3.5.5 Lanczos::Status Lanczos::getWarnWhen () const

Warning when a certain event occurs. Here you may specify a certain number of events, you want to be warned about when they occurs. The default value is NOTHING which means that no warning will occur. Events may be combined with operator `bitor`. Example:

```
warnWhen(EXCEED_ITERATIONS | EIGENVECTOR);
```

See also:

status().

9.3.5.6 Lanczos::Status Lanczos::minimumMode (GradientObject & *object*, blitz::Array< double, 1 > const & *coordinates*, double & *eigenvalue*, blitz::Array< double, 1 > & *eigenvector*, blitz::Array< double, 1 > & *gradient*)

Hessian lowest eigenvalue and corresponding eigenvector.

See also:

minimumMode().

9.3.5.7 Lanczos::Status Lanczos::minimumMode (GradientFunction *function*, blitz::Array< double, 1 > const & *coordinates*, double & *eigenvalue*, blitz::Array< double, 1 > & *eigenvector*, blitz::Array< double, 1 > & *gradient*)

Hessian lowest eigenvalue and corresponding eigenvector.

Parameters:

$\leftarrow$ ***function*** Function to comput e the gradient.

$\leftarrow$ ***coordinates*** Coordinates at which to compute the minimum mode.

$\leftrightarrow$ ***eigenvalue*** Output lowest eigenvalue. For input depends on getInitial().

$\leftrightarrow$ ***eigenvector*** Output eigenvector of the lowest eigenvalue. For input depends on getInitial().

$\rightarrow$ ***gradient*** Gradient at *coordinates*.

Returns:

The status (see status()).

Note:

Input values in *eigenvalue* and *eigenvector* are used only if getInitial() is USER.
When returned status is UNCOMPLETE, the function returns with the best estimation it has of the lowest eigenvalue and eigenvector. When the returned status is UNCOMPLETE because the initial vector was an eigenvector, the function returns the eigenvector (i.e. initial vector) and its eigenvalue.

Precondition:

If getInitial() is USER the norm of *eigenvector* must not be zero.

9.3.5.8 Lanczos const & Lanczos::operator= (Lanczos const & *lanczos*)

Copy parameters. Copy parameters controling how the Lanczos Iterative method is performed. Note that it does does copy the results displayed by *lanczos*. In other word, results produced by iteration(), eigenvalue(), etc ... will remain different.

9.3.5.9 Lanczos::Status Lanczos::status () const

Status of the Lanczos process. When the lanczos is not running, status indicates either NOT_RUNNING (before any call to minimumMode()) or why the last minimumMode() execution has ended (e.g. CONVERGED). Function status() may also be used while minimumMode() is running, during a call back to the function computing the gradient (see parameter *gradient* in minimumMode(GradientFunction, ...) and gradient in minimumMode(GradientObject&, ...)). In this case status() indicates where the minimisation is currently.

See also:

Status.

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

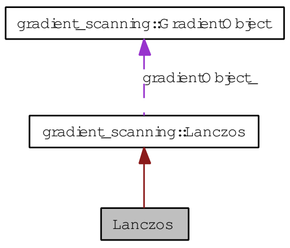
- /Users/berthet/eon/client/Lanczos/lanczos.hpp
- /Users/berthet/eon/client/Lanczos/lanczos.cpp

9.4 Lanczos Class Reference

```
#include <lanczos_for_eon.hpp>
```

Inherits gradient_scanning::Lanczos.

Collaboration diagram for Lanczos:



Public Member Functions

- Lanczos (Matter ∗const, Parameters ∗parameters)
- virtual void startNewSearchAndCompute (Matter const ∗matter, double ∗)

  Destructor.

- virtual void moveAndCompute (Matter const ∗matter)

  Execute Lanczos. It uses last eigenvector to start Lanczos. Use in subsequent step of a saddle point search.

- virtual double returnLowestEigenmode (double ∗result)

9.4.1 Detailed Description

Lanczos class compatible with EON.

9.4.2 Constructor & Destructor Documentation

9.4.2.1 Lanczos::Lanczos (Matter ∗ const, Parameters ∗ *parameters*)

Constructor.

Parameters:

← ∗***parameters*** Pointer to the Parameter object containing the runtime parameters.

9.4.3 Member Function Documentation

9.4.3.1 double Lanczos::returnLowestEigenmode (double ∗ *result*) `[virtual]`

Lowest eigenvalue and corresponding eigenvector.

Parameters:

← ***result*** Pointer to array to store the eigenvector. The length of the vector is $3 \times$ Number of Movable Atoms.

Returns:

Eigenvalue

9.4.3.2 void Lanczos::startNewSearchAndCompute (Matter const ∗ *matter*, double ∗) `[virtual]`

Destructor.

Execute Lanczos. It uses gradient to start lanczos. Use at the beggining of a saddle point search.

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

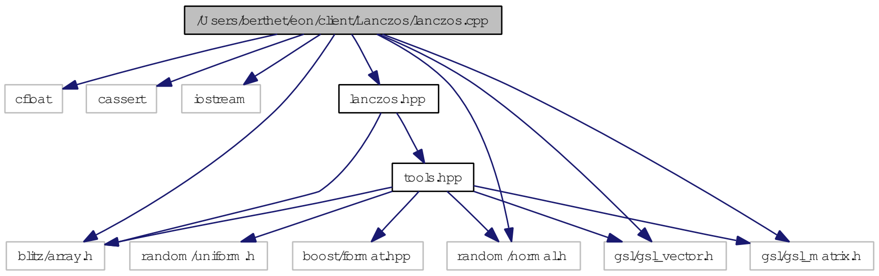
- /Users/berthet/eon/client/Lanczos/lanczos_for_eon.hpp
- /Users/berthet/eon/client/Lanczos/lanczos_for_eon.cpp

10 File Documentation

10.1 /Users/berthet/eon/client/Lanczos/lanczos.cpp File Reference

```
#include <cfloat>
#include <cassert>
#include <iostream>
#include <blitz/array.h>
#include <random/normal.h>
#include <gsl/gsl_vector.h>
#include <gsl/gsl_matrix.h>
#include "gsl_eigen.h"
#include "lanczos.hpp"
```

Include dependency graph for lanczos.cpp:



10.1.1 Detailed Description

Class Lanczos.

Author:

Jean Claude C. Berthet

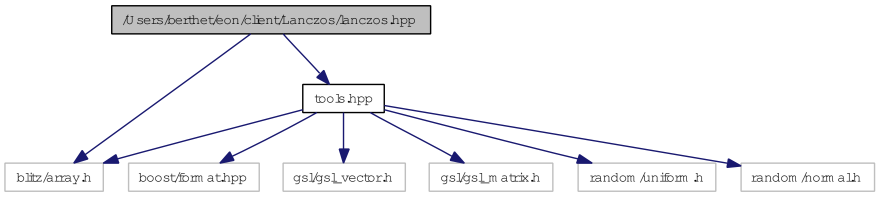
Date:

2007-2008 University of Iceland

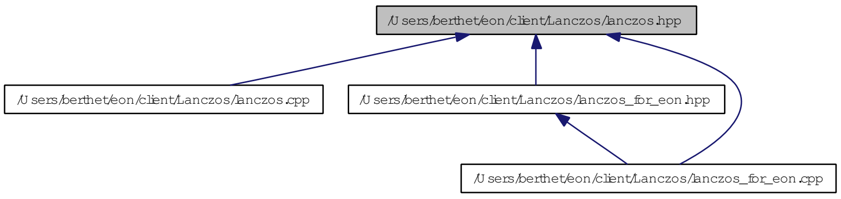
10.2 /Users/berthet/eon/client/Lanczos/lanczos.hpp File Reference

```
#include <blitz/array.h>
#include "tools.hpp"
```

Include dependency graph for lanczos.hpp:



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


/Users/berthet/eon/client/Lanczos/lanczos.hpp
/Users/berthet/eon/client/Lanczos/lanczos.cpp
/Users/berthet/eon/client/Lanczos/lanczos_for_eon.hpp
/Users/berthet/eon/client/Lanczos/lanczos_for_eon.cpp

Classes

- class gradient_scanning::Lanczos

  Compute the lowest eigenvalue and eigenvector.

Namespaces

- namespace gradient_scanning

  ConjugateGradient, SaddlePoint searches.

10.2.1 Detailed Description

Class Lanczos.

Author:

Jean Claude C. Berthet

Date:

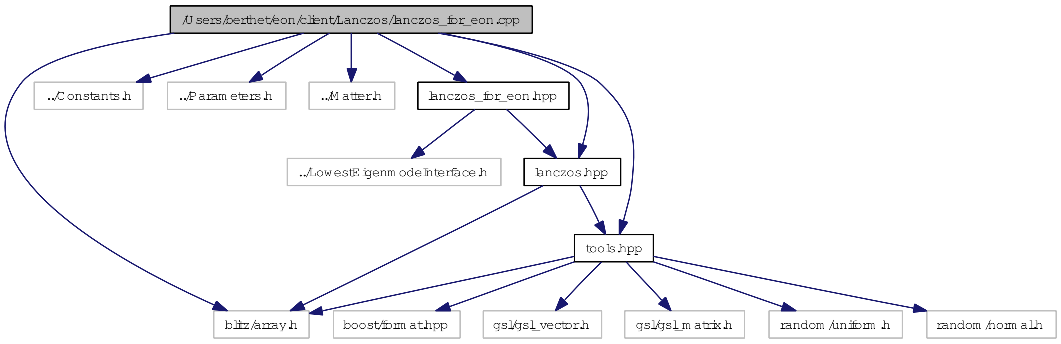
2007-2008 University of Iceland

10.3 /Users/berthet/eon/client/Lanczos/lanczos_for_eon.cpp File Reference

```
#include <blitz/array.h>
#include "../Constants.h"
#include "../Parameters.h"
#include "../Matter.h"
```

```
#include "lanczos_for_eon.hpp"
#include "tools.hpp"
#include "lanczos.hpp"
```

Include dependency graph for lanczos_for_eon.cpp:



10.3.1 Detailed Description

Interface Lanczos for EON.

Author:

Jean Claude C. Berthet

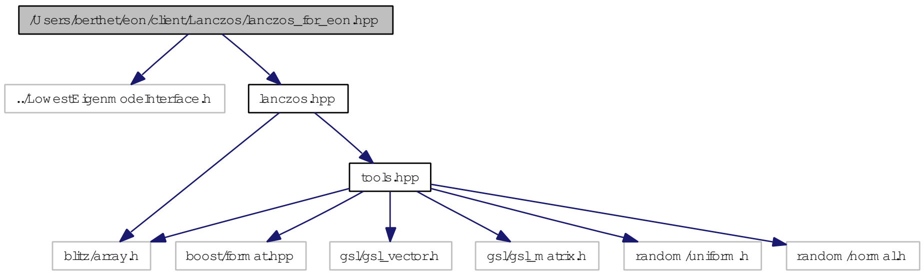
Date:

2007 University of Iceland

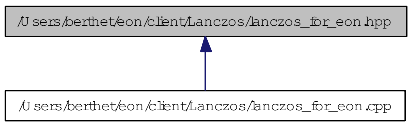
10.4 /Users/berthet/eon/client/Lanczos/lanczos_for_eon.hpp File Reference

```
#include "../LowestEigenmodeInterface.h"
#include "lanczos.hpp"
```

Include dependency graph for lanczos_for_eon.hpp:



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


/Users/berthet/eon/client/Lanczos/lanczos_for_eon.hpp
/Users/berthet/eon/client/Lanczos/lanczos_for_eon.cpp

Classes

- class Lanczos

10.4.1 Detailed Description

Interface Lanczos for EON.

Author:

Jean Claude C. Berthet

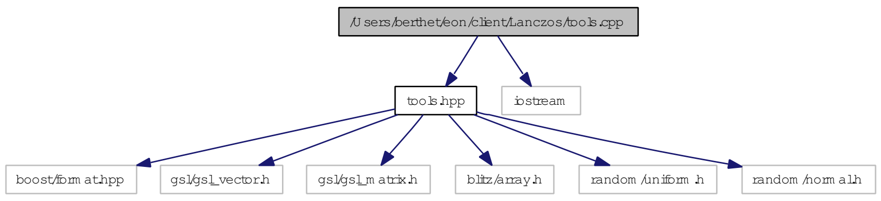
Date:

2007 University of Iceland

10.5 /Users/berthet/eon/client/Lanczos/tools.cpp File Reference

```
#include "tools.hpp"
#include <iostream>
```

Include dependency graph for tools.cpp:



Namespaces

- namespace gradient_scanning

  ConjugateGradient, SaddlePoint searches.

- namespace gradient_scanning::random

  random number generator.

10.5.1 Detailed Description

Tools to handle functions or object computing the gradient

Author:

Jean Claude C. Berthet

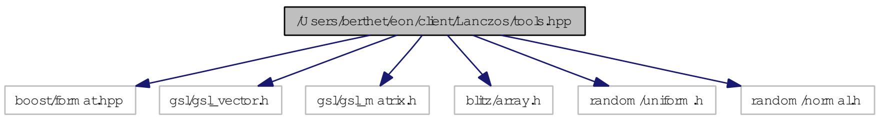
Date:

2006-2007 University of Iceland

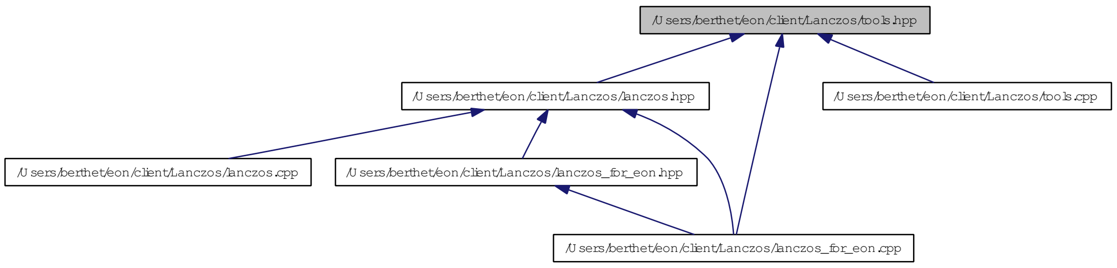
10.6 /Users/berthet/eon/client/Lanczos/tools.hpp File Reference

```
#include <boost/format.hpp>
#include <gsl/gsl_vector.h>
#include <gsl/gsl_matrix.h>
#include <blitz/array.h>
#include <random/uniform.h>
#include <random/normal.h>
```

Include dependency graph for tools.hpp:



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


/Users/berthet/eon/client/Lanczos/tools.hpp
/Users/berthet/eon/client/Lanczos/lanczos.hpp
/Users/berthet/eon/client/Lanczos/tools.cpp
/Users/berthet/eon/client/Lanczos/lanczos.cpp
/Users/berthet/eon/client/Lanczos/lanczos_for_eon.hpp
/Users/berthet/eon/client/Lanczos/lanczos_for_eon.cpp

Classes

- class gradient_scanning::GradientObject

  To compute gradient. To compute gradient from a function (GradientFunction), or from a member function (GradientTemplate).

- class gradient_scanning::GradientTemplate< T >

  When gradient is computed by member function.

Namespaces

- namespace gradient_scanning

ConjugateGradient, SaddlePoint searches.

- namespace gradient_scanning::random

  random number generator.

Typedefs

- typedef void(∗ gradient_scanning::GradientFunction)(blitz::Array< double, 1 > &coordinates, blitz::Array< double, 1 > &gradient)

Functions

- unsigned int gradient_scanning::random::seedWithTime ()

  Seed generator.

- blitz::Array< double, 1 > gradient_scanning::wrapToBlitz (gsl_vector ∗const vgsl)

  Wrap gsl_vector into blitz::Array.

- blitz::Array< double, 2 > gradient_scanning::wrapToBlitz (gsl_matrix ∗const mgsl)

  Wrap gsl_matrix into blitz::Array.

- gsl_vector gradient_scanning::wrapToGsl (blitz::Array< double, 1 > &vblitz)

  Wrap blitz::Array into gsl_vector.

- gsl_matrix gradient_scanning::wrapToGsl (blitz::Array< double, 2 > &mblitz)

  Wrap blitz::Array into gsl_matrix.

- gsl_vector const gradient_scanning::fastWrapToGsl (blitz::Array< double, 1 > const &vblitz)
- gsl_matrix const gradient_scanning::fastWrapToGsl (blitz::Array< double, 2 > const &mblitz)

Variables

- ranlib::Normal< double > gradient_scanning::random::normal

  Gaussian distribution of mean zero and variance 1.

- ranlib::Uniform< double > gradient_scanning::random::uniform

  Uniform distribution in interval [0;1].

10.6.1 Detailed Description

Tools to handle functions or object computing the gradient

Author:

Jean Claude C. Berthet

Date:

2006-2007 University of Iceland