jacobian-hessian-cpp

Generated by Doxygen 1.8.11

Contents

Chapter 1

Main Page

This codes calculate numerically the Jacobian and the Hessian of a vector of function. For matrix operation, Armadillo is used: http://arma.sourceforge.net/ To install, do the follwoing:

```
mkdir build
cd build
cmake ..
make
```

There is an example provided. To understand the concept, let's take an example of a function of vector, with 3 input parameters and 2 output parameters: Example: A function of vector, with 3 input parameters and 2 output parameters.

$$y = \begin{bmatrix} f_1(x_1, x_2, x_3) \\ f_2(x_1, x_2, x_3) \end{bmatrix}$$

$$Jac(y) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \end{bmatrix}$$

$$Hess(y(1)) = \begin{bmatrix} \frac{\partial^2 f_1}{\partial x_1^2} & \frac{\partial^2 f_1}{\partial x_1 \partial x_2} & \frac{\partial^2 f_1}{\partial x_1 \partial x_3} \\ \frac{\partial^2 f_1}{\partial x_2 \partial x_1} & \frac{\partial^2 f_1}{\partial x_2^2} & \frac{\partial^2 f_1}{\partial x_2 \partial x_3} \\ \frac{\partial^2 f_1}{\partial x_3 \partial x_1} & \frac{\partial^2 f_1}{\partial x_3 \partial x_2} & \frac{\partial^2 f_1}{\partial x_3^2} \end{bmatrix}$$

$$Hess(y(2)) = \begin{bmatrix} \frac{\partial^2 f_2}{\partial x_1^2} & \frac{\partial^2 f_2}{\partial x_1 \partial x_2} & \frac{\partial^2 f_2}{\partial x_1 \partial x_3} \\ \frac{\partial^2 f_2}{\partial x_2 \partial x_1} & \frac{\partial^2 f_2}{\partial x_2^2} & \frac{\partial^2 f_2}{\partial x_2 \partial x_3} \\ \frac{\partial^2 f_2}{\partial x_3 \partial x_1} & \frac{\partial^2 f_2}{\partial x_3 \partial x_2} & \frac{\partial^2 f_2}{\partial x_3^2} \end{bmatrix}$$

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

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

Chapter 4

Class Documentation

4.1 FX Class Reference

Public Member Functions

- FX (colvec(∗f)(colvec &x, colvec &some_constants))

  Constructor, create a mathematical function.
- ∼FX ()

  Destructor, nothing happens here.
- mat JacobianAt (colvec &x, colvec &some_constants)

  Calculate the Jacobian at certain inputs.
- mat HessianAt (colvec &x, colvec &some_constants, int i)

  Calculate the Hessian, at certain inputs.
- colvec SolveAt (colvec &x, colvec &some_constants)

  Solve the function at certain inputs.
- void SetEpsilon (double epsilon)

  A very small number.

4.1.1 Constructor & Destructor Documentation

4.1.1.1 FX::FX (colvec(∗)(colvec &x, colvec &some_constants) *f*)

Constructor, create a mathematical function.

Parameters

f	Address to the user defined mathematical function.

4.1.2 Member Function Documentation

4.1.2.1 mat FX::HessianAt (colvec & *x*, colvec & *some_constants*, int *i*)

Calculate the Hessian, at certain inputs.

Parameters

x	Location where the Hessian is computed.
some_constants	Optional constants used in the function.
i	For a function of vector, do Hessian at i-th element of the vector.

Returns

Hessian at location x.

4.1.2.2 mat FX::JacobianAt (colvec & *x*, colvec & *some_constants*)

Calculate the Jacobian at certain inputs.

Parameters

x	Location where the Jacobian is computed.
some_constants	Optional constants used in the function.

Returns

Jacobian at location x

4.1.2.3 void FX::SetEpsilon (double *epsilon*)

A very small number.

Parameters

epsilon	A very small number.

4.1.2.4 colvec FX::SolveAt (colvec & *x*, colvec & *some_constants*)

Solve the function at certain inputs.

Parameters

x	Location where the function is solved.
some_constants	Optional constants used in the function.

Returns

Result from solving the function.

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

- src/fx.h
- src/fx.cpp

Chapter 5

File Documentation

5.1 src/fx.cpp File Reference

Implement a user defined mathematical function.

```
#include "fx.h"
```

5.1.1 Detailed Description

Implement a user defined mathematical function.

Author

Auralius Manurung

Date

28 Jan 2017

5.2 src/fx.h File Reference

A header file for a user defined mathematical function.

```
#include <math.h>
#include <assert.h>
#include <armadillo>
```

Classes

- class FX

5.2.1 Detailed Description

A header file for a user defined mathematical function.

Author

Auralius Manurung

Date

28 Jan 2017

5.3 src/main1.cpp File Reference

This is to test the class FX.

```
#include "fx.h"
```

Functions

- colvec **foo** (colvec &x, colvec &a)
- int **main** (int argc, char **argv)

5.3.1 Detailed Description

This is to test the class FX.

Author

Auralius Manurung

Date

27 Jan 2017

Index