

Function optimization

Generated by Doxygen 1.9.5

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 BoxArea Class Reference	7
4.1.1 Detailed Description	7
4.1.2 Member Function Documentation	8
4.1.2.1 is_in()	8
4.2 dimensional_limits Struct Reference	9
4.2.1 Detailed Description	9
4.3 Function Class Reference	9
4.3.1 Detailed Description	10
4.3.2 Member Function Documentation	10
4.3.2.1 operator()	10
4.4 GDOptInfo Class Reference	11
4.4.1 Detailed Description	11
4.5 GradientDescent Class Reference	11
4.5.1 Detailed Description	12
4.5.2 Constructor & Destructor Documentation	12
4.5.2.1 ~GradientDescent()	12
4.5.3 Member Function Documentation	12
4.5.3.1 get_opt_info()	12
4.5.3.2 make_step()	13
4.5.3.3 optimize()	13
4.6 IterAfterImpSSC Class Reference	13
4.6.1 Detailed Description	14
4.6.2 Member Function Documentation	14
4.6.2.1 criterion()	14
4.7 IterOnlySSC Class Reference	14
4.7.1 Detailed Description	15
4.7.2 Member Function Documentation	15
4.7.2.1 criterion()	15
4.8 MDFunction Class Reference	15
4.8.1 Detailed Description	16
4.8.2 Member Function Documentation	16
4.8.2.1 operator()	16
4.9 MinGradNormGDSC Class Reference	16

4.9.1 Detailed Description	17
4.9.2 Member Function Documentation	17
4.9.2.1 criterion()	17
4.10 MinRelImpNormGDSC Class Reference	17
4.10.1 Detailed Description	18
4.10.2 Member Function Documentation	18
4.10.2.1 criterion()	18
4.11 MinStepNormGDSC Class Reference	18
4.11.1 Detailed Description	19
4.11.2 Member Function Documentation	19
4.11.2.1 criterion()	19
4.12 NormDiffSSC Class Reference	19
4.12.1 Detailed Description	20
4.12.2 Member Function Documentation	20
4.12.2.1 criterion()	20
4.13 OptimizationMethod Class Reference	20
4.13.1 Detailed Description	21
4.13.2 Member Function Documentation	21
4.13.2.1 make_step()	22
4.13.2.2 optimize()	22
4.14 OptInfo Class Reference	22
4.14.1 Detailed Description	23
4.15 OptResult Class Reference	23
4.15.1 Detailed Description	23
4.16 StochasticOptimization Class Reference	24
4.16.1 Detailed Description	24
4.16.2 Member Function Documentation	24
4.16.2.1 get_opt_info()	24
4.16.2.2 make_step()	25
4.16.2.3 optimize()	25
4.17 StohOptInfo Class Reference	25
4.17.1 Detailed Description	26
4.18 StopCriterion Class Reference	26
4.18.1 Detailed Description	26
4.18.2 Member Function Documentation	27
4.18.2.1 criterion()	27
4.18.2.2 operator()	27
5 File Documentation	29
5.1 BoxArea.h	29
5.2 Function.h	29
5.3 GDOptInfo.h	30

5.4 GradientDescent.h	30
5.5 IterAfterImpSSC.h	30
5.6 IterOnlySSC.h	31
5.7 MDFunction.h	31
5.8 MinGradNormGDSC.h	31
5.9 MinRelImpNormGDSC.h	32
5.10 MinStepNormGDSC.h	32
5.11 NormDiffSSC.h	32
5.12 OptimizationMethod.h	33
5.13 OptInfo.h	33
5.14 OptResult.h	33
5.15 StochasticOptimization.h	34
5.16 StohOptInfo.h	34
5.17 StopCriterion.h	34
6 Example Documentation	37
6.1 main.cpp	37
Index	39

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

BoxArea	7
dimensional_limits	9
Function	9
MDFunction	15
OptimizationMethod	20
GradientDescent	11
StochasticOptimization	24
OptInfo	22
GDOptInfo	11
StohOptInfo	25
OptResult	23
StopCriterion	26
IterAfterImpSSC	13
IterOnlySSC	14
MinGradNormGDSC	16
MinRelImpNormGDSC	17
MinStepNormGDSC	18
NormDiffSSC	19

Chapter 2

Class Index

2.1 Class List

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

BoxArea	A class that contains that stores axes limits	7
dimensional_limits	A structure that stores axis limits	9
Function	Class that contains a pointer on scalar function with several arguments	9
GDOptInfo	Class that contains information about gradient descent	11
GradientDescent	Class that performs gradient descent and stores information about optimization	11
IterAfterImpSSC	Class of stop criterion for stochastic optimization that triggeres after maximum number of iterations after last improvement or after maximum number of iterations	13
IterOnlySSC	Class of stop criterion of stochastic optimization that triggeres after maximum number of iterations	14
MDFunction	Class that stores vector-function with multidim argument	15
MinGradNormGDSC	Class of stop criterion for gradient descent that triggeres if gradient norm is less than threshold or after maximum number of iterations	16
MinRelImpNormGDSC	Class of stop criterion for gradient descent that triggeres if norm of relative difference is less than threshold or after maximum number of iterations	17
MinStepNormGDSC	Class of stop critreion for gradient descent that triggeres if step norm is less than threshold or after maximum number of iterations	18
NormDiffSSC	Class of stop criterion for stochastic optimization that triggeres if last improvement norm is less than threshold or after maximum number of iterations	19
OptimizationMethod	Abstract class for optimization methods	20
OptInfo	Abstract class that stores information about optimization iteration	22
OptResult	Class that stores result of optimization	23

StochasticOptimization	
Class that performs stochastic optimization	24
StohOptInfo	
Class that stores optimization info for stochastic optimization	25
StopCriterion	
Abstract class that used for performing a stop of optimization process	26

Chapter 3

File Index

3.1 File List

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

BoxArea.h	??
Function.h	??
GDOptInfo.h	??
GradientDescent.h	??
IterAfterImpSSC.h	??
IterOnlySSC.h	??
MDFunction.h	??
MinGradNormGDSC.h	??
MinRelImpNormGDSC.h	??
MinStepNormGDSC.h	??
NormDiffSSC.h	??
OptimizationMethod.h	??
OptInfo.h	??
OptResult.h	??
StochasticOptimization.h	??
StohOptInfo.h	??
StopCriterion.h	??

Chapter 4

Class Documentation

4.1 BoxArea Class Reference

A class that contains that stores axes limits.

```
#include <BoxArea.h>
```

Public Member Functions

- **BoxArea** (unsigned int n_dim_, const std::vector< [dimensional_limits](#) > &lims)
- **BoxArea** (const [BoxArea](#) &other)
- **BoxArea** ([BoxArea](#) &&other) noexcept
- unsigned int **get_n_dim** () const
- void **swap** ([BoxArea](#) &other) noexcept
- [BoxArea](#) & **operator=** ([BoxArea](#) other) noexcept
- [dimensional_limits](#) **get_limits** (unsigned int i) const
- bool **is_in** (const std::vector< double > &point) const

Protected Attributes

- unsigned int **n_dim** = 0
Number of dimensions.
- std::vector< [dimensional_limits](#) > **limits**
Axes limits.

4.1.1 Detailed Description

A class that contains that stores axes limits.

Examples

[main.cpp](#).

4.1.2 Member Function Documentation

4.1.2.1 `is_in()`

```
bool BoxArea::is_in (
    const std::vector< double > & point ) const
```

Checks if point is in area.

Parameters

<i>point</i>	Vector that contains coordinates of point.
--------------	--

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

- BoxArea.h
- BoxArea.cpp

4.2 dimensional_limits Struct Reference

A structure that stores axis limits.

```
#include <BoxArea.h>
```

Public Member Functions

- **dimensional_limits** (double low, double up)

Public Attributes

- double **lower** = -1
Lower axis limit.
- double **upper** = 1
Upper axis limit.

4.2.1 Detailed Description

A structure that stores axis limits.

Examples

[main.cpp](#).

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

- BoxArea.h

4.3 Function Class Reference

Class that contains a pointer on scalar function with several arguments.

```
#include <Function.h>
```

Public Member Functions

- **Function** (double(*func_)(std::vector< double >), unsigned int n_dim_)
- **Function** (const [Function](#) &other)
- **Function** ([Function](#) &&other) noexcept
- **Function** & **operator=** ([Function](#) other)
- void **swap** ([Function](#) &other) noexcept
- double **operator()** (const std::vector< double > &argument) const
- void **set_func** (double(*new_func)(std::vector< double >), unsigned int new_n_dim)
- unsigned int **get_n_dim** () const

Protected Attributes

- unsigned int **n_dim** = 0
Number of dimensions of argument.
- double(* **func**)(std::vector< double >) = nullptr
Pointer on a function.

4.3.1 Detailed Description

Class that contains a pointer on scalar function with several arguments.

Examples

[main.cpp](#).

4.3.2 Member Function Documentation

4.3.2.1 operator>()

```
double Function::operator() (
    const std::vector< double > & argument ) const
```

Returns value of function.

Parameters

<i>argument</i>	Vector that contains coordinates of a point.
-----------------	--

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

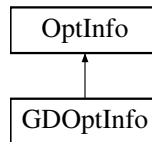
- Function.h
- Function.cpp

4.4 GDOptInfo Class Reference

Class that contains information about gradient descent.

```
#include <GDOptInfo.h>
```

Inheritance diagram for GDOptInfo:



Public Member Functions

- **GDOptInfo** (unsigned int n_iter_, double grad_norm_, double last_step_norm_, double rel_imp_norm_)
- **GDOptInfo** (const [GDOptInfo](#) &other)
- **GDOptInfo** ([GDOptInfo](#) &&other) noexcept
- void **swap** ([GDOptInfo](#) &other) noexcept
- [GDOptInfo](#) & **operator=** ([GDOptInfo](#) other) noexcept
- double **get_grad_norm** () const
- double **get_last_step_norm** () const
- double **get_rel_imp_norm** () const

Additional Inherited Members

4.4.1 Detailed Description

Class that contains information about gradient descent.

Fields of this class are result of iteration of gradient descent.

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

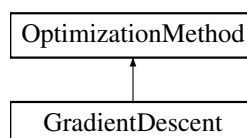
- GDOptInfo.h
- GDOptInfo.cpp

4.5 GradientDescent Class Reference

Class that performs gradient descent and stores information about optimization.

```
#include <GradientDescent.h>
```

Inheritance diagram for GradientDescent:



Public Member Functions

- **GradientDescent** ([Function](#) *func_, [MDFunction](#) *grad_, [BoxArea](#) *box_area_, [StopCriterion](#) *stop_criterion_, const std::vector< double > &first_point_, unsigned int lr_steps_, std::string norm_name_)
- **GradientDescent** (const [GradientDescent](#) &other)
- **GradientDescent** ([GradientDescent](#) &&other) noexcept
- void **swap** ([GradientDescent](#) &other) noexcept
- [GradientDescent](#) & **operator=** ([GradientDescent](#) other) noexcept
- void **set_first_point** (const std::vector< double > &point)
- void **make_step** ()
- [OptResult](#) **optimize** ()
- [OptInfo](#) * **get_opt_info** () const
- **~GradientDescent** ()

Additional Inherited Members

4.5.1 Detailed Description

Class that performs gradient descent and stores information about optimization.

Examples

[main.cpp](#).

4.5.2 Constructor & Destructor Documentation

4.5.2.1 ~GradientDescent()

```
GradientDescent::~GradientDescent ( )
```

Returns optimization information.

4.5.3 Member Function Documentation

4.5.3.1 get_opt_info()

```
OptInfo * GradientDescent::get_opt_info ( ) const [virtual]
```

Implements [OptimizationMethod](#).

4.5.3.2 make_step()

```
void GradientDescent::make_step ( ) [virtual]
```

Makes one iteration of gradient descent.

Implements [OptimizationMethod](#).

4.5.3.3 optimize()

```
OptResult GradientDescent::optimize ( ) [virtual]
```

Performs gradient descent.

Implements [OptimizationMethod](#).

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

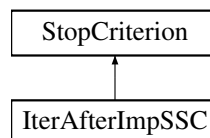
- GradientDescent.h
- GradientDescent.cpp

4.6 IterAfterImpSSC Class Reference

Class of stop criterion for stochastic optimization that triggeres after maximum number of iterations after last improvement or after maximum number of iterations.

```
#include <IterAfterImpSSC.h>
```

Inheritance diagram for IterAfterImpSSC:



Public Member Functions

- **IterAfterImpSSC** (unsigned int max_n_iter_, unsigned int max_n_iter_after_imp_)
- **IterAfterImpSSC** (const [IterAfterImpSSC](#) &other)
- **IterAfterImpSSC** ([IterAfterImpSSC](#) &&other) noexcept
- void **swap** ([IterAfterImpSSC](#) &other) noexcept
- [IterAfterImpSSC](#) & **operator=** ([IterAfterImpSSC](#) other) noexcept
- bool **criterion** ([OptInfo](#) *status)
- unsigned int **get_max_n_iter_after_imp** () const

Additional Inherited Members

4.6.1 Detailed Description

Class of stop criterion for stochastic optimization that triggers after maximum number of iterations after last improvement or after maximum number of iterations.

This stop criterion triggers if function value was not decreased after some number of iterations.

Examples

[main.cpp](#).

4.6.2 Member Function Documentation

4.6.2.1 criterion()

```
bool IterAfterImpSSC::criterion (
    OptInfo * status ) [virtual]
```

Returns true if stop criterion is triggered, otherwise returns false.

Implements [StopCriterion](#).

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

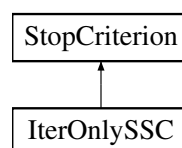
- IterAfterImpSSC.h
- IterAfterImpSSC.cpp

4.7 IterOnlySSC Class Reference

Class of stop criterion of stochastic optimization that triggers after maximum number of iterations.

```
#include <IterOnlySSC.h>
```

Inheritance diagram for IterOnlySSC:



Public Member Functions

- **IterOnlySSC** (unsigned int max_n_iter_)
- **IterOnlySSC** (const [IterOnlySSC](#) &other)
- **IterOnlySSC** ([IterOnlySSC](#) &&other) noexcept
- void **swap** ([IterOnlySSC](#) &other) noexcept
- [IterOnlySSC](#) & **operator=** ([IterOnlySSC](#) other) noexcept
- bool **criterion** ([OptInfo](#) *opt_info)

Additional Inherited Members

4.7.1 Detailed Description

Class of stop criterion of stochastic optimization that triggeres after maximum number of iterations.

4.7.2 Member Function Documentation

4.7.2.1 criterion()

```
bool IterOnlySSC::criterion (
    OptInfo * opt_info ) [virtual]
```

Returns true if stop criterion is triggered, otherwise returns false.

Implements [StopCriterion](#).

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

- IterOnlySSC.h
- IterOnlySSC.cpp

4.8 MDFunction Class Reference

Class that stores vector-function with multidim argument.

```
#include <MDFunction.h>
```

Public Member Functions

- **MDFunction** (std::vector< double >(*func_)(std::vector< double >), unsigned int arg_n_dim_)
- **MDFunction** (const [MDFunction](#) &other)
- **MDFunction** ([MDFunction](#) &&other) noexcept
- [MDFunction](#) & **operator=** ([MDFunction](#) other)
- void **swap** ([MDFunction](#) &other) noexcept
- std::vector< double > **operator()** (const std::vector< double > &argument) const
- void **set_func** (std::vector< double >(*new_func)(std::vector< double >), unsigned int new_arg_n_dim)
- unsigned int **get_arg_n_dim** () const

Protected Attributes

- unsigned int **arg_n_dim** = 0
Number of dimensions of argument.
- `std::vector< double >(* func)(std::vector< double >) = nullptr`
Pointer on a function.

4.8.1 Detailed Description

Class that stores vector-function with multidim argument.

Examples

[main.cpp](#).

4.8.2 Member Function Documentation

4.8.2.1 operator()

```
std::vector< double > MDFunction::operator() (
    const std::vector< double > & argument ) const
```

Returns value of function at the point.

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

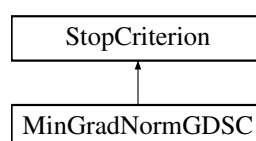
- MDFunction.h
- MDFunction.cpp

4.9 MinGradNormGDSC Class Reference

Class of stop criterion for gradient descent that triggeres if gradient norm is less than threshold or after maximum number of iterations.

```
#include <MinGradNormGDSC.h>
```

Inheritance diagram for MinGradNormGDSC:



Public Member Functions

- **MinGradNormGDSC** (unsigned int max_n_iter_, double min_grad_norm_)
- **MinGradNormGDSC** (const [MinGradNormGDSC](#) &other)
- **MinGradNormGDSC** ([MinGradNormGDSC](#) &&other) noexcept
- void **swap** ([MinGradNormGDSC](#) &other) noexcept
- [MinGradNormGDSC](#) & **operator=** ([MinGradNormGDSC](#) other) noexcept
- bool **criterion** ([OptInfo](#) *status)
- double **get_min_grad_norm** () const

Additional Inherited Members

4.9.1 Detailed Description

Class of stop criterion for gradient descent that triggeres if gradient norm is less than threshold or after maximum number of iterations.

Examples

[main.cpp](#).

4.9.2 Member Function Documentation

4.9.2.1 criterion()

```
bool MinGradNormGDSC::criterion (
    OptInfo * status ) [virtual]
```

Returns true if stop criterion is triggered, otherwise returns false.

Implements [StopCriterion](#).

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

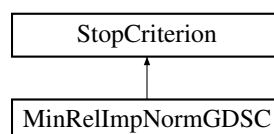
- MinGradNormGDSC.h
- MinGradNormGDSC.cpp

4.10 MinRelImpNormGDSC Class Reference

Class of stop criterion for gradient descent that triggeres if norm of relative difference is less than threshold or after maximum number of iterations.

```
#include <MinRelImpNormGDSC.h>
```

Inheritance diagram for MinRelImpNormGDSC:



Public Member Functions

- **MinRelImpNormGDSC** (unsigned int max_n_iter_, double min_rel_imp_norm_)
- **MinRelImpNormGDSC** (const [MinRelImpNormGDSC](#) &other)
- **MinRelImpNormGDSC** ([MinRelImpNormGDSC](#) &&other) noexcept
- void **swap** ([MinRelImpNormGDSC](#) &other) noexcept
- [MinRelImpNormGDSC](#) & **operator=** ([MinRelImpNormGDSC](#) other) noexcept
- bool **criterion** ([OptInfo](#) *status)
- double **get_min_rel_imp_norm** () const

Additional Inherited Members

4.10.1 Detailed Description

Class of stop criterion for gradient descent that triggeres if norm of relative difference is less than threshold or after maximum number of iterations.

Norm of relative difference is norm of difference of current value of function and previous value of function, divided by current value of function.

4.10.2 Member Function Documentation

4.10.2.1 criterion()

```
bool MinRelImpNormGDSC::criterion (
    OptInfo * status ) [virtual]
```

Returns true if stop criterion is triggered, otherwise returns false.

Implements [StopCriterion](#).

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

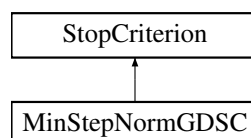
- MinRelImpNormGDSC.h
- MinRelImpNormGDSC.cpp

4.11 MinStepNormGDSC Class Reference

Class of stop critreion for gradient descent that triggeres if step norm is less than threshold or after maximum number of iterations.

```
#include <MinStepNormGDSC.h>
```

Inheritance diagram for MinStepNormGDSC:



Public Member Functions

- **MinStepNormGDSC** (unsigned int max_n_iter_, double min_step_norm_)
- **MinStepNormGDSC** (const [MinStepNormGDSC](#) &other)
- **MinStepNormGDSC** ([MinStepNormGDSC](#) &&other) noexcept
- void **swap** ([MinStepNormGDSC](#) &other) noexcept
- [MinStepNormGDSC](#) & **operator=** ([MinStepNormGDSC](#) other) noexcept
- bool **criterion** ([OptInfo](#) *status)
- double **get_min_step_norm** () const

Additional Inherited Members

4.11.1 Detailed Description

Class of stop critreion for gradient descent that triggeres if step norm is less than threshold or after maximum number of iterations.

Step norm is norm of difference of current point and previous point.

4.11.2 Member Function Documentation

4.11.2.1 criterion()

```
bool MinStepNormGDSC::criterion (
    OptInfo * status ) [virtual]
```

Returns true if stop criterion is triggered, otherwise returns false.

Implements [StopCriterion](#).

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

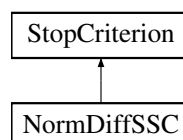
- MinStepNormGDSC.h
- MinStepNormGDSC.cpp

4.12 NormDiffSSC Class Reference

Class of stop criterion for stohastic optimization that triggeres if last improvement norm is less than threshold or after maximum number of iterations.

```
#include <NormDiffSSC.h>
```

Inheritance diagram for NormDiffSSC:



Public Member Functions

- **NormDiffSSC** (unsigned int max_n_iter_, double min_last_imp_norm_)
- **NormDiffSSC** (const [NormDiffSSC](#) &other)
- **NormDiffSSC** ([NormDiffSSC](#) &&other) noexcept
- void **swap** ([NormDiffSSC](#) &other) noexcept
- [NormDiffSSC](#) & **operator=** ([NormDiffSSC](#) other) noexcept
- bool **criterion** ([OptInfo](#) *opt_info)
- double **get_min_last_imp_norm** () const

Additional Inherited Members

4.12.1 Detailed Description

Class of stop criterion for stochastic optimization that triggers if last improvement norm is less than threshold or after maximum number of iterations.

Last improvement norm is norm of difference of current value of function and previous value of function.

4.12.2 Member Function Documentation

4.12.2.1 criterion()

```
bool NormDiffSSC::criterion (
    OptInfo * opt_info ) [virtual]
```

Returns true if stop criterion is triggered, otherwise returns false.

Implements [StopCriterion](#).

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

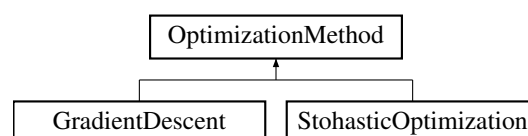
- NormDiffSSC.h
- NormDiffSSC.cpp

4.13 OptimizationMethod Class Reference

Abstract class for optimization methods.

```
#include <OptimizationMethod.h>
```

Inheritance diagram for OptimizationMethod:



Public Member Functions

- virtual [OptInfo](#) * **get_opt_info** () const =0
- virtual [OptResult](#) **optimize** ()=0
- unsigned int **get_n_dim** () const
- unsigned int **get_n_iter** () const

Protected Member Functions

- **OptimizationMethod** ([Function](#) *func_, [BoxArea](#) *box_area_, [StopCriterion](#) *stop_criterion_)
- **OptimizationMethod** (const [OptimizationMethod](#) &other)
- **OptimizationMethod** ([OptimizationMethod](#) &&other) noexcept
- void **swap** ([OptimizationMethod](#) &other) noexcept
- virtual void **make_step** ()=0

Protected Attributes

- std::vector< std::vector< double > > **point_history**
Vector that stores all points of optimization process.
- unsigned int **n_iter** = 0
Current number of iterations.
- unsigned int **n_dim** = 0
Number of dimensions of points.
- [BoxArea](#) * **box_area** = nullptr
Limits of search area.
- [Function](#) * **func** = nullptr
Pointer on function that should be optimized.
- [StopCriterion](#) * **stop_criterion** = nullptr
Pointer on stop criterion.

4.13.1 Detailed Description

Abstract class for optimization methods.

Methods optimize, make_step and get_opt_info should be defined in inheritors.

Examples

[main.cpp](#).

4.13.2 Member Function Documentation

4.13.2.1 make_step()

```
virtual void OptimizationMethod::make_step ( ) [protected], [pure virtual]
```

Abstract method that makes one iteration of optimization.

Implemented in [GradientDescent](#), and [StochasticOptimization](#).

4.13.2.2 optimize()

```
virtual OptResult OptimizationMethod::optimize ( ) [pure virtual]
```

Abstract method that performs optimization.

Implemented in [GradientDescent](#), and [StochasticOptimization](#).

Examples

[main.cpp](#).

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

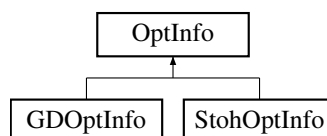
- OptimizationMethod.h
- OptimizationMethod.cpp

4.14 OptInfo Class Reference

Abstract class that stores information about optimization iteration.

```
#include <OptInfo.h>
```

Inheritance diagram for OptInfo:



Public Member Functions

- **OptInfo** (const [OptInfo](#) &other)
- **OptInfo** ([OptInfo](#) &&other) noexcept
- **OptInfo** (unsigned int n_iter_)
- void **swap** ([OptInfo](#) &other) noexcept
- [OptInfo](#) & **operator=** ([OptInfo](#) other)
- unsigned int **get_n_iter** () const

Protected Attributes

- unsigned int **n_iter**
Current number of iterations.

4.14.1 Detailed Description

Abstract class that stores information about optimization iteration.

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

- OptInfo.h
- OptInfo.cpp

4.15 OptResult Class Reference

Class that stores result of optimization.

```
#include <OptResult.h>
```

Public Member Functions

- **OptResult** (const std::vector< double > &min_point_, double min_value_, unsigned int n_iter_)
- **OptResult** (const [OptResult](#) &other)
- **OptResult** ([OptResult](#) &&other) noexcept
- void **swap** ([OptResult](#) &other) noexcept
- [OptResult](#) & **operator=** ([OptResult](#) other) noexcept
- unsigned int **get_n_iter** () const
- std::vector< double > **get_min_point** () const
- double **get_min_value** () const

4.15.1 Detailed Description

Class that stores result of optimization.

Examples

[main.cpp](#).

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

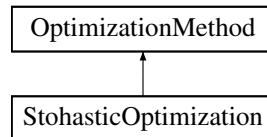
- OptResult.h
- OptResult.cpp

4.16 StochasticOptimization Class Reference

Class that performs stochastic optimization.

```
#include <StochasticOptimization.h>
```

Inheritance diagram for StochasticOptimization:



Public Member Functions

- **StochasticOptimization** ([Function](#) *func_, [BoxArea](#) *box_area_, [StopCriterion](#) *stop_criterion_, const std::vector< double > &first_point_, double delta_, double p_)
- **StochasticOptimization** (const [StochasticOptimization](#) &other)
- **StochasticOptimization** ([StochasticOptimization](#) &&other)
- void **swap** ([StochasticOptimization](#) &other) noexcept
- [StochasticOptimization](#) & **operator=** ([StochasticOptimization](#) other) noexcept
- void **set_first_point** (const std::vector< double > &point)
- void **make_step** ()
- [OptResult](#) **optimize** ()
- [OptInfo](#) * **get_opt_info** () const

Additional Inherited Members

4.16.1 Detailed Description

Class that performs stochastic optimization.

Examples

[main.cpp](#).

4.16.2 Member Function Documentation

4.16.2.1 get_opt_info()

```
OptInfo * StochasticOptimization::get_opt_info ( ) const [virtual]
```

Implements [OptimizationMethod](#).

4.16.2.2 make_step()

```
void StochasticOptimization::make_step ( ) [virtual]
```

Performs one iteration of stochastic optimization.

Implements [OptimizationMethod](#).

4.16.2.3 optimize()

```
OptResult StochasticOptimization::optimize ( ) [virtual]
```

Performs stochastic optimization.

Implements [OptimizationMethod](#).

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

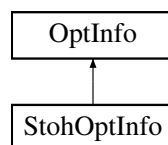
- StochasticOptimization.h
- StochasticOptimization.cpp

4.17 StohOptInfo Class Reference

Class that stores optimization info for stochastic optimization.

```
#include <StohOptInfo.h>
```

Inheritance diagram for StohOptInfo:



Public Member Functions

- **StohOptInfo** (unsigned int n_iter_, unsigned int n_iter_after_imp, double last_imp_norm_)
- **StohOptInfo** (const [StohOptInfo](#) &other)
- **StohOptInfo** ([StohOptInfo](#) &&other) noexcept
- void **swap** ([StohOptInfo](#) &other) noexcept
- [StohOptInfo](#) & **operator=** ([StohOptInfo](#) other) noexcept
- unsigned int **get_n_iter_after_imp** () const
- double **get_last_imp_norm** () const

Additional Inherited Members

4.17.1 Detailed Description

Class that stores optimization info for stochastic optimization.

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

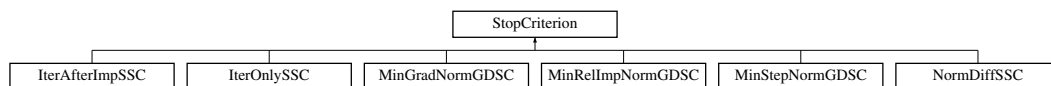
- StohOptInfo.h
- StohOptInfo.cpp

4.18 StopCriterion Class Reference

Abstract class that used for performing a stop of optimization process.

```
#include <StopCriterion.h>
```

Inheritance diagram for StopCriterion:



Public Member Functions

- virtual bool [criterion](#) ([OptInfo](#) *opt_info)=0
- bool [operator\(\)](#) ([OptInfo](#) *opt_info)

Protected Member Functions

- **StopCriterion** (unsigned int max_n_iter_)
- **StopCriterion** (const [StopCriterion](#) &other)
- **StopCriterion** ([StopCriterion](#) &&other) noexcept
- void **swap** ([StopCriterion](#) &other) noexcept

Protected Attributes

- unsigned int **max_n_iter**
Maximum number of iterations.

4.18.1 Detailed Description

Abstract class that used for performing a stop of optimization process.

Method criterion should be defined in inheritors.

Examples

[main.cpp](#).

4.18.2 Member Function Documentation

4.18.2.1 criterion()

```
virtual bool StopCriterion::criterion (
    OptInfo * opt_info ) [pure virtual]
```

Abstract method, should return true if stop criterion is triggered, otherwise should return false.

Implemented in [IterOnlySSC](#), [NormDiffSSC](#), [IterAfterImpSSC](#), [MinGradNormGDSC](#), [MinRelImpNormGDSC](#), and [MinStepNormGDSC](#).

4.18.2.2 operator()

```
bool StopCriterion::operator() (
    OptInfo * opt_info )
```

Set fields of object of this class.

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

- StopCriterion.h
- StopCriterion.cpp

Chapter 5

File Documentation

5.1 BoxArea.h

```
1 #pragma once
2 #include <vector>
3
4
5
6
7 struct dimensional_limits {
8     double lower = -1;
9     double upper = 1;
10    dimensional_limits() {}
11    dimensional_limits(double low, double up) : lower(low), upper(up) {}
12 };
13
14
15
16
17
18
19 class BoxArea
20 {
21 protected:
22     unsigned int n_dim = 0;
23     std::vector<dimensional_limits> limits;
24
25 public:
26     BoxArea();
27     BoxArea(unsigned int n_dim_, const std::vector<dimensional_limits>& lms);
28     BoxArea(const BoxArea& other);
29     BoxArea(BoxArea&& other) noexcept;
30
31     unsigned int get_n_dim() const;
32     void swap(BoxArea& other) noexcept;
33     BoxArea& operator=(BoxArea other) noexcept;
34     dimensional_limits get_limits(unsigned int i) const;
35     bool is_in(const std::vector<double>& point) const;
36     ~BoxArea();
37 };
38
```

5.2 Function.h

```
1 #pragma once
2 #include <vector>
3
4
5
6
7 class Function
8 {
9 protected:
10     unsigned int n_dim = 0;
11     double (*func)(std::vector<double>) = nullptr;
12
13 public:
14     Function();
15     Function(double (*func_)(std::vector<double>), unsigned int n_dim_);
16     Function(const Function& other);
17     Function(Function&& other) noexcept;
18
19     Function& operator=(Function other);
20     void swap(Function& other) noexcept;
21     double operator()(const std::vector<double>& argument) const;
22     void set_func(double (*new_func)(std::vector<double>), unsigned int new_n_dim);
23     unsigned int get_n_dim() const;
24     ~Function();
25 };
26
27
28
29
30
31
32
33
```

5.3 GDOptInfo.h

```

1 #pragma once
2 #include "OptInfo.h"
3
4 class GDOptInfo : public OptInfo {
5 private:
6     double grad_norm;
7     double last_step_norm;
8     double rel_imp_norm;
9 public:
10     GDOptInfo();
11     GDOptInfo(unsigned int n_iter_, double grad_norm_, double last_step_norm_, double rel_imp_norm_);
12     GDOptInfo(const GDOptInfo& other);
13     GDOptInfo(GDOptInfo&& other) noexcept;
14
15     void swap(GDOptInfo& other) noexcept;
16     GDOptInfo& operator=(GDOptInfo other) noexcept;
17     double get_grad_norm() const;
18     double get_last_step_norm() const;
19     double get_rel_imp_norm() const;
20     ~GDOptInfo();
21 };
22
23
24

```

5.4 GradientDescent.h

```

1 #pragma once
2 #include "OptimizationMethod.h"
3 #include "MDFunction.h"
4 #include "GDOptInfo.h"
5 #include "MinGradNormGDSC.h"
6 #include "MinStepNormGDSC.h"
7 #include "MinRelImpNormGDSC.h"
8
9 class GradientDescent : public OptimizationMethod {
10 private:
11     std::vector<double> value_history;
12     MDFunction* grad = nullptr;
13     std::vector<double> first_point;
14     double curr_min_value;
15     unsigned int lr_steps;
16     double grad_norm;
17     double last_step_norm;
18     double rel_imp_norm;
19     unsigned int max_updates_lr = 10;
20     double scale = 10.;
21     std::string norm_name;
22 public:
23     GradientDescent();
24     GradientDescent(Function* func_, MDFunction* grad_, BoxArea* box_area_, StopCriterion*
25 stop_criterion_,
26     const std::vector<double>& first_point_, unsigned int lr_steps_, std::string norm_name_);
27     GradientDescent(const GradientDescent& other);
28     GradientDescent(GradientDescent&& other) noexcept;
29
30     void swap(GradientDescent& other) noexcept;
31     GradientDescent& operator=(GradientDescent other) noexcept;
32     void set_first_point(const std::vector<double>& point);
33     void make_step();
34     OptResult optimize();
35     OptInfo* get_opt_info() const;
36     ~GradientDescent();
37 };
38
39 double l2_norm(std::vector<double> arg);
40 double l1_norm(std::vector<double> arg);
41
42 std::vector<double> operator*(double scalar, const std::vector<double>& vec);
43 std::vector<double> operator+(const std::vector<double>& vec1, const std::vector<double>& vec2);
44 std::vector<double> operator-(const std::vector<double>& vec1, const std::vector<double>& vec2);
45

```

5.5 IterAfterImpSSC.h

```

1 #pragma once
2 #include "StopCriterion.h"
3 #include "StohOptInfo.h"
4

```

```

10 class IterAfterImpSSC : public StopCriterion {
11 private:
12     unsigned int max_n_iter_after_imp;
13 public:
14     IterAfterImpSSC();
15     IterAfterImpSSC(unsigned int max_n_iter_, unsigned int max_n_iter_after_imp_);
16     IterAfterImpSSC(const IterAfterImpSSC& other);
17     IterAfterImpSSC(IterAfterImpSSC&& other) noexcept;
18
19     void swap(IterAfterImpSSC& other) noexcept;
20     IterAfterImpSSC& operator=(IterAfterImpSSC other) noexcept;
21     bool criterion(OptInfo* status);
22     unsigned int get_max_n_iter_after_imp() const;
23     ~IterAfterImpSSC();
24 };
25

```

5.6 IterOnlySSC.h

```

1 #pragma once
2 #include "StopCriterion.h"
3 #include "StohOptInfo.h"
4
5 class IterOnlySSC : public StopCriterion{
6 public:
7     IterOnlySSC();
8     IterOnlySSC(unsigned int max_n_iter_);
9     IterOnlySSC(const IterOnlySSC& other);
10    IterOnlySSC(IterOnlySSC&& other) noexcept;
11
12    void swap(IterOnlySSC& other) noexcept;
13    IterOnlySSC& operator=(IterOnlySSC other) noexcept;
14    bool criterion(OptInfo* opt_info);
15    ~IterOnlySSC();
16 };
17

```

5.7 MDFFunction.h

```

1 #pragma once
2 #include<vector>
3
4 class MDFFunction
5 {
6 protected:
7     unsigned int arg_n_dim = 0;
8     std::vector<double> (*func)(std::vector<double>) = nullptr;
9 public:
10    MDFFunction(); //
11    MDFFunction(std::vector<double>(*func_)(std::vector<double>), unsigned int arg_n_dim_);
12    MDFFunction(const MDFFunction& other); //
13    MDFFunction(MDFFunction&& other) noexcept;
14
15    MDFFunction& operator=(MDFFunction other);
16    void swap(MDFFunction& other) noexcept;
17    std::vector<double> operator() (const std::vector<double>& argument) const;
18    void set_func(std::vector<double> (*new_func)(std::vector<double>), unsigned int new_arg_n_dim);
19    unsigned int get_arg_n_dim() const;
20    ~MDFFunction();
21 };
22

```

5.8 MinGradNormGDSC.h

```

1 #pragma once
2 #include "StopCriterion.h"
3 #include "GDOptInfo.h"
4
5 class MinGradNormGDSC : public StopCriterion {
6 private:
7     double min_grad_norm;
8 public:
9     MinGradNormGDSC();
10    MinGradNormGDSC(unsigned int max_n_iter_, double min_grad_norm_);
11    MinGradNormGDSC(const MinGradNormGDSC& other);
12

```

```

17     MinGradNormGDSC(MinGradNormGDSC&& other) noexcept;
18
19     void swap(MinGradNormGDSC& other) noexcept;
20     MinGradNormGDSC& operator=(MinGradNormGDSC other) noexcept;
24     bool criterion(OptInfo* status);
25     double get_min_grad_norm() const;
26     ~MinGradNormGDSC();
27 };
28

```

5.9 MinRelImpNormGDSC.h

```

1 #pragma once
2 #include "StopCriterion.h"
3 #include "GDOptInfo.h"
4
10 class MinRelImpNormGDSC : public StopCriterion {
11 private:
13     double min_rel_imp_norm;
14 public:
15     MinRelImpNormGDSC();
16     MinRelImpNormGDSC(unsigned int max_n_iter_, double min_rel_imp_norm_);
17     MinRelImpNormGDSC(const MinRelImpNormGDSC& other);
18     MinRelImpNormGDSC(MinRelImpNormGDSC&& other) noexcept;
19
20     void swap(MinRelImpNormGDSC& other) noexcept;
21     MinRelImpNormGDSC& operator=(MinRelImpNormGDSC other) noexcept;
25     bool criterion(OptInfo* status);
26     double get_min_rel_imp_norm() const;
27     ~MinRelImpNormGDSC();
28 };
29

```

5.10 MinStepNormGDSC.h

```

1 #pragma once
2 #include "StopCriterion.h"
3 #include "GDOptInfo.h"
4
10 class MinStepNormGDSC : public StopCriterion {
11 private:
13     double min_step_norm;
14 public:
15     MinStepNormGDSC();
16     MinStepNormGDSC(unsigned int max_n_iter_, double min_step_norm_);
17     MinStepNormGDSC(const MinStepNormGDSC& other);
18     MinStepNormGDSC(MinStepNormGDSC&& other) noexcept;
19
20     void swap(MinStepNormGDSC& other) noexcept;
21     MinStepNormGDSC& operator=(MinStepNormGDSC other) noexcept;
25     bool criterion(OptInfo* status);
26     double get_min_step_norm() const;
27     ~MinStepNormGDSC();
28 };
29

```

5.11 NormDiffSSC.h

```

1 #pragma once
2 #include "StopCriterion.h"
3 #include "StohOptInfo.h"
4
10 class NormDiffSSC : public StopCriterion {
11 private:
13     double min_last_imp_norm;
14 public:
15     NormDiffSSC();
16     NormDiffSSC(unsigned int max_n_iter_, double min_last_imp_norm_);
17     NormDiffSSC(const NormDiffSSC& other);
18     NormDiffSSC(NormDiffSSC&& other) noexcept;
19
20     void swap(NormDiffSSC& other) noexcept;
21     NormDiffSSC& operator=(NormDiffSSC other) noexcept;
25     bool criterion(OptInfo* opt_info);
26     double get_min_last_imp_norm() const;
27     ~NormDiffSSC();
28 };
29

```

5.12 OptimizationMethod.h

```

1 #pragma once
2 #include "StopCriterion.h"
3 #include <stdexcept>
4 #include <cmath>
5 #include <stdexcept>
6 #include "BoxArea.h"
7 #include "Function.h"
8 #include "OptInfo.h"
9 #include "OptResult.h"
10 #include <string>
11
12 class OptimizationMethod
13 {
14 protected:
15     OptimizationMethod();
16     OptimizationMethod(Function* func_, BoxArea* box_area_, StopCriterion* stop_criterion_);
17     OptimizationMethod(const OptimizationMethod& other);
18     OptimizationMethod(OptimizationMethod&& other) noexcept;
19
20     void swap(OptimizationMethod& other) noexcept;
21     std::vector<std::vector<double> > point_history;
22     unsigned int n_iter = 0;
23     unsigned int n_dim = 0;
24     BoxArea* box_area = nullptr;
25     Function* func = nullptr;
26     StopCriterion* stop_criterion = nullptr;
27     virtual void make_step() = 0;
28 public:
29     virtual OptInfo* get_opt_info() const = 0;
30     virtual OptResult optimize() = 0;
31     virtual ~OptimizationMethod();
32     unsigned int get_n_dim() const;
33     unsigned int get_n_iter() const;
34 };
35

```

5.13 OptInfo.h

```

1 #pragma once
2 #include <stdexcept>
3
4 class OptInfo {
5 protected:
6     unsigned int n_iter;
7 public:
8     OptInfo();
9     OptInfo(const OptInfo& other);
10    OptInfo(OptInfo&& other) noexcept;
11    OptInfo(unsigned int n_iter_);
12
13    void swap(OptInfo& other) noexcept;
14    OptInfo& operator=(OptInfo other);
15    unsigned int get_n_iter() const;
16    virtual ~OptInfo() {};
17 };
18

```

5.14 OptResult.h

```

1 #pragma once
2 #include <vector>
3
4 class OptResult
5 {
6 private:
7     unsigned int n_iter;
8     std::vector<double> min_point;
9     double min_value;
10 public:
11     OptResult();
12     OptResult(const std::vector<double>& min_point_, double min_value_, unsigned int n_iter_);
13     OptResult(const OptResult& other);
14     OptResult(OptResult&& other) noexcept;
15
16     void swap(OptResult& other) noexcept;
17     OptResult& operator=(OptResult other) noexcept;
18     unsigned int get_n_iter() const;
19

```

```

25     std::vector<double> get_min_point() const;
26     double get_min_value() const;
27 };
28

```

5.15 StochasticOptimization.h

```

1 #pragma once
2 #include <float.h>
3 #include "OptimizationMethod.h"
4 #include "IterOnlySSC.h"
5 #include "IterAfterImpSSC.h"
6 #include "NormDiffSSC.h"
7 #include <random>
8 #include <chrono>
9
13 class StochasticOptimization : public OptimizationMethod
14 {
15 private:
16     std::vector<double> first_point;
17     double curr_min_value;
18     double delta;
19     double p;
20     unsigned int n_iter_after_improvement;
21     double last_imp_norm = DBL_MAX;
22     std::uniform_real_distribution<double> distribution;
23     std::default_random_engine generator;
24     unsigned long long seed;
25     double unif(double lower, double upper);
26
27 public:
28     StochasticOptimization();
29     StochasticOptimization(Function* func_, BoxArea* box_area_, StopCriterion* stop_criterion_, const
std::vector<double>& first_point_,
30         double delta_, double p_);
31     StochasticOptimization(const StochasticOptimization& other);
32     StochasticOptimization(StochasticOptimization&& other);
33
34     void swap(StochasticOptimization& other) noexcept;
35     StochasticOptimization& operator=(StochasticOptimization other) noexcept;
36     void set_first_point(const std::vector<double>& point);
37     void make_step();
38     OptResult optimize();
39     OptInfo* get_opt_info() const;
40     ~StochasticOptimization();
41 };
42

```

5.16 StohOptInfo.h

```

1 #pragma once
2 #include "OptInfo.h"
3
7 class StohOptInfo : public OptInfo {
8 private:
9     unsigned int n_iter_after_imp;
10     double last_imp_norm;
11
12 public:
13     StohOptInfo();
14     StohOptInfo(unsigned int n_iter_, unsigned int n_iter_after_imp, double last_imp_norm_);
15     StohOptInfo(const StohOptInfo& other);
16     StohOptInfo(StohOptInfo&& other) noexcept;
17
18     void swap(StohOptInfo& other) noexcept;
19     StohOptInfo& operator=(StohOptInfo other) noexcept;
20     unsigned int get_n_iter_after_imp() const;
21     double get_last_imp_norm() const;
22     ~StohOptInfo() {};
23 };
24

```

5.17 StopCriterion.h

```

1 #pragma once
2 #include "OptInfo.h"
3 #include <vector>

```



```
4
9 class StopCriterion
10 {
11 protected:
12     unsigned int max_n_iter;
13
14     StopCriterion();
15     StopCriterion(unsigned int max_n_iter_);
16     StopCriterion(const StopCriterion& other);
17     StopCriterion(StopCriterion&& other) noexcept;
18
19     void swap(StopCriterion& other) noexcept;
20 public:
21     virtual bool criterion(OptInfo* opt_info) = 0;
22     bool operator()(OptInfo* opt_info);
23     virtual ~StopCriterion() {};
24 };
25
26
```


Chapter 6

Example Documentation

6.1 main.cpp

An example of gradient descent

```
/* \file main.cpp
 *
 */
#define _CRT_SECURE_NO_WARNINGS
#define _USE_MATH_DEFINES
#include <iostream>
#include <math.h>
#include <cstdio>
#include "../StochasticOptimization.h"
#include "../GradientDescent.h"
//https://en.wikipedia.org/wiki/Rosenbrock_function
// Minimum point: {1, 1}
double rosenbrock_func2(std::vector<double> arg) {
    return pow(1 - arg[0], 2) + 100 * pow(arg[1] - pow(arg[0], 2), 2);
}
std::vector<double> rosenbrock_func2_grad(std::vector<double> arg) {
    return std::vector<double>({ 2 * (200 * pow(arg[0], 3) - 200 * arg[0] * arg[1] + arg[0] - 1), 200 *
        (arg[1] - pow(arg[0], 2)) });
}
int main() {
    /*
    * Lets perform a stochastic optimization of Rosenbrock function.
    */
    dimensional_limits x_limits(-10, 10), y_limits(-10, 10); // Dimensional limits
    BoxArea box_area(2, { x_limits, y_limits }); // Area of search
    std::vector<double> first_point = {-4.5, 3.4}; // Initial point of search
    Function func(rosenbrock_func2, 2); // Function that we will optimize
    /*
    * Lets set stop criterion. Let the maximum number of iterations be 5000 and maximum number of iterations
    after last improvement be 200.
    */
    StopCriterion* stop_criterion = new IterAfterImpSSC(5000, 200);
    double delta = 1; // Delta parameter
    double p = 0.5; // Probability parameter
    /*
    * Lets set optimization method with the parameters.
    */
    OptimizationMethod* optimization_method = new StochasticOptimization(&func, &box_area, stop_criterion,
        first_point, delta, p);
    /*
    * Perform optimization.
    */
    OptResult opt_result = optimization_method->optimize();
    std::cout << "Optimization result: \n";
    std::cout << "\t Iteration num: " << opt_result.get_n_iter();
    std::cout << "\n\t Optimal point: ";
    std::vector<double> opt_point = opt_result.get_min_point();
    // Print coordinates of min point.
    for (int i = 0; i < opt_point.size(); ++i) {
        std::cout << opt_point[i] << ' ';
    }
    std::cout << "\n\t Function value: " << opt_result.get_min_value() << "\n";
    delete optimization_method;
    delete stop_criterion;
    /*
```

```

* Now lets perform a gradient descent
*/
    dimensional_limits x_limits2(-5, 5), y_limits2(-10, 15); // Dimensional limits
    BoxArea box_area2(2, { x_limits2, y_limits2 }); // Area of search
    std::vector<double> first_point2 = { -3, 3 }; // Initial point of search
    Function func2(rosenbrock_func2, 2); // Function that we will optimize
    MDFunction grad(rosenbrock_func2_grad, 2); // Gradient of function
    std::string norm_name = "l2"; // Lets use l2 norm
    unsigned int steps = 100; // Lets do 100 steps on each iteration
    /*
* Lets set stop criterion. Let the maximum number of iterations be 5000 and min grad norm threshold be
    1e-6.
*/
    StopCriterion* stop_criterion2 = new MinGradNormGDSC(5000, 1e-6);
    /*
* Lets set optimization method with the parameters
*/
    OptimizationMethod* optimization_method2 = new GradientDescent(&func2, &grad, &box_area2,
        stop_criterion2, first_point2, steps, norm_name);
    /*
* Perform optimization.
*/
    opt_result = optimization_method2->optimize();
    std::cout << "Optimization result:  \n";
    std::cout << "\t Iteration num:  " << opt_result.get_n_iter();
    std::cout << "\n\t Optimal point:  ";
    opt_point = opt_result.get_min_point();
    // Print coordinates of min point.
    for (int i = 0; i < opt_point.size(); ++i) {
        std::cout << opt_point[i] << ' ';
    }
    std::cout << "\n\t Function value:  " << opt_result.get_min_value() << "\n";
    delete optimization_method2;
    delete stop_criterion2;
    return 0;
}

```

Index

- ~GradientDescent
 - GradientDescent, [12](#)
- BoxArea, [7](#)
 - is_in, [8](#)
- criterion
 - IterAfterImpSSC, [14](#)
 - IterOnlySSC, [15](#)
 - MinGradNormGDSC, [17](#)
 - MinRelImpNormGDSC, [18](#)
 - MinStepNormGDSC, [19](#)
 - NormDiffSSC, [20](#)
 - StopCriterion, [27](#)
- dimensional_limits, [9](#)
- Function, [9](#)
 - operator(), [10](#)
- GDOptInfo, [11](#)
- get_opt_info
 - GradientDescent, [12](#)
 - StochasticOptimization, [24](#)
- GradientDescent, [11](#)
 - ~GradientDescent, [12](#)
 - get_opt_info, [12](#)
 - make_step, [12](#)
 - optimize, [13](#)
- is_in
 - BoxArea, [8](#)
- IterAfterImpSSC, [13](#)
 - criterion, [14](#)
- IterOnlySSC, [14](#)
 - criterion, [15](#)
- make_step
 - GradientDescent, [12](#)
 - OptimizationMethod, [21](#)
 - StochasticOptimization, [24](#)
- MDFunction, [15](#)
 - operator(), [16](#)
- MinGradNormGDSC, [16](#)
 - criterion, [17](#)
- MinRelImpNormGDSC, [17](#)
 - criterion, [18](#)
- MinStepNormGDSC, [18](#)
 - criterion, [19](#)
- NormDiffSSC, [19](#)
- criterion, [20](#)
- operator()
 - Function, [10](#)
 - MDFunction, [16](#)
 - StopCriterion, [27](#)
- OptimizationMethod, [20](#)
 - make_step, [21](#)
 - optimize, [22](#)
- optimize
 - GradientDescent, [13](#)
 - OptimizationMethod, [22](#)
 - StochasticOptimization, [25](#)
- OptInfo, [22](#)
- OptResult, [23](#)
- StochasticOptimization, [24](#)
 - get_opt_info, [24](#)
 - make_step, [24](#)
 - optimize, [25](#)
- StohOptInfo, [25](#)
- StopCriterion, [26](#)
 - criterion, [27](#)
 - operator(), [27](#)