ROL

Generated by Doxygen 1.6.1

Wed Dec 11 11:39:40 2013

Contents

Chapter 1

ROL Documentation (Development Version)

Figure 1.1: Rapid Optimization Library

1.1 Introduction

ROL, the Rapid Optimization Library, is a Trilinos package for matrix-free optimization.

1.2 Overview

Current release of ROL includes the following features:

- Unconstrained optimization algorithms.

1.3 Quick Start

The Rosenbrock example (rol/example/rosenbrock/example_01.cpp) demonstrates the use of ROL. It amounts to sixsteps:

1.3.1 Step 1: Implement linear algebra / vector interface.

--- or try one of the provided implementations, such as ROL::StdVector in rol/vector.

```
ROL::Vector
```

1.3.2 Step 2: Implement objective function interface.

--- or try one of the provided functions, such as ROL::Objective_Rosenbrock in rol/zoo.

```
ROL::Objective
```

1.3.3 Step 3: Choose optimization step.

--- with ParameterList settings in the variable parlist.

```
ROL::LineSearchStep<RealT> step(parlist);
```

1.3.4 Step 4: Set status test.

--- with gradient tolerance {gtol}, step tolerance {stol} and the maximum number of iterations {maxit}.

```
ROL::StatusTest<RealT> status(gtol, stol, maxit);
```

1.3.5 Step 5: Define an algorithm.

--- based on the status test and the step.

```
ROL::DefaultAlgorithm<RealT> algo(step,status);
```

1.3.6 Step 6: Run algorithm.

--- starting from the initial iterate {x}, applied to objective function {obj}.

```
algo.run(x, obj);
```

1.3.7 Done!

1.4 Development Plans

Constrained optimization, optimization under uncertainty, etc.

Chapter 2

Class Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

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

Chapter 5

Class Documentation

5.1 ROL::Algorithm Class Reference

Provides an interface to run optimization algorithms.

`#include <ROL_Algorithm.hpp>`

5.1.1 Detailed Description

Provides an interface to run optimization algorithms.

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

- ROL_Algorithm.hpp

5.2 ROL::AlgorithmState< Real > Struct Template Reference

Public Attributes

- int **iter**
- int **nfval**
- int **ngrad**
- Real **value**
- Real **gnorm**
- Real **snorm**
- Teuchos::RCP< Vector< Real > > **iterateVec**

5.2.1 Detailed Description

template<class Real> struct ROL::AlgorithmState< Real >

Definition at line 58 of file ROL_Step.hpp.

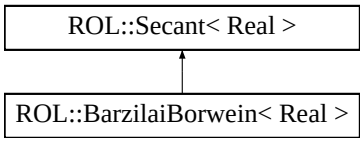
The documentation for this struct was generated from the following file:

- ROL_Step.hpp

5.3 ROL::BarzilaiBorwein< Real > Class Template Reference

Provides definitions for Barzilai-Borwein operators.

`#include <ROL_BarzilaiBorwein.hpp>`Inheritance diagram for ROL::BarzilaiBorwein< Real >::



Public Member Functions

- **BarzilaiBorwein** (int type=1)
- void **applyH** (Vector< Real > &Hv, const Vector< Real > &v, const Vector< Real > &x)
- void **applyB** (Vector< Real > &Bv, const Vector< Real > &v, const Vector< Real > &x)

Private Attributes

- int **type_**

5.3.1 Detailed Description

template<class Real> class ROL::BarzilaiBorwein< Real >

Provides definitions for Barzilai-Borwein operators.

Definition at line 54 of file ROL_BarzilaiBorwein.hpp.

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

- ROL_BarzilaiBorwein.hpp

5.4 ROL::DefaultAlgorithm< Real > Class Template Reference

Public Member Functions

- **DefaultAlgorithm** (Step< Real > &step, StatusTest< Real > &status, bool printHeader=false)
- virtual std::vector< std::string > run (Vector< Real > &x, Objective< Real > &obj, bool print=false)

  Run algorithm.

- std::string **getIterHeader** (void)
- std::string **getIterInfo** (bool withHeader=false)

Private Attributes

- Teuchos::RCP< Step< Real > > **step_**
- Teuchos::RCP< StatusTest< Real > > **status_**
- Teuchos::RCP< AlgorithmState< Real > > **state_**
- bool **printHeader_**

5.4.1 Detailed Description

template<class Real> class ROL::DefaultAlgorithm< Real >

Definition at line 58 of file ROL_Algorithm.hpp.

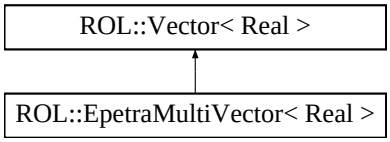
The documentation for this class was generated from the following file:

- ROL_Algorithm.hpp

5.5 ROL::EpetraMultiVector< Real > Class Template Reference

Implements the ROL::Vector interface for an Epetra_MultiVector.

`#include <ROL_EpetraMultiVector.hpp>`Inheritance diagram for ROL::EpetraMultiVector< Real >::



Public Member Functions

- **EpetraMultiVector** (const Teuchos::RCP< Epetra_MultiVector > &epetra_vec)
- void plus (const Vector< Real > &x)

  *Compute $y \leftarrow x + y$ where $y =$ *this.*

- void scale (const Real alpha)

  *Compute $y \leftarrow \alpha y$ where $y =$ *this.*

- Real dot (const Vector< Real > &x) const

  *Returns $\langle y, x \rangle$ where $y =$ *this.*

- Real norm () const

  *Returns $\|y\|$ where $y =$ *this.*

- Teuchos::RCP< Vector< Real > > clone () const

  Clone to make a new (uninitialized) vector.

- virtual void axpy (const Real alpha, const Vector< Real > &x)

  *Compute $y \leftarrow \alpha x + y$ where $y =$ *this.*

- virtual void zero ()

  Set to zero vector.

- virtual void set (const Vector< Real > &x)

  *Set $y \leftarrow x$ where $y =$ *this.*

- Teuchos::RCP< const Epetra_MultiVector > **getVector** () const
- Teuchos::RCP< Vector< Real > > basis (const int i) const

  Return i-th basis vector: define if finite-difference gradients and Hessians are used.

- int **dimension** () const

Private Attributes

- Teuchos::RCP< Epetra_MultiVector > **epetra_vec_**

5.5.1 Detailed Description

template<class Real> class ROL::EpetraMultiVector< Real >

Implements the ROL::Vector interface for an Epetra_MultiVector.

Definition at line 61 of file ROL_EpetraMultiVector.hpp.

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

- ROL_EpetraMultiVector.hpp

5.6 ROL::Krylov< Real > Class Template Reference

Provides definitions for Krylov solvers.

`#include <ROL_Krylov.hpp>`

Public Member Functions

- **Krylov** (Real tol1=1.e-4, Real tol2=1.e-2, int maxit=100, bool useInexact=false)
- void **CG** (Vector< Real > &s, int &iter, int &flag, const Vector< Real > &g, const Vector< Real > &x, Objective< Real > &obj, Teuchos::RCP< Secant< Real > > secant=Teuchos::null)

Private Attributes

- Real **tol1_**
- Real **tol2_**
- int **maxit_**
- bool **useInexact_**

5.6.1 Detailed Description

template<class Real> class ROL::Krylov< Real >

Provides definitions for Krylov solvers.

Definition at line 54 of file ROL_Krylov.hpp.

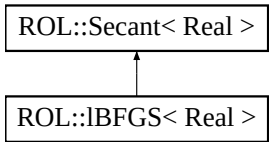
The documentation for this class was generated from the following file:

- ROL_Krylov.hpp

5.7 ROL::lBFGS< Real > Class Template Reference

Provides definitions for limited-memory BFGS operators.

`#include <ROL_lBFGS.hpp>`Inheritance diagram for ROL::lBFGS< Real >::



Public Member Functions

- **lBFGS** (int M)
- void **applyH** (Vector< Real > &Hv, const Vector< Real > &v, const Vector< Real > &x)
- void **applyB** (Vector< Real > &Bv, const Vector< Real > &v, const Vector< Real > &x)

5.7.1 Detailed Description

template<class Real> class ROL::lBFGS< Real >

Provides definitions for limited-memory BFGS operators.

Definition at line 54 of file ROL_lBFGS.hpp.

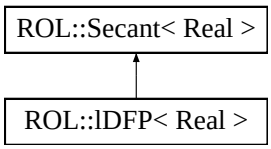
The documentation for this class was generated from the following file:

- ROL_lBFGS.hpp

5.8 ROL::lDFP< Real > Class Template Reference

Provides definitions for limited-memory DFP operators.

`#include <ROL_lDFP.hpp>`Inheritance diagram for ROL::lDFP< Real >::



Public Member Functions

- **lDFP** (int M)
- void **applyH** (Vector< Real > &Hv, const Vector< Real > &v, const Vector< Real > &x)
- virtual void **applyH0** (Vector< Real > &Hv, const Vector< Real > &v, const Vector< Real > &x)
- void **applyB** (Vector< Real > &Bv, const Vector< Real > &v, const Vector< Real > &x)
- virtual void **applyB0** (Vector< Real > &Bv, const Vector< Real > &v, const Vector< Real > &x)

5.8.1 Detailed Description

template<class Real> class ROL::lDFP< Real >

Provides definitions for limited-memory DFP operators.

Definition at line 54 of file ROL_lDFP.hpp.

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

- ROL_lDFP.hpp

5.9 ROL::LineSearch< Real > Class Template Reference

Provides interface for and implements line searches.

`#include <ROL_LineSearch.hpp>`

Public Member Functions

- **LineSearch** (Teuchos::ParameterList &parlist)
- bool **status** (const ELineSearch type, int &ls_neval, int &ls_ngrad, const Real alpha, const Real fold, const Real sgold, const Real fnew, const Vector< Real > &x, const Vector< Real > &s, Objective< Real > &obj)
- void **run** (Real &alpha, Real &fval, int &ls_neval, int &ls_ngrad, const Real &gs, const Vector< Real > &s, const Vector< Real > &x, Objective< Real > &obj)
- void **simplebacktracking** (Real &alpha, Real &fval, int &ls_neval, int &ls_ngrad, const Real &gs, const Vector< Real > &s, const Vector< Real > &x, Objective< Real > &obj)
- void **backtracking** (Real &alpha, Real &fval, int &ls_neval, int &ls_ngrad, const Real &gs, const Vector< Real > &s, const Vector< Real > &x, Objective< Real > &obj)
- void **bisection** (Real &alpha, Real &fval, int &ls_neval, int &ls_ngrad, const Real &gs, const Vector< Real > &s, const Vector< Real > &x, Objective< Real > &obj)
- void **goldensection** (Real &alpha, Real &fval, int &ls_neval, int &ls_ngrad, const Real &gs, const Vector< Real > &s, const Vector< Real > &x, Objective< Real > &obj)
- void **brents** (Real &alpha, Real &fval, int &ls_neval, int &ls_ngrad, const Real &gs, const Vector< Real > &s, const Vector< Real > &x, Objective< Real > &obj)

Private Attributes

- ELineSearch **els_**
- ECurvatureCondition **econd_**
- EDescent **edesc_**
- int **maxit_**
- Real **c1_**
- Real **c2_**
- Real **tol_**
- Real **rho_**
- Real **alpha0_**
- bool **useralpha_**

5.9.1 Detailed Description

template<class Real> class ROL::LineSearch< Real >

Provides interface for and implements line searches.

Definition at line 54 of file ROL_LineSearch.hpp.

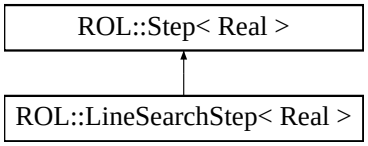
The documentation for this class was generated from the following file:

- ROL_LineSearch.hpp

5.10 ROL::LineSearchStep< Real > Class Template Reference

Provides the interface to compute optimization steps with line search.

`#include <ROL_LineSearchStep.hpp>`Inheritance diagram for ROL::LineSearchStep< Real >::



Public Member Functions

- **LineSearchStep** (Teuchos::ParameterList &parlist)
- **LineSearchStep** (Teuchos::RCP< Secant< Real > > &secant, Teuchos::ParameterList &parlist)
- void compute (Vector< Real > &s, const Vector< Real > &x, Objective< Real > &obj, AlgorithmState< Real > &algo_state)

  Compute step.

- void update (Vector< Real > &x, const Vector< Real > &s, Objective< Real > &obj, AlgorithmState< Real > &algo_state)

  Update step, if successful.

- std::string printHeader (void) const

  Print iterate header.

- std::string printName (void) const

  Print step name.

- std::string print (AlgorithmState< Real > &algo_state, bool printHeader=false) const

  Print iterate status.

Private Attributes

- Teuchos::RCP< Secant< Real > > **secant_**
- Teuchos::RCP< Krylov< Real > > **krylov_**

- Teuchos::RCP< NonlinearCG< Real > > **nlcg_**
- Teuchos::RCP< LineSearch< Real > > **lineSearch_**
- int **iterKrylov_**
- int **flagKrylov_**
- ELineSearch **els_**
- ECurvatureCondition **econd_**
- EDescent **edesc_**
- ESecant **esec_**
- int **ls_nfval_**
- int **ls_ngrad_**
- std::vector< bool > **useInexact_**

5.10.1 Detailed Description

template<class Real> class ROL::LineSearchStep< Real >

Provides the interface to compute optimization steps with line search.

Definition at line 65 of file ROL_LineSearchStep.hpp.

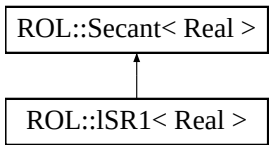
The documentation for this class was generated from the following file:

- ROL_LineSearchStep.hpp

5.11 ROL::lSR1< Real > Class Template Reference

Provides definitions for limited-memory SR1 operators.

`#include <ROL_lSR1.hpp>`Inheritance diagram for ROL::lSR1< Real >::



Public Member Functions

- **lSR1** (int M)
- void **update** (const Vector< Real > &grad, const Vector< Real > &gp, const Vector< Real > &s, const Real snorm, const int iter)
- virtual void **applyH0** (Vector< Real > &Hv, const Vector< Real > &v, const Vector< Real > &x)
- void **applyH** (Vector< Real > &Hv, const Vector< Real > &v, const Vector< Real > &x)
- virtual void **applyB0** (Vector< Real > &Bv, const Vector< Real > &v, const Vector< Real > &x)
- void **applyB** (Vector< Real > &Bv, const Vector< Real > &v, const Vector< Real > &x)

Private Attributes

- bool **updateIterate_**

5.11.1 Detailed Description

template<class Real> class ROL::lSR1< Real >

Provides definitions for limited-memory SR1 operators.

Definition at line 54 of file ROL_lSR1.hpp.

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

- ROL_lSR1.hpp

5.12 ROL::NonlinearCG< Real > Class Template Reference

Implementats nonlinear conjugate gradient methods.

`#include <ROL_NonlinearCG.hpp>`

Public Member Functions

- **NonlinearCG** (ENonlinearCG type, int restart=100)
- Teuchos::RCP< NonlinearCGState< Real > > & **get_state** ()
- virtual void **run** (Vector< Real > &s, const Vector< Real > &g, const Vector< Real > &x, Objective< Real > &obj)

Private Attributes

- Teuchos::RCP< NonlinearCGState< Real > > **state_**

5.12.1 Detailed Description

template<class Real> class ROL::NonlinearCG< Real >

Implementats nonlinear conjugate gradient methods.

Definition at line 65 of file ROL_NonlinearCG.hpp.

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

- ROL_NonlinearCG.hpp

5.13 ROL::NonlinearCGState< Real > Struct Template Reference

Public Attributes

- std::vector< Teuchos::RCP< Vector< Real > > > **grad**
- std::vector< Teuchos::RCP< Vector< Real > > > **pstep**
- int **iter**
- int **restart**
- ENonlinearCG **nlcg_type**

5.13.1 Detailed Description

template<class Real> struct ROL::NonlinearCGState< Real >

Definition at line 56 of file ROL_NonlinearCG.hpp.

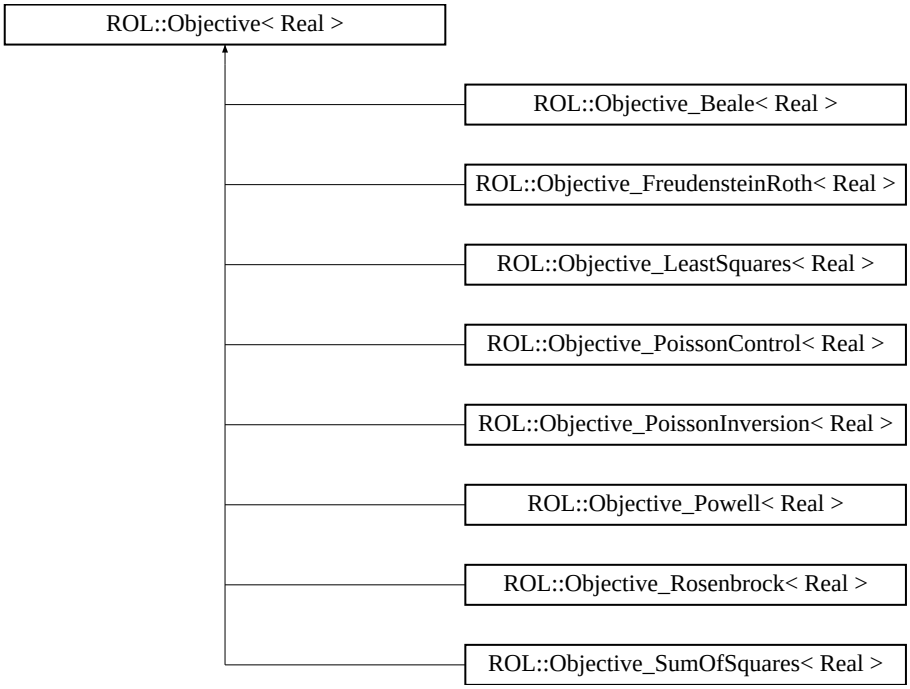
The documentation for this struct was generated from the following file:

- ROL_NonlinearCG.hpp

5.14 ROL::Objective< Real > Class Template Reference

Provides the interface to evaluate objective functions.

`#include <ROL_Objective.hpp>`Inheritance diagram for ROL::Objective< Real >::



Public Member Functions

- virtual Real value (const Vector< Real > &x, Real &tol)=0

  Compute value.

- virtual void gradient (Vector< Real > &g, const Vector< Real > &x, Real &tol)

  Compute gradient.

- virtual Real dirDeriv (const Vector< Real > &x, const Vector< Real > &d, Real &tol)

  Compute directional derivative.

- virtual void hessVec (Vector< Real > &hv, const Vector< Real > &v, const Vector< Real > &x, Real &tol)

  Apply Hessian approximation to vector.

- virtual void invHessVec (Vector< Real > &hv, const Vector< Real > &v, const Vector< Real > &x, Real &tol)

  Apply inverse Hessian approximation to vector.

- virtual void precond (Vector< Real > &Pv, const Vector< Real > &v, const Vector< Real > &x)

  Apply preconditioner to vector.

- virtual std::vector< std::vector< Real > > checkGradient (const Vector< Real > &x, const Vector< Real > &d, const bool printToScreen=true, const int numSteps=ROL_NUM_CHECKDERIV_STEPS)

  Finite-difference gradient check.

- virtual std::vector< std::vector< Real > > checkHessVec (const Vector< Real > &x, const Vector< Real > &v, const bool printToScreen=true, const int numSteps=ROL_NUM_CHECKDERIV_STEPS)

  Finite-difference Hessian-applied-to-vector check.

5.14.1 Detailed Description

template<class Real> class ROL::Objective< Real >

Provides the interface to evaluate objective functions. Provides the definition of the objective function interface.

Definition at line 59 of file ROL_Objective.hpp.

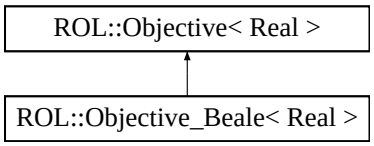
The documentation for this class was generated from the following files:

- ROL_Objective.hpp
- ROL_ObjectiveDef.hpp

5.15 ROL::Objective_Beale< Real > Class Template Reference

Beale's function.

`#include <ROL_Beale.hpp>`Inheritance diagram for ROL::Objective_Beale< Real >::



Public Member Functions

- Real value (const Vector< Real > &x, Real &tol)

  Compute value.

- void gradient (Vector< Real > &g, const Vector< Real > &x, Real &tol)

  Compute gradient.

- void hessVec (Vector< Real > &hv, const Vector< Real > &v, const Vector< Real > &x, Real &tol)

  Apply Hessian approximation to vector.

- void invHessVec (Vector< Real > &hv, const Vector< Real > &v, const Vector< Real > &x, Real &tol)

  Apply inverse Hessian approximation to vector.

Private Attributes

- std::vector< Real > **y_**

5.15.1 Detailed Description

template<class Real> class ROL::Objective_Beale< Real >

Beale's function.

Definition at line 64 of file ROL_Beale.hpp.

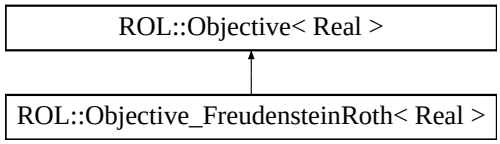
The documentation for this class was generated from the following file:

- ROL_Beale.hpp

5.16 ROL::Objective_FreudensteinRoth< Real > Class Template Reference

Freudenstein and Roth's function.

`#include <ROL_FreudensteinRoth.hpp>`Inheritance diagram for ROL::Objective_FreudensteinRoth< Real >::



Public Member Functions

- Real value (const Vector< Real > &x, Real &tol)

  Compute value.

- void gradient (Vector< Real > &g, const Vector< Real > &x, Real &tol)

  Compute gradient.

- void hessVec (Vector< Real > &hv, const Vector< Real > &v, const Vector< Real > &x, Real &tol)

  Apply Hessian approximation to vector.

- void invHessVec (Vector< Real > &hv, const Vector< Real > &v, const Vector< Real > &x, Real &tol)

  Apply inverse Hessian approximation to vector.

5.16.1 Detailed Description

template<class Real> class ROL::Objective_FreudensteinRoth< Real >

Freudenstein and Roth's function.

Definition at line 64 of file ROL_FreudensteinRoth.hpp.

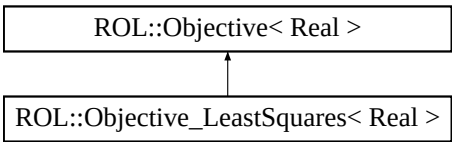
The documentation for this class was generated from the following file:

- ROL_FreudensteinRoth.hpp

5.17 ROL::Objective_LeastSquares< Real > Class Template Reference

Least squares function.

`#include <ROL_LeastSquares.hpp>`Inheritance diagram for ROL::Objective_LeastSquares< Real >::



Public Member Functions

- Real value (const Vector< Real > &x, Real &tol)

  Compute value.

- void gradient (Vector< Real > &g, const Vector< Real > &x, Real &tol)

  Compute gradient.

- void hessVec (Vector< Real > &hv, const Vector< Real > &v, const Vector< Real > &x, Real &tol)

  Apply Hessian approximation to vector.

5.17.1 Detailed Description

template<class Real> class ROL::Objective_LeastSquares< Real >

Least squares function.

Definition at line 64 of file ROL_LeastSquares.hpp.

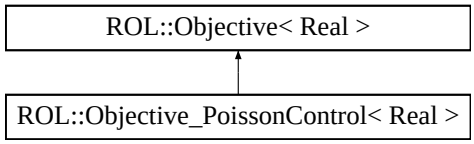
The documentation for this class was generated from the following file:

- ROL_LeastSquares.hpp

5.18 ROL::Objective_PoissonControl< Real > Class Template Reference

Poisson distributed control.

`#include <ROL_PoissonControl.hpp>`Inheritance diagram for ROL::Objective_PoissonControl< Real >::



Public Member Functions

- **Objective_PoissonControl** (Real alpha=1.e-4)
- void **apply_mass** (Vector< Real > &Mz, const Vector< Real > &z)
- void **solve_poisson** (Vector< Real > &u, const Vector< Real > &z)
- Real **evaluate_target** (Real x)
- Real value (const Vector< Real > &z, Real &tol)

  Compute value.

- void gradient (Vector< Real > &g, const Vector< Real > &z, Real &tol)

  Compute gradient.

- void hessVec (Vector< Real > &hv, const Vector< Real > &v, const Vector< Real > &z, Real &tol)

  Apply Hessian approximation to vector.

Private Attributes

- Real **alpha_**

5.18.1 Detailed Description

template<class Real> class ROL::Objective_PoissonControl< Real >

Poisson distributed control.

Definition at line 64 of file ROL_PoissonControl.hpp.

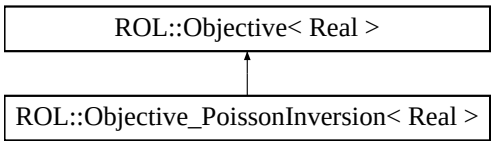
The documentation for this class was generated from the following file:

- ROL_PoissonControl.hpp

5.19 ROL::Objective_PoissonInversion< Real > Class Template Reference

Poisson material inversion.

`#include <ROL_PoissonInversion.hpp>`Inheritance diagram for ROL::Objective_PoissonInversion< Real >::



Public Member Functions

- **Objective_PoissonInversion** (int nz=32, Real alpha=1.e-4)
- Real **reg_value** (const Vector< Real > &z)
- void **reg_gradient** (Vector< Real > &g, const Vector< Real > &z)
- void **reg_hessVec** (Vector< Real > &hv, const Vector< Real > &v, const Vector< Real > &z)
- void **apply_mass** (Vector< Real > &Mf, const Vector< Real > &f)
- void **solve_poisson** (Vector< Real > &u, const Vector< Real > &z, Vector< Real > &b)
- Real **evaluate_target** (Real x)
- void **apply_linearized_control_operator** (Vector< Real > &Bd, const Vector< Real > &z, const Vector< Real > &d, const Vector< Real > &u)
- void **apply_transposed_linearized_control_operator** (Vector< Real > &Bd, const Vector< Real > &z, const Vector< Real > &d, const Vector< Real > &u)
- void **apply_transposed_linearized_control_operator_2** (Vector< Real > &Bd, const Vector< Real > &z, const Vector< Real > &v, const Vector< Real > &d, const Vector< Real > &u)
- void **solve_state_equation** (Vector< Real > &u, const Vector< Real > &z)
- void **solve_adjoint_equation** (Vector< Real > &p, const Vector< Real > &u, const Vector< Real > &z)
- void **solve_state_sensitivity_equation** (Vector< Real > &w, const Vector< Real > &v, const Vector< Real > &u, const Vector< Real > &z)
- void **solve_adjoint_sensitivity_equation** (Vector< Real > &q, const Vector< Real > &w, const Vector< Real > &v, const Vector< Real > &p, const Vector< Real > &u, const Vector< Real > &z)
- Real value (const Vector< Real > &z, Real &tol)

  Compute value.

- void gradient (Vector< Real > &g, const Vector< Real > &z, Real &tol)

  Compute gradient.

- void hessVec (Vector< Real > &hv, const Vector< Real > &v, const Vector< Real > &z, Real &tol)

  Apply Hessian approximation to vector.

Private Attributes

- int **nu_**
- int **nz_**
- Real **hu_**
- Real **hz_**
- Real **alpha_**
- Real **eps_**
- int **reg_type_**

5.19.1 Detailed Description

template<class Real> class ROL::Objective_PoissonInversion< Real >

Poisson material inversion.

Definition at line 66 of file ROL_PoissonInversion.hpp.

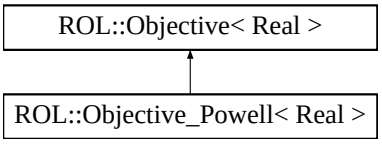
The documentation for this class was generated from the following file:

- ROL_PoissonInversion.hpp

5.20 ROL::Objective_Powell< Real > Class Template Reference

Powell's badly scaled function.

`#include <ROL_Powell.hpp>`Inheritance diagram for ROL::Objective_-Powell< Real >::



Public Member Functions

- Real value (const Vector< Real > &x, Real &tol)

  Compute value.

- void gradient (Vector< Real > &g, const Vector< Real > &x, Real &tol)

  Compute gradient.

- void hessVec (Vector< Real > &hv, const Vector< Real > &v, const Vector< Real > &x, Real &tol)

  Apply Hessian approximation to vector.

- void invHessVec (Vector< Real > &hv, const Vector< Real > &v, const Vector< Real > &x, Real &tol)

  Apply inverse Hessian approximation to vector.

5.20.1 Detailed Description

template<class Real> class ROL::Objective_Powell< Real >

Powell's badly scaled function.

Definition at line 64 of file ROL_Powell.hpp.

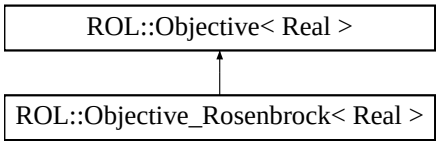
The documentation for this class was generated from the following file:

- ROL_Powell.hpp

5.21 ROL::Objective_Rosenbrock< Real > Class Template Reference

Rosenbrock's function.

`#include <ROL_Rosenbrock.hpp>`Inheritance diagram for ROL::Objective_Rosenbrock< Real >::



Public Member Functions

- **Objective_Rosenbrock** (Real alpha=100.0)
- Real value (const Vector< Real > &x, Real &tol)

  Compute value.

- void gradient (Vector< Real > &g, const Vector< Real > &x, Real &tol)

  Compute gradient.

- void hessVec (Vector< Real > &hv, const Vector< Real > &v, const Vector< Real > &x, Real &tol)

  Apply Hessian approximation to vector.

- void invHessVec (Vector< Real > &hv, const Vector< Real > &v, const Vector< Real > &x, Real &tol)

  Apply inverse Hessian approximation to vector.

Private Attributes

- Real **alpha_**

5.21.1 Detailed Description

template<class Real> class ROL::Objective_Rosenbrock< Real >

Rosenbrock's function.

Definition at line 64 of file ROL_Rosenbrock.hpp.

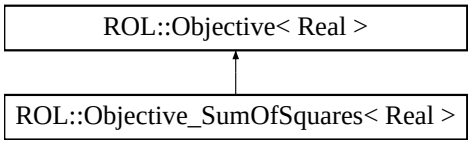
The documentation for this class was generated from the following file:

- ROL_Rosenbrock.hpp

5.22 ROL::Objective_SumOfSquares< Real > Class Template Reference

Sum of squares function.

`#include <ROL_SumOfSquares.hpp>`Inheritance diagram for ROL::Objective_SumOfSquares< Real >::



Public Member Functions

- Real value (const Vector< Real > &x, Real &tol)

  Compute value.

- void gradient (Vector< Real > &g, const Vector< Real > &x, Real &tol)

  Compute gradient.

- void hessVec (Vector< Real > &hv, const Vector< Real > &v, const Vector< Real > &x, Real &tol)

  Apply Hessian approximation to vector.

- void invHessVec (Vector< Real > &hv, const Vector< Real > &v, const Vector< Real > &x, Real &tol)

  Apply inverse Hessian approximation to vector.

5.22.1 Detailed Description

template<class Real> class ROL::Objective_SumOfSquares< Real >

Sum of squares function.

Definition at line 64 of file ROL_SumOfSquares.hpp.

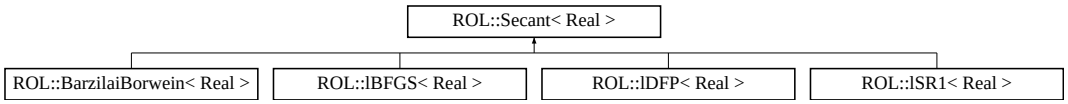
The documentation for this class was generated from the following file:

- ROL_SumOfSquares.hpp

5.23 ROL::Secant< Real > Class Template Reference

Provides interface for and implements limited-memory secant operators.

`#include <ROL_Secant.hpp>`Inheritance diagram for ROL::Secant< Real >::



Public Member Functions

- **Secant** (int M=10)
- Teuchos::RCP< SecantState< Real > > & **get_state** ()
- virtual void **update** (const Vector< Real > &grad, const Vector< Real > &gp, const Vector< Real > &s, const Real snorm, const int iter)
- virtual void **applyH** (Vector< Real > &Hv, const Vector< Real > &v, const Vector< Real > &x)=0
- virtual void **applyH0** (Vector< Real > &Hv, const Vector< Real > &v, const Vector< Real > &x)
- virtual void **applyB** (Vector< Real > &Bv, const Vector< Real > &v, const Vector< Real > &x)=0
- virtual void **applyB0** (Vector< Real > &Bv, const Vector< Real > &v, const Vector< Real > &x)
- void **test** (const Vector< Real > &x, const Vector< Real > &s)

Private Attributes

- Teuchos::RCP< SecantState< Real > > **state_**

5.23.1 Detailed Description

template<class Real> class ROL::Secant< Real >

Provides interface for and implements limited-memory secant operators.

Definition at line 67 of file ROL_Secant.hpp.

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

- ROL_Secant.hpp

5.24 ROL::SecantState< Real > Struct Template Reference

Public Attributes

- std::vector< Teuchos::RCP< Vector< Real > > > **iterDiff**
- std::vector< Teuchos::RCP< Vector< Real > > > **gradDiff**
- std::vector< Real > **product**
- std::vector< Real > **product2**
- int **storage**
- int **current**
- int **iter**

5.24.1 Detailed Description

template<class Real> struct ROL::SecantState< Real >

Definition at line 56 of file ROL_Secant.hpp.

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

- ROL_Secant.hpp

5.25 ROL::StatusTest< Real > Class Template Reference

Provides an interface to check status of optimization algorithms.

`#include <ROL_StatusTest.hpp>`

Public Member Functions

- **StatusTest** (Real gtol=1.e-6, Real stol=1.e-12, int max_iter=100)
- virtual bool check (AlgorithmState< Real > &state)

  Check algorithm status.

Private Attributes

- Real **gtol_**
- Real **stol_**
- int **max_iter_**

5.25.1 Detailed Description

template<class Real> class ROL::StatusTest< Real >

Provides an interface to check status of optimization algorithms.

Definition at line 56 of file ROL_StatusTest.hpp.

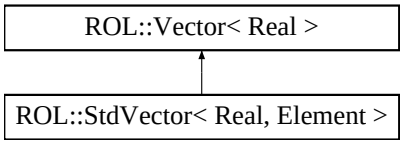
The documentation for this class was generated from the following file:

- ROL_StatusTest.hpp

5.26 ROL::StdVector< Real, Element > Class Template Reference

Provides the std::vector implementation of the ROL::Vector interface.

`#include <ROL_StdVector.hpp>`Inheritance diagram for ROL::StdVector< Real, Element >::



Public Member Functions

- **StdVector** (const Teuchos::RCP< std::vector< Element > > &std_vec)
- void **plus** (const Vector< Real > &x)
- void scale (const Real alpha)

  *Compute $y \leftarrow \alpha y$ where $y =$ *this.*

- Real **dot** (const Vector< Real > &x) const
- Real norm () const

  *Returns $\|y\|$ where $y =$ *this.*

- Teuchos::RCP< Vector< Real > > clone () const

  Clone to make a new (uninitialized) vector.

- Teuchos::RCP< const std::vector< Element > > **getVector** () const
- Teuchos::RCP< Vector< Real > > basis (const int i) const

  Return i-th basis vector: define if finite-difference gradients and Hessians are used.

- int dimension ()

  Return dimension of the vector space.

Private Attributes

- Teuchos::RCP< std::vector< Element > > **std_vec_**

5.26.1 Detailed Description

template<class Real, class Element = Real> class ROL::StdVector< Real, Element >

Provides the std::vector implementation of the ROL::Vector interface.

Definition at line 57 of file ROL_StdVector.hpp.

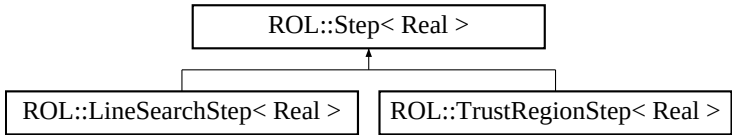
The documentation for this class was generated from the following file:

- ROL_StdVector.hpp

5.27 ROL::Step< Real > Class Template Reference

Provides the interface to compute optimization steps.

`#include <ROL_Step.hpp>`Inheritance diagram for ROL::Step< Real >::



Public Member Functions

- Teuchos::RCP< StepState< Real > > & **get_state** ()
- virtual void initialize (const Vector< Real > &x, Objective< Real > &obj, AlgorithmState< Real > &algo_state)

  Initialize step.

- virtual void compute (Vector< Real > &s, const Vector< Real > &x, Objective< Real > &obj, AlgorithmState< Real > &algo_state)=0

  Compute step.

- virtual void update (Vector< Real > &x, const Vector< Real > &s, Objective< Real > &obj, AlgorithmState< Real > &algo_state)=0

  Update step, if successful.

- virtual std::string printHeader (void) const =0

  Print iterate header.

- virtual std::string printName (void) const =0

  Print step name.

- virtual std::string print (AlgorithmState< Real > &algo_state, bool printHeader=false) const =0

  Print iterate status.

Public Attributes

- Teuchos::RCP< StepState< Real > > **state_**

5.27.1 Detailed Description

template<class Real> class ROL::Step< Real >

Provides the interface to compute optimization steps.

Definition at line 76 of file ROL_Step.hpp.

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

- ROL_Step.hpp

5.28 ROL::StepState< Real > Struct Template Reference

Public Attributes

- Teuchos::RCP< Vector< Real > > **gradientVec**
- Teuchos::RCP< Vector< Real > > **descentVec**

5.28.1 Detailed Description

template<class Real> struct ROL::StepState< Real >

Definition at line 69 of file ROL_Step.hpp.

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

- ROL_Step.hpp

5.29 ROL::TrustRegion< Real > Class Template Reference

Provides interface for and implements trust-region subproblem solvers.

`#include <ROL_TrustRegion.hpp>`

Public Member Functions

- **TrustRegion** (Teuchos::ParameterList &parlist)
- void **update** (Vector< Real > &x, Real &fnew, Real &del, int &nfval, int &ngrad, int &flagTR, const Vector< Real > &s, const Real snorm, const Real fold, const Vector< Real > &g, Objective< Real > &obj, Teuchos::RCP< Secant< Real > > &secant=Teuchos::null)
- void **run** (Vector< Real > &s, Real &snorm, Real &del, int &iflag, int &iter, const Vector< Real > &x, const Vector< Real > &grad, const Real &gnorm, Objective< Real > &obj, Teuchos::RCP< Secant< Real > > &secant=Teuchos::null)
- void **cauchypoint** (Vector< Real > &s, Real &snorm, Real &del, int &iflag, int &iter, const Vector< Real > &x, const Vector< Real > &grad, const Real &gnorm, Objective< Real > &obj, Teuchos::RCP< Secant< Real > > &secant=Teuchos::null)
- void **truncatedCG** (Vector< Real > &s, Real &snorm, Real &del, int &iflag, int &iter, const Vector< Real > &x, const Vector< Real > &grad, const Real &gnorm, Objective< Real > &obj, Teuchos::RCP< Secant< Real > > &secant=Teuchos::null)
- void **dogleg** (Vector< Real > &s, Real &snorm, Real &del, int &iflag, int &iter, const Vector< Real > &x, const Vector< Real > &grad, const Real &gnorm, Objective< Real > &obj, Teuchos::RCP< Secant< Real > > &secant=Teuchos::null)
- void **doubledogleg** (Vector< Real > &s, Real &snorm, Real &del, int &iflag, int &iter, const Vector< Real > &x, const Vector< Real > &grad, const Real &gnorm, Objective< Real > &obj, Teuchos::RCP< Secant< Real > > &secant=Teuchos::null)

Private Attributes

- ETrustRegion **etr_**
- bool **useSecantPrecond_**
- bool **useSecantHessVec_**
- int **maxit_**
- Real **tol1_**
- Real **tol2_**

- Real **delmin_**
- Real **delmax_**
- Real **eta0_**
- Real **eta1_**
- Real **eta2_**
- Real **gamma0_**
- Real **gamma1_**
- Real **gamma2_**
- Real **pRed_**
- Real **TRsafe_**
- Real **eps_**

5.29.1 Detailed Description

template<class Real> class ROL::TrustRegion< Real >

Provides interface for and implements trust-region subproblem solvers.

Definition at line 56 of file ROL_TrustRegion.hpp.

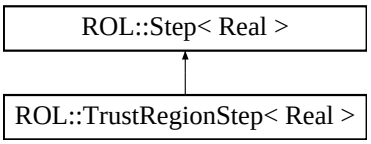
The documentation for this class was generated from the following file:

- ROL_TrustRegion.hpp

5.30 ROL::TrustRegionStep< Real > Class Template Reference

Provides the interface to compute optimization steps with trust regions.

`#include <ROL_TrustRegionStep.hpp>`Inheritance diagram for ROL::TrustRegionStep< Real >::



Public Member Functions

- **TrustRegionStep** (Teuchos::ParameterList &parlist)
- **TrustRegionStep** (Teuchos::RCP< Secant< Real > > &secant, Teuchos::ParameterList &parlist)
- void initialize (const Vector< Real > &x, Objective< Real > &obj, AlgorithmState< Real > &algo_state)

  Initialize step.

- void compute (Vector< Real > &s, const Vector< Real > &x, Objective< Real > &obj, AlgorithmState< Real > &algo_state)

  Compute step.

- void update (Vector< Real > &x, const Vector< Real > &s, Objective< Real > &obj, AlgorithmState< Real > &algo_state)

  Update step, if successful.

- std::string printHeader (void) const

  Print iterate header.

- std::string printName (void) const

  Print step name.

- std::string print (AlgorithmState< Real > &algo_state, bool printHeader=false) const

  Print iterate status.

Private Attributes

- Teuchos::RCP< Secant< Real > > **secant_**
- Teuchos::RCP< TrustRegion< Real > > **trustRegion_**
- ETrustRegion **etr_**
- ESecant **esec_**
- bool **useSecantHessVec_**
- bool **useSecantPrecond_**
- Real **del_**
- std::vector< bool > **useInexact_**
- int **TRflag_**
- int **TR_nfval_**
- int **TR_ngrad_**
- int **CGflag_**
- int **CGiter_**

5.30.1 Detailed Description

template<class Real> class ROL::TrustRegionStep< Real >

Provides the interface to compute optimization steps with trust regions.

Definition at line 63 of file ROL_TrustRegionStep.hpp.

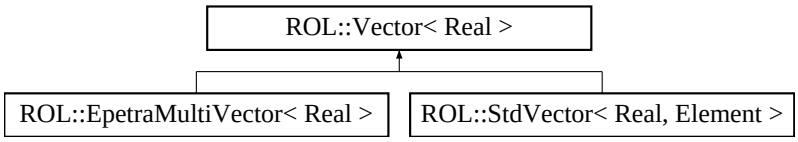
The documentation for this class was generated from the following file:

- ROL_TrustRegionStep.hpp

5.31 ROL::Vector< Real > Class Template Reference

Provides the vector space interface.

`#include <ROL_Vector.hpp>`Inheritance diagram for ROL::Vector< Real >::



Public Member Functions

- virtual void plus (const Vector &x)=0
  *Compute $y \leftarrow x + y$ where $y =$ *this.*
- virtual void scale (const Real alpha)=0
  *Compute $y \leftarrow \alpha y$ where $y =$ *this.*
- virtual Real dot (const Vector &x) const =0
  *Returns $\langle y, x \rangle$ where $y =$ *this.*
- virtual Real norm () const =0
  *Returns $\|y\|$ where $y =$ *this.*
- virtual Teuchos::RCP< Vector > clone () const =0
  Clone to make a new (uninitialized) vector.
- virtual void axpy (const Real alpha, const Vector &x)
  *Compute $y \leftarrow \alpha x + y$ where $y =$ *this.*
- virtual void zero ()
  Set to zero vector.
- virtual Teuchos::RCP< Vector > basis (const int i) const
  Return i-th basis vector: define if finite-difference gradients and Hessians are used.
- virtual int dimension ()
  Return dimension of the vector space.
- virtual void set (const Vector &x)
  *Set $y \leftarrow x$ where $y =$ *this.*

5.31.1 Detailed Description

template<class Real> class ROL::Vector< Real >

Provides the vector space interface. The basic interface to be supplied by the user includes:

- vector addition,
- scalar multiplication,
- dot (scalar) product of vectors,
- vector norm,
- cloning of vectors.

The dot product can represent an inner product (in Hilbert space) or a duality pairing (in general Banach space).

There are additional virtual member functions that the user may want to reimplement for added efficiency.

Definition at line 70 of file ROL_Vector.hpp.

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

- ROL_Vector.hpp

Chapter 6

File Documentation

6.1 example_01.cpp File Reference

Shows how to minimize Rosenbrock's function using Newton-Krylov. `#include "ROL_Rosenbrock.hpp"`

```
#include "ROL_LineSearchStep.hpp"
#include "ROL_Algorithm.hpp"
#include "Teuchos_oblackholestream.hpp"
#include "Teuchos_GlobalMPISession.hpp"
#include <iostream>
```

Defines

- #define **USE_HESSVEC** 1

Typedefs

- typedef double **RealT**

Functions

- int **main** (int argc, char *argv[ ])

6.1.1 Detailed Description

Shows how to minimize Rosenbrock's function using Newton-Krylov.

Definition in file example_01.cpp.

6.2 ROL_Beale.hpp File Reference

Contains definitions for Beale's function. `#include "ROL_StdVector.hpp"`

`#include "ROL_Objective.hpp"`

Classes

- class ROL::Objective_Beale< Real >

  Beale's function.

Defines

- #define **USE_HESSVEC** 1

Functions

- template<class Real >
  void **ROL::getBeale** (Teuchos::RCP< Objective< Real > > &obj, Vector< Real > &x0, Vector< Real > &x)

6.2.1 Detailed Description

Contains definitions for Beale's function.

Author:

Created by D. Ridzal and D. Kouri.

Definition in file ROL_Beale.hpp.

6.3 ROL_FreudensteinRoth.hpp File Reference

Contains definitions for Freudenstein and Roth's function. `#include "ROL_StdVector.hpp"`

`#include "ROL_Objective.hpp"`

Classes

- class ROL::Objective_FreudensteinRoth< Real >

  Freudenstein and Roth's function.

Functions

- template<class Real >
  void **ROL::getFreudensteinRoth** (Teuchos::RCP< Objective< Real > > &obj, Vector< Real > &x0, Vector< Real > &x)

6.3.1 Detailed Description

Contains definitions for Freudenstein and Roth's function.

Author:

Created by D. Ridzal and D. Kouri.

Definition in file ROL_FreudensteinRoth.hpp.

6.4 ROL_LeastSquares.hpp File Reference

Contains definitions for least squares function. `#include "ROL_StdVector.hpp"`

`#include "ROL_Objective.hpp"`

Classes

- class ROL::Objective_LeastSquares< Real >

  Least squares function.

Functions

- template<class Real >
  void **ROL::getLeastSquares** (Teuchos::RCP< Objective< Real > > &obj, Vector< Real > &x0, Vector< Real > &x)

6.4.1 Detailed Description

Contains definitions for least squares function.

Author:

Created by D. Ridzal and D. Kouri.

Definition in file ROL_LeastSquares.hpp.

6.5 ROL_PoissonControl.hpp File Reference

Contains definitions for Poisson optimal control. `#include "ROL_StdVector.hpp"`

`#include "ROL_Objective.hpp"`

Classes

- class ROL::Objective_PoissonControl< Real >

  Poisson distributed control.

Functions

- template<class Real >
  void **ROL::getPoissonControl** (Teuchos::RCP< Objective< Real > > &obj, Vector< Real > &x0, Vector< Real > &x)

6.5.1 Detailed Description

Contains definitions for Poisson optimal control.

Author:

Created by D. Ridzal and D. Kouri.

Definition in file ROL_PoissonControl.hpp.

6.6 ROL_PoissonInversion.hpp File Reference

Contains definitions for Poisson material inversion. `#include "ROL_StdVector.hpp"`

`#include "ROL_Objective.hpp"`

`#include "Teuchos_LAPACK.hpp"`

Classes

- class ROL::Objective_PoissonInversion< Real >

  Poisson material inversion.

Functions

- template<class Real >
  void **ROL::getPoissonInversion** (Teuchos::RCP< Objective< Real > > &obj, Vector< Real > &x0, Vector< Real > &x)

6.6.1 Detailed Description

Contains definitions for Poisson material inversion.

Author:

Created by D. Ridzal and D. Kouri.

Definition in file ROL_PoissonInversion.hpp.

6.7 ROL_Powell.hpp File Reference

Contains definitions for Powell's badly scaled function. `#include "ROL_StdVector.hpp"`

`#include "ROL_Objective.hpp"`

Classes

- class ROL::Objective_Powell< Real >

  Powell's badly scaled function.

Functions

- template<class Real >
  void **ROL::getPowell** (Teuchos::RCP< Objective< Real > > &obj, Vector< Real > &x0, Vector< Real > &x)

6.7.1 Detailed Description

Contains definitions for Powell's badly scaled function.

Author:

Created by D. Ridzal and D. Kouri.

Definition in file ROL_Powell.hpp.

6.8 ROL_Rosenbrock.hpp File Reference

Contains definitions for Rosenbrock's function. `#include "ROL_StdVector.hpp"`

`#include "ROL_Objective.hpp"`

Classes

- class ROL::Objective_Rosenbrock< Real >

  Rosenbrock's function.

Functions

- template<class Real >
  void **ROL::getRosenbrock** (Teuchos::RCP< Objective< Real > > &obj, Vector< Real > &x0, Vector< Real > &x)

6.8.1 Detailed Description

Contains definitions for Rosenbrock's function.

Author:

Created by D. Ridzal and D. Kouri.

Definition in file ROL_Rosenbrock.hpp.

6.9 ROL_SumOfSquares.hpp File Reference

Contains definitions for sum of squares function. `#include "ROL_StdVector.hpp"`

`#include "ROL_Objective.hpp"`

Classes

- class ROL::Objective_SumOfSquares< Real >

  Sum of squares function.

Functions

- template<class Real >
  void **ROL::getSumOfSquares** (Teuchos::RCP< Objective< Real > > &obj, Vector< Real > &x0, Vector< Real > &x)

6.9.1 Detailed Description

Contains definitions for sum of squares function.

Author:

Created by D. Ridzal and D. Kouri.

Definition in file ROL_SumOfSquares.hpp.

6.10 ROL_TestObjectives.hpp File Reference

Contains definitions of test objective functions. `#include "ROL_Rosenbrock.hpp"`

```
#include "ROL_FreudensteinRoth.hpp"
#include "ROL_Beale.hpp"
#include "ROL_Powell.hpp"
#include "ROL_SumOfSquares.hpp"
#include "ROL_LeastSquares.hpp"
#include "ROL_PoissonControl.hpp"
#include "ROL_PoissonInversion.hpp"
#include "ROL_Types.hpp"
#include "ROL_StdVector.hpp"
#include "ROL_Objective.hpp"
```

Functions

- template<class Real >
  void **ROL::getTestObjectives** (Teuchos::RCP< Objective< Real > > &obj, Vector< Real > &x0, Vector< Real > &x, const ETestObjectives test)

6.10.1 Detailed Description

Contains definitions of test objective functions.

Author:

Created by D. Ridzal and D. Kouri.

Definition in file ROL_TestObjectives.hpp.

6.11 ROL_Types.hpp File Reference

Contains definitions of custom data types in ROL. `#include <Teuchos_ScalarTraits.hpp>`

`#include <Teuchos_TestForException.hpp>`

Defines

- #define **ROL_VALIDATE**(A)
- #define ROL_NUM_CHECKDERIV_STEPS 13

  Number of steps for derivative checks.

Enumerations

- enum **EDescent** {

  DESCENT_STEEPEST = 0, **DESCENT_NONLINEARCG**, **DESCENT_SECANT**, **DESCENT_NEWTON**,

  DESCENT_NEWTONKRYLOV, **DESCENT_SECANTPRECOND**, **DESCENT_LAST** }

  Enumeration of descent direction types.

- enum **ESecant** {

  SECANT_LBFGS = 0, **SECANT_LDFP**, **SECANT_LSR1**, **SECANT_BARZILAIBORWEIN**,

  SECANT_USERDEFINED, **SECANT_LAST** }

  Enumeration of secant update algorithms.

- enum **ENonlinearCG** {

  NONLINEARCG_HESTENES_STIEFEL = 0, **NONLINEARCG_FLETCHER_REEVES**, **NONLINEARCG_DANIEL**, **NONLINEARCG_POLAK_RIBIERE**,

  NONLINEARCG_FLETCHER_CONJDESC, **NONLINEARCG_LIU_STOREY**, **NONLINEARCG_DAI_YUAN**, **NONLINEARCG_HAGAR_ZHANG**,

  NONLINEARCG_LAST }

  Enumeration of nonlinear CG algorithms.

- enum **ELineSearch** {

  LINESEARCH_BACKTRACKING = 0, **LINESEARCH_BISECTION**, **LINESEARCH_GOLDENSECTION**, **LINESEARCH_CUBICINTERP**,

  LINESEARCH_BRENTS, **LINESEARCH_LAST** }

  Enumeration of line-search types.

- enum **ECurvatureCondition** { **CURVATURECONDITION_WOLFE** = 0, **CURVATURECONDITION_STRONGWOLFE**, **CURVATURECONDITION_GOLDSTEIN**, **CURVATURECONDITION_LAST** }

  Enumeration of line-search curvature conditions.

- enum **ETrustRegion** {

  TRUSTREGION_CAUCHYPOINT = 0, **TRUSTREGION_TRUNCATEDCG**, **TRUSTREGION_DOGLEG**, **TRUSTREGION_DOUBLEDOGLEG**,

  TRUSTREGION_LAST }

  Enumeration of trust-region solver types.

- enum **ETestObjectives** {

  TESTOBJECTIVES_ROSENBROCK = 0, **TESTOBJECTIVES_FREUDENSTEINANDROTH**, **TESTOBJECTIVES_BEALE**, **TESTOBJECTIVES_POWELL**,

  TESTOBJECTIVES_SUMOFSQUARES, **TESTOBJECTIVES_LEASTSQUARES**, **TESTOBJECTIVES_POISSONCONTROL**, **TESTOBJECTIVES_POISSONINVERSION**,

  TESTOBJECTIVES_LAST }

  Enumeration of test objective functions.

Functions

- std::string **ROL::EDescentToString** (EDescent tr)
- int ROL::isValidDescent (EDescent d)

  Verifies validity of a Secant enum.

- EDescent & **ROL::operator++** (EDescent &type)
- EDescent **ROL::operator++** (EDescent &type, int)
- EDescent & **ROL::operator--** (EDescent &type)
- EDescent **ROL::operator--** (EDescent &type, int)
- std::string **ROL::ESecantToString** (ESecant tr)

- int ROL::isValidSecant (ESecant s)

  Verifies validity of a Secant enum.

- ESecant & **ROL::operator++** (ESecant &type)
- ESecant **ROL::operator++** (ESecant &type, int)
- ESecant & **ROL::operator--** (ESecant &type)
- ESecant **ROL::operator--** (ESecant &type, int)
- std::string **ROL::ENonlinearCGToString** (ENonlinearCG tr)
- int ROL::isValidNonlinearCG (ENonlinearCG s)

  Verifies validity of a NonlinearCG enum.

- ENonlinearCG & **ROL::operator++** (ENonlinearCG &type)
- ENonlinearCG **ROL::operator++** (ENonlinearCG &type, int)
- ENonlinearCG & **ROL::operator--** (ENonlinearCG &type)
- ENonlinearCG **ROL::operator--** (ENonlinearCG &type, int)
- std::string **ROL::ELineSearchToString** (ELineSearch ls)
- int ROL::isValidLineSearch (ELineSearch ls)

  Verifies validity of a LineSearch enum.

- ELineSearch & **ROL::operator++** (ELineSearch &type)
- ELineSearch **ROL::operator++** (ELineSearch &type, int)
- ELineSearch & **ROL::operator--** (ELineSearch &type)
- ELineSearch **ROL::operator--** (ELineSearch &type, int)
- std::string **ROL::ECurvatureConditionToString** (ECurvatureCondition ls)
- int ROL::isValidCurvatureCondition (ECurvatureCondition ls)

  Verifies validity of a CurvatureCondition enum.

- ECurvatureCondition & **ROL::operator++** (ECurvatureCondition &type)
- ECurvatureCondition **ROL::operator++** (ECurvatureCondition &type, int)
- ECurvatureCondition & **ROL::operator--** (ECurvatureCondition &type)
- ECurvatureCondition **ROL::operator--** (ECurvatureCondition &type, int)
- std::string **ROL::ETrustRegionToString** (ETrustRegion tr)
- int ROL::isValidTrustRegion (ETrustRegion ls)

  Verifies validity of a TrustRegion enum.

- ETrustRegion & **ROL::operator++** (ETrustRegion &type)
- ETrustRegion **ROL::operator++** (ETrustRegion &type, int)
- ETrustRegion & **ROL::operator--** (ETrustRegion &type)
- ETrustRegion **ROL::operator--** (ETrustRegion &type, int)
- std::string **ROL::ETestObjectivesToString** (ETestObjectives to)
- int ROL::isValidTestObjectives (ETestObjectives to)

  Verifies validity of a TestObjectives enum.

- ETestObjectives & **ROL::operator++** (ETestObjectives &type)
- ETestObjectives **ROL::operator++** (ETestObjectives &type, int)
- ETestObjectives & **ROL::operator--** (ETestObjectives &type)
- ETestObjectives **ROL::operator--** (ETestObjectives &type, int)

Variables

- static const double ROL::ROL_EPSILON = std::abs(Teuchos::ScalarTraits<double>::eps())

  Platform-dependent machine epsilon.

- static const double ROL::ROL_THRESHOLD = 10.0 * ROL_EPSILON

  Tolerance for various equality tests.

6.11.1 Detailed Description

Contains definitions of custom data types in ROL.

Author:

Created by D. Ridzal and D. Kouri.

Definition in file ROL_Types.hpp.

Index