

NAME

LevenbergMarquardt – Levenberg-Marquardt optimization method.

SYNOPSIS

```
#include <ql/math/optimization/levenbergmarquardt.hpp>
```

Inherits **OptimizationMethod**.

Public Member Functions

LevenbergMarquardt (**Real** epsfcn=1.0e-8, **Real** xtol=1.0e-8, **Real** gtol=1.0e-8, **bool** useCostFunctionsJacobian=false)

virtual **EndCriteria::Type** **minimize** (**Problem** &P, const **EndCriteria** &endCriteria)
minimize the optimization problem P

virtual **Integer** **getInfo** () const

void **fcn** (int m, int n, **Real** *x, **Real** *fvec, int *iflag)

void **jacFcn** (int m, int n, **Real** *x, **Real** *fjac, int *iflag)

Detailed Description

Levenberg-Marquardt optimization method.

This implementation is based on MINPACK (<http://www.netlib.org/minpack>, <http://www.netlib.org/cephes/linalg.tgz>) It has a built in fd scheme to compute the jacobian, which is used by default. If useCostFunctionsJacobian is true the corresponding method in the cost function of the problem is used instead. Note that the default implementation of the jacobian in **CostFunction** uses a central difference (order 2, but requiring more function evaluations) compared to the forward difference implemented here (order 1).

Examples:

BermudanSwaption.cpp.

Author

Generated automatically by Doxygen for QuantLib from the source code.

