RAISE PARTNER

Wall Risk Engine

# Wall Risk Engine® User Guide

August 21, 2015

# Contents

# List of Figures

# 1 Introducing Wall Risk Engine®

## 1.1 What is Wall Risk Engine®?

### 1.1.1 Overview

Wall Risk Engine® is a software component composed of 5 modules:

- Risk and expert modeling: estimating the risk (volatilities, correlations,...) and the performance (expected returns) of the components of your investment universe;
- Risk and performance analysis: analyzing (ex-ante and ex-post) the risk and performance properties of your financial instruments and portfolios;
- Robust portfolio optimization: designing tailored, dynamic and robust allocation strategies;
- Simulation and stress-tests: Simulating future values of financial instruments (simple and structured products) and portfolios, computing large-scale risk measures (VaR) based on Monte-Carlo approaches.
- Data preparation: outlier detection and data completion functionalities to prepare the data for the risk modeling and risk analysis.

### 1.1.2 Main features

**Risk and performance modeling**

This module provides advanced functionalities to build tailored risk and performance models, using both historical data and exogenous information such as anticipations, market-implied information or in-house scores. Monitoring and estimating the correlations (or covariances) across all asset classes is a very sensitive issue especially within a portfolio optimization or a pricing process. The risk models provided in Wall Risk Engine® rely on 3 main pillars:

- Computing reactive and dynamic historical estimations of the correlations and volatilities (or the variance/covariance matrix);
- Taking into account exogenous information (if any) about the correlations and the volatilities;

- Applying a relevant filter to ensure the robustness of the risk model (by removing the noise) and merge historical and exogenous information.

**Risk and performance analysis**

This module embeds a wide range of ex-ante and ex-post analysis designed to monitor the risk and performance of a given portfolio:

- Performance measures;
- Risk measures: Volatility, semi-volatility, tracking-error, robust VaR, drawdown,...;
- Ratios: Sharpe, Sortino, Information, Omega;
- 3rd and 4th moments of the distribution: skewness, kurtosis;
- Risk and performance contributions.

**Robust portfolio optimization**

This module provides optimization functions to design tailored portfolios:

- Minimization of the risk of the portfolio given some performance constraint;
- Risk Budgeting: maximization of the expected return of the portfolio given some volatility budget constraints;
- Market portfolio (with maximum Sharpe ratio);
- Off-the-shelf adaptive allocation model to build reactive and robust portfolios;

All these functionalities embed patented robust optimization techniques and provide reliable and pertinent portfolios, ensuring a smooth behaviour with respect to market changes. These allocation models are compatible with any type of linear constraints on the weights such as sectorial constraints, asset-level constraints.

**Simulation and stress-testing**

This module provides tools to:

- simulate financial instruments using factorial or dense simulation models,
- build scenario simulations based on exogenous factors,
- stress-test a financial instrument or a portfolio,
- compute risk measures (Value at Risk) on large-scale portfolios based on Monte-Carlo simulations.

**Quant data preparation**

This module provides functionalities to prepare the market data for the risk modeling and analysis:

- detecting outliers in multi-dimensional time series,
- completing historical time series with robust regression approaches.

## 1.2 Installation

### 1.2.1 Technical requirements

**System requirements**

All the functions are available for:

- Win32/Win64 architecture on Intel processors : Dynamic link library (DLL)
- Linux platforms - Shared library (SO)

The components are designed to be highly efficient on all types of scalar machines (PC's, workstations). The components can be used and integrated in different types of applications such as:

- Win32/Win64 applications (.NET, .COM, Visual C++ 6.0, Visual Basic 6.0),
- spreadsheets in Microsoft Office applications such as Excel and Access (VBA),
- web-based applications (ASP, PHP, VB script, Java),
- client applications (Java, VB),
- high-performance computing, computation servers (Fortran 77/90, C/C++),
- quant research tools: Matlab, Scilab.

### 1.2.2 Installation procedure

The installation procedure depends on the chosen API. Please refer to the *Getting started with Wall Risk Engine®* guide delivered with the API.

### 1.2.3 License

Each installation of the Wall Risk Engine® library requires a license which is generated from the expiration date and the MAC (or physical) address of the user's computer. To find this MAC address, you need to open a DOS command window, type "ipconfig /all" and read the MAC address corresponding to your network connexion.

## 1.3 Documentation

The available documentation for Wall Risk Engine® includes:

- The Wall Risk Engine® User Guide describes practical information for developers who use Wall Risk Engine®.
- The Wall Risk Engine® Quant Guide provides detailed information about the quant models.
- For each API, we provide a *Getting started* guide (pdf) driving you through your first experience with Wall Risk Engine®. You will find the detailed installation procedure as well as simple examples to get familiar with the functionalities and the API.

- A technical manual and integration examples for each language (Java, C/C++, Visual C++, VB/VBA) providing functions description, instructions and annotated sample programs.

## 1.4 Support

Any bug report or question regarding the use of Wall Risk Engine® should be sent to the consultant in charge of your project at Raise Partner.

# 2 How To's

We present hereafter examples explaining how to organize Wall Risk Engine® functionalities in order to build allocation, backtesting, stress-testing and simulation processes.

## 2.1 Portfolio allocation process

The figure 2.1 describes how to articulate Wall Risk Engine® functionalities to implement a portfolio allocation process, from the computation of historical returns to the ex-ante analysis of the optimal portfolio:

- Step 1: Computation of the arithmetic historical returns from the historical prices with the *NORMmodelingReturns* function
- Step 2: Construction of the risk model:
  - Computation of the historical estimation of the covariance matrix given the chosen estimation window, even if the historical dataset is incomplete (cf. Risk and Performance Modeling module).
  - Calibration of the covariance matrix with additionnal views (explicit or interval) on the volatilities or correlations.
- Step 3: Construction of the performance model:
  - Computation of the historical mean returns (historical trend approach) given the chosen estimation window.
  - This step can be completed or replaced by any in-house performance modeling process (based on a scoring approach for instance).
- Step 4: Computation of the optimal portfolio based on the risk and performance models, the chosen strategy (minimum volatility, risk budgeting...), the set of strategy parameters and the user-defined constraints.
- Step 5: Ex-ante analysis of the optimal portfolio with the ex-ante functions of the Risk and Performance module.

## 2.2 Back-testing of an investment strategy

The figure 2.2 describes how to implement a back-testing process to analyze an investment strategy on real past scenarios.

At each historical rebalancing date, the returns corresponding to the rolling estimation window are extracted and fed into the risk and performance models. Then the optimal portfolio at this date is computed, and the portfolio is left untouched until the next rebalancing date.

At the end of the back-test, the performances of the simulated portfolio are recomposed and compared to the underlying instruments (or to any other benchmark) via an ex-post risk and performance analysis.

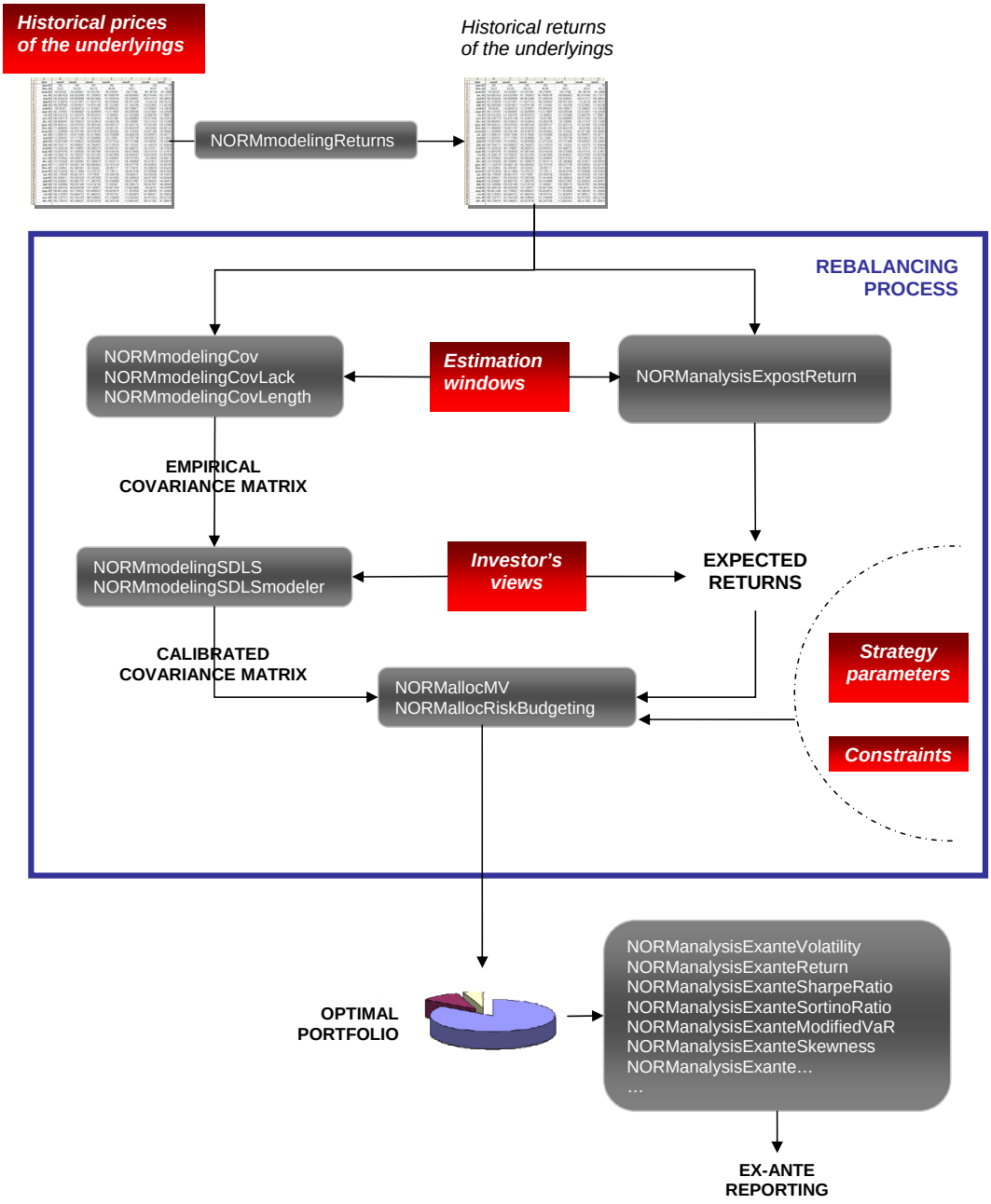


Figure 2.1: Portfolio rebalancing process with Wall Risk Engine® functions

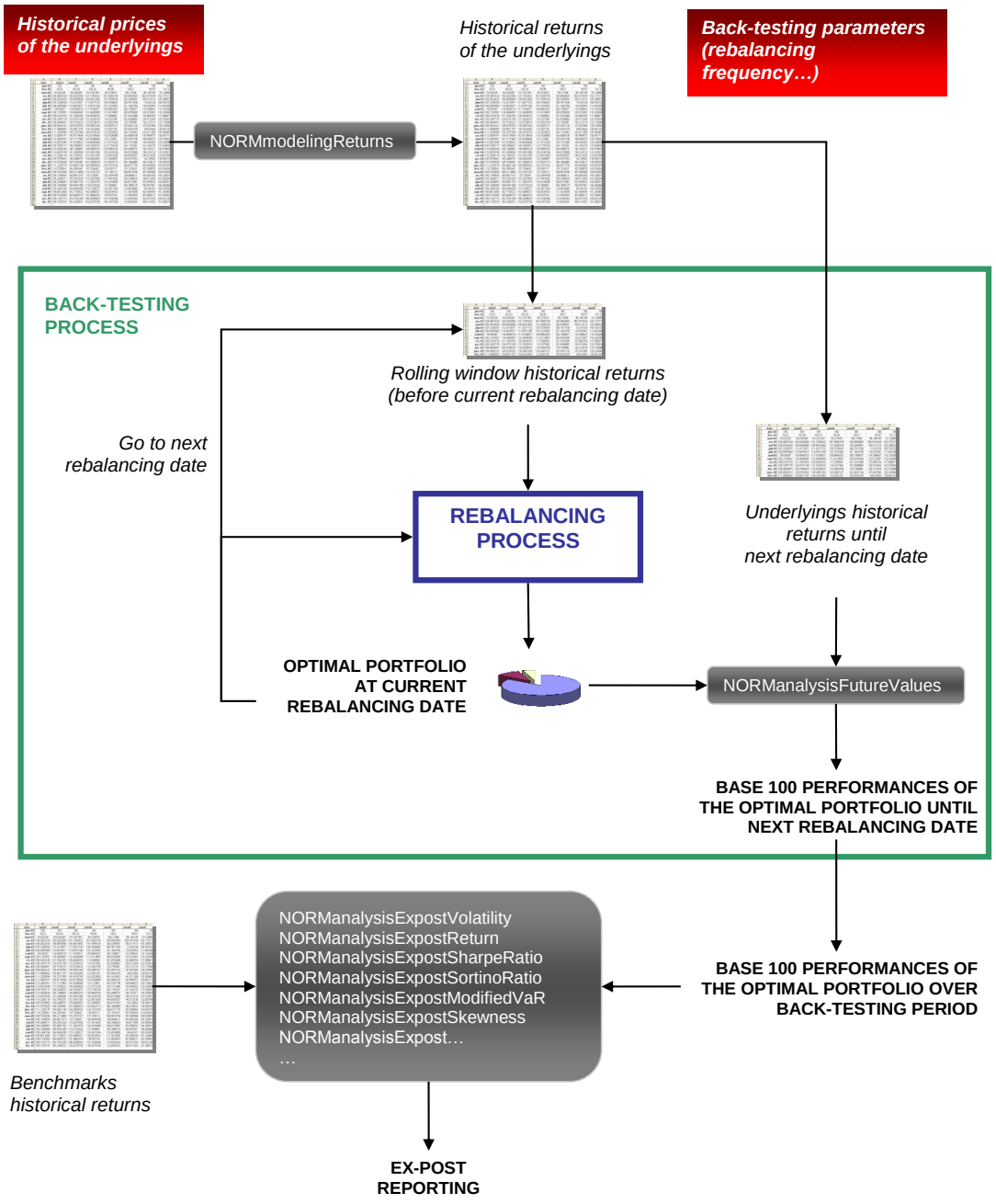


Figure 2.2: Back-testing process with Wall Risk Engine® functions

## 2.3 Simulating an analyze a complex financial instrument (such as a structured product)

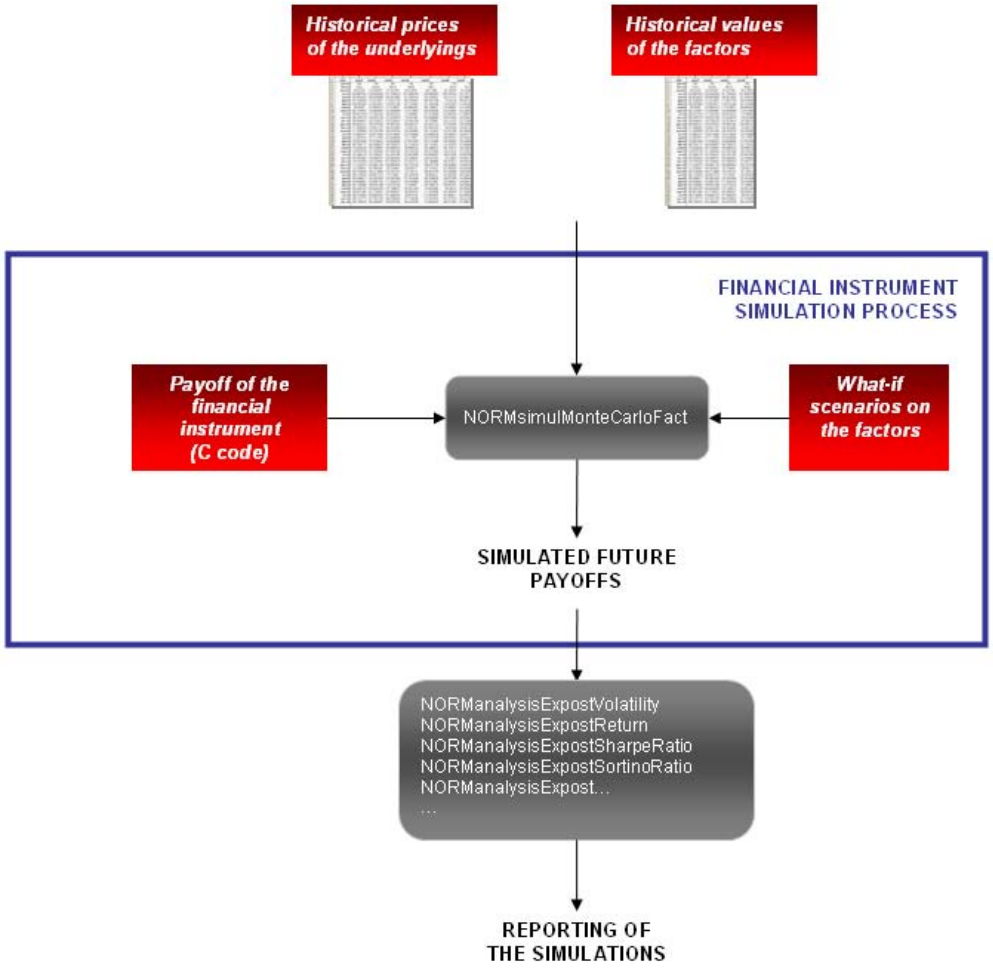


Figure 2.3: Financial instrument simulation process with Wall Risk Engine® functions

## 2.4 Building stress-testing scenarios for a portfolio

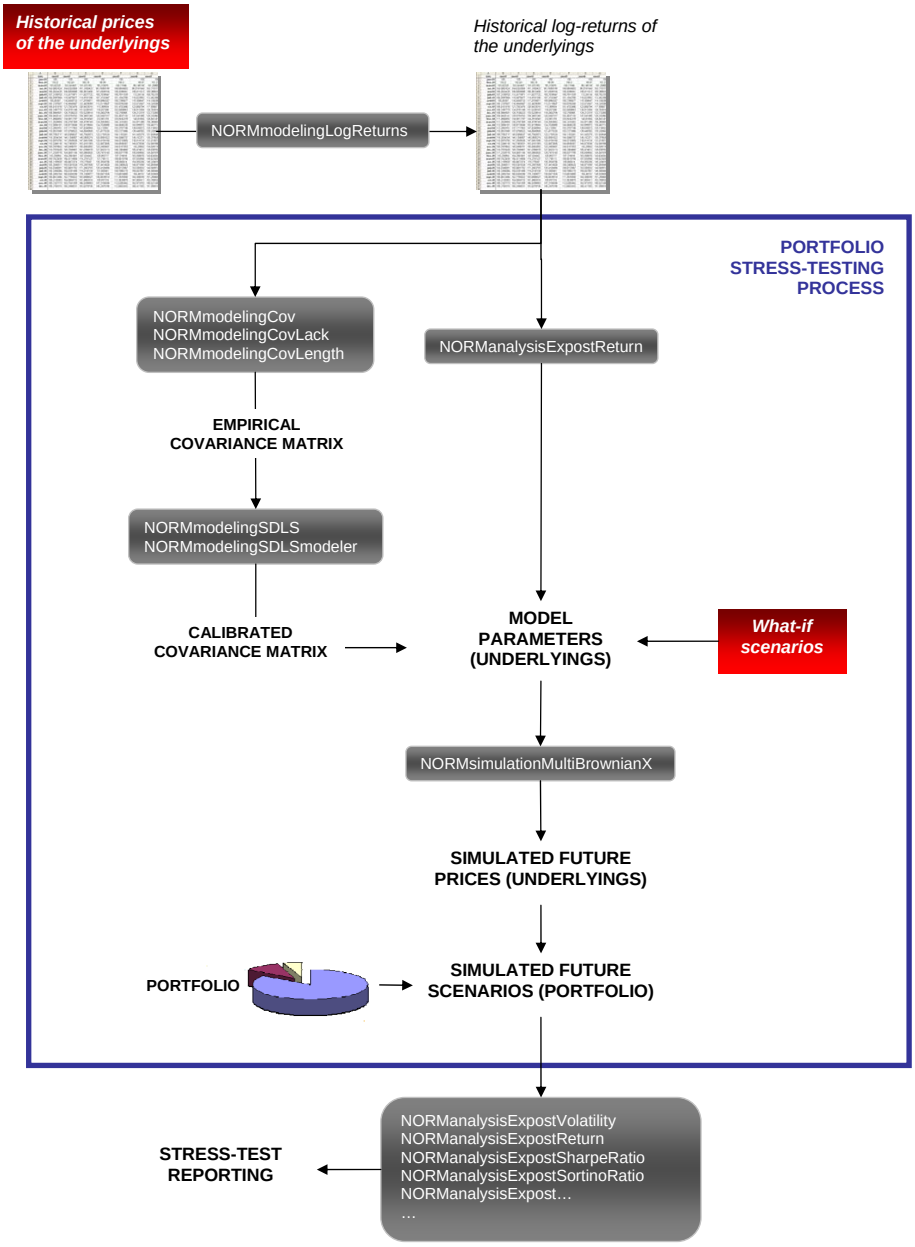


Figure 2.4: Portfolio stress-testing process with Wall Risk Engine® functions

# Part I

# Functions description

## 2.5 Allocation module

### 2.5.1 WREallocConstSector - Sectors Constraints Modelling

#### Overview

This component is a tool for sector constraints modelling.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of asset(s) (greater than or equal to 1) |
| nbSectors | integer | number of sector(s) (greater than or equal to 1) |
| nbSubSectors | integer array of dimension nbSectors | number of sub-sectors per sector |
| assetsSectors | integer array of dimension nbAssets by nbSectors | sub-sector number for each asset in each sector |
| minSectWeights | double array of dimension nbSectors by nssmax | sub-sectors min weights constraints per sector, nssmax is the maximum number of sub-sectors per sector |
| eqSectWeights | double array of dimension nbSectors by nssmax | sub-sectors equality weights constraints per sector, nssmax is the maximum number of sub-sectors per sector |
| maxSectWeights | double array of dimension nbSectors by nssmax | sub-sectors max weights constraints per sector, nssmax is the maximum number of sub-sectors per sector |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| nbEqConst | integer array of dimension 1 | number of equality constraints |
| nbIneqConst | integer array of dimension 1 | number of inequality constraints (=2sum(nbSubSectors[i]), i=1,nbSectors) |
| C | double array of dimension nbAssets by nbeq+2*sum | matrix of constraints |
| b | double array of dimension nbeq+2*sum | vector of constraints |
| info | integer array of dimension 1 | diagnostic argument |

### 2.5.2 WREallocERC - Risk Parity Allocation

#### Overview

test.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of asset(s) (greater than 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| riskweights | double array of dimension nbAssets | mean return(s) |
| riskMeasureType | integer | Type of measure. Equal to 1. |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| optimalWeights | double array of dimension nbAssets | optimal portfolio |
| info | integer array of dimension 1 | diagnostic argument |

### 2.5.3 WREallocGKCVaR - Minimize Kernel CVaR

#### Overview

It minimizes Kernel CVaR.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of asset(s) (greater than 1) |
| kappa2 | double | adjusting coefficient for covariance matrix |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| kappa1 | double | adjusting coefficient for 1st moment |
| expret | double array of dimension nbAssets | expected returns |
| nbDates | integer | number of dates |
| kappa3 | double | aversion coefficient VaR |
| historet | double array of dimension nbDates by nbAssets | Historical returns |
| alpha | double | Confidence level of VaR |
| nbEqConst | integer | number of equality constraints (greater than or equal to 0) |
| nbIneqConst | integer | number of inequality constraints (greater than or equal to 0) |
| C | double array of dimension nbAssets by nbEqConst+nbIneqConst | matrix of constraints. If no constraints are given, C=0 |
| b | double array of dimension nbEqConst+nbIneqConst | vector of constraints. If no constraints are given, b=0 |
| minWeights | double array of dimension nbAssets | lower bounds |
| maxWeights | double array of dimension nbAssets | upper bounds |
| istargetPerf | integer | 1 : there is a target performance. 0 : no. |
| targetPerf | double | target performance |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| optimalWeights | double array of dimension nbAssets | optimal portfolio |
| info | integer array of dimension 1 | diagnostic argument |

### 2.5.4 WREallocGKCVaRTC - Minimize CVaR with TC constraints

#### Overview

It minimizes Kernel CVaR with piece-wise polynomial transaction costs constraints.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of asset(s) (greater than 1) |
| kappa2 | double | adjusting coefficient for covariance matrix |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| kappa1 | double | adjusting coefficient for 1st moment |
| expret | double array of dimension nbAssets | expected returns |
| nbDates | integer | number of dates |
| kappa3 | double | aversion coefficient VaR |
| historet | double array of dimension nbDates by nbAssets | Historical returns |
| alpha | double | Confidence level of VaR |
| nbEqConst | integer | number of equality constraints (greater than or equal to 0) |
| nbIneqConst | integer | number of inequality constraints (greater than or equal to 0) |
| C | double array of dimension nbAssets by nbEqConst+nbIneqConst | matrix of constraints. If no constraints are given, C=0 |
| b | double array of dimension nbEqConst+nbIneqConst | vector of constraints. If no constraints are given, b=0 |
| minWeights | double array of dimension nbAssets | lower bounds |
| maxWeights | double array of dimension nbAssets | upper bounds |
| istargetPerf | integer | 1 : there is a target performance. 0 : no. |
| targetPerf | double | target performance |
| iskappa4 | integer | 1 if kappa4 provided. 0 if not (then it auto-calibrated by exact penalization): TC are then a constraint, using tcmax. |
| kappa4 | double | coefficient for transaction costs |
| maxTC | double | Maximal Transaction cost authorized. |
| omega0 | double array of dimension nbAssets | initial weights |
| nbKnotsMaxTC | integer | Maximal number of knots for the piecewise polynomial transaction costs. |
| nbKnotsTC | integer array of dimension nbAssets | Number of knots for the piecewise polynomial transaction costs, for each asset. |
| knotsTC | double array of dimension nbKnotsMaxTC by nbAssets | Knots of the pieces. |
| nbMonomialMaxTC | integer | number of monomials for each polynomial. |

| | | |
|---|---|---|
| orderTC | double array of dimension nbMonomialMaxTC by (nbKnotsMaxTC+1)*nbAssets | Orders of the pieces. |
| coefTC | double array of dimension nbMonomialMaxTC by (nbKnotsMaxTC+1)*nbAssets | Coef of the pieces. |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| optimalWeights | double array of dimension nbAssets | optimal portfolio |
| TC | double array of dimension 1 | Transaction cost |
| info | integer array of dimension 1 | diagnostic argument |

### 2.5.5 WREallocGKVaR - Minimize Kernel VaR

#### Overview

It minimizes Kernel VaR.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of asset(s) (greater than 1) |
| kappa2 | double | adjusting coefficient for covariance matrix |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| kappa1 | double | adjusting coefficient for 1st moment |
| expret | double array of dimension nbAssets | expected returns |
| nbDates | integer | number of dates |
| kappa3 | double | aversion coefficient VaR |
| historet | double array of dimension nbDates by nbAssets | Historical returns |
| alpha | double | Confidence level of VaR |
| nbEqConst | integer | number of equality constraints (greater than or equal to 0) |
| nbIneqConst | integer | number of inequality constraints (greater than or equal to 0) |
| C | double array of dimension nbAssets by nbEqConst+nbIneqConst | matrix of constraints. If no constraints are given, C=0 |
| b | double array of dimension nbEqConst+nbIneqConst | vector of constraints. If no constraints are given, b=0 |
| minWeights | double array of dimension nbAssets | lower bounds |
| maxWeights | double array of dimension nbAssets | upper bounds |
| istargetPerf | integer | 1 : there is a target performance. 0 : no. |
| targetPerf | double | target performance |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| optimalWeights | double array of dimension nbAssets | optimal portfolio |
| info | integer array of dimension 1 | diagnostic argument |

### 2.5.6 WREallocGKVaRTC - Minimize VaR with TC constraints

#### Overview

It minimizes Kernel VaR with piece-wise polynomial transaction costs constraints.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of asset(s) (greater than 1) |
| kappa2 | double | adjusting coefficient for covariance matrix |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| kappa1 | double | adjusting coefficient for 1st moment |
| expret | double array of dimension nbAssets | expected returns |
| nbDates | integer | number of dates |
| kappa3 | double | aversion coefficient VaR |
| historet | double array of dimension nbDates by nbAssets | Historical returns |
| alpha | double | Confidence level of VaR |
| nbEqConst | integer | number of equality constraints (greater than or equal to 0) |
| nbIneqConst | integer | number of inequality constraints (greater than or equal to 0) |
| C | double array of dimension nbAssets by nbEqConst+nbIneqConst | matrix of constraints. If no constraints are given, C=0 |
| b | double array of dimension nbEqConst+nbIneqConst | vector of constraints. If no constraints are given, b=0 |
| minWeights | double array of dimension nbAssets | lower bounds |
| maxWeights | double array of dimension nbAssets | upper bounds |
| istargetPerf | integer | 1 : there is a target performance. 0 : no. |
| targetPerf | double | target performance |
| iskappa4 | integer | 1 if kappa4 provided. 0 if not (then it auto-calibrated by exact penalization): TC are then a constraint, using tcmax. |
| kappa4 | double | coefficient for transaction costs |
| maxTC | double | Maximal Transaction cost authorized. |
| omega0 | double array of dimension nbAssets | initial weights |
| nbKnotsMaxTC | integer | Maximal number of knots for the piecewise polynomial transaction costs. |
| nbKnotsTC | integer array of dimension nbAssets | Number of knots for the piecewise polynomial transaction costs, for each asset. |
| knotsTC | double array of dimension nbKnotsMaxTC by nbAssets | Knots of the pieces. |
| nbMonomialMaxTC | integer | number of monomials for each polynomial. |

| | | |
|---|---|---|
| orderTC | double array of dimension nbMonomialMaxTC by (nbKnotsMaxTC+1)*nbAssets | Orders of the pieces. |
| coefTC | double array of dimension nbMonomialMaxTC by (nbKnotsMaxTC+1)*nbAssets | Coef of the pieces. |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| optimalWeights | double array of dimension nbAssets | optimal portfolio |
| TC | double array of dimension 1 | Transaction cost |
| info | integer array of dimension 1 | diagnostic argument |

### 2.5.7 WREallocIT - Index Tracking Minimization

#### Overview

This component computes the optimal portfolio which minimizing the tracking error vs. a benchmark.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of asset(s) (greater than 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| benchmarkCov | double array of dimension nbAssets | covariance of benchmark |
| benchmarkExpectedReturn | double | benchmark mean return supplied |
| nbEqConst | integer | number of equality constraints (greater than or equal to 0) |
| nbIneqConst | integer | number of inequality constraints (greater than or equal to 0) |
| C | double array of dimension nbAssets by nbEqConst+nbIneqConst | matrix of constraints. If no constraints are given, C=0 |
| b | double array of dimension nbEqConst+nbIneqConst | vector of constraints. If no constraints are given, b=0 |
| minWeights | double array of dimension nbAssets | lower bounds |
| maxWeights | double array of dimension nbAssets | upper bounds |
| relativeTargetReturn | double | target performance relative to the benchmark |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| optimalWeights | double array of dimension nbAssets | optimal portfolio |
| info | integer array of dimension 1 | diagnostic argument |

### 2.5.8 WREallocITrfr - Index Tracking Minimization with a risk-free asset

#### Overview

This component computes the optimal portfolio which minimizing the tracking error vs. a benchmark.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of asset(s) (greater than 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| benchmarkCov | double array of dimension nbAssets | covariance of benchmark |
| benchmarkExpectedReturn | double | benchmark mean return supplied |
| riskFreeRate | double | risk-free rate |
| nbEqConst | integer | number of equality constraints (greater than or equal to 0) |
| nbIneqConst | integer | number of inequality constraints (greater than or equal to 0) |
| C | double array of dimension nbAssets by nbEqConst+nbIneqConst | matrix of constraints. If no constraints are given, C=0 |
| b | double array of dimension nbEqConst+nbIneqConst | vector of constraints. If no constraints are given, b=0 |
| minWeights | double array of dimension nbAssets | lower bounds |
| maxWeights | double array of dimension nbAssets | upper bounds |
| minWeightRfr | double | risk-free asset lower bounds |
| maxWeightRfr | double | risk-free asset upper bounds |
| relativeTargetReturn | double | target performance relative to the benchmark |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| optimalWeights | double array of dimension nbAssets | optimal portfolio |
| optimalWeightRfr | double array of dimension 1 | optimal risk-free asset |
| info | integer array of dimension 1 | diagnostic argument |

### 2.5.9 WREallocMMDD - Minimize a maxdrawdown function

#### Overview

This component computes the optimal portfolio with minimal maxdrawdown (and usual 1st and 2nd moments) under a performance constraint (target performance), various linear constraints. Max Drawdown can be put as a constraint too.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of asset(s) (greater than 1) |
| kappa2 | double | adjusting coefficient for 2nd moment (covariance matrix) |
| cov | double array of dimension nbAssets by nbAssets | 2nd moment (covariance matrix) |
| kappa1 | double | adjusting coefficient for 1st moment |
| expret | double array of dimension nbAssets | 1st moment (expected returns) |
| p | integer | number of dates |
| S | double array of dimension p by nbAssets | Historical prices |
| nbEqConst | integer | number of equality constraints (greater than or equal to 0) |
| nbIneqConst | integer | number of inequality constraints (greater than or equal to 0) |
| C | double array of dimension nbAssets by nbEqConst+nbIneqConst | matrix of constraints. If no constraints are given, C=0 |
| b | double array of dimension nbEqConst+nbIneqConst | vector of constraints. If no constraints are given, b=0 |
| minWeights | double array of dimension nbAssets | lower bounds |
| maxWeights | double array of dimension nbAssets | upper bounds |
| isMaxDrawDownConstraint | integer | 1 : there is a hard maxDrawdown constraint performance. 0 : mas drawdown is a soft constraint (objective function) ; in this last case, you should put DDmax equals to 0. |
| DDmax | double | MaxDrawDown target |
| kappa3 | double | adjusting coefficient for drawdown (only used when isMaxDrawDownConstraint=0) |
| istargetPerf | integer | 1 : there is a target performance. 0 : no. |
| targetPerf | double | target performance |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| optimalWeights | double array of dimension nbAssets | optimal portfolio |

| | | |
|---|---|---|
| lowestMaxDDposs | double array of dimension 1 | The lowest ddmax target which could be possible : only available if isMaxDrawDownConstraint = 1 |
| info | integer array of dimension 1 | diagnostic argument |

### 2.5.10 WREallocMRrfr - Mean-Variance (Markowitz) : max return with a risk-free asset

#### Overview

This component computes the optimal portfolio with maximal performance under a volatility constraint (target volatility) and various linear constraints.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of asset(s) (greater than 0) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| riskFreeRate | double | risk-free rate |
| nbEqConst | integer | number of equality constraints (greater than or equal to 0) |
| nbIneqConst | integer | number of inequality constraints (greater than or equal to 0) |
| C | double array of dimension nbAssets by nbEqConst+nbIneqConst | matrix of constraints. If no constraints are given, C=0 |
| b | double array of dimension nbEqConst+nbIneqConst | vector of constraints. If no constraints are given, b=0 |
| minWeights | double array of dimension nbAssets | lower bounds |
| maxWeights | double array of dimension nbAssets | upper bounds |
| minWeightRfr | double | risk-free asset lower bounds |
| maxWeightRfr | double | risk-free asset upper bounds |
| targetVolatility | double | target volatility |
| targetVolatilityPrecision | double | target volatility precision |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| optimalWeights | double array of dimension nbAssets | optimal portfolio (risky assets) |
| optimalWeightRfr | double array of dimension 1 | optimal risk-free asset |
| lambda | double array of dimension 1 | Lagrange multiplier |
| info | integer array of dimension 1 | diagnostic argument |

### 2.5.11 WREallocMV - Mean-Variance (Markowitz)

#### Overview

This component computes the optimal portfolio with minimal volatility under a performance constraint (target return) and various linear constraints.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of asset(s) (greater than 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| nbEqConst | integer | number of equality constraints (greater than or equal to 0) |
| nbIneqConst | integer | number of inequality constraints (greater than or equal to 0) |
| C | double array of dimension nbAssets by nbEqConst+nbIneqConst | matrix of constraints. If no constraints are given, C=0 |
| b | double array of dimension nbEqConst+nbIneqConst | vector of constraints. If no constraints are given, b=0 |
| minWeights | double array of dimension nbAssets | lower bounds |
| maxWeights | double array of dimension nbAssets | upper bounds |
| targetReturn | double | target return |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| optimalWeights | double array of dimension nbAssets | optimal portfolio |
| info | integer array of dimension 1 | diagnostic argument |

### 2.5.12 WREallocMVE - Efficient Frontier (Lagrange Method)

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of point(s) (greater than 1) |
| expectedReturnA | double | mean return of portfolio A |
| expectedReturnB | double | mean return of portfolio B |
| expectedReturnC | double | mean return of portfolio C |
| volatilityA | double | volatility of portfolio A (greater than or equal to 0) |
| volatilityB | double | volatility of portfolio B (greater than or equal to 0) |
| volatilityC | double | volatility of portfolio C (greater than or equal to 0) |
| minExpectedReturn | double | minimum return of universe |
| maxExpectedReturn | double | maximum return of universe |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| mveReturns | double array of dimension nbValues | vector of mean returns |
| mveVolatilities | double array of dimension nbValues | vector of volatilities |
| info | integer array of dimension 1 | diagnostic argument |

### 2.5.13 WREallocMVTC - Asset Allocation with Transaction Costs constraints

#### Overview

This component computes the optimal portfolio with minimal volatility/tracking error subject to transaction costs.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of asset(s) (greater than 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| nbEqConst | integer | number of equality constraints (greater than or equal to 0) |
| nbIneqConst | integer | number of inequality constraints (greater than or equal to 0) |
| C | double array of dimension nbAssets by nbEqConst+nbIneqConst | matrix of constraints. If no constraints are given, C=0 |
| b | double array of dimension nbEqConst+nbIneqConst | vector of constraints. If no constraints are given, b=0 |
| minWeights | double array of dimension nbAssets | lower bounds |
| maxWeights | double array of dimension nbAssets | upper bounds |
| costs | double array of dimension nbAssets | transaction costs (greater than or equal to 0) |
| initWeights | double array of dimension nbAssets | initial portfolio weights |
| alpha | double | transaction costs parameter (greater than or equal to 0) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| optimalWeights | double array of dimension nbAssets | optimal portfolio |
| info | integer array of dimension 1 | diagnostic argument |

### 2.5.14 WREallocMVrfr - Mean-Variance (Markowitz) with a risk-free asset

#### Overview

This component computes the optimal portfolio with minimal volatility under a performance constraint (target return) and various linear constraints.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of asset(s) (greater than 0) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| riskFreeRate | double | risk-free rate |
| nbEqConst | integer | number of equality constraints (greater than or equal to 0) |
| nbIneqConst | integer | number of inequality constraints (greater than or equal to 0) |
| C | double array of dimension nbAssets by nbEqConst+nbIneqConst | matrix of constraints. If no constraints are given, C=0 |
| b | double array of dimension nbEqConst+nbIneqConst | vector of constraints. If no constraints are given, b=0 |
| minWeights | double array of dimension nbAssets | lower bounds |
| maxWeights | double array of dimension nbAssets | upper bounds |
| minWeightRfr | double | risk-free asset lower bounds |
| maxWeightRfr | double | risk-free asset upper bounds |
| targetReturn | double | target return |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| optimalWeights | double array of dimension nbAssets | optimal portfolio (risky assets) |
| optimalWeightRfr | double array of dimension 1 | optimal risk-free asset |
| info | integer array of dimension 1 | diagnostic argument |

### 2.5.15 WREallocRiskBudgeting - Multi-Volatility constrained optimization

#### Overview

This component constructs the portfolio with the highest performance under some volatility budget constraints and various linear constraints.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of asset(s) (greater than 1) |
| Gamma | double array of dimension nbAssets by nbAssets | input matrix (risk constraint) |
| Q | double array of dimension nbAssets by nbAssets | input matrix (objective function) |
| kappa | double | risk aversion coeficient (greater than 0) |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| nbEqConst | integer | number of equality constraints (greater than or equal to 0) |
| nbIneqConst | integer | number of inequality constraints (greater than or equal to 0) |
| C | double array of dimension nbAssets by nbEqConst+nbIneqConst | matrix of constraints. If no constraints are given, C=0 |
| b | double array of dimension nbEqConst+nbIneqConst | vector of constraints. If no constraints are given, b=0 |
| minWeights | double array of dimension nbAssets | lower bounds |
| maxWeights | double array of dimension nbAssets | upper bounds |
| nbClasses | integer | number of risk budgeting constraints (greater than 1) |
| classes | integer array of dimension nbAssets | classes definition |
| volatilityBudget | double array of dimension nbClasses | volatility budget constraints |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| optimalWeights | double array of dimension nbAssets | optimal portfolio |
| lambda | double array of dimension nbClasses | optimal lambda corresponding to the risk budget constraint |
| info | integer array of dimension 1 | diagnostic argument |

### 2.5.16 WREallocRiskBudgetingIT - Multi-Volatility constrained optimization

#### Overview

This component constructs the portfolio with the highest performance under some volatility budget constraints and various linear constraints.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of asset(s) (greater than 1) |
| Gamma | double array of dimension nbAssets by nbAssets | input matrix (risk constraint) |
| kappa | double | risk aversion coeficient (greater than 0) |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| benchmarkCov | double array of dimension nbAssets | covariance of the benchmark with respect to the portfolio |
| benchmarkVar | double | variance of the benchmark |
| nbEqConst | integer | number of equality constraints (greater than or equal to 0) |
| nbIneqConst | integer | number of inequality constraints (greater than or equal to 0) |
| C | double array of dimension nbAssets by nbEqConst+nbIneqConst | matrix of constraints. If no constraints are given, C=0 |
| b | double array of dimension nbEqConst+nbIneqConst | vector of constraints. If no constraints are given, b=0 |
| minWeights | double array of dimension nbAssets | lower bounds |
| maxWeights | double array of dimension nbAssets | upper bounds |
| nbClasses | integer | number of risk budgeting constraints (greater than 1) |
| classes | integer array of dimension nbAssets | classes definition |
| volatilityBudget | double array of dimension nbClasses | index tracking budget constraints |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| optimalWeights | double array of dimension nbAssets | optimal portfolio |
| lambda | double array of dimension nbClasses | optimal lambda corresponding to the risk budget constraint |
| info | integer array of dimension 1 | diagnostic argument |

### 2.5.17 WREallocSharpeRatio - Maximum Sharpe Ratio

#### Overview

This component computes the optimal portfolio with maximum Sharpe Ratio under various linear constraints .

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of asset(s) (greater than 0) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| nbEqConst | integer | number of equality constraints (greater than or equal to 0) |
| nbIneqConst | integer | number of inequality constraints (greater than or equal to 0) |
| C | double array of dimension nbAssets by nbEqConst+nbIneqConst | matrix of constraints. If no constraints are given, C=0. The constraint 'sum of weights equal to 1' already included) |
| b | double array of dimension nbEqConst+nbIneqConst | vector of constraints. If no constraints are given, b=0 |
| minWeights | double array of dimension nbAssets | lower bounds |
| maxWeights | double array of dimension nbAssets | upper bounds |
| riskFreeRate | double | the risk-free rate (with the same frequency as expected returns) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| optimalWeights | double array of dimension nbAssets | optimal portfolio |
| info | integer array of dimension 1 | diagnostic argument |

### 2.5.18 WREallocSharpeRatioRfr - Maximum Sharpe Ratio

#### Overview

This component computes the optimal portfolio with maximum Sharpe Ratio under various linear constraints.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of asset(s) (greater than 0) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| riskFreeRate | double | the risk-free rate (with the same frequency as expected returns) |
| nbEqConst | integer | number of equality constraints (greater than or equal to 0) |
| nbIneqConst | integer | number of inequality constraints (greater than or equal to 0) |
| C | double array of dimension nbAssets by nbEqConst+nbIneqConst | matrix of constraints. If no constraints are given, C=0 |
| b | double array of dimension nbEqConst+nbIneqConst | vector of constraints. If no constraints are given, b=0 |
| minWeights | double array of dimension nbAssets | lower bounds |
| maxWeights | double array of dimension nbAssets | upper bounds |
| minWeightsRfr | double | risk-free asset lower bound |
| maxWeightsRfr | double | risk-free asset upper bound |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| optimalWeights | double array of dimension nbAssets | optimal portfolio |
| optimalWeightRfr | double array of dimension 1 | optimal risk-free asset |
| info | integer array of dimension 1 | diagnostic argument |

## 2.6 Analysis module

### 2.6.1 WREanalysisExanteCC - Ex-ante concentration coefficient (CC)

#### Overview

The concentration coefficient measures portfolio concentration in terms of the asset weightings. In an equal weighted portfolio the CC will be equal to the number of assets. As the portfolio becomes more concentrated in particular assets the CC will be proportionally reduced.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of assets (greater than or equal to 1) |
| weights | double array of dimension nbAssets | portfolio's weight(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| exanteCC | double array of dimension 1 | concentration coefficient |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.2 WREanalysisExanteDRisk - Ex-ante diversifiable risk eliminated

#### Overview

A value of 100 percent represents perfect diversification.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | covariance matrix size (greater than or equal to 1) |
| corr | double array of dimension nbAssets by nbAssets | correlation matrix |
| weights | double array of dimension nbAssets | portfolio's weight(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| riskDiversification | double array of dimension 1 | portfolio's ex-ante percent of diversifiable risk eliminated |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.3 WREanalysisExanteIPC - Ex-ante intra-portfolio correlation (IPC)

#### Overview

Intra-portfolio correlation is a means to quantify diversification. The range is from -1 to 1, with -1 being the most diversified and 1 being the least.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | covariance matrix size (greater than or equal to 1) |
| corr | double array of dimension nbAssets by nbAssets | correlation matrix |
| weights | double array of dimension nbAssets | portfolio's weight(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| exanteIPC | double array of dimension 1 | portfolio's ex-ante intra correlation |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.4 WREanalysisExanteIR - Ex-ante Information Ratio

#### Overview

The information ratio is a risk-adjusted measure of a portfolio relative out-performance to a referential benchmark.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than or equal to 1) |
| nbAssets | integer | covariance matrix size (greater than or equal to 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| benchmarkCov | double array of dimension nbAssets | covariance between portfolio and benchmark |
| benchmarkReturns | double array of dimension nbValues | vector of benchmark's return(s) |
| benchmarkVariance | double | variance of benchmark |
| weights | double array of dimension nbAssets | portfolio's weight(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| exanteInformationRatio | double array of dimension 1 | portfolio's ex-ante Information Ratio |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.5 WREanalysisExanteKurtosis - Ex-ante Kurtosis

#### Overview

Ex-ante Excess Kurtosis.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of values (greater than or equal to 4) |
| nbAssets | integer | number of asset(s) (greater than or equal to 1) |
| weights | double array of dimension nbAssets | portfolio's weight(s) |
| assetsReturns | double array of dimension nbValues by nbAssets | asset(s) returns |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| exanteKurtosis | double array of dimension 1 | portfolio's ex-ante Kurtosis |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.6 WREanalysisExanteMaxDGTVaR - Ex-ante VaR for portfolio including options

#### Overview

Ex-ante VaR for portfolio including options mono-underlyings.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbProducts | integer | number of products in the portfolio |
| V | double array of dimension nbProducts | Values of the products at the current time (m) |
| nbUnderlyings | integer | number of underlyings |
| S | double array of dimension nbUnderlyings | Values of the underlyings at the current time (n) |
| T | double | Time where the VaR will be estimated. |
| weights | double array of dimension nbProducts | weights (nbProducts) |
| nbdates | integer | number of dates |
| ret | double array of dimension nbdates by nbUnderlyings | historical returns |
| expret | double array of dimension nbUnderlyings | expected returns of the underlyings |
| cov | double array of dimension nbUnderlyings by nbUnderlyings | Covariance of the underlyings |
| corres | integer array of dimension nbProducts | Correspondance between underlyings and products (1 to nbUnderlyings) |
| Theta | double array of dimension nbProducts | greek theta |
| Delta | double array of dimension nbProducts | greek delta |
| Gamma | double array of dimension nbProducts | greek gamma |
| alpha | double | Confidence level of the VaR |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| maxvar | double array of dimension 1 | Option Porfolio's Conditional Value-at-Risk |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.7 WREanalysisExanteMaxDGTVaRContrib - Ex-ante VaR for portfolio including options

#### Overview

Ex-ante VaR for portfolio including options mono-underlyings.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbProducts | integer | number of products in the portfolio |
| V | double array of dimension nbProducts | Values of the products at the current time (m) |
| nbUnderlyings | integer | number of underlyings |
| S | double array of dimension nbUnderlyings | Values of the underlyings at the current time (n) |
| T | double | Time where the VaR will be estimated. |
| weights | double array of dimension nbProducts | weights (nbProducts) |
| nbdates | integer | number of dates |
| ret | double array of dimension nbdates by nbUnderlyings | historical returns |
| expret | double array of dimension nbUnderlyings | expected returns of the underlyings |
| cov | double array of dimension nbUnderlyings by nbUnderlyings | Covariance of the underlyings |
| corres | integer array of dimension nbProducts | Correspondance between underlyings and products (1 to nbUnderlyings) |
| Theta | double array of dimension nbProducts | greek theta |
| Delta | double array of dimension nbProducts | greek delta |
| Gamma | double array of dimension nbProducts | greek gamma |
| alpha | double | Confidence level of the VaR |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| marginalVar | double array of dimension nbProducts | Marginal Value-at-risk |
| varcontrib | double array of dimension nbProducts | Contrib Value-at-risk |
| varcontribp | double array of dimension nbProducts | Contrib Value-at-risk (percentage) |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.8 WREanalysisExanteMaxDrawDown - Ex-ante relative Maxdrawdown (percentage)

#### Overview

Ex-ante relative Maxdrawdown (percentage)

#### Inputs

| Variable | Type | Description |
|---|---|---|
| p | integer | number of dates |
| n | integer | number of assets |
| S | double array of dimension p by n | input prices |
| omega | double array of dimension n | input data |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| mdd | double array of dimension 1 | |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.9 WREanalysisExanteMaxDrawDownX - Ex-ante relative Maxdrawdown (percentage)

#### Overview

Ex-ante relative Maxdrawdown (percentage) with indices

#### Inputs

| Variable | Type | Description |
|---|---|---|
| p | integer | number of dates |
| n | integer | number of assets |
| S | double array of dimension p by n | input prices |
| omega | double array of dimension n | input data |
| epsi | double | number of values |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| mdd | double array of dimension 1 | |
| nbind | integer array of dimension 1 | |
| ind1 | integer array of dimension n | index 1 |
| ind2 | integer array of dimension n | index 2 |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.10 WREanalysisExanteMaxVaR - Ex-ante Value-at-Risk (Upper Bound)

#### Overview

This component computes a robust approximation of the Value-at-Risk which allows deals with non Gaussian distributions and provides a more robust ex-ante Value-at-Risk.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | covariance matrix size (greater than or equal to 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| weights | double array of dimension nbAssets | portfolio's weight(s) |
| probabilityLevel | double | probability level (greater than 0 and lower than 1) - if exanteMaxValueAtRisk lower than 0: probability that the loss will be lower to exanteMaxValueAtRisk - if exanteMaxValueAtRisk greater than 0: probability that the gain will be upper to exanteMaxValueAtRisk |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| exanteMaxValueAtRisk | double array of dimension 1 | portfolio's ex-ante Value-at-Risk |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.11 WREanalysisExanteModifiedSharpeRatio - Ex-ante Modified Sharpe Ratio

#### Overview

The Modified Sharpe ratio is the ratio of the excess return divided by the Modified Value-at-Risk.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of values (greater than or equal to 4) |
| nbAssets | integer | number of asset(s) (greater than or equal to 1) |
| weights | double array of dimension nbAssets | portfolio's weight(s) |
| assetsReturns | double array of dimension nbValues by nbAssets | asset(s) returns |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| riskFreeRate | double | risk free rate (ex. LIBOR 3 months) |
| probabilityLevel | double | probability level (0 lower than probabilityLevel lower than 1) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| exanteModifiedSharpeRatio | double array of dimension 1 | portfolio's ex-ante modified Sharpe ratio |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.12 WREanalysisExanteModifiedVaR - Ex-ante Modified Value-at-Risk

#### Overview

The modified Value-at-Risk (VaR) adjusts the traditional Gaussian VaR with the Skewness and Kurtosis of the distribution.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of values (greater than or equal to 4) |
| nbAssets | integer | number of asset(s) (greater than or equal to 1) |
| weights | double array of dimension nbAssets | portfolio's weight(s) |
| assetsReturns | double array of dimension nbValues by nbAssets | asset(s) returns |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| probabilityLevel | double | probability level (greater than 0 and lower than 1) - if exanteModifiedValueAtRisk lower than 0: probability that the loss will be lower to exanteModifiedValueAtRisk - if exanteModifiedValueAtRisk greater than 0: probability that the gain will be upper to exanteModifiedValueAtRisk |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| exanteModifiedValueAtRisk | double array of dimension 1 | portfolio's ex-ante modified Value-at-Risk |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.13 WREanalysisExanteNormalCVaR - Ex-ante Conditional Value-at-Risk (Gaussian)

#### Overview

This component computes the Gaussian ex-ante Conditional Value-at-Risk (CVaR).

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | covariance matrix size (greater than or equal to 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| weights | double array of dimension nbAssets | portfolio's weight(s) |
| probabilityLevel | double | probability level (0lower than probabilityLevellower than 1) -if exanteNormalConditionalValueAtRisklower than 0: probability that the loss will be lower to exanteNormalConditionalValueAtRisk -if exanteNormalConditionalValueAtRiskgreater than 0: probability that the gain will be upper to it |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| exanteNormalConditionalValueAt | double array of dimension 1 | portfolio's ex-ante Normal conditional Value-at-Risk |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.14 WREanalysisExanteNormalDGTCVaR - Ex-ante VaR for portfolio including options

#### Overview

Ex-ante VaR for portfolio including options mono-underlyings.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbProducts | integer | number of products in the portfolio |
| V | double array of dimension nbProducts | Values of the products at the current time (m) |
| nbUnderlyings | integer | number of underlyings |
| S | double array of dimension nbUnderlyings | Values of the underlyings at the current time (n) |
| T | double | Time where the VaR will be estimated. |
| weights | double array of dimension nbProducts | weights (nbProducts) |
| nbdates | integer | number of dates |
| ret | double array of dimension nbdates by nbUnderlyings | historical returns |
| expret | double array of dimension nbUnderlyings | expected returns of the underlyings |
| cov | double array of dimension nbUnderlyings by nbUnderlyings | Covariance of the underlyings |
| corres | integer array of dimension nbProducts | Correspondance between underlyings and products (1 to nbUnderlyings) |
| Theta | double array of dimension nbProducts | greek theta |
| Delta | double array of dimension nbProducts | greek delta |
| Gamma | double array of dimension nbProducts | greek gamma |
| alpha | double | Confidence level of the VaR |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| ncvar | double array of dimension 1 | Option Porfolio's Conditional Value-at-Risk |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.15 WREanalysisExanteNormalDGTCVaRContrib - Ex-ante VaR for portfolio including options

#### Overview

Ex-ante VaR for portfolio including options mono-underlyings.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbProducts | integer | number of products in the portfolio |
| V | double array of dimension nbProducts | Values of the products at the current time (m) |
| nbUnderlyings | integer | number of underlyings |
| S | double array of dimension nbUnderlyings | Values of the underlyings at the current time (n) |
| T | double | Time where the VaR will be estimated. |
| weights | double array of dimension nbProducts | weights (nbProducts) |
| nbdates | integer | number of dates |
| ret | double array of dimension nbdates by nbUnderlyings | historical returns |
| expret | double array of dimension nbUnderlyings | expected returns of the underlyings |
| cov | double array of dimension nbUnderlyings by nbUnderlyings | Covariance of the underlyings |
| corres | integer array of dimension nbProducts | Correspondance between underlyings and products (1 to nbUnderlyings) |
| Theta | double array of dimension nbProducts | greek theta |
| Delta | double array of dimension nbProducts | greek delta |
| Gamma | double array of dimension nbProducts | greek gamma |
| alpha | double | Confidence level of the VaR |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| mncvar | double array of dimension nbProducts | Marginal Conditional Value-at-risk |
| ncvarcontrib | double array of dimension nbProducts | Contrib Conditional Value-at-risk |
| ncvarcontribp | double array of dimension nbProducts | Contrib Conditional Value-at-risk (percentage) |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.16 WREanalysisExanteNormalDGTVaR - Ex-ante VaR for portfolio including options

#### Overview

Ex-ante VaR for portfolio including options mono-underlyings.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbProducts | integer | number of products in the portfolio |
| V | double array of dimension nbProducts | Values of the products at the current time (m) |
| nbUnderlyings | integer | number of underlyings |
| S | double array of dimension nbUnderlyings | Values of the underlyings at the current time (n) |
| T | double | Time where the VaR will be estimated. |
| weights | double array of dimension nbProducts | weights (nbProducts) |
| nbdates | integer | number of dates |
| ret | double array of dimension nbdates by nbUnderlyings | historical returns |
| expret | double array of dimension nbUnderlyings | expected returns of the underlyings |
| cov | double array of dimension nbUnderlyings by nbUnderlyings | Covariance of the underlyings |
| corres | integer array of dimension nbProducts | Correspondance between underlyings and products (1 to nbUnderlyings) |
| Theta | double array of dimension nbProducts | greek theta |
| Delta | double array of dimension nbProducts | greek delta |
| Gamma | double array of dimension nbProducts | greek gamma |
| alpha | double | Confidence level of the VaR |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| nvar | double array of dimension 1 | Option Porfolio's Value-at-Risk |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.17 WREanalysisExanteNormalDGTVaRContrib - Ex-ante VaR for portfolio including options

#### Overview

Ex-ante VaR for portfolio including options mono-underlyings.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbProducts | integer | number of products in the portfolio |
| V | double array of dimension nbProducts | Values of the products at the current time (m) |
| nbUnderlyings | integer | number of underlyings |
| S | double array of dimension nbUnderlyings | Values of the underlyings at the current time (n) |
| T | double | Time where the VaR will be estimated. |
| weights | double array of dimension nbProducts | weights (nbProducts) |
| nbdates | integer | number of dates |
| ret | double array of dimension nbdates by nbUnderlyings | historical returns |
| expret | double array of dimension nbUnderlyings | expected returns of the underlyings |
| cov | double array of dimension nbUnderlyings by nbUnderlyings | Covariance of the underlyings |
| corres | integer array of dimension nbProducts | Correspondance between underlyings and products (1 to nbUnderlyings) |
| Theta | double array of dimension nbProducts | greek theta |
| Delta | double array of dimension nbProducts | greek delta |
| Gamma | double array of dimension nbProducts | greek gamma |
| alpha | double | Confidence level of the VaR |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| mnvar | double array of dimension nbProducts | Marginal Value-at-risk |
| nvarcontrib | double array of dimension nbProducts | Marginal value-at-risk contribution |
| nvarcontribp | double array of dimension nbProducts | Marginal value-at-risk contribution (percentage) |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.18 WREanalysisExanteNormalShortfall - Ex-ante Shortfall Probability (Gaussian)

#### Overview

This component computes the Gaussian ex-ante shortfall probability. Shortfall is the probability (Gaussian) of return falling short of a certain threshold return mu.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | covariance matrix size (greater than or equal to 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| weights | double array of dimension nbAssets | portfolio's weight(s) |
| mu | double | threshold return |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| exanteNormalShortfall | double array of dimension 1 | portfolio's ex-ante normal Shortfall |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.19 WREanalysisExanteNormalVaR - Ex-ante Value-at-Risk (Gaussian)

#### Overview

This component computes the Gaussian ex-ante Value-at-Risk (VaR) with the parametric Variance-Covariance method.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | covariance matrix size (greater than or equal to 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| weights | double array of dimension nbAssets | portfolio's weight(s) |
| probabilityLevel | double | probability level (greater than 0 and lower than 1) - if exanteNormalValueAtRisk lower than 0: probability that the loss will be lower to exanteNormalValueAtRisk - if exanteNormalValueAtRisk greater than 0: probability that the gain will be upper to exanteNormalValueAtRisk |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| exanteNormalValueAtRisk | double array of dimension 1 | portfolio's ex-ante Normal Value-at-Risk |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.20 WREanalysisExanteReturn - Ex-ante Mean Return

#### Overview

This function computes the ex-ante mean return (performance) of a given portfolio.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of asset(s) (greater than or equal to 1) |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| weights | double array of dimension nbAssets | weight(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| exanteReturn | double array of dimension 1 | portfolio's mean return |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.21 WREanalysisExanteSharpeRatio - Ex-ante Sharpe Ratio

#### Overview

The Sharpe ratio is a risk-adjusted measure calculated using the portfolio's ex-ante volatility (standard deviation) and portfolio's ex-ante excess return (portfolio return - risk-free rate) to determine reward per unit of risk.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | covariance matrix size (greater than or equal to 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| weights | double array of dimension nbAssets | weight(s) |
| riskFreeRate | double | risk free rate (ex. LIBOR 3 months) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| exanteSharpeRatio | double array of dimension 1 | portfolio's ex-ante Sharpe Ratio |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.22 WREanalysisExanteSkewness - Ex-ante Skewness

#### Overview

Ex-ante Skewness.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of values (greater than or equal to 3) |
| nbAssets | integer | number of asset(s) (greater than or equal to 1) |
| weights | double array of dimension nbAssets | weight(s) |
| assetsReturns | double array of dimension nbValues by nbAssets | asset(s) returns |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| exanteSkewness | double array of dimension 1 | portfolio's ex-ante Skewness |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.23 WREanalysisExanteSmoothedNonParametricCVaR - Ex-ante Non Parametric Conditional Value-at-Risk

#### Overview

This function computes the non parametric ex-ante conditional Value-at-Risk with gaussian kernels.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbdates | integer | number of dates |
| nbAssets | integer | number of assets |
| ret | double array of dimension nbdates by nbAssets | historical returns |
| weights | double array of dimension nbAssets | portfolio's weights |
| alpha | double | probability level (for example: 0.95) |
| prec | double | precision for density integration |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| CValueAtRisk | double array of dimension 1 | non parametric conditional Value-at-Risk |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.24 WREanalysisExanteSmoothedNonParametricVaR - Ex-ante Non Parametric Value-at-Risk

#### Overview

This function computes the non parametric ex-ante Value-at-Risk with gaussian kernels.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbdates | integer | number of dates |
| nbAssets | integer | number of assets |
| ret | double array of dimension nbdates by nbAssets | historical returns |
| weights | double array of dimension nbAssets | portfolio's weights |
| alpha | double | probability level (for example: 0.95) |
| prec | double | precision for density integration |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| ValueAtRisk | double array of dimension 1 | non parametric Value-at-Risk |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.25 WREanalysisExanteTE - Ex-ante Tracking Error

#### Overview

The tracking error quantifies the degree to which the strategy differed from an index or a benchmark when using an indexing or any other benchmarking strategy.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | covariance matrix size (greater than or equal to 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| benchmarkCov | double array of dimension nbAssets | covariance between portfolio and benchmark |
| benchmarkVariance | double | variance of benchmark |
| weights | double array of dimension nbAssets | weight(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| exanteTrackingError | double array of dimension 1 | portfolio's ex-ante Tracking Error |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.26 WREanalysisExanteVolatility - Ex-ante Volatility

#### Overview

This component computes the ex-ante volatility (standard deviation) of a portfolio.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | covariance matrix size (greater than or equal to 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| weights | double array of dimension nbAssets | weight(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| exanteVolatility | double array of dimension 1 | portfolio's ex-ante volatility |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.27 WREanalysisExpostCorrCoef - Correlation Coefficient (Ex-post)

#### Overview

This function computes the ex-post correlation coefficient between two random variables X and Y.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of values (greater than 1) |
| X | double array of dimension nbValues | random variable X |
| Y | double array of dimension nbValues | random variable Y |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostCorrCoef | double array of dimension 1 | expost correlation coefficient |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.28 WREanalysisExpostCorrCoefX - Correlation Coefficient (Ex-post) with width estimation

#### Overview

This function computes the ex-post correlation coefficient between two random variables X and Y.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of values (greater than 1) |
| widthPerf | integer | performance width estimation (greater than 1) |
| X | double array of dimension nbValues | random variable X |
| Y | double array of dimension nbValues | random variable Y |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostCorrCoef | double array of dimension 1 | expost correlation coefficient |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.29 WREanalysisExpostDownSide - Ex-post Downside Risk

#### Overview

Downside is a generalization of the semi-volatility as is defined as the k-moment of returns below a certain threshold return.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| k | integer | k-moment (k=1,2,3,4) |
| mu | double | threshold return |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostDownSide | double array of dimension 1 | ex-post down side |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.30 WREanalysisExpostDownSideX - Ex-post Downside Risk with width estimation

#### Overview

Downside is a generalization of the semi-volatility as is defined as the k-moment of returns below a certain threshold return.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| widthPerf | integer | performance width estimation (greater than 1) |
| widthRisk | integer | risk width estimation (greater than 1) |
| k | integer | k-moment (k=1,2,3,4) |
| mu | double | threshold return |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostDownSide | double array of dimension 1 | ex-post down side |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.31 WREanalysisExpostHistoricalVaR - Historical Value-at-Risk

#### Overview

Historical Value-at-Risk (with interpolation if needed).

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbsimul | integer | number of simulations |
| returns | double array of dimension nbsimul | return(s) |
| alpha | double | probability level |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| var | double array of dimension 1 | Historical Value-at-Risk |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.32 WREanalysisExpostIR - Ex-post Information Ratio

#### Overview

The information ratio is a risk-adjusted measure of a portfolio relative out-performance to a referential benchmark.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| benchmarkReturns | double array of dimension nbValues | benchmark's return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostInformationRatio | double array of dimension 1 | ex-post information ratio |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.33 WREanalysisExpostIRX - Ex-post Information Ratio with width estimation

#### Overview

The information ratio is a risk-adjusted measure of a portfolio relative out-performance to a referential benchmark.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| widthPerf | integer | performance width estimation (greater than 1) |
| widthRisk | integer | risk width estimation (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| benchmarkReturns | double array of dimension nbValues | benchmark's return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostInformationRatio | double array of dimension 1 | ex-post information ratio |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.34 WREanalysisExpostKurtosis - Ex-post Kurtosis

#### Overview

Ex-post Excess kurtosis.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 3) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostKurtosis | double array of dimension 1 | ex-post kurtosis |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.35 WREanalysisExpostKurtosisX - Ex-post Kurtosis with width estimation

#### Overview

Ex-post kurtosis with width estimation.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 3) |
| widthPerf | integer | performance width estimation (greater than 1) |
| widthRisk | integer | risk width estimation (greater than 3) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostKurtosis | double array of dimension 1 | ex-post kurtosis |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.36 WREanalysisExpostMaxDrawDown - Relative Maxdrawdown (percentage)

**Overview**

Relative Maxdrawdown (percentage)

**Inputs**

| Variable | Type | Description |
|---|---|---|
| n | integer | number of values |
| S | double array of dimension n | input prices |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| mdd | double array of dimension 1 | |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.37 WREanalysisExpostMaxDrawDownX - Relative Maxdrawdown (percentage)

#### Overview

Relative Maxdrawdown (percentage) with indices

#### Inputs

| Variable | Type | Description |
|---|---|---|
| n | integer | number of values |
| S | double array of dimension n | input prices |
| epsi | double | number of values |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| mdd | double array of dimension 1 | |
| nbind | integer array of dimension 1 | |
| ind1 | integer array of dimension n | index 1 |
| ind2 | integer array of dimension n | index 2 |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.38 WREanalysisExpostMaxLoss - Ex-post Maximum Loss

#### Overview

The Maximum Loss is the minimum cumulated return from the beginning in a certain time period.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than or equal to 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostMaxLoss | double array of dimension 1 | ex-post maximum loss |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.39 WREanalysisExpostMaxLossX - Ex-post Maximum Lo¿ss with width estimation

#### Overview

The Maximum Loss is the minimum cumulated return from the beginning in a certain time period.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than or equal to 1) |
| widthPerf | integer | performance width estimation (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostMaxLoss | double array of dimension 1 | ex-post maximum loss |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.40 WREanalysisExpostMaxVaR - Ex-post Value-at-Risk (Upper Bound)

#### Overview

This function computes a robust bound of the Value-at-Risk which allows deals with non Gaussian distributions and provides a more robust ex-post Value-at-Risk.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| probabilityLevel | double | probability level (0 lower than probabilityLevel lower than 1) - if expostMaxValueAtRisk lower than 0: probability that the loss will be lower to expostMaxValueAtRisk - if expostMaxValueAtRisk greater than 0: probability that the gain will be upper to expostMaxValueAtRisk |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostMaxValueAtRisk | double array of dimension 1 | ex-post Value-at-Risk max |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.41 WREanalysisExpostMaxVaRX - Ex-post Value-at-Risk (Upper Bound) with width estimation

#### Overview

This function computes a robust bound of the Value-at-Risk which allows deals with non Gaussian distributions and provides a more robust ex-post Value-at-Risk.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| widthPerf | integer | performance width estimation (greater than 1) |
| widthRisk | integer | risk width estimation (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| probabilityLevel | double | probability level (0 lower than probabilityLevel lower than 1) - if expostMaxValueAtRisk lower than 0: probability that the loss will be lower to expostMaxValueAtRisk - if expostMaxValueAtRisk greater than 0: probability that the gain will be upper to expostMaxValueAtRisk |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostMaxValueAtRisk | double array of dimension 1 | ex-post Value-at-Risk max |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.42 WREanalysisExpostModifiedSharpeRatio - Ex-post Modified Sharpe ratio

#### Overview

The ex-post Modified Sharpe ratio is the ratio of the excess return (ex-post portfolio's mean return - risk-free rate, divided by the ex-post Modified Value-at-Risk.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 3) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| riskFreeRate | double | risk free rate (ex. LIBOR 3 months) |
| probabilityLevel | double | probability level (0 lower than probabilityLevel lower than 1) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostModifiedSharpeRatio | double array of dimension 1 | ex-post modified Sharpe ratio |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.43 WREanalysisExpostModifiedSharpeRatioX - Ex-post Modified Sharpe ratio with width estimation

#### Overview

The ex-post Modified Sharpe ratio is the ratio of the excess return (ex-post portfolio's mean return - risk-free rate, divided by the ex-post Modified Value-at-Risk.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 3) |
| widthPerf | integer | performance width estimation (greater than 1) |
| widthRisk | integer | risk width estimation (greater than 3) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| riskFreeRate | double | risk free rate (ex. LIBOR 3 months) |
| probabilityLevel | double | probability level (0 lower than probabilityLevel lower than 1) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostModifiedSharpeRatio | double array of dimension 1 | ex-post modified Sharpe ratio |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.44 WREanalysisExpostModifiedVaR - Ex-post Modified Value-at-Risk

#### Overview

The modified Value-at-Risk (VaR) adjusts the traditional Gaussian VaR with the skewness and kurtosis of the distribution.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 3) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| probabilityLevel | double | probability level (greater than 0 and lower than 1) - if expostModifiedValueAtRisk lower than 0: probability that the loss will be lower to expostModifiedValueAtRisk - if expostModifiedValueAtRisk greater than 0: probability that the gain will be upper to expostModifiedValueAtRisk |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostModifiedValueAtRisk | double array of dimension 1 | ex-post modified Value-at-Risk |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.45 WREanalysisExpostModifiedVaRX - Ex-post Modified Value-at-Risk with width estimation

#### Overview

The modified Value-at-Risk (VaR) adjusts the traditional Gaussian VaR with the skewness and kurtosis of the distribution.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 3) |
| widthPerf | integer | performance width estimation (greater than 1) |
| widthRisk | integer | risk width estimation (greater than 3) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| probabilityLevel | double | probability level (greater than 0 and lower than 1) - if expostModifiedValueAtRisk lower than 0: probability that the loss will be lower to expostModifiedValueAtRisk - if expostModifiedValueAtRisk greater than 0: probability that the gain will be upper to expostModifiedValueAtRisk |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostModifiedValueAtRisk | double array of dimension 1 | ex-post modified Value-at-Risk |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.46 WREanalysisExpostNormalCVaR - Ex-post Conditional Value-at-Risk (Gaussian)

**Overview**

This component computes the Gaussian ex-post Conditional Value-at-Risk (CVaR).

**Inputs**

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| probabilityLevel | double | probability level (0lower than probabilityLevellower than 1) -if expostNormalConditionalValueAtRisklower than 0: probability that the loss will be lower to expostNormalConditionalValueAtRisk -if expostNormalConditionalValueAtRiskgreater than 0: probability that the gain will be upper to it |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| expostNormalConditionalValueAt | double array of dimension 1 | ex-post Normal conditional Value-at-Risk |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.47 WREanalysisExpostNormalCVaRX - Ex-post Conditional Value-at-Risk (Gaussian) with width estimation

#### Overview

This component computes the Gaussian ex-post Conditional Value-at-Risk (CVaR).

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| widthPerf | integer | performance width estimation (greater than 1) |
| widthRisk | integer | risk width estimation (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| probabilityLevel | double | probability level (0lower than probabilityLevellower than 1) -if expostNormalConditionalValueAtRisklower than 0: probability that the loss will be lower to expostNormalConditionalValueAtRisk -if expostNormalConditionalValueAtRiskgreater than 0: probability that the gain will be upper to it |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostNormalConditionalValueAt | double array of dimension 1 | ex-post Normal conditional Value-at-Risk |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.48 WREanalysisExpostNormalShortfall - Ex-post Shortfall Probability (Gaussian)

#### Overview

Shortfall is the probability (Gaussian) of return falling short of a certain threshold return mu.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| mu | double | threshold return |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostNormalShortfall | double array of dimension 1 | ex-post normal Shortfall |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.49 WREanalysisExpostNormalShortfallX - Ex-post Shortfall Probability (Gaussian) with width estimation

#### Overview

Shortfall is the probability (Gaussian) of return falling short of a certain threshold return mu.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| widthPerf | integer | performance width estimation (greater than 1) |
| widthRisk | integer | risk width estimation (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| mu | double | threshold return |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostNormalShortfall | double array of dimension 1 | ex-post normal Shortfall |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.50 WREanalysisExpostNormalVaR - Ex-post Value-at-Risk (Gaussian)

#### Overview

This function computes the Gaussian ex-post Value-at-Risk (VaR).

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| probabilityLevel | double | probability level (greater than 0 and lower than 1) - if expostNormalValueAtRisk lower than 0: probability that the loss will be lower to expostNormalValueAtRisk - if expostNormalValueAtRiskgreater than 0: probability that the gain will be upper to expostNormalValueAtRisk |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostNormalValueAtRisk | double array of dimension 1 | ex-post Normal Value-at-Risk |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.51 WREanalysisExpostNormalVaRX - Ex-post Value-at-Risk (Gaussian) with width estimation

#### Overview

This function computes the Gaussian ex-post Value-at-Risk (VaR).

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| widthPerf | integer | performance width estimation (greater than 1) |
| widthRisk | integer | risk width estimation (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| probabilityLevel | double | probability level (greater than 0 and lower than 1) - if expostNormalValueAtRisk lower than 0: probability that the loss will be lower to expostNormalValueAtRisk - if expostNormalValueAtRiskgreater than 0: probability that the gain will be upper to expostNormalValueAtRisk |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostNormalValueAtRisk | double array of dimension 1 | ex-post Normal Value-at-Risk |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.52 WREanalysisExpostReturn - Ex-post Mean Return

#### Overview

This function computes the ex-post mean return (performance).

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than or equal to 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostReturn | double array of dimension 1 | ex-post mean return |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.53 WREanalysisExpostReturnX - Ex-post Mean Return with width estimation

#### Overview

This function computes the ex-post mean return (performance).

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than or equal to 1) |
| widthPerf | integer | performance width estimation (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostReturn | double array of dimension 1 | ex-post mean return |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.54 WREanalysisExpostSemiVolatility - Ex-post Semi-Volatility

#### Overview

The ex-post semi-volatility is the volatility (standard deviation) of negative deviations from mean.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostSemiVolatility | double array of dimension 1 | ex-post semi-volatility |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.55 WREanalysisExpostSemiVolatilityX - Ex-post Semi-Volatility with width estimation

#### Overview

The ex-post semi-volatility is the volatility (standard deviation) of negative deviations from mean.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| widthRisk | integer | risk width estimation (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostSemiVolatility | double array of dimension 1 | ex-post semi-volatility |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.56 WREanalysisExpostSharpeRatio - Ex-post Sharpe Ratio

#### Overview

The ex-post Sharpe ratio is a risk-adjusted measure, calculated using the portfolio's ex-post volatility (standard deviation) and portfolio's ex-post excess return (portfolio return - risk-free rate) to determine reward per unit of risk.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| riskFreeRate | double | risk free rate (ex. LIBOR 3 months) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostSharpeRatio | double array of dimension 1 | ex-post Sharpe ratio |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.57 WREanalysisExpostSharpeRatioX - Ex-post Sharpe Ratio with width estimation

#### Overview

The ex-post Sharpe ratio is a risk-adjusted measure, calculated using the portfolio's ex-post volatility (standard deviation) and portfolio's ex-post excess return (portfolio return - risk-free rate) to determine reward per unit of risk.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| widthPerf | integer | performance width estimation (greater than 1) |
| widthRisk | integer | risk width estimation (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| riskFreeRate | double | risk free rate (ex. LIBOR 3 months) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostSharpeRatio | double array of dimension 1 | ex-post Sharpe ratio |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.58 WREanalysisExpostSkewness - Ex-post Skewness

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 2) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostSkewness | double array of dimension 1 | ex-post skewness |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.59 WREanalysisExpostSkewnessX - Ex-post Skewness with width estimation

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 2) |
| widthPerf | integer | performance width estimation (greater than 1) |
| widthRisk | integer | risk width estimation (greater than 2) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostSkewness | double array of dimension 1 | ex-post skewness |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.60 WREanalysisExpostSmoothedNonParametricCVaR - Ex-post Non Parametric Conditional Value-at-Risk

#### Overview

This function computes the non parametric ex-post Conditional Value-at-Risk with gaussian kernels.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbdates | integer | number of historical returns |
| ret | double array of dimension nbdates | historical returns |
| alpha | double | probability level (for example: 0.95) |
| prec | double | precision for density integration |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostNonParametricValueAtRisk | double array of dimension 1 | non parametric Value-at-Risk |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.61 WREanalysisExpostSmoothedNonParametricVaR - Ex-post Non Parametric Value-at-Risk

#### Overview

This function computes the non parametric ex-post Value-at-Risk with gaussian kernels.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbdates | integer | number of historical returns |
| ret | double array of dimension nbdates | historical returns |
| alpha | double | probability level (for example: 0.95) |
| prec | double | precision for density integration |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostNonParametricValueAtRisk | double array of dimension 1 | non parametric Value-at-Risk |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.62 WREanalysisExpostSortinoRatio - Ex-post Sortino Ratio

#### Overview

The Sortino ratio is the excess return over risk-free rate over the downside semi-variance, so it measures the return to bad volatility.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| riskFreeRate | double | risk free rate (ex. LIBOR 3 months) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostSortinoRatio | double array of dimension 1 | ex-post Sortino ratio |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.63 WREanalysisExpostSortinoRatioX - Ex-post Sortino Ratio with width estimation

#### Overview

The Sortino ratio is the excess return over risk-free rate over the downside semi-variance, so it measures the return to bad volatility.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| widthPerf | integer | performance width estimation (greater than 1) |
| widthRisk | integer | risk width estimation (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| riskFreeRate | double | risk free rate (ex. LIBOR 3 months) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostSortinoRatio | double array of dimension 1 | ex-post Sortino ratio |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.64 WREanalysisExpostTE - Ex-post Tracking Error

#### Overview

The ex-post tracking error measure the variance of the difference between the portfolio's returns and benchmark's returns.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| benchmarkReturns | double array of dimension nbValues | benchmark's return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostTrackingError | double array of dimension 1 | ex-post tracking error |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.65 WREanalysisExpostTEX - Ex-post Tracking Error with width estimation

#### Overview

The ex-post tracking error measure the variance of the difference between the portfolio's returns and benchmark's returns.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| widthRisk | integer | risk width estimation (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| benchmarkReturns | double array of dimension nbValues | benchmark's return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostTrackingError | double array of dimension 1 | ex-post tracking error |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.66 WREanalysisExpostVolatility - Ex-post Volatility

#### Overview

The volatility (or standard deviation) is a statistical measure of the historical returns. The ex-post volatility is usually computed using daily or monthly returns.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostVolatility | double array of dimension 1 | ex-post volatility |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.67 WREanalysisExpostVolatilityExp - Ex-post Exponentially Weighted Volatility

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| lambda | double | decay factor (greater than or equal to 0 and lower than or equal to1) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostVolatilityExp | double array of dimension 1 | ex-post exponentially weighted volatility |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.68 WREanalysisExpostVolatilityExpX - Ex-post Exponentially Weighted Volatility with width estimation

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| widthRisk | integer | risk width estimation (greater than X) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |
| lambda | double | decay factor (greater than or equal to 0 and lower than or equal to1) |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| expostVolatilityExp | double array of dimension 1 | ex-post exponentially weighted volatility |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.69 WREanalysisExpostVolatilityX - Ex-post Volatility with width estimation

#### Overview

The volatility (or standard deviation) is a statistical measure of the historical returns. The ex-post volatility is usually computed using daily or monthly returns.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| widthRisk | integer | risk width estimation (greater than 1) |
| portfolioReturns | double array of dimension nbValues | portfolio's return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expostVolatility | double array of dimension 1 | ex-post volatility |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.70 WREanalysisFutureValues - Portfolio's Future Values (NAV)

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of point(s) (greater than or equal to 1) |
| nbAssets | integer | number of asset(s) (greater than or equal to 1) |
| assetsValues | double array of dimension nbValues by nbAssets | asset(s) values |
| weights | double array of dimension nbAssets | weight(s) |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| futureValues | double array of dimension nbValues | portfolio's future values |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.71 WREanalysisGaussianKernel - Density Estimation by Gaussian Kernel Method

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| n | integer | number of return(s) (greater than or equal to 2) |
| m | integer | number of grid point(s) (greater than or equal to 1) |
| x | double array of dimension n | return(s) |
| y | double array of dimension m | grid (increasing order) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| z | double array of dimension m | Gaussian kernel estimation |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.72 WREanalysisMaxVaRContrib - Max Value-at-Risk Contribution (Marginal)

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | covariance matrix size (greater than or equal to 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| weights | double array of dimension nbAssets | portfolio's weight(s) |
| probabilityLevel | double | probability level (0 lower than probabilityLevel lower than 1) - if exanteNormalValueAtRisk lower than 0: probability that the loss will be lower to exanteNormalValueAtRisk - if exanteNormalValueAtRisk greater than 0: probability that the gain will be upper to exanteNormalValueAtRisk |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| marginalMaxVaR | double array of dimension nbAssets | Marginal Max VaR |
| maxVaRContribution | double array of dimension nbAssets | marginal Max VaR contribution |
| maxVaRContributionP | double array of dimension nbAssets | marginal Max VaR contribution in percent |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.73 WREanalysisNormalCVaRContrib - Gaussian Conditional Value-at-Risk Contribution (Marginal)

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | covariance matrix size (greater than or equal to 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| weights | double array of dimension nbAssets | portfolio's weight(s) |
| probabilityLevel | double | probability level (0 lower than probabilityLevel lower than 1) - if exanteNormalValueAtRisk lower than 0: probability that the loss will be lower to exanteNormalValueAtRisk - if exanteNormalValueAtRisk greater than 0: probability that the gain will be upper to exanteNormalValueAtRisk |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| marginalNormalCVaR | double array of dimension nbAssets | Marginal Gaussian CVaR |
| normalCVaRContribution | double array of dimension nbAssets | marginal Gaussian CVaR contribution |
| normalCVaRContributionP | double array of dimension nbAssets | marginal Gaussian CVaR contribution in percent |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.74 WREanalysisNormalVaRContrib - Gaussian Value-at-Risk Contribution (Marginal)

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | covariance matrix size (greater than or equal to 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| weights | double array of dimension nbAssets | portfolio's weight(s) |
| probabilityLevel | double | probability level (0 lower than probabilityLevel lower than 1) - if exanteNormalValueAtRisk lower than 0: probability that the loss will be lower to exanteNormalValueAtRisk - if exanteNormalValueAtRisk greater than 0: probability that the gain will be upper to exanteNormalValueAtRisk |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| marginalNormalVaR | double array of dimension nbAssets | Marginal Gaussian VaR |
| normalVaRContribution | double array of dimension nbAssets | marginal Gaussian VaR contribution |
| normalVaRContributionP | double array of dimension nbAssets | marginal Gaussian VaR contribution in percent |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.75 WREanalysisPastValues - Portfolio's Past Values (NAV)

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of point(s) (greater than or equal to 1) |
| nbAssets | integer | number of asset(s) (greater than or equal to 1) |
| assetsValues | double array of dimension nbValues by nbAssets | asset(s) values |
| weights | double array of dimension nbAssets | weight(s) |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| pastValues | double array of dimension nbValues | portfolio's past values |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.76 WREanalysisPerformanceContrib - Performance contribution

#### Overview

This function implements the performance decomposition of a portfolio.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | number of assets, portfolio size (greater than or equal to 1) |
| weights | double array of dimension nbAssets | portfolio's weight(s) |
| expectedReturns | double array of dimension nbAssets | asset(s) expected return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| performanceContribution | double array of dimension nbAssets | marginal performance contribution |
| performanceContributionP | double array of dimension nbAssets | marginal performance contribution in percent |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.77 WREanalysisRollingPies - Historical Weights (Rolling Pies)

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of values (greater than 1) |
| initialWeight | double | initial weight |
| assetValues | double array of dimension nbValues | asset's values |
| portfolioValues | double array of dimension nbValues | portfolio's values |
| mode | integer | type of values (1) past values, (2) future values |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| weights | double array of dimension nbValues | historical weights |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.78 WREanalysisTrackingErrorContrib - Tracking Error Contribution (Marginal)

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | covariance matrix size (greater than or equal to 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| benchmarkCov | double array of dimension nbAssets | covariance between portfolio and benchmark |
| benchmarkVariance | double | variance of benchmark |
| weights | double array of dimension nbAssets | weight(s) |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| trackingError | double array of dimension 1 | Tracking Error |
| marginalTrackingError | double array of dimension nbAssets | Marginal Tracking Error |
| trackingErrorContribution | double array of dimension nbAssets | marginal Tracking Error contribution |
| trackingErrorContributionP | double array of dimension nbAssets | marginal Tracking Error contribution in percent |
| info | integer array of dimension 1 | diagnostic argument |

### 2.6.79 WREanalysisVolatilityContrib - Volatility Contribution (Marginal)

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | covariance matrix size (greater than or equal to 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| weights | double array of dimension nbAssets | weight(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| volatility | double array of dimension 1 | portfolio volatility |
| marginalVolatility | double array of dimension nbAssets | marginal volatility |
| volatilityContribution | double array of dimension nbAssets | marginal volatility contribution |
| volatilityContributionP | double array of dimension nbAssets | marginal volatility contribution in percent |
| info | integer array of dimension 1 | diagnostic argument |

## 2.7 Data module

### 2.7.1 WREdataAdjBondPrices - Time to maturity adjusted bond returns, for Value-at-Risk calculation

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| nbvalue | integer | number of values |
| histo | double array of dimension nbvalue | historical prices |
| St | double | last price (at now) |
| nominal | double | Nominal |
| deltaT | double | time to maturity |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| adjrets | double array of dimension nbvalue | adjusted historical returns |
| info | integer array of dimension 1 | diagnostic argument |

### 2.7.2 WREdataAdjCBondPrices - Time to maturity adjusted coupon bond returns, for Value-at-Risk calculation

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbvalue | integer | number of values |
| histo | double array of dimension nbvalue | history of prices |
| St | double | last price (at now) |
| nbcoupon | integer | number of coupons |
| coupon | double array of dimension nbcoupon | coupons |
| nominal | double | Nominal |
| deltaT | double array of dimension nbcoupon | time to maturity |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| adjrets | double array of dimension nbvalue | adjusted historical returns |
| info | integer array of dimension 1 | diagnostic argument |

### 2.7.3 WREdataMissing - Fill (for example) monthly data X thanks to a (for example) daily benchmark Y

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| nX | integer | Number of data |
| datesX | integer array of dimension nX | Dates |
| X | double array of dimension nX | Data to fill (prices) |
| nY | integer | Number of data |
| datesY | integer array of dimension nY | Dates |
| Y | double array of dimension nY | Data proxy (prices) |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| nZ | integer array of dimension 1 | Number of data |
| datesZ | integer array of dimension nY | Dates |
| Z | double array of dimension nY | Filled data (prices) |
| info | integer array of dimension 1 | diagnostic argument |

### 2.7.4 WREdataOutlierRet - Univariate outlier detection in return series

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| n | integer | number of value(s) (n ¿= 5) |
| ret | double array of dimension n | return series (one-dimensional) |
| coeflimit | double | descrimination parameter |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| nbliers | integer array of dimension 1 | number of outliers founded |
| liersindex | integer array of dimension n | Indices of outliers founded |
| info | integer array of dimension 1 | diagnostic argument |

### 2.7.5 WREdataOutlierRet2 - Univariate outlier detection in return series (hypothesis of null median)

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| n | integer | number of value(s) (n ¿= 5) |
| ret | double array of dimension n | return series (one-dimensional) |
| coeflimit | double | descrimination parameter |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| nbliers | integer array of dimension 1 | number of outliers founded |
| liersindex | integer array of dimension n | Indices of outliers founded |
| info | integer array of dimension 1 | diagnostic argument |

## 2.8 Modeling module

### 2.8.1 WREmodelingAPT - APT Model (Arbitrage Pricing Theory)

#### Overview

This function implements the APT (Arbitrage Pricing Theory) model.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than or equal to nbFactors+1) |
| nbAssets | integer | number of risky asset(s) (greater than or equal to 1) |
| nbFactors | integer | number of factor(s) (greater than or equal to 1) |
| assetsReturns | double array of dimension nbValues by nbAssets | asset(s) return(s) |
| factorsReturns | double array of dimension nbValues by nbFactors | factor(s) return(s) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| alpha | double array of dimension nbAssets | coefficients alpha |
| beta | double array of dimension nbFactors by nbAssets | coefficients beta |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.2 WREmodelingAPTconstraint - APT Model (Arbitrage Pricing Theory) with alpha/beta constraints

#### Overview

This function implements the APT (Arbitrage Pricing Theory) model with linear constraints on Alpha and Beta.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than or equal to nbFactors+1) |
| nbFactors | integer | number of factor(s) (greater than or equal to 1) |
| assetReturns | double array of dimension nbValues | asset return(s) |
| factorsReturns | double array of dimension nbValues by nbFactors | factor(s) return(s) |
| nbEqConst | integer | number equality constraints (greater than or equal to 0) |
| nbIneqConst | integer | number inequality constraints (greater than or equal to 0) |
| C | double array of dimension nbFactors+1 by nbEqConst+nbIneqConst | matrix of linear constraints |
| b | double array of dimension nbEqConst+nbIneqConst | vector of linear constraints |
| lowerBounds | double array of dimension nbFactors+1 | lower bounds |
| upperBounds | double array of dimension nbFactors+1 | upper bounds |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| alpha | double array of dimension 1 | coefficient alpha |
| beta | double array of dimension nbFactors | coefficients beta |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.3 WREmodelingCalEpsVar - Level of calibration computed by explained variance

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| p | integer | size of the matrix (greater than or equal to 1) |
| Q | double array of dimension p by p | input matrix |
| explainedVariance | double | explained variance (0 lower than explainedVariance lower than 1) |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| minEigenValue | double array of dimension 1 | minimum level of eigenvalue |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.4 WREmodelingCoKurtosis - Co-Kurtosis Matrix

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| nbAssets | integer | number of variables (greater than or equal to 1) |
| assetsReturns | double array of dimension nbValues by nbAssets | returns matrix |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| cokurtosis | double array of dimension nbAssets by nbAssets*nbAssets*nbAssets | co-Kurtosis matrix |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.5 WREmodelingCoSkewness - Co-Skewness Matrix

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| nbAssets | integer | number of variables (greater than or equal to 1) |
| assetsReturns | double array of dimension nbValues by nbAssets | returns matrix |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| coskewness | double array of dimension nbAssets by nbAssets*nbAssets | co-Skewness matrix |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.6 WREmodelingCorr - Correlation Matrix

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| nbAssets | integer | number of variables (greater than or equal to 1) |
| assetsReturns | double array of dimension nbValues by nbAssets | returns matrix |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| corr | double array of dimension nbAssets by nbAssets | correlation matrix |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.7 WREmodelingCorrExp - Exponentially Weighted Correlation Matrix

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| nbAssets | integer | number of variables (greater than or equal to 1) |
| assetsReturns | double array of dimension nbValues by nbAssets | returns matrix |
| lambda | double | exponential parameter (greater than or equal to 0, lower than or equal to1) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| corr | double array of dimension nbAssets by nbAssets | exponentially weighted correlation matrix |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.8 WREmodelingCorrLack - Correlation matrix with missing data (lack of data)

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than or equal to 2) |
| nbAssets | integer | number of variables (greater than or equal to 1) |
| assetsReturns | double array of dimension nbValues by nbAssets | returns matrix (lower than or equal to-1000 if there's a lack) |
| matBackUp | double array of dimension nbAssets by nbAssets | back-up matrix |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| corr | double array of dimension nbAssets by nbAssets | correlation matrix |
| indexMatBackUp | integer array of dimension nbAssets by nbAssets | index used in back-up matrix |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.9 WREmodelingCov - Covariance Matrix

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| nbAssets | integer | number of variables (greater than or equal to 1) |
| assetsReturns | double array of dimension nbValues by nbAssets | returns matrix |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.10 WREmodelingCovAsema - Covariance matrix estimation with ASEMA method

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbDates | integer | number of values |
| nbAssets | integer | number of assets |
| rets | double array of dimension nbDates by nbAssets | historical returns |
| nbPeriods | integer | number of periods |
| tabPeriods | integer array of dimension nbPeriods | lengths of screening periods |
| filteringLevel | double | filtering level for the cov eigenvalues |
| optionMarkov | integer | markovian property activation parameter |
| covPreviousf | double array of dimension nbAssets by nbAssets | reference cov matrix for markov case |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| calibratedCov | double array of dimension nbAssets by nbAssets | calibrated Cov Matrix |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.11 WREmodelingCovExp - Exponentially Weighted Covariance Matrix

#### Overview

Exponentially Weighted Covariance Matrix.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| nbAssets | integer | number of variables (greater than or equal to 1) |
| assetsReturns | double array of dimension nbValues by nbAssets | returns matrix |
| lambda | double | exponential parameter (greater than or equal to 0, lower than or equal to1) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.12 WREmodelingCovExpLack - Exponentially Weighted Covariance Matrix (managing missing data)

#### Overview

Exponentially Weighted Covariance Matrix (managing missing data : values inferior to -1000).

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| nbAssets | integer | number of variables (greater than or equal to 1) |
| assetsReturns | double array of dimension nbValues by nbAssets | returns matrix |
| backupMatrix | double array of dimension nbAssets by nbAssets | backup matrix |
| lambda | double | exponential parameter (greater than or equal to 0, lower than or equal to 1) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| ind | integer array of dimension nbAssets by nbAssets | index matrix |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.13 WREmodelingCovFiltering - Semi-Definite Least Square (SDLS) optimization with filtering level

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| p | integer | size of the matrix (greater than or equal to 1) |
| cov | double array of dimension p by p | input matrix |
| filteringLevel | double | filtering level (greater than or equal to 0 and lower than 1) |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| X | double array of dimension p by p | corrected or calibred matrix |
| parameter | double array of dimension 1 | calibration parameter |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.14 WREmodelingCovFilteringX - Semi-Definite Least Square (SDLS) optimization with filtering level - X

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| p | integer | size of the matrix (greater than or equal to 1) |
| cov | double array of dimension p by p | input matrix |
| filteringLevel | double | filtering level (greater than or equal to 0 and lower than 1) |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| X | double array of dimension p by p | corrected or calibred matrix |
| parameter | double array of dimension 1 | calibration parameter |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.15 WREmodelingCovLack - Empirical covariance matrix with missing data (lack of data)

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than or equal to 2) |
| nbAssets | integer | number of variables (greater than or equal to 1) |
| assetsReturns | double array of dimension nbValues by nbAssets | returns matrix (lower than or equal to-1000 if there's a lack) |
| matBackUp | double array of dimension nbAssets by nbAssets | back-up matrix |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| indexMatBackUp | integer array of dimension nbAssets by nbAssets | index used in back-up matrix |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.16 WREmodelingCovLackAsema - Covariance matrix estimation with ASEMA-lack method (missing data)

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| nbDates | integer | number of values |
| nbAssets | integer | number of assets |
| rets | double array of dimension nbDates by nbAssets | historical returns |
| nbPeriods | integer | number of periods |
| tabPeriods | integer array of dimension nbPeriods | lengths of screening periods |
| filteringLevel | double | filtering level for the cov eigenvalues |
| optionMarkov | integer | markovian property activation parameter |
| covPreviousf | double array of dimension nbAssets by nbAssets | reference cov matrix for markov case |
| matBackup | double array of dimension nbAssets by nbAssets | backup cov matrix. If optionMarkov = 1 then matBackup = covPreviousf |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| calibratedCov | double array of dimension nbAssets by nbAssets | calibrated Cov Matrix |
| indBackup | integer array of dimension nbAssets by nbAssets | indices of used backup elements |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.17 WREmodelingCovLength - Covariance Matrix with different values lengths

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| nbAssets | integer | number of variables (greater than or equal to 1) |
| length | integer array of dimension nbAssets | length of each asset historical returns (greater than 1 and lower than or equal tonbValues) |
| assetsReturns | double array of dimension nbValues by nbAssets | returns matrix |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.18 WREmodelingEM - EM algorithm for missing data

#### Overview

This function implements the Expectation-Maximization (EM) algorithm to recover missing observations in a time series .

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points ( greater than or equal to nbFactors+1) |
| nbAssets | integer | number of assets ( greater than or equal to 1) |
| assetsValues | double array of dimension nbValues by nbAssets | matrix of values |
| missValue | double | value of missing data ( lower than or equal to 0) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| assetsValuesCorrected | double array of dimension nbValues by nbAssets | EM-recover values |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.19 WREmodelingGarchCov - GARCH Covariance Matrix

#### Overview

This function implements the variance-covariance matrix estimation by the multivariate GARCH(1,1) model.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| nbAssets | integer | number of variables (greater than or equal to 1) |
| assetsReturns | double array of dimension nbValues by nbAssets | returns matrix |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.20 WREmodelingImpliedReturns - Implied returns

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| nbAssets | integer | covariance matrix size (greater than or equal to 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| expectedReturns | double array of dimension nbAssets | mean return(s) |
| weights | double array of dimension nbAssets | portfolio's weight(s) |
| riskFreeRate | double | risk free rate (ex. LIBOR 3 months) |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| impliedReturn | double array of dimension nbAssets | implied returns |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.21 WREmodelingLogReturns - Logarithmic return(s)

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of historical points (greater than or equal to 2) |
| nbAssets | integer | number of variables (greater than or equal to 1) |
| assetsValues | double array of dimension nbValues by nbAssets | matrix of values (greater than 0) |
| horizon | integer | investment horizon (0 greater than horizon lower than nbValues) |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| assetsReturns | double array of dimension nbValues-horizon by nbAssets | matrix of returns |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.22 WREmodelingLogReturnsLack - Logarithmic return(s) with missing data (lack of data)

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of historical points (greater than or equal to 2) |
| nbAssets | integer | number of variables (greater than or equal to 1) |
| assetsValues | double array of dimension nbValues by nbAssets | matrix of values (greater than 0 or lower than or equal to-1000 if there's a lack) |
| horizon | integer | investment horizon (0 lower than horizon lower than nbValues) |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| assetsReturns | double array of dimension nbValues-horizon by nbAssets | matrix of returns |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.23 WREmodelingMeanReturn - Mean return calculation

#### Overview

It calculates mean returns.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbDates | integer | Number of dates |
| nbAssets | integer | Number of assets |
| HistoricalReturns | double array of dimension nbDates by nbAssets | Historical returns |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| MeanReturn | double array of dimension nbAssets | Mean returns |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.24 WREmodelingMeanReturnExp - Exponentially weighted mean return calculation

#### Overview

It calculates exponentially weighted mean returns.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbDates | integer | Number of dates |
| nbAssets | integer | Number of assets |
| HistoricalReturns | double array of dimension nbDates by nbAssets | Historical returns |
| lambda | double | Exponentially weighting coefficient |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| MeanReturn | double array of dimension nbAssets | Mean returns |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.25 WREmodelingMeanReturnExpLack - Exponentially weighted mean return calculation (managing missing data)

#### Overview

It calculates exponentially weighted mean returns (managing missing data : values inferior to -1000).

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbDates | integer | Number of dates |
| nbAssets | integer | Number of assets |
| HistoricalReturns | double array of dimension nbDates by nbAssets | Historical returns |
| lambda | double | Exponentially weighting coefficient |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| MeanReturn | double array of dimension nbAssets | Mean returns |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.26 WREmodelingMeanReturnLack - Mean return calculation (manage missing data)

#### Overview

It calculates mean returns, managing missing data (values inferior to -1000).

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbDates | integer | Number of dates |
| nbAssets | integer | Number of assets |
| HistoricalReturns | double array of dimension nbDates by nbAssets | Historical returns |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| MeanReturn | double array of dimension nbAssets | Mean returns |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.27 WREmodelingMedianReturn - Median return calculation

#### Overview

It calculates median returns.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbDates | integer | Number of dates |
| nbAssets | integer | Number of assets |
| HistoricalReturns | double array of dimension nbDates by nbAssets | Historical returns |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| MedianReturn | double array of dimension nbAssets | Median returns |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.28 WREmodelingMedianReturnLack - Median return calculation (managing missing data)

#### Overview

It calculates median returns (managing missing data : values inferior to -1000).

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbDates | integer | Number of dates |
| nbAssets | integer | Number of assets |
| HistoricalReturns | double array of dimension nbDates by nbAssets | Historical returns |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| MedianReturn | double array of dimension nbAssets | Median returns |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.29 WREmodelingMomDGT - Moments of options mono-underlyings.

#### Overview

Moments of options mono-underlyings.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbProducts | integer | number of products in the portfolio |
| V | double array of dimension nbProducts | Values of the products at the current time (m) |
| nbUnderlyings | integer | number of underlyings |
| S | double array of dimension nbUnderlyings | Values of the underlyings at the current time (n) |
| T | double | Time where the VaR will be estimated. |
| nbdates | integer | number of dates |
| ret | double array of dimension nbdates by nbUnderlyings | historical returns |
| expret | double array of dimension nbUnderlyings | expected returns of the underlyings |
| cov | double array of dimension nbUnderlyings by nbUnderlyings | Covariance of the underlyings |
| corres | integer array of dimension nbProducts | Correspondance between underlyings and products (1 to nbUnderlyings) |
| Theta | double array of dimension nbProducts | greek theta |
| Delta | double array of dimension nbProducts | greek delta |
| Gamma | double array of dimension nbProducts | greek gamma |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| expectedreturn | double array of dimension nbProducts | Expected return |
| covariance | double array of dimension nbProducts by nbProducts | covariance matrix |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.30 WREmodelingPASreturns - Expected returns with PAS methodology

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | size 2 of the matrix (greater than or equal to 1) |
| nbAssets | integer | size 1 of the matrix (greater than or equal to 1) |
| histo | double array of dimension nbValues by nbAssets | historical matrix |
| cond | double | conditionning number (greater than or equal to 0 ) |
| nbSimul | integer | number of simulations |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| estimpas | double array of dimension nbAssets | calibration parameter |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.31 WREmodelingPCA - Principal Components Analysis (PCA) using the covariance matrix

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| n | integer | size of the matrix (greater than or equal to 1) |
| M | double array of dimension n by n | covariance matrix |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| eigval | double array of dimension n | eigenvalues (variances) |
| eigvect | double array of dimension n by n | eigenvectors (principal components) |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.32 WREmodelingReturns - Simple Net return(s) (Arithmetic)

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of historical points (greater than or equal to 2) |
| nbAssets | integer | number of variables (greater than or equal to 1) |
| assetsValues | double array of dimension nbValues by nbAssets | matrix of values (greater than 0) |
| horizon | integer | investment horizon (0 greater than horizon lower than nbValues) |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| assetsReturns | double array of dimension nbValues-horizon by nbAssets | matrix of returns |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.33 WREmodelingReturnsLack - Simple Net return(s) (Arithmetic) with missing data (lack of data)

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of historical points (greater than or equal to 2) |
| nbAssets | integer | number of variables (greater than or equal to 1) |
| assetsValues | double array of dimension nbValues by nbAssets | matrix of values (greater than 0 or lower than or equal to -1000 if there's a lack) |
| horizon | integer | investment horizon (0 lower than horizon lower than nbValues) |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| assetsReturns | double array of dimension nbValues-horizon by nbAssets | matrix of returns |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.34 WREmodelingRobustBetas - Robust betas solver

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| p | integer | size of the matrix (¿=1) |
| Q | double array of dimension p by p | quadratic matrix |
| linearPart | double array of dimension p | linear part |
| nbEqConst | integer | number of equality constraints (¿=0) |
| nbIneqConst | integer | number of inequality constraints (¿=0) |
| C | double array of dimension p by nbEqConst+nbIneqConst | matrix of constraints. If no constraints are given, C=0 |
| b | double array of dimension nbEqConst+nbIneqConst | vector of constraints. If no constraints are given, b=0 |
| lowerBounds | double array of dimension p | lower bounds. If some components are bounded from below, set the other (unconstrained) values to a very large negative number |
| upperBounds | double array of dimension p | upper bounds (same remark as above) |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| lagrangeMultipliers | double array of dimension p+nbEqConst+nbIneqConst | Lagrange multipliers |
| optimalPoints | double array of dimension p | optimal points |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.35 WREmodelingSDLS - Semi-Definite Least Square (SDLS) optimization

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| p | integer | size of the matrix (greater than or equal to 1) |
| Q | double array of dimension p by p | input matrix |
| nbEqConst | integer | number of equality constraints (greater than or equal to 0) |
| Ceq | double array of dimension p by nbEqConst*(p) | symmetric matrices of equality constraints |
| bEq | double array of dimension nbEqConst | vector of equality constraints |
| nbIneqConst | integer | number of inequality constraints (greater than or equal to 0) |
| Cineq | double array of dimension p by nbIneqConst*p | symmetric matrices of inequality constraints |
| bLowerIneq | double array of dimension nbIneqConst | vector of lower constraints |
| bUpperIneq | double array of dimension nbIneqConst | vector of upper constraints |
| constPrecision | double | constraints precision (greater than 0) |
| minEigenValue | double | minimum level of eigenvalue desired in the output matrix (greater than or equal to 0) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| X | double array of dimension p by p | corrected matrix |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.36 WREmodelingSDLSModeler - Semi-Definite Least Square optimization with equality and inequality constraints

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| p | integer | size of the matrix (greater than or equal to 1) |
| Q | double array of dimension p by p | input matrix |
| C | integer array of dimension p by p | matrix of equality and inequality constraints indicators. 1 : X(i,j) = Q(i,j). 2 : X(i,j) = b(i,j). 3 : Q(i,j)-b(i,j) lower than X(i,j) lower than Q(i,j)+b(i,j). 4 : bLowerIneq(i,j) lower than X(i,j) lower than b(i,j). |
| b | double array of dimension p by p | matrix of equality and inequality constraints values |
| bLowerIneq | double array of dimension p by p | matrix of lower inequality constraints values |
| constPrecision | double | constraints precision (greater than 0) |
| minEigenValue | double | minimum level of eigenvalue desired in the output matrix (greater than or equal to 0) |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| X | double array of dimension p by p | corrected matrix |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.37 WREmodelingSDLScorr - Semi-Definite Least Square (SDLS) optimization for a correlation matrix

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| p | integer | size of the matrix (greater than or equal to 1) |
| Q | double array of dimension p by p | input matrix |
| constPrecision | double | constraints precision (greater than 0) |
| minEigenValue | double | minimum level of eigenvalue desired in the output matrix (greater than or equal to 0 and lower than 1) |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| X | double array of dimension p by p | corrected correlation matrix |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.38 WREmodelingSDLStrace - Semi-Definite Least Square (SDLS) optimization with trace constraint (constant)

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| p | integer | size of the matrix (greater than or equal to 1) |
| Q | double array of dimension p by p | input matrix |
| constPrecision | double | constraints precision (greater than 0) |
| minEigenValue | double | minimum level of eigenvalue desired in the output matrix (greater than or equal to 0 and lower than 1) |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| X | double array of dimension p by p | corrected matrix |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.39 WREmodelingShrinkCov - Shrinkage estimator of the covariance matrix with Ledoit Wolf's method

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| nbAssets | integer | number of variables (greater than or equal to 2) |
| assetsReturns | double array of dimension nbValues by nbAssets | returns matrix |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| shrinkagecov | double array of dimension nbAssets by nbAssets | covariance matrix |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.40 WREmodelingShrinkMeanBS - Bayes-Stein shrinkage estimator of the expected returns

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| nbAssets | integer | covariance matrix size (greater than or equal to 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| expectedReturns | double array of dimension nbAssets | expected returns |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| shrinkReturn | double array of dimension nbAssets | shrinkage expected returns |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.41 WREmodelingShrinkMeanJS - James-Stein shrinkage estimator of the expected returns

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| nbValues | integer | number of points (greater than 1) |
| nbAssets | integer | covariance matrix size (greater than or equal to 1) |
| cov | double array of dimension nbAssets by nbAssets | covariance matrix |
| expectedReturns | double array of dimension nbAssets | expected returns |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| shrinkReturn | double array of dimension nbAssets | shrinkage expected returns |
| info | integer array of dimension 1 | diagnostic argument |

### 2.8.42 WREmodelingStructuredFiltering - Structured matrix filtering for cross asset class matrices (based on blocks)

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| | integer | matrix size |
| | integer | number of blocks |
| classes | | definition of the classes (classes(i) = j greater than asset i is in the class j) |
| margin | | |
| permissiveness | double | permissiveness for constraints violation |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| info | integer array of dimension 1 | diagnostic argument |

## 2.9 Pricing module

### 2.9.1 WREpricingBC - Pricing a bonus certificate

#### Overview

Pricing a European Barrier Put Up Out.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K | double | strike |
| B | double | Barrier |
| DeltaT | double | Maturity - horizon |
| breached | integer | whether the product is breached (0 if not, 1 else ) |
| ratio | double | ratio |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.2 WREpricingBCCap - Pricing a bonus certificate

#### Overview

Pricing a European Barrier Put Up Out.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K | double | strike |
| Cap | double | Cap |
| B | double | Barrier |
| DeltaT | double | Maturity - horizon |
| breached | integer | whether the product is breached (0 if not, 1 else ) |
| ratio | double | ratio |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.3 WREpricingBarrierCdi - Pricing a European Barrier Call Down In

#### Overview

Pricing a European Barrier Call Down In.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K | double | strike |
| B | double | Barrier |
| Reb | double | Rebate (aka Stop Loss) |
| DeltaT | double | Maturity - horizon |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.4 WREpricingBarrierCdo - Pricing a European Barrier Call Down Out

#### Overview

Pricing a European Barrier Call Down Out.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K | double | strike |
| B | double | Barrier |
| Reb | double | Rebate (aka Stop Loss) |
| DeltaT | double | Maturity - horizon |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.5 WREpricingBarrierCui - Pricing a European Barrier Call Up In

#### Overview

Pricing a European Barrier Call Up In.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K | double | strike |
| B | double | Barrier |
| Reb | double | Rebate (aka Stop Loss) |
| DeltaT | double | Maturity - horizon |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.6 WREpricingBarrierCuo - Pricing a European Barrier Call Up Out

#### Overview

Pricing a European Barrier Call Up Out.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K | double | strike |
| B | double | Barrier |
| Reb | double | Rebate (aka Stop Loss) |
| DeltaT | double | Maturity - horizon |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.7 WREpricingBarrierPdi - Pricing a European Barrier Put Down In

#### Overview

Pricing a European Barrier Put Down In.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K | double | strike |
| B | double | Barrier |
| Reb | double | Rebate (aka Stop Loss) |
| DeltaT | double | Maturity - horizon |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.8 WREpricingBarrierPdo - Pricing a European Barrier Put Down Out

#### Overview

Pricing a European Barrier Put Down Out.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K | double | strike |
| B | double | Barrier |
| Reb | double | Rebate (aka Stop Loss) |
| DeltaT | double | Maturity - horizon |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.9 WREpricingBarrierPui - Pricing a European Barrier Put Up In

#### Overview

Pricing a European Barrier Put Up In.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K | double | strike |
| B | double | Barrier |
| Reb | double | Rebate (aka Stop Loss) |
| DeltaT | double | Maturity - horizon |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.10 WREpricingBarrierPuo - Pricing a European Barrier Put Up Out

#### Overview

Pricing a European Barrier Put Up Out.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K | double | strike |
| B | double | Barrier |
| Reb | double | Rebate (aka Stop Loss) |
| DeltaT | double | Maturity - horizon |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.11 WREpricingDGTPrice - Price a prouct from its greeks

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| n | integer | number of underlyings |
| S0 | double array of dimension n | underlyings at time 0 [n] |
| St | double array of dimension n | underlyings at time t [n] |
| m | integer | number of products |
| P0 | double array of dimension m | products at time 0 [n] |
| t | double | time after time 0 |
| Theta | double array of dimension m | greek theta |
| Delta | double array of dimension n by m | greek delta |
| Gamma | double array of dimension n*n by m | greek gamma |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| Pt | double array of dimension m | products at time 0 [n] |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.12 WREpricingDiscountCertificate - Pricing a discount certificate

#### Overview

It prices a discount-certificate (EUSIPA code : 1200).

#### Inputs

| Variable | Type | Description |
|---|---|---|
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K | double | strike |
| DeltaT | double | Maturity - horizon |
| ratio | double | ratio |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.13 WREpricingMiniFuture - Pricing a mini-future

#### Overview

It prices a mini-future (EUSIPA code : 2210).

#### Inputs

| Variable | Type | Description |
|---|---|---|
| S | double | Undelying price |
| FL | double | Finance level |
| Barrier | double | Barrier |
| longorshort | integer | (1 if long, 0 if short) |
| ratio | double | ratio |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.14 WREpricingOP - Pricing a bonus certificate

#### Overview

Pricing a European Barrier Put Up Out.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K | double | strike |
| Leverage | double | Leverage |
| DeltaT | double | Maturity - horizon |
| ratio | double | ratio |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.15 WREpricingOPBC - Pricing an outperformance bonus certificate

#### Overview

Pricing an outperformance bonus certificate.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K | double | strike |
| B | double | Barrier |
| Leverage | double | Leverage |
| DeltaT | double | Maturity - horizon |
| ratio | double | ratio |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.16 WREpricingOPBCCap - Pricing a capped bonus certificate

#### Overview

It prices a capped bonus certificate (EUSIPA code : 1250).

#### Inputs

| Variable | Type | Description |
|---|---|---|
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K1 | double | strike 1 |
| cap | double | cap |
| B | double | Barrier |
| Leverage | double | Leverage |
| DeltaT | double | Maturity - horizon |
| ratio | double | ratio |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.17 WREpricingOPCap - Pricing a bonus certificate

#### Overview

Pricing a European Barrier Put Up Out.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K1 | double | strike 1 |
| K2 | double | strike 2 |
| Leverage | double | Leverage |
| DeltaT | double | Maturity - horizon |
| ratio | double | ratio |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.18 WREpricingReverseConvertible - Pricing a reverse-convertible

#### Overview

It prices a Reverse-convertible (EUSIPA code : 1220).

#### Inputs

| Variable | Type | Description |
|---|---|---|
| n | integer | Number of cash flows |
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K | double | strike |
| Nominal | double | Nominal |
| Coupon | double array of dimension n | Coupon |
| DeltaT | double array of dimension n | cash flows dates - horizon |
| ratio | double | ratio |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.19 WREpricingVanillaCall - Pricing a European Call

#### Overview

Pricing a European Call.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K | double | strike |
| DeltaT | double | Maturity - horizon |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| gtheta | double array of dimension 1 | greek |
| gdelta | double array of dimension 1 | greek |
| ggamma | double array of dimension 1 | greek |
| grho | double array of dimension 1 | greek |
| gvega | double array of dimension 1 | greek |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.20 WREpricingVanillaCallCap - Pricing a European Call with cap

#### Overview

Pricing a European Call with cap.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K | double | strike |
| Cap | double | Cap |
| DeltaT | double | Maturity - horizon |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.21 WREpricingVanillaPut - Pricing a European Put

#### Overview

Pricing a European Put.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K | double | strike |
| DeltaT | double | Maturity - horizon |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| gtheta | double array of dimension 1 | greek |
| gdelta | double array of dimension 1 | greek |
| ggamma | double array of dimension 1 | greek |
| grho | double array of dimension 1 | greek |
| gvega | double array of dimension 1 | greek |
| info | integer array of dimension 1 | diagnostic argument |

### 2.9.22 WREpricingVanillaPutCap - Pricing a European Put with cap

#### Overview

Pricing a European Put with cap.

#### Inputs

| Variable | Type | Description |
|---|---|---|
| rfr | double | risk free rate |
| d | double | dividend rate |
| sigma | double | volatility |
| S | double | undelying price |
| K | double | strike |
| Cap | double | Cap |
| DeltaT | double | Maturity - horizon |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| P | double array of dimension 1 | Option price |
| info | integer array of dimension 1 | diagnostic argument |

## 2.10 Simulation module

### 2.10.1 WREsimulBMDiffusion - BM diffusion

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| S | double | Initial value |
| mean | double | mean of the (Log) returns |
| vol | double | vol of the (Log) returns |
| nbquantile | integer | number of quantiles |
| alpha | double array of dimension nbquantile | Level of quantile |
| nbstep | integer | number of steps |
| step | double | Length of the step |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| quantileNAV | double array of dimension nbstep by nbquantile | quantiles NAV pathes |
| info | integer array of dimension 1 | diagnostic argument |

### 2.10.2 WREsimulGBMDiffusion - GBM diffusion

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| S | double | Initial value |
| mean | double | mean of the (Log) returns |
| vol | double | vol of the (Log) returns |
| nbquantile | integer | number of quantiles |
| alpha | double array of dimension nbquantile | Level of quantile |
| nbstep | integer | number of steps |
| step | double | Length of the step |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| quantileNAV | double array of dimension nbstep by nbquantile | quantiles NAV pathes |
| meanNAV | double array of dimension nbstep | mean NAV path |
| info | integer array of dimension 1 | diagnostic argument |

### 2.10.3 WREsimulGeometricBrownianX - Simulation of a Multidimentional Geometric Brownian Motion

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| p | integer | number of assets (greater than 1) |
| T | integer | time (days, months, years) (greater than 0) |
| N | integer | number of sub-division(s) (greater than 1) |
| S | double array of dimension p | initial values |
| mu | double array of dimension p | process drifts |
| cov | double array of dimension p by p | covariance matrix |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| y | double array of dimension N by p | generated process |
| info | integer array of dimension 1 | diagnostic argument |

### 2.10.4 WREsimulMonteCarlo - Monte-Carlo method, BASic version, input diffusion parameters

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| userpo | | entry point of an external subroutine payoff provided by the user |
| p | integer | size of problem (greater than or equal to 1) |
| mu | double array of dimension p | drift |
| chol | double array of dimension p by p | Choleski factor of covariance |
| prices | double array of dimension p | initial prices |
| nbsimul | integer | number of simulations |
| maturity | double | maturity in years |
| nbstep | integer | number of steps to maturity in years |
| riskFreeRate | double | risk free rate |
| liudat | integer | size of the integer workspace |
| iusdat | integer array of dimension liudat | user data for USERPO payoff function provided by the user |
| ldudat | integer | size of the double workspace |
| dusdat | double array of dimension ldudat | user data for USERPO payoff function provided by the user |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| payoff | double array of dimension nbsimul | payoff |
| price | double array of dimension 1 | price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.10.5 WREsimulMonteCarloCorr - Monte-Carlo method, Basic version, input CORrelation matrix

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| userpo | | entry point of an external subroutine payoff provided by the user |
| p | integer | size of problem (greater than or equal to 1) |
| corr | double array of dimension p by p | correlation matrix |
| mu | double array of dimension p | drift |
| vol | double array of dimension p | volatilities |
| prices | double array of dimension p | initial prices |
| nbsimul | integer | number of simulations |
| maturity | double | maturity in years |
| nbstep | integer | number of steps to maturity in years |
| riskFreeRate | double | risk free rate |
| liudat | integer | size of the integer workspace |
| iusdat | integer array of dimension liudat | user data for USERPO payoff function provided by the user |
| ldudat | integer | size of the double workspace |
| dusdat | double array of dimension ldudat | user data for USERPO payoff function provided by the user |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| payoff | double array of dimension nbsimul | payoff |
| price | double array of dimension 1 | price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.10.6 WREsimulMonteCarloFact - Monte-Carlo method with factors

**Overview**

**Inputs**

| Variable | Type | Description |
|---|---|---|
| userpo | | entry point of an external subroutine payoff provided by the user |
| n | integer | number of historical points (greater than or equal to 2) |
| p | integer | number of variables (greater than or equal to 1) |
| q | integer | number of factors (greater than or equal to 1) |
| X | double array of dimension n by p | matrix of assets values (greater than 0) |
| Y | double array of dimension n by q | matrix of factors values (greater than 0) |
| h | integer | investment horizon (greater than 0 and lower than n) |
| nbsimul | integer | number of simulations |
| maturity | double | maturity in years |
| nbstep | integer | number of steps to maturity in years |
| riskFreeRate | double | risk free rate |
| muind | integer array of dimension q | indicators for mu, =1: mu computed, =2: mu given |
| givenmu | double array of dimension q | given mu |
| liudat | integer | size of the integer workspace |
| iusdat | integer array of dimension liudat | user data for USERPO payoff function provided by the user |
| ldudat | integer | size of the double workspace |
| dusdat | double array of dimension ldudat | user data for USERPO payoff function provided by the user |

**Outputs**

| Variable | Type | Description |
|---|---|---|
| payoff | double array of dimension nbsimul | payoff |
| fpayoff | double array of dimension q by nbsimul | factors payoff |
| price | double array of dimension 1 | price |
| info | integer array of dimension 1 | diagnostic argument |

### 2.10.7 WREsimulMonteCarloHist - Monte-Carlo method, Basic version, input assets prices HIStory

#### Overview

#### Inputs

| Variable | Type | Description |
|---|---|---|
| userpo | | entry point of an external subroutine payoff provided by the user |
| n | integer | number of historical points (greater than or equal to 2) |
| p | integer | number of variables (greater than or equal to 1) |
| X | double array of dimension n by p | matrix of values (greater than 0) |
| h | integer | investment horizon (greater than 0 and lower than n) |
| nbsimul | integer | number of simulations |
| maturity | double | maturity in years |
| nbstep | integer | number of steps to maturity in years |
| riskFreeRate | double | risk free rate |
| muind | integer array of dimension p | indicators for mu, =1: mu computed, =2: mu given |
| givenmu | double array of dimension p | given mu |
| liudat | integer | size of the integer workspace |
| iusdat | integer array of dimension liudat | user data for USERPO payoff function provided by the user |
| ldudat | integer | size of the double workspace |
| dusdat | double array of dimension ldudat | user data for USERPO payoff function provided by the user |

#### Outputs

| Variable | Type | Description |
|---|---|---|
| payoff | double array of dimension nbsimul | payoff |
| price | double array of dimension 1 | price |
| info | integer array of dimension 1 | diagnostic argument |

# 3 Troubleshooting

This chapter provides the user with instructions on how to access and interpret the error handling contained within the Wall Risk Engine® library. If these suggestions do not help evaluate specific difficulties, we suggest that the user review the "bug report" section (how to report a problem) and contact the Raise Partner support at support@raisepartner.com.

## 3.1 Common errors

We list in this section the most common programming errors when calling Wall Risk Engine® functions. These errors may cause the function to report a failure or they may lead to wrong results:

- wrong number of arguments,
- arguments in the wrong order,
- an argument of the wrong type (especially double and integer arguments of the wrong precision),
- wrong dimension for an array argument,
- bad value for an argument,
- insufficient space.

## 3.2 Bug report

All requests should be sent to the consultant in charge of your project at Raise Partner. The first questions we will ask are the following:

- the client's name and/or reference code,
- the type of machine on which the function was run (e.g. Intel based),
- Operating system (e.g. Windows XP Professional),
- the program language used, e.g. Visual Basic 6.0,

- the name of the Wall Risk Engine® function, e.g. NORMallocIT,
- the value of the function return code (res),
- the value of the diagnostic argument (info),
- if appropriate, a copy of the input file(s) and output file(s).

## 3.3 Error codes

| Code | Description |
|---|---|
| 0 | Successful Termination |
| 1i | Error in the $i$th argument size - computation performed.<br>Ex. code = 102, the second argument has bad size. |
| 2 | Memory allocation problem - no computation performed. |
| 3i | Illegal value(s) of the $i$th argument - no computation performed.<br>Ex. code = 301, the first argument has bad size. |
| 4* | License has expired or bad license - no computation performed.<br>41 or 42: license date expired.<br>43: MAC address not found.<br>44: MAC address access denied. |
| 5 | Numerical problem in computation - no computation performed.<br>For further information see the diagnostic argument *info* <0. |
| 6 | Warning - computation performed (but not optimal).<br>For further information see the diagnostic argument *info* >0. |

### 3.3.1 Exit code *res*

Every function of Wall Risk Engine® returns an integer exit code that should be tested by the calling program. The exit code is to be interpretable as follows:

### 3.3.2 Diagnostic argument *info*

All routines have a diagnostic argument info that indicates the success or failure of the numerical computation (in case the error code is 5 or 6) for each function, as follows:

- info = 0 : successful termination, no problem occurred.
- info = * : computation problem defined for each routine - cf. table hereafter for the diagnostic argument interpretation.
    - If info > 0: Warning - computation performed (but not optimal)
    - If info < 0: Problem in computation - no computation performed

### 3.3.3 Arithmetic errors

| Error Code | Warning | Description |
|---|---|---|
| -1 | | Division by zero - Divisor is zero and dividend is a finite nonzero number (ex. 10.8/0.0). |
| -2 | | Not enough data/value(s) for computation. |
| -3 | | Underflow - Operation produces a result that is too small to be represented as a normal number. |
| -4 | | Overflow - Operation produces a result that exceeds the range of exponent. |
| -5 | | Invalid operand - Operation with mathematically invalid operand. |
| 6 | x | Inexact - Operation produces a result that cannot be represented with finite precision. |

### 3.3.4 Asset allocation errors

| Error Code | Warning | Description |
|---|---|---|
| -100 | | The constraints $C_{inf} <= w <= C_{sup}$ or $C <= b$ are incompatibles. |
| -101 | | The target return is too large (ex. Superior to the maximum of the underlying specified returns). |
| -102 | | The probability confidence level a is not in [0%, 100%]. |
| -103 | | A quote/price value is too small (or zero). |
| -104 | | The variance/volatility is too small (or zero). |
| -105 | | The Tracking Error is too small (or zero). |
| 106 | x | The number of maximum iterations is reached without finding an optimal portfolio. |
| 107 | x | The TE/VaR constraint is too small The portfolio with minimum TE/VaR is computed. |
| -108 | | Covariance matrix must be definite positive (use matrix correction function). |
| -109 | | The input matrix (Risk Budgeting objective function) is not definite positive. |
| 110 | x | A risk budget constraint is too small (the dual solution of this constraint is equal to 1.E+15). |
| 111 | x | A risk budget constraint is to high, maximum number of iterations reached without saturation of this risk budget constraint (the dual solution of this constraint is equal to 0). |
| -112 | | A risk budget Index tracking constraint is negative, should be positive. |
| -113 | | Max. Sharpe ratio unattainable - No problem solution |
| -114 | | The risk-free rate is too large. No problem solution. |

### 3.3.5 Optimization errors

| Error Code | Warning | Description |
|---|---|---|
| 1001 | x | There are no admissible optimal points with these specified constraints. |
| -1002 | | No inferior bound supplied. |
| -1003 | | Degenerate point with infinite cycle. |
| -1004 | | Problem of bound, too large gap between two successive iterations. |
| -1005 | | Incorrect input(s) data. |
| -1006 | | Incompatibility of equality constraints. |
| -1101 | | Error in the simulation/objective function (non-linear optimization). |
| -1102 | | An input parameter is badly initialized (non-linear optimization). |
| -1103 | | Quasi-Newton matrix is no definite positive (non-linear optimization). |
| 1104 | x | Maximum number of iterations reached without finding an admissible point (non-linear optimization). |
| 1105 | x | Maximum number of "simulator" call reached without finding an admissible point. |
| 1106 | x | Maximum precision reached without finding an admissible point (non-linear optimization). |
| -1107 | | Error in the Hessian matrix factorization (non-linear optimization). |
| -1108 | | The problem have no solution (info = 1106 with dxmin < 1.E-15) |
| -1109 | | At least one equality constraint unverified. |
| -1110 | | Problem with constraints, failed to pass checks |
| -1201 | | Problem in the dichotomy function. |
| 1202 | x | Maximum number of iterations reached without finding an admissible point (dichotomy). |
| 5001 | x | Absolute tolerance reached (SOCP). |
| 5002 | x | Relative tolerance reached (SOCP). |
| 5003 | x | Target value achieved (SOCP). |
| 5004 | x | Maximum number of iterations reached (SOCP). |
| 5006 | x | If feasible original problem is unbounded (SOCP). |
| -5001 | | Incorrect input(s) data (SOCP). |
| -5002 | | Error in the initialization of the size of the constraints (must be greater than 0) (SOCP). |
| -5003 | | Dual constraints no verified (SOCP). |
| -5004 | | Primal constraints not respected, return (SOCP). |
| -5005 | | Primal constraints no verified, unfeasible problem (SOCP). |
| -5006 | | The matrix A is not full ranked (SOCP). |
| -5007 | | Problem with lower/upper bounds (SOCP). |
| -5008 | | Target value unachievable (SOCP). |
| -5009 | | The constraint matrix not a symmetric matrix (SOCP). |
| -5010 | | The constraint matrix not a strictly positive matrix (SOCP). |

| | | |
|---|---|---|
| -5011 | | None inverse matrix (SOCP). |
| -5014 | | The lambda parameter is negative (SOCP). |
| -5016 | | The linear primal constraints are not respected (SOCP). |
| -5019 | | Problem with the computation of pseudo-inverse. Constraints badly initialized (SOCP). |
| -5021 | | Problem (kappa > 0) amplitude of the ellipsoid is strictly negative (SOCP). |

### 3.3.6 Matrix calibration errors

| Error Code | Warning | Description |
|---|---|---|
| -1300 | | The input parameter $\alpha$ is too large, the constraint $\alpha \leq n$*Trace(C) is not verified (SDLS with trace constraint). |
| 1301 | | The % of explain variance is too low. |
| 1302 | | The % of explain variance is too large. |
| 1310 | | No correction. The minimum eigenvalue is equal the maximum eigenvalue. |
| -2001 | | The input parameter $\alpha$ is not in [0, 1], X $\geq \alpha$*Id (SDLS for correlation matrix). |
| -2003 | | Specified variance not in [0, 1] (Kato correction). |
| -2004 | | Trace(X) is too small or zero (Kato correction). |

### 3.3.7 Risk factor model errors

| Error Code | Warning | Description |
|---|---|---|
| -4001 | | Bad index value (not equal to 11, 12, 13, 21, 22 or 23). |
| -4002 | | Number of point(s) used in iteration is too small or zero. |
| -4003 | | Incoherent number of tracking factor(s). |
| -4006 | | Numbers of iteration is too small or zero (Tracking Factor). |
| -4008 | | L2 norm is too small or zero (Tracking Factor). |
| 4009 | x | The covariance matrix Kato-structure is completely changing. |

### 3.3.8 Simulation errors

| Error Code | Warning | Description |
|---|---|---|
| -3001 | | The investment horizon is too small (or zero). |
| -3002 | | Initial price(s) is too small (or zero). |
| -3003 | | Number of simulation(s) is too small (or zero). |
| -3101 | | Bad process order (ARCH, GARCH). |
| -3102 | | Condition of process stationary not yields (ARCH, GARCH). |

### 3.3.9 Pricing errors

| Error Code | Warning | Description |
|---|---|---|
| -3201 | | First moment is negative |
| -3202 | | Second moment is negative |
| -3203 | | Underlying price is negative or null |
| -3204 | | Strike is negative or null |
| -3205 | | Time to maturity is negative |
| -3206 | | Barrier is negative or null |
| -3207 | | Initial value is negative (Brownian bridge) |
| -3208 | | Final value is negative or null (Brownian bridge) |
| -3209 | | Nominal is negative or null |
| -3210 | | Capital protection is negative or null |
| -3211 | | Yield to maturity is lower than -1 (formula: rym = ((N-C0)/(CP*N))**(T) -1) |
| -3212 | | Barrier is lower than financial level (ex mini future) |
| -3213 | | Price of the underlying is lower or equal to Barrier (ex mini future) |
| -3214 | | Barrier is greater than or equal to strike |
| -3215 | | Leverage is negative or null |
| -3216 | | Coupon is greater than 1 |
| -3217 | | Number of observation dates has to be >= 1 |
| -3218 | | Negative price got in the pricing process (ex. Exchangeable certificate, call at the emission) |

### 3.3.10 Portfolio simulation errors

| Error Code | Warning | Description |
|---|---|---|
| -3301 | | Threshold1-threshold2 is equal to zero in ConfidScore |
| -3302 | | Sum of underlying weights is null |
| -3303 | | Unrecognized type of product |
| 3304 | x | One or more than one product has experienced an error when pricing |
| -3305 | | Currency index is out of limit (must be <= nbcrncy and >= 1) |
| -3306 | | Time to maturity is negative or null |
| -3307 | | Bond last price St is negative or null |
| -3308 | | Nominal is negative or null (ex. Bonds) |
| -3309 | | One or more of bond historical prices are negative or null |
| -3310 | x | Coefficient of diversification experienced a problem (possibly, correlations calculus) |
| -3311 | | All the instruments have experienced a problem when pricing, no calculation is done |
| 3312 | x | Error occurred while calculating VaR contributions |
| 3313 | x | Error occurred while calculating Performance contributions |
| -3314 | | Error occurred while calculating the VaR |
| -3315 | | Error occurred while calculating the confidence score |
| -3316 | | Error occurred while calculating the portfolio ex-post mean return |
| -3317 | | Error were occurred when adjusting bond returns (division by 0) |
| -3318 | | Product underlyings must be of the same currency (except straight certificate) |
| -3319 | | FX rate return of the product is less or equal than -1 (devision by 0) |
| -3320 | | Simulated FX is negative or null (devision by 0) |
| -3321 | | Product must have underlying currency index (see Ucrncies and Underlyings parameters) |
| -3322 | x | Coefficient of diversification of the equi-weighted Ptf experienced a problem |
| -3323 | | Implicit volatility calculation problem |
| -3324 | | S and/or A are less then its threshold |
| -3325 | | NaN value detected in the outputs (maybe internal error or problem with data) |
| 3326 | x | |
| -3327 | | Market to theo problem |

### 3.3.11 Data management errors

| Error Code | Warning | Description |
|---|---|---|
| -9001 | | All data are missing for a date. |
| -9003 | | No enough observed data. |

### 3.3.12 Data management errors

| Error Code | Warning | Description |
|---|---|---|
| -301 | | Argument 1 has bad value (bad link to the source of data) |

# Copyright