OCTARISK Documentation

version 0.2.0

Stefan Schloegl (schinzilord@octarisk.com)

This is the documentation for the market risk measurement project OCTARISK.

Table of Contents

OCTARISK Documentation

This manual is for the market risk measurement project OCTARISK (version 0.2.0).

1 Introduction

1.1 Why quantifying market risk?

This ongoing project is made for all investors who want to dig deeper into their portfolio than just looking at the yearly profit or loss. Although most financial reports of more sophisticated brokers contain risk measures like standard deviations, the volatility alone cannot cover the risk inherent in non-linear financial products like options. Moreover, potential investors care about their portfolio values under certain market conditions, e.g. they want to compare their perceived personal stress levels during the financial crisis but with the financial instruments in their portfolio losses during stress scenarios.

If questions like

- What happens to my portfolio value, if the ECB increases the interest rate by 100bp?
- What happens to my portfolio value, if an emerging country is defaulting and the US Dollar increases in value?
- How would my portfolio perform, if we have a new crash comparable to Black Friday 1987?

are of potential interest for you, then the OCTARISK market risk project might be satisfying your needs for a professional risk modelling framework.

Since most investors do not give excessive credits to debtors or bear operational or liquidity risks, the OCTARISK project focuses on market risk only - all remaining types of risk which are relevant for your investment portfolio: equity risk, interest rate risk, volatility risk, commodity risk, ...

For the assessment of these market risk types, a sophisticated full valuation approach with Monte-Carlo based value-at-risk and expected shortfall calculation is performed. The underlying principles are state-of-the-art in financial institutions and are used in internal models to fulfill the requirements set by regulators (for Basel III and Solvency 2). The important concepts are adopted, the unnecessary overhead was skipped - resulting in a fast, lightweight yet flexible approach for quantifying market risk.

1.2 Features

The OCTARISK quantifying market risk projects features

- a full valuation approach for financial instruments
- Monte-Carlo method for scenario generation of underlying risk factors
- simple input interface via static and dynamic files
- a processable report incl. graphical representation of portfolio profits and losses as well as an overview over the riskiest instruments and positions (see See Section 3.3 [Output files], page 20 for examples)
- an easily customizable framework based on full implementation on Octave with compact source code

1.3 Prerequisites

The only requirement is GNU Octave (tested for versions > 4.0) with installed financial and statistical package and hardware with minimum of 5Gb of memory. Calculation time decreases significantly while using optimized linear algebra packages of OpenBLAS and LAPACK (or comparable). For automatic processing of the input data (e.g. to get actual market data from Quandl or Yahoo finance), to make the parameter estimation as well as process the report files, some programming language like Perl, Python and a running LaTeX environment are recommended, but not required.

Nevertheless, a basic understanding of a high level programming language like Octave is required to adjust the source code and to customize the calculation. Furthermore, a thorough understanding of financial markets, instruments and valuation will be needed in order to select appropriate models and to interpret results. The implemented models and the underlying concepts are explained in detail for example in following literature:

Risk Management and Financial Institutions, John C. Hull, 2015
Paul Wilmott on Quantitative Finance, 2nd Edition, Paul Wilmott, 2006
Options, Futures and other Derivatives, 7th Edition, John C. Hull, 2008

The next chapter contains the background and details needed for running the market risk valuation and aggregation software and understanding the risk measures.

2 User guide

This chapter gives a general introduction to market risk and how the market risk is captured by OCTARISK. Thereby the following convention is made: market movement means the movement around the mean value or, in other words, the possible deviation from the expected value as time passes. Stronger movement around the expected value means more risk and results in a higher standard deviation, the most important statistic parameter for capturing market risk.

2.1 Introducing market risk

Typically, market risk can be divided into several sub types. Most of the splitting is obvious, but some of the more exotic risk types remain very broad. The ideal breakdown depends heavily on the specific portfolio whose market risk shall be quantified. A basic approach fitting most portfolios of private investors, not too broad, not too granular, is chosen in OCTARISK. The following types of risk are defined:

- *FX*: Forex risk captures the market movements of exchange rate values. These movements relative to the reporting currency (in our case EUR) affect financial assets in foreign currencies only.
- *EQ*: Equity risk is related to the movement of equity markets. It will be typically broken down into sub-categories like countries, regions or other types of aggregation (e.g. developed market, emerging market, frontier markets).
- *COM*: Commodity risk. This is a rather broad type of risk, often correlated to other risk types (e.g. equity). Commodity risk tries to capture the movement of spot and future / forward commodity prices, as well as commodity linked equities.
- *IR*: Interest rate risk is related to the movement of interest rate values. This is the most important type of risk for cash flow bearing instruments.
- *RE*: Real estate risk, where typically it will be distinguished between movements of market values of REITs (Real Estate Investment Trusts) and housing prices. This risk type can also be broken down into sub-categories like different countries, regions or other categories (e.g. developed markets, emerging markets).
- *VOLA*: Implied volatility risk, e.g. the anticipated future movement of implied volatility of equity instruments or swaps.
- *ALT*: Alternative market risk, used as a container for every other type of risk not already captured (e.g. Bitcoins, infrastructure)

The OCTARISK projects focuses on these seven risk types. For each risk type, appropriate risk factors can be chosen. One risk factor is a typical representative of a particular risk type. Most often, several risk factors are needed to describe the granular behavior of a market risk type, e.g. it is necessary to describe the specific characteristics of countries, regions or currencies. For example, two risk factors can be used to describe the international equity market movements: Developed markets and emerging market. If desired, developed market can be further split into North America, Europe and Asia to describe risk diversification effects between these broad regions. After the selection of risk factors, stochastic models are chosen, calibrated and subsequently used for scenario generation.

2.2 Market risk measures

2.2.1 Value-at-Risk

Value-at-risk (VAR) is defined as the monetary loss which the portfolio won't exceed for a specific probability on a certain time horizon. As an example, a 250 trading day (one calender year) VAR of 1000 EUR at the 99% confidence interval means there is a probability of 99% that the portfolio loss within one year (250 trading days) is equal to or less than 1000 EUR. It is important to note that no forecast is made for the possible loss which can occur in 1% of the remaining cases. Furthermore, within a proper calibrated risk setup one **must** expect a loss greater than the VAR amount in 1% of all cases, that means a loss of more than 1000 EUR will occur in two to three trading days per year. Otherwise, if there are no trading days observed where the loss is greater than predicted by VAR, the risk is overstated, leaving room for better usage of risk capital.

2.2.2 Expected shortfall (ES)

Expected shortfall is an additional risk measure which is defined as the arithmetic average (mean) loss in the remaining tail of the sorted profit and loss distribution of all simulated MC scenarios, which are not covered by the 99% VAR. The ES should always be seen in context of the VAR and is a more coherent risk measure, which can make stronger predictions about diversification benefits of portfolios.

2.3 Scenario generation

A scenario is a specific set of shocks to risk factors. Typically, the directions of the shocks are correlated, and the value of the shock is dependent on the stochastic properties of the risk factors (e.g. volatility or mean reversion parameters). Although these properties can be also chosen customary, for evaluating risk measures like value-at-risk these parameters are typically extracted from past, real market movements.

2.3.1 Stochastic models

In order to describe movements of risk factors in time, a connection has to be made between statistical behavior of real time series and stochastic processes for modeling synthetic time series. OCTARISK concentrates on three stochastic processes: Wiener, Ornstein-Uhlenbeck and root-diffusion processes.

2.3.1.1 Random walk and the Wiener process

For the Wiener process, two different possible definitions exist. In the first case, both the drift and the normally-distributed random number W_t (the so called Wiener process) are proportional to the variable at the former time step, resulting in the process S_t which satisfies the following stochastic differential equation:

$dS_t = mu * S_{t-1} * dt + sigma * S_t * dW_t$
The following analytic solution to this stochastic differential equation is derived:

$S_t = S_0 * exp((mu - sigma^2/2) * t + sigma * W_t)$
This solution ensures positive values at all time steps.

In the second case, a continuous time random walk with independent, normally-distributed random numbers independent of the variable at the former time step is given by the following stochastic differential equation:
$dS_t = mu * dt + sigma * W_t$
This process shows self-similarity and scaling behavior.

2.3.1.2 Ornstein-Uhlenbeck process

The Wiener process can be extended to incorporate a serial dependency (like a memory) - tomorrows values are dependent on the level of todays values. The Ornstein-Uhlenbeck (OU) process has a mean-reversion term, which is directly proportional to the difference of the actual value from the mean reversion level:
$dX_t = mr_{rate} * (mr_{level} - X_{t-1}) * dt + sigma * dW_t$
where the mean reversion rate mr_{rate} can be seen as a proportional parameter of a restoring force. The increments dX_t tend to point to the mean-reversion level mr_{level}. The result of the formula is an addditive term to the risk factor depending on the past level and a stochastic term (modeled by the Wiener process).

2.3.1.3 Square-root diffusion process

In order to exclude negative values in the OU mean-reversion process, an additional repelling force is needed, which ensures that the level of the stochastic variables stays away from zero. The square-root diffusion process (SRD) has an additional term, which is multiplied with the standard deviation and the random variable. This term is identified as the square root of the variable at the former time step:
$dX_t = mr_{rate} * (mr_{level} - X_{t-1}) * dt + sigma * sqrt(X_{t-1}) * dW_t$
Negative values for the variables are excluded if the following equation is fulfilled:
$2 * mr_{rate} * mr_{level} >= sigma$

2.3.2 Financial models

The stochastic models which have been presented in the last section, are used as basis for financial models. In order to map stochastic processes to financial models, properties of financial models are identified and subsequently stochastic models chosen in order to generate simulated time series of risk factors with appropriate and desired behavior.

2.3.2.1 Geometric Brownian motion

The standard financial model for equity, real-estate and commodity risk factors is a geometric Brownian motion (GBM), which utilizes the extended Wiener process, where the drift and random variable are proportional to the risk factor value. Due to this proportionality the time series can not reach zero and the modeled risk factors values always stay positive. This is reflected in the real world behavior of equity or commodity prices, where the intrinsic value of these assets cannot fall below zero.

2.3.2.2 Brownian motion

In a Brownian motion (BM) model, the time-series increments are a function of drift, time and standard deviation, but not dependent on the actual level of the financial variable. For certain types of risk factors (e.g. interest rates), one assumes a Brownian motion so that the modeled price movements are given as additive shocks which are completely independent on the actual risk factor value. Therefore, negative values of the modeled risk factors are allowed.

2.3.2.3 Vasicek model

In the long run, the market movements of exchange rates and interest rates seem to be mean-reverting. This behavior can be modeled by a Ornstein-Uhlenbeck process resulting in a so called one factor short rate model proposed by Vasicek. The Ornstein-Uhlenbeck process allows for negative values, which reflects the real world behavior of short rates since the financial crisis, where interest rates of AAA-rated government bonds had negative yields for at east some time.

2.3.2.4 Cox-Ingersoll-Ross and Heston model

If one doesn't want to allow negative values for interest rates or other mean-reverting risk factors, a square-root diffusion process can be chosen as stochastic model. This results in the short-rate model of Cox-Ingersoll-Ross or the Heaston model for modelling at-the-money volatility used in option pricing.

2.3.3 Parameter estimation

Once an appropriate model is chosen for the risk factor, one has to define input parameters for the stochastic differential equations. One approach is to use historical data to estimate statistic parameters like volatility, correlations or mean-reversion parameters. Another approach is to apply expert judgment in selecting input parameters for the models. OCTARISK uses the specified parameters to generate Monte-Carlo scenarios - the selection of the parameter estimation approach is up to the reader.

Typically, parameters are extracted from historical time series on weekly or monthly data on the past three to five years. Unfortunately, availability of historical time series for all risk factors is one of the main constraints in parameter estimation for private investors. Most often, one has to overcome problems of missing data and the need for interpolation or extrapolation. In future versions scripts for parameter estimation will be provided.
In an ideal world, at first appropriate risk types and risk factors are selected, then the validation of a stochastic model is performed and the appropriateness of the model and the assumptions (like length of historical time series) is verified in back-tests. Nevertheless, no stochastic model can completely describe the financial markets, giving rise to model error (both in parameter estimation and model selection).

2.3.4 Monte-Carlo Simulation

A Monte-Carlo approach is chosen to generate the risk factor shocks in all scenarios. Therefore a risk factor correlation matrix (e.g. estimated from historical time series) and additional parameters describing statistical distributions can be used to generate random numbers, which are utilized as input parameters to stochastic models.

In order to account for non-normal distributions and higher order correlation effects in the Monte-Carlo simulation, a copula approach is chosen to generate dependent, correlated random numbers. In a first step, the input correlation matrix is used to generate normally distributed, correlated random numbers with zero drift and unit variance. In a next step, either a Gaussian copula or a student-t copula is utilized to transform the normally distributed random variables to uniformly distributed random variables, while either the linear correlation dependence (for Gaussian copulas) or additionally the non-linear dependence structures (for student-t copulas) is preserved. In a last step, these random numbers are incorporated into a function which chooses for each risk factor the appropriate distribution in a certain way, that standard deviation, skewness and kurtosis are matched with the input parameters. Therefore the Pearson distribution system is used as a basis for generation of random variables. Subsequently, in each of the Monte-Carlo scenarios a specific set of correlated risk factor shocks, dependent on the selected financial models, is generated, while the marginal distributions have desired standard deviation, skewness and kurtosis.

A further potential problem is the model error from the Monte-Carlo method. Only a limited number of Monte-Carlo scenarios can be generated and valuated (in the range of 10000 to 100000), thus leaving space for not represented scenarios which could alter the risk measures.

2.3.5 Stress testing

A complementary method to the stochastic scenario generation is to directly define shocks to risk factors. This scenario analysis is normally done in order to calculate the portfolio behavior in well known historic scenarios (like Financial Crisis 2008, devaluation of Asian currencies in the mid 90s, terrorist attack on 9/11, Black Friday in October 1987) or in scenarios, where one only is interested in the behavior of the portfolio value in isolated shocks (like all equities decline by 30pct in value, a +100bp parallel shift in interest rates). Possible sources of scenarios are provided by regulators and can be easily adapted for personalized stress scenarios.

2.4 Instrument valuation

After the scenarios are generated, one has to calculate the behavior of financial instruments to movements in the underlying risk factors. Therefore two different approaches are chosen: the sensitivity approach applies relative valuation adjustments to the instrument base values proportional to the defined sensitivity. The valuation adjustments are derived from movements in value of the underlying risk factors. Secondly, in a full valuation approach, the scenario dependent input parameters are fed into a pricing function which (re)calculates the new absolute value of the instrument in each particular scenario.

2.4.1 Sensitivity approach

The sensitivity approach is performed for all instruments which have a linear dependency on the movement of underlying risk factor values, or where not enough data or no appropriate pricing function exists for a full valuation approach.

2.4.1.1 Linear dependency on risk factor shocks

For the risk assessment of equities, commodities or real estate instruments it is appropriate to take only the linear dependency on risk factor movements into account. Each risk factor has sensitivities to underlying risk factors. An instrument inherits the amount of shock proportional to the defined sensitivity value from the underlying risk factors.

A typical example is a developed market exchanged traded fund (ETF) with exposure to North America, Europe and Asia. The ETF will follow the price movements of these three risk factors, so the sensitivities are simply the relative exposure to the three equity markets (e.g. 0.5, 0.3 and 0.2). These sensitivities are then used to calculate the weighted shock that is applied to the actual instrument value. The same principle holds for single stock, where sensitivities to appropriate risk factors and to an idiosyncratic risk term (an uncorrelated random number) can be selected. A useful method for calibration is the multi-dimensional linear regression. The resulting betas from this regression can be taken for the sensitivities to regressed risk factors. The remaining alpha and estimation error resembles the sensitivity to the idiosyncratic risk. The exposure to the uncorrelated random number can be derived from one minus the adjusted R_square of the regression. Since the R_square gives the amount of variability that is explained by the regression model, one minus R_square is equal to the amount of uncorrelated random fluctuations which are not covered by the input parameters.

2.4.1.2 Approximation with sensitivity approach

For instruments with insufficient information one can also choose the sensitivity approach. One example are funds consisting mainly of bonds. Without look-through, one has no information about the exact cash flows of the underlying bonds. Instead, often the duration (and convexity) of the fund is known. These two types of sensitivity (duration and convexity) can then be used to calculate the change in value to interest rate shocks. Therefore, the absolute interest rate shock at the node, which equals the Macaulay duration, is calculated as absolute difference compared to the base rate. This change in interest rate level (dIR) is directly transformed to a relative value shock (dV) by the formula
$dV = duration * dIR + convexity * dIR^2$
incorporating both sensitivities.

2.4.2 Full valuation approach

The core competency of a quantitative risk measurement project is full valuation, where the absolute value of financial instruments is calculated from raw input parameters by a special pricing function. Some input parameters to the pricing function are scenario dependent, other are inherent to the instrument. The most important input parameters have to be modeled by stochastic processes and can subsequently be fed into the pricing function, where the new, scenario dependent absolute value of the instrument is calculated. At the moment, the following full valuation pricing functions are implemented in OCTARISK:

2.4.2.1 Option pricing

European plain-vanilla options are priced by the Black-Scholes model (See Section 4.2 [option_bs], page 24). The Black-Scholes equation provides a best estimate of the option price. The underlying financial instrument and implied volatility are modeled as risk factors in order to calculate the new option price in each scenario. The risk free rate will be also made scenario dependent in order to capture the interest rate sensitivity of the option price. Further information is provided by numerous textbooks.

For American options a more sophisticated model has to be used for pricing. Unfortunately, binomial models (like the Cox-Rubinstein-Ross model) or finite-difference models are not feasible for a full valuation Monte-Carlo based approach, since the computation time for a large amount of time steps and MC scenarios is too high due to missing parallelization opportunities. Instead, a Willow-Tree model is implemented to price American options. Within that model, instead of using the full binomial tree with increasing number of nodes per time step, a constant number of nodes at each pricing time step is utilized to approximate the movements of the underlying price. With optimized transition probabilities, the whole model relies on a smaller amount of total nodes which significantly decreases computation time and lowers memory consumption (See Section 4.3 [option_willowtree], page 25 implementation for further details).

A calibration is performed to align the model price based on provided input parameters with the observed market price. This calibration calculates an implied spread which is added to the modeled volatility as a constant offset.

The implied volatility itself is dependent on the option strike level and time to maturity (term). In order to grasp that behavior, the so called volatility smile is modeled by a moneyness vs. term volatility surface, where changes in the spot price lead to moneyness changes. Therefore, the actual implied volatility behavior (at-the-money implied volatility vs. moneyness vs. term) of the market is preserved for the pricing.

2.4.2.2 Swaption pricing

European plain-vanilla swaptions are priced via the closed form solution of the Black-76 model. Again, a calibration is performed to align the model price based on provided input parameters with the observed market price. The calibrated implied volatility spread is subsequently added to the modeled volatility as a constant offset. The volatility smile for the specific term is also given by volatility surfaces.

2.4.2.3 Cash flow instrument pricing

Cash flow instruments are specified by the following sets of variables: cash flow dates and corresponding cash flow values. Moreover, each cash flow instrument has an actual market price and an underlying interest rate curve, which has to be provided as a separate risk factor. Before the full valuation can be carried out, the spread over yield is calculated to align the observed market price with the value given by the pricing function. The spread over yield is then assumed to be a constant offset to the scenario dependent interest rate spot curve. For each scenario, all cash flows are discounted with the appropriate interest rate. The present value is then given by the sum of all discounted cash flows. In the future, credit spreads will also be modeled as separate risk factors, thus incorporating another risk factor which captures the risk of cash flow instruments.

2.5 Aggregation

After the valuation of all instruments in each scenario, one has to aggregate all parameters of instruments contained in a fund. Therefore all fund position values are derived by multiplying the position size with the instrument value in each scenario. The resulting sorted fund profit and loss distribution is then used to calculate the value-at-risk at fund level.

If less than 50001 MC scenarios are used in OCTARISK, the VAR is smoothened by a Harrell-Davis estimator (See Section 4.6 [harrell_davis_weight], page 27 for details). A weighted average of the scenarios around the confidence interval scenarios is calculated. The HD VAR shall reduce the Monte-Carlo error.

2.6 Reporting

After the aggregation of instrument data to fund level, a risk report is generated. The report contains VAR on position level, the riskiest instruments and positions per fund, as well as the diversified (with observed correlation) and undiversified (correlation set to 1) VAR and ES on fund level. Moreover, stress test results per fund, profit and loss distributions and histograms on both the one day and 250 day time horizon are generated (See Section 3.3 [Output files], page 20 for examples)

3 Developer guide

This chapter describes the actual implementation of the project. All calculation steps and the input and output files will be described in detail as well as examples are provided.

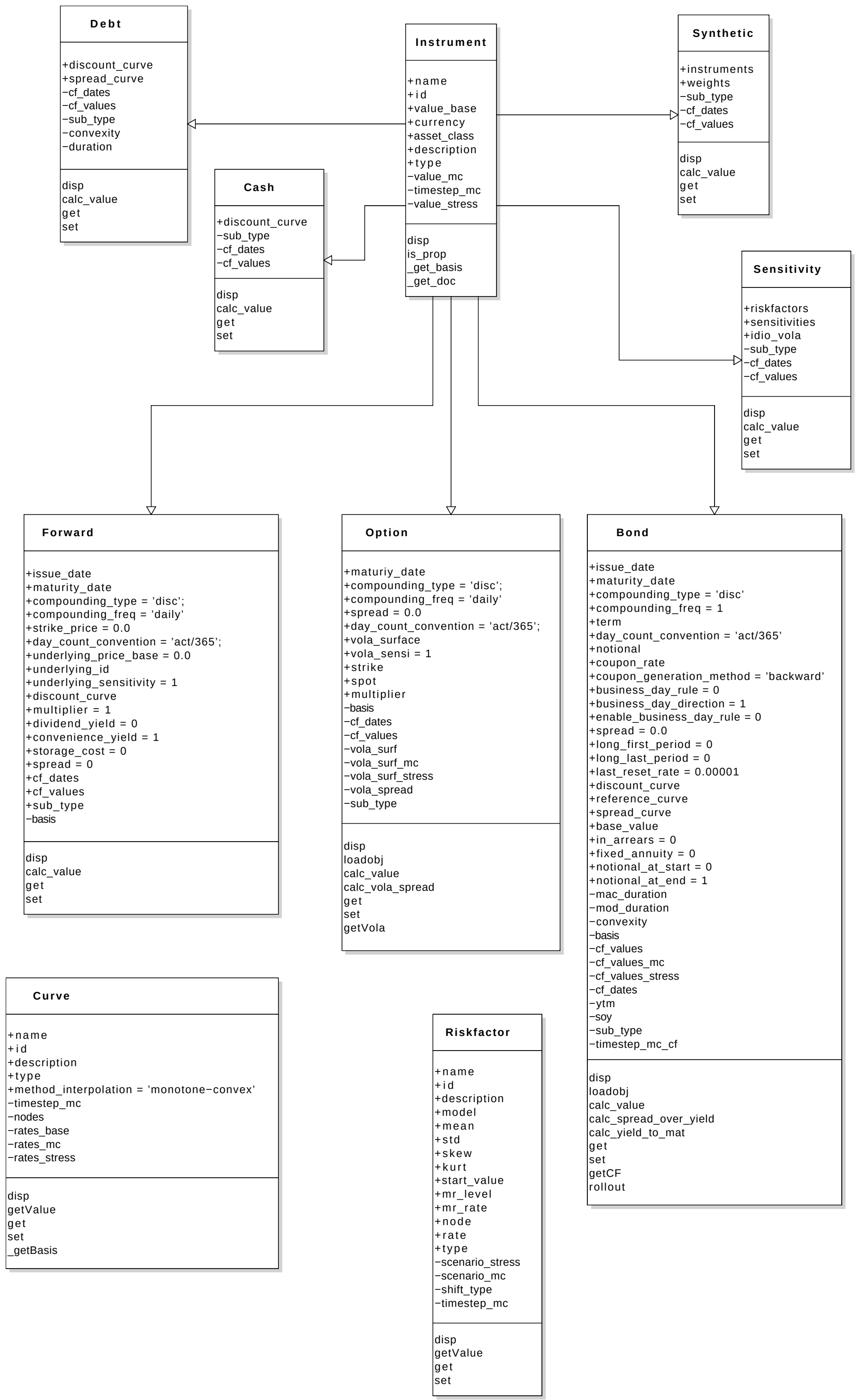
3.1 Implementation concept

Fast, lightweight, accurate are the three main buzz words for the implementation concept. The following enumeration gives an overview of the script:

1. DEFINITION OF VARIABLES
   1. general variables
   2. VAR specific variables
2. INPUT
   1. Processing Instruments data
   2. Processing Riskfactor data
   3. Processing Positions data
   4. Processing Stresstest data
3. CALCULATION
   1. Model Riskfactor Scenario Generation
      - Load input correlation matrix
      - Get distribution parameters from riskfactors
      - call MC scenario generations
   2. Monte Carlo Riskfactor Simulation for all timesteps
   3. Take risk factor stress scenarios from stressdefinition
   4. Process yield curves
   5. Full Valuation of all Instruments in all MC and stress scenarios
   6. Portfolio Aggregation
      - loop over all portfolios / positions
      - VaR Calculation
        - sort arrays
        - Get Value of confidence scenario
        - make vector with Harrel-Davis Weights
      - Calculate Expected Shortfall
   7. Print Report including position VaRs
   8. Plotting
4. HELPER FUNCTIONS

3.1.1 Class diagram

OCTARISK was set up in an object oriented programming style for all objects like instruments, risk factors and curves. Inside the methods of the aforementioned classes, pricing or interpolation functions are called. The following class diagram gives an overview of all classes:


Debt
+discount_curve
+spread_curve
−cf_dates
−cf_values
−sub_type
−convexity
−duration
disp
calc_value
get
set
Instrument
+name
+id
+value_base
+currency
+asset_class
+description
+type
−value_mc
−timestep_mc
−value_stress
disp
is_prop
_get_basis
_get_doc
Synthetic
+instruments
+weights
−sub_type
−cf_dates
−cf_values
disp
calc_value
get
set
Cash
+discount_curve
−sub_type
−cf_dates
−cf_values
disp
calc_value
get
set
Sensitivity
+riskfactors
+sensitivities
+idio_vola
−sub_type
−cf_dates
−cf_values
disp
calc_value
get
set
Forward
+issue_date
+maturity_date
+compounding_type = 'disc';
+compounding_freq = 'daily'
+strike_price = 0.0
+day_count_convention = 'act/365';
+underlying_price_base = 0.0
+underlying_id
+underlying_sensitivity = 1
+discount_curve
+multiplier = 1
+dividend_yield = 0
+convenience_yield = 1
+storage_cost = 0
+spread = 0
+cf_dates
+cf_values
+sub_type
−basis
disp
calc_value
get
set
Option
+maturiy_date
+compounding_type = 'disc';
+compounding_freq = 'daily'
+spread = 0.0
+day_count_convention = 'act/365';
+vola_surface
+vola_sensi = 1
+strike
+spot
+multiplier
−basis
−cf_dates
−cf_values
−vola_surf
−vola_surf_mc
−vola_surf_stress
−vola_spread
−sub_type
disp
loadobj
calc_value
calc_vola_spread
get
set
getVola
Bond
+issue_date
+maturity_date
+compounding_type = 'disc'
+compounding_freq = 1
+term
+day_count_convention = 'act/365'
+notional
+coupon_rate
+coupon_generation_method = 'backward'
+business_day_rule = 0
+business_day_direction = 1
+enable_business_day_rule = 0
+spread = 0.0
+long_first_period = 0
+long_last_period = 0
+last_reset_rate = 0.00001
+discount_curve
+reference_curve
+spread_curve
+base_value
+in_arrears = 0
+fixed_annuity = 0
+notional_at_start = 0
+notional_at_end = 1
−mac_duration
−mod_duration
−convexity
−basis
−cf_values
−cf_values_mc
−cf_values_stress
−cf_dates
−ytm
−soy
−sub_type
−timestep_mc_cf
disp
loadobj
calc_value
calc_spread_over_yield
calc_yield_to_mat
get
set
getCF
rollout
Curve
+name
+id
+description
+type
+method_interpolation = 'monotone−convex'
−timestep_mc
−nodes
−rates_base
−rates_mc
−rates_stress
disp
getValue
get
set
_getBasis
Riskfactor
+name
+id
+description
+model
+mean
+std
+skew
+kurt
+start_value
+mr_level
+mr_rate
+node
+rate
+type
−scenario_stress
−scenario_mc
−shift_type
−timestep_mc
disp
getValue
get
set

3.2 Input files

In the following, all required input files are introduced in detail. An example portfolio is given with position of equity ETFs, global bond ETFs, short positions in equity options and exposures to commodity risk, real estate risk and Bitcoins.

3.2.1 Risk factors

The risk factors input file contains all risk factors which are modeled by stochastic processes. The shocks of these risk factors are then used as in put to the calculation of the new instrument value which have these risk factors as underlying risk drivers.
The columns of the risk factors file consist of the following entries:

- *name*: Name of the riskfactors, this follows the convention RF_TYPE_XYZ (string).
- *id*: Unique ID of the risk factors. To keep it simple, just take name (string).
- *type*: Risk factor types follow typical asset class conventions (string). These types are explained in Section 2.1 [Introducing market risk], page 4.
- *description*: A short description of the risk factor. String maximum length of 255 characters.
- *model*: Model ID of the underlying stochastic process (string). See section Models for further explanation.
- *parameters*: Comma separated parameters used for modeling the stochastic process (floats).

Generalized column format for risk factors input file with colums field separator ';':

name	**ID**	**type**	**description**	**model**	**parameters**
RF_XX_YY	RF_XX_YY	RF_XX	useful description	GBM, BM, OU, SRD	x, y, ..., z

The input format is restricted to these six columns. However, to define various number of columns for the different stochastic models, each of the colums can itself be used as a vector of arbitrary length with field separator ','. The items of the vectors contain model specific information:

model	**para1**	**para2**	**para3**	**para4**	**para5**	**para6**	**para7**
GBM, BM	mean	std dev	skewness	kurtosis	IR node	IR value	
OU	mean	std dev	skewness	kurtosis	start value	mr-level	mr-rate
SRD	mean	std dev	skewness	kurtosis	start value	mr-level	mr-rate

For stochastic models GMB and GM the parameters 5 and 6 are used only for definition of interest rate risk factors. The following is an example definition for some risk factors:

```
name;id;type;description;model;parameters;
RF_EQ_DE;RF_EQ_DE;RF_EQ;Equity Germany;GBM;0.0,0.18,-0.5,5.0;
RF_EQ_EUR;RF_EQ_EUR;RF_EQ;Equity Euro;GBM;0.0,0.18,-0.5,5.0;
RF_EQ_EU;RF_EQ_EU;RF_EQ;Equity Europe;GBM;0.0,0.144,-0.5,5.0;
RF_EQ_NA;RF_EQ_NA;RF_EQ;Equity NorthAmerica;GBM;0.0,0.16,-0.5,4.0;
RF_EQ_AP;RF_EQ_AP;RF_EQ;Equity AsiaPacific;GBM;0.0,0.16,-0.5,4.0;
RF_EQ_EM;RF_EQ_EM;RF_EQ;Equity EmergingMarkets;GBM;0.0,0.20,-0.5,4.0;
RF_COM_DIV;RF_COM_DIV;RF_COM;Commodity Diversified;GBM;0.0,0.13,-0.5,4.0;
RF_RE_DM;RF_RE_DM;RF_RE;RealEstate World DM;GBM;0.0,0.15,-0.5,5.5;
RF_COM_GOLD;RF_COM_GOLD;RF_COM;Physical Gold;GBM;0.0,0.12,-0.3,6.5;
RF_COM_SILVER;RF_COM_SILVER;RF_COM;Physical Silver;GBM;0.0,0.20,-0.3,6.5;
RF_ALT_BC;RF_ALT_BC;RF_ALT;Bitcin EUR;GBM;0.0,0.5,-0.3,4.5;
RF_FX_EURUSD;RF_FX_EURUSD;RF_FX; EUR USD;SRD;0.0,0.08,0.0,3.0,1.09,1.2,0.001;
RF_VOLA_DE;RF_VOLA_DE;RF_VOLA;Vola Germany;SRD;0.0,0.1,0.0,3.0,0.21,0.19,0.027;
RF_IR_EUR_1Y;RF_IR_EUR_1Y;RF_IR;IR EUR 1y;BM;0.0,0.0011,0.0,3.0,365,0.001;
RF_IR_EUR_5Y;RF_IR_EUR_5Y;RF_IR;IR EUR 10y;BM;0.0,0.0032,0.0,3.0,3650,0.002;
RF_IR_EUR_20Y;RF_IR_EUR_20Y;RF_IR;IR EUR 20y;BM;0.0,0.0060,0.0,3.0,7300,0.02;
RF_IR_USD_1Y;RF_IR_USD_1Y;RF_IR;IR USD 1y;BM;0.0,0.0023,0.0,3.0,365,0.0066;
RF_IR_USD_5Y;RF_IR_USD_5Y;RF_IR;IR USD 5y;BM;0.0,0.006,0.0,3.0,1825,0.017;
RF_IR_USD_10Y;RF_IR_USD_10Y;RF_IR;IR USD 10y;BM;0.0,0.006,0.0,3.0,3650,0.024;
RF_VOLA_GOLD;RF_VOLA_GOLD;RF_VOLA;Vola Gold;SRD;0.0,0.1,0.0,3.0,0.15,0.17,0.04;
```

3.2.2 Positions

The positions input file contains all positions, which are essentially a bunch of instruments in a portfolio. Positions must point to instruments which are defined in the instruments input file. Portfolios containing these instruments are just increasing integers.
The columns of the positions contain the following characteristics :

- *port*: ID of the portfolio (integer). No further information of the portfolio is needed.
- *id*: Unique ID of the instrument in the portfolio. Must reference a valid instrument which is defined in the instruments input file.
- *quantity*: The number of shares of the particular instrument in the portfolio (float).

Example definitions for some positions (positive quantity: long position, negative quantity: short position):

```
port;id;quantity;
1;A0RFFT;65;
1;A1JB4Q;179;
1;A1J7CK;135;
1;A0RM43;141;
1;ETF127;543;
1;DBX1A7;24;
1;513700;44;
1;DBX1EU;33.35;
1;ETF060;107;
1;ETF114;159;
1;LYX0BX;129.8;
1;LYX0AG;20.18;
1;ETF012;71;
1;A0DPMW;63.52;
1;BASF11;39;
1;A0LGQL;723;
1;ETF090;53;
1;GOLDPHYS;10.31;
1;CFANLEIHE1;10;
1;BTCOIN;1.98;
1;ODAXC201501217;-1;
```

3.2.3 Covariance matrix

The covariance file is an Octave readable full matrix containing the covariances of all risk factors. Therefore the dimension of the covariance matrix is n x n, where n must be the number of defined risk factors (otherwise an exception will be raised). The covariance matrix is then split into a correlation matrix and a standard deviation matrix. After that, the correlation matrix undergoes a Cholesky decomposition to generate correlated random variables. The standard deviation vector is no longer used for modelling the stochastic processes, since the four moments of the distributions are directly given in the risk factor input file. The correlation matrix must be positive definite, otherwise an error will be raised.

3.2.4 Instruments

The instruments input file contains the definition of all instruments which are used as portfolio positions and are referencing underlying risk factors.
The columns of the positions file consist of following entries:

- *name*: Name of the instrument which will be used for the reports (string).
- *id*: Unique ID of the instrument. Just take WKN or ISIN (string). Used as a reference in position input file.
- *value*: Actual market value of the instrument. Relative shocks of risk factors will be applied to this market value. In case of absolut shocks (for e.g. Options of cash flow instruments), a calibration of the spread over yield or volatility spread will be performed (float).
- *type*: Instrument type should follow the same convention as applied to risk factor types. Shock values from stress tests will be applied to all instruments of one type (string).
- *description*: A short description of the instrument. Maximum length of 255 characters (string).

- *currency*: Currency of the instrument. If no appropriate currency risk factor is defined, they are mapped to USD (string).
- *riskfactors*: Comma separated IDs of underlying risk factors. Vector must be equal in length to sensitivities (string).
- *sensitivity*: Comma separated values of sensitivities to the formerly defined risk factors. Vector must be equal in length to risk factors (float).
- *special_num*: Special field for definition of numbers, e.g. used for volatility or idiosyncratic risk of stocks (float).
- *special_str*: Special field for definition of strings, no actual usage (string).
- *cf_dates*: Comma separated strings of cash flow dates, used to describe cash flow instruments (e.g. 16-Jul-2016,17-Jul-2017) (string).
- *cf_values*: Comma separated cash flow amounts in the instrument currency, one amount per cash flow date. Vector must be of equal length to cash flow dates (float).

The following table gives an overview of the instrument type dependent comma separated fields:

type	**item**	**values**
Option	riskfactors	RF impl. vol., RF underlying, strike, RF IR rate
Option	sensitivity	beta to RF impl. vol., underlying spot value, strike value, IR spread
Option	special_num	multiplier value
Option	special_str	expiration date (format DD-MMM-YYYY)
CF instr.	riskfactors	RF IR rate
CF instr.	sensitivity	IR spread added to RF IR curve in absolute terms
CF instr.	special_num	null
CF instr.	special_str	null
CF instr.	cf_dates	comma separated strings of cash flow dates
CF instr.	cf_values	comma separated floats of cash flow amounts
Swaption	riskfactors	RF impl. vol., RF underlying, strike, RF IR rate, swap tenor, swap no. payments per year
Swaption	sensitivity	beta to RF impl. vol., underlying spot value, strike value, IR spread, tenor value, payments amount
Swaption	special_num	multiplier value
Swaption	special_str	expiration date (format DD-MMM-YYYY)

Example definitions for some instruments:

```
name;id;value;type;description;currency;riskfactors;sensitivity;special_num;special_str;cf_date
...
```

Instrument description for Debt instruments: Riskfactor is underlying discount curve (spread to be implemented) and sensitivity is duration.
Instrument description for Equity / Real Estate / Commodity instruments: Riskfactors are several underlying riskfactors, sensitivities are like betas in multi-dimensional linear regression approach.
Speciality for Stocks: Riskfactors [Real Riskfactors;IDIO]. IDIO means normal distributed random variable with standard deviation of underlying riskfactor and with sensitivity equals [1 - R^2] of a linear regression to cover idiosyncratic risk component of stocks.

3.2.5 Stress tests

The stress test input file contains the definition of all stress test. Each stress test describes the behavior of one or more risk factor in a particular scenario. The risk factor shock values are directly applied to all risk factor which follow the same name convention.
The columns of the stress test file consists of following entries:

- *number*: Unique number (integer) of the particular stresstest.
- *id*: Unique ID (string) of the stresstest.
- *name*: Name (string) of the stresstest, used in reporting.
- *risktype*: Comma separated IDs of risk types OR risk factors. If only a risktype (e.g. RF_FX) is given, all risk factors with that risk type are shocks in the stresstest scenario. If a whole risk factor ID is given (e.g. RF_FX_EURUSD or RF_IR_EUR_5Y), only the single risk factors are shocked (string).
- *shiftvalue*: The value of the applied shocks. Relative shocks are given in decimals (e.g. -0.5 for a -50% down shock). Absolute shocks to interest rate nodes are given in basis points (e.g. 100 means a 100bp upshock) (float).
- *shifttype*: Integer 0 or 1 defining the shock type. 1 means relative shock (multiplied by risk factor base value), 0 means absolute shock (added to risk factor base value).

An example for two possible stress test definitions are given. The financial crisis stress test shocks all equity positions, lowers the USD spot rate curve by 200 bp and increases the implied equity volatility by 150%:

```
1;FinancialCrisis2008;Finl Cri 2008;RF_IR,RF_EQ,RF_VOLA; -200,-0.4,1.5;0,1,1;
2;IR_USDTwistPos;IR +- Twist;RF_IR_USD_1Y,RF_IR_USD_10Y;-100,100;0,0;
```

3.2.6 Volatility surface

For all options and swaptions, the implied volatility is necessary to calculate the derivative theoretical value. In order to feed the implied volatility into octarisk, a moneyness / term surface has to be specified in a separate file. For all underlying risk factors the impl. volatility data file has to be named like "vol_index_RF_XX_YY.dat". The risk factor ID will be used to automatically identify the appropriate file. The structure of the file follows the volatility surface conventions: the first row and column contain the days to maturities and the moneynesses, e.g.

```
0 30 93
1.2 0.326 0.274
1.0 0.319 0.283
0.8 0.289 0.274
```

This example describes options with two different days to maturity (30 days and 93 days) and three different moneynesses (1.2, 1.0, 0.8). During the full valuation approach the moneyness is a function of the underlying risk factor spot price / rate and the constant strike price / rate. TA linear two-dimensional interpolation and constant extrapolation will be performed to calculate the new scenario dependent implied volatililty. Since the at-the-money volatility is itself a risk factor (modeled as a factor), the interpolated volatility will be multiplied by this scenario dependent factor. This process combines the conservation of volatility surface shape with the single factor stochastic modeling of the at-the-money volatility. Furthermore, it is possible to generate a file with just one constant volatility.

3.3 Output files

Overview of report summary and graphics.

3.3.1 VAR Report

For each fund, a VAR report is generated. The report shall give an overview of total risk measure, allow a break down to position level and estimate the diversification effect. An example for the report looks like:

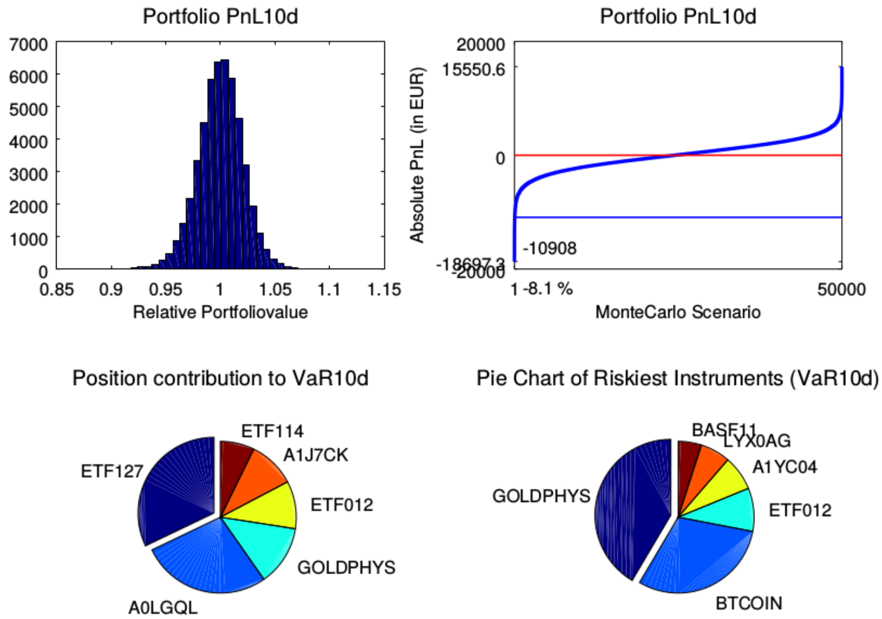
```
=== Value-At-Risk Report for Portfolio 1 ===
VaR calculated 99pct Confidence Intervall:
Number of Monte Carlo Scenarios: 500000
Confidence Scenarionummer: 5000
Valuation Date: 09-Oct-2015
VaR on Positional Level:
|VaR 1D for Position   |iShares JPMorgan $ EM|A0RFFT| = | 618.16 EUR|
|VaR 250D for Position |iShares JPMorgan $ EM|A0RFFT| = | 1755.21 EUR|
...
=== Total Portfolio VaR ===
|Portfolio VaR   1D@99Pct|   |    -1.32%|
|Portfolio VaR   1D@99Pct|   |  1779.97 EUR|
|Portfolio VaR 250D@99Pct|   |   -19.34%|
|Portfolio VaR 250D@99Pct|   | 25983.52 EUR|

|Expected Shortfall 1D@99Pct|   |    -1.51%|
|Expected Shortfall 1D@99Pct|   |  2027.10 EUR|
|Expected Shortfall 250D@99Pct|   |   -21.58%|
|Expected Shortfall 250D@99Pct|   | 28996.72 EUR|
```

The unique field separator '|' allows efficient use of the data in LaTeX based report files.

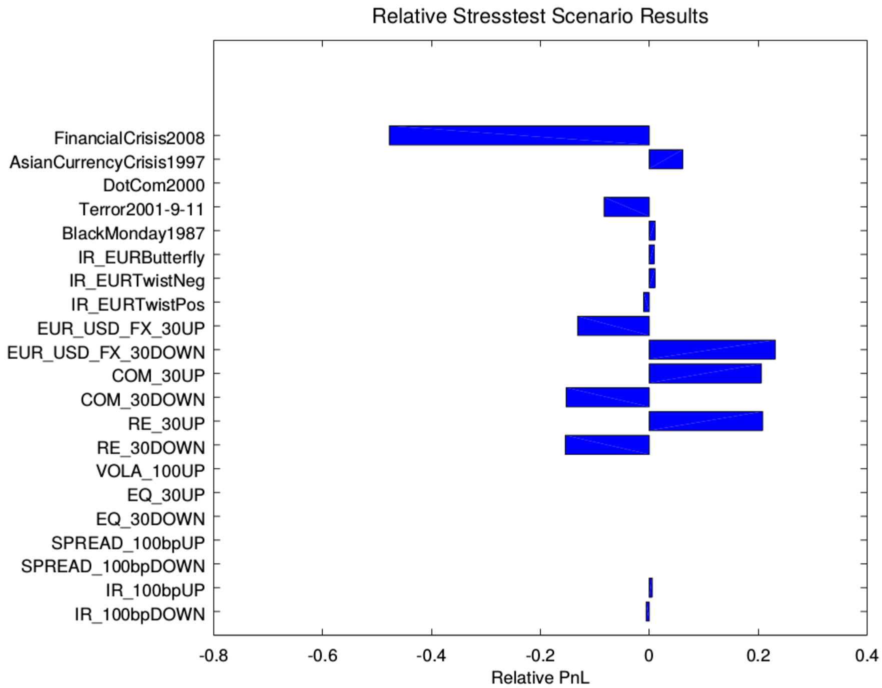
3.3.2 Overview images

Additionally to the printed report, overview images of the profit and loss distribution, both sorted and histogram as well as the riskiest instruments and positions are plotted.



Portfolio Value-at-Risk at the 99.9% confidence level on 10 day time horizon. There are shown four different results: On the top left corner a histogram of the profit and loss distribution is presented for the whole portfolio. On the x-axis, the relative portfolio value is given. On the top right corner the P'n'L distribution is shown as sorted absolute profits or losses for each of the MC scenarios. The red base line marks the base scenario. The blue line indicates the VAR scenario number, where 99.9% of all profits and losses are shown on the right side. Losses greater than the VAR occur in 0.1% of all cases. The two lower charts show the VAR contribution of the six riskiest positions (left chart) or instruments (right chart). The positional view is determined by weighing the instruments with their position size in the portfolio.

Moreover, stress test results are plotted in a bar chart, indicating the relative profit or loss of the fund in each stress test scenario:


Relative Stresstest Scenario Results
FinancialCrisis2008
AsianCurrencyCrisis1997
DotCom2000
Terror2001-9-11
BlackMonday1987
IR_EURButterfly
IR_EURTwistNeg
IR_EURTwistPos
EUR_USD_FX_30UP
EUR_USD_FX_30DOWN
COM_30UP
COM_30DOWN
RE_30UP
RE_30DOWN
VOLA_100UP
EQ_30UP
EQ_30DOWN
SPREAD_100bpUP
SPREAD_100bpDOWN
IR_100bpUP
IR_100bpDOWN
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
Relative PnL

4 Octave Functions and Scripts

In the following sections you find the Octave function texinfo.

4.1 octarisk

`octarisk ()` [Function File]

Version: 0.1, 2015/11/24, Schinzilord: initial version
0.2, 2015/12/16, Schinzilord: added Willow Tree model for pricing american equity options, added volatility surface model for term / moneyness structure of volatility
0.3, 2016/01/19, Schinzilord: added new instrument types FRB, FRN, FAB, ZCB added synthetic instruments (linear combinations of other instruments) added equity forwards and Black-Karasinski stochastic process 0.4, 2015/02/05, Schinzilord: added spread risk factors, general cash flow pricing

Calculate Monte-Carlo Value-at-Risk (VAR) and Expected Shortfall (ES) for instruments, positions and portfolios at a given confidence level on 1D and 250D time horizon with a full valuation approach.
See octarisk documentation for further information.

Input files in csv format:

- Instruments data: specification of instrument universe (name, id, market value, underlying risk factor, cash flows etc.)
- Riskfactors data: specification of risk factors (name, id, stochastic model, statistic parameters)
- Positions data: specification of portfolio and position data (portfolio id, instrument id, position size)
- Stresstest data: specification of stresstest risk factor shocks (stresstest name, risk factor shock values and types)
- Covariance matrix: covariance matrix of all risk factors
- Volatility surfaces (index volatility: term vs. moneyness, call moneyness spot / strike, linear interpolation and constant extrapolation)

Output data:

- portfolio report: instruments and position VAR and ES, diversification effects
- profit and loss distributions: plot of profit and loss histogramm and distribution, most important positions and instruments

Supported instrument types:

- equity (stocks and funds priced via multi-factor model and idiosyncratic risk)
- commodity (physical and funds priced via multi-factor model and idiosyncratic risk)
- real estate (stocks and funds priced via multi-factor model and idiosyncratic risk)
- custom cash flow instruments (NPV of all custom CFs)

- bond funds priced via duration-based sensitivity approach
- fixed rate bonds (NPV of all CFs)
- floating rate notes (scenario dependent cash flow values, NPV of all CFs)
- fixed amortizing bonds (either annuity bonds or amortizable bonds, NPV of all CFs)
- zero coupon bonds (NPV of notional)
- European equity options (Black-Scholes model)
- American equity options (Willow Tree model)
- European swaptions (Black76 model)
- Equity forward
- Synthetic instruments (linear combinations of other valuated instruments)

Supported stochastic processes for risk factors:

- Geometric Brownian Motion
- Black-Karasinski process
- Brownian Motion
- Ornstein-Uhlenbeck process
- Square-root diffusion process

Supported copulas for MC scenario generation:

- Gaussian copula
- t-copula with one parameter specification for common degrees of freedom

Further functionality will be implemented in the future (e.g. inflation linked instruments)

See also: option_willowtree, option_bs, harrell_davis_weight, swaption_black76, pricing_forward, rollout_cashflows, scenario_generation_MC.

4.2 option_bs

`[value delta gamma vega theta rho omega] = option_bs (CallPutFlag, S, X, T, r, sigma, divrate)` [Function File]

Compute the prices of european call or put options according to Black-Scholes valuation formula:

```
C(S,T) = N(d_1)*S - N(d_2)*X*exp(-rT)
P(S,T) = N(-d_2)*X*exp(-rT) - N(-d_1)*S
d1 = (log(S/X) + (r + 0.5*sigma^2)*T)/(sigma*sqrt(T))
d2 = d1 - sigma*sqrt(T)
```

The Greeks are also computed (delta, gamma, vega, theta, rho, omega) by their closed form solution.

Parallel computation for column vectors of S,X,r and sigma is possible.

Variables:

- *CallPutFlag*: Call: "1", Put: "0"
- *S*: stock price at time 0
- *X*: strike price
- *T*: time to maturity in days
- *r*: annual risk-free interest rate (continuously compounded)
- *sigma*: implied volatility of the stock price measured as annual standard deviation
- *divrate*: dividend rate p.a., continously compounded

See also: option_willowtree, crr_binomialoption, option_exotic_mc.

4.3 option_willowtree

value = option_willowtree (*CallPutFlag*, *AmericanFlag*, *S*, *X*, *T*, *r*, *sigma*, *dividend*, *dk*) [Function File]

value = option_willowtree (*CallPutFlag*, *AmericanFlag*, *S*, *X*, *T*, *r*, *sigma*, *dividend*, *dk*, *nodes*) [Function File]

Computes the price of european or american equity options according to the willow tree model.

The willow tree approach provides a fast and accurate way of calculating option prices. Furthermore, massive parallelization due to litte memory consumption is possible. This implementation of the willow tree concept is based on following literature:

- "Willow Tree", Andy C.T. Ho, Master thesis, May 2000
- "Willow Power: Optimizing Derivative Pricing Trees", Michael Curran, ALGO RESEARCH QUARTERLY, Vol. 4, No. 4, December 2001

Efficient parallel computation for column vectors of S,X,r and sigma is possible (advantage: linear increase of calculation time in timesteps and nodes).

Runtime of parallel computations incl. tree transition optimization (360 days maturity, 5 day stepsize, 20 willow tree nodes) are performed (at 46 GFlops machine, 4 Gb Ram) in:

50 | 0.5s
500 | 0.5s
5000 | 1.1s
50000 | 9.0s
200000 | 32s

Example of an American Call Option with continuous dividends:
(365 days to maturity, vector with different spot prices and volatilities, strike = 8, r = 0.06, dividend = 0.05, timestep 5 days, 20 nodes): `option_willowtree(1,1,[7;8;9;7;8;9],8,365,0.06,[0.2;0.2;0.2;0.3;0.3;0.3],0.05,5,20)`

Variables:

- *CallPutFlag*: Call: "1", Put: "0"
- *AmericanFlag*: American option: "1", European Option: "0"
- *S*: stock price at time 0
- *X*: strike price
- *T*: time in days to maturity

- *r*: annual risk-free interest rate (continuously compounded, act/365)
- *sigma*: implied volatility of the stock price measured as annual standard deviation
- *dividend*: continuous dividend yield, act/365
- *dk*: size of timesteps for valuation points (optimal accuracy vs. runtime choice : 5 days timestep)
- *nodes*: number of nodes for willow tree setup. Number of nodes must be in list [10,15,20,30,40,50]. These vectors are optimized by Currans suggested Method to fulfill variance constraint (optimal accuracy vs. runtime choice: 20 nodes)

See also: option_binomial, option_bs, option_exotic_mc.

4.4 interpolate_curve

`interpolate_curve (`*`nodes`*`, `*`rates`*`, `*`timestep`*`)` [Function File]
`interpolate_curve (`*`nodes`*`, `*`rates`*`, `*`timestep`*`, `*`ufr`*`, `*`alpha`*`)` [Function File]

Calculate an interpolated return on a curve for a given timestep Supported methods are: linear (default), moneymarket, exponential, loglinear, spline and smith-wilson. A constant extrapolation is assumed, excep for smith-wilson, where the ultimate forward rate will be reached with reversion speed alpha For all methods except splines a fast taylormade algorithm is used. For splines see function interp1 for more details. Explanation of Input Parameters of the linear interpolation curve function:

- *nodes*: is a 1xN vector with all timesteps of the given curve
- *rates*: is MxN matrix with curve rates per timestep defined in columns. Each row contains a specific scenario with different curve structure
- *timestep*: is a scalar, specifiying the interpolated timestep on vector nodes
- *ufr*: (only used for smith-wilson): ultimate forward rate
- *alpha*: (only used for smith-wilson): reversion speed to ultimate forward rate

See also: interp1, interp2, interp3, interpn.

4.5 pricing_npv

`[`*`npv MacDur`*` ] = pricing_npv(`*`valuation_date,`* [Function File]
`cashflow_dates`, *`cashflow_values`*,*`spread_constant`* ...
,*discount_nodes*, *discount_rates*, *spread_nodes*, *spread_rates*, *basis*, *comp_type*, *comp_freq*)

Computes the net present value and Maccaulay Duration of a given cash flow pattern according to a given discount curve, spread curve and day count convention etc. Pre-requirements:

- installed octave finance package
- custom functions timefactor, discount_factor, interpolate_curve

Input and output variables:

- *valuation_date*: Structure with relevant information for specification of the forward:

- *cashflow_dates*: cashflow_dates is a 1xN vector with all timesteps of the cash flow pattern
- *cashflow_values*: cashflow_values is a MxN matrix with cash flow pattern.
- *spread_constant*: a constant spread added to the total yield extracted from discount curve and spread curve (can be used to spread over yield)
- *discount_nodes*: tmp_nodes is a 1xN vector with all timesteps of the given curve
- *discount_rates*: tmp_rates is a MxN matrix with discount curve rates defined in columns. Each row contains a specific scenario with different curve structure
- *spread_nodes*: OPTIONAL: spread_nodes is a 1xN vector with all timesteps of the given spread curve
- *spread_rates*: OPTIONAL: spread_rates is a MxN matrix with spread curve rates defined in columns. Each row contains a specific scenario with different curve structure
- *basis*: OPTIONAL: day-count convention (either basis number between 1 and 11, or specified as string (act/365 etc.)
- *comp_type*: OPTIONAL: compounding type (disc, cont, simple)
- *comp_freq*: OPTIONAL: compounding frequency (1,2,3,4,6,12 payments per year)
- *npv*: returs a 1xN vector with all net present values per scenario
- *MacDur*: returs a 1xN vector with all Maccaulay durations

See also: timefactor, discount_factor, interpolate_curve.

4.6 harrell_davis_weight

`harrell_davis_weight` (*scenarios*, *observation*, *alpha*) [Function File]

Computes the Harrell-Davis (1982) quantile estimator and jacknife standard errors of quantiles. The quantile estimator is a weighted linear combination or order statistics in which the order statistics used in traditional nonparametric quantile estimators are given the greatest weight. In small samples the H-D estimator is more efficient than traditional ones, and the two methods are asymptotically equivalent. The H-D estimator is the limit of a bootstrap average as the number of bootstrap resamples becomes infinitely large.

See also: ..., ...

4.7 doc_instrument

object = *Instrument* (*name*,*id*,*description*,*type*,*currency*,*base_value*,*asset_class*,*valuation_date*) [Function File]

Instrument Superclass Inputs:

- *name* (string): Name of object

- *id* (string): Id of object
- *description* (string): Description of object
- *type* (string): instrument type in list [cash, bond, debt, forward, option, sensitivity, synthetic]
- *currency* (string): ISO code of currency
- *base_value* (float): Actual base (spot) value of object
- *asset_class* (sring): Instrument asset class
- *valuation_date* (datenum): serial day number from Jan 1, 0000 defined as day 1.

The constructor of the instrument class constructs an object with the following properties and inherits them to all sub classes:

- name: Name of object
- id: Id of object
- description: Description of object
- value_base: Actual base (spot) value of object
- currency: ISO code of currency
- asset_class: Instrument asset class
- type: Type of Instrument class (Bond,Forward,...)
- valuation_date: date format DD-MMM-YYYY
- value_stress: Vector with values under stress scenarios
- value_mc: Matrix with values under MC scenarios (values per timestep per column)
- timestep_mc: MC timestep per column (cell string)

Superclass Method getValue
Return the scenario (shock) value for an instrument object. Specify the desired return values with a property parameter. If the second argument abs is set, the absolut scenario value is calculated from scenario shocks and the risk factor start value.
Timestep properties:

- base: return base value
- stress: return stress values
- 1d: return MC timestep

Instrument Method isProp
Query all properties from the Instrument Superclass and sub classes and returns 1 in case of a valid property.

See also: Instrument.

Index