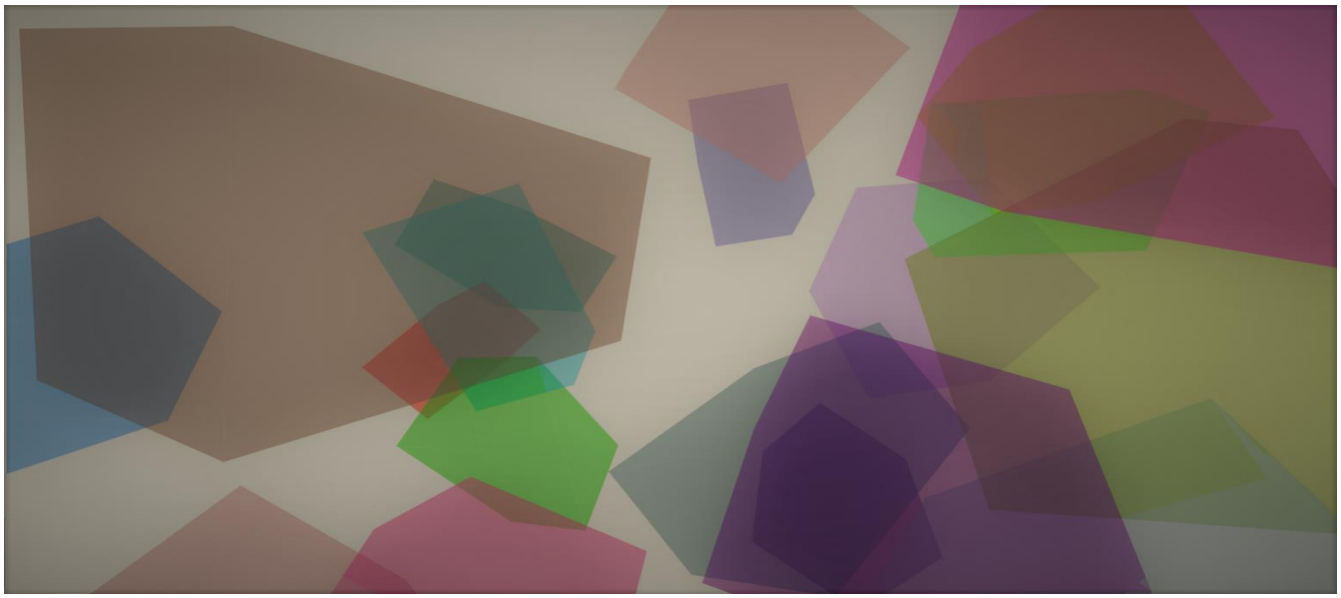


MATHEMATICAL MODELLING METHODS FOR TIME SERIES

Eurorisk Systems Ltd.
31, General Kiselov str.
BG-9002 Varna, Bulgaria
Phone +359 52 612 367
Fax +359 52 612 371
info@eurorisksystems.com
www.eurorisksystems.com

Contents

1. Preface

This report outlines important aspects of models for time series processing methods. It aims at proposing a set of advanced approaches for the represenation and modelling of time series, that can be used for pricing, sensitivity calculation and simulation of Value at Risk, on short and long term horizons. These approaches should ensure easy integration into the evaluation and management systems, to cover functionalities from front to back office. Moreover, the intention behind them is to develop approaches for multi-factor and multi-step simulation on several future time points, to produce distributions based on simulation paths, and to allow pricing and risk estimation for instruments that work with underlying time series.

In addition to the model definition, development tasks include the creation of prototype evaluation, as well as analysis of studies, thereby identifying the functional content and results of the evaluation and simulation. The development is based on historical price time series (price index per unit) and current prices of futures (forwards) of time series.

The evaluation approaches include the following main steps:

- Representation of expected future developments of time series, using prediction based learning or neural networks, as well as auto-regressive logic.

- Representation of implied or historical volatility of the series, according to maturity.

- 2D performance (Multi-Factor and Multi-Step) of the Monte Carlo simulation on selected market factors and future time points, in order to obtain development paths and distributions.

- Running of path-dependent pricing expressions that produce mainly non-normal distributions and obtain means as expectations and VaR statistics.

A set of additional approaches should enhance the quality of the pricing and VaR estimation, including:

- Stochastic interpolation (Volatility Bridge), used to produce daily price movements for a future time period.

- Non-normal distributions (including automatic identification of the distribution type and its parameters) of market time series and copula approach for market factors in the Monte Carlo simulation.

- Multifactor models that represent price development of unknown factors by known market factors. This dependence is automatically calibrated on a pricing expression that can then be used for the calculation of prices, sensitivities and VaR simulations.

- Clustering of time series, used for grouping time series in such a way that series with similar historical behaviour are categorized into the same group. This approach reduces the analysis and simulation space and is able to make orthogonal removing dependencies.

For further information, please contact our Hotline at +359 52 612 367 or send an e-mail to info@eurorisksystems.com.

2. Clustering of time series

2.1 Purpose

The purpose of clustering is creating groups of time series in a way that series with similar historical behaviour are organized into the same group.

2.2 Input data

- A set of time series.
- Number of clusters.

2.3 Output data

- Clusters of time series. Every cluster consists of:
  - The series belonging to the cluster.
  - A generated synthetic series, called prototype series, considered a representative of the cluster. This synthetic series has the same dimensionality as all other series and is usually near the cluster's centre.

- In addition to clusters – together with their series and prototypes – clustering quality statistics are generated [21] [22] [23] in the following manner:
  - Inter and intra cluster statistics, adjuster R squared, average linkage, etc. These elements represent the clustering quality.
  - Some of these statistics can be used to determine the optimal number of clusters.

The clustering module is able to reduce large sets of thousands of individual time series to small sets of several dozens synthetic series [21] [22]. The main reason for this is the using of prototypes instead of real series for time and memory consuming operations, such as correlations estimation, with as minimal error as possible. In this way, the number of time series is reduced, as any calculation should be done with a given time series, it is identified in which series the cluster has been classified and the cluster prototype is used instead of the real series. Thus, the number of all available series is reduced to the number of clusters. That way, calculations that needed to be performed with the series decrease and can be performed practically. Otherwise, the huge data causes too many calculations, which often cannot be finished ¸within an acceptable time period. Some examples of clusters, along with their series and prototypes, are shown in Fig. 1. The red bold lines represent prototypes that are near clusters centres and substitute the real series in time consuming calculations.

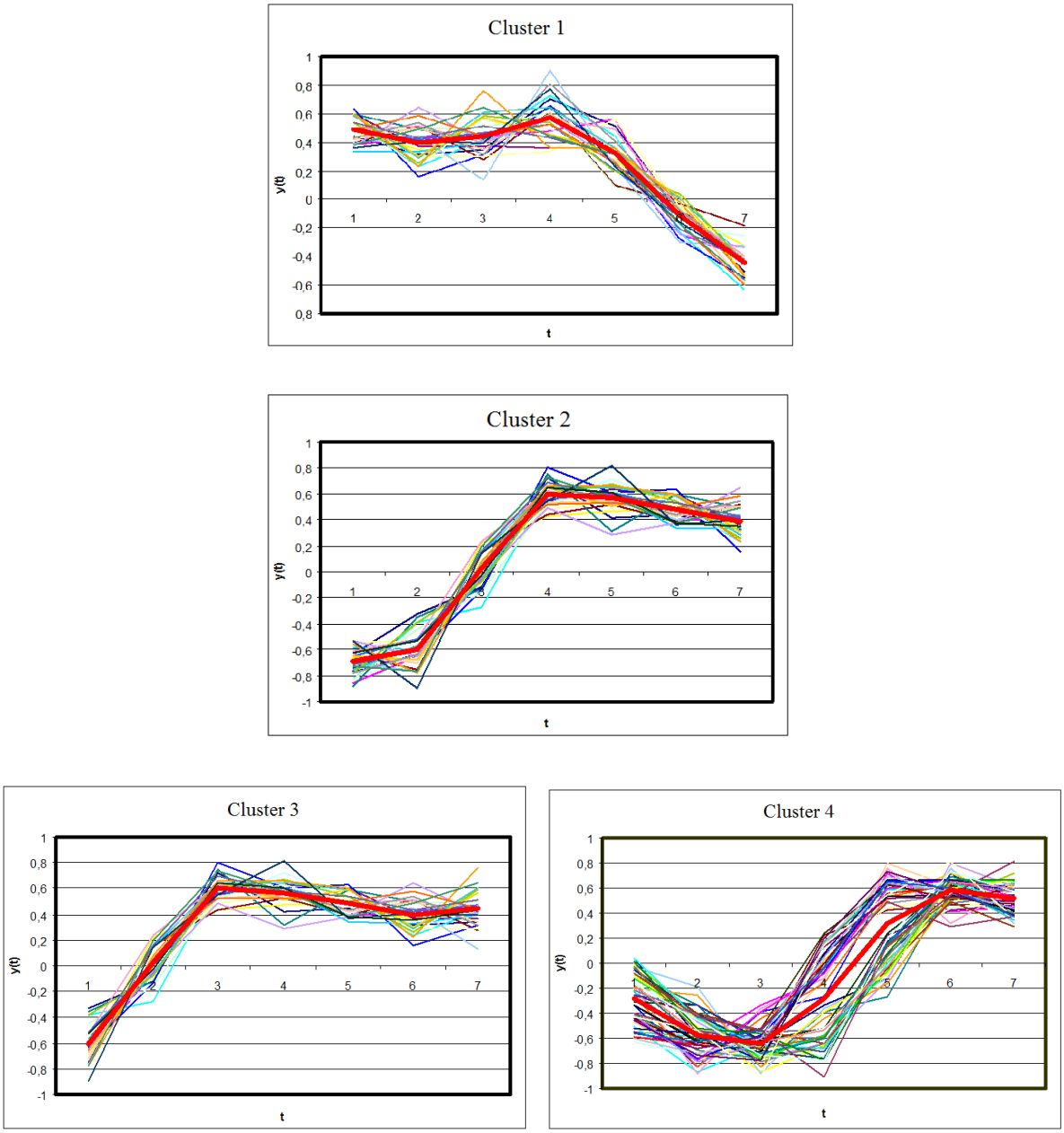


Fig. 1 *Examples of clusters, their prototypes illustrated as a bold red line*

2.4 Properties

The historical behaviour of the series is considered

While determining clusters, the similarity of the series behaviour is investigated, not only for actual rates, but for all other historical rates, as well. In this way, it is possible to put a given series, with high or specific current rates, into a cluster with lower rates series. This may occur when a series has different actual rates for the current date, but has had similar historical developments and behaviours in the past.

Series weights

Individual series may also have weights. If the weights are all equal to 1, then all series are equally treated in the clustering process. Otherwise, some of the series may be considered more or less important than others. If the weight of a series is two times bigger than the weight of another series, the former is regarded as two series with the same values. Weights can not only be integer, but any arbitrary real values as well.
The effect of the usage of weights is shown in Fig. 2. The centre between Series 1 and Series 2 is illustrated with a dashed red line. If those two series have the same weight, the centre will be very close to the prototype. Series 2, however, has the greater weight and thus attracts the prototype to itself.

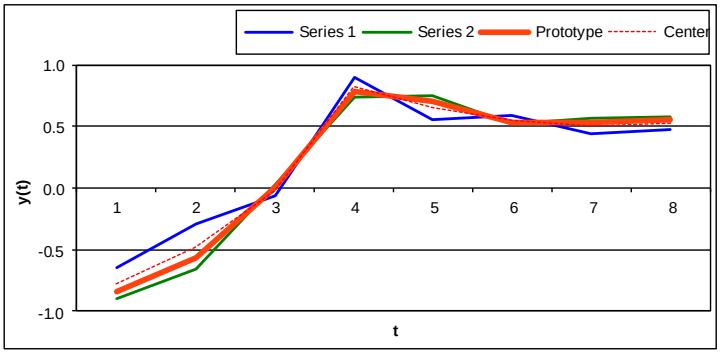


Fig. 2 *The effect of a series weight to the cluster prototype*

Historical values of the series can be taken into account

The comparison of series in the clustering process can be achieved by using a decay factor. Most commonly, the series comparison is done via the Euclidean distance (1), where the corresponding values are subtracted and powered. The decay factor, however, reduces the weight of the older historical values and increases the weight of more recent values (2). Decay λ is defined to be between 0 and 1. The greater the decay, the more weight is given to the last values in the series.

Euclidean distance without decay:

$$d = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2} \tag{1}$$

Euclidean distance with decay λ:

$$d = \sqrt{\frac{1}{\sum_{i=1}^{N}\lambda^{i-1}}\sum_{i=1}^{N}\lambda^{N-i}(x_i - y_i)^2} \tag{2}$$

where x and y are the series between which the distance is determined.

Determination of the optimal number of clustering

- The optimal number of clusters could be found by clustering in 2, 3, …, n clusters and by estimating an indicator for the clustering quality (e.g. adjusted R^2)
- A graphic demonstrates the plot where axis X is the number of clusters and axis Y is the clustering quality. The sharp that is represented in the graph, determines the optimal number of clusters (s. Fig. 3)
- The optimal number of clusters can be achieved in the following manner:
  - Performing the clustering many times over, increasing the number of clusters with a fixed value k>1. For example, clustering in 5, 10, …, n clusters
  - When the sharp drop (starting at point i) is found, additional clustering in i+1, i+2, …,i+k-1 clusters is done, in order to find the optimal number of clusters more precisely,

  This is shown in Fig. 4.

Num Clusters	Adjusted R squared	Error	Time (sec.)
2	0.7888814	0.2111186	20
3	0.8856307	0.1143693	32
4	0.9281010	0.0718990	40
5	0.9351225	0.0648775	60
6	0.9360977	0.0639023	70
7	0.9361842	0.0638158	95
8	0.9647925	0.0352075	109
9	0.9543623	0.0456377	122
10	0.9544117	0.0455883	144
11	0.9758081	0.0241919	154
12	0.9757913	0.0242087	173
13	0.9572335	0.0427665	180
14	0.9572007	0.0427993	194
15	0.9571655	0.0428345	218
16	0.9573212	0.0426788	225
17	0.9572855	0.0427145	260
18	0.9861978	0.0138022	276
19	0.9861863	0.0138137	287
20	0.9861746	0.0138254	305
21	0.9861818	0.0138182	314
22	0.9861700	0.0138300	326
23	0.9861583	0.0138417	362
24	0.9861466	0.0138534	381
25	0.9861356	0.0138644	379
26	0.9861230	0.0138770	429

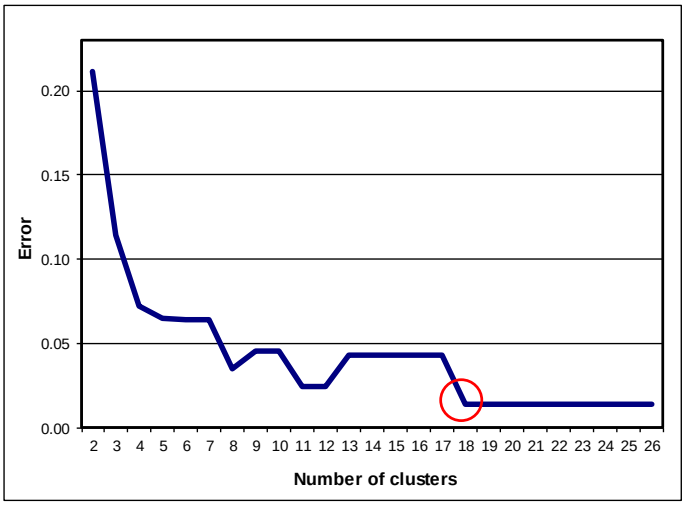
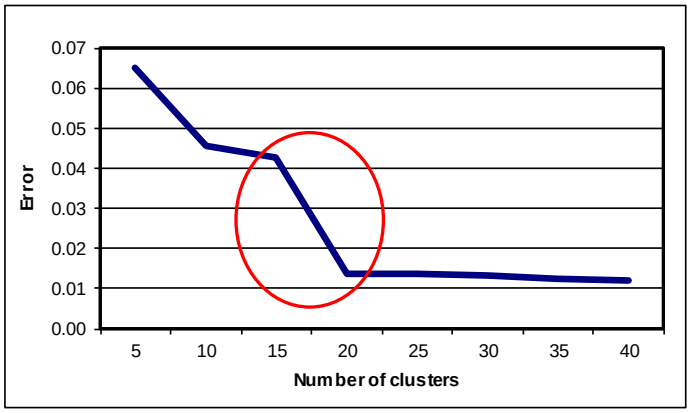


Fig. 3 *Clustering of 1200 shares in a different number of clusters and error calculation based on an adjusted R^2 to find the optimal number of clusters*

If there are too many data elements, the computation time could last too long. For this reason, searching for the optimal number of clusters could be performed on two levels, as shown in Fig. 5.

Num Clusters	Adjusted R Squared	Error
5	0.9351225	0.0648775
10	0.9544117	0.0455883
15	0.9571655	0.0428345
20	0.9861746	0.0138254
25	0.9861356	0.0138644
30	0.9868221	0.0131779
35	0.9875898	0.0124102
40	0.9878886	0.0121114



Num Clusters	Adjusted R Squared	Error
15	0.9571655	0.0428345
16	0.9573212	0.0426788
17	0.9572855	0.0427145
18	0.9861978	0.0138022
19	0.9861863	0.0138137
20	0.9861746	0.0138254

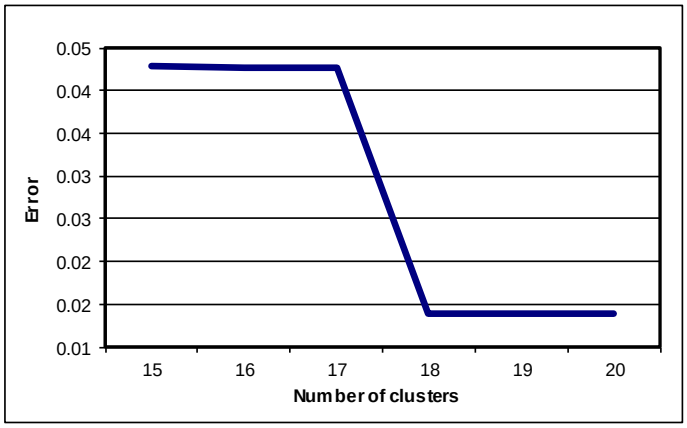
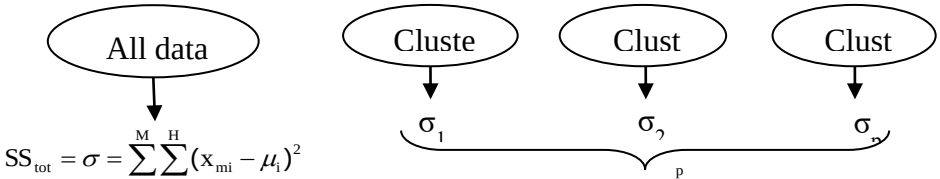


Fig. 4 *Overall number of clustering is 8 + 4 = 12, compared to 26 in the successive search*

Statistics about clustering quality are computed showing how good the clustering has been performed

- Adjusted R squared



$$SS_{tot} = \sigma = \sum_{m=1}^{M}\sum_{i=1}^{H}(x_{mi} - \mu_i)^2$$

$$SS_{sum} = \sum_{j=1}^{p}\sigma_j$$

$$\sigma_j = \sum_{m=1}^{M_j}\sum_{i=1}^{H}(x_{jmi} - \mu_{ji})^2$$

M – Number of all elements (shares or curves)
μ– center of all data elements
M_j – Number of elements in cluster j
H – Number of historical values
μ_j – Synthetic share/curve of cluster j

$$R^2 = 1 - \frac{SS_{sum}}{SS_{tot}}$$

$$\text{Adjusted}R^2 = 1 - (1 - R^2)\frac{M-1}{M-p-1} = 1 - \frac{SS_{sum}}{SS_{tot}}\frac{M-1}{M-p-1}$$

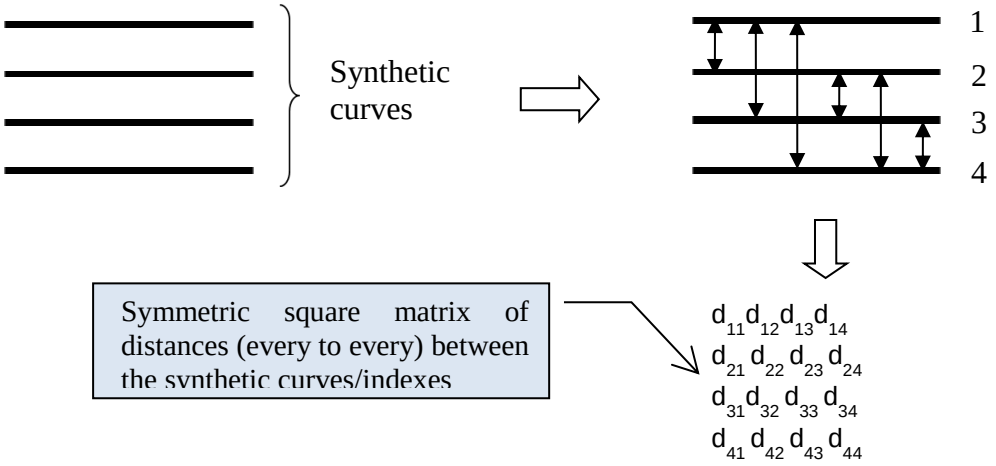
- Euclidean distances

  – Every to every

  Given synthetic curves with H historical values, the Euclidean distance between synthetic curves i and j is calculated as:

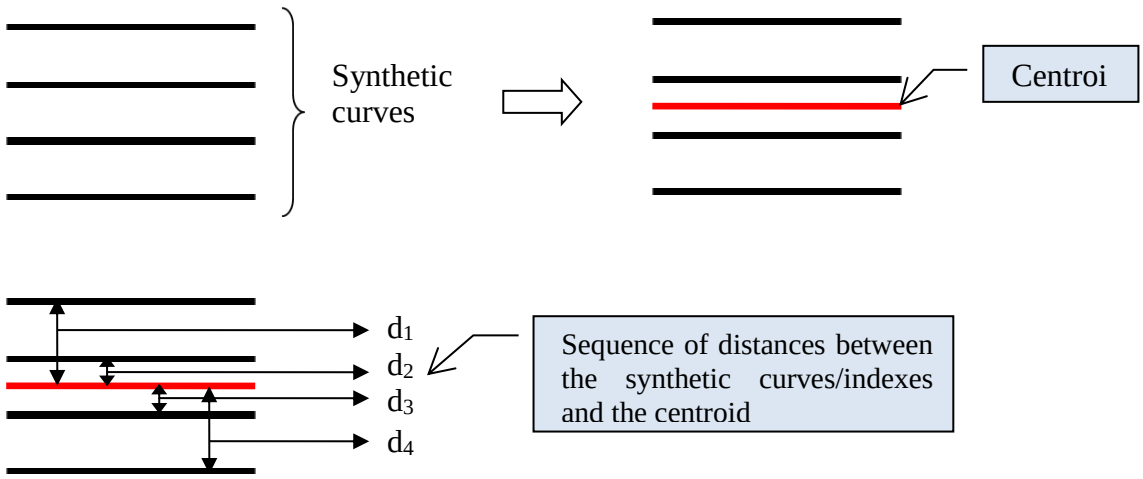
$$d_{ij} = \sqrt{\sum_{k=1}^{H}(c_{ik} - c_{jk})^2} \qquad (3)$$

Thus, a symmetric square matrix is obtained, showing the distances between synthetic curves/indexes.



– Every to centroid

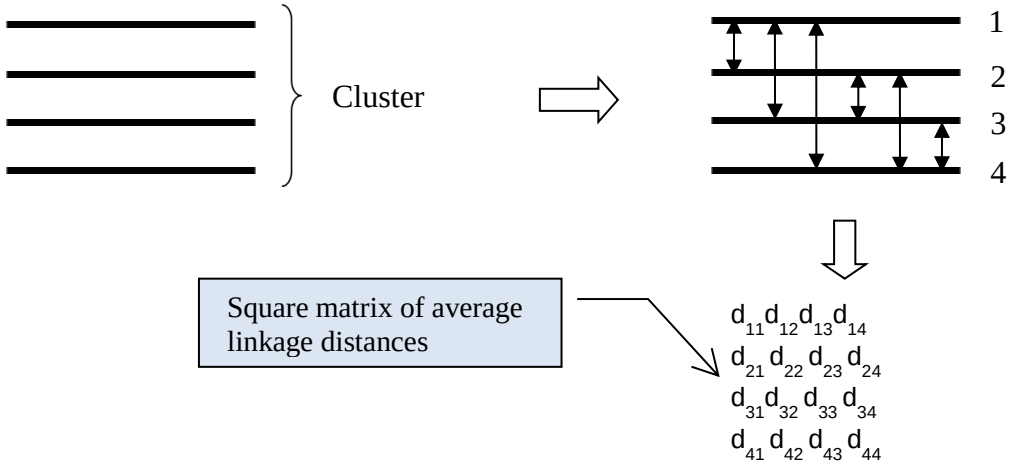
A sequence of values can be calculated, showing the distances between each curve/index and its calculated centre (centroid). The centroid has the same dimensionality as the curves/indexes and every value of the centroid can be defined as the average number of the corresponding values of synthetic curves/indexes.



- Average linkage
  Average linkage is a method of calculating the distance between clusters. The distance between two clusters is the average distance between objects (shares/indexes) from the first cluster and objects from the second cluster. The distance between cluster X and cluster Y is:

$$d_{xy} = \frac{1}{N_x N_y} \sum_{i=1}^{Nx} \sum_{j=1}^{Ny} d(x_i, y_j) \ , \quad x_i \in X, y_j \in Y \tag{4}$$

$d(x_i, y_j)$ - Euclidean distance between curve i from cluster X and curve j from cluster Y.



The usage of clusters prototypes

After performing the clustering, cluster prototypes are used for the correlation matrix and volatility vector in the Monte Carlo simulations. However, the volatilities are calculated for real market factors, too, so that Beta factors to prototype series can be obtained. In fact, Monte Carlo simulates the correlated prototypes, but each real factor is related to the prototype factor via Beta factors, i.e. the correlation is used from prototypes and the volatility is mapped by Beta factors. Thus, the data shrinking is performed for the correlation matrix, but real volatilities are also used in the simulation.

3. Non-normal distributions

3.1 Purpose

The purpose of the non-normal distributions identification module is an automatical determination of theoretical distribution types and their parameters, for a given time series. It utilizes a Copula approach, to simulate market factors in Monte Carlo VaR simulations, using mapped distributions. Thus, the best modelling of empirical distribution shape is achieved by reproducible theoretical distribution shape.

3.2 Input data

- Time series.
- Chosen standard distribution types (e.g. Beta, Cauchy, Student, Weibull, etc.).

3.3 Output data

- An identified distribution type that best fits the given time series.
- Parameters specific to the identified distribution type.
- Numerical estimation of the distance between the empirical distribution and all other distribution types. Thus, the distribution types can be ordered and other, better fitting distribution types can be manually selected.

3.4 Properties

The best distribution type determination

- The assumption that historical series of market risk factors are normally distributed is not necessarily always true. Real distributions are mostly skewed, having flat tails, which is related to the underestimation of market risk for improbable large losses. That is why the distribution identification is used to improve the Monte Carlo VaR simulation by non-normal distributions.
  - Automatical identification and mapping of 15 theoretical distribution types based on market factor time series.
  - Estimation of parameters of the selected, best distribution.
  - The distribution type and its parameters, as well as the comparison to empirical market distribution, can be manually set.
  - Two distance types are used to select the distributions: histogram and cumulative.
  - Using mapped distributions, the Copula approach enables the simulation of market factors in Monte Carlo VaR.

The non-normal distribution identification improves the Monte Carlo VaR simulation (Fig. 5) by using correlated non-normal distribution samples, instead of correlated normal ones. Steps on finding the best distribution type for a given data are:

1) Empirical distribution of the time series of a market factor is built in the market risk matrix.
2) For all theoretical distribution types, the following actions are performed:
   a) Parameters of the distribution type are estimated using available market data.
   b) The theoretical distribution is built using estimated parameters.
   c) Theoretical and empirical distributions are compared, and model distances are calculated. These calculation can be performed in two alternative ways, using either histogram or cumulative distances.
3) Order the distributions according to minimal distances and select the most fitting distribution, store it and for later usage in the Copula simulation.

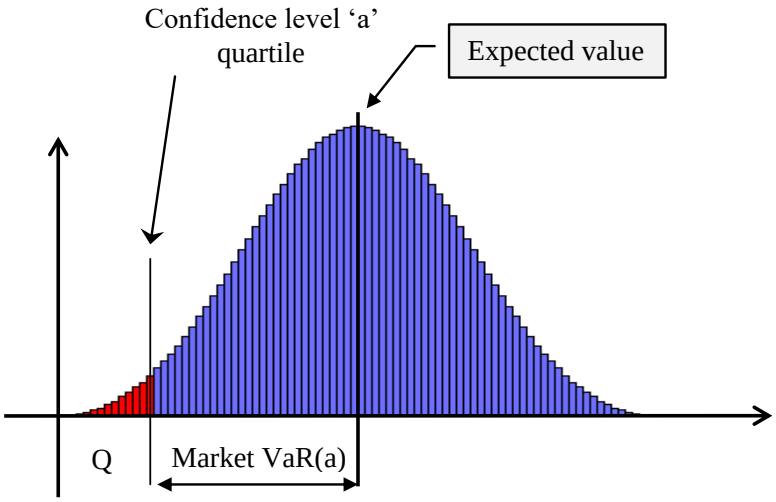


Fig. 5 *VaR calculation*

In Fig. 6, an example is given illustrating a beta distribution that better fits the time series distribution, than a normal distribution.

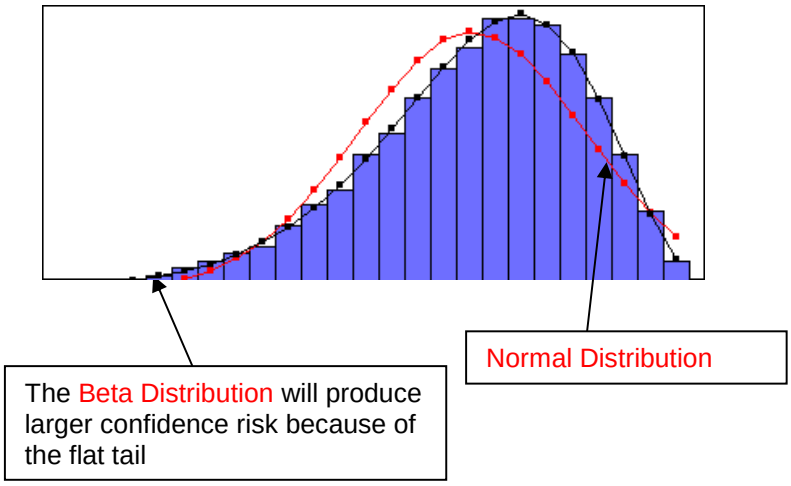


Fig. 6 *Flat tail distribution*

The hypothesis “goodness of fit test” generally indicates whether or not the data represented as time series belongs to a preliminary specified distribution type. Examples of such methods are “chi squared”, Kolmogorov-Smirnov, Anderson-Darling and others. In our case, the best distribution type is determined by choosing distributions from a list of specified pre-defined distributions. This can be achieved using two alternative measures: histogram measure and cumulative measure.

Histogram measure

Here, the average squared distance between the histogram bin frequencies of empirical and theoretical histograms is calculated (3).

$$d^2 = \frac{1}{k}\sum_{i=1}^{k}(O_i - E_i)^2 \tag{5}$$

The distances are shown in Fig. 7. This type of measure is similar to the “chi squared” goodness of fit test statistic, expressed by formula (4) [17] [20]. It requires the building of empirical and theoretical histograms, as well as work on binned data.

$$\chi^2 = \sum_{i=1}^{k}\frac{(O_i - E_i)^2}{E_i} \tag{6}$$

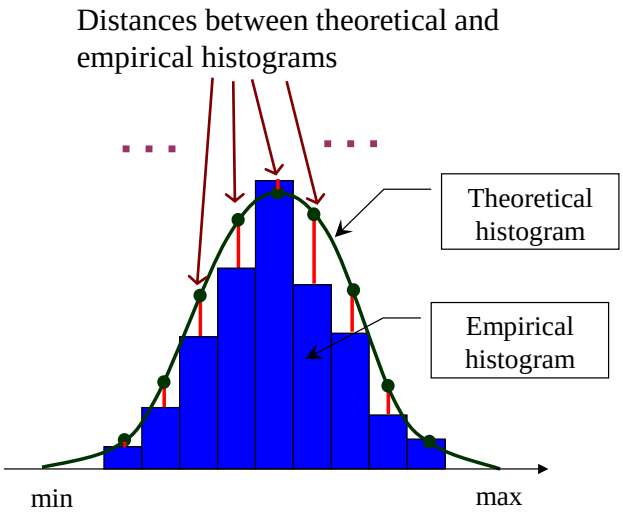


Fig. 7 *Histogram measure*

The empirical histogram can be easily built from sample data, but the theoretical histogram uses the Cumulative Distribution Function (CDF) for every distribution type. This means that, for each distribution type, the distribution parameters should be

estimated beforehand, in order to be applied to the CDF formula. Table 1 shows the most common distributions types with their parameters [12] [13] [14] [16] [19].

Distribution	*Distribution parameters*		*Additional parameters*	
	Parameter 1	*Parameter 2*	*Parameter 1*	*Parameter 2*
Beta	Shape	Shape	Shift	Scale
Cauchy	Location	Scale	---	---
Exponential	Rate	---	Shift	---
Inverse Normal	Mu	Lambda	Shift	---
Log Normal	Log Scale	Shape	Shift	---
Normal	Mean	Variance	Shift	---
Pareto	Scale	Shape	---	---
Rayleigh	Sigma	---	Shift	---
Student	Nu	---	Shift	Scale
Weibull	Scale	Shape	Shift	---
Logistic	Location	Scale	---	---
Gumbel	Location	Scale	---	---
Maxwell-Boltzmann	A	---	Shift	---
Gamma	Shape	Scale	Shift	---
Pearson type 7	Shape	Scale	---	---

Table 1 *Distribution Parameters*

There are one or two distribution specific parameters, along with one or two additional parameters.

- *Distribution specific parameters*. For each distribution type, specific parameters are estimated from the empirical sample data, using the method of moments, least squares regression, or the maximum likelihood estimation. In most cases, there is more than one possible way for the estimation. Thus, the choice is based on the established researches about their effectiveness.

- *Additional parameters*. From a practical point of view, it is possible for the sample data, or a part of it, to fall into a region in which the distribution is not defined. When taking into account that the main goal is to identify the distribution based on the probability distribution curve shape, the data is shifted and scaled, in order to compare the empirical to the theoretical shape (s, Fig. 8).

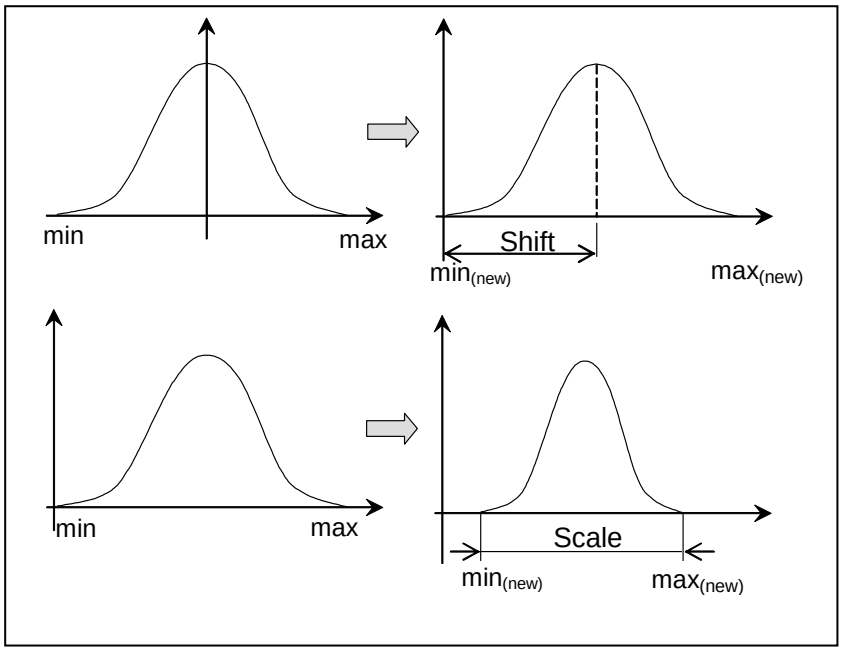


Fig. 8 *Additional parameters*

After obtaining both distribution specific and additional parameters. the theoretical histogram is built. The starting point is the known CDF:

$$F_x(x) = P(X <= x) \tag{7}$$

The theoretical histogram is built as the probability distribution function (PDF) from a distribution with known CDF (s. Fig. 9).

$$P(a < X <= b) = F_x(b) - F_x(a) \tag{8}$$

where Fx(x) is the CDF.

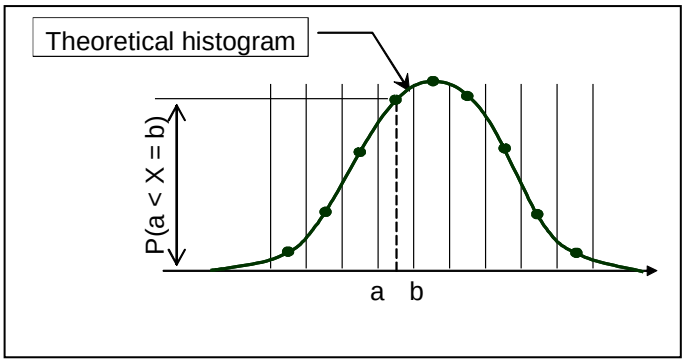


Fig. 9 *Theoretical histogram bins*

The empirical histogram is built by counting the values belonging to the bins. If a value is on the border between two bins, then half of the value is added to both of them. The number of bins in a theoretical histogram coincides with that of an empirical histogram. There are different ways to determine the number of bins. In our implementation, the following formula is used:

$$B = 5\log(N) \tag{9}$$

and

$$B < 5 \Rightarrow B = 5 \tag{10}$$

$$B > 25 \Rightarrow B = 25 \tag{11}$$

where:

N - series size.

Cumulative measure

Here, cumulative distribution values are used. This approach aims at increasing the accuracy of the best distribution identification, taking into account all values in the distribution identification process. The cumulative values approach works on non-binned data and therefore more computational operations are needed, leading to more time consumation, compared to the histogram measure. The method is as follows.

- Having the sample values in axis Y (shown on the left in Fig. 10), the differences between them are accumulated and the graphic of this function is considered.
- A shift of the function is performed, in order for its mean to coincide to the mean of the identical empirical data function.

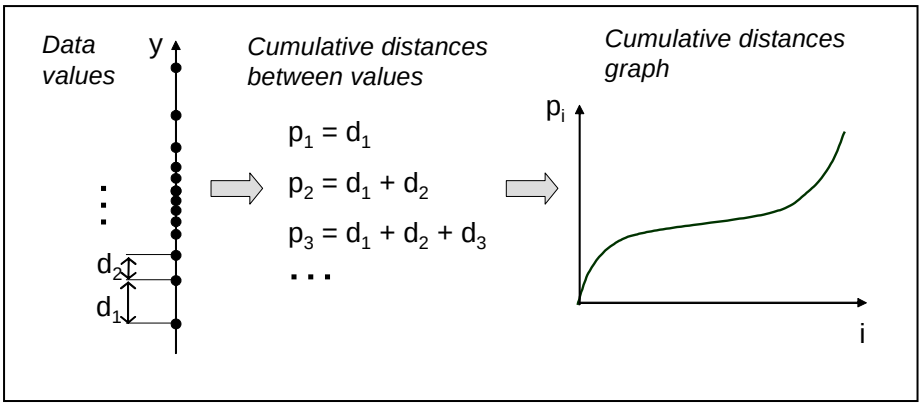


Fig. 10 *Cumulative distances*

The most difficult task in this approach is obtaining the sample with the theoretical distribution that corresponds to characteristics of the empirical sample data. Here again, as in the histogram measure, parameter estimations are used. The full theoretical sample is generated using the inverse CDF (12).

$$x = F^{-1}(\xi) \tag{12}$$

The same sample values generator is also used to test the distributions identification system. The generation of the sample, distributed according to a specified distribution type, is performed using uniformly distributed values in an interval from 0 to 1, generated either randomly or in equal distances.

Graphics of the histogram and cumulative measures are shown in Fig. 11 and Fig. 12.

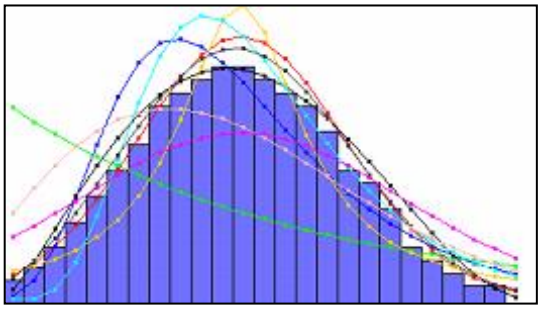


Fig. 11 *Graphics of the empirical and all theoretical histograms*

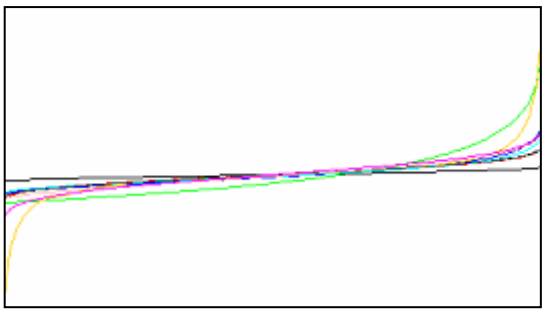


Fig. 12 *Graphics of empirical and cumulative values for all theoretical distributions*

The distribution of random numbers in a Monte Carlo simulation is important for the accuracy of the computed risks, as well as for the calculation performance. The market environment consists of many dependent risk factors and they are modelled as a multivariate random variable. Here, non-normal distributions can be used for the description of various risk factors. The Monte Carlo simulation then applies the mapped distribution with its parameters, to calculate the Value at Risk. Fig. 13 shows the usage of non-normally distributed factors in the Monte Carlo simulation.

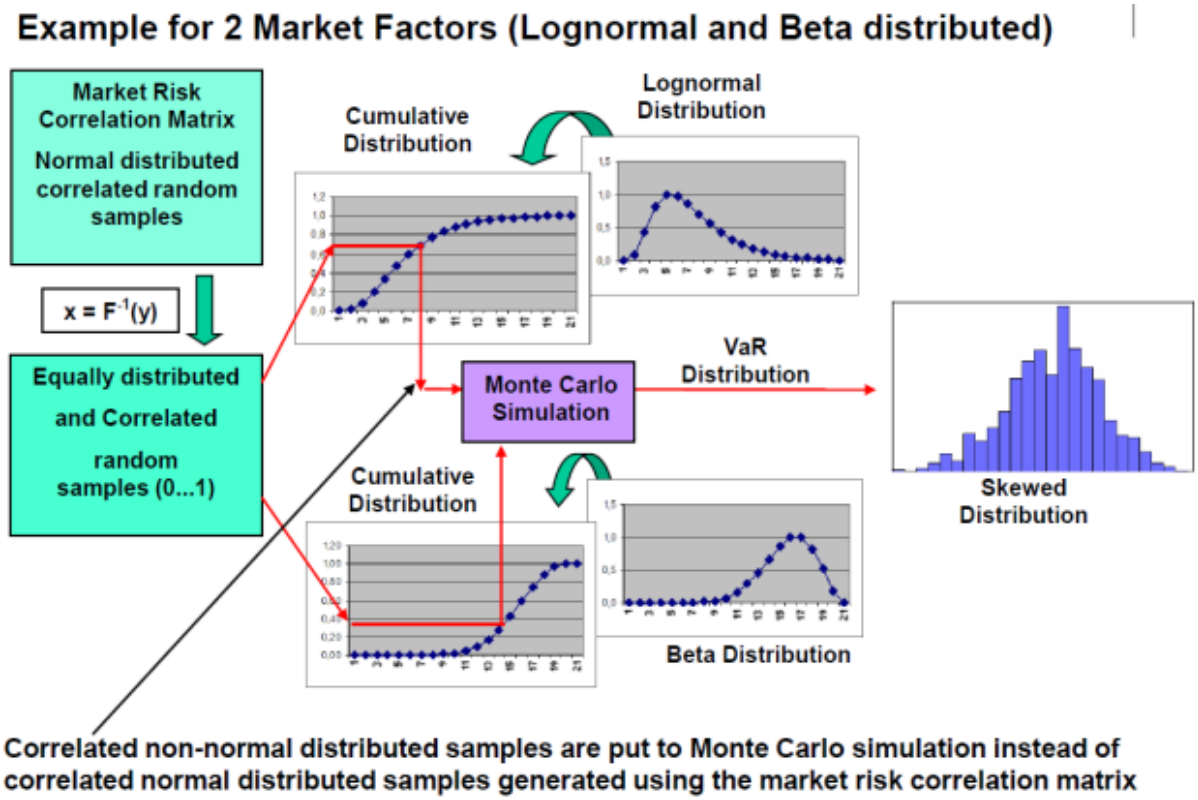


Fig. 13 *Usage of non-normal distributions*

When the simulation is performed, the distribution types of different market factors are already determined. Random, uniformly distributed values are generated for each series. Applying the inverse cumulative function of the preliminary identified distribution type for the series, the correlated non-normally distributed samples are used for the Monte Carlo simulation, instead of the normally distributed samples, generated using the risk correlated

matrix. In this way,VaR distribution is obtained more accurately. This process should be performed for every series. Immediately after the determination of the distribution type, the CDF is numerically built and preserved, in order to improve the performance for a simulation of a large number of samples (Fig. 14).

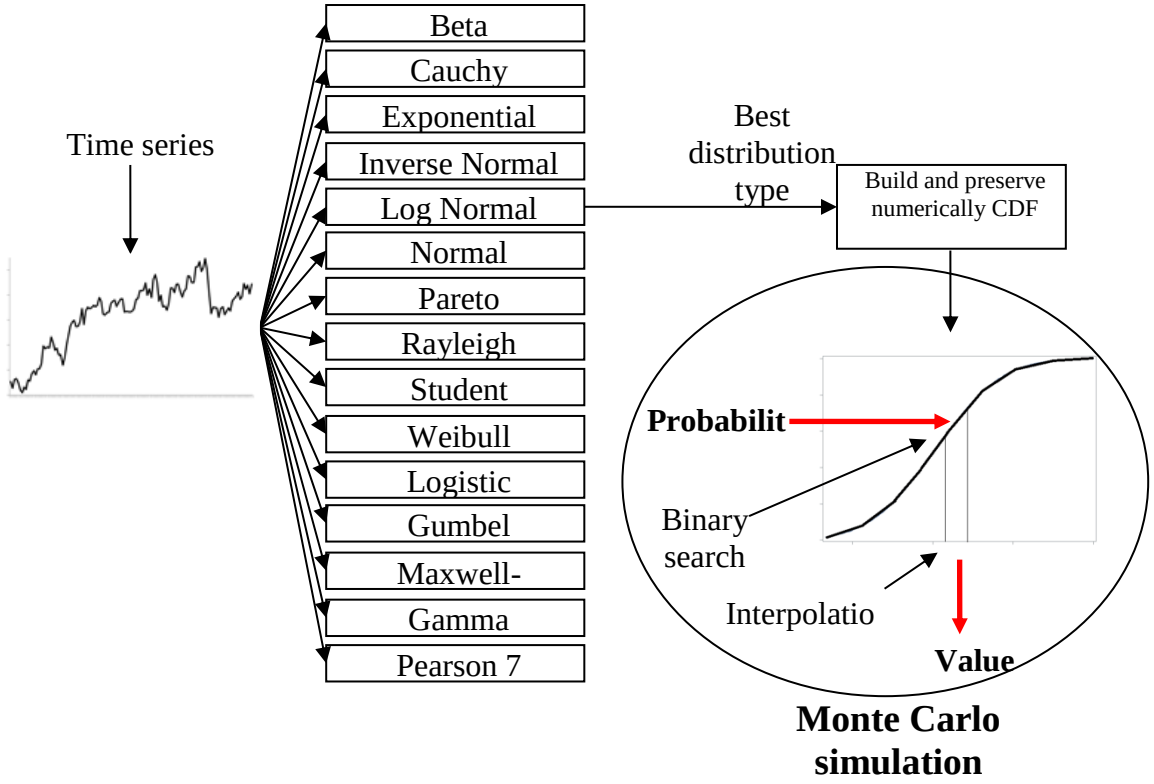


Fig. 14 *Determination and usage of the best distribution type*

Thus, the CDF resides in the memory and can later be used in the Monte Carlo simulation stage. Given that CDF values are sorted, ascending the next equally distributed value is transformed into a specifically distributed value ($x=F^{-1}(y)$) by a binary search in the numerically built CDF.

4. Multifactor modelling

4.1 Purpose

The purpose of multifactor modelling is to build formulas that describe time series by other multiple time series. The described series is called target series (or factor) and the series that describe the target are called explanatory series (factors).

4.2 Input data

- Historical time series of the target factor.
- Other available historical time series to be used as explanatory factors.
- Settings regarding the formula: functions that can participate, coefficients restrictions, etc.

4.3 Output data

- Polynomial formula, describing the dependency of target factors from explanatory factors.
- Target factor generated by the formula.

4.4 Properties

Multifactor modeling requires both target, as well as all explanatory factors, in order to be on the same time horizon. However, because of their different natures, these factors often have missing and incomplete data values. In cases where the time series are received from different sources in different countries, that have different holiday or non-working days, or in cases where the database contains some corrupted values, interpolation and volatility bridge approach can be used.

When time series of the modeled target factor and modeling explanatory factors are available, the goal is to use a multifactor approach for the target. This approach is build up on an estimation of a polynomial-based expression, found by regression and other intelligent techniques over known market factors and functions over them. The pricing expression should reproduce the original target time series as best as it possibly fits.

After the expression estimation step, the pricing expression can be used in the near future as a pricing model for the target factor, which allows pricing, VaR, volatility estimation and price projection. If the future price estimation deviates enough from the market price of the target series, a new expression calibration is performed /Fig. 15).

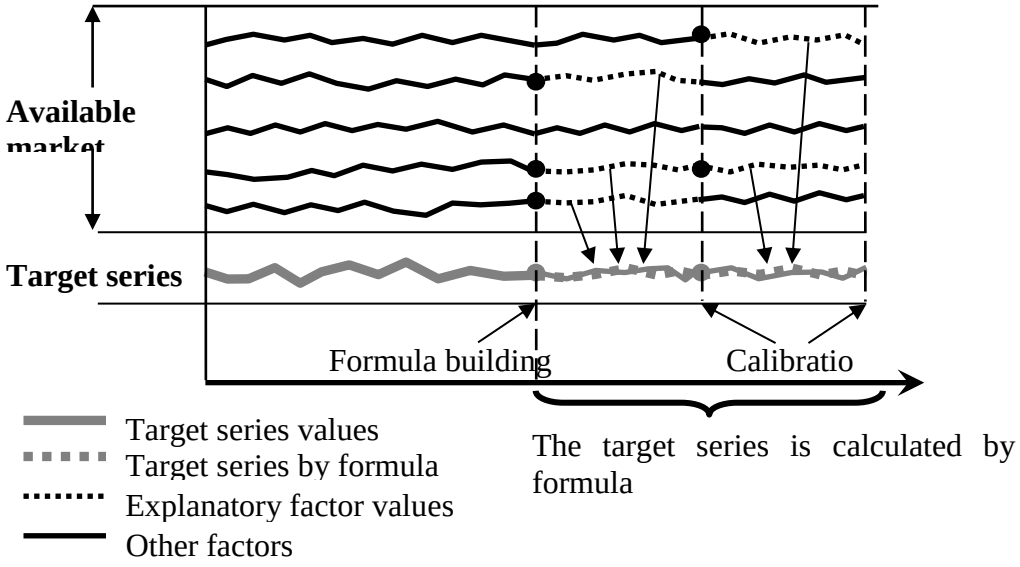


Fig.15 *Multifactor modeling*

The formula (expression estimation) looks as follows:

$$y = \beta_1 f_1(x_1) + \beta_2 f_2(x_2) + \ldots + \beta_n f_m(x_n) + \beta_{n+1} + \varepsilon \qquad (13)$$

where:

y	- target factor
$x_1, x_2, \ldots, x_n$	- explanatory factors
$\beta_1, \beta_2, \ldots, \beta_n, \beta_{n+1}$	- regression coefficients
$f_1, f_2, \ldots, f_m$	- basis functions
ε	- error

Stages of the modeling process are shown in Fig. 16.

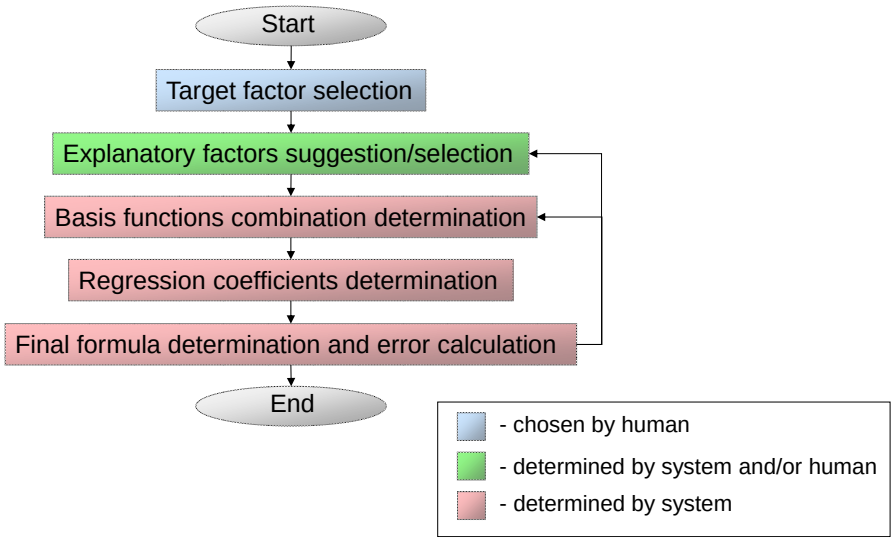


Fig. 16 *Stages of the multifactor modelling*

The first step in the modelling is target series selection. After that, explanatory factors should be selected via an automatic suggestion and/or hand choosing. The whole set of explanatory candidates would be too big and some of the series are not as descriptive as others.

An automatic suggestion can be performed in one of the following ways.

- *Clustering*. Explanatory factors are obtained from the cluster in which the target factor is classified. If the number of explanatory factors, determined in this way, is insufficient, the number of clusters can be decreased in order to increase the number of elements in the cluster. This approach of automatic suggestion is based on the clustering that is described earlier in this document.
- *Minimal covariances between candidate factors*. The covariances between all factors are calculated, where first minimal values are used to determine explanatory factors.
- *Maximal covariances between candidate factors and the target factor*. The first maximal values are used to determine the explanatory factors in this case.

The suggested factors can be manually changed,or other factors can be added or removed. When both target and explanatory factors are selected, the formula building process can be started in which the system should find a combination of basic functions to the explanatory factors and regression coefficients β_i calculation. The basic functions are used to improve the accuracy and possibly to avoid linear dependencies between factors that cause matrices equations problems.
Given target and explanatory factors and basic functions combination, regression coefficients β are estimated by solving the matrices equations (14-16) [4] [10] [11].

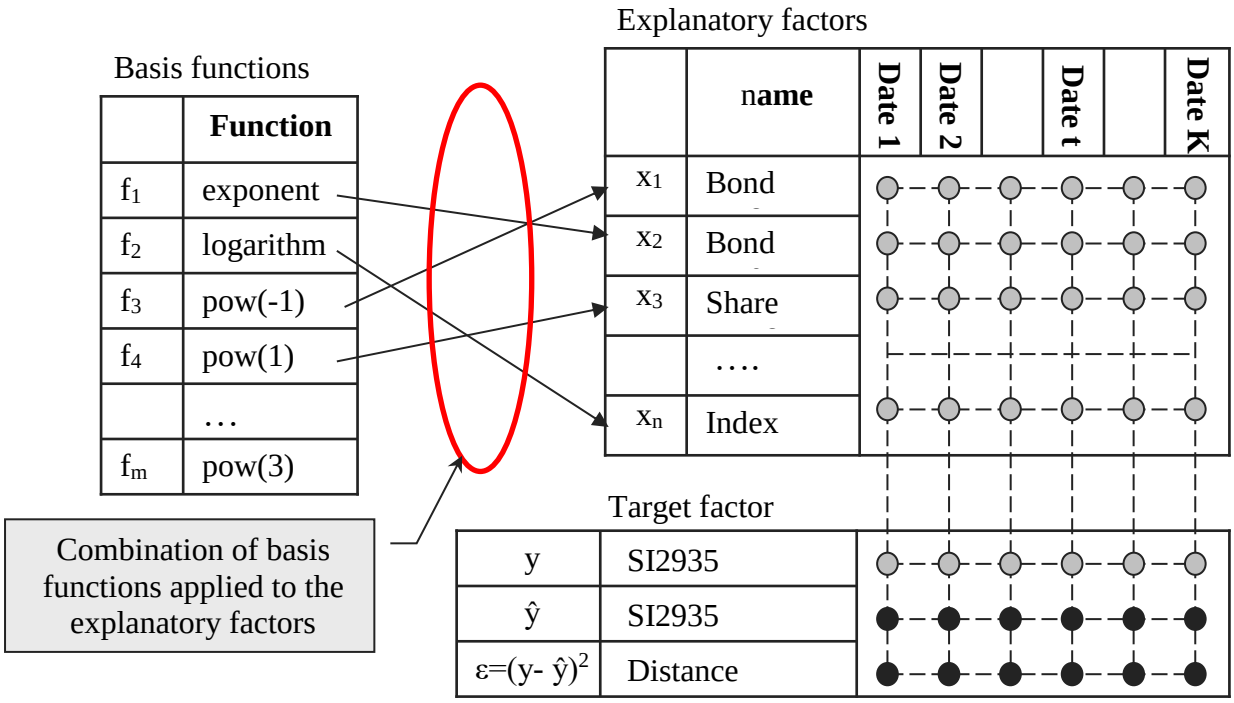
Target factor | Explanatory factors (factor 1, factor 2, … factor n) | Regression coefficients

$$\begin{matrix} \text{date 1} \\ \text{date 2} \\ \dots \\ \text{date k} \end{matrix} \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_k \end{pmatrix} = \begin{pmatrix} f_1(x_{11}) & f_2(x_{21}) & \dots & f_m(x_{n1}) \\ f_1(x_{12}) & f_2(x_{22}) & \dots & f_m(x_{n2}) \\ \dots & \dots & \dots & \dots \\ f_1(x_{1k}) & f_2(x_{2k}) & \dots & f_m(x_{nk}) \end{pmatrix} \times \begin{pmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_m \end{pmatrix}$$

Y = A × B (14)

$$\hat{B} = (A^T A)^{-1} A^T Y \quad (15)$$

$$\hat{Y} = A \times \hat{B} \quad (16)$$



$$\hat{y} = \beta_1 f_3(x_1) + \beta_2 f_1(x_2) + \ldots + \beta_n f_2(x_n) + \beta_{n+1} + \varepsilon$$

Fig. 17 *Multifactor modelling*

The formula is built in such a way that the distance ε must be minimized. To attain a notion of the modeling quality, additional statistics for the fitting quality should be shown – correlation, R squared, adjusted R squared, target and synthetic volatility. In Fig. 18, settings of the multifactor are presented.

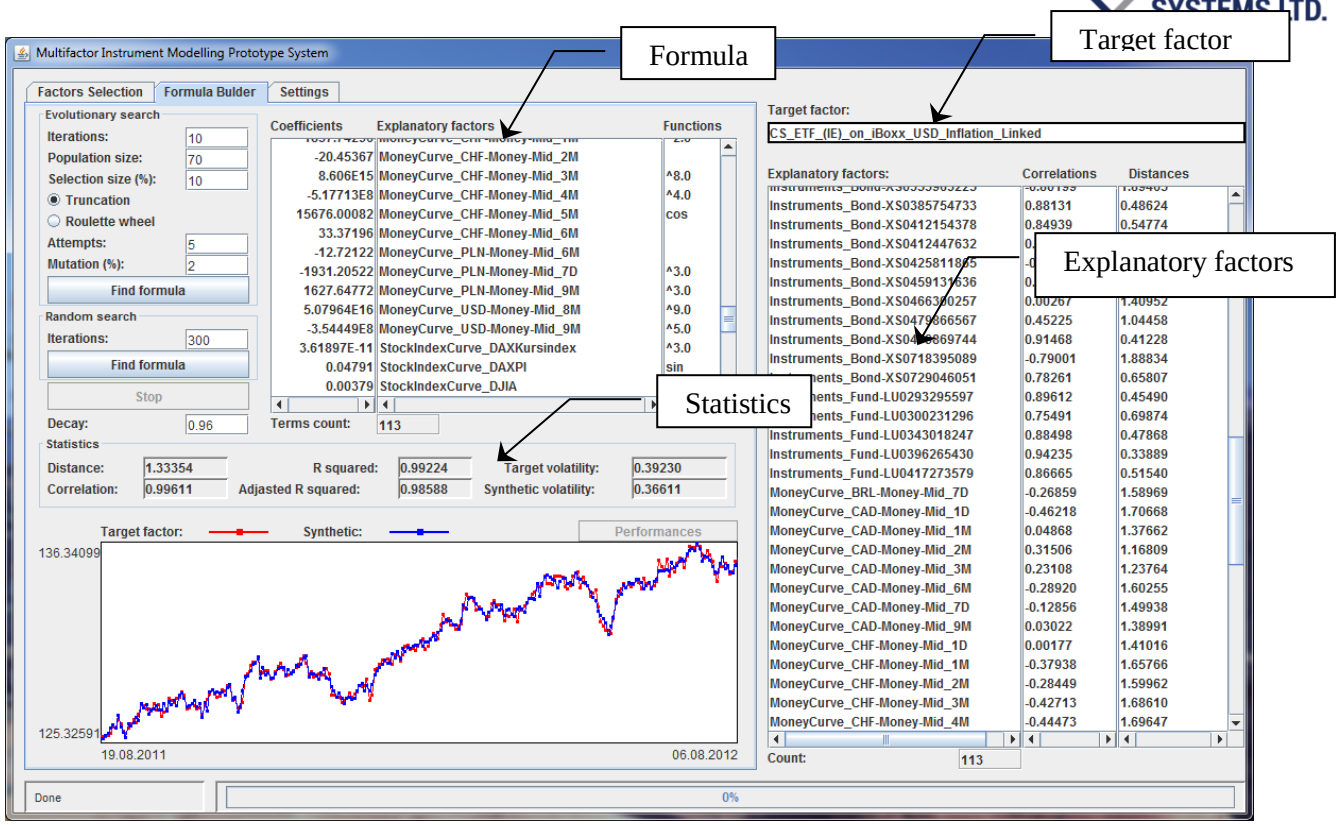


Fig. 18 *Multifactor modeling*

- Formula terms
  - Remove small coefficients. Small formula term coefficients can be removed, as they do not significantly influence the formula results.
  - Include free term without explanatory factor. The free term does not include an available factor. It is a constant and allows more degrees of freedom in improving the formula accuracy.
  - Include the target factor into explanatory factors. In most cases, this is not achieved, because the formula is not applicable to real-life situations.
  - Auto seed. If the seed stays fixed, different attempts produce the same results with the same data. By choosing the auto seed option, each attempt will generate different formulas. The reason is that, in the formula generation approach, a random generator used that can reproduce the same or different results in every generation.
- Factors
  - Performance. Factors can be pre-processed prior to the modeling and post-processed after the modeling stage.
  - Performance for clustering. Factors can be pre-processed only for the factors selection stage, via the clustering approach.
- Basis functions selection

  The basic functions that participate in the formula must be selected beforehand. For some functions it is possible to not participate in the final formula, but they are all used in the formula finding step. If no basis functions are selected, factors participate with their raw values. Depending on the factors' nature, it is possible for some basis functions to be

unsuitable for factor values (for example, a large factor value applied to an exponential basis function will produce a value near to infinity and would cause an error).
Formula terms could be reduced by removing the ones whose regression coefficients are smaller than the given threshold. In this case, the corresponding factor should be excluded from the formula.

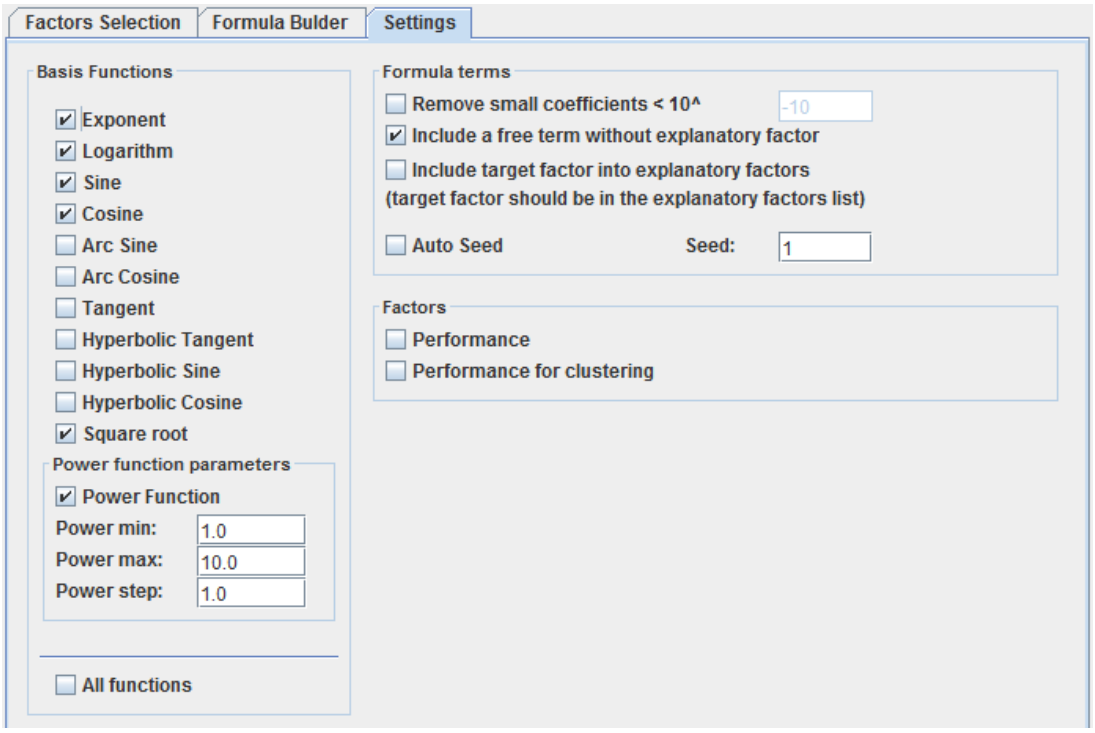


Fig.19 *Multifactor settings*

A constant (free regression coefficient without a factor) could also be included in the formula. It is represented as a factor with all historical values equal to 1.0 and without a basis function applied.

The following standard basis functions can be included:

- Sine
- Cosine
- Tangent
- Hyperbolic tangent
- Hyperbolic sine
- Hyperbolic Cosine
- Square root
- Exponent
- Logarithm
- Arc sine
- Arc cosine

- Power function, etc.

Fig. 20 illustrates an example of a target series with synthetic series generated by applying the formula in which they are juxtaposed.
A good modeling is seen, where a decay factor is applied additionally, enabling the actual values (near the end of the series) to be more important in the modeling. Thus, the given and generated series are more alike at the end of the series than in the beginning.

Multifactor calibration

This formula calibration is shown in Fig. 21. First, target and explanatory factors are selected and loaded. The first version of the formula is created and calculations are performed with this formula version. Afterwards, the generated target begins to deviate from real market values and the formula must be corrected. The selected explanatory factors, that are used for building the first formula version, are used again, but the formula is calibrated and new functions and coefficients for the formula terms are produced.

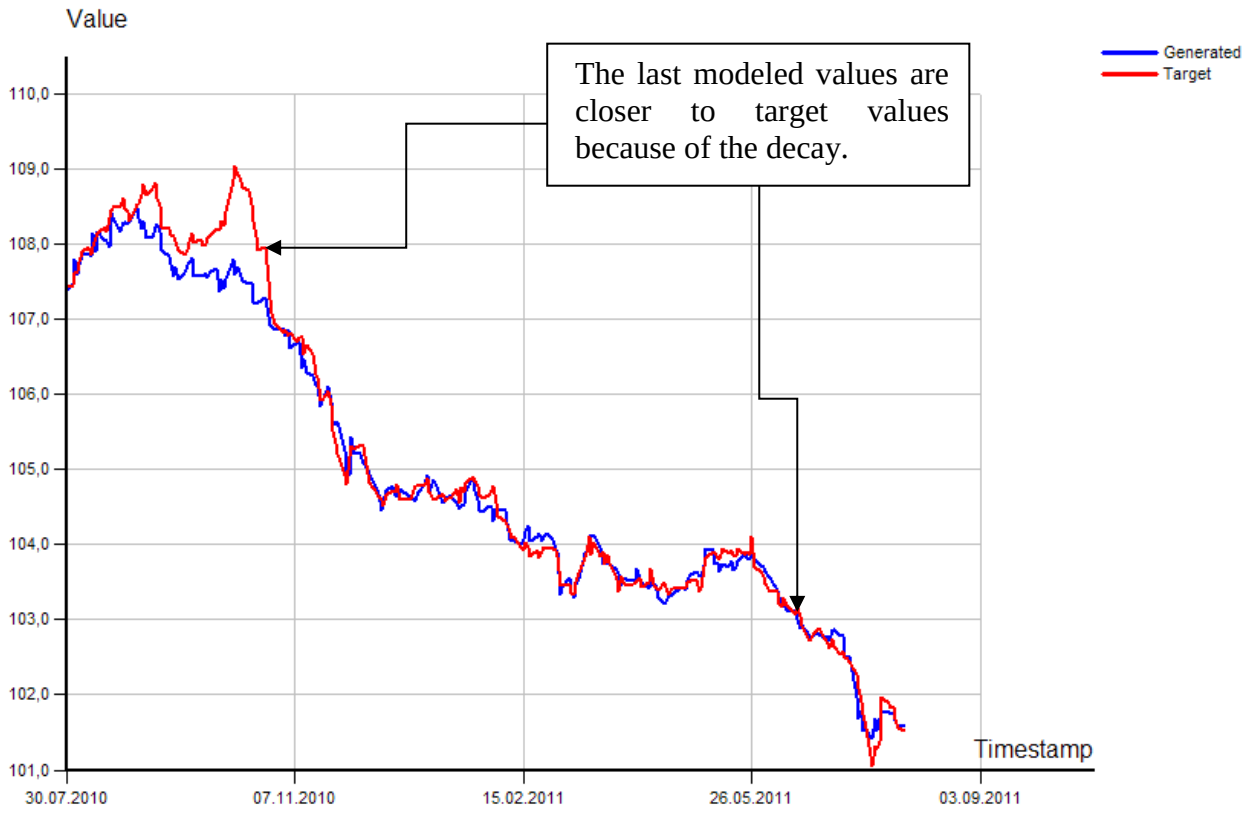


Fig. 20 *A given target series and a synthetic series generated by the multifactor module*

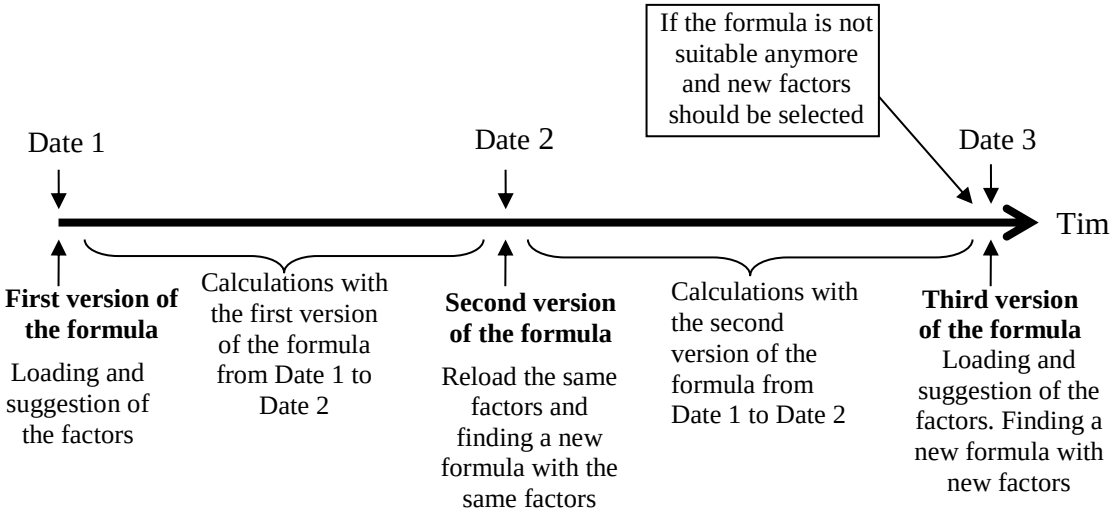


Fig. 21 *Formula calibration*

For this reason, a second formula version is created and it is further used to generate the target, until its values start to deviate from the real target values. If this deviation is too significant, new explanatory factors should be selected. The factors are loaded again, a selection of new factors is performed by one of the above mentioned automated approaches (clustering, min or max correlated) and they can be manually changed. The selected factors are used to generate the third formula version, that, as opposed to previous formula versions, includes different explanatory factors, basis functions and coefficients. This formula can later be used in further calculations, until new calibration is needed.

There are two kinds of calibrations:

1) Preserving currently selected explanatory factors and changing only the functions applied to them, as well as and term coefficients, including the free term.
2) Selecting new explanatory factors. In this case, new factors can be added, existing factors removed, or both. The formula has completely changed regarding basis functions and terms coefficients.

In each formula calibration, settings can be changed, i.e. the set of basis functions can be used in the formula, removing terms with small coefficients, etc.

Sensitive analysis

Multifactor analysis can be used for the sensitivity analysis. Coefficients β_1, β_2, . . . β_n in formula (13), also shown in Fig. 18 and Fig. 22, can be used to analyze the factor participation in the modeling of the target factor.

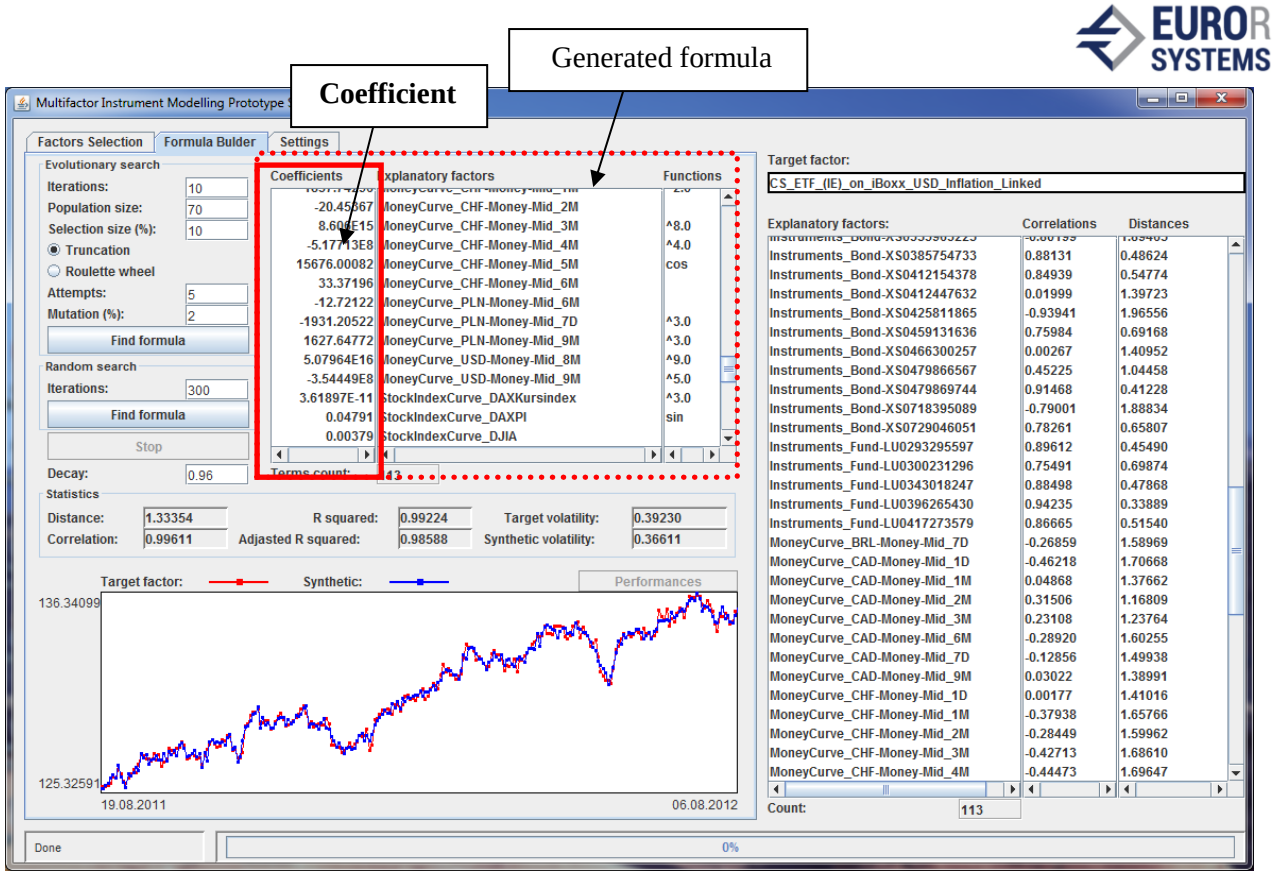


Fig. 22 *The generated formula*

This allows the relationship between price changes (P) the risk factors to be expressed. According to the delta-gamma approach, in which the change in the price is:

$$\Delta P \approx \delta^T \Delta Z + \frac{1}{2}\Delta Z^T \Gamma \Delta Z \tag{17}$$

where δ_i is the first derivative of the risk factor:

$$\delta_i = \frac{\partial P}{\partial z_i} \tag{18}$$

estimated by

$$\frac{P(\ldots,z_i+h_i,\ldots)-P(\ldots,z_i-h_i,\ldots)}{2h_i} = \frac{s_i^+ - s_i^-}{2h_i} \tag{19}$$

The diagonal elements of Γ are:

$$\Gamma_{ii} = \frac{\partial^2 P}{\partial z_i^2} \tag{20}$$

estimated by:

$$\frac{P(\ldots,z_i+h_i,\ldots)-P(\ldots,z_i,\ldots)}{h_i^2} - \frac{P(\ldots,z_i-h_i,\ldots)-P(\ldots,z_i,\ldots)}{h_i^2} = \frac{s_i^+ - s_i^-}{h_i^2} \tag{21}$$

The cross elements of the Γ are:

$$\Gamma_{ik} = \frac{\partial^2 P}{\partial z_i \partial z_k}, (i \neq k) \tag{22}$$

estimated by:

$$\frac{P(\dots, z_i + h_i, \dots, z_k + h_k, \dots) - P(\dots, z_i + h_i, \dots, z_k - h_k, \dots)}{4h_i h_k} + \frac{P(\dots, z_i - h_i, \dots, z_k - h_k, \dots) - P(\dots, z_i - h_i, \dots, z_k + h_k, \dots)}{4h_i h_k} \tag{23}$$

The changes and their relationships can be modeled and analyzed by a multifactor modeling approach.

5. Prediction

5.1 Purpose

The purpose of the prediction module is forecasting a given time series for a given time horizon, by analysing the series' historical development.

5.2 Input data

- Historical time series.
- Settings according to the used approach (for example, learning iterations, time window size, etc.).
- Predictions of the time horizon (the length of the predicted sub-series).

5.3 Output data

- Time series, with additionally predicted values added at the end.
- Prediction quality statistics.

5.4 Properties

- The time series prediction is a field with too many prediction methods. Some of the most commonly used are:
  - Averages (Moving Average (MA), Weighted Moving Average (WMA), Exponential Weighted Moving Average (EWMA), etc.
  - Autoregressive methods (Autoregressive (AR), Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA), Seasonal

Autoregressive Integrated Moving Average (SARIMA), Autoregressive Moving Averages with Exogenous Input Model (ARMAX), Self-Exciting Threshold Autoregressive (SETAR), etc.) with Box-Jenkins methodology.
  - Trend-extrapolation (based on Least Squares Error (LSE), trend polynomial finding, etc.).
  - Neural Networks (Multilayer perceptron, Radial basis functions, Self-organizing map, Adaptive resonant theory, recurrent Elman/Jordan networks, etc.).
  - Other regression based (e.g. Observers) and econometric models.
  - Kalman, Wiener and other filters.
  - Wavelet based methods.
  - Holt-Winter decomposition.
  - Hybrid approaches.
- Predictions can be used for technical analysis, algorithmic trading and other similar purposes.
- Excluding the exact predictions, the confidence bands could also be used to determine intervals in which the predictions can fall, as shown in Fig. 23, marked in grey colour around the predicted values. This figure illustrates an example of historical values of a time series and its prediction for the horizon of some future values.
- Predictability indicators can be suggested (Hurst exponent, etc.). When a prediction is performed, some indicators for the prediction quality should be calculated for preliminary separated test values.
- Fig. 24 shows one of the most commonly used approaches for time series prediction using the sliding window [7] [8] [9].

There are different mathematical models using this approach. Box-Jenkins approach, autoregressive methods, as well as neural networks are amongst them. The work included in these methods can be separated into two stages: model identification and prediction. The former is based on historical values analysis, while the latter uses a built mathematical model from the first stage in order to generate predictions. The model parameter identification can be considered an optimization problem and the prediction is propagation of the input through the model.

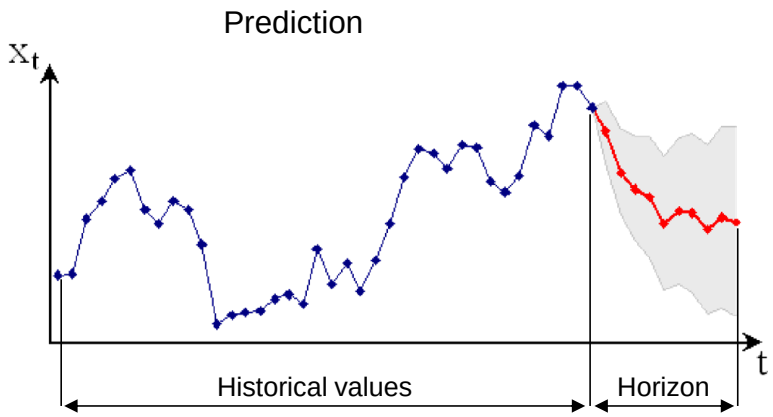


Fig. 23 *Time series prediction for a given time horizon with confidence bands*

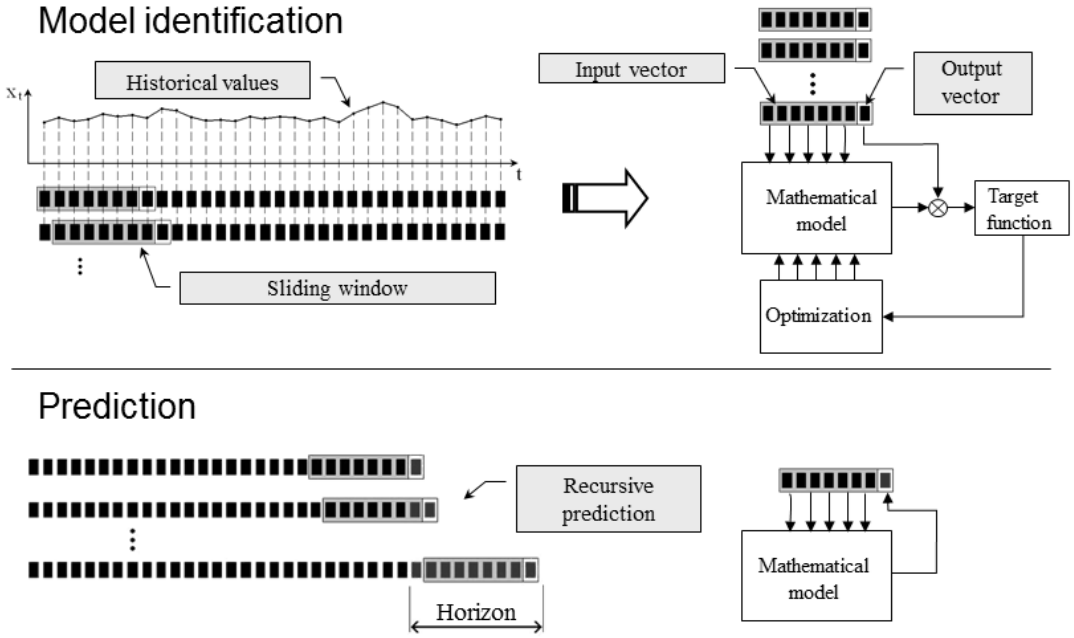


Fig. 24 *Time series prediction via the sliding window approach*

This approach is used in both neural network and autoregressive techniques. One of the most difficult tasks here is the window size determination that coincides to the number of model parameters. For this purpose, one of the following methods can be used:

- Analysing the autocorrelation function (ACF) and Partial Autocorrelation function (PACF). These functions are built for the time series. The point in which they become smaller than a given significance level is chosen to determine the time window. This approach emerges from the Box-Jenkins methodology. It is commonly used in the autoregressive prediction, but also in other similar methods, such as neural network prediction.

- Brute force searching. This method performs a prediction for all possible window sizes and errors are calculated according to different criteria.

- Trend brute force searching
  If there is *t* historical values $x_1 \ldots x_t$ the linear trend L_a is built for them and it is extrapolated in the future time horizon for which predictions will be performed. Here, the future time horizon is equal to the historical series size *t*. After that, for every possible window size *p* from *minWindowSize* to *maxWindowSizet* predictions $x_{t+1} \ldots x_{2t}$ are generated in the future and the linear trend L_f is built for the whole series $x_1 \ldots x_{2t}$, consisting of both historical and predicted values. Value *p.* for which the Euclidean distance d_t between the extrapolated L_a and L_f is minimal, is chosen to be the window size. After our investigation, *minWindowSize* and *maxWindowSize* are chosen to be $\frac{t}{5}$ and $\frac{t}{2}$ respectively.

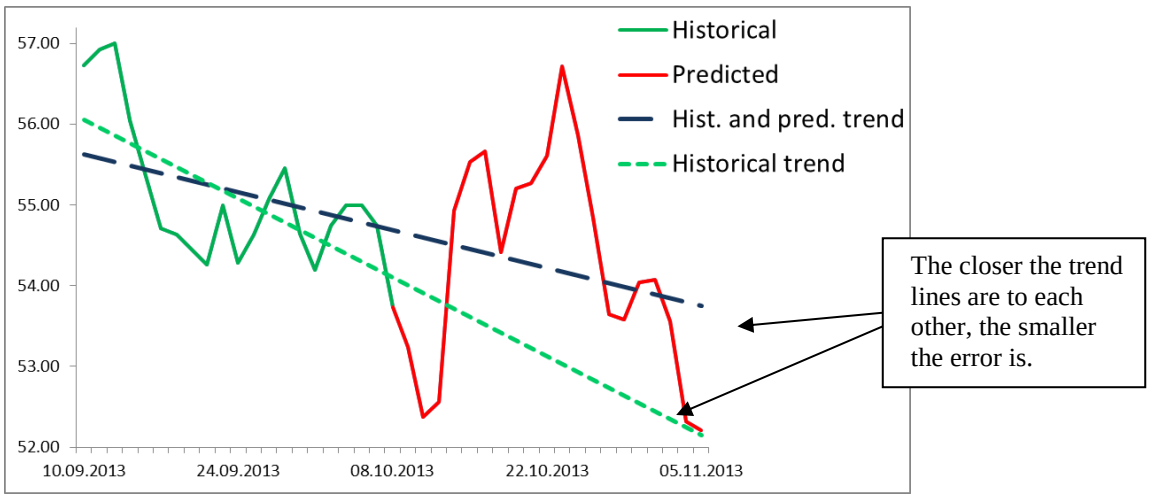


Fig. 25 *Trend brute force searching*

- Variation brute force searching
  In this approach, variation S_a is calculated for the historical values $x_1 \ldots x_t$. For every possible window size *p,* from *minWindowSize* to *maxWindowSizet,* predictions $x_{t+1} \ldots x_{2t}$ are generated and variation S_f is calculated for $x_{t+1} \ldots x_{2t}$. Value *p* is chosen to be the window size, for which the distance $d_v = |S_a - S_f|$ between historical and predicted values variation, is minimal.

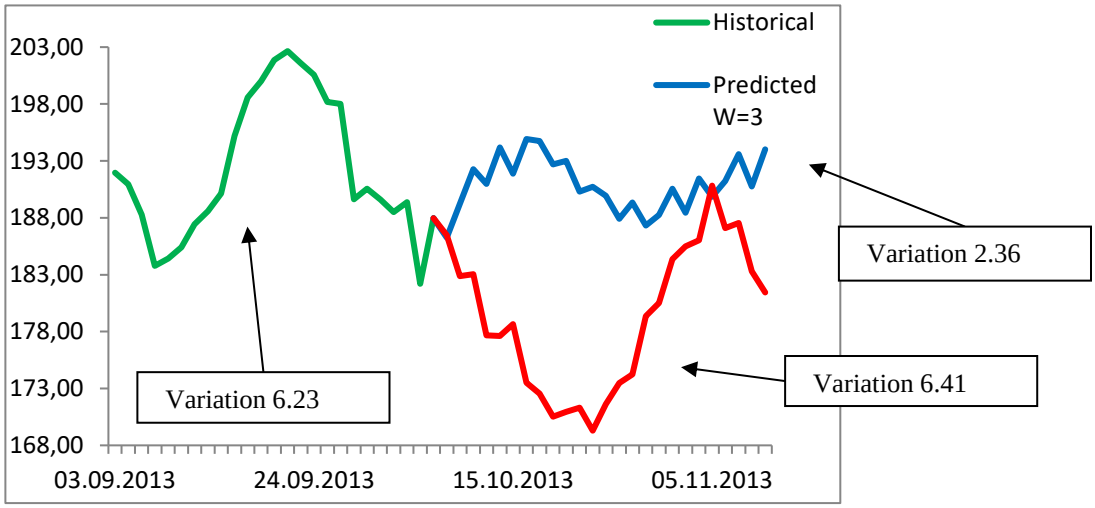


Fig. 26 *Predictions in red colour are better because their variation is closer to the variation of the historical time series*

- Trend and variation brute force searching
  This approach represents a combination between the two previous approaches. Both d_t and d_v are calculated and their sum $d_s=d_t+d_v$ is calculated for every *p* from *minWindowSize* to *maxWindowSize*. Value *p,* for which d_s is minimal, is chosen to be the window size. Here, different weights can be applied to both d_t and d_v in the calculation of d_s.

– An inconvenience in the time series prediction methods is the great number of settings required for the mathematical model. Most of them are precised using a trial and error approach. Settings also depend on time series properties. as shown in Table 2.

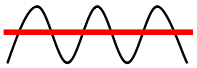
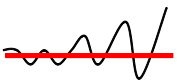
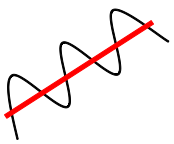
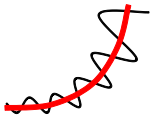
	No	Additive seasonality	Multiplicative seasonality
No			
Additive			
Multiplicative			

Table 2 *Time Series Probabilities*

In order to enable predictions of time series, not depending from its characteristics, some pre-processing should be performed, prior to the building of the model, as well as some post-processing after the prediction.

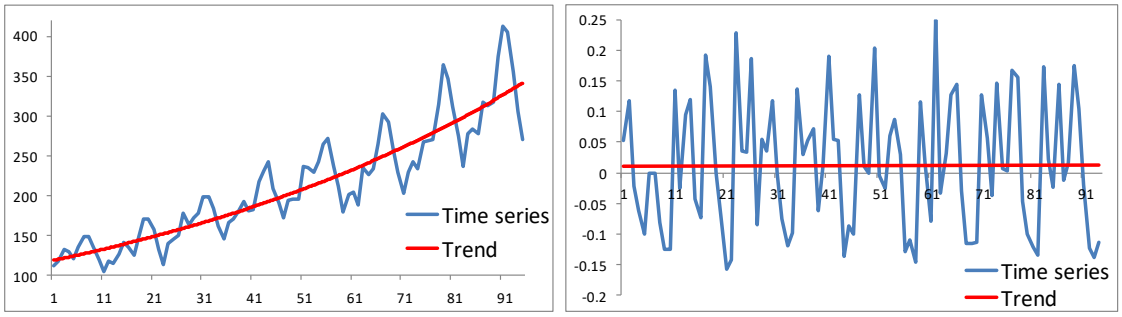


Fig. 27 *An original series with its trend on the left and the series after pre-processing on the right*

The series is rarely used in its pure values. Most often it is pre-processed by:

$$x_t = \frac{y_t - y_{t-1}}{y_{t-1}} \tag{24}$$

Thus, the series x is one value shorter than the original series. After the prediction, the reverse operation is performed by:

$$y_t = x_t, \text{ t=0} \tag{25}$$

$$y_t = (x_t + 1)y_{t-1}, \text{ t > 0} \tag{26}$$

The aim of the time series is to predict each value by some previous values of the series in the following way:

$$x_{t+1} = \phi_1 x_t + \phi_2 x_{t-1} + \ldots + \phi_p x_{t-p+1} + e_{t+1} \tag{27}$$

This is, in fact, an autoregressive (AR) process [4] [5] [6] [10] [11]. In order to use it, order p and coefficients ϕ_i should be determined. One of the methods to achieve do that is to solve the matrix equation (28) and finding of Φ.

$$\underbrace{\begin{pmatrix} x_{t+1} \\ x_{t+2} \\ \vdots \\ x_{t+s} \end{pmatrix}}_{b} = \underbrace{\begin{pmatrix} x_t & x_{t-1} & \cdots & x_{t-p+2} & x_{t-p+1} \\ x_{t+1} & x_t & \cdots & x_{t-p+3} & x_{t-p+2} \\ \vdots & & & & \\ x_{t+s-1} & x_{t+s-2} & \cdots & x_{t+s-p+1} & x_{t+s-p} \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{p-1} \\ \phi_p \end{pmatrix}}_{\Phi} \tag{28}$$

Another way is to use the Yule-Walker equations [3] [5] [6]. The order *p* is equal to the window size shown in Fig. 24. Thus, after determining coefficients ϕ they can be applied to the last values *x* of the series, obtaining the first prediction. Taking it as a real value, the next prediction can be obtained using the same formula, and so on. Applying this method, the generated predictions are shown in Fig. 28, where the black line shows the history and the red line expresses the predicted future.

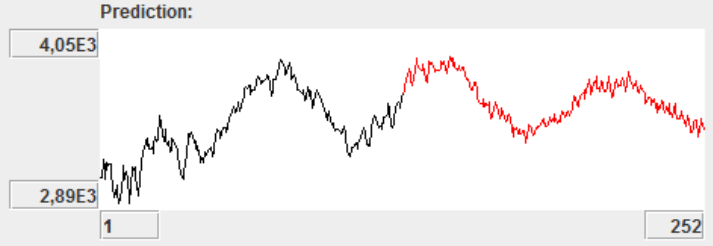


Fig. 28 *A time series prediction*

Time series is shown with seasonality. The autoregressive predicted values are shown in red colour. It can be seen that the seasonal effect is successfully predicted, which is important for data processing. Additionally, with pre-processing and post-processing, the trend is removed and restored before and after the model building respectively, thus obtaining qualitative prediction and taking into account both seasonality and the trend, independently of their additive or multiplicative behaviour.

6. Monte Carlo Simulation

6.1 Purpose

The purpose of the Monte Carlo simulation is to find the Value-at-Risk (VaR) for a given data and portfolio.

6.2 Input data

- Data represented as time series and portfolio data.
- Settings, such as the number of runs, confidence interval, and default categorization in case of missing data and risk type.

6.3 Output data

Market risk distributions graphically displayed via histograms or reports.

6.4 Properties

A market risk involves the uncertainty of future earnings, resulting from changes in various independent series in the market environment for a certain future time point (risk horizon) [2]. The market risk of a series is measured by means of a single value, called market Value-at-Risk (VaR). Market VaR represents unexpected losses at adverse market movements and measures the risk based on a probability of loss and a specific time horizon in which this loss can be expected to occur. The regulators use VaR to define the requirements for the trading, since VaR models can be used to estimate the loss of capital due to the market risk.

There are different methods to calculate Market Risk, amongst which the most powerful are parametric VaR/CoVaR, historical simulation and structured Monte Carlo simulation [1] [15] [18]. The module that carries out the structured Monte Carlo Simulation, can create scenarios for the market development in the future and can compute the series to be analyzed successively, by using scenarios (Fig. 29).

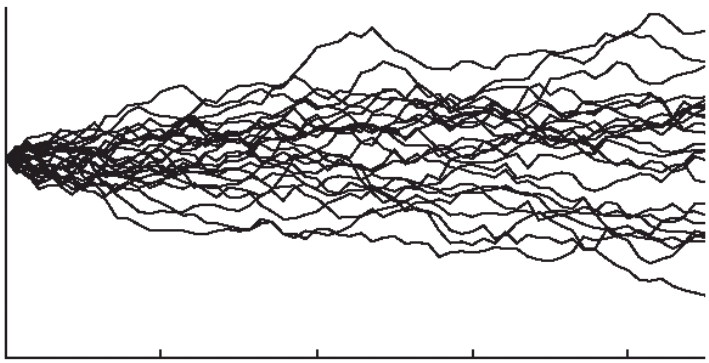


Fig. 29 *Generated scenarios*

The resulting outcome series represents a price distribution, indicating the risks with confidence.

The systematic evaluation of the Monte Carlo Simulation is shown in Fig. 30 and is based on the steps described below.

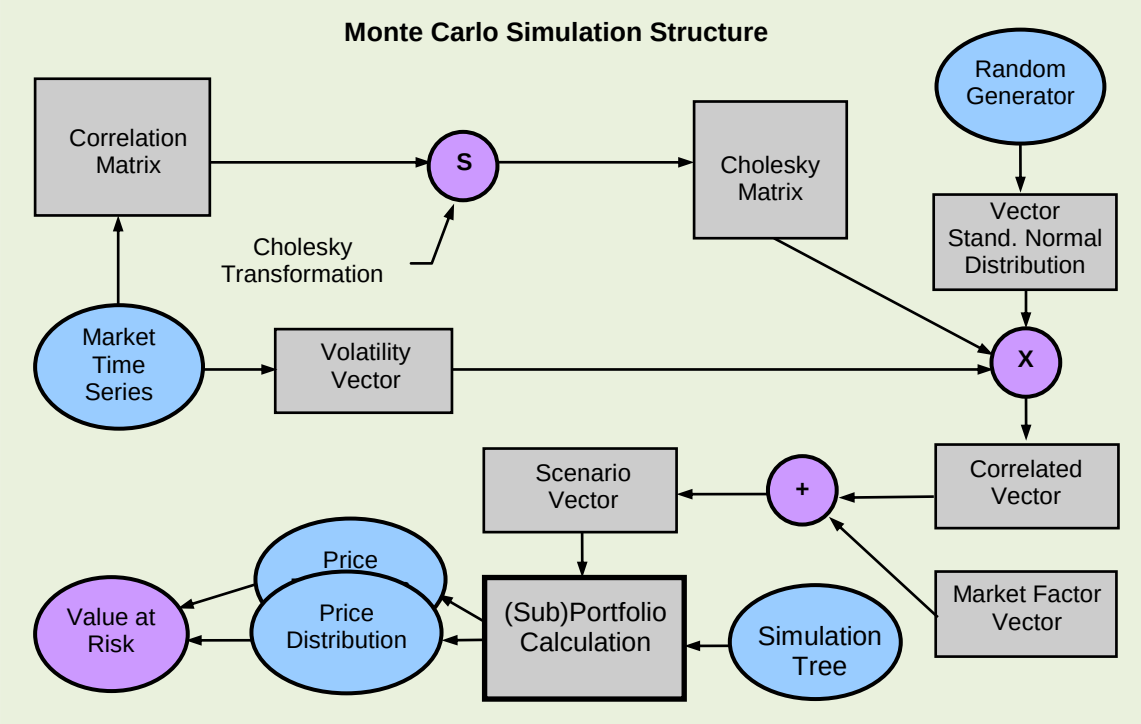


Fig.30 *Structured Monte Carlo simulation*

1) Volatility vector (V) and correlation matrix (R) for a set of market risk drivers, such as interest rates, FX rates, prices and spreads, are either produced from statistics of its own historical series in the PMS data base, or are imported from JPMorgan standard risk data sets.
2) Covariance matrix (C) is obtained by a matrix multiplication of volatility vector (V) and correlation matrix (R).
3) Cholesky matrix S is constructed from the covariance matrix (C), so that S x S' = C,c where S' is the transposed matrix.
4) Matrix Perturbation or Spectral Decomposition adjusts the covariance matrix to be positive definite, in case the original matrix is not positive definite, as a result from dependencies of risk drivers.
5) A random generator produces independent samples for each market driver, using standard normal distribution SND (0,1) stored into deviation vectors (D). A SVD correction and normal form correction applied to random vectors ensures normal distribution, where mean = 0, standard deviation = 1, skewness = 0, kurtosis = 0 and correlation between

factors = 0. The high quality of the random series reduces the number of Monte Carlo runs. 5.000 to 10.000 runs are usually enough to obtain good results.

6) Deviation vector (D) is multiplied by the Cholesky matrix (S), taking into account the covariance between every risk driver pair. This produces delta vectors (B) of normal distributed as well as correlated deviations.
7) Vectors (B) are applied to actual risk driver values (interest rates, FX rates, spreads, etc.) in the asset vectors (A). This produces scenario vectors (E) for Monte Carlo runs that compute the pricing expression value, using pricing trees to represent instruments and portfolios supplied by a Pricing Tree Generator. Pricing trees are specialized data structures ensuring a simulation's high performance.
8) Steps 6 is repeated a number of times (5000…10000). The large scenario set affects the calculation of price distributions of pricing expression. The resulted price distribution is constructed by counting the appearance of the values within many adjacent small ranges. Thereafter, Value at Risk is calculated via a numerical integration of the distribution density, using a confidence percentage, for example 1%, for the upper limit of the price distribution.

The Monte Carlo simulation steps described above use a set of well-known mathematical operations:

Calculation of the covariance matrix

$$C_{i,j} = R_{i,j} * V_i * V_j \quad , \qquad \text{i = 1...n (matrix width), j<= i (triangle matrix)} \tag{29}$$

where:

C_{ij} - Element from covariance matrix

R_{ij} - Element from correlation matrix (R_{ii} =1)

V_i, V_j - Element from volatility vector

Construction of the Cholesky matrix S

$$S \times S' = C \tag{30}$$

where:

S' - Transposed matrix,

a_{ij} - Elements of the matrix S

For a matrix where the width = 3, the following matrix operation applies:

$$C = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix} = \begin{pmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \times \begin{pmatrix} a_{11} & a_{21} & a_{31} \\ 0 & a_{22} & a_{32} \\ 0 & 0 & a_{33} \end{pmatrix} \tag{31}$$

The matrix multiplication involves formulas for the calculation elements of the Cholesky matrix:

$$\begin{aligned} a_{ii} &= \left(c_{ii} - \sum_{k=1}^{i-1} a_{ik}^2 \right)^{1/2} \\ a_{ij} &= \frac{1}{a_{jj}} \left(c_{ij} - \sum_{k=1}^{i-1} a_{ik} \cdot a_{jk} \right) \qquad j = i+1, i+2, \ldots, N \end{aligned} \tag{32}$$

Positive definite matrix

A matrix is said to be symmetric if A=A', where A' is the transpose of A. Therefore, covariance and correlation matrices are real and symmetric. A real symmetric matrix A is said to be positive definite if x'Ax is positive for every vector x different from 0. Given this definition, one can prove that a real symmetric matrix is positive definite if, and only if, all its eigenvalues are greater than zero.
The question is: "How do the eigenvalues of a matrix A change if it is subject to a perturbation A → A+E?" Because the eigenvalues are continuous functions of the entries of A, it is natural to think that, if the perturbation matrix E is small enough, the eigenvalues should not change too much. For our purposes, it would be undesirable to drastically change a matrix in order to turn it into a proper covariance matrix.
There are two methods that can transform a non-positive definite matrix into positive one.

Matrix Perturbation

The Monte Carlo engine uses the following perturbation:

A→A+e(I-A) = B, where e is a scalar and I is the identity matrix.

If e = 0 then A→A and if e = 1, then A→I, which is clearly a positive definite matrix. Therefore, since the eigenvalues of a matrix are continuous functions of its entries, there is an infinite number of scalars e in the interval [0, 1] that, given the equation above, are a positive definite matrix. The task is to obtain the min of e, which is calculated from the min of eigenvalues.

Spectral Decomposition

Given the right-hand-side eigensystem S' of the real and symmetric matrix R and its associated set of eigenvalues $\{\lambda_i\}$ such that

$$R \,.\, S = \Lambda \,.\, S \text{ with } \Lambda = \text{diag}(\lambda_i), \tag{34}$$

define the non-zero elements of the diagonal matrix Λ' as

$$\Lambda' : \lambda'_i = \begin{cases} \lambda_i : \lambda_i \geq 0 \\ 0 : \lambda_i < 0 \end{cases} \tag{35}$$

If the target matrix is not positive-semi-definite, it has at least one negative eigenvalue and at least one of the λ'_i will be zero.
In addition, define the non-zero elements of the diagonal scaling matrix T with respect to the eigensystem S by

$$T : t_i = \left[\sum_m s_{im}^2 \lambda'_m \right]^{-1} \tag{36}$$

$$B' := S\sqrt{\Lambda'} \tag{37}$$

$$B := \sqrt{T} B' = \sqrt{T} S \sqrt{\Lambda'} \tag{38}$$

By the construction,

$$R := BB^T \tag{39}$$

is now both positive semi-definite and has unit diagonal elements. A procedural description of the method above may clarify what, in fact, has to be done:

- Calculate the eigenvalues λ_i and the right-hand-side eigenvectors s_i of R
- Set all negative λ_i to zero.
- Multiply the vectors s_i with their associated "corrected" eigenvalues λ'_i and arrange as columns of R'.
- Finally, R results from R' are obtained by normalising the row vectors of R' to unit length.

Correction of random series

1) SVD Correction of expected value, standard deviation and correlation

For the first correction of expected value, standard deviation and correlation is created by a common post-processing. The approach is based on calculating the statistical figures of random numbers and applying the differences to the ideal case for the correction.

Formal Task

Given the matrix X(n,k), where n lines contain the generated k-dimensional normally distributed random samples. The statistical parameters of these samples – m(k) vector of mean values and K(k,k) covariance matrix – can be calculated. The distribution, caused by Matrix X, is represented and follows the normal distribution $N^{(k)}(m, K)$. It also requires that the sample distribution of the target $N^{(k)}$ (v, C) should have specific parameters. In this particular case, v = (0, 0, ..., 0) and matrix C should be the unit matrix (overall 0 and everywhere else on the main diagonal = 1). There are two formal challenges:

- The task of shifting the statistical means of point m to point v;
- The task of transforming the statistical structure, represented by the covariance matrix K, to the target matrix, represented by C.

Solution to the tasks

The first challenge is easily resolved by a shift of the statistical means by the difference vector r.
The second challenge requires the conversion of a statistical sample X into a new Y, that owns the target parameter. The solution is in the form of a linear operator, with the conversion matrix S sought. Every vector Yi is determined in the following manner:

$$y_i = S^T \cdot x_i + r, \qquad i = \overline{1,n} \tag{40}$$

Statistical parameters of sample Y are calculated as follows:
The vector of mean values:

$$\begin{aligned} &\underset{i}{E}\{y_i\} = v = \\ &\underset{i}{E}\{S^T \cdot x_i + r\} = r + \underset{i}{E}\{S^T \cdot x_i\} = \\ &r + S^T \cdot \underset{i}{E}\{x_i\} = r + S^T \cdot m \end{aligned} \tag{41}$$

The difference vector r can be determined using the above expression:

$$r = v - S^T \cdot m \tag{41}$$

TCovariance matrix elements are:

$$
\begin{aligned}
&\underset{i}{E}\{(y_i - v)\cdot(y_i - v)^T\} = C = \\
&= \underset{i}{E}\{(S^T \cdot x_i + r - v)\cdot(S^T \cdot x_i + r - v)^T\} = \\
&= \underset{i}{E}\{[S^T \cdot x_i + r - (r + S^T \cdot m)]\cdot[S^T \cdot x_i + r - (r + S^T \cdot m)]^T = \\
&= \underset{i}{E}\{[S^T \cdot (x_i - m)]\cdot[S^T \cdot (x_i - m)]^{T\}} = \\
&= S^T \cdot \underset{i}{E}\{(x_i - m)\cdot(x_i - m)^T\}\cdot S = \\
&= S^T \cdot K \cdot S
\end{aligned}
\qquad (42)
$$

For the existence of the matrix S, it is necessary that matrices K and C are commutative. This condition is satisfied, because the covariance matrices are symmetrical. The correction implemented for random series uses a SVD algorithm (Singular Value Decomposition) in determining the conversion matrix S.

2) Correction for normal distribution form

The purpose of the form correction is to correct figures of the second order (Kurtosis = 0, Skewness = 0), i.e. the distribution form of the ideal normal distribution should be reached, but the results of the first correction shall be retained.
The following aspects are important in the implementation of this correction:

- The inverse function of a standard normal distribution SND (0.1) is known and represents a strict linear function, i.e. a synthetic linear numerical series could produce an ideal standard normal distribution after the convertion.
- The necessary correlation between random series, which is equal to zero, is given by the sequence of random numbers and through their mutual order in the random series. The sequence of each random number in a random order, with regards to the sequence of all other random numbers in other random series, should be preserved.

From the above stated, an approach follows to the implementation the correction, using the following steps:

- Each time series is converted by the inverse function.
- Converted random numbers should be equally distributed, because of the origin standard normal distribution. Of course, this is not the case, because of the random generator, and the task of correcting the form to ideal equal distribution.
- The random numbers are then sorted, with their original order saved. The new order should follow a linear function. Again, this is not the case; the numbers vary somewhat within the linear function.
- The figures are then overwritten by a synthetic linear function (from 0 to 1) and converted back to normal distribution again, thus enabling the rise of random series with a very good quality.

- As a final step, the original order is restored, in order to reconstruct the initial series correlation.

Obtaining the delta normal distribution vector

$$\overline{B} = |S| \times \overline{D} \quad ,$$

where
S - Cholesky matrix
D - standard normal distribution vector
B - delta normal distribution vector

Calculation of the scenario vector

The scenario vector is obtained by application the delta normal distribution vector on the asset vector, that contains interest rates, foreign exchange rates, etc.

for interest rates:

$$A_i^{Scenario} = A_i * (1 + B_i) \tag{43}$$

for prices and foreign exchange rates

$$A_i^{Scenario} = A_{t_i} \cdot e^{B_t} \tag{44}$$

where:

B_i - Scenario
A_i - Asset

7. Volatility Bridge

7.1 Purpose

The purpose of the Volatility Bridge is generating smaller time interval values (e.g. daily) between the bigger time interval values (e.g. monthly) of a time series in the Monte Carlo simulation.

7.2 Input data

TValues of the time series generated via the simulation scenarios.

7.3 Output data

After the volatility bridge method is applied, the time series has additional values amongst the bigger time interval values.

7.4 Properties

The Monte Carlo simulation, presented in the previous section, generates a smaller number of scenarios over longer horizons. The volatility bridge generates scenarios for arbitrary horizons in between these values, as shown in Fig. 31 [1].

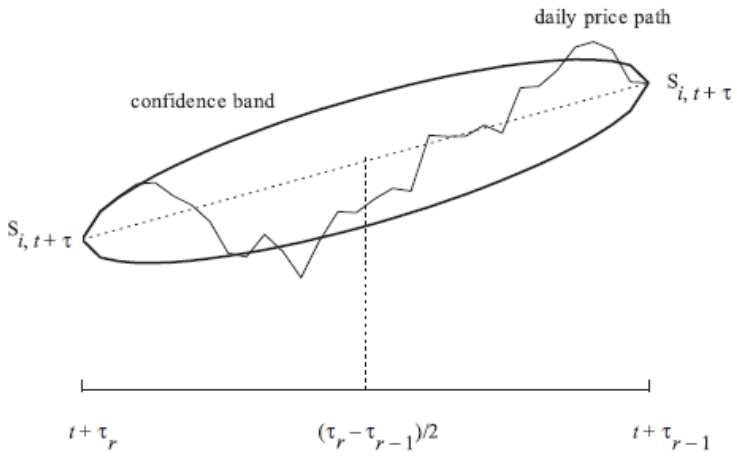


Fig. 31 *Volatility Bridge*

The volatility bridge process generates values with a normal distribution in between two-time points, where at each time point the value is known. Daily price scenarios, generated in this way, are normally distributed.

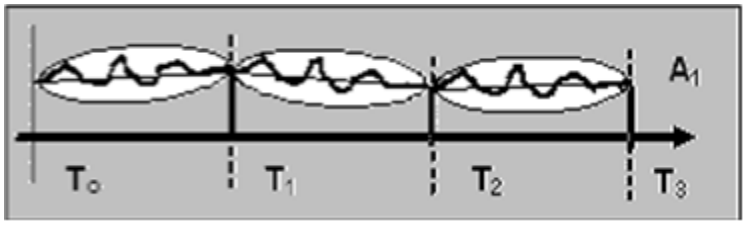


Fig.32 *Volatility Bridge, generating values between several time points*

Generated from Monte Carlo, monthly values represent level 1, while volatility bridge generated values are at level 2 of the simulation. For each simulation run for level 1, the values between time points are recalculated. The level 2 simulation performs a stochastic interpolation between the support points, using the volatility bridge. The following properties are taken into account during this process.

- Expected values between the control points are linearly interpolated.
- Daily fluctuations are generated by a second level Monte Carlo simulation, which generates the daily changes derived from the historical volatility and correlation.

The simulation of scenarios is independent of the time of simulated planned items.

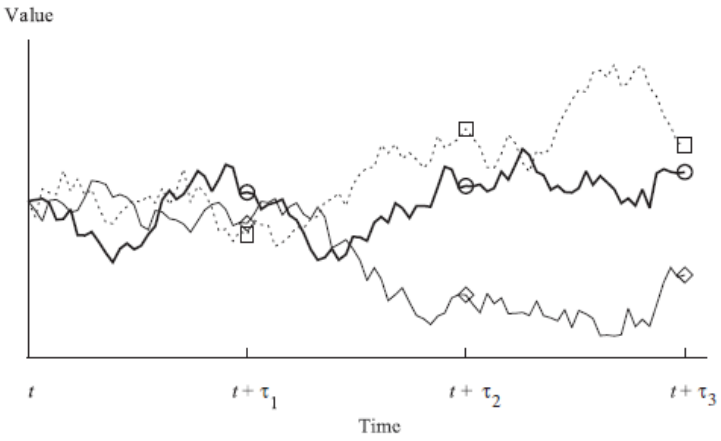


Fig. 33 *Multiply simulated daily price paths with their level 1 and level 2 points*

For b=1, 2, …, τ_1, τ_1 + 1, …, τ_R the daily prices are generated with the following properties:

- At level 1 dates, that is b = τ_1, …,τ_R, level 2 scenarios match level 1 scenarios, so that $d^q_{t+\tau_r} = s^q_{t+\tau_r}$ where $s^q_{t+\tau_r}$ are level 1 prices in scenario q.

- Between level 1 dates, for $\tau_{r-1} < b < \tau_r$, level 2 scenarios are distributed according to the assumptions of daily dynamics of *d*, conditional on *d* passing through $s^q_{t+\tau_{r-1}}$ and $s^q_{t+\tau_r}$.

In order to simulate the volatility bridge process for *d,* the following algorithm has to be used:

1) Independent normally distributed random variables u^q_l for *l* = 1, …,TR are generated with mean 0 and variance σ^2.
2) At level 1 dates (that is, for $b = \tau_r, r = 1,...,R$), set $d^q_{t+\tau_r} = s^q_{t+\tau_r}$
3) For all other dates (that is $b \in (\tau_{r-1}, \tau_r)$ for *r* = 1, …,*R*), set

$$d^q_{t+b} = \frac{\tau_r - b}{\tau_r - \tau_{r-1}} s^q_{t+\tau_{r-1}} + \frac{b - \tau_{r-1}}{\tau_r - \tau_{r-1}} s^q_{t+\tau_r} + \sum_{l=\tau_{r-1}+1}^{b} u^q_l - \frac{b - \tau_{r-1}}{\tau_r - \tau_{r-1}} \sum_{l=\tau_{r-1}+1}^{\tau_r} u^q_l \qquad (45)$$

The algorithm above is also applicable; a random walk is not assumed for the daily process. In general, the u^q_l can be generated according to an arbitrary distribution. This allows the accounting for any special characteristics of daily movements, while maintaining the covariance structure of the volatility bridge.

8. Literature

1. http://www.riskmetrics.com: Long Run Technical Document.
2. Eugene Stern, Risk management for non-financial corporations, RiskMetrics Group JOURNAL Volume 5, Number 1 Winter 2004, pp 42-70.
3. Eshel, G. The Yule Walker Equations for the AR Coefficients, Technical report, Bard College at Simon's Rock, 2009.
4. Hamilton, J. Time Series Analysis. Princeton University Press, ISBN: 0-691-04289-6, 1994.
5. Chatfield, C. The analysis of time series. An introduction. Fifth edition. Chapman & Hall/CRC, ISBN-10: 0412716402, ISBN-13: 978-0412716409, 1996.
6. Box G. E. P., G. M. Jenkins. Time Series Analysis: Forecasting and Control, San Francisco, Holden-Day, 1970.
7. Ji, Y., J. Hao, N. Reyhani, A. Lendasse. Direct and recursive prediction of time series using mutual information selection. Computational Intelligence and Bioinspired Systems: 8th International Workshop on Artificial Neural Networks, IWANN'05, Vilanovai la Geltra, Barcelona, Spain, ser. Lecture Notes in Computer Science, vol. 3512. Springer-Verlag GmbH, 2005, pp. 1010-1017.
8. McNames, J., J. A. K. Suykens, J. Vandewalle. Time Series Prediction. Competition. Internation Journal of Bifurcation and Chaos, Vol. 9, No. 8, 1999, pp. 1485-1500.
9. Palit, A. K., D. Popovic. Computational intelligence in time series forecasting. Theory and engineering applications. Springer-Verlag, 2005.
10. Cameron, C. Trivedi, P. Regression Analysis of Count Data. Cambridge university press, 1998.
11. Draper, N. Smith, H. Applied Regression Analysis. Wiley Series in Probability and Statistics, 1998.
12. Balakrishnan, N., Nevzorov, V. A primer on statistical distributions. John Wiley &Ssons Inc., 2003.
13. Bury, K. Statistical Distributions in engineering. Cambridge University Press, 1999.
14. Krishnamoorthy, K. Handbook of statistical distributions with applications. Chapman & Hall, 2006.
15. Mun, J. Modeling Risk. Applying Monte Carlo Simulation, Real Options Analysis, Forecasting, and Optimization Techniques. Wiley, 2006.
16. R-forge distributions Core Team. A guide on probability distributions. 2009.
17. Ricci, V. Fitting Distributions with R. 2005.
18. Robert, C., Casella, G. Monte Carlo statistical methods. Springer-Verlag, 1999.
19. Vallentin, M. Probability and Statistics Cookbook, 2011.
20. Vose, D. Fitting distributions to data and why you are probably doing it wrong. 2010.
21. Anil K. Jain, Richard C. Dubes. Algorithms for Clustering Data. Prentice-Hall, New Jersey, 1998.
22. Xu, R. Wunsch, C. Clustering. Wiley, 2009.
23. Risk Engine, Valuation Methodology, Version 2015.1.
24. И. Д. Мандель. Кластерныйанализ. Финансы и статистика, Москва, 1988.