Slovak University of Technology in Bratislava
Faculty of Mechanical Engineering
13th Conference on Applied Mathematics **APLIMAT 2014**

Proceedings

’NORTHEAST VOLATILITY WIND’ EFFECT

PUCKOVS Andrejs (LV), MATVEJEVS Andrejs (LV)

Abstract. This paper describes volatility forecasting approach applicable for stock indexes, allowing to reveal the instability of financial time series initially. This approach is based on time series (signal) decomposition into components by using wavelet filtering with subsequent volatility evolution research of each signal component. According to research, a slight increase in volatility in the low-frequency components of the signal leads to significant disturbances in high-frequency components destine entire signal volatility growth.

Keywords. ’North-East Volatility Wind’ Effect, Wavelet filtering, Direct Continuous wavelet transform (Direct CWT), Inverse Continuous wavelet transform (Inverse CWT), Signal Decomposition, Volatility evolution, Disturbances transmission, Time Series, Stock Index

Mathematics Subject Classification: Primary 42C40, 65T60; Secondary 46N30.

1 Introduction

An approach described in abstract is based on ’North-East Volatility Wind’ effect which is describing behavior of stock market indices. An exposure of this allows to predict stock indices volatility bursts, what is is extremely important for investment risks management.

This approach is based on signal decomposition by using the wavelet filtering. Wavelet filtering is applied by using Direct and Inverse CWT for each scaling parameter. Thus for each scaling parameter the signal component (which is part of the original signal) is obtained.

For subsequent research volatility indicator is analysed by using 20-days time window, which is shifted on the time axis. Volatility analysis is done for each signal component. As a result volatility evolution in time is obtained for each signal component.

Since original signal can be reconstructed by summing up all component, volatility of original signal can be obtained by summing up volatility of each component, or, in other words, by summing up all **volatility layers**.

As a next step volatility evolution crosscorrelation analysis is done for each signal component in order to describe volatility transmission from one volatility component to another. In other words volatility transmission from one volatility layer to another is analysed. The process of volatility transmission is similar to disturbulence transmission from one physical object to another.

’North-East Volatility Wind’ effect is described as effect of volatility transmission from low-frequncy components of the signal to high-frequency components. ’North-East Volatility Wind’ effect is illustrated graphically next, it represents a ’wave’ originated by ’North-East Volatility Wind’, which is moving from low-frquency components to a high-frequency components. Respectively opposite effect has a name of ’South-West Volatility Wind’ effect.

’North-East Volatility Wind’ implies volatility increase of original signal and destabilize stock markets, while ’South-West Volatility Wind’ implies volatility decrease and destabilize stock markets. [7]

Unstable stock market periods can be forecasted by tracing volatility growth in low-frquency signal components. Volatility growth forecasting is extremely important for investment risks management.

2 Research algorithm

Here and further, an algorithm proposed in introduction is shown in detail. Research algorithm block-scheme is illustrated in the figure 1. Research includes three main blocks - Data mining and preprocessing block, Signal decomposition block and Signal analysis block. All blocks are considered in subsequent sections in detail. Research algorithm explanation is made by blocks. All operations and operands in each block are explained by using mathematical language and Matlab code. Explanation is extended with example, for which the Dow Jones Industrial Average Index is selected. With this the graphical representation of research algorithm at key-steps of the research is made.

3 Data Mining and Preprocessing Block

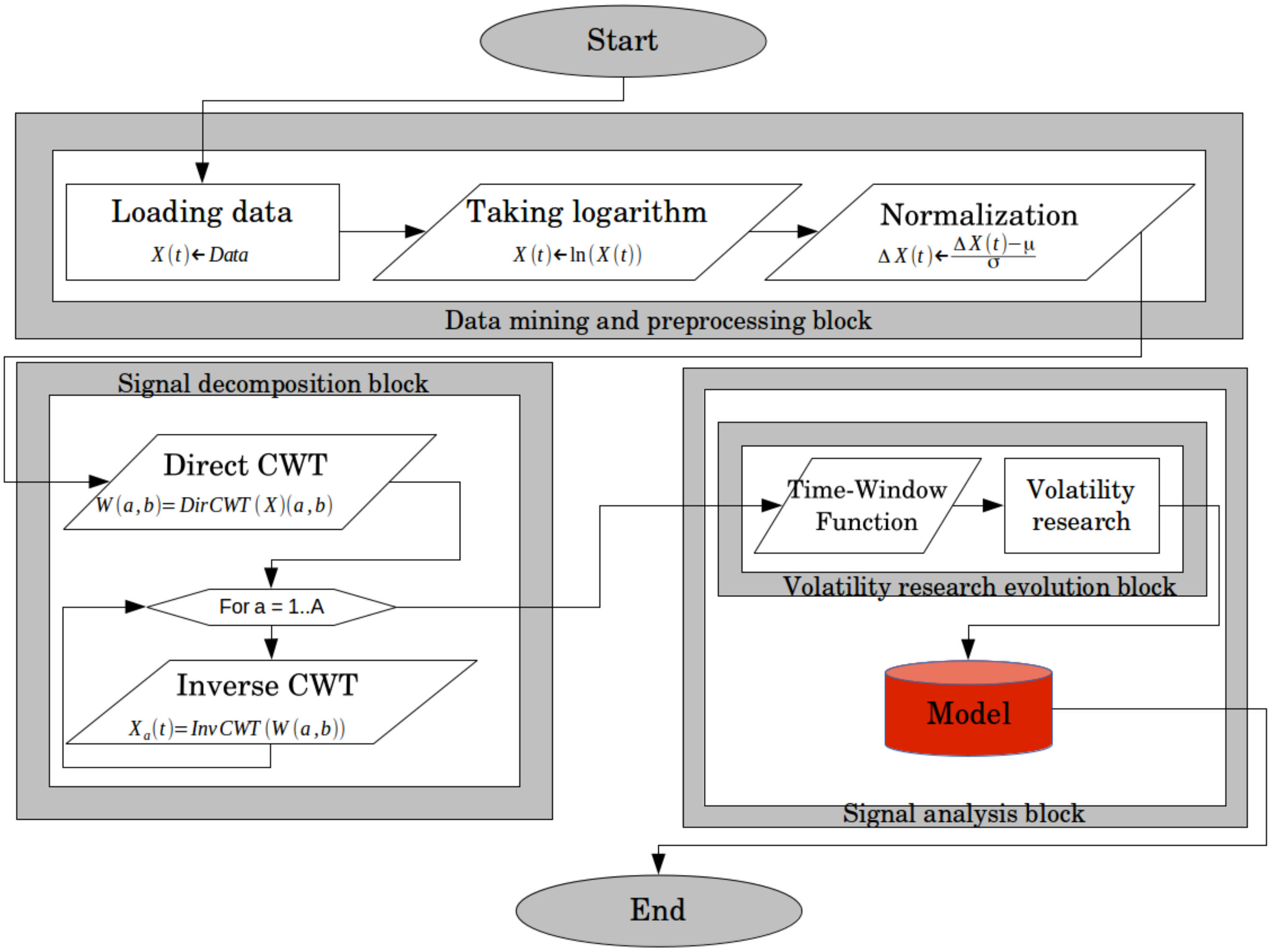
This block intend for financial data mining and preprocessing. The block is placed before Signal decomposition and analysis blocks. Data mining and preprocessing block has three main elements: Data loading operation, Taking logarithm operand and Normalization operand. ’Loading data’ operation is completed in early beginning by using automatic financial time series data download from ’finance.yahoo.com’ source. ’Taking logarithm’ operand is located after ’Loading data’ operation and intend to eliminate the network effect in time series data. ’Normalization’ operand operates after ’Taking logarithm’ operand and intend to normalize financial time series data from stochastic process perspective.

3.1 Loading data

’Loading data’ operation is realized by using automatic data download from ’finance.yahoo.com’ server acting in Matlab enviroment. For this purpose the Matlab function `get_yahoo_stockdata2` is used. The function includes three main parameters: stock index ticket (e.g. for the Dow Jones Industrial Average it is `'DJI'`), two additional parameters showing data sampling start date and end date. In current research stock indexes daily close prices are used.

3.2 Taking logarithm

’Taking logarithm’ operand intend to make power law behavior of financial time series linear. Power law behaviour of financial time series can be explained by a network-effect. In financial mathematics



Obr. 1. Research algorithm block-scheme

taking logarithm operation (with subsequent differentiation operation) has a meaning of Continuous Compound Interest rate calculation. [3] From stochastic process perspective taking logarithm operation has a meaning of transition from Log-normal distribution to the Normal distribution. Without fail 'out-doing' of power law behavior leads us to a better understanding of the financial markets. Mathematical explanation of operand is shown next.

Mathematical Explanation

Stock index additions ΔX are calculated by equation 1.

$$\Delta X(t) \leftarrow ln[X(t+1) - X(t)], \quad t \in [1, T-1]; T \leftarrow T-1. \tag{1}$$

As a result financial time series data are modified, network-effect is eliminated, time series value at time $t = 0$ is zero. For similar activities in Matlab environment the following code is proposed.

Matlab code

```
clear all; clc; close all;
```

```
% Loading data
Data = get_yahoo_stockdata2('^DJI', datenum(2004,1,1),datenum(2013,11,1),'d',true);
Data = Data.Close;
Data = log(Data);
```

This code downloads The Dow Jones Industrial Average Index data from finace.yahoo.com server for specified period of time and takes logarithm of index values.

3.3 Normalization

Normalization operand is optional. Operand is placed next to 'Taking logarithm' operand and operates with logarithmic stock index additions. This operand normalizes stock index additions from stochastic process perspective. Normalization operand provides stochastic process N(0,1) in the output. Stochastic process with mathematical expectation $\mu = 0$ and standard deviation $\sigma = 1$ is familiar, and in some sense, it is 'basic' in stochastic analysis, for example a Random Walk $\Delta x = \pm 1$ and Classical Brownian motion has the same properties. [9] Elimination of average and conversion of stock index additions standard deviation to $\sigma = 1$ in current research has a meaning of switch from the stock index stochastic process with specific properties to generalized stochastic process. Mathematical explanation is shown next.

Mathematical explanation
Normalization operand provides normalized stock index additions ΔX with mathematical expectation $\mu = 0$ and standard deviation $\sigma = 1$ in the output. See equation 2.

$$\begin{aligned} \Delta X(t) \leftarrow \frac{(\Delta X(t) - \mu)}{\sigma}, \\ \mu = T^{-1} \cdot \sum_{t=1}^{T} \Delta X(t), \\ \sigma^2 = T^{-1} \cdot \sum_{t=1}^{T} (\Delta X(t) - \mu)^2 \end{aligned} \tag{2}$$

In current operation first of all, properties of stock index additions (mathematical expectation μ and standard deviation σ) are calculated. Then normalization operation $(\Delta X(t) - \mu)/\sigma$ is implemented. After that all new variables of normalized stock index additions are assigned to $\Delta X(t)$. At the end by summing up variables $\Delta X(t)$ the normalized stochastic process $X(t)$ is obtained and its variables are assigned to $X(t)$.

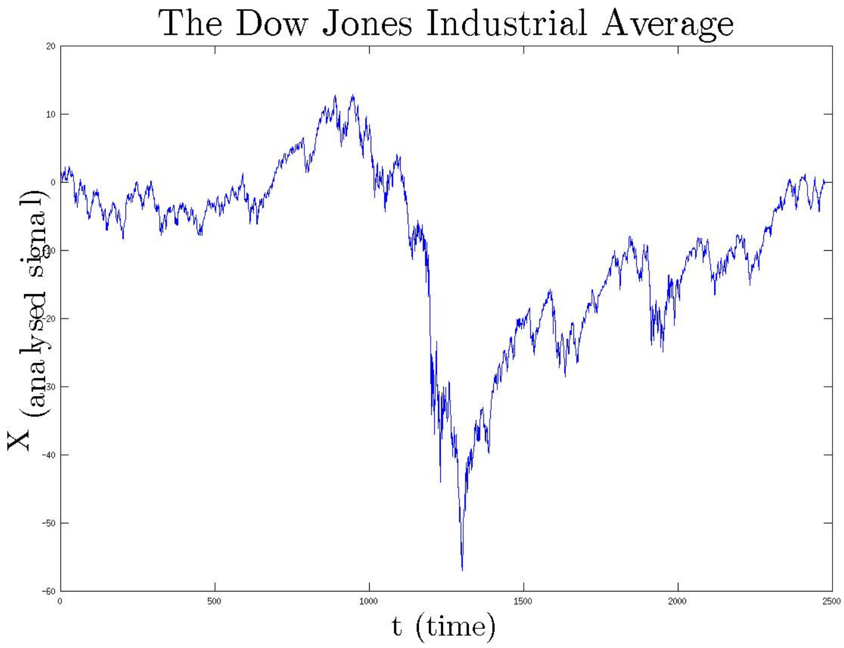
$$X(t) \leftarrow \sum_{i=1}^{t} \Delta X(i). \tag{3}$$

Normalization operand provides normalized stock index with properties $N(0,1)$ which also is the output of Data analysis and preprocessing block and input for Signal decomposition block. From this point $X(t)$ has the name of analysed signal. For similar activities in Matlab environment the following code is proposed.
Matlab code

```
% Signal preprocessing
```

```
 wn = diff(Data);
% Normalization
 wn = (wn - mean(wn))/std(wn);
% Time
 T = length(wn); t = 1:T;
% Create signal
  X = cumsum(wn);
```

This code normalize The Dow Jones Industrial Average Index additions and provides analysed signal in the output, which is also illustrated graphically in the figure 2.



Obr. 2. Analysed signal - The Dow Jones Industrial Average

Since the purpose of research is stock indices Volatility evolution research, describing 'North-East Volatility Wind' effect, volatility of analysed signal 'parts' to be discovered next. In order to decompose analysed signal in parts a wavelet filtering is realized in the next block. [4]

4 Signal decomposition block

Signal decomposition block is placed between Data mining and preprocessing block and Signal analysis block. The block consist of two parts: Direct Continuous Wavelet Transform (Direct CWT) and Inverse Continuous Wavelet Transform (Inverse CWT) operands. Since analyzed signal is decomposed in parts, Direct CWT and Inverse CWT operands are working in loop, providing decomposed parts of analyzed signal for each scaling parameter a. In the output Signal decomposition block provides decomposed parts of the signal for each scaling parameter a, in fact, these are a components of (original) analyzed signal, which are analyzed separately in the Signal analysis block.
Signal decomposition by using wavelet transform is keeping following idea: the number of signal components depends only on scaling parameter a and by summing all components of decomposed signal original (analysed) signal can be reconstructed. [4] Analogically Volatility of analysed signal

can be reconstructed by summing all signal components volatilities. Since the object of current research is a volatility evolution, volatility evolution on particular decomposed parts of the signal to be analysed in subsequent sections, after Signal decomposition procedure is cleared.

4.1 Direct Continuous Wavelet Transform operand

Direct CWT operand operates on analysed signal, transforming it from signal, defined in time domain, to the signal, defined in the 'wavelet domain'. In some sense, 'wavelet domain' operates in time and frequency domain simultaneously. In fact, Direct CWT constructs a Wavelet Image of analysed signal defined by scaling parameter (which has serious analogy with frequency) and shift parameter (which has serious analogy with time). [1] The Wavelet Transform operation by itself is a dot-product operation of analysed signal and (shifted and scaled) mother wavelet function. This operation illuminates part of the signal of particular frequency in particular interval of time. Mother wavelet scaling operation has a meaning of selection of analysed signal in defined frequency band, while mother wavelet shift operation has a meaning of selection of analysed signal in defined time interval. Mathematical explanation of operand is shown next.

Mathematical explanation
Direct CWT operand takes analysed signal $X(t)$ in input and provides the Wavelet Image $W(a,b)$ in output. The Wavelet image of analysed signal $W(a,b)$ is defined in wavelet domain by scaling parameter a and shift parameter b, while analysed signal $X(t)$ is defined in time domain by time parameter t. [6]Direct CWT is the dot-product operation of analysed signal $X(t)$ and shifted and scaled mother wavelet function $\psi(\frac{t-b}{a})$. [10]

$$W(a,b) = a^{-\frac{1}{2}} \cdot \int_0^T X(t) \cdot \psi(\frac{t-b}{a})dt, \qquad a \in (1, A), A \leq T/2 \tag{4}$$

Wavelet image of analysed signal, provided by Direct CWT operand in output, is used as input by Inverse CWT operand. As the signal decomposition is provided for each scaling parameter a, the signal decomposition process can be prioivided in loop, for each scaling parameter a. Such activities in Matlab environment can be realized by following code is proposed.
Matlab code

```
% Wavelet analysis

% Define parameters before analysis
dt = 1;
maxsca = floor(T); s0 = 2*dt; ds = 2*dt;
scales = s0:ds:maxsca;
wname = 'morl';
SIG = {X,dt};
WAV = {wname,[]};

% Compute the CWT using cwtft with linear scales
cwtS = cwtft(SIG, 'scales', scales, 'wavelet', WAV);
```

In Matlab environment Direct CWT operand is realized by using `cwtft` function, which, in fact, is operating not with analysed signal and scaled and shifted mother wavelet function but with their Fourier images. [4] But for algorithm exploration it is not terribly important. In the output a Wavelet image of analysed signal or let's call Wavelet Coefficients Matrix (since Matlab operates with matrices) is obtained.

4.2 Inverse Continuous Wavelet Transform operand

Inverse CWT operand operates with a Wavelet Image of analysed signal. By its nature Inverse CWT operand reconstructs original signal from its Wavelet Image. Since whole signal can be obtained by the 'whole' Wavelet Image, the signal component can be obtained from part of its Wavelet Image. In current research authors are interested to focus not on the whole signal, but on signal components, since volatility research is made on particular signal components.

Mathematical explanation
Inverse CWT operand reconstructs parts of original signal $X(a,t)$ from specified Wavelet Image parts $W(a,b)$ by using equation 5.

$$\forall a : X(a,t) = C^{-1}a^{-2} \cdot \int_0^B W(a,b) \cdot \psi_{ab}(t)db \tag{5}$$

In equation 5 C is the constant, dependent only on mother wavelet function ψ. [11]Inverse CWT operand accomplishes original signal $X(t)$ decomposition on particular components $X(a,t)$, which are used for volatility research in the Signal analysis block. Since signal decomposition equation 5 is provided, signal recovery or perfect reconstruction equation 6 can be provided as well. [5]

$$\hat{X}(t) = \int_0^A X(a,t)da \tag{6}$$

In Matlab environment Inverse CWT operand is realized by using following code.
Matlab code

```
% Compute inverse CWT using linear scales
X2 = icwtlin(cwtS,'Signal',X);
%Signal decomposition
Im = cwtS;
WF = Im.cfs;
[A B] = size(WF);

W0 = zeros(A,B);
for a = 1:A
    Wmod = W0;    Wmod(a,:) = WF(a,:);
    Im.cfs = Wmod;
    Arg = icwtlin(Im);
    Xpart(a,:) = Arg- mean(Arg);
clear Arg
end
clear Im
```

```
% Adding constant for perfect reconstruction
X4 = sum(Xpart);
C = mean(X-X4');
for a = 1:A
    Xpart(a,:) = C/A + Xpart(a,:);
end
X4 = sum(Xpart);
```

This Matlab code accomplishes signal decomposition by using Wavelet Coefficient Matrix. Signal part recovery from Wavelet Coefficient Matrix is made by using Matlab function `icwtlin`. An abbreviation 'lin' means linear scales for Wavelet Coefficient Matrix. Should say, Matlab does not check whether scales are linear. In current research linear scales are chosen, but in CWT for 'log' scales can also be used.
Provided code generates decomposed parts of the signal for each scaling parameter a. Decomposed parts of the signal named `Xpart` are used in subsequent volatility analysis.

5 Signal analysis block

This block brings light on volatility evolution research. The block is placed after Signal decomposition block and it works with decomposed parts of analysed signal. Signal decomposition, made in previous step, has divided original signal in components by using wavelet filtration. The purpose of current research is volatility analysis and volatility evolution analysis. Starting current section it is important to recapitulate why volatility analysis is important for this research.
From financial perspective volatility is a measure of risk. The risk by itself has no sense-feeling.The meaning of volatility depends on investment strategy. But most investors relate volatility with some degree of 'awareness', because most of them are keeping long positions. For them it is very important to know the volatility to manage risks and understand markets by measuring volatility.

In order to understand volatility and its evolution in detail, authors propose to consider volatility on 'different level of abstraction' as the Wavelet Transform is considering the signal on 'different level of abstraction'. For this purpose volatility is measured on different signal components; the signal decomposition is made by using Direct and Inverse CWT with integer scales, which are located closely.

But there is one an issue: stock indexes lower scales (high-frequency components) components volatility is incomparably greater than volatility for components at higher scales (low-frequency components).This situation is observed because the energy of stock indexes signal is concentrated on high-frequency components. [12] [2]For that reason authors propose to use logarithm of volatility as the risk indcator. The switch from volatility indicator to volatility logarithm brings serious advantage - volatility indicators at different scales become comparable.

The second issue to be solved is a choice of volatility argument - time. Since volatility evolution is the research object, volatility argument - time (or let's call time-window for volatility estimation) should be selected short enough to illuminate changes in volatility indicator. In current research 20-days volatility logarithm indicator is selected. Calculations of this indicator is provided in the next subsection.

5.1 Volatility research subblock

Volatility research subblock use decomposed parts of analysed signal in input and provides volatility indicator in output. Volatility research subblock consist of two parts: time-window function (which selects 20-days data and shifts time-window to the end) and volatility analysis operand (which is measuring volatility indicator of selected data). Volatility research subblock is measuring volatility for each part of decomposed signal and in each time-window. Mathematical explanation for Volatility research subblock operations is provided next.

Mathematical explanation

Time-Window operand operates on parts of decomposed signal $X(a,t)$ and selects 20-days window interval $Win_\tau(a,t)$ by using equation 7.

$$\begin{gathered} \Delta X(a,t) \leftarrow X(a,t+1) - X(a,t), \\ Win_\tau(a,t) \leftarrow (\Delta X(a,t) \cup \Delta X(a,t+1) \cup ... \cup \Delta X(a,t+\tau)), \\ \forall a, \forall t | (t+\tau \leq T), \tau = 20. \end{gathered} \tag{7}$$

Volatility analysis operand use output of Time-Window operand in input and provides 20-days volatility indicator for each part of decomposed signal and in each time-window by using 8.

$$\begin{gathered} D_\tau(a,t) = \tau^{-1} \cdot \sum_{t=1}^{\tau} (Win_\tau(a,t) - \mu(a,t))^2, \\ \mu(a,t) = \tau^{-1} \cdot \sum_{t=1}^{\tau} (Win_\tau(a,t)). \end{gathered} \tag{8}$$

Equation 8 explains 20-days (τ days) volatility calculation [9], but as it is discussed before, instead of volatility, logarithmic volatility should be calculated by using equation 9.

$$D_\tau(a,t) \leftarrow ln(D_\tau(a,t)). \tag{9}$$

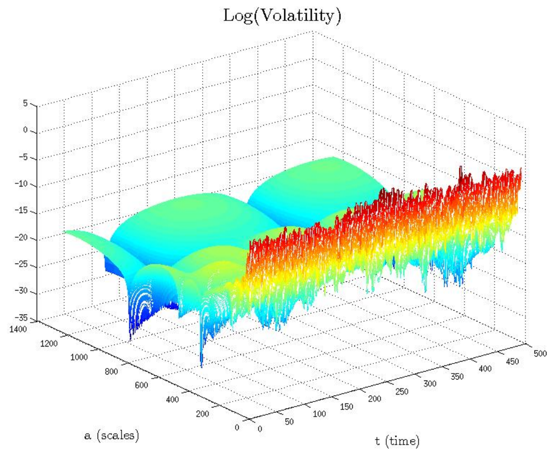
Equation 9 explains changes in variables when $ln(D_\tau(a,t))$ variables are assigned to $D_\tau(a,t)$. Let (or assign) operator was implemented also in previous equations, that was made for the reason of economy in order to not use too much variables. For similar activities in Matlab environment the following code is proposed.

Matlab code

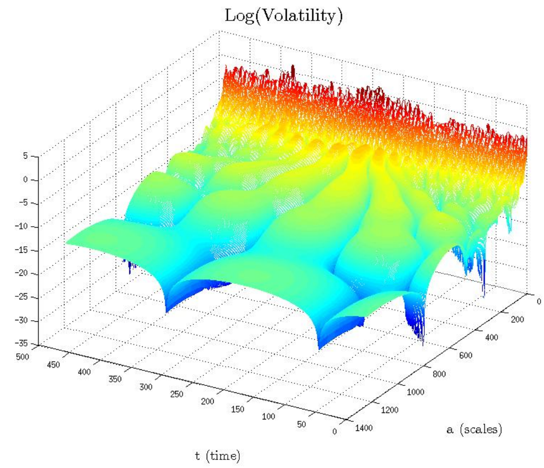
```
for a = 1:A
    st = 1; win = 20; del = 1;
    en = st + win;
    bmod = floor(win/del)+1;
    while (en < B)
        D_a(a,bmod) = log(std(Xpart(a,st:en)));

        st = st + del;
        en = st + win;
        bmod = bmod +1;
    end
end
```

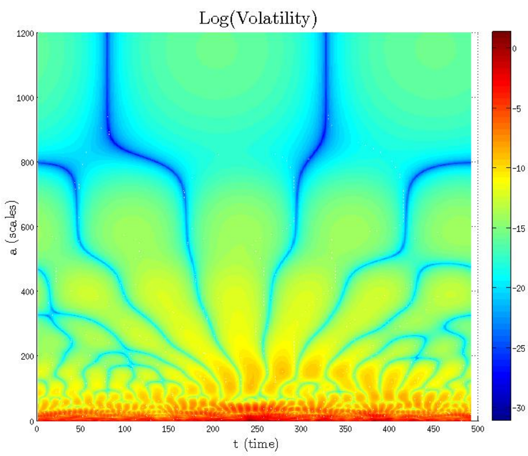
Current code use `Xpart` variable (whcih is calculated in previous step) as input, and provides `D_a` variable in output. `D_a` variable abbreviated $D_\tau(a,t)$ in mathematical explanation and in fact this is a matrix, which contains volatility variables for each scaling parameter a for each time-window $Win_\tau(a,t)$. One remark should be made here, for machine-time economy the variable `del=1` should be increased in order to reduce the number of time-windows. In current research `del=5` is used. The next figure illustrates the `D_a` variable (20-days logarithmic volatility indicator) for The Dow Jones Industrial Average for all parts of decomposed signal for analysed period of time.



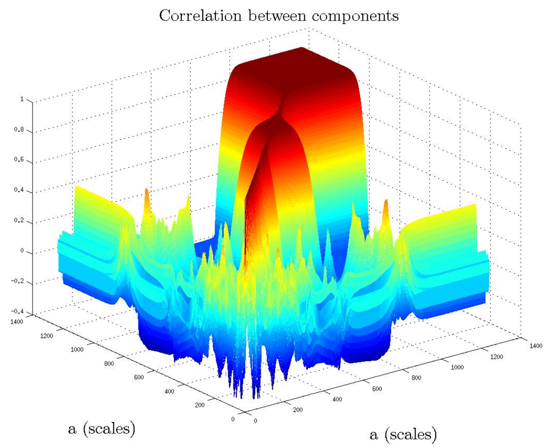
(a) Volatility indicator of the he Dow Jones Industrial Average(1 point of view)



(b) Volatility indicator of the The Dow Jones Industrial Average(2 point of view)



(c) Volatility indicator of the The Dow Jones Industrial Average(top-position view)



(d) Correlation between volatility layers of The Dow Jones Industrial Average

Obr. 3. Volatility indicators of the The Dow Jones Industrial Average and correlation between Volatility Layers

5.2 Volatility analysis and 'North-East Volatility Wind' Effect

Calculated logarithmic volatility indicator provides the picture of volatility evolution in each component of analysed signal or let's call in each **Volatility Layer**. It is important to discover not only volatility evolution but also complicated interdependences between Volatility Layers. As it is shown

in 3d there is a strong correlation between some Volatility Layers. If we look at figure 3b the volatility indicator at higher scales (low-frequency) components is very small, in the the same time volatility indicator at lower scale (high-frequency) components is higher. One interesting effect should be pointed out, disturbances apear on higher volatility layers and eventually go to the lower volatility layers. When disturbances come to the lowest volatility layers, where energy of signal is high, a very conspicuous overall-signal volatility burst is happening. After conspicuous volatility burst, disturbances are rolling back to the highest volatility layers. This is extremely important effect, which is useful for volatility forecasting.
There is some analogy between volatility transmission between layers and ocean wave. The wave far from the cost line is hardly noticeable, but when it comes to the cost line it becomes very sharp and noticeable. The same thing is happening in the stock markets, when hardly noticeable volatility wave comes to the lower scale (high-frequency) components. This effect is called 'North-East Volatility Wind' effect. The name of the effect explained by direction of volatility transmission between layers in time. This effect is explained in details next.
It is important to discover evolution of volatility in time. In the same time it is important to discover relations between volatility layers. For volatility evolution research in time in each layer the one research model is proposed, which is explained next.

Mathematical explaination
The difference between volatilities $\Delta D_\tau(a, a+\Delta a, t)$ on each volatility layer in each time moment is calculated in equation 10.

$$\begin{aligned} \Delta D_\tau(a, a+\Delta a, t) = D_\tau(a, t+1) - D_\tau(a+\Delta a, t), & \\ \forall a, \forall \Delta a|(a+\Delta a \leq A), & \\ \forall t|(t+1 \leq T) & \end{aligned} \tag{10}$$

As the result a 3D matrix ΔD_τ is acquired. To illustrate this matrix, authors use 3D plot evolute in time (actually this is a movie). But in order to understand the dynamics of ΔD_τ and to illustrate the 'North-East Volatility Wind' effect, the f ΔD_τ matrix is shown at sampled moments of time. See the figure 4 for illustration.
As it is shown in figure 4, as volatility wave goes from higher volatility layers (see figure 4a) towards the lower volatility layers 4f), overall-volatility (or volatility of analysed signal in certain time-window) is rising dramatically. See corespondent figures 5a - 6c in order to trace overall-volatility evolution. When volatility wave reaches the lowest volatility layer 4f, overall-volatility demonstrates the burst 6c. As it is mentioned before, the volatility transmission between layers are analogical to ocean wave roll: by reaching the cost line, ocean wave becomes sharp and noticeable; as ocean wave is rolling back after the slide on coast line, volatility wave is moving from the lowest volatility layer to the higher volatility layers (where it becomes less noticeable). The wind which is blowing from the 'sea side' (which is moving volatility towards the lowest volatility layers) is called 'North-East Volatility Wind'. The wind which is blowing from the 'coast side' (which is moving volatility towards the highest volatility layers) is called South-West Volatility Wind. The direction of wind is explained by direction of wave relatively to scaling axis (x axis correspond to a, while y axis correspond to $a + \Delta a$). But direction of wind also depends on direction of coordinate system.
The main conclusion to be made in paper is following - **a slight increase in volatility in the low-frequency components of the signal leads to significant disturbances in high-frequency components destine entire signal volatility growth. This effect is called 'North-East Volatility Wind'**

Effect. This conclusion is very important for risk management on the stock markets. 'North-East Volatility Wind' Effect described in current paper brings out deeper understanding of volatility evolution and opportunity to illuminate most dramatical market drawdowns initially. This opportunity is explained by ability to see a very small changes in volatility logarithm in the low-frequency components of the signal.

6 Future work

In future research 'North-East Volatility' Wind effect expected to be formalized and complicated volatility evolution and complicated relationships between volatility layers to be discovered next. For that reason Genetic Algorithms and Neural Networks expected to be used for hidden relationships discovery. [8]

References

[1] BASU, T. K. *Lecture Series on Digital Signal Processing*. IIT Kharagpur, 2013, 555 s.

[2] BURNAEV *Primenenije vejvlet preobrazovanija dla analiza signalov*. MFTI, 2007, ISBN 5-93517-103-1, 138 s.

[3] CAPINSKI, M., T., Z. *Mathematics for Finance An Introduction to Financial Engineering*. Springer, 2010, 309 s.

[4] GADRE, V. M. *Lecture series in Multirate Digital Signal Processing and Multiresolution analysis*. IIT Bombay, 2013, 555 s.

[5] JAKOVLEV, A. N. *Vvedenije v vejvlet preobrazovanija*. NGTU, 2003, ISBN 5-7782-0405-1, 576 s.

[6] PUCKOVS, A., A, M. Wavelet analysis in Wavelet Transform Modulus Maxima Approach. *Information Technology and Management Science. Vol.15*, ročník 15, č. 15, 2012: s. 76–86, ISSN 22559086.

[7] PUCKOVS, A., A, M. Equity Indexes Analysis and Synthesis by using Wavelet Transforms. In *CFE 2013 7-th International Conference on Computational and Financial Econometrics*, London: Birkbeck University of London, 2013, ISBN 978-84-937822-3-8, str. 98.

[8] RUTKOVSKAJA, P. M., D., RUTKOVSKIJ, L. *Nejronnije seti, Geneticheskije algoritmi i Nechetkije sistemi*. Gorachaja Linija - Telekom, 2006, ISBN 5-93517-103-1, 383 s.

[9] SHIRAEV, A. *Osnovi stohasticheskoj i finansovoj matematiki*. Fathis, 2003, ISBN 5-7036-0043-ÐL', 608 s.

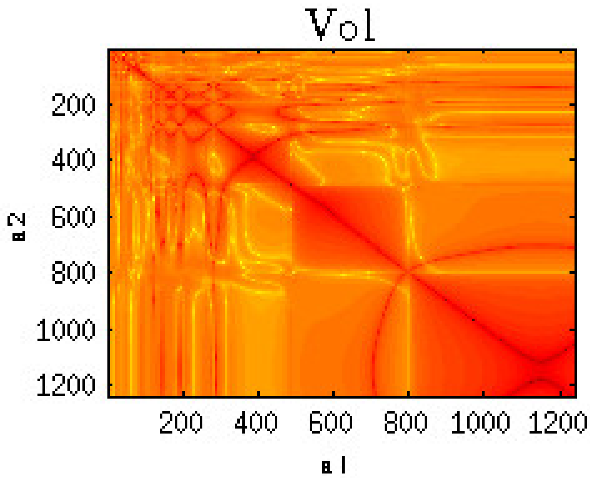
[10] SHOLUHIN, O. I. *Multifraktali. Infokomunikacionnije prilozhenija*. Gorachaja Linija - Telekom, 2010, ISBN 978-5-9912-0142-1, 576 s.

[11] SMOLENCEV *Osnovi Teoriii Vejvletov v MATLAB*. DMK Press, 2005, ISBN 5-94074-122-3, 304 s.

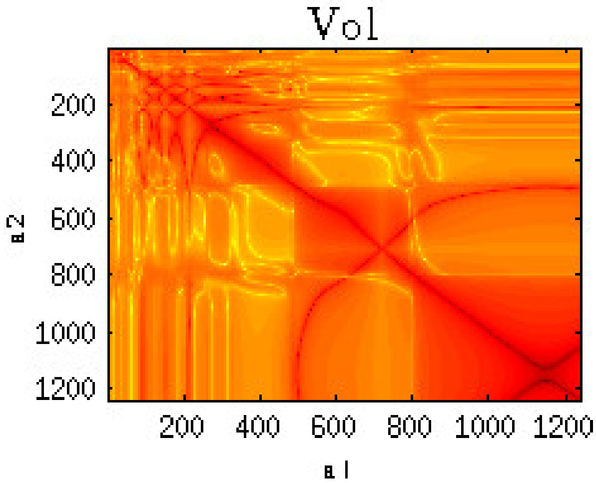
[12] TREFETHEN, L. N. *Spectral Methods in MATLAB*. Comlab, 2007, 160 s.

Riga Technical university, 1 Kalku iela, Riga, Latvia

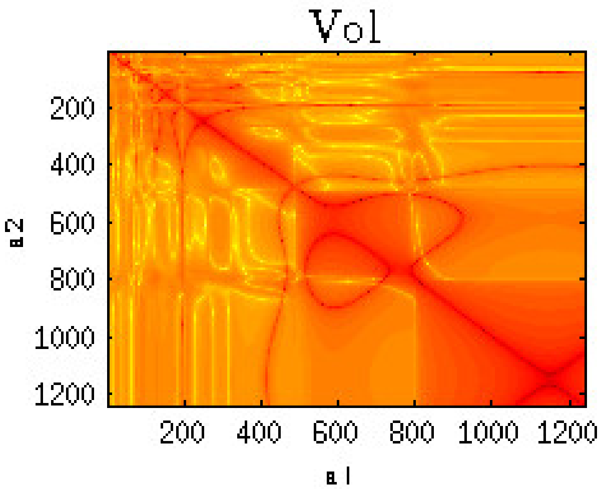
Andrejs Puckovs (PhD Student), Andrejs Matvejevs (Dr.sc.ing.)
The Chair of the Probability Theory and Mathematical Statistics
Riga Technical University
1 Kalku iela, Riga, Latvia



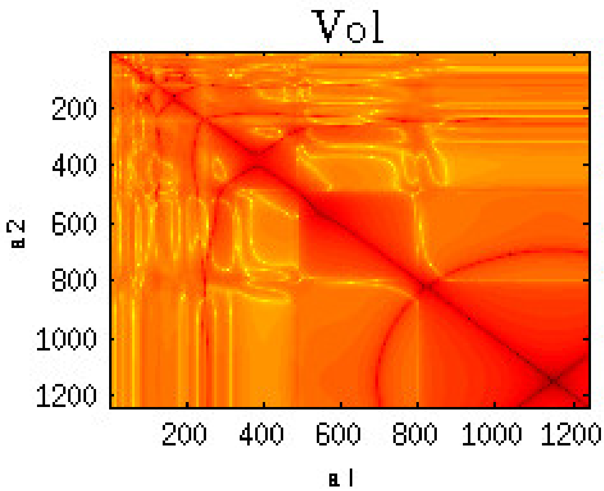
(a) 'North-East Volatility Wind' in moment t=140)



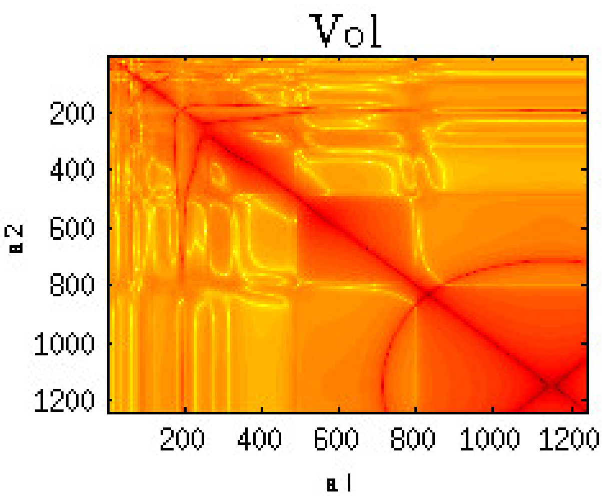
(b) 'North-East Volatility Wind' in moment t=160



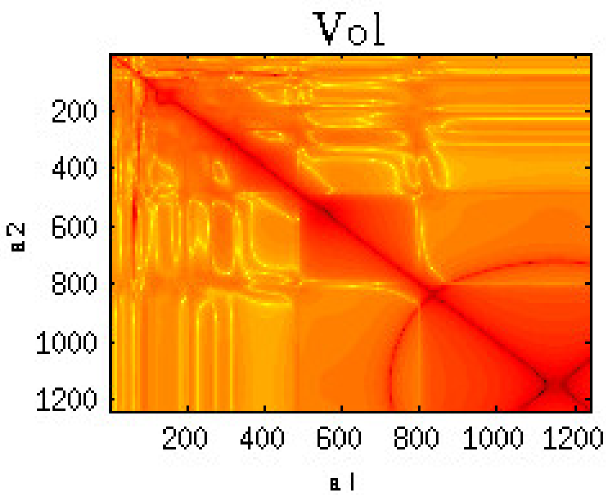
(c) 'North-East Volatility Wind' in moment t=180



(d) 'North-East Volatility Wind' in moment t=205

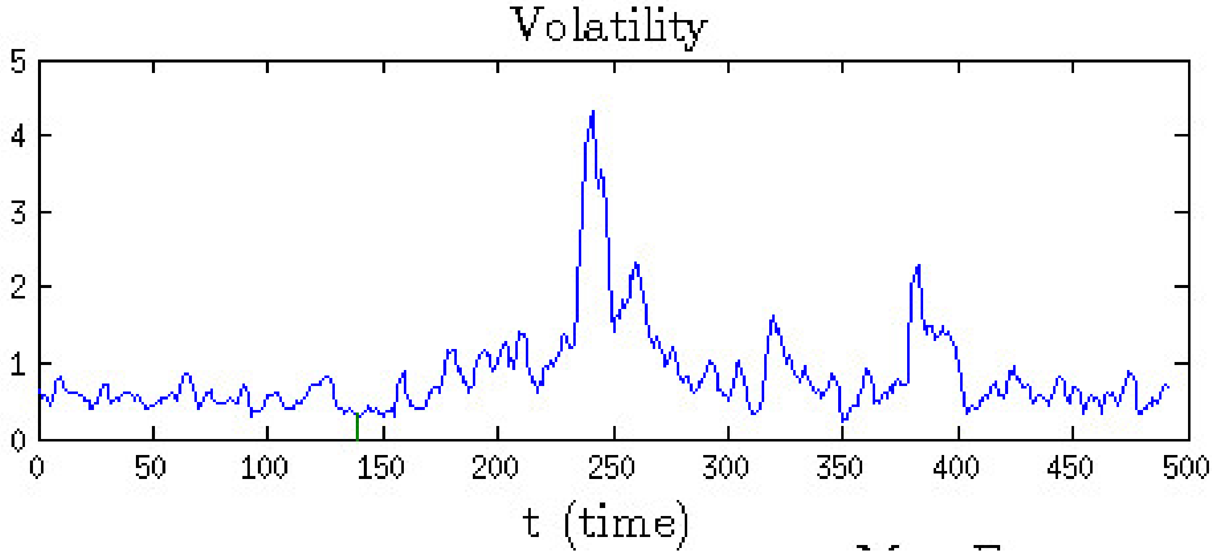


(e) 'North-East Volatility Wind' in moment t=215 (moment before volatility peak)

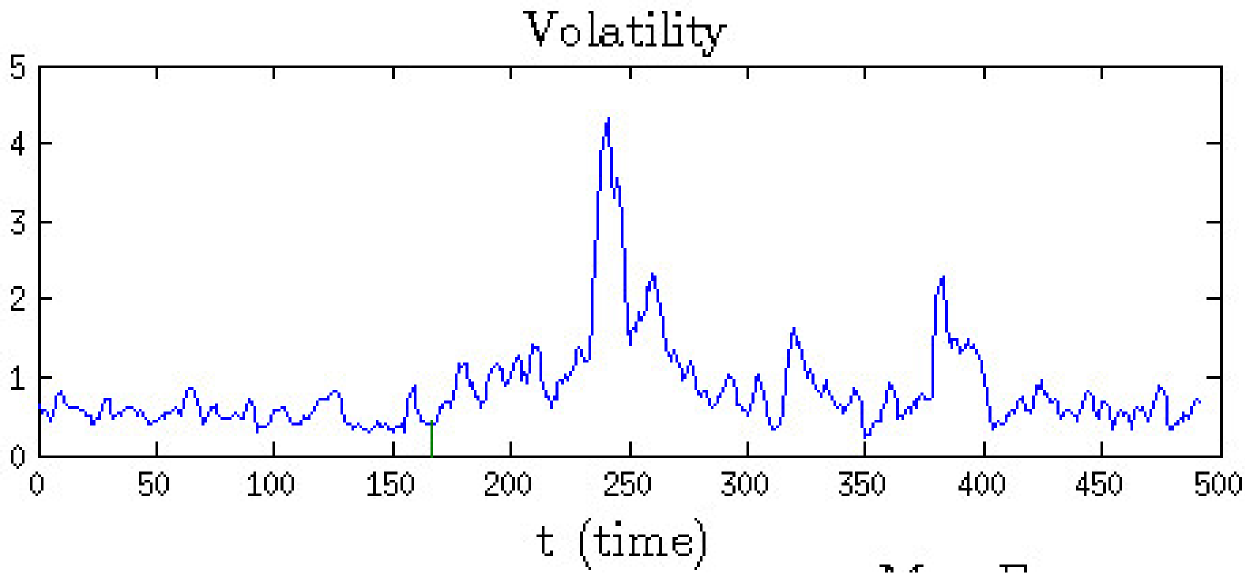


(f) 'North-East Volatility Wind' in moment t=240 (volatility peak)

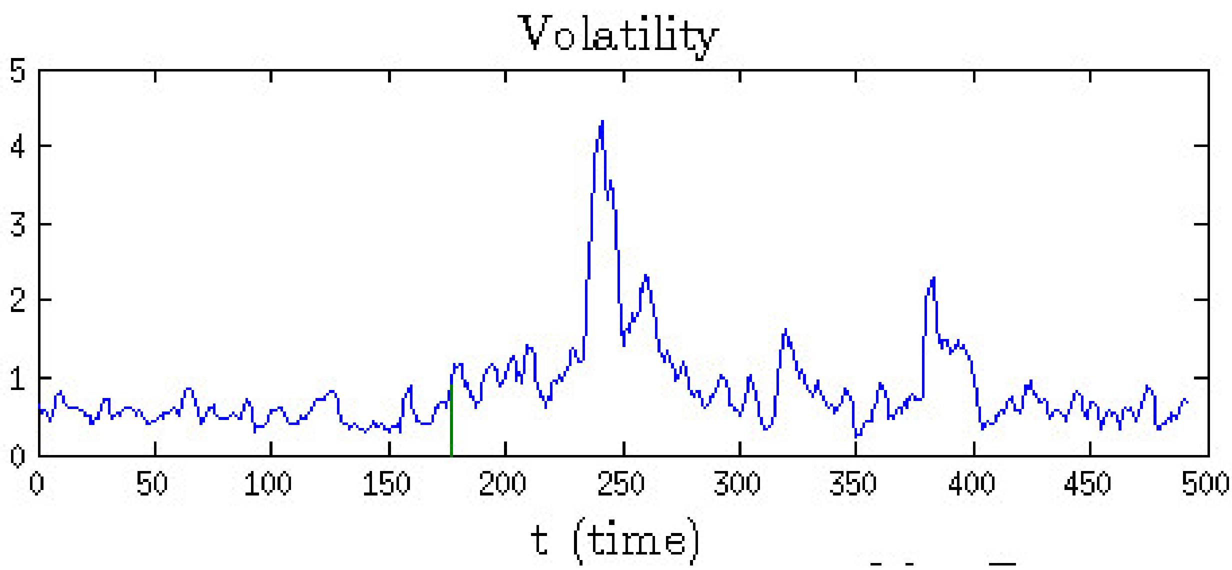
Obr. 4. 'North-East Volatility Wind' Effect in different moments of time



(a) Overall-volatility in moment t=140)

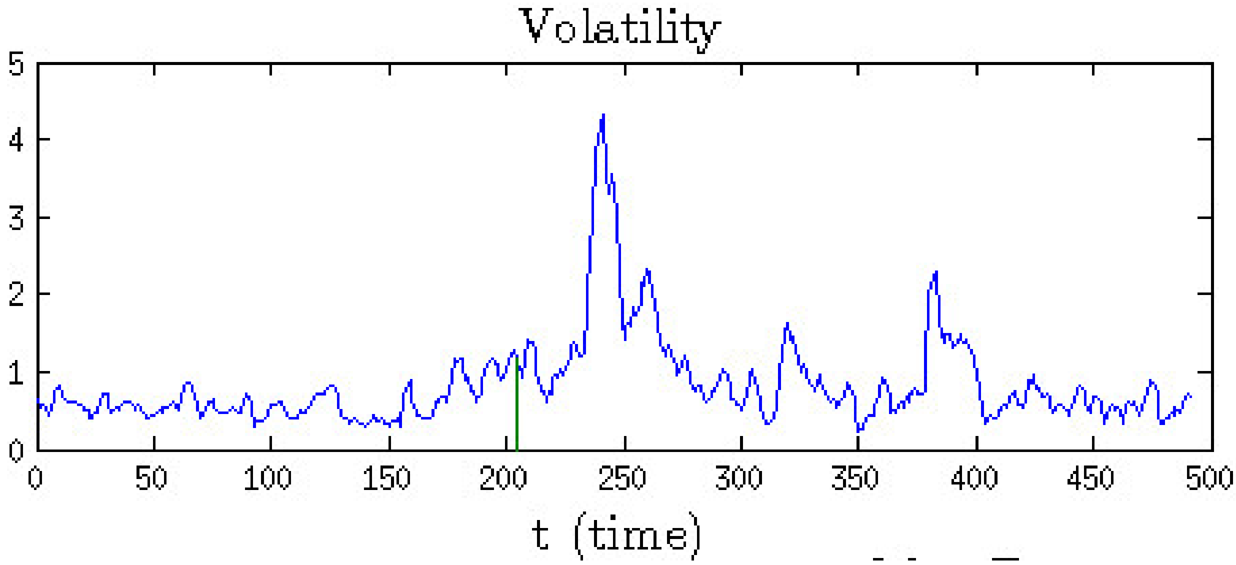


(b) Overall-volatility in moment t=160

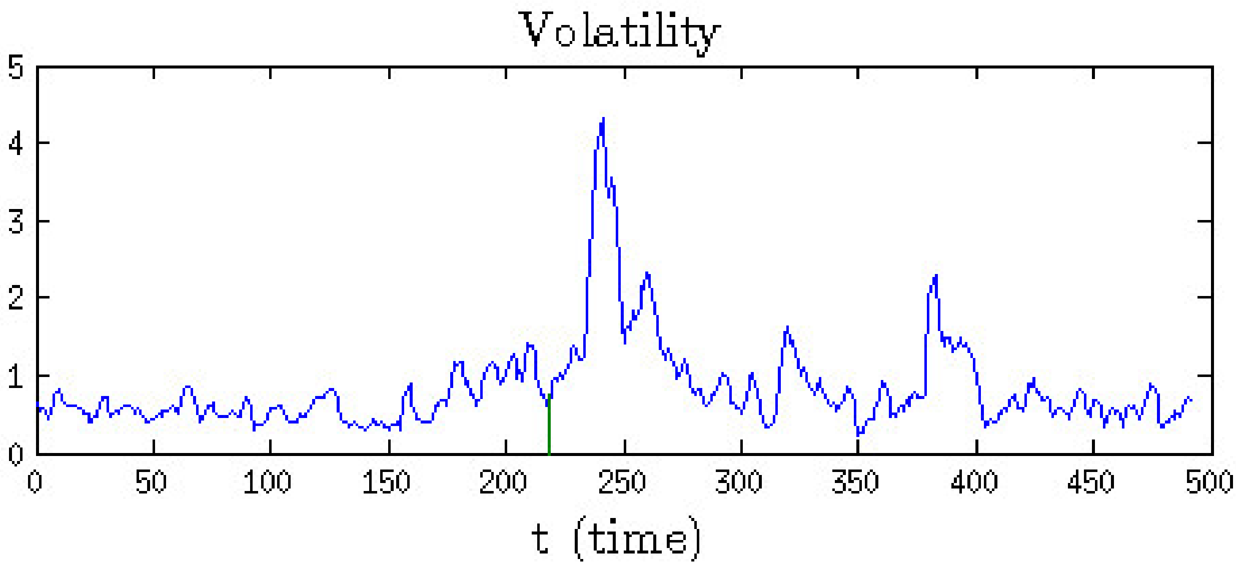


(c) Overall-volatility in moment t=180

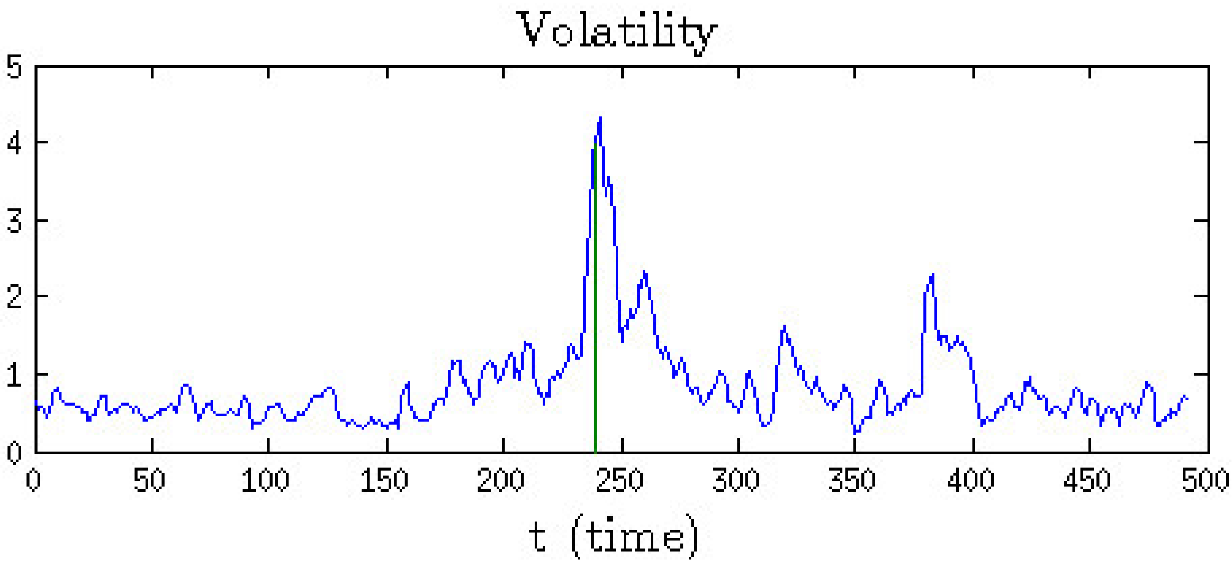
Obr. 5. Overall-volatility in different moments of time



(a) Overall-volatility in moment t=205



(b) Overall-volatility in moment t=215 (before crisis)



(c) Overall-volatility in moment t=240 (crisis peak)

Obr. 6. Overall-volatility in different moments of time