Journal of Business Economics and Management
2007, Vol VIII, No 3, 165–168
www.jbem.lt

ISSN 1611-1699

APPLICATION OF MONTE CARLO SIMULATION METHODS IN RISK MANAGEMENT

Alexander Suhobokov

Head of Market Risk Management Division, Parex Bank,
Smilšu iela 3, LV-1522 Riga, Latvia
E-mail: Aleksandrs.Suhobokovs@parex.lv

Received 4 April 2007; accepted 15 June 2007

Abstract. The paper deals with Monte Carlo simulation method and its application in Risk Management. The author with the help of MATLAB 7.0 introduces new modification of Monte Carlo algorithm aimed at fast and effective calculation of financial organization's Value at Risk (VaR) by the example of Parex Bank's FOREX exposure.

Keywords: market risk management, Monte Carlo simulation, Value at Risk (VaR).

1. Introduction

During the last 20 years Value at Risk (VaR) has become one of the most important tools in the science of Risk Management. The major reason for that is the ability of VaR to provide a precise quantitative measure of downside risk. By the moment the scientific world has produced tens of different approaches to VaR calculation. In practice, the objective of any financial organization is to pick one approach that would provide a reasonably accurate estimate of risk at reasonable costs.

2. Approaches to VaR

In general, approaches to VaR can be split into two major categories [1]. The first category uses local methods for VaR calculations. ***Local-valuation methods*** measure risk by valuing the portfolio once, at the initial position, and then using local derivatives to forecast possible movements. The classic example of methods in this category is the delta normal approach developed by investment bank J. P. Morgan [2]. This method uses linear derivatives and assumes normal distributions of price changes. The second major category for VaR assessment implies full valuation of risk factors. ***Full-valuation methods*** measure risk by fully repricing the portfolio over a range of scenarios. These methods are generally represented by two key approaches: Historical Simulation approach and Monte Carlo Simulation approach. The historical simulation method provides a straightforward implementation of full valuation. It takes a portfolio of assets at a particular point in time and then revalues the portfolio a number of times, using the history of prices for the assets in the portfolio. The portfolio revaluations produce a distribution of profits and losses which can be examined to determine the VaR of the portfolio with a chosen level of confidence.

3. Description of the model

In this article the author describes the computation techniques of the second major full valuation approach, Monte Carlo Simulation. As long as price histories for the assets in a portfolio are available for an appropriate length of history then historical simulation is a particularly effective way to calculate VaR. However, if a sufficient history of price changes are difficult to derive (which is usual for fixed income portfolios, for example) it may become impossible to apply historical simulation. There is also another important argument which says that the history of actual prices is a limited source of possible scenarios, from which to derive VaR. Both arguments are certainly answered by Monte Carlo Simulation. This approach involves artificially generating a very large set of hypothetical events (correlated asset price changes), from which VaR is derived. Mathematically Monte Carlo method is the most complicated of the three mentioned approaches. As it is generally possible to calculate portfolio VaR in Microsoft Excel with delta normal and historical simulation methods, it is hardly possible to get relatively fast and precise figure of portfolio VaR in Excel using Monte Carlo approach. Especially this is true for big correlated portfolios with many risk factors [3]. That is why a special software must be chosen that would help to quantify VaR with Monte Carlo method in an effective, easy and fast way. Generally, there are several IT solutions that could help in dealing with this issue. These are mainly powerful mathematical packages. Author’s choice is MATLAB Version 7.0. This solution represents a multifunctional mathematical tool aimed at solving a large variety of tasks in different disciplines from medicine to financial modelling. MATLAB has a number of important advantages [4]. Firstly, besides the ready-made algorithms, user can create his own algorithms to solve definite types of tasks. Secondly, MATLAB can easily operate with large amounts of data, representing them as variables in matrix form. Thirdly, MATLAB is perfectly integrated with Excel. This allows moving huge input matrixes from Excel to MATLAB and easily moving the results of operations back to Excel.

The interface of MATLAB consists of three windows. On the right there is the biggest window, representing the command window. It is a place of entering variables and operations. Up on the left there is a window called workspace that keeps all previously entered variables. Down on the left there is a history window. Here you can retrace the history of operations done in MATLAB.

In order to demonstrate how to work with Monte Carlo method in MATLAB we will use FOREX portfolio of one of the biggest Latvian banks. The portfolio includes 44 different world currencies (mainly basic world currencies as well as CIS and Eastern European currencies), and its base currency is Euro (EUR). *The goal* is to calculate portfolio VaR with the given time horizon and confidence level that would incorporate correlations between currencies.

In principle, MATLAB has a built-in function to realize Monte Carlo method. This function is called “portsim”. But it has a number of peculiarities that make it invalid in the observed case [5]. As a result, the author had to create his own algorithm (programme code). It is depicted below.

```
1)  function[mvar]= movar(C,exrate,portvalue,t,p,l)
2)  n = size(C,1);
3)  A = chol(C);
4)  m = length(exrate);
5)  for j = 1:m;
6)  x1(j) = 1/exrate(j);
7)  end
8)  portvalue1 = portvalue’*x1’;
9)   for i = 1:l
10)  for j = 1:t
11) y = randn(n,1);
12) end
13)  fc = A*y;
14) newexrate = exrate.*exp(fc*sqrt(t));
15)  for k = 1:m
16) x(k) = newexrate(k);
17) s(k) = 1/x(k);
18) end
19) newportvalue = portvalue’*s’;
20) dvp(i) = newportvalue - portvalue1;
21) end
22) sdvp = sort(dvp)’;
23) movar = sdvp(fix(l-l*p))
24) plot(dvp,’b:*’)
```

The first line consists of two parts. The first part represents the name of the function we are going to address. This is the form of the function that is recognized and memorized by MATLAB. The second part of the line (after “=”) shows the variables that are needed for this function. Variable “C” denotes the correlation table of all 44 currencies of our portfolio. Variable “exrate” is the last exchange rate of every currency relative to Euro. Variable “l” is the number of simulations for every currency. Variable “portvalue” defines the open position of every currency. And, finally, variable “t” serves as the term of VaR forecast.

The algorithm uses a number of functions that are built in MATLAB. *“Size”* function defines the size of a variable; *“Chol”* function is the Cholesky factorization, which decomposes the correlation matrix with the purpose to generate correlated random hypothetical exchange rates; *"length"* function defines the length of a variable; *"randn"* function generates random prices. The algorithm also includes a number of cycles, which use standard functions *"for"* and *"end"*. The last line introduces function "plot", that is a graph drawing function. Expression *(dvp,'b:*')* means that the program has got to mark all meanings of "dvp" variable in blue colour on the graph, (letter *b* is responsible for this action). In addition, every meaning of the variable will be marked by symbol (*).

Now we can test the algorithm in operations with the real data. We will analyze the influence of variables "l", "p" and "t" modification on the final result. At first we will experiment with "l", leaving "p" and "t" unchanged at the level of 0.95 and 1 accordingly. First we will equate "l" to 100, 1000, 10000 and 100000. We will make 100 runs of the algorithm with every "l". Results are found in Table 1. "Average" stands for average value; St. Dev. stands for Standard deviation and Coef. var. stands for Coefficient of variation. It is clearly seen that as number of simulations increases Standard Deviations decrease, which means that the result gets more precise. The same tendency is observable in Fig 1, which depicts Coefficients of Variation.

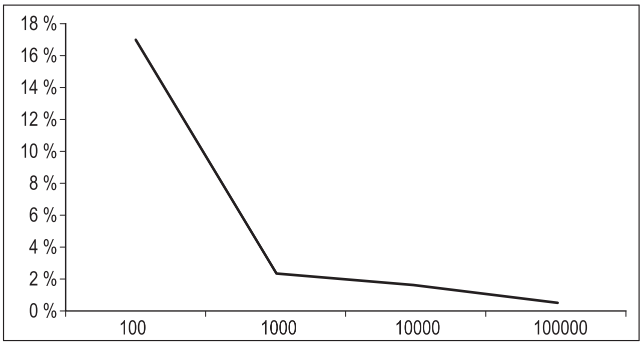

Fig 1. Coefficients of variations for different notions of "l" variable

However, the exactness is not the only important argument for choosing the proper notion of "l" variable. Not less important is time that it takes a computer to make necessary calculations. For this work a powerful Dual Core computer with the processor of Intel Centrino and clock rate of 1.6 Hz was used. With "l" equal to 100 each run of the algorithm it took about a second, with "l" equal to 1 000 – 5 seconds, with "l" equal to 10 000 – about a minute, at "l" equal to 100 000 – about 10 minutes. When this time is multiplied by the number of periods of the forecast (variable "t") the number of iterations increases dramatically. Therefore, taking into account both factors - exactness and time – it seems more logical to use 10 000 simulations. All 10 000 simulated profit/loss values of portfolio are shown in Fig 2 (this figure is built by the 24th line of the algorithm).

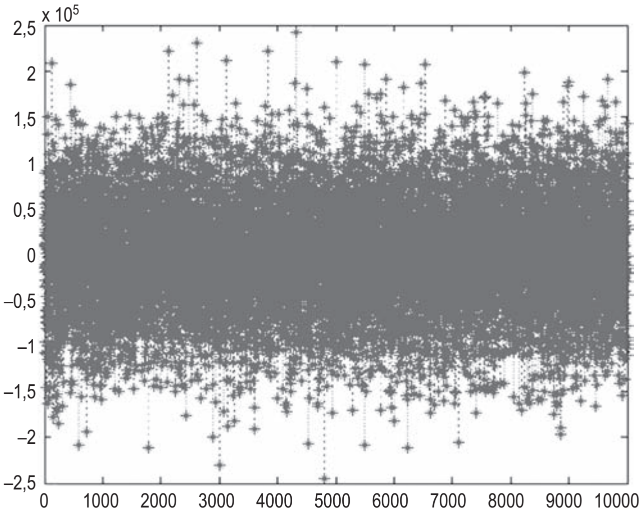

Fig 2. Simulated profit/loss values of the portfolio ("dvp" variable)

Table 1. VaR statistics with different notions of "l" variable

l = 100		l = 1 000		l = 10 000		l = 100 000	
Average	104,184	Average	100,272	Average	100,778	Average	101,410
St. Dev	17,730	St. Dev	2,433	St. Dev	1,629	St. Dev	579
Coef. var	0,170	Coef. var	0,024	Coef. var	0,016	Coef. var	0,006

Now we will experiment with the variable "t". We will take the number of periods from one to ten. The result of calculations is presented in Table 2.

Table 2. VaR values with different notions of "t" variable

N#	VaR
1	100,778
2	140,520
3	173,830
4	199,930
5	228,250
6	242,100
7	264,060
8	284,880
9	295,140
10	317,680

Comparing VaR value for 1 day and for 10 days, it is clearly seen, that VaR grew by $\sqrt{t}$ times, which corresponds to the formula used in the 14th line of the algorithm. Usage of this method is generally accepted in literature, however it is necessary to mention that during significant volatility periods this approach can result in the understated VaR.

The next experiment is a change of the set of confidence level, variable "p". Again, the other parameters remain unchanged. The results of calculations are evaluated in Table 3.

Table 3. VaR values with different notions of "p" variable

Probability	VaR
0,66	25,392
0,85	78,699
0,95	100,778
0,97	120,470
0,99	145,650

Confidence interval 0,66 approximately corresponds to one sigma. With such probability, maximum loss on the next day will be EUR 25,392. The highest confidence interval 0.99 gives VaR of 145,650. The widely known international Basel 2 Accord that controls Capital Adequacy in financial organizations when implementing an internal approach for Capital Adequacy calculations advises to use a confidence interval of 0,99 and forecast term of 10 days. With such parameters VaR value is equal to EUR 447,430. The standard approach, which uses 8 % risk weight for the FOREX risk factors gives the result of EUR 701,560. It means that the usage of internal approach (based on VaR methodology) allows economizing on 37 % of the bank capital reserving less for the currency risk.

4. Conclusions

VaR's popularity keeps growing at a fast pace. Especially this is true for financial organizations in the Post Soviet countries, which require more thorough risk management techniques as their financial systems become more and more sophisticated. The reason of such high interest in VaR systems is in its multifunctionality. VaR is successfully applied in three important financial areas: 1) risk limitation (VaR limits), 2) proper return calculations (in terms of RAROC, Risk Adjusted Return on Capital) and 3) capital adequacy calculations with incorporation of correlations between assets as it was shown in this article.

References

1. JORION, PH. *Value at risk – the new benchmark for managing financial risk*. 2nd Edition. McGraw-Hill, London, 2000, p. 291–311.
2. WILSON, T. Value at risk. *Journal of Risk Management and Analysis*, 1999, Vol 1, p. 61–124. C. Alexander, ed., Wiley, Chichester, England.
3. LOBANOV, A. and CHUGUNOV, A. *Encyclopaedia of financial risk management.* 2nd Edition. Alpina Business Books, Moscow, 2006, p. 307–311.
4. DOWD, K. *Measuring market risk.* 2nd Edition. John Wiley&Sons, London, 2003, p. 210–217.
5. BREDIN, D.; HYDE, S. FOREX Risk: measurement and evaluation using value risk. *Journal of Business Finance Accounting,* 2004, Vol 31, Issue 9–10, p. 1389.