CreditCruncher - Technical Document

Gerard Torrent Gironella

Version 1.1 - R381

Abstract

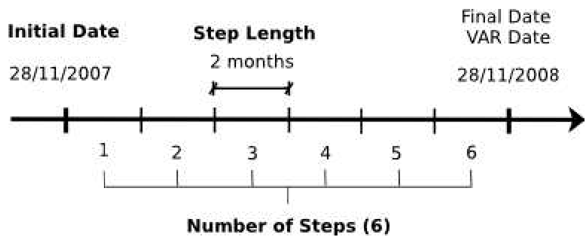
The CCruncher goal is compute the credit risk of portfolios where investments are fixed income assets taking into account default correlations between sectors. This is done determining the probability distribution of the portfolio loss at time T using Monte Carlo simulation method and computing risk statistics over there (Expected Loss, Standard Deviation, Value at Risk, Expected Shortfall).

Keywords: credit risk, Monte Carlo, gaussian copula, Value at Risk, Expected Shortfall.

1 Parameters

1.1 Time

In order to speed up the calculations we precomputes asset losses at fixed times. To fix this time nodes we need an *initial date*, the *number of steps* and the *step length*. The probability distribution is computed at date $T = initial\ date + number\ of\ steps \times step\ length$.



1.2 Ratings and Survival Functions

A credit rating tells a lender or investor the probability of the subject being able to pay back a loan. A poor credit rating indicates a high risk of defaulting. Every rating has associated a survival function. This function indicates the probability that a borrower with initial rating X be non-defaulted at time t. The rating system creation and the construction of the survivals functions is outside the scope of this paper[1].

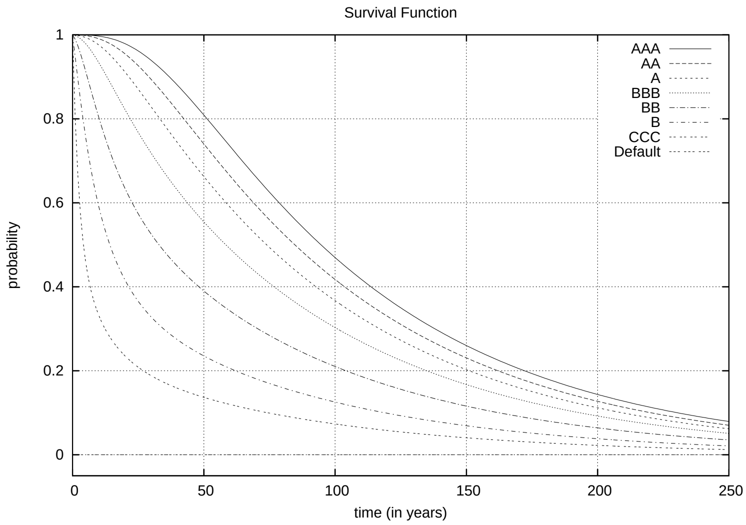


Figure 1: Survival functions

1.3 Sectors and correlations

The risk of a credit portfolio depends crucially on defaults correlations between economic sectors. Sectors are groupings of companies that react similarly to given economic conditions. Sectors examples are: energy, financial, technology, media and entertainment, utilities, health care, etc. Default correlation between sectors measures the dependence to default between the defined sectors. This can be expressed in table form where $\rho_{i,j} = Corr(Sector_i, Sector_j)$:

	$Sector_1$	$\dots$	$Sector_m$
$Sector_1$	$\rho_{1,1}$	$\cdots$	$\rho_{1,m}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$Sector_m$	$\rho_{1,m}$	$\cdots$	$\rho_{m,m}$

[1] Appendix A.3 shows how determine the survival functions using the transition matrix.

1.4 Portfolio

Portfolio is composed by borrowers where each one has an initial rating and belongs to a sector. Each borrower have one or more assets. Each asset is defined by the date when it was incorporated in the portfolio, a expected cashflow and a forecasted recovery at certain dates.

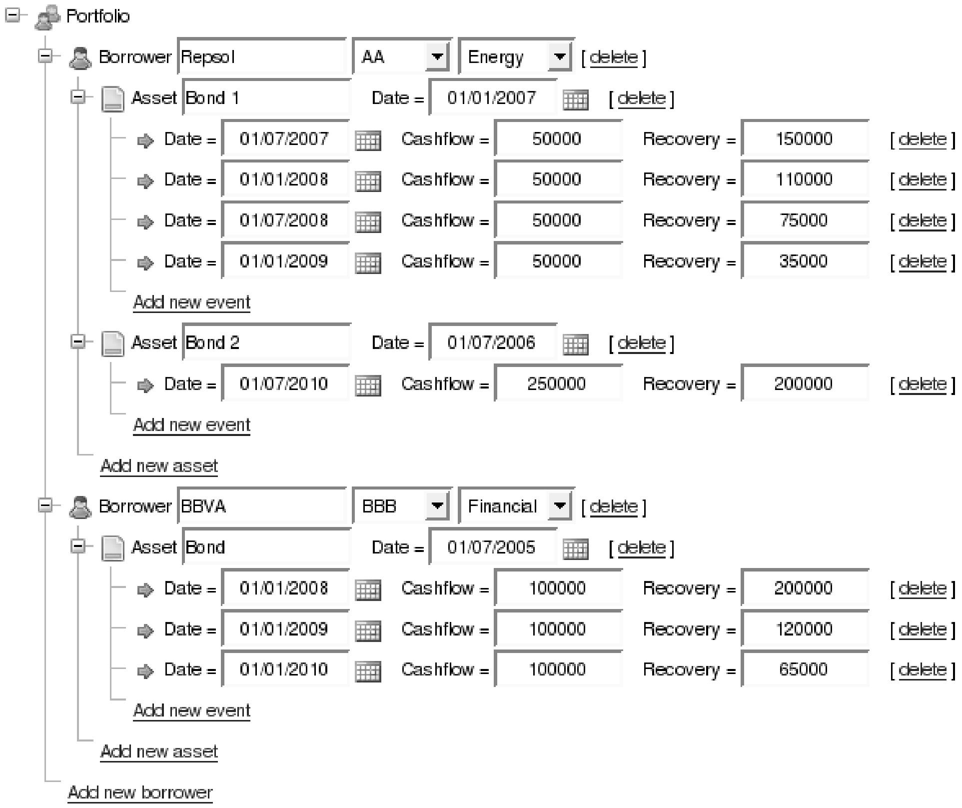


Figure 2: Portfolio example

Cashflow indicates the cash given to the borrower (negative amounts) and the cash received from borrower (positive amounts) at each date (see figure3).

Recovery indicates the cash recovered (after to take legal action against the borrower) if the borrower default at this date (see figure4).

If a borrower defaults, the loss due to this default is the sum of all remaining cashflows at default time minus the recovery at default time.

2 Resolution

2.1 Correlation matrix between borrowers

We need translate the correlations between sectors to a correlations between borrowers. Suppose that we have n borrowers and m sectors. Each borrower belongs to a sector. Correlations between sectors are known:

	$Sector_1$	$\ldots$	$Sector_m$
$Sector_1$	$\rho_{1,1}$	$\cdots$	$\rho_{1,m}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$Sector_m$	$\rho_{1,m}$	$\cdots$	$\rho_{m,m}$

where $\rho_{i,j} = Corr(Sector_i, Sector_j)$. We sort the borrowers so that at the beginning are those of sector 1 and at the end those of sector m. Then we create the borrowers correlation matrix taking as correlation between two borrowers the correlation between his sectors.

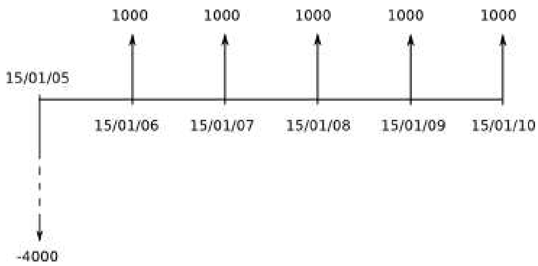


Figure 3: Asset cashflow example

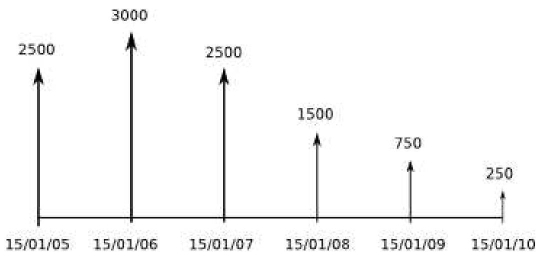


Figure 4: Asset recovery example

$$\begin{pmatrix}
1 & \cdots & \rho_{1,1} & & \rho_{1,k} & \cdots & \rho_{1,k} & & \rho_{1,m} & \cdots & \rho_{1,m} \\
\vdots & \ddots & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\
\rho_{1,1} & \cdots & 1 & & \rho_{1,k} & \cdots & \rho_{1,k} & & \rho_{1,m} & \cdots & \rho_{1,m} \\
 & & & \ddots & & & & & & & \\
\rho_{1,k} & \cdots & \rho_{1,k} & & 1 & \cdots & \rho_{k,k} & & \rho_{k,m} & \cdots & \rho_{k,m} \\
\vdots & \ddots & \vdots & & \vdots & \ddots & \vdots & & \vdots & & \vdots \\
\rho_{1,k} & \cdots & \rho_{1,} & & \rho_{k,k} & \cdots & 1 & & \rho_{k,m} & \cdots & \rho_{k,m} \\
 & & & & & & & \ddots & & & \\
\rho_{1,m} & \cdots & \rho_{1,m} & & \rho_{k,m} & \cdots & \rho_{k,m} & & 1 & \cdots & \rho_{m,m} \\
\vdots & \ddots & \vdots & & \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\
\rho_{1,m} & \cdots & \rho_{1,m} & & \rho_{k,m} & \cdots & \rho_{k,m} & & \rho_{m,m} & \cdots & 1
\end{pmatrix}$$

Observe that this is a correlation matrix (symmetric, definite positive, $|\rho_{i,j}| < 1$, $|\rho_{i,i}| = 1$) composed by blocks. We will use this characteristic to adapt the Cholesky algorithm with the aim to improve memory size and speed.

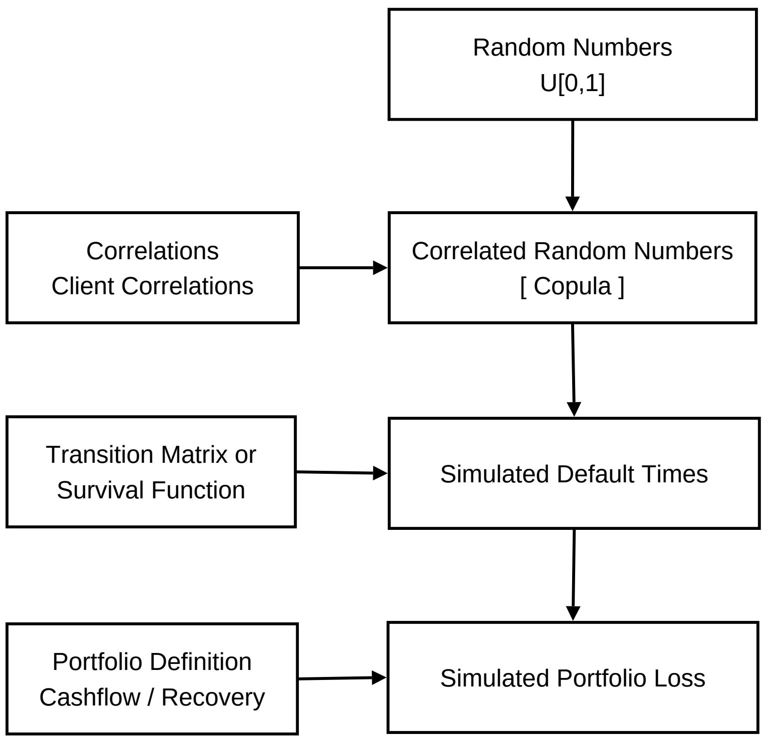


Figure 5: Monte Carlo simulation schema

2.2 Mapping losses at time nodes

Because we want a very fast implementation we precompute losses for each asset at fixed time nodes. Time nodes can be distinct than cashflow or recovery dates. The following paragraphs tells how compute losses at time nodes.

2.2.1 Recovery at time nodes

We compute the recovery at time nodes as a linear interpolation using defined asset recoveries just below and over. If don't exist previous recovery we considerer a value of 0. If don't exist subsequent recovery we considerer a value of 0. Figure 6 shows recovery displayed in figure 4 mapped to biannual time nodes.

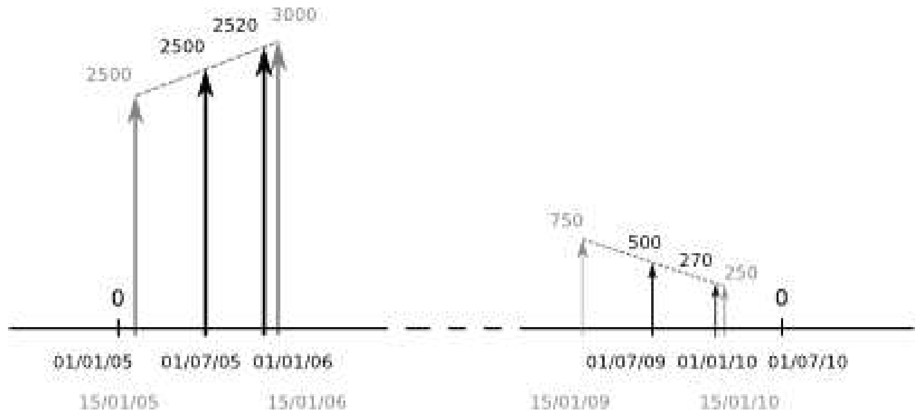


Figure 6: Asset recoveries mapped to time nodes

2.2.2 Losses at time nodes

If borrower defaults before the date of asset risk assumption then loss is 0. If borrower defaults after asset finalization date then loss is 0. If borrower defaults at time node k and exists pending cashflows then loss is the sum of all pending cashflows minus the recovery at time node k. Figure 7 shows asset loss mapped to biannual time nodes of asset displayed at figures 3, 4 and 6.

2.3 Monte Carlo simulation

Monte Carlo methods [5] are class of computational algorithms for simulating the behavior of various physical and mathematical systems. Each simulation consist to compute a random default time for each borrower and found the portfolio loss value. We need that every default time fulfill the borrower survival functions and the borrowers correlation matrix. This is achieved with the simulation of copulas. A copula [6] [8]is a multivariate random variable where each component is a uniform $U[0,1]$.

2.3.1 Random numbers generation

Each simulation requires a set of random numbers between $[0,1]$ correlated by the borrowers correlation matrix (see section 2.1), Σ_1. We generate them using the gaussian copula simulation algorithm:

Step 1. We create the covariance[2] matrix Σ_2 mapping Σ_1 component by component:

$$\Sigma_2 = \begin{pmatrix} 2sin(\frac{\pi}{6}) & 2sin(\rho_{12}\frac{\pi}{6}) & \dots & 2sin(\rho_{1n}\frac{\pi}{6}) \\ 2sin(\rho_{12}\frac{\pi}{6}) & 2sin(\frac{\pi}{6}) & \dots & 2sin(\rho_{2n}\frac{\pi}{6}) \\ \vdots & \vdots & \ddots & \vdots \\ 2sin(\rho_{1n}\frac{\pi}{6}) & 2sin(\rho_{2n}\frac{\pi}{6}) & \dots & 2sin(\frac{\pi}{6}) \end{pmatrix}$$

Observe that Σ_2 have diagonal elements with 1 because $2sin(\frac{\pi}{6}) = 1$.

[2]Correlation and covariance matrix are the same because diagonal elements are 1.

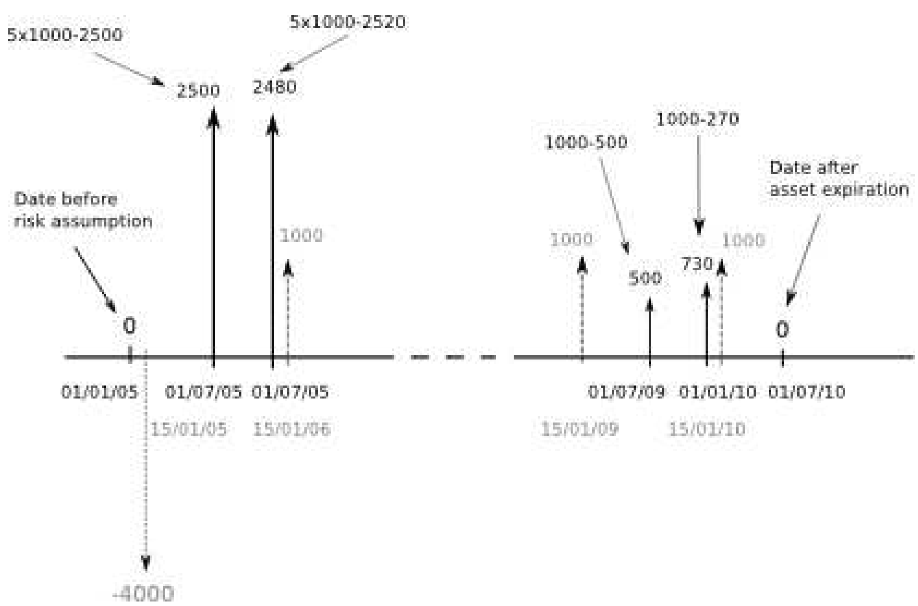


Figure 7: Asset losses mapped to time nodes

Step 2. We apply Cholesky to Σ_2, then $\Sigma_2 = B \cdot B^\top$, where:

$$B = \begin{pmatrix} b_{11} & 0 & \dots & 0 \\ b_{21} & b_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{pmatrix}$$

Step 3. We simulates[3] a $N(0,1)$ n times:

$$\vec{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \qquad y_k \sim N(0,1) \text{ independents}$$

Step 4. We simulate a normal multivariate $\vec{Z} \sim N(\vec{0}, \Sigma_2)$ doing:

$$B \cdot \vec{Y} = \begin{pmatrix} b_{11} & 0 & \dots & 0 \\ b_{21} & b_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix} = \vec{Z}$$

Step 5. Finally we obtain the copula, $\vec{U}$.

$$\begin{pmatrix} \Phi(z_1) \\ \vdots \\ \Phi(z_n) \end{pmatrix} = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} = \vec{X}$$

where $\Phi(x)$ is the $N(0,1)$ cumulative distribution function.

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}\, dt$$

Steps 1 and 2 are done only one time previous to simulation. The rest of steps are performed for every Monte Carlo simulation.

In appendix A.2 is described a Cholesky decomposition adapted to block matrix that allows to work with matrix of size bigger than 50000.

2.3.2 Default time simulation

Given the borrower k and a random number $u_k \in [0,1]$ (generated using a copula in the previous step) we simulates the default time considering the inverse of the borrower survival function at u_k. Figure 8 shows this in a graphic manner.

[3] The algorithm used to simulates a $N(0,1)$ is explained in appendix A.1

2.3.3 Portfolio loss evaluation

At this point, given any borrower, we have a simulated default time. We map this default time to time node closest by the right where we have the precomputed losses for his assets. To obtain the simulated portfolio loss value we sum all assets losses and we kept this value in a list.

2.4 Risk computation

After N simulations (eg. 20000, 50000 or more) we have a list of numbers, x_i, where each number represents a simulated portfolio loss. More values imply more accuracy in results. All risk statistics (eg. Expected Loss) have an error that will be estimated. First of all we can approximate the portfolio loss probability distribution creating an histogram with the simulated values.

CCruncher uses *R package*[4] to perform the statistical computations described in this section.

2.4.1 Expected Loss

Expected Loss is the mean of the portfolio loss probability distribution. Central Limit Theorem [7] grants that:

[4] http://www.r-project.org

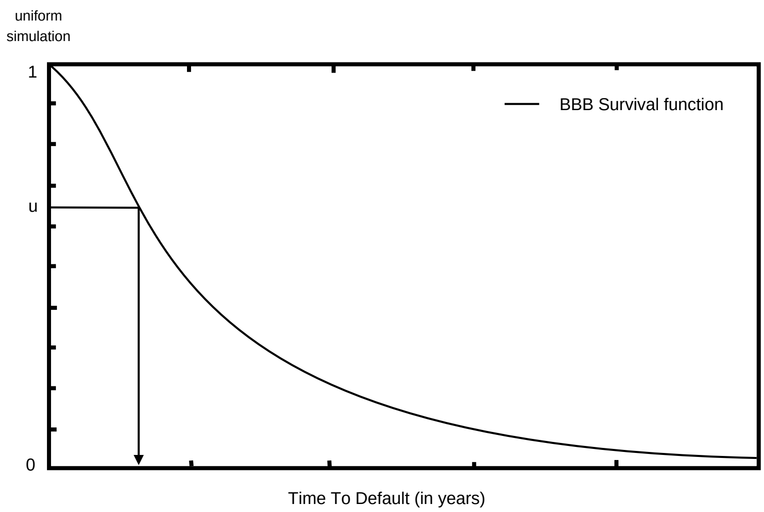


Figure 8: Default time simulation with initial rating BBB

$$\mu = \widehat{\mu} \pm \phi^{-1}\left(\frac{1-\alpha}{2}\right) \cdot \frac{\widehat{\sigma}}{\sqrt{N}}$$

where α is the confidence level for error, ϕ^{-1} the N(0,1) inverse cumulative distribution function and $\widehat{\mu}$ and $\widehat{\sigma}$ are the mean and stddev estimators:

$$\widehat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\widehat{\sigma} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \widehat{\mu})^2} = \sqrt{\frac{1}{N-1}\left(\sum_{i=1}^{N} x_i^2 - \frac{\left(\sum_{i=1}^{N} x_i\right)^2}{N}\right)}$$

2.4.2 Portfolio Loss Standard Deviation

Other usual risk statistic is the portfolio loss standard deviation as measure of dispersion from mean. Central Limit Theorem [7] grants that:

$$\sigma = \widehat{\sigma} \pm \phi^{-1}\left(\frac{1-\alpha}{2}\right) \cdot \frac{\widehat{\sigma}}{\sqrt{2N}}$$

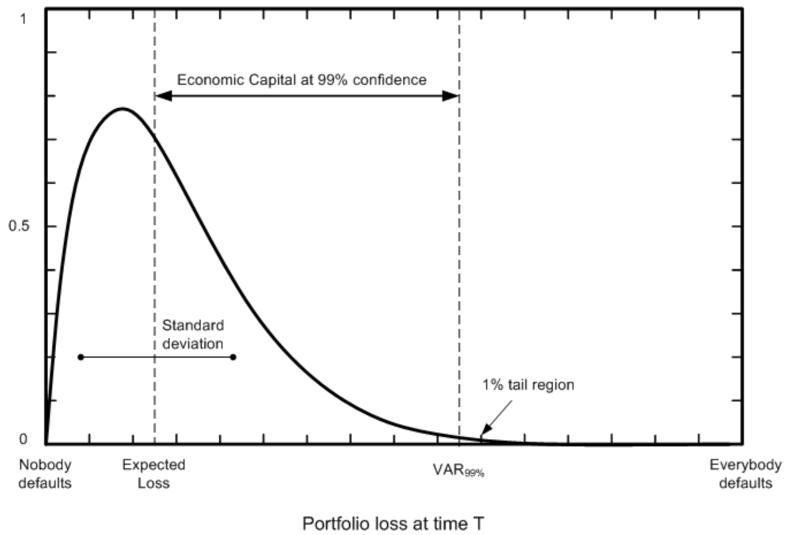


Figure 9: Portfolio loss at time T

where α is the confidence level for error, ϕ^{-1} the N(0,1) inverse cumulative distribution function and $\widehat{\sigma}$ is stddev estimator:

$$\widehat{\sigma} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \widehat{\mu})^2} = \sqrt{\frac{1}{N-1}\left(\sum_{i=1}^{N} x_i^2 - \frac{\left(\sum_{i=1}^{N} x_i\right)^2}{N}\right)}$$

2.4.3 Value At Risk

Value at Risk [4] is the most used risk value. We note as VAR_β where β is the VAR confidence level (eg. VAR at 95%). VAR is another form to say quantile. Then $VAR_\beta = q_\beta = \inf\{x|F(x) \geq \beta\}$.

$$VAR_\beta = \widehat{q_\beta} \pm \phi^{-1}\left(\frac{1-\alpha}{2}\right) \cdot \text{stderr}(q_\beta)$$

where α is the confidence level for error, β is the VAR confidence level, ϕ^{-1} the N(0,1) inverse cumulative distribution function, $\widehat{q_\beta}$ is the quantile estimator, and stderr(q_β) is a estimation of the standard error.

$$\widehat{q_\beta} = x_{k:N}$$

where,

- k fulfills $\frac{k}{N} \leq \beta < \frac{k+1}{N}$
- $x_{k:N}$ is the k-th element of ascendent sorted values

We determine stderr(q_β) using Maritz-Jarret method described in [2].

$$\begin{aligned}
M &= [N\beta + 0.5] \\
a &= M - 1 \\
b &= N - M \\
W_i &= B(a,b,\frac{i+1}{N}) - B(a,b,\frac{i}{N}) \\
C_k &= \sum_{i=1}^{N} W_i \cdot x_i
\end{aligned}$$

where $[x]$ is the integer part of x and $B(a,b,x)$ is the incomplete beta function:

$$B(a,b,x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\int_0^x t^{a-1}(1-t)^{b-1}dt$$

Then,

$$\text{stderr}(q_\beta) = \sqrt{C_2 - C_1^2}$$

2.4.4 Expected Shortfall

VAR is not a distance because it don't fulfills sub-additive property [3], that is $VAR(A+B) \nleq VAR(A) + VAR(B)$. Expected Shortfall is a consistent risk measure [1] similar to VAR. Can be described as the average of the $\beta\%$ worst losses.

$$ES_\beta = \widehat{ES_\beta} \pm \phi^{-1}\left(\frac{1-\alpha}{2}\right) \cdot \text{stderr}(ES_\beta)$$

where α is the confidence level for error, β is the ES confidence level, ϕ^{-1} the N(0,1) inverse cumulative distribution function, $\widehat{ES_\beta}$ is the ES estimator and stderr(ES_β) is a estimation of the standard error.

We select the simulation portfolio loss values that are bigger than VAR_β.

$$y_1, y_2, y_3, \cdots, y_K \qquad \text{where} \quad y_i > VAR_\beta$$

Then,

$$\widehat{ES_\beta} = \frac{1}{K}\sum_{i=1}^{K} y_i$$

$$\text{stderr}(ES_\beta) = \frac{\sqrt{\frac{1}{K-1}\sum_{i=1}^{K}\left(y_i - \widehat{ES_\beta}\right)^2}}{\sqrt{K}} = \frac{\sqrt{\frac{1}{K-1}\left(\sum_{i=1}^{K} y_i^2 - \frac{\left(\sum_{i=1}^{K} y_i\right)^2}{K}\right)}}{\sqrt{K}}$$

2.4.5 Economic Capital

We can compute the Economic Capital at confidence level β as:

$$\text{Economic Capital} = VAR_\beta - \text{Expected Loss}$$

3 Other considerations

CCruncher takes other considerations that they have not been exposed until this moment with the purpose of simplifying the content.

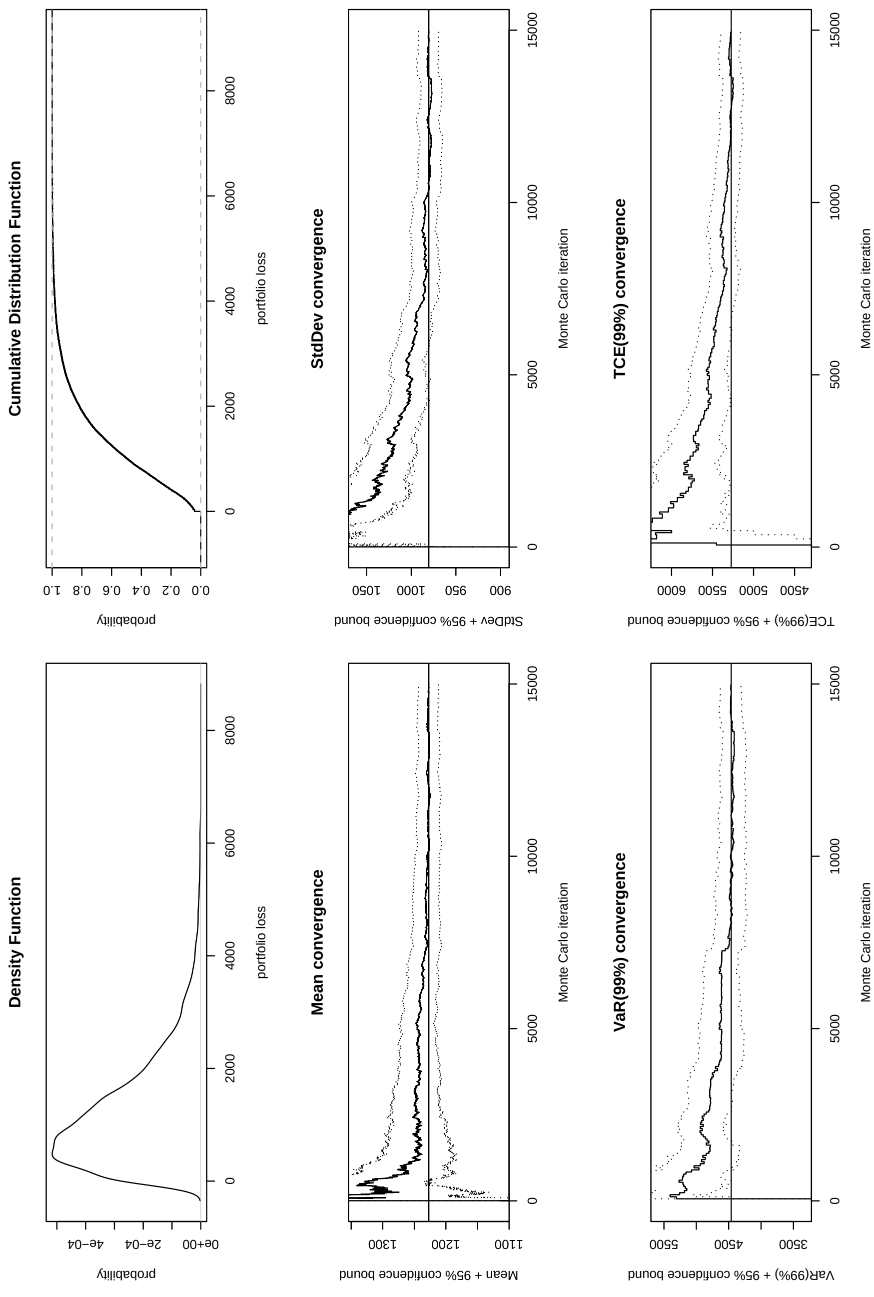


Figure 10: CCruncher results

3.1 Risk aggregation

CCruncher simulates the whole portfolio loss at time T. Also allows simulates (in the same execution), subportfolios defining one or more aggregators. Consequently you can determine what are the branches (or products or borrowers or regions or assets, etc.) that contributes to increase the risk of your portfolio. Every subportfolio have its own list of simulated values that can be processed to obtain the risk indicators for this subportfolio.

3.2 Time updates

CCruncher can considerer that money value decrease along the time following a fixed curve (eg. yield curve). CCruncher allows to specify the yield curve and perform all computations that involves cash transport along the time (recovery mapping, losses mapping and portfolio value at time T) taking into account this fact. The time transport coeficient is determined using the compound interest formula:

$$\Upsilon(r, t_0, t_1) = (1 + r)^{(t_1 - t_0)}$$

3.3 Antithetic technique

The random number generation using a copula is time expensive. Gaussian copula is symmetric, that means that $(u_1, u_2, \cdots, u_N)$ is equiprobable to $(1 - u_1, 1 - u_2, \cdots, 1 - u_N)$. In CCruncher antithetic mode, each copula generation is used 2 times, as $(u_1, u_2, \cdots, u_N)$ and as $(1 - u_1, 1 - u_2, \cdots, 1 - u_N)$ reducing in this way the number of generated copulas.

A Appendices

A.1 How simulate a N(0,1)

In order to simulate $Z \sim N(\mu, \sigma^2)$ values we use:

$$z = \mu + \sigma \cdot \sqrt{-2ln(u_1)} \cdot cos(2\pi \cdot u_2) \qquad u_1, u_2 \sim U[0,1]$$

Where u_i are generated by a Mersenne Twister random number generator.

A.2 Cholesky decomposition for block matrix

Cholesky algorithm decompose a symmetric positive definite matrix into a lower triangular matrix and the transpose of the lower triangular matrix. Algorithm description can be found in *Numerical Recipes in C*[5].

$$U^\top \cdot U = A$$

If we have a portfolio of 50000 borrowers, our correlation matrix size will be a 50000×50000 matrix. This requires up to 19 Gb. of RAM memory and multiply this matrix by a vector suppose 2500000000 multiplications.

We adapt the Cholesky algorithm in order to consider that borrower correlations matrix is a block matrix with 1's in diagonal like this example:

$$A = \left(\begin{array}{cccc|ccc} 1 & 0.5 & 0.5 & 0.5 & 0.1 & 0.1 & 0.1 \\ 0.5 & 1 & 0.5 & 0.5 & 0.1 & 0.1 & 0.1 \\ 0.5 & 0.5 & 1 & 0.5 & 0.1 & 0.1 & 0.1 \\ 0.5 & 0.5 & 0.5 & 1 & 0.1 & 0.1 & 0.1 \\ \hline 0.1 & 0.1 & 0.1 & 0.1 & 1 & 0.3 & 0.3 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.3 & 1 & 0.3 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.3 & 0.3 & 1 \end{array}\right)$$

We decompose the previous matrix using the standard Cholesky decomposition:

$$U = \left(\begin{array}{cccc|ccc} 1.00000 & 0.50000 & 0.50000 & 0.50000 & 0.10000 & 0.10000 & 0.10000 \\ 0 & 0.86603 & 0.28868 & 0.28868 & 0.05774 & 0.05774 & 0.05774 \\ 0 & 0 & 0.81650 & 0.20412 & 0.04082 & 0.04082 & 0.04082 \\ 0 & 0 & 0 & 0.79057 & 0.03162 & 0.03162 & 0.03162 \\ \hline 0 & 0 & 0 & 0 & 0.99197 & 0.28630 & 0.28630 \\ 0 & 0 & 0 & 0 & 0 & 0.94975 & 0.21272 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.92563 \end{array}\right)$$

[5] http://www.nr.com

Observe that U have repeated elements that allows to kept in RAM memory like this:

$$U = \left| \begin{array}{c|cc} 1.00000 & 0.50000 & 0.10000 \\ 0.86603 & 0.28868 & 0.05774 \\ 0.81650 & 0.20412 & 0.04082 \\ 0.79057 & 0 & 0.03162 \\ 0.99197 & 0 & 0.28630 \\ 0.94975 & 0 & 0.21272 \\ 0.92563 & 0 & 0 \end{array} \right|$$

that is, for each row we kept the diagonal value and the value of each sector. With this strategy the required memory size is $N \times (M + 1)$ where N is the number of borrowers and M the number of sectors. With this consideration the memory required to store a 50000 borrowers correlation matrix is only 4.2 Mb.

We use the fact that matrix U have repeated elements to reduce the number of operations required to multiply A by a vector. Let us to see an example:

$$\begin{aligned} &(U \cdot x)_2 = 0.86603 \cdot x_2 + 0.28868 \cdot x_3 + 0.28868 \cdot x_4 + \\ &0.05774 \cdot x_5 + 0.05774 \cdot x_6 + 0.05774 \cdot x_7 \\ &= 0.86603 \cdot x_2 + 2 \cdot 0.28868 \cdot (x_3 + x_4) + 3 \cdot 0.05774 \cdot (x_5 + x_6 + x_7) \end{aligned}$$

With this consideration we can reduce the number of operations from N^2 to $N \times (M + 1)$ where N is the number of borrowers and M the number of sectors. In the 50000 borrowers example, the operations number reduce from 2500000000 to only 500000.

A.3 From transition matrix to survival function

The T-years transition matrix gives the probability to jump from rating r_i to rating r_j in a term of T years.

$$M_T = \begin{pmatrix} m_{1,1} & \dots & m_{1,n} \\ \vdots & \ddots & \vdots \\ m_{n,1} & \dots & m_{n,n} \end{pmatrix} \qquad m_{i,j} = P(r_i \to r_j; T)$$

where n is the number of ratings and $m_{i,j}$ is the probability that a borrower with rating r_i jumps to r_j in T years. Figure 11 shows a transition matrix example with the probability that a borrower with rating AA jumps to rating B in 1 year is 0.14%.

Transition matrix can be scaled in time using the following rules:

$$\begin{array}{l} M_{T_1+T_2} = M_{T_1} \cdot M_{T_2} \\ M_{k \cdot T} = M_T^k \\ M_{\frac{T}{k}} = \sqrt[k]{M_T} \end{array} \quad (1)$$

The root of matrix can be computed as:

$$M = P^{-1} \cdot D \cdot P \longrightarrow M^{\gamma} = P^{-1} \cdot D^{\gamma} \cdot P$$

where P is a matrix composed by the eigenvectors of M and D is a diagonal matrix composed by the eigenvalues of M. The inverse of a matrix can be computed using the LU decomposition as it explained in *Numerical Recipes in C*[6]:

$$Id = P^{-1} \cdot P = L \cdot U \cdot P$$

This allows us to compute probability default for any time and any initial rating (that is, the survival functions) doing:

$$Survival(r_i, t) = (M_t)_{i,n}$$

where r_i is the initial rating, t is the time, M_t is the transition matrix for time t (scaled from M_T using (1)) and n is the index of default rating, r_n.

[6]http://www.nr.com

	AAA	AA	A	BBB	BB	B	CCC	Default
AAA	90.81	8.33	0.68	0.06	0.12	0.00	0.00	0.00
AA	0.70	90.65	7.79	0.64	0.06	0.14	0.02	0.00
A	0.09	2.27	91.05	5.52	0.74	0.26	0.01	0.06
BBB	0.02	0.33	5.95	86.93	5.30	1.17	0.12	0.18
BB	0.03	0.14	0.67	7.73	80.53	8.84	1.00	1.06
B	0.00	0.11	0.24	0.43	6.48	83.46	4.07	5.21
CCC	0.22	0.00	0.22	1.30	2.38	11.24	64.86	19.78
Default	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00

Figure 11: 1-year transition matrix

References

[1] Dirk Tasche Carlo Acerbi. Expected shortfall: a natural coherent alternative to value at risk. *BIS*, 2001.

[2] Jonathan E. Grindlay Jaesub Hong, Eric M. Schlegel. New spectral classification technique for x-ray sources: quantile analysis. 2004.

[3] Stefan R. Jaschke. Quantile-var is the wrong measure to quantify market risk for regulatory purposes. *BIS*, 2001.

[4] Philippe Jorion. *Value at Risk*. McGraw-Hill, 1997.

[5] Mervyn Marasinghe. Monte carlo methods. Class notes for Iowa State University, Dept. of Statistics.

[6] Alexander McNeil Paul Embrechts and Daniel Straumann. Correlation and dependence in risk management: properties and pitfalls. *RiskLab*, 1999.

[7] Murray R. Spiegel. *Estadística*. Schaum, 1997.

[8] Shaun S. Wang. Aggregation of correlated risk portfolios: Models & algorithms. *CAS Committee on Theory of Risk*.