Portfolio Credit Risk: Models and Numerical Methods

Guillermo Navas Palencia

Universitat Politècnica de Catalunya

guillermo.navas@estudiant.upc.edu

January 22, 2016

Contents

Introduction

- Credit risk is the risk of loss resulting from an obligors inability to meet its legal obligation according to the debt contract.
- For financial institutions it is essential to quantify the credit risk at a portfolio level. Portfolio credit loss modelling requires the default dependence among obligors. A common approach is utilizing one or multiple factor models, such as the obligors are independent conditional on some latent common factor, which in some cases are assumed to follow a standard normal distribution.
- Financial institutions are also interested in the computation of common risk measures, such as Value-at-Risk (VaR) and Expected Shortfall (ES). VaR is the measure chosen in the Basel II Accord (Basel Committee on Bank Supervision).
- Credit rating agencies.

Generalized One-factor Model

Assumption (Generalized One-Factor Model, Schönbucher (2000))

The values of the assets of the obligors are driven by a common factor Y which has distribution function $G(y)$, and an idiosyncratic noise component ϵ_n which is distributed according to the distribution function $H(\epsilon_n)$

$$V_n(t) = \sqrt{\rho_n}\, Y + \sqrt{1-\rho_n}\, \epsilon_n \quad n \leq N \tag{1}$$

where $Y \sim G$ and ϵ_n, $n \leq N$ are independent and identically $H(\epsilon_n)$-distributed with mean 0 and variance 1 and $\rho \in [0,1]$.

The generalized default probability or conditional default risk is given by

$$p_n(y) = H\left(\frac{\alpha_n - \sqrt{\rho_n}\, y}{\sqrt{1-\rho_n}}\right) \tag{2}$$

and must satisfy

$$E[p_n(y)] = \int_{-\infty}^{\infty} H\left(\frac{\alpha_n - \sqrt{\rho_n}\, y}{\sqrt{1-\rho_n}}\right) G'(y)\, dy = p_n \tag{3}$$

The generalized default probability function given by

$$P[X = n] = \int_{-\infty}^{\infty} \binom{N}{n} \left(H\left(\frac{\alpha - \sqrt{\rho}\, y}{\sqrt{1-\rho}}\right)\right)^n \left(1 - H\left(\frac{\alpha - \sqrt{\rho}\, y}{\sqrt{1-\rho}}\right)\right)^{N-n} G'(y)\, dy \tag{4}$$

Computing threshold α (1)

1. Apply Double Exponential transformation to Equation (3)

$$E[p(y)] \approx \frac{\pi}{2} h \sum_{k=1}^{N} w_k H\left(\frac{\alpha - \sqrt{\rho} x_k}{\sqrt{1-\rho}}\right) g(x_k) = p \tag{5}$$

where $t_k = -ta + (k-1)h$, $ta = 12^{0.46}$, $h = 2ta/(N-1)$ and

$$x_k = \sinh\left(\frac{\pi}{2}\sinh(t_k)\right) \quad w_k = \cosh(t_k)\cosh\left(\frac{\pi}{2}\sinh(t_k)\right) \tag{6}$$

2. Root-finding algorithm
   - Bisection algorithm

$$f(\alpha; \rho, p) = \frac{\pi}{2} h \sum_{k=1}^{N} w_k H\left(\frac{\alpha - \sqrt{\rho} x_k}{\sqrt{1-\rho}}\right) g(x_k) - p \tag{7}$$

   - Newton-Raphson algorithm

$$f'(\alpha; \rho, p) = \frac{\partial f(\alpha, \rho, p)}{\partial \alpha} = \frac{\pi h}{2\sqrt{1-\rho}} \sum_{k=1}^{N} w_k H'\left(\frac{\alpha - \sqrt{\rho} x_k}{\sqrt{1-\rho}}\right) g(x_k) \tag{8}$$

$$\alpha_{k+1} = \alpha_k - \frac{f(\alpha_k)}{f'(\alpha_k)} \qquad \alpha_0 = \texttt{Bisection}(f(\alpha), lb, ub, 1e{-}01) \tag{9}$$

   - Brent's method

Computing threshold α (2) - Computational results

In this case the derivative $f'(\alpha, \rho, p)$ is expensive to compute, as much as the function $f(\alpha, \rho, p)$. Therefore, $f'(\alpha, \rho, p)$ can be approximated by a second order centered finite difference approximation [1].

$$f'(\alpha; \rho, p) \approx \frac{f(a+h) - f(a-h)}{2h} + \mathcal{O}(h^2) \tag{10}$$

or a more accurate fourth order finite difference approximation

$$f'(\alpha; \rho, p) \approx \frac{-f(a+2h) + 8f(a+h) - 8f(a-h) + f(a-2h)}{12h} + \mathcal{O}(h^4) \tag{11}$$

Discretization	Method	Iterations	Abs.Error	Time (milliseconds)
Riemann	B+NR	11	$5.23 \cdot 10^{-11}$	8.70
	B+NRFD	11	$5.23 \cdot 10^{-11}$	10.71
	Brent	15	$5.23 \cdot 10^{-11}$	5.99
DE	B+NR	23	$1.78 \cdot 10^{-10}$	5.45
	B+NRFD	11	$1.78 \cdot 10^{-10}$	3.28
	Brent	15	$\mathbf{1.78 \cdot 10^{-11}}$	**1.98**

Table: Benchmark for computing the threshold α for two discretization schemes and 3 root-finding algorithms. B+NR = Bisection + Newton-Raphson, B+NRFD = Bisection + Newton-Raphson with centered finite difference order 2. Tolerance=$1e{-}15$. $\alpha_0 = -1.03125$. DE: $N = 100$, $d = 11$. Riemann midpoint: $N = 100$. Gaussian factor model: $Y \sim \mathcal{N}(0,1)$ and $\epsilon_n \sim \mathcal{N}(0,1)$ where $\alpha^* = \Phi^{-1}(0.15)$.

[1] The value of the parameter h is $u^{2/3}$ where u is the **unit roundoff** typically 10^{-16} in double-precision.

Generalized default probability function

Once the threshold α is computed, the following integral is numerically solved

$$P[X = n] = \int_{-\infty}^{\infty} \binom{N}{n} \left(H\left(\frac{\alpha - \sqrt{\rho}\, y}{\sqrt{1-\rho}} \right) \right)^n \left(1 - H\left(\frac{\alpha - \sqrt{\rho}\, y}{\sqrt{1-\rho}} \right) \right)^{N-n} G'(y)\, dy$$

Computation methods:

- Double exponential transformation.
- Monte Carlo + Median Latin Hypercube sampling.

$$P[X = n] = 2 \int_{-\pi/2}^{\pi/2} \mathcal{B}\left(n, N, H\left(\frac{\alpha - \sqrt{\rho}\, \tan(x)}{\sqrt{1-\rho}} \right) \right) \frac{G'(\tan(x))}{\cos(2x) + 1}\, dx \qquad (12)$$

  where $\mathcal{B}(n, N, p)$ denotes the probability mass function of the binomial distribution.

- Gauss-Legendre quadrature.

Efficient computation of the binomial distribution

Binomial distribution

$$f(k;n,p) = e^{\ln\binom{n}{k}+k\ln p+(n-k)\ln(1-p)} \tag{13}$$

Necdet Batir (2010) established a new Stirling-type approximation formula for the factorial function, which so far is one of the most accurate approximations to $n!$

$$n! \approx \sqrt{2\pi}n^n e^{-n}\sqrt[4]{n+\frac{1}{6}+\frac{1}{72n}-\frac{31}{6480n^2}-\frac{139}{155520n^3}+\frac{9871}{6531840n^4}} \tag{14}$$

We make use of (14) to write an expression for the logarithm of the binomial coefficient,

$$\ln n! \approx n\ln n - n + \tfrac{1}{2}\ln(2\pi) + \tfrac{1}{4}\ln\left(n+\tfrac{1}{6}+\tfrac{1}{72n}\left(1+\tfrac{1}{90n}\left(-31+\tfrac{1}{24n}\left(-139+\tfrac{9871}{42n}\right)\right)\right)\right) \tag{15}$$

and having written $\ln n!$ we can easily write $\ln\binom{n}{k}$ as follows,

$$\begin{aligned}\ln\binom{n}{k} \approx\ & n\ln n-(n-k)\ln(n-k)-k\ln k-\tfrac{1}{2}\ln(2\pi)\\ &+\tfrac{1}{4}\Big[\ln\left(n+\tfrac{1}{6}+\tfrac{1}{72n}\left(1+\tfrac{1}{90n}\left(-31+\tfrac{1}{24n}\left(-139+\tfrac{9871}{42n}\right)\right)\right)\right)\\ &-\ln\left((n-k)+\tfrac{1}{6}+\tfrac{1}{72(n-k)}\left(1+\tfrac{1}{90(n-k)}\left(-31+\tfrac{1}{24(n-k)}\left(-139+\tfrac{9871}{42(n-k)}\right)\right)\right)\right)\\ &-\ln\left(k+\tfrac{1}{6}+\tfrac{1}{72k}\left(1+\tfrac{1}{90k}\left(-31+\tfrac{1}{24k}\left(-139+\tfrac{9871}{42k}\right)\right)\right)\right)\Big]\end{aligned} \tag{16}$$

Median Latin Hypercube sampling

Latin hypercube sampling (LHS) is one form of stratified sampling that can yield more precise estimates of the distribution function.

Algorithm 1 Median Latin Hypercube Sampling

procedure MLHS(samples, dimensions, lbound, ubound)
  $a \leftarrow lbound$
  $b \leftarrow ubound$
  for $j := 1$ to dimensions **do**
    vector of indexes: $\pi_i, \quad i = 1, \ldots, samples$
    random shuffle of π_i
    $v_{ij} = (b - a)\dfrac{\pi_i - 0.5}{samples} + a$
    to extend to not uniform distribution, where F is the cumulative distribution:
    $z_{ij} = F^{-1}(v_{ij})$
  end for
  return z
end procedure

Theoretical results for the univariate case show that the sampling error of MC decreases as $\mathcal{O}(1/\sqrt{N})$, whereas the sampling error for LHS is $\mathcal{O}(1/N)$, therefore it is quadratically faster.

Remark

MC and LHS are both unbiased estimation techniques, therefore computed statistics approach their theoretical values as $N \to \infty$.

Numerical examples (1)

Model	Method	Mean	Stdev	$VaR_{0.999}$	Time(s)
$L-L$	MC+MLHS	0.1499999756	0.0852446935	0.629988	0.199
$L-L$	DE	0.1499999756	0.0852446935	0.629988	0.132
$L-L$	GL	0.1499999610	0.0852446130	0.626293	0.020
$E-E$	MC+MLHS	0.1500000000	0.1058402431	0.476118	0.255
$E-E$	DE	0.1500000000	0.1058402431	0.476118	0.191
$E-E$	GL	0.1500000005	0.1058402453	0.476121	0.032

Table: Metrics and computation time for generalized portfolios. $L = Logistic(0, \sqrt{3}/\pi)$. $E = EMG(\mu = -0.95)$. Methods (MC+MLHS(500), DE(12, 300) and GL(60)).

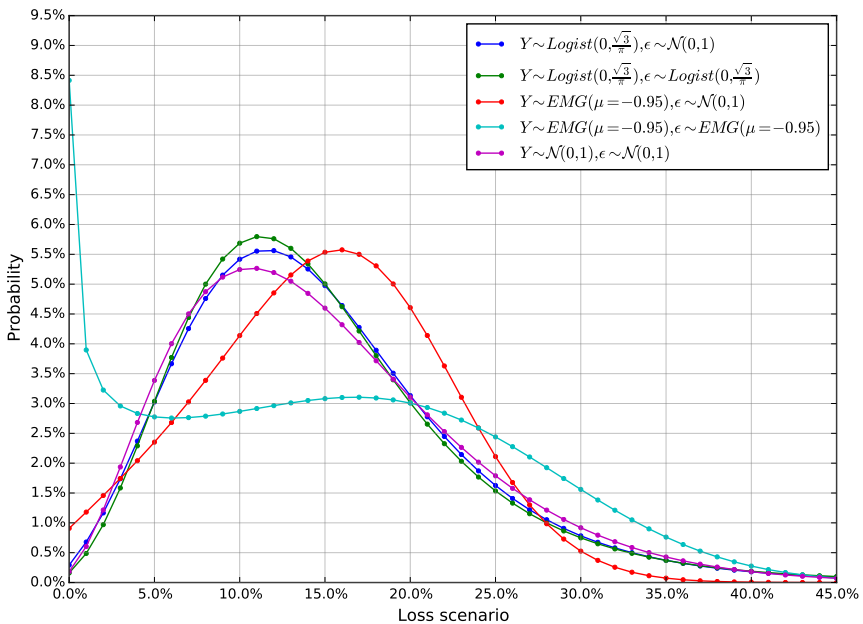
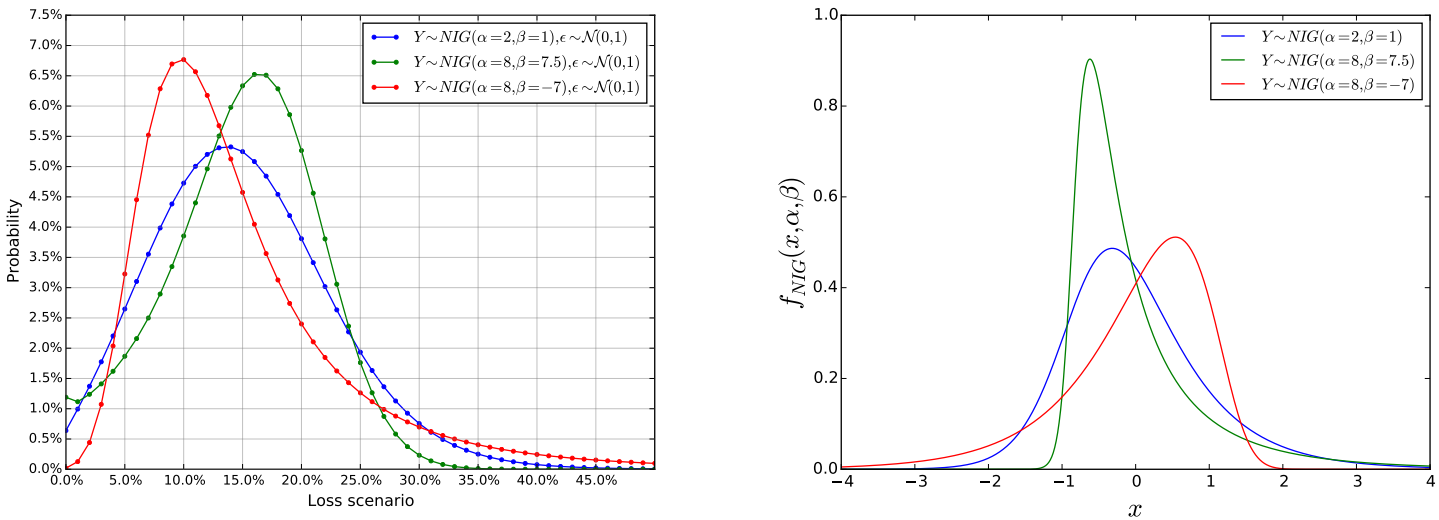


Figure: Probability distribution of the defaults of several special factor models.

Numerical examples (2)

Special factor models

- L - N: Logistic - Normal factor model
- L - L: Logistic factor model
- E - N: Exponentially Modified Gaussian - Normal factor model
- E - E: Exponentially Modified Gaussian factor model
- NIG - N: Normal Inverse Gaussian - Normal factor model
- NIG - NIG: Normal Inverse Gaussian factor model



(a) *Probability distribution.* (b) *Density of some NIG distribution.*

Figure: Probability distribution of the defaults of special factor model NIG-Normal.

General loss distribution

The Vasicek distribution implies a strong assumption; the systematic and idiosyncratic factors are normally and standardized distributed. Schönbucher provided a general portfolio cumulative distribution function given by

$$F(x;\alpha,\rho) = P[X \leq x] = 1 - G\left(\frac{\alpha}{\sqrt{\rho}} - \sqrt{\frac{1-\rho}{\rho}}H^{-1}(x)\right) \tag{17}$$

We can derive Equation (17) to obtain the general portfolio loss density function

$$f(x;\alpha,\rho) = \frac{\partial F(x;\alpha,\rho)}{\partial x} = \sqrt{\frac{1-\rho}{\rho}}G'\left(\frac{\alpha}{\sqrt{\rho}} - \sqrt{\frac{1-\rho}{\rho}}H^{-1}(x)\right)\frac{\partial}{\partial x}H^{-1}(x) \tag{18}$$

and the quantile function or inverse cumulative distribution function can be easily obtained from Equation (17)

$$F^{-1}(\beta;\alpha,\rho) = H\left(\frac{\alpha - \sqrt{\rho}G^{-1}(1-\beta)}{\sqrt{1-\rho}}\right) \tag{19}$$

Special factor model - Logistic factor model

The following special factor model is the *Logistic-Logistic* defined as $Y \sim Logistic(0, \frac{\sqrt{3}}{\pi})$ and $H \sim Logistic(0, \frac{\sqrt{3}}{\pi})$ $(L - L)$.

The $L - L$ distribution function is given by

$$F(x; \alpha, \rho) = \frac{1}{2}\left(1 - \tanh\left(\frac{\alpha\pi}{2\sqrt{3\rho}} - \frac{\sqrt{1-\rho}\ln\left(\frac{x}{1-x}\right)}{2\sqrt{\rho}}\right)\right) \tag{20}$$

and its derivative leads to the $L - L$ density function

$$f(x; \alpha) = \sqrt{\frac{(1-\rho)}{16\rho}}\operatorname{sech}^2\left(\frac{\alpha\pi}{2\sqrt{3\rho}} - \frac{\sqrt{1-\rho}\ln\left(\frac{x}{1-x}\right)}{2\sqrt{\rho}}\right)\frac{1}{(1-x)x} \tag{21}$$

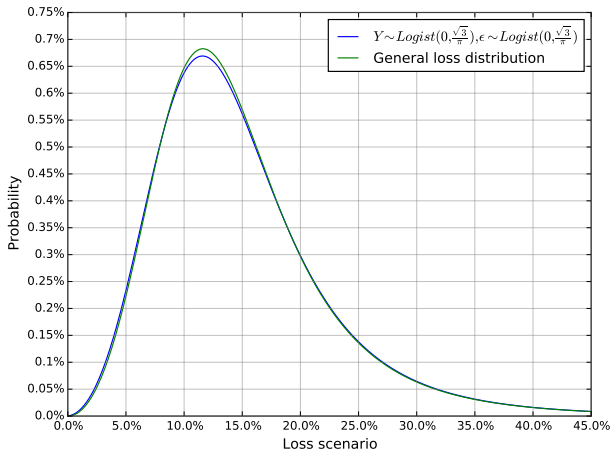
$L - L$ quantile function

$$F^{-1}(\beta; \alpha, \rho) = \frac{1}{2}\left(1 + \tanh\left(\frac{\pi}{2\sqrt{3}}\frac{\alpha - \sqrt{\rho}\ln\left(\frac{1-\beta}{\beta}\right)\frac{\sqrt{3}}{\pi}}{\sqrt{1-\rho}}\right)\right) \tag{22}$$

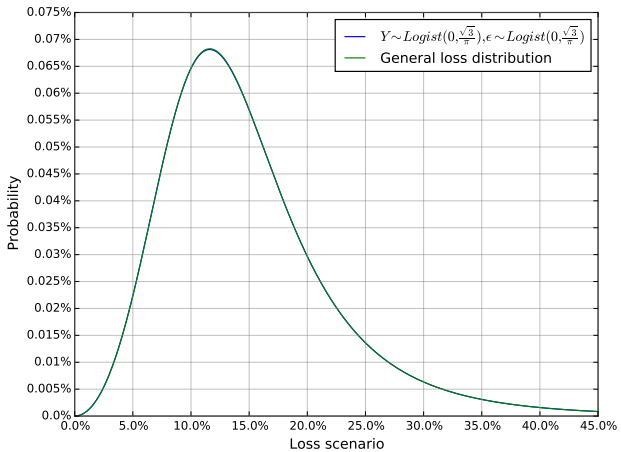
Special factor models - Logistic factor model - Results

Model	N	Mean	Stdev	$VaR_{0.999}$
GLD	1000	0.1500000000	0.0777958665	0.610054
GVM(DE)	1000	0.1500000000	0.0785725438	0.612051
GLD	10000	0.1500000000	0.0777958665	0.610054
GVM(DE)	10000	0.1500000000	0.0778738828	0.610253

Table: Main metrics. General Loss Distribution vs. Generalized Vasicek Model: $Y \sim Logistic(0, \sqrt{3}/\pi)$ and $H \sim Logistic(0, \sqrt{3}/\pi)$.



(a) *Assets=1000.*



(b) *Assets=10000.*

Figure: General Loss Distribution vs. Generalized Vasicek Model: $Y \sim Logistic(0, \sqrt{3}/\pi)$ and $H \sim Logistic(0, \sqrt{3}/\pi)$.

The model framework (1)

Consider a credit portfolio consisting of N obligors, where any obligor $n = 1, \ldots, N$ can be characterized by three parameters: the exposure at default (EAD) denoted by E_n, loss given default (LGD) denoted by Λ_n and the probability of default PD_n.

$$D_n = \begin{cases} 1 & \text{if obligor } n \text{ is in default with probability } PD_n \\ 0 & \text{if obligor } n \text{ is not in default with probability } 1 - PD_n \end{cases} \tag{23}$$

The loss incurred due to default of obligor n is given by

$$L_n = E_n \cdot \Lambda_n \cdot D_n = w_n \cdot D_n \tag{24}$$

where $w_n = E_n \cdot \Lambda_n$ is the effective exposure of obligor n. Then the portfolio loss is defined as

$$L = \sum_{n=1}^{N} L_n \tag{25}$$

For a given state of the economy $Y = y$, the conditional Fourier transform of the portfolio default distribution $(f_L(x), 0 < x < 1)$ is defined as

$$\hat{f}_{L|Y=y}(t) = E[e^{-itL} \mid Y = y] = E[e^{-it\sum_{n=1}^{N} w_n D_n} \mid Y = y] \tag{26}$$

The model framework (2)

Given that $P[D_n = 1 \mid Y = y] = p_n(y)$ and $P[D_n = 0 \mid Y = y] = 1 - p_n(y)$, the conditional Fourier transform of the portfolio default distribution is given by

$$\hat{f}_{L|Y=y}(t) = \prod_{n=1}^{N} [1 + p_n(y) + p_n(y) e^{-itw_n}] \tag{27}$$

and the unconditional Fourier transform is obtained by considering the density function of the factor Y, $\phi(y)$

$$\hat{f}_L(t) = \int_{-\infty}^{\infty} \prod_{n=1}^{N} [1 + p_n(y) + p_n(y) e^{-itw_n}] \phi(y)\, dy \tag{28}$$

where

$$p_n(y) = \Phi\left(\frac{\alpha_n - \sqrt{\rho_n} y}{\sqrt{1 - \rho_n}} \right) \quad \text{and} \quad \phi(x) = \frac{e^{-x^2/2}}{\sqrt{2\pi}}$$

Stochastic loss given default (1)

Assuming D_n and Λ_n are conditionally independent given $Y = y$.

$$\hat{f}_{L|Y=y}(t) = \prod_{n=1}^{N} E[e^{-itE_n D_n \Lambda_n} \mid Y = y] \tag{29}$$

$$E[e^{-itE_n D_n \Lambda_n} \mid Y = y] = 1 - p_n(y) + p_n(y)\hat{f}_{\Lambda_n}(t, En; Y) \tag{30}$$

where

$$\hat{f}_{\Lambda_n}(t, En; Y) = E[e^{-itE_n \cdot 1 \cdot \Lambda_n} \mid Y = y] \tag{31}$$

Hence, the conditional portfolio Fourier Transform is given by

$$\hat{f}_{L|Y=y}(t) = \prod_{n=1}^{N}[1 - p_n(y) + p_n(y)\hat{f}_{\Lambda_n}(t, En; Y)] \tag{32}$$

and therefore the unconditional portfolio Fourier transform is expressed as

$$\hat{f}_L(t) = \int_{-\infty}^{\infty} \prod_{n=1}^{N}[1 - p_n(y) + p_n(y)\hat{f}_{\Lambda_n}(t, En; Y)]\phi(y)\, dy \tag{33}$$

Stochastic loss given default (2)

Case 1: Normal distribution

Applying the Fourier transform definition of the Normal distribution:

$$\hat{f}_{\Lambda_n}(t, E_n) = e^{-itE_n\mu_n} e^{\frac{\sigma_n^2}{2} t^2 E_n^2} \tag{34}$$

the portfolio Fourier transform is given by

$$\hat{f}_L(t) = \int_{-\infty}^{\infty} \prod_{n=1}^{N} [1 - p_n(y) + p_n(y) e^{-itE_n\mu_n} e^{\frac{\sigma_n^2}{2} t^2 E_n^2}] \phi(y)\, dy \tag{35}$$

Case 2: Beta distribution

I use the fact that the characteristic function of a beta distributed random variable with parameters α and β can be expressed as follows

$$\varphi_X(t) = \sum_{k=0}^{\infty} \frac{(it)^k B(\alpha + k, \beta)}{k! B(\alpha, \beta)} = 1 + \sum_{k=1}^{\infty} \frac{(it)^k}{k!} \prod_{n=0}^{k-1} \frac{\alpha + n}{\alpha + \beta + n} = {}_1F_1(\alpha; \alpha+\beta; it) \tag{36}$$

The last function ${}_1F_1(a; b; z)$ is the confluent hypergeometric function of the first kind. Thus, the portfolio Fourier transform with beta distributed LGD is given by

$$\hat{f}_L(t) = \int_{-\infty}^{\infty} \prod_{n=1}^{N} [1 - p_n(y) + p_n(y) {}_1F_1(\alpha_n; \alpha_n + \beta_n; -itE_n)] \phi(y)\, dy \tag{37}$$

CF Beta distribution - new algorithm (1)

Algorithm 2 HypBeta: Characteristic function of the Beta distribution

```
1:  function HypBeta(a, b, z, nodes = 64, tol = 1e−15, maxiter = 1000)
2:      a, b ∈ ℝ and x, z ∈ ℂ
3:      if a, b ≳ |z| then
4:          if a, b < 50 then
5:              x = HypTaylor(a, b, z, tol, maxiter)        ▷ Taylor series method
6:          else
7:              x = HypGJ(a, b, z, nodes)                   ▷ Gauss-Jacobi method
8:          end if
9:      else
10:         x = HypSD(a, b, z, nodes)                       ▷ Numerical Steepest Descent method
11:     end if
12:     return x
13: end function
```

Integral representation: Amongst the several methods for computing ${}_1F_1(a;b;z)$, Abramowitz and Stegun give a useful integral form for $\Re(b) > \Re(a) > 0$

$$ {}_1F_1(a;b;z) = \frac{\Gamma(b)}{\Gamma(a)\Gamma(b-a)} \int_0^1 e^{zt} t^{a-1}(1-t)^{b-a-1}\, dt \tag{38} $$

Taylor series: In practice the simplest method for computing the confluent hypergeometric function is to truncate the Taylor series defined as follows

$$ {}_1F_1(a;b;z) \approx S_N = \sum_{j=0}^{N} \frac{(a)_j}{(b)_j} \frac{1}{j!} z^j \tag{39} $$

CF Beta distribution - new algorithm (2)

Numerical Steepest Descent method

1. Transform integral representation into highly oscillatory integral

$$ {}_1F_1(a;b;z) = \frac{\Gamma(b)}{\Gamma(a)\Gamma(b-a)} \int_0^1 e^{zt} t^{a-1}(1-t)^{b-a-1}\, dt \tag{40} $$

$$ {}_1F_1(a;b;z) = \frac{\Gamma(b)}{\Gamma(a)\Gamma(b-a)} \int_0^1 \underbrace{e^{\Re(z)t} t^{a-1}(1-t)^{b-a-1}}_{r(t)} \underbrace{e^{i\Im(z)t}}_{e^{i\omega t},\omega=\Im(z)}\, dt \tag{41} $$

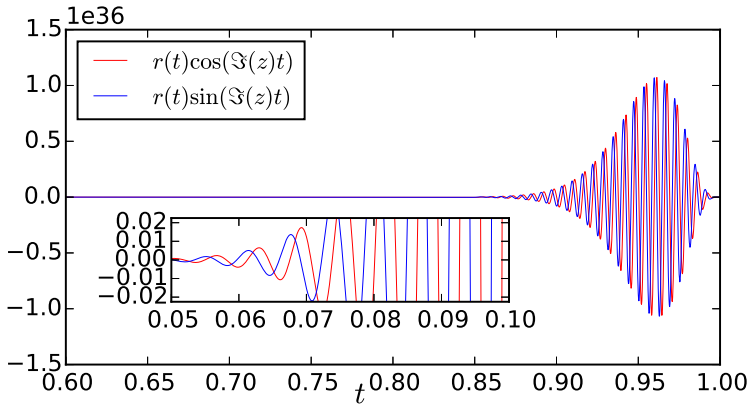
2. Deformation of integration path onto the path of steepest descent

$$ \begin{aligned} {}_1F_1(a;b;z) = \frac{\Gamma(b)}{\Gamma(a)\Gamma(b-a)} \Bigg[& \frac{i}{\omega} \int_0^\infty e^{\Re(z) i\frac{q}{\omega}} \left(i\frac{q}{\omega}\right)^{a-1} \left(1 - i\frac{q}{\omega}\right)^{b-a-1} e^{-q}\, dq \\ & - \frac{ie^{i\omega}}{\omega} \int_0^\infty e^{\Re(z)(1+i\frac{q}{\omega})} \left(1 + i\frac{q}{\omega}\right)^{a-1} \left(-i\frac{q}{\omega}\right)^{b-a-1} e^{-q}\, dq \Bigg] \end{aligned} \tag{42} $$

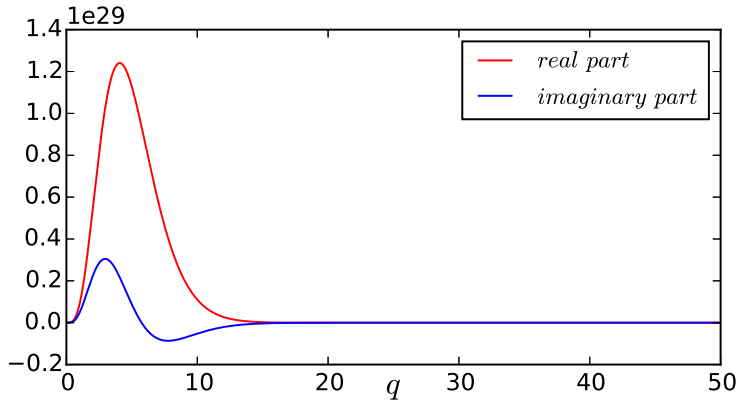
3. Numerical integration: Gauss-Laguerre quadrature

$$ \begin{aligned} {}_1F_1(a;b;z) \approx \frac{\Gamma(b)}{\Gamma(a)\Gamma(b-a)} \Bigg[& \frac{i}{\omega} \sum_{j=1}^{N_{mesh}} w_j e^{\Re(z) i\frac{x_j}{\omega}} \left(i\frac{x_j}{\omega}\right)^{a-1} \left(1 - i\frac{x_j}{\omega}\right)^{b-a-1} \\ & - \frac{ie^{i\omega}}{\omega} \sum_{j=1}^{N_{mesh}} w_j e^{\Re(z)(1+i\frac{x_j}{\omega})} \left(1 + i\frac{x_j}{\omega}\right)^{a-1} \left(-i\frac{x_j}{\omega}\right)^{b-a-1} \Bigg] \end{aligned} \tag{43} $$

CF Beta distribution - new algorithm (3)



(a) *Integrand by adaptive quadrature.*



(b) *Integrand by steepest descent method.*

Figure: ${}_1F_1(5, 10, 100 - 1000i)$.

(a, b, z)	Mathematica	SD (N=64)	Rel.error
$(500, 510, 100 - 1000i)$	$-6.2228244518594344 \times 10^{39}$	$-6.2228244518589960 \times 10^{39}$	-8.10×10^{-14}
	$+6.5628468755115970 \times 10^{38}i$	$+6.5628468755089959 \times 10^{38}i$	$-8.54 \times 10^{-15}i$
$(700, 800, -10^5 i)$	$-1.2009316709301143 \times 10^{-113}$	$-1.2009316709306315 \times 10^{-113}$	-1.22×10^{-13}
	$+1.9151353691419210 \times 10^{-113}i$	$+1.9151353691419011 \times 10^{-113}i$	$-1.94 \times 10^{-13}i$
$(900, 1000, 100 - 10^5 i)$	$6.0430633105995064 \times 10^{-60}$	$6.0430633106045338 \times 10^{-60}$	3.73×10^{-13}
	$-6.8982723514693311 \times 10^{-60}i$	$-6.8982723514706421 \times 10^{-60}i$	$4.26 \times 10^{-13}i$
$(999, 1000, 700 - 10^5 i)$	$-4.6020552283174788 \times 10^{302}$	$-4.6020552283192628 \times 10^{302}$	-1.82×10^{-13}
	$-8.8660573516817291 \times 10^{302}i$	$-8.8660573516852506 \times 10^{302}i$	$3.51 \times 10^{-13}i$
$(999, 1000, -700 + 10^5 i)$	$-3.2868461235577501 \times 10^{-306}$	$-3.2868461235590394 \times 10^{-306}$	-1.32×10^{-13}
	$+9.2806995639762948 \times 10^{-306}i$	$+9.2806995639799548 \times 10^{-306}i$	$-3.72 \times 10^{-13}i$
$(900, 930, -10^{10} i)$	$-5.9703815795271339 \times 10^{-212}$	$-5.9703815795709398 \times 10^{-212}$	-1.12×10^{-11}

Table: Results by applying the numerical steepest descent method to compute the confluent hypergeometric function on the complex plane for large values of a, b and $|z|$.

Fourier transform method - Implementation

The implementation has two steps: **generation** of portfolio Fourier transform and **inversion** to obtain the portfolio loss distribution. The inversion step of the portfolio is performed by using Fast Fourier Transform (FFT).

1. computation of $\hat{f}(t)$ generates a vector of complex numbers.
2. given a Fourier resolution M the inversion is performed using the Discrete Inverse Fourier Transform (DIFT)

$$[f_0, f_1, \ldots, f_{M-1}] = \mathrm{DIFT}([\hat{f}(t_0), \ldots, \hat{f}(t_{M/2}), \overline{\hat{f}(t_{M(2-1)})}, \ldots, \overline{\hat{f}(t_1)}]) \tag{44}$$

where

$$\mathrm{DIFT}([\hat{f}(t_0), \ldots, \hat{f}(t_{M/2})]) = \frac{1}{M} \sum_{k=0}^{M/2} \hat{f}(t_k) e^{2\pi i k m/M}, \ m = 0, \ldots, M/2 \tag{45}$$

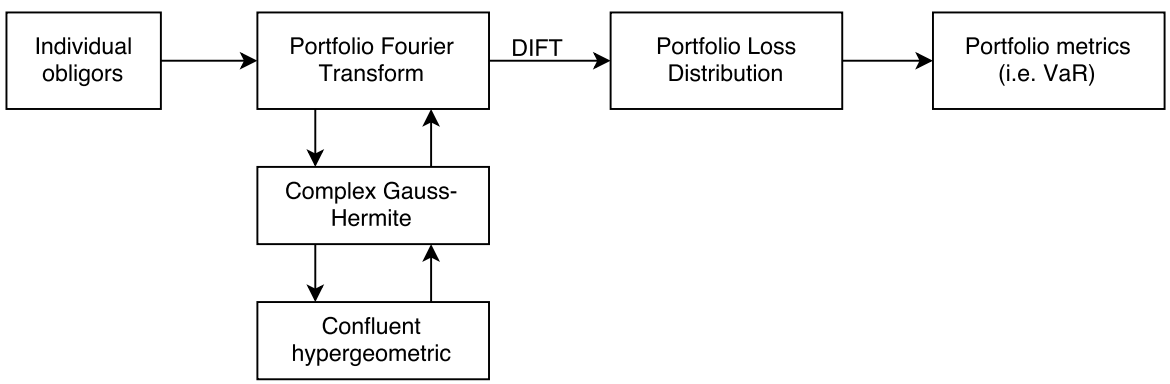


Figure: Steps of the FTM computation with Beta distributed LGDs.

The model framework (1)

Let F be the cumulative distribution function of L. Without loss of generality, we can assume $\sum_{n=1}^{N} E_n = 1$ and consider

$$F(x) = \begin{cases} \overline{F}(x) & \text{if } 0 \leq x \leq 1 \\ 1 & \text{if } x > 1 \end{cases} \tag{46}$$

for a certain $\overline{F}$ defined in $[0, 1]$. $\overline{F}$ can be approximated by a finite summation of scaling functions, Haar wavelets basis functions

$$\overline{F}(x) \approx \overline{F}_m(x) = \sum_{i=1}^{2^m - 1} c_{m,k} \phi_{m,k}(x) \tag{47}$$

where $\phi_{m,k}(x) = 2^{m/2} \phi(2^m x - k)$ and

$$\psi(x) = \begin{cases} 1 & \text{if } 0 \leq x < 1/2 \\ -1 & \text{if } 1/2 \leq x < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\phi(x) = \begin{cases} 1 & \text{if } 0 \leq x < 1 \\ 0 & \text{otherwise} \end{cases}$$

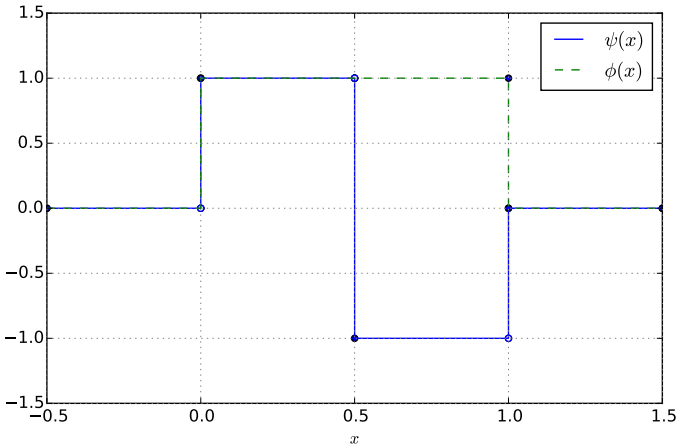


Figure: The Haar wavelet.

The model framework (2)

Assuming non-stochastic LGD, the unconditional MGF obtained by taking the expectation value of the conditional MGF is given by

$$M_L(s) = E[e^{-sL}] = E[M_L(s;Y)] = E\left[\prod_{n=1}^{N}[1 - p_n(y) + p_n(y)e^{-sE_n\Lambda_n}]\right]$$

$$= \int_{-\infty}^{\infty}\prod_{n=1}^{N}[1 - p_n(y) + p_n(y)e^{-sE_n\Lambda_n}]f(y)\,dy \tag{48}$$

Coefficients $c_{m,k}$

$$Q(z) \equiv \sum_{k=0}^{2^m-1} c_{m,k}z^k \approx \frac{M_L(-2^m\ln(z)) - z^{2^m}}{2^{m/2}(1-z)} \tag{49}$$

$$\lim_{z\to 0} Q(z) = c_{m,0} = \frac{\int_{-\infty}^{\infty}\prod_{n=1}^{N}[1 - p_n(y)]f(y)\,dy}{2^{m/2}} \tag{50}$$

$$c_{m,k} = \frac{2}{\pi r^k}\int_0^{\pi}\Re(Q(re^{iu}))\cos(ku)\,du \tag{51}$$

Computation of $\Re(Q(re^{iu}))$

$$Q(re^{iu}) = \frac{M_L(-2^m \ln(re^{iu}) - (re^{iu})^{2^m}}{2^{m/2}(1 - re^{iu})} \tag{52}$$

$$\Re(Q(re^{iu})) = \frac{\Re(z_1)\Re(z_2) + \Im(z_1)\Im(z_2)}{(\Re(z_2))^2 + (\Im(z_2))^2} \tag{53}$$

where

$$\Re(z_1) = \Re(M_L(-2^m \ln(re^{iu})) - r^{2^m} \cos(2^m u), \quad \Re(z_2) = 2^{m/2}(1 - r\cos(u)) \tag{54}$$

and

$$\Im(z_1) = \Im(M_L(-2^m \ln(re^{iu})) - r^{2^m} \sin(2^m u), \quad \Re(z_2) = -2^{m/2} r \sin(u) \tag{55}$$

$$M_L(-2^m \ln(re^{iu})) = \int_{-\infty}^{\infty} \prod_{n=1}^{N} [1 - p_n(y) + p_n(y) r^{2^m E_n \Lambda_n} e^{i 2^m E_n \Lambda_n u}] f(y)\, dy \tag{56}$$

$$\Re(M_L(-2^m \ln(re^{iu}))) \approx \frac{\pi}{2} h \sum_{j=1}^{N} w_j \Re(\overline{M_L}(-2^m \ln(re^{iu}); x_j)) \tag{57}$$

$$\Im(M_L(-2^m \ln(re^{iu}))) \approx \frac{\pi}{2} h \sum_{j=1}^{N} w_j \Im(\overline{M_L}(-2^m \ln(re^{iu}); x_j)) \tag{58}$$

Computation of the coefficients $c_{m,k}$

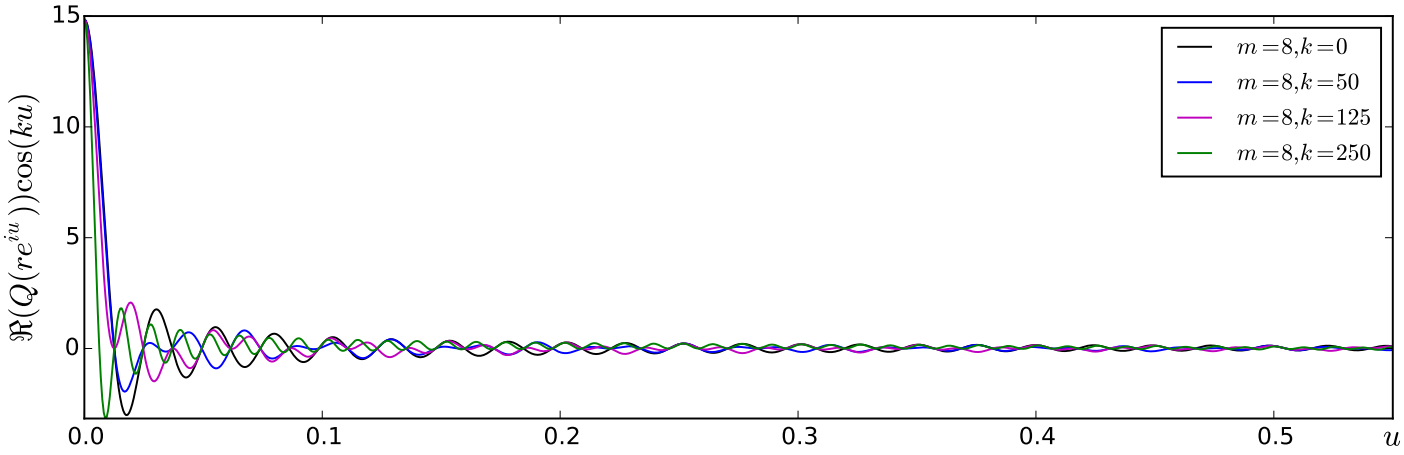


Figure: Plot function $\Re(Q(re^{iu}))\cos(ku)$ in $u = [0, \pi]$.

The extended trapezoidal rule stands as the fastest method with remarkable accuracy for this kind of integrals.

Method	iterations	result	Rel.Error	Ratio
QAWO (GSL)	108	0.052649594467716	-	8
QAG GK41(GSL)	16	0.052649594467716	-	3.5
Trapezoidal	256	0.052649557000781	7.12×10^{-7}	2
Ext. Trapezoidal(10^{-4})	9	0.052649589486008	9.46×10^{-8}	1

Table: Methods for computing $c_{m,k}$.

Numerical examples (1)

Portfolio 1. This portfolio has $N = 1001$ obligors with $p_n = 0.0033$, $\rho_n = 0.2$ and $E_n = 1$ for $n = 1, \ldots, N-1$ and $E_{1001} = 100$.

Scale	$VaR_{0.999}$	Rel.Error	Time(s)	$VaR_{0.9999}$	Rel.Error	Time(s)
$m = 10$	0.106934	0.71%	27.61	0.153809	0.40%	29.36
$m = 11$	0.108154	0.42%	56.76	0.153076	0.08%	64.86
FTM	0.107400	0.28%	58.50	0.152861	0.22%	58.50

Table: VaR of portfolio 1 for $m \in \{10, 11\}$. Relative error with respect to Monte Carlo with 5 millions scenarios.

Portfolio 2. This portfolio has $N = 1000$ obligors with $p_n = 0.003$, $\rho_n = 0.15$ and $E_n = 1/n$ for $n = 1, \ldots, N$.

Scale	$VaR_{0.999}$	Rel.Error	Time(s)
$m = 8$	0.138672	1.30%	6.77
$m = 9$	0.139648	0.61%	15.00
$m = 10$	0.140137	0.26%	30.93
$m = 11$	0.140381	0.08%	53.91
$m = 12$	0.140503	0.00%	110.23

Table: VaR of portfolio 2 for $m \in \{8, 9, 10, 11, 12\}$. Relative error with respect to Monte Carlo with 5 millions scenarios.

Numerical examples (2)

Portfolio 3. This portfolio has $N = 10000$ obligors with $p_n = 0.01$, $\rho_n = 0.15$ and $E_n = 1/n$ for $n = 1, \ldots, N$.

Scale	$VaR_{0.999}$	Rel.Error	Time(s)
$m = 8$	0.162109	0.25%	82.76
$m = 9$	0.163086	0.86%	164.13
$m = 10$	0.161621	0.05%	388.36

Table: VaR of portfolio 3 for $m \in \{8, 9, 10\}$. Relative error with respect to Monte Carlo with 5 millions scenarios.

Portfolio 4. This portfolio has $N = 1000$ obligors with $p_n = [0.001, 0.01, 0.1, 0.2]$, $\rho_n = [0.02, 0.1, 0.2, 0.5]$ and $E_n = 1$ for $n = 1, \ldots, N$.

$p \setminus \rho$	0.02	0.1	0.2	0.5
0.001	0.007324 **(4.13%)**	0.015137 **(0.15%)**	0.028809 **(3.53%)**	0.087402 **(17.07%)**
0.01	0.032715 **(0.57%)**	0.079590 **(0.81%)**	0.142090 **(3.94%)**	0.386230 **(13.05%)**
0.1	0.202637 **(0.31%)**	0.374512 **(0.85%)**	0.536621 **(1.99%)**	0.884277 **(2.11%)**
0.2	0.347168 **(0.34%)**	0.556152 **(0.76%)**	0.720215 **(1.30%)**	0.968262 **(0.42%)**

Table: VaR value at 99.9% for portfolio 4. Relative error with respect to Monte Carlo using Median Latin Hypercube Sampling and applying tan transformation.

Conclusions

- The generalization of the Vasicek model increases the flexibility.
- Fourier transform method very competent for small/medium portfolios.
- Haar-wavelet approximation good performance for large portfolios.
- Advanced numerical methods are required.
- There is no ideal method for all portfolios.

Numerical software

Most of contents of this thesis is oriented towards an efficient implementation of numerical methods and computational models in portfolio credit risk.

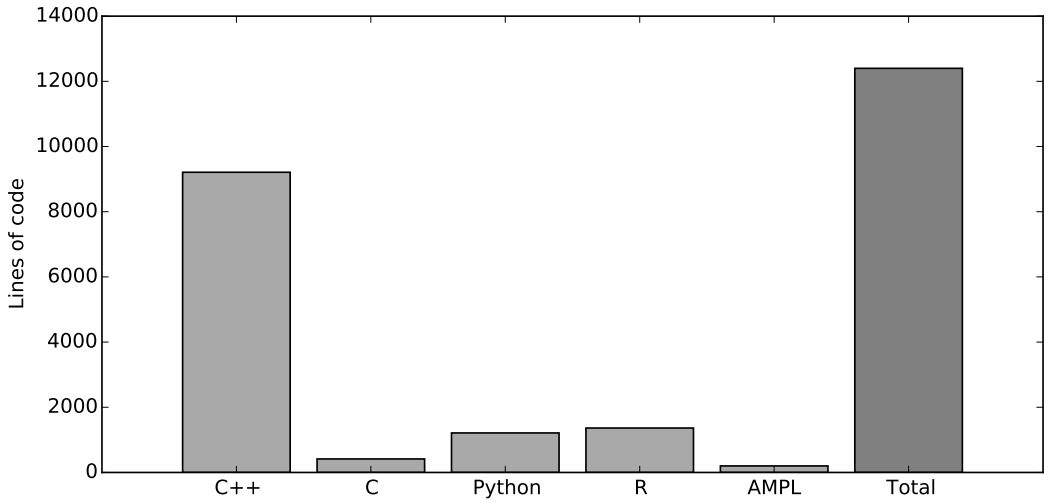


Figure: Lines of code for each programming language. Total = 12399 lines.

The total project is over 12000 lines of source code. The main code in C++ has been coded following a software development process called Test-driven development (TDD).

About me

- **Moody's Investors Service**, Frankfurt 06/2013 - 05/2014
  - Associate in Structured Finance Group - Technology team.
- **Numerical Algorithms Group (NAG)**, Oxford 11/2015 - Present
  - Numerical Software Developer in mathematical optimization.
- **Universitat Politècnica de Catalunya** 09/2015 - Present
  - PhD in Computing. Working on the efficient computation of special functions.

Questions

Questions?

Thank you