Economic Scenarios Generation for Insurance: `ESGtoolkit` (and friends)

Institut de Sciences Financières et d'Assurances, Université Lyon 1

Thierry Moudiki (@moudikithierry)

September 26, 2014

Contents

Why ?

- **As an insurer**, basically :
  - How much should I keep aside today at $t = 0$, for the payment of a **guaranteed** compensation tomorrow at $t = T$:

$$Liabs_0 = BestEstimateLiabs_0 + Margin_0$$

$\Longrightarrow$ **Reserving** : **pricing** the guaranteed compensation

  - Having calculated $Liabs_0$, I'd like to derive $x_\alpha > 0$, so that :

$$\mathbb{P}(Assets_T - Liabs_T > x_\alpha) \geq 1 - \alpha\%$$

$\Longrightarrow$ **Capital modeling** : determining the **future distribution** of my Own Funds

Why ? (cont'd)

Regulatory point of view

- **Market Consistent Embedded value (MCEV) :** Time Value of Financial Options & Guarantees (TVOFG)
- **Solvency II :
  - ** Best Estimate valuation of the technical reserves (BEL)
  - **Solvency II :** Own Risk Solvency Assessment (ORSA)

But :

- For **pricing (reserving)** : not always **closed formulas** available for pricing the guarantees (risk asymmetry induced by the optional features)
- For **capital modeling** : not always (never !) entirely-specified probability distributions available for the Own Funds

Context

What ?

From what's been said : **Modeling and simulation of risk factors are needed.**

ESG : Economic Scenarios Generator

- Tool for **modeling** and **simulation** of economic factors' future values
- **Purpose :** Asset & Liability Management in Banking and Insurance.
- **Our focus :** Insurance

How ?

How can we do that ?

- With **Monte Carlo simulation**
- But also, **Bootstrapping** can be used

Generally, 2 types of simulations

- **Real-world simulations** under the objective probability : for **capital modeling**
- **Risk-neutral simulations** under a martingale probability measure : for **reserving/pricing**

The package ESGtoolkit

- An **R** package providing tools, for constructing custom Economic Scenario Generators (ESG)
- Version `0.1` released in june 2014
- A vignette is available (now in PDF, in HTML soon)
- Under development
- **Suggestions, bug reports, and features request** are welcome.

Why **R** ?

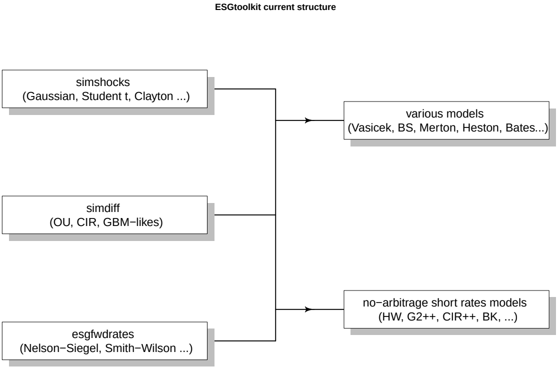
- The **language of Analytics**; 2 million users worldwide (Source : *Seven quick facts about R*)
- Free + Open source
- Over **5800** contributed packages on CRAN repository, doing almost anything you could think of (Visit : *the Task Views*)
- "***Researchers in statistics and machine learning will often publish an R package to accompany their articles. This means immediate access to the very latest statistical techniques and implementations.***" Hadley Wickham (RStudio)

Contents

ESGtoolkit

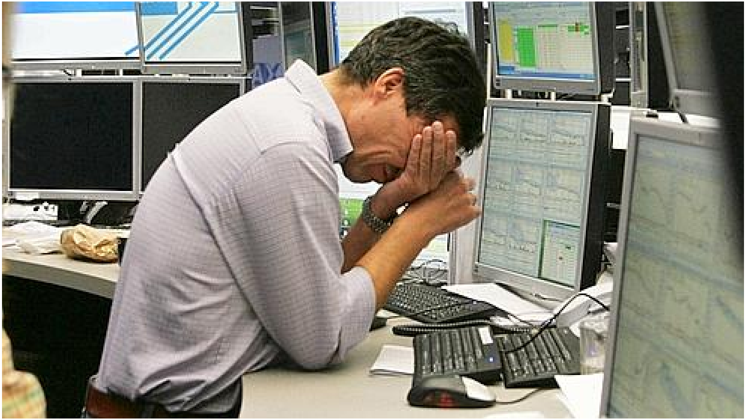
- **Currently, 3 main functions**
  - `simdiff` (underlying C++ code *via* Rcpp) :
    - Ornstein-Uhlenbeck process simulation
    - Cox-Ingersoll-Ross process simulation
    - Geometric Brownian motion with constant or time-dependent drift or volatility, and optional (lognormal or double-exponential) jumps
  - `simshocks` :
    - simulation of gaussian shocks with highly flexible dependence structure
  - `esgfwdrates` :
    - instantaneous forward rates for no-arbitrage short rate models
- **And additional functions for diagnostics** : `esgplotbands`, `esgplotshocks`, `esgplotmartingaletest`, ...

ESGtoolkit


ESGtoolkit current structure
simshocks
(Gaussian, Student t, Clayton ...)
simdiff
(OU, CIR, GBM−likes)
esgfwdrates
(Nelson−Siegel, Smith−Wilson ...)
various models
(Vasicek, BS, Merton, Heston, Bates...)
no−arbitrage short rates models
(HW, G2++, CIR++, BK, ...)

However, remember

- "*Essentially, all models are wrong, but some are useful*" George E. P. Box
- ... also, some can lead to disasters.



From Felix Salmon's *Recipe for Disaster: The Formula That Killed Wall Street* (available online)

$$\Pr[T_A < 1, T_B < 1] = \Phi_2(\Phi^{-1}(F_A(1)), \Phi^{-1}(F_B(1)), \gamma)$$

Here's what killed your 401(k) *David X. Li's Gaussian copula function as first published in 2000. Investors exploited it as a quick—and fatally flawed—way to assess risk. A shorter version appears on this month's cover of* Wired.

Probability
Specifically, this is a joint default probability—the likelihood that any two members of the pool (A and B) will both default. It's what investors are looking for, and the rest of the formula provides the answer.

Survival times
The amount of time between now and when A and B can be expected to default. Li took the idea from a concept in actuarial science that charts what happens to someone's life expectancy when their spouse dies.

Equality
A dangerously precise concept, since it leaves no room for error. Clean equations help both quants and their managers forget that the real world contains a surprising amount of uncertainty, fuzziness, and precariousness.

Copula
This couples (hence the Latinate term copula) the individual probabilities associated with A and B to come up with a single number. Errors here massively increase the risk of the whole equation blowing up.

Distribution functions
The probabilities of how long A and B are likely to survive. Since these are not certainties, they can be dangerous: Small miscalculations may leave you facing much more risk than the formula indicates.

Gamma
The all-powerful correlation parameter, which reduces correlation to a single constant—something that should be highly improbable, if not impossible. This is the magic number that made Li's copula function irresistible.

However, remember

3 types of risks in the process of modeling and simulation of risk factors :

- **Model risk**
  - Choice of the model
  - Choice of the dependence structure
- **Discretization** error
- **Calibration** error

Contents

About model risk

Choice of the model and choice of the dependence structure

A simple example : Insurance company, ABC corp., that has, on December 30, 2011, **2 assets in its portfolio** :

$$A_0 = Nominal \times (40\% CAC\,40 + 60\% S\&P\,500)$$

The liabilities on December 30, 2011 are :

$$L_0 = 45\% \times A_0$$

ABC corp. has to pay daily **guaranteed benefits** K to the insured, plus an **additional benefits** depending on the daily performance of the assets :

$$\forall i > 0,\ L_i = L_{i-1} - K \times \left(1 + 95\% max\left(\frac{A_i}{A_{i-1}} - 1, 0\right)\right)$$

Analyst wants to assess ABC corp.'s 6-month solvency and uses R

```r
# loading the package quantmod
# to obtain financial index time series
library(quantmod)
```

```r
# Importing the values of the CAC40 and S&P500
# CAC40, as a time series (ts) object
getSymbols('^FCHI', src='yahoo',
           return.class = 'ts',
           from = "2011-06-30",
           to = "2011-12-30")
# S&P500, as a time series (ts) object
getSymbols('^GSPC', src='yahoo',
           return.class = 'ts',
           from = "2011-06-27",
           to = "2011-12-30")
```

The type of data we can get from quantmod

```r
head(FCHI)[, 1:4]
```

```
##        FCHI.Open FCHI.High FCHI.Low FCHI.Close
## [1,]        3937      3982     3926       3982
## [2,]        3982      4024     3967       4007
## [3,]        4010      4010     3997       4003
## [4,]        3997      3999     3974       3979
## [5,]        3981      3982     3942       3961
## [6,]        3982      4020     3960       3980
```

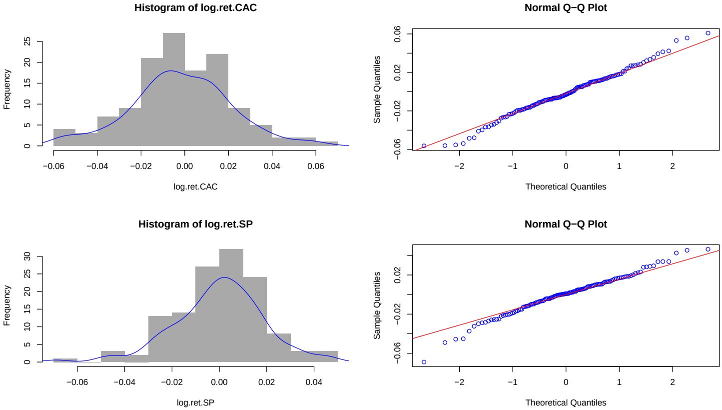
- Using the closing prices $S_t^{(CAC)}$, $S_t^{(SP)}$ analyst calculates **the daily log-returns**

$$log\left(\frac{S_{t_{i+1}}^{(CAC)}}{S_{t_i}^{(CAC)}}\right)$$

and

$$log\left(\frac{S_{t_{i+1}}^{(SP)}}{S_{t_i}^{(SP)}}\right)$$

Normality of the log-returns ?



Normality of the log-returns ? (cont'd)

```
shapiro.test(log.ret.CAC)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log.ret.CAC
## W = 0.9906, p-value = 0.5268
```

```
shapiro.test(log.ret.SP)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log.ret.SP
## W = 0.9827, p-value = 0.0962
```

Maximum likelihood estimation

- Analyst **assumes that** :
  - **the distribution of the assets over the next 6 months will be the same that she observed in the last 6-months period**.

Maximum likelihood is used to calibrate the lognormal models :

```
# Parameters for the projection of the CAC 40
delta <- 1/252 # for daily sampling
sigma.CAC <- sqrt((n-1)/n)*sd(log.ret.CAC)/sqrt(delta)
mu.CAC <- mean(log.ret.CAC)/delta +
  0.5*sigma.CAC^2
```

Correlation ?

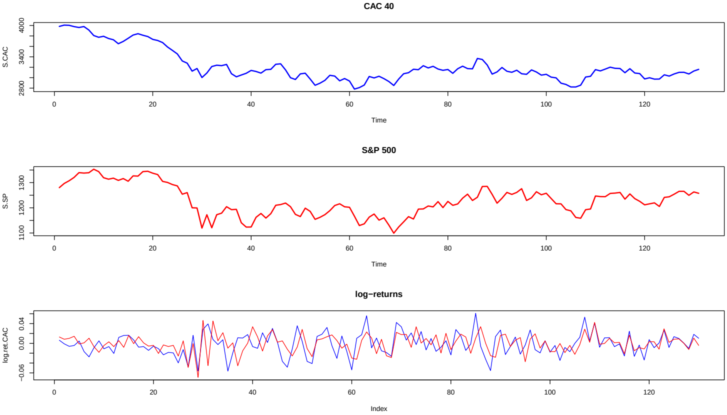
- Based on :
    - the histograms
    - the Normal qqplot
    - the results of the tests (Normality not rejected)

ABC corp.'s analyst **assumes that the distribution of the assets on the period of interest is lognormal**.

- But she now wants to know more about the **dependence between the assets**

Correlation ? (cont'd)

- Visualizing the indices on the 6-month period, and the log-returns



Correlation ? (cont'd)

- She decides to
  - **assume that the assets are correlated (Gaussian dependence)**.

The correlation coefficients between the shocks is :

```
U.S.CAC <- pnorm((log.ret.CAC - mean(log.ret.CAC))
                  /sd(log.ret.CAC))
U.S.SP <- pnorm((log.ret.SP - mean(log.ret.SP))/
                  sd(log.ret.SP))
(correlation <- cor(U.S.CAC, U.S.SP))
```

```
## [1] 0.3852
```

Correlation ? (cont'd)

However. . .

```
# Package CDVine : Statistical inference of
# canonical vine (C-vine) and D-vine copulas
library(CDVine)
copula.selection <- BiCopSelect(U.S.CAC, U.S.SP)
print(c(copula.selection$family, copula.selection$par,
        copula.selection$par2))
```

```
## [1] 2.0000 0.4227 9.1056
```

- **R** package `CDVine` (see jstatsoft paper) says :
  - **Student t dependence**, with 9.11 degrees of freedom, and dependence parameter equal to 0.42

- Simulation of shocks with **ESGtoolkit**

```r
library(ESGtoolkit)
# Nb of simulations, horizon, and frequency
nb <- 10000
horizon <- 1
freq <- "daily"
# Simulation of shocks
set.seed(1)
# With student dependence
eps.student <- simshocks(n = nb,
horizon = horizon, frequency = freq,
family = copula.selection$family,
par = copula.selection$par,
par2 = copula.selection$par2)
```

- Simulation of future values of assets with `simdiff`

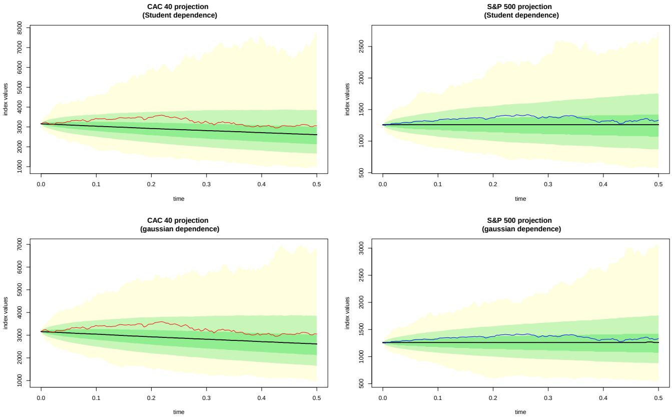
Simulation with Gaussian dependence is exactly the same, but uses `eps.gaussian` for the shocks

```
# With Student dependence
# CAC 40
sim.CAC.t <- window(simdiff(n = nb,
horizon = horizon, model = "GBM",
frequency = freq, x0 = S.CAC[n],
theta1 = mu.CAC, theta2 = sigma.CAC,
eps = eps.student[[1]]), end = 0.5)
# S&P 500
sim.SP.t <- window(simdiff(n = nb,
horizon = horizon,
model = "GBM",
frequency = freq, x0 = S.SP[n],
theta1 = mu.SP, theta2 = sigma.SP,
eps = eps.student[[2]]), end = 0.5)
```

- Visualizing the projections of assets' values with `esgplotbands`

```
par(mfrow = c(2, 2))
esgplotbands(sim.CAC.t, xlab = "time",
             ylab = "index values")
lines(S.CAC.future, col = "red")
esgplotbands(sim.SP.t, xlab = "time",
             ylab = "index values")
lines(S.SP.future, col = "blue")
esgplotbands(sim.CAC.g, xlab = "time",
             ylab = "index values")
lines(S.CAC.future, col = "red")
esgplotbands(sim.SP.g, xlab = "time",
             ylab = "index values")
lines(S.SP.future, col = "blue")
```

- Visualizing the projections of assets' values with `esgplotbands`. The true values (from January to June 2012) are the red lines and the blue lines.



- Assets and Liabilities simulation for the Net Asset Value (the whole code is available)

```
### The insurer's Asset
w1 <- 0.4
w2 <- 1-w1
Nominal <- 100000
# With gaussian dependence
S.g <- Nominal*(w1*sim.CAC.g + w2*sim.SP.g)
# With Student t dependence
S.t <- Nominal*(w1*sim.CAC.t + w2*sim.SP.t)

### The insurer's liability
# With gaussian and Student t dependence (at t=0)
L0.g <- L0.t <- S.g[1,1]*0.45
# Guaranteed capital
K <- 80
# Pct. for additional benefits
pct.PB <- .95
```

- Assets and Liabilities simulation for the Net Asset Value

```
# variation of liabilities
fact.growth.t <- -K*(1 + pct.PB*(S.t.mat[-1, ]/
                                   S.t.mat[-nrowS, ] - 1)*
                   (S.t.mat[-1, ]/S.t.mat[-nrowS, ] - 1 > 0))

# Liabilities simulation (Student)
L.t <- ts(apply(rbind(rep(L0.t, ncolS), fact.growth.t), 2,
                cumsum),
   start = start(S.t), deltat = deltat(S.t))

# Net Asset Value
NAV.t <- S.t - L.t
```

- Estimating the 6-month required capital with Gaussian and Student dependences (the whole code is available)

```
# VaR Gaussian vs Student dep.
qt.NAV.t <- quantile(NAV.t.last,
         probs = 1 - c(0.9, 0.95, 0.995))
(qt.NAV.t/qt.NAV.g - 1)*100
```

```
##      10%       5%     0.5%
## -1.9779 -1.4868 -0.4514
```

```
# TVaR Gaussian vs Student dep.
TVaR.NAV.t <- mean(sort(NAV.t.last)[1:50])
(TVaR.NAV.t/TVaR.NAV.g - 1)*100
```

```
## [1] -5.767
```

But still. . .

- Historical stock prices are **NOT** lognormal.
- Even though it's based on real data, this was a "nice" example.
- Try other models : Heston model (stochastic volatility) or Bates model (stochastic volatility with jumps diffusion)
- **Example :** The package's vignette explains how to make simulations of Bates model
- But :
  - **calibration** must be carried out with **care**
  - **validation** of statistical properties as well

Contents

About discretization error

- Example

A Stochastic Differential Equation for an equilibrium short rate model (Vasicek model's SDE) :

$$dr_t = a(\theta - r_t)dt + \sigma dW_t$$

A simple and intuitive way to make simulations of the SDE : **Euler scheme** (1st order Ito development) :

- Deterministic mean-reverting part

$$r_{t_{i+1}} - r_{t_i} = a(\theta - r_{t_i})(t_{i+1} - t_i) + \dots$$

- ... A part with random shocks ($\epsilon \sim \mathcal{N}(0,1)$)

$$\sigma \epsilon \sqrt{t_{i+1} - t_i}$$

About discretization error (cont'd)

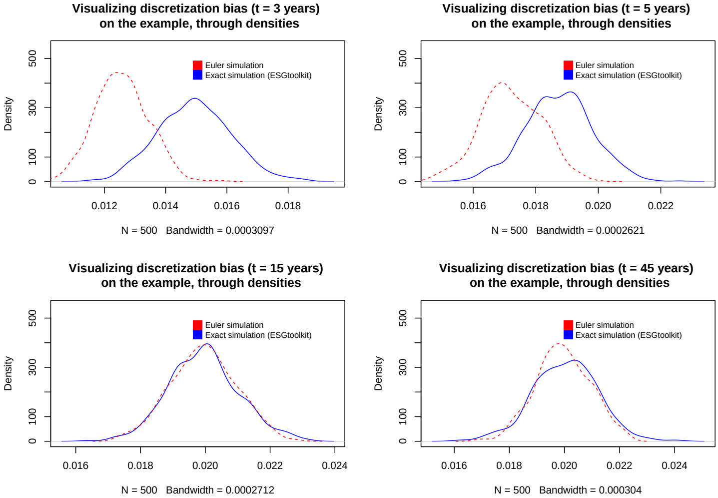
- Other method for the simulation of SDE : **Milstein scheme** (2nd order Ito development). But when e.g the volatility σ is constant, it's not necessary. Otherwise, leads to more complicated formulas.
- Euler and Milstein schemes imply **discretization bias**.
- Currently in **ESGtoolkit**, we use **exact simulation** of the transition distribution between t_{i+1} and $t_i \implies$ **No discretization bias**.

Here, using the SDE's exact solution provides the following **alternative scheme** :

$$r_{t_{i+1}} = e^{-a(t_{i+1}-t_i)} r_{t_i} + \theta(1 - e^{-a(t_{i+1}-t_i)}) + \sigma\epsilon\sqrt{\frac{1 - e^{-a(t_{i+1}-t_i)}}{2a}}$$

- Illustration using **ESGtoolkit** (`set.seed(1)`)

```
m <- 50 # number of projection dates
n <- 500 # number of simulations
a <- 0.5 # speed of mean-reversion
theta <- 0.02 # long term rate, mean-rev. level
sigma <- 0.001 # volatility

# Short rate with Euler discretization
r.Euler <- matrix(0, nrow = m, ncol = n)
for (i in 1:(m-1))
{r.Euler[i+1, ] <- r.Euler[i, ] + a*(theta -
r.Euler[i, ]) + sigma*rnorm(n)}

# Short rate with Exact simulation (ESGtoolkit)
r.Exact <- simdiff(n = n, horizon = 50,
                   model = "OU",
              x0 = 0, theta1 = a*theta,
              theta2 = a, theta3 = sigma)
```


Visualizing discretization bias (t = 3 years) on the example, through densities
Euler simulation
Exact simulation (ESGtoolkit)
Density
0
100
300
500
0.012
0.014
0.016
0.018
N = 500 Bandwidth = 0.0003097
Visualizing discretization bias (t = 5 years) on the example, through densities
Euler simulation
Exact simulation (ESGtoolkit)
Density
0
100
300
500
0.016
0.018
0.020
0.022
N = 500 Bandwidth = 0.0002621
Visualizing discretization bias (t = 15 years) on the example, through densities
Euler simulation
Exact simulation (ESGtoolkit)
Density
0
100
300
500
0.016
0.018
0.020
0.022
0.024
N = 500 Bandwidth = 0.0002712
Visualizing discretization bias (t = 45 years) on the example, through densities
Euler simulation
Exact simulation (ESGtoolkit)
Density
0
100
300
500
0.016
0.018
0.020
0.022
0.024
N = 500 Bandwidth = 0.000304

About discretization error (cont'd)

- **Comparing the implied zero-coupon prices**

```
price.Euler <- rowMeans(exp(-apply(r.Euler, 2,
                                    cumsum)))
price.Exact <- rowMeans(exp(-apply(r.Exact, 2,
                                    cumsum)))

(head(price.Exact)/head(price.Euler) - 1)*100
```

```
## [1] 0.0000 0.2071 0.4452 0.6392 0.7936 0.9002
```

Contents

Calibration (the Art of)

- Choice of **projection models parameters** using **market data**
- **Real world**
  - Methods of moments (Generalized, Simulated)
  - Maximum likelihood (or Quasi-Maximum likelihood, Normal approximation, Simulated maximum likelihood)
  - Non-linear filtering
- **Market Consistent** (Risk-neutral)
  - Minimizing (for N given financial instruments)

$$\sum_{i=1}^{N} w_i g(Price_0^{(i,Model)}(\Theta) - Price_0^{(i,Market)})$$

  - With model parameters $\Theta \in \mathbb{R}^d$, weights $(w_i)_{i=1,\ldots,N}$ and for example :
    - $g : x \mapsto x^2$
    - $g : x \mapsto |x|$

Focus on market consistent calibration

CRO Forum, in *Extrapolation of Market Data (2010)*

- *The complexity of the stochastic model used to value embedded options and guarantees is directly linked to the complexity of the underlying insurance contract*
- *The complexity of the model used should take into account the complexity of the liability and its embedded equity guarantees [...]. The calibration of any model should at least consider the at-the-money term-structure and for more complex models (e.g. Heston) should also consider the full volatility surface of the liquid part of the volatility market. This then automatically implies extrapolated volatilities for in- and out-the-money volatilities in the extrapolated part of the curve.*

Focus on market consistent calibration (cont'd)

- **Example :** G2++ model (2-factor Hull & White) calibrated to ATM Euro Caps on December 31, 2011

$$dx_t = -ax_t + \sigma dW_t^{(x)}$$

$$dy_t = -by_t + \eta dW_t^{(y)}$$

$$dW_t^{(x)} dW_t^{(y)} = \rho dt$$

$$r_t = x_t + y_t + \Phi_t$$

  - 5 parameters to find : a, b, σ, η, ρ
  - Objective function with many local optima
  - Optimization with **R** package **mcGlobaloptim**
    - Monte Carlo simulation of multiple starting points in a given region
    - Running local optimizations
    - Finding the best parameters

Focus on market consistent calibration (cont'd)

```r
# Parameters found for the G2++
a_opt <- 0.50000000
b_opt <- 0.35412030
sigma_opt <- 0.09416266
rho_opt <- -0.99855687
eta_opt <- 0.08439934

horizon <- 20
n <- 500
freq <- "semi-annual"
delta_t <- 1/2

# Simulation of gaussian correlated shocks
eps <- simshocks(n = n, horizon = horizon,
                 frequency = freq,
                 method = "anti",
                 family = 1, par = rho_opt)
```

- **Simulation of the model factors**

```
x <- simdiff(n = n, horizon = horizon,
             frequency = freq,
             model = "OU",
             x0 = 0, theta1 = 0, theta2 = a_opt, theta3 = sig
             eps = eps[[1]])
y <- simdiff(n = n, horizon = horizon,
             frequency = freq,
             model = "OU",
             x0 = 0, theta1 = 0, theta2 = b_opt, theta3 = eta
             eps = eps[[2]])
```

- **Forward rates and final model**

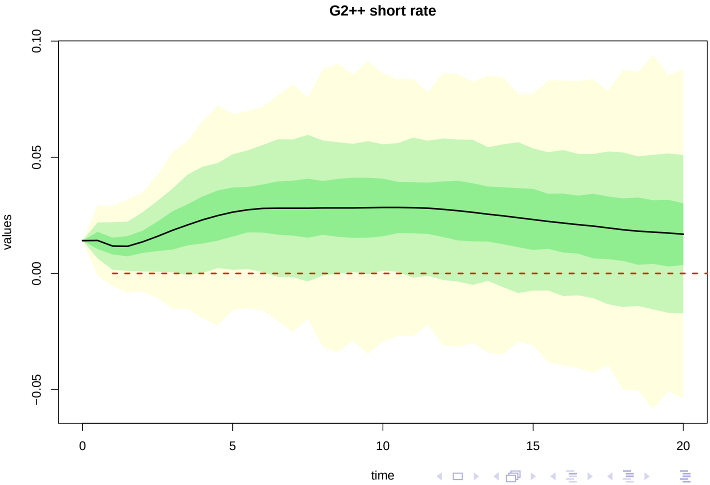
```
fwdrates <- esgfwdrates(n = n, horizon = horizon,
out.frequency = freq, in.maturities = u,
in.zerorates = txZC, method = "SW")
fwdrates <- window(fwdrates, end = horizon)

t.out <- seq(from = 0, to = horizon,
             by = delta_t)

param.phi <- 0.5*(sigma_opt^2)*(1 -
exp(-a_opt*t.out))^2/(a_opt^2) +
0.5*(eta_opt^2)*(1 - exp(-b_opt*t.out))^2/
  (b_opt^2) +
(rho_opt*sigma_opt*eta_opt)*(1 - exp(-a_opt*t.out))*
  (1 - exp(-b_opt*t.out))/(a_opt*b_opt)
```

- **Forward rates and final model (cont'd)**

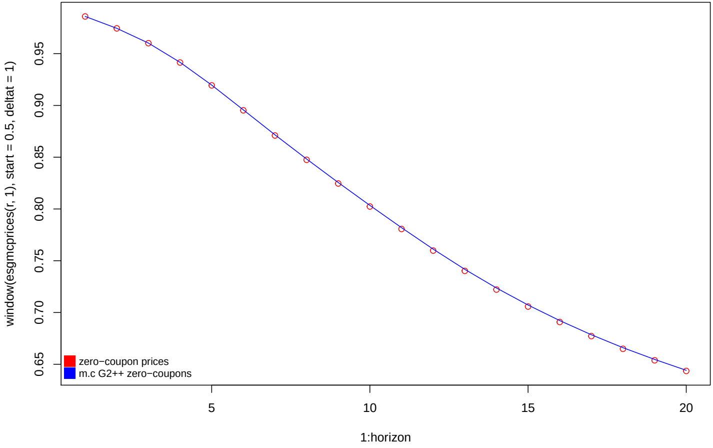
```
param.phi <- ts(replicate(n, param.phi),
                start = start(x), deltat = deltat(x))

phi <- fwdrates + param.phi
colnames(phi) <- c(paste0("Series ", 1:n))

r <- x + y + phi
colnames(r) <- c(paste0("Series ", 1:n))
```

- The result

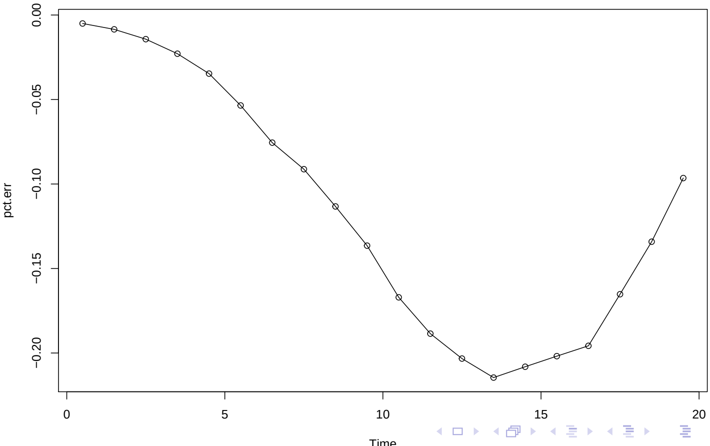
```r
esgplotbands(r, xlab = "time", ylab = "values",
             main = "G2++ short rate")
lines(rep(0, ncol(r)), col = "red", lty = 2, lwd = 2)
```



- Monte Carlo prices of zero-coupons vs true prices



```r
# Checking error manually
pct.err <- (window(esgmcprices(r, 1),
start = 0.5, deltat = 1)/p[1:horizon] - 1)*100
plot(pct.err)
points(pct.err)
```



One further step is required

- Verifying by simulation that the discounted Caps payoffs are martingales (**Market consistency test**)
- Typically, a Student t-test
- An example of **Market consistency test** can be found in the package vignette.