Equity/Interest-rates ESGs - Implementation, Statistical analysis & Parameter influence

Final project presentation - Cutting-Edge project

Amal BACHA - Dalia BARBI - Khalil BATTIKH - Lucas RODRIGUEZ - Naïm SOUNI

Group 2

May 16th, 2023

Master in Quantitative Finance (M2QF)

⇝ Overview of academic deliverables

As of today, 6 documents :

1. Technical **report**
2. **Slides**
3. **GitHub repository**[1] [2]
4. **Online project homepage**[3]
5. **Online documentation**[4]
6. **Architecture charts** (UML classes & packages)

1. https://github.com/lcsrodriguez/CuttingEdge-Milliman
2. https://github.dev/lcsrodriguez/CuttingEdge-Milliman
3. https://lcsrodriguez.github.io/qf/cutting-edge/
4. https://lcsrodriguez.github.io/qf/cutting-edge/doc/

Quick outline

1. Introduction, Project's assumptions & General Framework
2. Implementation of :
   - Interest rates
   - Equity prices
3. In-depth study of the impact of parameters (model & simulation)
4. Conclusion, Critiques & Further extensions

Introduction

Problem definition

Context

- Financial market $\mathcal{M}$ on $(\Omega, \mathcal{F}, \mathbb{F} := (\mathcal{F}_t)_{t \in \mathbb{R}^+}, \mathbb{P})$
- Continuous-time modeling

Objectives

- State-of-the-art of existing Python libraries
- Implementation of several ESGs for interest rates & equity prices
- Implementation of European and Asian pricers
- Parameters influence study & Statistical analysis
- Final analysis & Conclusions

Constraints

- Python & Jupyter Notebook
- OOP architecture
- Adoption of a highly-professional framework

Financial framework

Hypothesis of research project

1. Underlying perfectly divisible
2. Friction-less market
3. No calibration [5]

5. Possible extension of the current project

Technical framework
Development environment

General informations

- Development : **Python 3.10+**
- Environment : **Jupyter Notebook** *(local, Google Colab, Kaggle)*
- Dependencies tracking : **pip**[6] & **Dependabot**
- Version control : **Git/GitHub**[7]
- LaTeX report writing
- Data handling & Numerical analysis : **NumPy**, **Pandas** & **PyArrow**
- Plotting : **Matplotlib**, **Pyplot** & **Seaborn**
- UML & Class diagram **Pyreverse**
- CI/CD workflow : GitHub Actions *UML, Dependabot, release*

$\Longrightarrow$ Professional development framework for **best implementation quality**

6. See complete list of dependencies on GitHub
7. GitHub project repo : https://github.com/lcsrodriguez/CuttingEdge-Milliman

Technical framework

Simplified UML graph & Class diagram

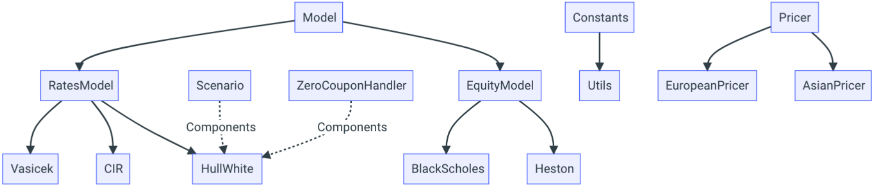


Figure – Simplified class diagram of the project [8]

Complete diagrams [9]

- UML class diagram
- UML package diagram

8. As of May 14th (*final version*)
9. See https://github.com/lcsrodriguez/CuttingEdge-Milliman

Implementation : Interest rates

Considered models

- Vasicek
- Cox-Ingersoll-Ross (CIR)
- Hull & White — As a refinement of Vasicek

Extensions : Ho-Lee, Black-Karakinski [10]

Implementation : 1 class for 1 model, all depending from the abstract class `RatesModel`

Simulation : Euler-Maruyama & Milstein numerical schemes

10. Used for hyprid credit/rates models

Implementation : Equity rates

Considered models

- Black-Scholes — Stochastic rates, constant diffusion
- Heston — As a refinement of BS

Implementation : 1 class for 1 model, all depending from the abstract class `EquityModel`

Simulation : Euler-Maruyama & Milstein numerical schemes Same as rates simulation

Implementation : Simulation of $k \geq 2$ Brownian motions (1/3)

Context :

- Simulation of several correlated Brownian motions for each combination
- In this project, $\boxed{k \in \{2, 3\}}$

Solution : $\longrightarrow$ Use of **Cholesky technique**

Example

$$\Sigma := \begin{bmatrix} \rho_{11} & \rho_{12} & \rho_{13} \\ \rho_{21} & \rho_{22} & \rho_{23} \\ \rho_{31} & \rho_{32} & \rho_{33} \end{bmatrix} = \begin{bmatrix} 1 & \rho_{21} & \rho_{31} \\ \rho_{21} & 1 & \rho_{32} \\ \rho_{31} & \rho_{32} & 1 \end{bmatrix} \in \mathcal{S}_3^{++} \tag{1}$$

where : $\forall (i, j) \in \{1, 2, 3\}^2,\ \rho_{ij} := \mathrm{Cov}(W^i, W^j)$

Implementation : Simulation of $k \geq 2$ Brownian motions (2/3)

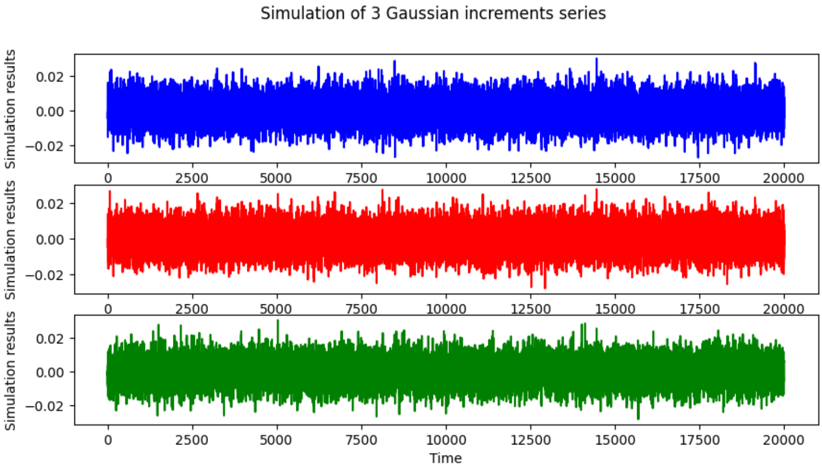


Figure – Brownian increments for $k = 3$ and a given $\Sigma \in \mathcal{S}_3^{++}$

Implementation : Simulation of $k \geq 2$ Brownian motions (3/3)

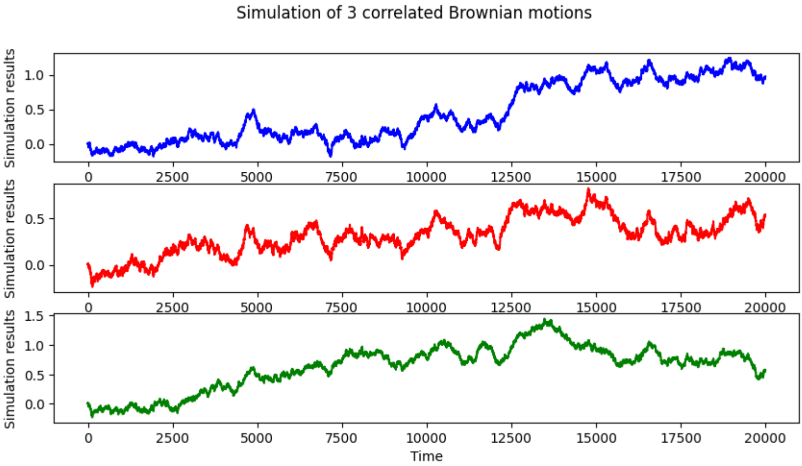


Figure – Brownian cumulative sums $\sim$ BMs trajectories for $k = 3$ and a given $\Sigma \in \mathcal{S}_3^{++}$

Implementation : European & Asian option pricing (1/2)

Context : Option prices $\rightsquigarrow$ Relevant analysis

Studies : Implementation of 2 pricers

- **European**

$$C^{\text{EUR}}(T,K) := \mathbb{E}\left[e^{-\int_0^T r_u \, \mathrm{d}u}(S_T - K)_+\right] \tag{2}$$

- **Asian**

$$C^{\text{ASIAN}}(T,K) := \mathbb{E}\left[e^{-\int_0^T r_u \, \mathrm{d}u}\left(\frac{1}{T}\int_0^T S_u \, \mathrm{d}u - K\right)_+\right] \tag{3}$$

with maturity date $T > 0$ and strike (exercise price) $K > 0$

Numerical implementation : Use of Monte-Carlo experiment *MC*

- Motivated by the previously-developed equity path-generators

Implementation : European & Asian option pricing (2/2)

Results

- $N_{MC} \in [\![10^2, 10^4]\!] \Longrightarrow$ **Good performance**
- **Main constraint** : Desired samples number : $N_{MC} \in [\![10^5, 10^6]\!]$
- Serial MC $\Longrightarrow$ Bad overall performance

Solution : Speed-up MC computations

1. GPU acceleration — CUDA, OpenCL, ...
2. Multi-threading — I/O-bound tasks
3. Multi-processing — CPU-bound tasks

Results : Speed-up by ~ 3 times with 7 CPU logical cores [11] [12] involved
$\Longrightarrow N_{MC} \sim 10^6$ reachable

Additional features :

- Confidence intervals computations [13]
- OOP architecture
- Pre-computed simulations to improve overall performances

11. See quantitative study in Appendix
12. MacBook Pro i7 - 8 logical cores
13. Implemented for several Z-scores : 80, 85, 90, 95, 99, 99.5, 99.9

Parameters impact : Overview (1/2)

Context : Relevant and easy-to-use simulation framework

Studies outline

Equity	Interest rates	Studied ?
	Vasicek	**Studied**
Black-Scholes	CIR	**Studied**
	Hull-White	**Studied**
	Vasicek	**Studied**
Heston	CIR	**Not studied**
	Hull-White	**Not studied**

Table – Potential combinations for future studies

For each combination, $\longrightarrow$ **relevant selection of parameters to be studied**

Parameters : Divided into 2 families : Model parameters & Simulation parameters

Parameters impact : Overview (2/2)

Method

- Fixed randomness to clearly compare relevant trends
- Selection of 4 parameters **max.** per combination
- Study over simulation (equity + index) & derivatives pricing results

Models [14]

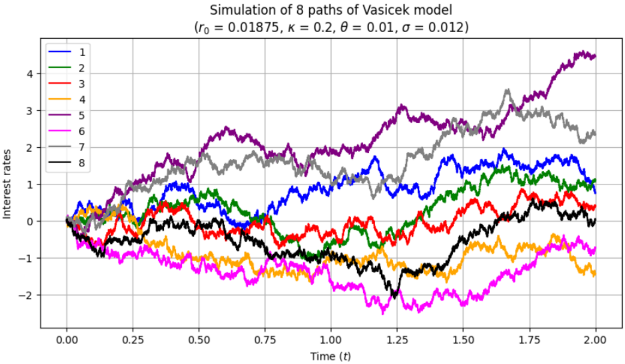
$$\begin{cases} \mathrm{d}r_t = \kappa(\theta - r_t)\mathrm{d}t + \eta \mathrm{d}B_t \\ r(0) = r_0 \quad \text{deterministic} \end{cases} \tag{4}$$

$$\begin{cases} \dfrac{\mathrm{d}S_t}{S_t} = r_t \mathrm{d}t + \sigma \mathrm{d}W_t \\ S_0 \geq 0 \end{cases} \tag{5}$$

Studied parameters We change the model's parameters : κ , θ , σ , T and N .

14. Generation of 2 correlated BMs needed : $(B_t)_t$ & $(W_t)_t$

Simulation of 8 paths of BS+Vasicek



Effect of κ on the asset price S_t and interest rate r_t

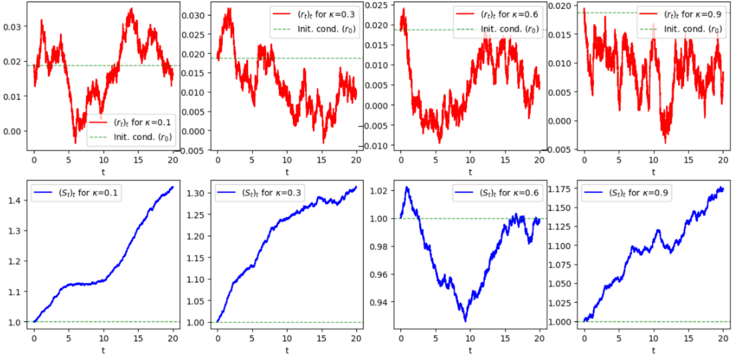


Figure – Effect of κ

Effect of θ on the asset price S_t and interest rate r_t

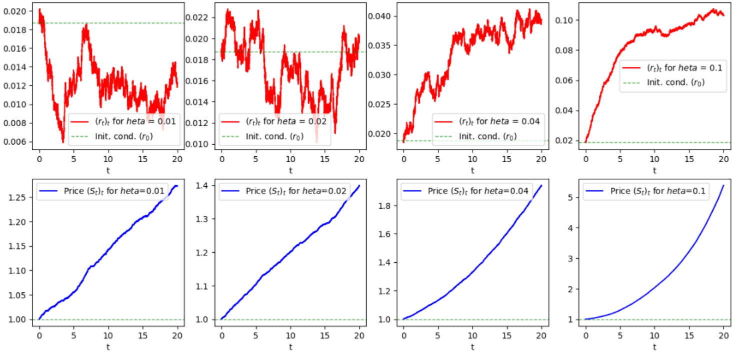


Figure – Effect of θ

Effect of the volatility σ on the asset price S_t and interest rate r_t

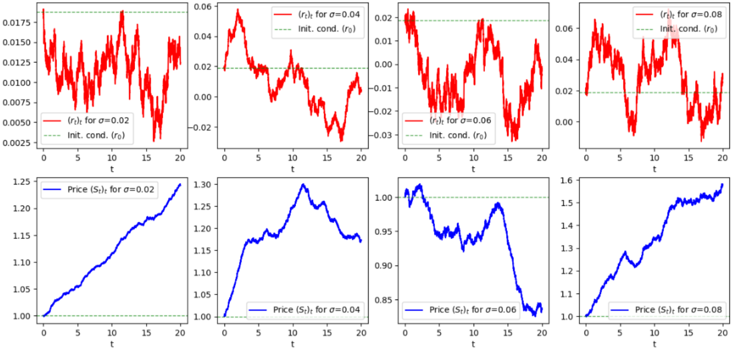


Figure – Effect of σ

Effect of number de steps N on the asset price S_t and interest rate r_t

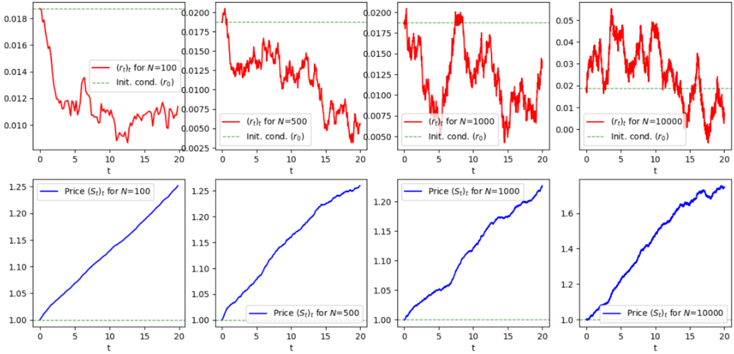


Figure – Effect of number of steps

Effect of the horizon T on the asset price S_t and interest rate r_t

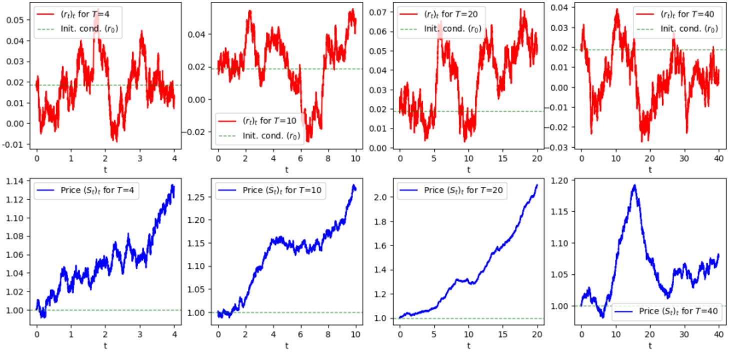


Figure – Effect of time

Joint Distribution

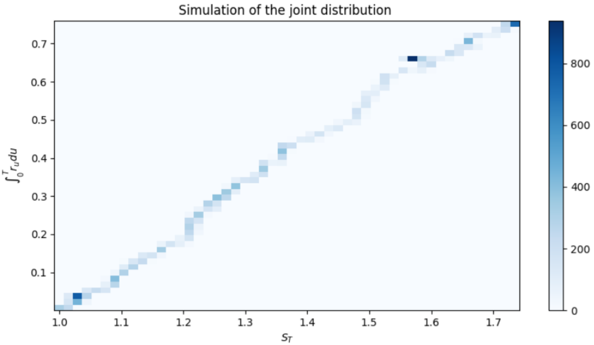


Figure – Simulation of the joint distribution

Analysis : Black-Scholes + CIR **Combination 2**

Models[15]

$$\begin{cases} \mathrm{d}r_t = \kappa(\theta - r_t)\mathrm{d}t + \sigma\sqrt{r_t}\mathrm{d}B_t \\ r(0) = r_0 \quad \text{deterministic} \end{cases} \tag{6}$$

$$\begin{cases} \dfrac{\mathrm{d}S_t}{S_t} = r_t\mathrm{d}t + \sigma\mathrm{d}W_t \\ S_0 \geq 0 \end{cases} \tag{7}$$

Studied parameters :

- Model : κ, θ, ρ
- Simulation : joint distribution

15. Generation of 2 correlated BMs needed : $(B_t)_t$ & $(W_t)_t$

Simulation of one path of the CIR model

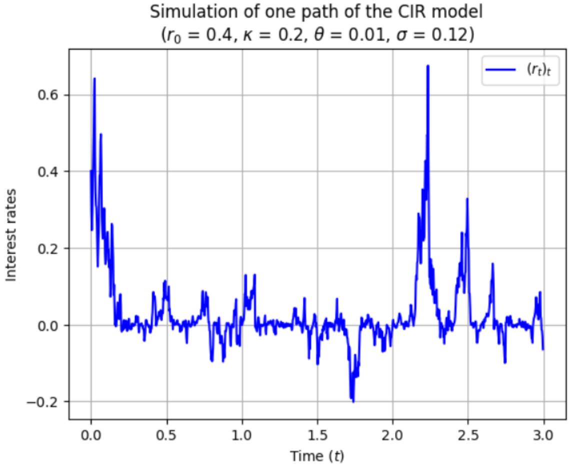


Figure – Simulation of one path of the CIR model

Simulation of 8 paths of the CIR model

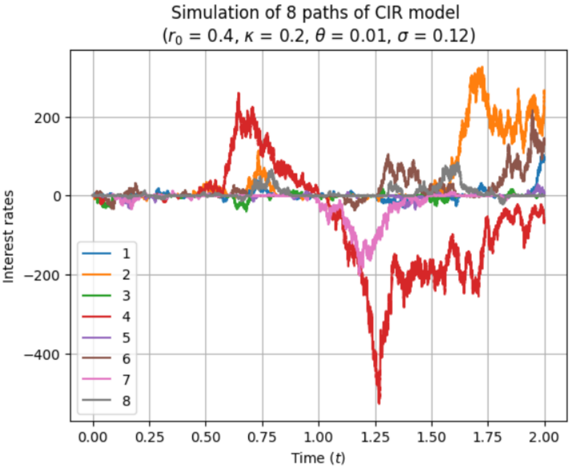


Figure – Simulation of 8 paths of the CIR model

Effect of κ on the asset price S_t and interest rates r_t

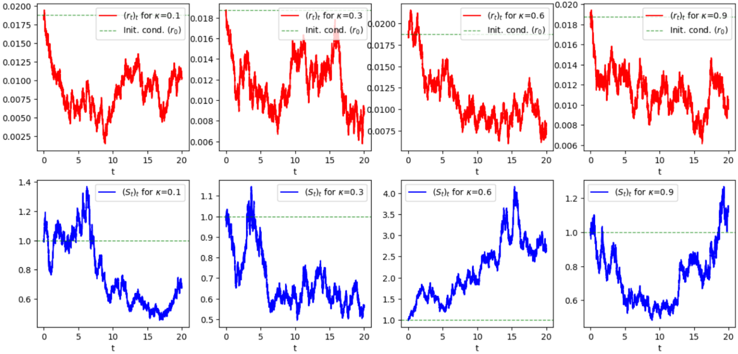


Figure – Effect of κ

Effect of θ on the asset price S_t and interest rates r_t

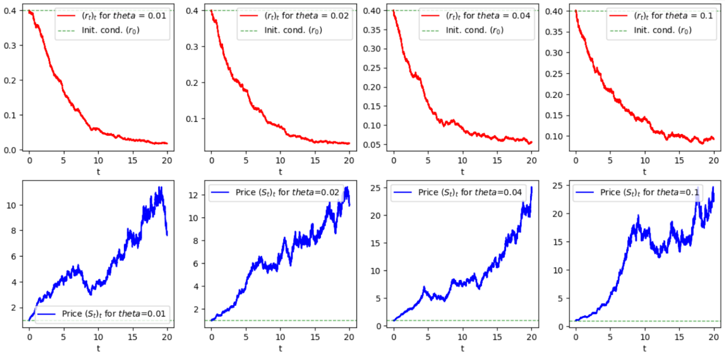


Figure – Effect of θ

Effect of ρ on the asset price S_t and interest rates r_t

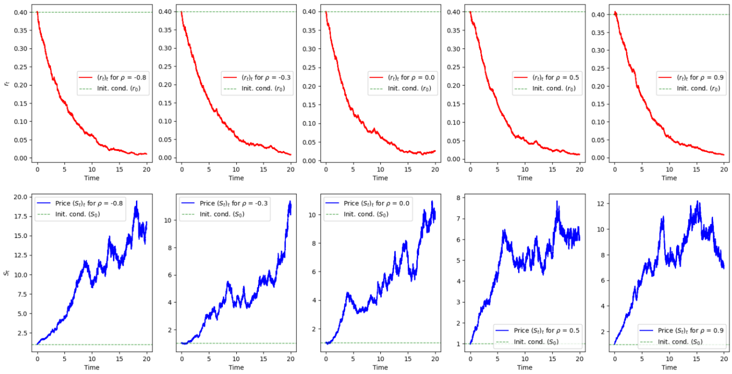


Figure – Effect of ρ

Simulation of the joint distribution

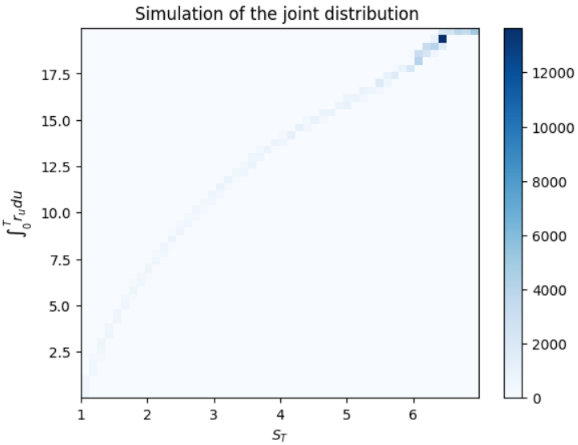


Figure – Simulation of the joint distribution

Analysis : Black-Scholes + Hull & White **Combination 3**

Models [16]

$$\begin{cases} \mathrm{d}r_t = (\theta(t) - ar_t)\mathrm{d}t + \sigma \mathrm{d}B_t \\ r(0) = r_0 \quad \text{deterministic} \end{cases} \tag{8}$$

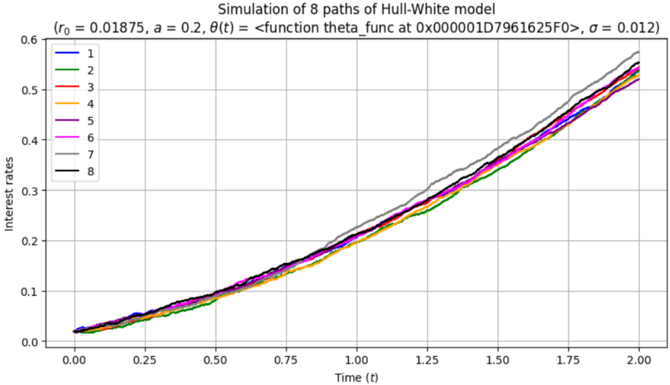
$$\begin{cases} \dfrac{\mathrm{d}S_t}{S_t} = r_t \mathrm{d}t + \sigma \mathrm{d}W_t \\ S_0 \geq 0 \end{cases} \tag{9}$$

Results : We chose different functions for $\theta(t)$, such as :

- $\theta(t)$ constant
- $\theta(t)$ exponential
- $\theta(t)$ logarithmic
- $\theta(t)$ sinusoidal

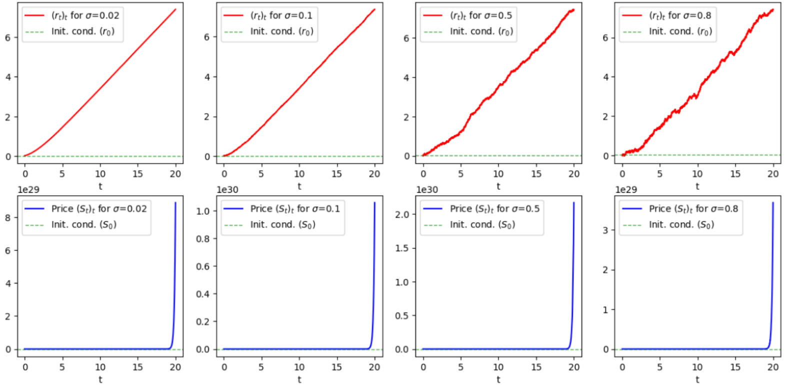
16. Generation of 2 correlated BMs needed : $(B_t)_t$ & $(W_t)_t$

Simulation of 8 paths of Hull-White



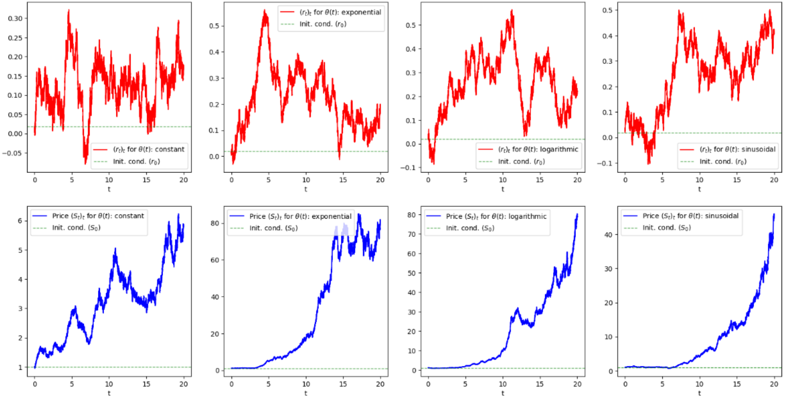
Effect of the volatility σ on the asset price S_t and interest rates r_t

:



Effect of the mean reversion $\theta(t)$

:



Analysis : Heston + Vasicek   Combination 4

Models [17]

$$\begin{cases} \mathrm{d}r_t = \kappa(\theta - r_t)\mathrm{d}t + \eta \mathrm{d}B_t \\ r(0) = r_0 \quad \text{deterministic} \end{cases} \tag{10}$$

$$\begin{cases} \dfrac{\mathrm{d}S_t}{S_t} = r_t \mathrm{d}t + \sqrt{V_t}\mathrm{d}W_t \\ S_0 \geq 0 \end{cases} \tag{11}$$

$$\begin{cases} \mathrm{d}V_t = \kappa(\theta - V_t)\mathrm{d}t + \eta\sqrt{V_t}\mathrm{d}\widetilde{W_t} \\ V_0 \geq 0 \end{cases} \tag{12}$$

Studied parameters

- Assumption : Fixing Heston parameters to respect Feller conditions $2\kappa\theta > \eta^2$
- Strategy of study : Compute the mean and the variance of the trajectories studied for every simulation

17. Generation of 3 correlated BMs needed : $(B_t)_t$, $(W_t)_t$ and $(\widetilde{W_t})_t$

Analysis : Heston + Vasicek **Combination 4**

Results

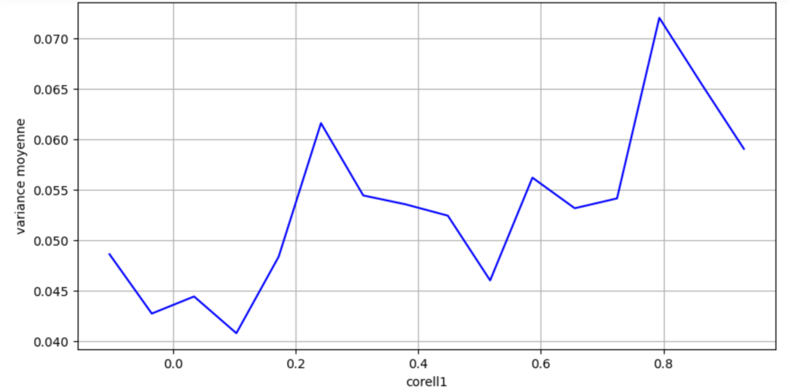


Figure – Effect of the correlation (with the variance)

Analysis : Heston + Vasicek **Combination 4**

Results

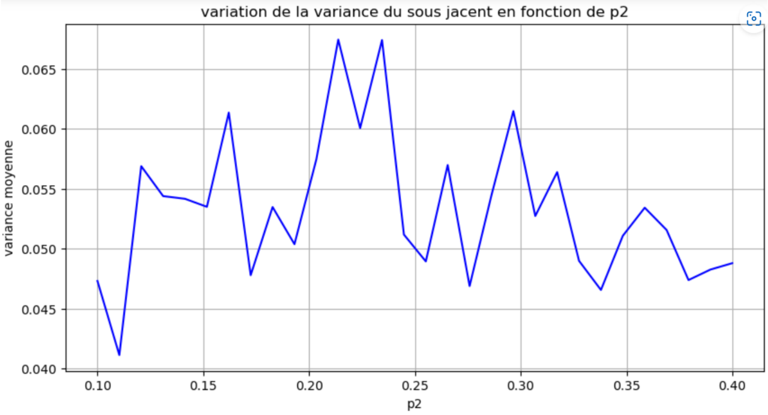


Figure – Effect of θ (with the variance)

Results

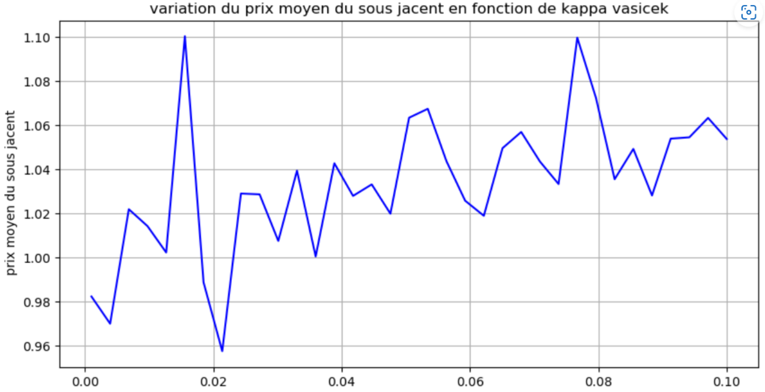


Figure – Effect of κ (with the underlying)

Conclusion & Perspectives

Criticism

- Use of independent parameters for the study , and could do some more technical work on the influence of the parameters
- Truncated study due to complexity and **high number of parameters pool**
- Complex organization with internships and other deadlines

Synthesis

- State-of-the-art of existing Python/R libraries for ESGs implementation
- Implementing rates, then equity models
- Implementation of efficient EURO & ASIA pricers
- In-depth analysis for the impact of model/simulation parameters

Conclusion & Perspectives

Extensions

- Introduce better parallelism/multithreading to speed up MC pricing
- Use of cloud computing instances with larger CPU cores *AWS Lambda, SageMaker*
- Exploit other ways to speed up the study of parameters impacts
- Building a REST API[18] to automate in a *user-friendly* UI the strategy runs
- Building a CLI[19] for strategies running automation

18. Flask, FastAPI
19. Click, argparse, ...

Appendices

Appendix : Euler & Milstein schemes (1/2)

Context (SDE)

$$\mathrm{d}X_t := a(t, X_t)\mathrm{d}t + b(t, X_t)\mathrm{d}W_t \tag{13}$$

with $X_0 = x_0 \in \mathbb{R}$ and time horizon in $[0, T]$ with $T > 0$.

Euler-Maruyama scheme

$$\forall n \in [\![0, N-1]\!],\ \ Y_{n+1} := Y_n + a(t_n, Y_n)\Delta t + b(t_n, Y_n)\Delta W_n \tag{14}$$

Milstein scheme

$$\begin{aligned}\forall n \in [\![0, N-1]\!],\ \ Y_{n+1} := Y_n &+ a(t_n, Y_n)\Delta t + b(t_n, Y_n)\Delta W_n \\ &+ \frac{1}{2} b(t_n, Y_n) b'(t_n, Y_n)\Big((\Delta W_n)^2 - \Delta t\Big)\end{aligned} \tag{15}$$

Framework

$$0 =: t_0 < t_1 < \ldots < t_N := T \text{ and } \Delta t := \frac{T}{N} \text{ and } t_k = k\Delta t$$

and $\Delta W_n := W_{t_{n+1}} - W_{t_n}$ independent and identically distributed normal random variables with zero mean and variance equals to Δt.

Appendix : Euler & Milstein schemes (2/2)

Euler-Maruyama scheme

- Strong error of order $\mathcal{O}(\sqrt{\Delta t})$
- Weak error of order $\mathcal{O}(\Delta t)$

Milstein scheme

- Strong error of order $\mathcal{O}(\Delta t)$
- Weak error of order $\mathcal{O}(\Delta t)$

Appendix : Multi-processing efficiency wrt CPU logical cores

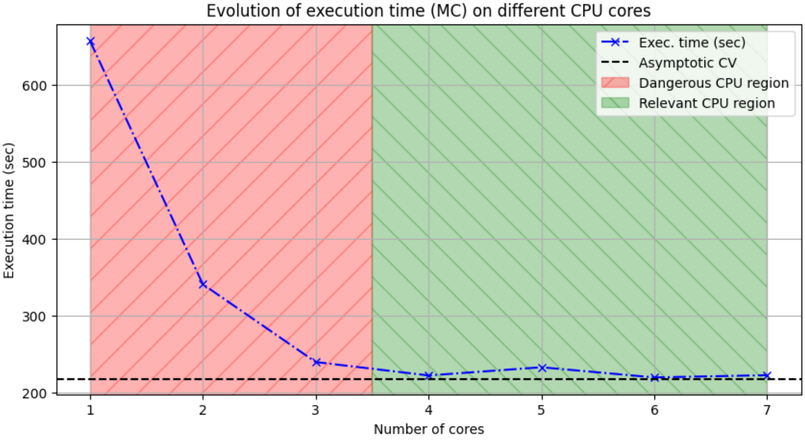


Figure – Monte-Carlo experiment - Confidence intervals

Appendix : Confidence intervals

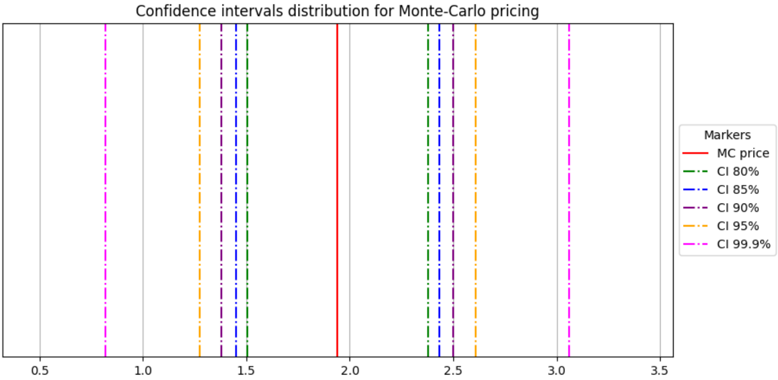


Figure – Monte-Carlo experiment - Confidence intervals

Appendix : MC convergence analysis

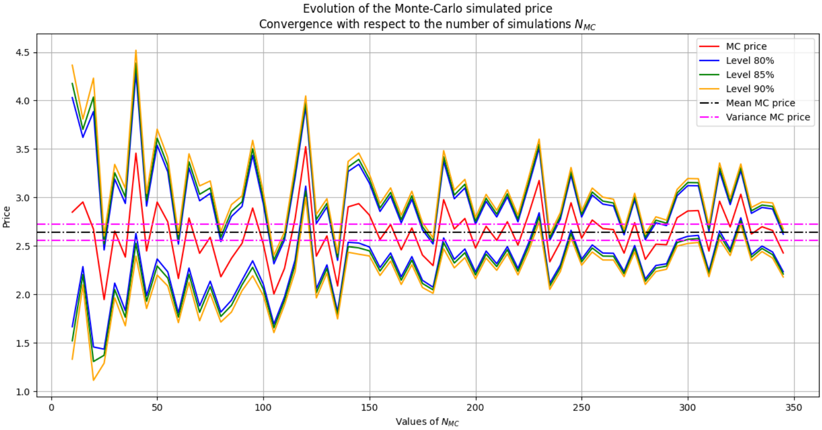


Figure – Monte-Carlo experiment - Convergence study for $N_{MC} \uparrow +\infty$