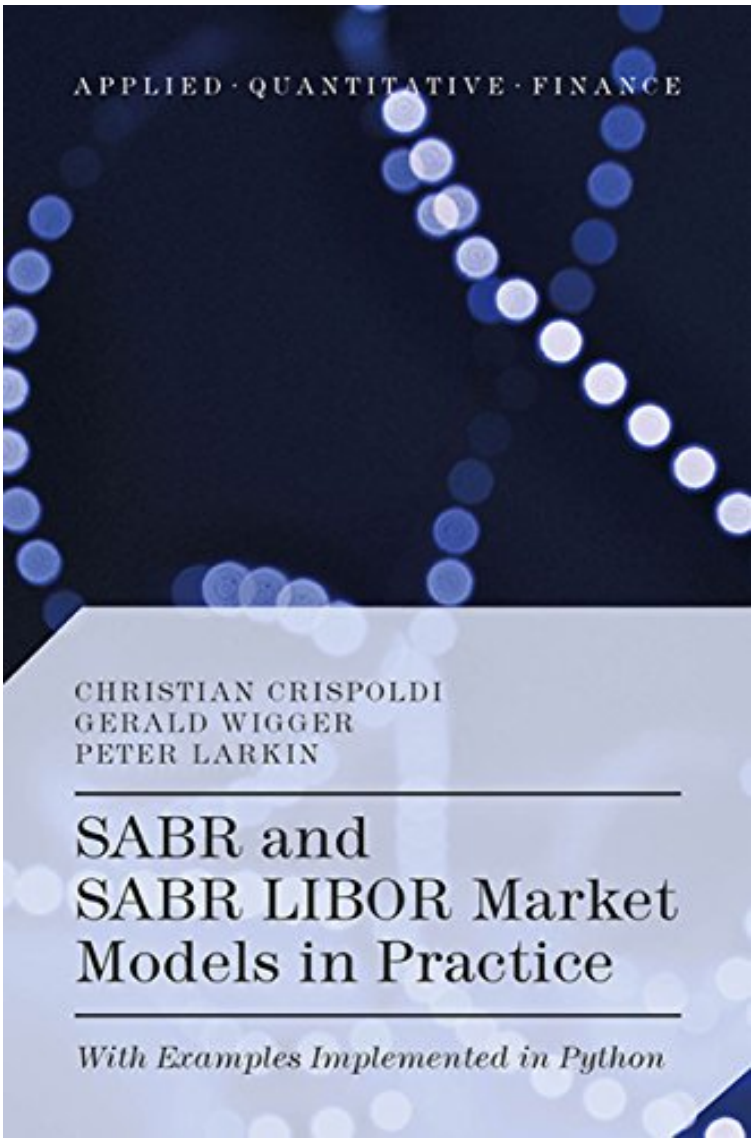
# SABR and SABR LIBOR Market Models in Practice: With Examples Implemented in Python (Applied Quantitative Finance)

*Christian Crispoldi, Geacute;rald Wigger, Peter Larkin*
**Download PDF | ePub | DOC | audiobook | ebooks*






#1539422 in eBooks 2016-04-29 2015-08-26File Name: B013J7RAVM | File size: 32.Mb

**Christian Crispoldi, Geacute;rald Wigger, Peter Larkin : SABR and SABR LIBOR Market Models in Practice: With Examples Implemented in Python (Applied Quantitative Finance)** before purchasing it in order to gage whether or not it would be worth my time, and all praised SABR and SABR LIBOR Market Models in Practice: With Examples Implemented in Python (Applied Quantitative Finance):

0 of 0 people found the following review helpful. Useful technical tips and detailsBy jsA hands on book where the

code and the technical tips can help speed up a lot the understanding of the model(s) treated. Both if you are still studying or just started on the desk.0 of 0 people found the following review helpful. Two StarsBy Stephan ChangThe code is not so clear.0 of 1 people found the following review helpful. Expensive for what it offers, superfluous codeBy CustomerRating 2.5 stars.I found this book's utility limited to the *only* useful piece of actual code - the simulation of Sabr LMM. If you really want to learn Sabr LMM, I strongly recommend The SABR/LIBOR Market Model: Pricing, Calibration and Hedging for Complex Interest-Rate Derivatives.A lot of python code seems superfluous. For somebody trying to learn Sabr LMM, do we really need code to -1. calculate first derivative (essentially (a-b)/h)2. calculate second derivative (essentially (a-2*b+c)/(h*h))3. draw 2 correlated random numbers (essentially X and rho*X + sqrt(1-rho*rho)*Y). Then there is a separate snippet to calculate more than 2 correlated random numbers (run of the mill cholesky decomposition, can effectively be done in two lines)!4. calculate (a+b*t)*exp(-c*t)+d (yes, this one liner is a separate python function in the book and mentioned as such in the book and all the marketing material).and so on, I hope you get the idea.The authors will say this code is used elsewhere in the book but then effectively the count of *examples implemented in python* (included in the book title as well) is down to two or three. Plus you will have to manually type the code from book, but hey at least no body can steal their code without paying for it! Another way to put it would be - no body can use their code even after paying for it!This being said, the book does have bits of useful information here and there. I suggest a buy recommendation if price goes down to 25 GBP/ 40 USD , its way too expensive otherwise. You are better off spending your money on Rebonato's book.

Interest rate traders have been using the SABR model to price vanilla products for more than a decade. However this model suffers however from a severe limitation: its inability to value exotic products. A term structure model agrave; la LIBOR Market Model (LMM) is often employed to value these more complex derivatives, however the LMM is unable to capture the volatility smile. A joint SABR LIBOR Market Model is the natural evolution towards a consistent pricing of vanilla and exotic products. Knowledge of these models is essential to all aspiring interest rate quants, traders and risk managers, as well an understanding of their failings and alternatives.SABR and SABR Libor Market Models in Practice is an accessible guide to modern interest rate modelling. Rather than covering an array of models which are seldom used in practice, it focuses on the SABR model, the market standard for vanilla products, the LIBOR Market Model, the most commonly used model for exotic products and the extended SABR LIBOR Market Model. The book takes a hands-on approach, demonstrating simply how to implement and work with these models in a market setting. It bridges the gap between the understanding of the models from a conceptual and mathematical perspective and the actual implementation by supplementing the interest rate theory with modelling specific, practical code examples written in Python. nbsp; nbsp;

About the AuthorChristian Crispoldi is a Vice President at Nomura Holding America Inc., in New York where he is responsible for the valuation and pricing of interest rate derivatives. Previously he worked as a financial engineer in various banks across Europe. Christian holds a Masters degree in Mathematical Finance from the University of York, UK, and a bachelor degree in Computer Engineering from the University of Bologna, Italy. Geacute;rald Wigger is Head of Quantitative Analysis at Weisshorn Re. He previously worked in various roles such as Head of Pricing at Axa Winterthur, Head of Risk Modeling at Zuuml;rcher Kantonalbank and Interest Rate Derivatives Quant at Bank of America Merril Lynch. Geacute;rald holds a PhD in Solid State Physics from ETH Zurich. Peter Larkin is a Data Scientist working on building predictive models using big data in the (re) insurance industry. Previously he worked as a Quantitative Analyst in the financial services industry working on projects spanning the pricing of structured products, credit and market risk, and asset management. Peter has a background in Theoretical Physics and received his PhD from the University of York in 2008, previously having obtained his Masters at Cambridge University and BSc at Imperial College London. In 2012 he also completed a MSc in Mathematical Finance from the University of Oxford.nbsp;