Global Calibration

Claudio Albanese [1]

August 18, 2009

Abstract

Current technology advances in computer engineering broaden substantially the realm of possibilities in the art of risk management of derivative portfolios. In this article, we discuss the benefits and technical feasibility of global calibration strategies. Although the industry is largely based on local calibration, we argue that global calibration is nowadays emerging as technically feasible and represents a useful complement to existing methodologies.

1 Calibration Strategies

Calibrating financial models is a challenging computational task, particularly difficult for exotic, long dated and hybrid derivatives. Local calibration methodologies are based on special models which are mathematical solvable in closed form for a handful of vanilla derivatives to be used as both hedging vehicles and calibration targets. Global calibration provides an antipodal alternative tackling the computational issues in engineering rather than in mathematics, thus avoiding to impose solvability restrictions on the underlying dynamics. Flexibility in the dynamic specification is obviously beneficial as it confers robustness and economic realism to the modeling exercise. A model with an unrealistic dynamic specification may calibrate to a handful of targets but will never be consistent across a broad spectrum of assets. By striving to achieve economic realism, global calibration is an exercise in information data mining more than an exercise in fitting.

In this section we give a bird view of local and global calibration methodologies, postponing to the following question a discussion of more technical issues and a case study. Both are methodologies for the valuation and risk management of derivative portfolios and implement with different strategies similar steps, such as:

- Assigning a pricing model to each derivative position.
- Identifying hedging vehicles;
- Identifying a strategy to calibrate pricing models consistently with hedging instruments

- Carrying out VaR analysis on the portfolio by shocking inputs (i.e. prices of hedging instruments);
- Evaluating hedge ratios.

The local calibration approach is by far the one of most widespread use to implement the tasks just mentioned. It involves the following steps:

- Identify a small set out of a list of models named after authors who have discovered closed form mathematical solutions for vanilla derivatives, such as the Black-Scholes model for equity and FX derivatives, the Hull-White model, the Heath-Jarow-Morton model and the Brace-Gatarek-Musiela model for interest rate derivatives, the Constant Elasticity of Variance model by Cox and Ross, the Heston model, Dupire's Local Volatility model, etc.. To each named model, one associates a specific class of derivative instruments;
- Attribute to each derivative position a handful of vanilla derivatives which at least in principle could be used as hedges for it;
- Calibrate position-specific pricing models consistently with the chosen hedging vehicles on an instrument-by-instrument basis. The expression "local calibration" derives precisely from the policy to tailor a different set of model parameters to each instrument;
- Apply adjustors for valuation purposes based on empirical rules in such a way to compensate for the systematic biases that simple models usually exhibit;
- Map a portfolio of exotic derivatives to a portfolio of vanilla options providing the theoretical individual hedges, using as hedge ratios the sensitivities obtained by the individually calibrated derivatives;
- Apply an econometrically more realistic model such as Hagan's SABR to assess the risk profile of the mapped portfolio of vanilla options and determine a strategy for global portfolio hedging;
- Use historical shocks on the mapped portoflio to finally arrive to a VaR figure valid for capital allocation purposes.

If local calibration sounds like black magic, it's because it is more an art than a science. Adjustors are often determined via a complex set of empirical rules, see for instance the patented implementation of the vanna-volga methodology to FX options by the firm Superderivatives (US Patent 7315838). The empirical rules work in most cases and are reinforced by their status of broadly adopted market standard. However, empirical rules are validated through back-testing and are not guaranteed to work in case markets dislocate in yet unseen ways.

One concern is that there is no theoretical foundation in support of the use of different model specifications across different instruments sharing the same underlying. On the contrary, the Fundamental Theorem of Finance precisely stipulates that the prices obtained by using consistently a unique model are guaranteed to be arbitrage free. Remarkably, the theorem admits also a converse statement: given a set of arbitrage free prices, there exists one unique model which reproduces them all. If such a model was not there, then the prices would allow for arbitrage.

Global models consistent with all available information are thus the theoretical bedrock on which Finance lies. They were also elevated to the status of holy grail because of the technical difficulties in deriving them. The search for global models will be frustrated if one does not have the technical means of capturing in the same model specification all the realistic features of the underlying process that ultimately find an expression in derivative prices. This is certainly the case if one imposes a constraint of mathematical tractability in terms of closed form solutions: reality always finds a way to break away from such a cage. But if one has the technology means to consider and analyze econometrically realistic models, then the search for globally consistent models has a realistic hope to succeed. Not only, when such a search fails because one reaches consistency with a large basket with the exception of a few outliers, this finding may well be an indication that the outliers are actually miss-priced. Theoretically, global models should be the cement that would stabilize and keep derivative markets together, as long as they were broadly known by a sufficient number of competing market participants.

Notwithstanding these solid theoretical reasons, the industry has based itself on the antithetic methodology of local calibration mostly for practical reasons. In particular:

- Implementation difficulties due to the limits of heritage computer systems;
- Mathematical difficulties in seizing the opportunity of innovations in computer engineering.

Current technology implementations for local calibration systems are based on CPU cluster farm technology. Namely:

- Local calibration tasks on an instrument-by-instrument basis are simplified by selecting very few hedging vehicles for calibration and analytically solvable models for which fitting parameters can be determined within one or two seconds at most on a CPU node;
- Cluster computing implementations are based on middleware and load balancers that seamlessly orchestrate the revaluation of each individual instrument once its pricing information is detected as outdated;
- Pricing and risk management tasks are carried out by executables spawned on individual cluster nodes, which process data from a shared drive and write the output back on the same when finished, typically without using interprocess communication or shared memory;

- Financial firms typically do not require quants to write thread-safe code and actually demand that thread-safety not be relied upon.
- The use of analytic solvability and implicit PDE schemes makes double precision a mandatory requirement, as these algorithms would be unstable in single precision.

A trading system based on global calibration would be engineered based on different principles. Namely, one would do the following:

- Allocate a special team and dedicated computing resources to the task of calibrating each individual risk factor globally across all available information. This is a sort of data mining exercise to capture in a single, realistic model description all market information and translate it into a mathematical format that allows one to generate realistic future scenarios which are consistent with all the input information;
- Apply the globally calibrated pricing models to each instrument. If a derivative depends on more than one risk factor, one would still take single factor marginal distributions calibrated globally and correlate them using dynamic copulas to produce correlated scenarios. Correlation coefficients can also be estimated globally to achieve the broadest possible consistency across the known universe of derivative pricing information, also including when appropriate historical information;
- Identify a set of hedging instruments to be used for a specific portfolio for aggregate hedging;
- Price all exotic single factor derivatives with the same globally calibrated model;
- Price all hybrid derivatives by correlating the globally calibrated single factors models by means of dynamic copulas;
- Carry out VaR analysis on the portfolio by shocking inputs (i.e. prices of hedging instruments);
- Evaluate hedge ratios in aggregate on a portfolio basis and against all designated hedging instruments at once.

Technology implementations for global calibration systems would have to leverage on the emerging computing technologies and be based on multi-GPU workstations endowed with the recent processors.

- GPUs (Graphic Processing Units) such as nVidia Tesla 1060 are ideal matrix engines. They are marketed at very low cost since they are produced in great voluments for the graphics and game market. Each GPU device is able to achieve a sustained performance of 350 GF/sec carrying out the linear algebra required to price derivatives by backward induction, the operation needed for global calibration. A workstation with 4 GPUs showcases an impressive sustained performance of 1.4TF/sec. This equipment is thus capable of executing high quality global optimization algorithms that ordinarily require 1-5 petaflops per risk factor. (One petaflop amounts to 1,000,000,000,000 floating point operations);

- Recent multi-core processors of the Intel Nehamel (iCore7) class are ideal chipsets for scenario generation in virtue of the large dedicated third level cache of 2 MB per core where one can store hash keys and look up tables. A kit with two Nehamel class processors achieves a performance of over 700 milion single period evaluations per second on an interest rate simulation. A similar performance can also be achieved by 3 Teslas 1060. Considering that most simulations stop at 20,000 scenarios, one sees that this order of performance is in excess of about four orders of magnitude with respect to today's standards. Traditional methods based on analytically closed form solvable models are often memory bound, not being able to keep full the instruction pipelines feeding current processors;

- Competing chipsets with similar functionality by ATI-AMD and IBM also leveraging on volume markets promise a healthy race toward steadily increasing performance;

- Multi-GPU equipment and multi-core CPUs require shared-memory engineering and asynchronous multi-threading programming, with several threading models and memory layers coexisting in the same equipment. Managing this kind of hardware platform requires a radical break away from the established coding practices which typically either place no requirement upon thread-safety or stick to simple patterns such as enforcing that all functions be re-entrant.

Using globally calibrated models as opposed to insisting on analytic solvability has several business benefits to financial organizations:

- One can divide roles between engineers dedicated to the maintanance of computational engines and economists or market experts devoted to the design of realistic models, the two being separated by a Chinese wall provided by a programmatic interface;

- Pricing libraries can be designed on a much reduced code base as the base computational engines can be shared across all asset classes, as opposed to having a separate code base for each of the named models;

- The calibration and modeling tasks can be separated away from the pricing and risk management tasks, possibly even delegated to third party providers;

- Profit attribution and mark-to-market can be separated from trade execution functions in an organization whereby models are centrally maintained;

- A global calibration system offers an economically meaningful representation and at least an alternative viewpoint vis-a-vis the market dominant practice of local calibration methodologies and can be useful to detect both outliers and systematic pricing biases;

- Staggering performance up to four orders of magnitude better then the current industry standards can be achieved by leveraging on various factors such as the ability to describe processes by means of matrices which fit as look-up tables in cache and the ability to share the same model across all instruments in large derivative portfolios.

2 Mathematical Framework

(This section is rather technical. The uninterested reader can safely skip this section and continue to the next one which dwells over a case study.)

The needed mathematics and numerical analysis change substantially when one builds models around matrix manipulation technology. However, beneath the formal and technical differences, the basic postulate of arbitrage freedom still plays the role of the pivotal linchpin of Financial Mathematics.

The traditional mathematical framework of Finance is given by stochastic calculus. This branch of Mathematics is largely aimed at obtaining closed form solutions whenever possible and otherwise bypassing the need to evaluate transition probabilities by means of a number of techniques of infinitesimal calculus. To optimally lever upon the emerging computing infrastructure it is useful to depart from this tradition and shift to a new mathematical framework built upon matrix manipualations, also called operator methods. The intent of this framework is to make use of matrix-multiplication engines to numerically evaluate transition probability kernels expressed as large matrices and then obtaining all quantities of financial interest out of composing and numerically differentiating these matrices. In this section we give a brief description of the key ideas.

To model the evolution of a risk factor, it is convenient to discretize it by identifying a finite number of state variables $x, y = 0, ..d - 1$. For instance, in the case study reviewed in the next section we choose $d = 512$ possible values for a state variable x and then associate to each value of x a particular value for a short rate and a state of monetary policy.

At any given time, one assumes that the risk factor corresponds to a given state variable $x = 0, ..d - 1$. To describe a process, one assigns a matrix $u_{\delta t}(x, y; t)$ giving the transition probabilities for the state x to evolve into the state y at any given date t and over a fixed but small time interval δt equal to one day or a fraction thereof such as 12 or 6 hours. This matrix is called elementary transition probability kernel. To be valid, an elementary transition probability kernel must satisfy the following two properties:

- $u_{\delta t}(x, y; t) \geq 0$,
- $\sum_y u_{\delta t}(x, y; t) = 1 \quad \forall x$.

The first property states that transition probabilities are positive, the second that they add up to 1.

To find the kernel over time horizons longer than δt, one can make use of matrix multiplications. In fact, the law of compounded probabilities indicates that the transition probability kernel over the double length interval $2\delta t$ is given by

$$u_{2\delta t}(x,y) = \sum_z u_{\delta t}(x,z) u_{\delta t}(z,y). \tag{1}$$

Notice that this rule amounts to the rule of matrix multiplication. This is the link that establish contact with the graphic hardware engineering as also 3-d visualization hinges on the ability to multiply matrices effectively. Iterating this equation, one then arrives to kernels over time steps of length $4\delta t$, $8\delta t$, etc..., i.e. we iterate as follows:

$$u_{2\delta t} = u_{\delta t}^2, \quad u_{4\delta t} = u_{2\delta t}^2, \quad ..... \quad u_{2^n\delta t} = u_{2^{n-1}\delta t}^2. \tag{2}$$

The algorithm we just described is called *fast exponentiation* and it is certainly not new. It was known to ancient Greek mathematician as an effective way of computing exponentials. However, fast exponentiation has not found many industry applications yet except for cryptographic algorithms which involve small matrices. The reason for its scarce popularity is that until this decade there was no available equipment specifically designed for the purpose of multiplying matrices and offering staggering performance at executing this task. Now that the situation has changed on that front, fast-exponentiation is likely to become a prominent tool in numerical analysis. In Finance, this technique will arguably give the modeler direct control over transition probability kernels. But this is a recent development and we live in a transition period: at the moment, the current practice is to use induction methods devised to bypassing the need of computing kernels.

With fast exponentiation, the length of the elementary time interval δt can easily be chosen very small. In fact, halving this interval has only the marginal impact of adding one more step in the iteration. It turns out that there is a critical threshold given by the so called Courant condition such that whenever the elementary time interval is below that threshold, the resulting long step kernels are very smooth. Smoothness is of great practical importance because operator methods are all about manipulating kernel matrices and if these are affected by a high level of noise errors would propagate fast. Remarkably, this is not the case here.

A way to understand what is at work is by reference to the butterfly effect: a small perturbation that inflates exponentially to produce a hurricane of vast proportions. This is what happens when the elementary time step δt is chosen above the Courant threshold, thus yielding a marginally unstable algorithm and no action is taken to iron out irregularities in the kernels with one method or another. The phenomenon can be further amplified whenever one uses single precision as opposed to double precision, as the noisier single precision arithmetics causes greater disturbances. This is in fact what happens in traditional backward induction algorithms used in practice whereby one proceeds step by step, moving of δt days at a time and one is forced to choose time steps δt exceeding the Courant bound.

On the other hand, whenever δt is below the Courant threshold there is no butterfly effect. What happens is that the roundoff errors one inevitably incurs at every step largely compensate against each other, the positive errors offsetting the negative errors, thus cleaning off the signal to a surprising degree and without outside intervention.

If the Courant condition is respected, then one can evaluate efficiently probability kernels even using single precision arithmetics and the resulting kernels are smooth. Single precision is a major technical issue as GPU chipsets have grown out of the graphics market and are thus based primarily on 32-bit floating point arithmetic engines. The most recent releases of GPUs also carry out double precision arithmetics, but at a tremendous performance cost of about a factor 8 which greatly reduces their competitive advantage with respect to high end CPUs.

Another consideration of engineering importance concerns the algorithms to generate scenarios. A transition matrix of size 512-512 in single precision takes about 1 MB of memory. Current processors have large Level-3 caches: the Nehamel offers an abundant 2 MB per core. Thus the transition probability kernels fit well within local caches and can be efficiently used for scenario generation. GPUs offer a very complex memory architecture with global, shared, constant, texture and registry memory, but they do not have traditional caches usable for the purpose of storing look up tables for scenario generation. Because of this reason, they are at a competitive disadvantage in this task. The added benefit of generating scenarios CPU side is that payoff valuation functions are currently implemented only CPU side and are very ill suited to the SIMD structure of GPUs because of the extensive conditional constructs they involve which typically gives rise to extensive branching.

The optimal arrangement is to have GPUs process kernels by fast exponentiation and CPUs run Monte Carlo scenario generation and formula valuations based on those kernels. As of 2009, Supermicro has been releasing motherboards with precisely this design, combining nVidia Teslas and Intel Nehamel processors, thus creating the ideal computational platforms for the new methodology. Going forward, one can only imagine that this hardware design will establish itself as the dominant mainstream architecture.

The quantum leap in performance for calibration tasks is impressive, as CPU based workstations could process at best 20 GF/sec on matrix-intensive tasks, while the new multi-GPU kits can sustain performances in excess of 1.4 TF/sec. The vast improvement in performance makes global calibration possible and has thus the potential of having a major business impact on business practices.

Heightened performance in fact translates into quality. When using operator methods, models don't need to be analytically solvable. They need to be expressible as Markov chains on lattices of size in the range 500-1000 on current hardware, and this already yields tremendous latitude to the modeler. The size restriction enables one to reproduce quite faithfully single factor models with a finely discretized factor. One can also accommodate regime switching models, i.e. collections of models describing separate market regimes whereby transitions between model specifications which ordinarily are achieved extrinsically by recalibration can instead be achieved endogenously. Regime switching features are essential for the task of global calibration as they allow one to separate market effects of different time scales and create a more robust calibration framework whereby fewer updates are needed. This in turn gives rise to better risk management and more effective hedging strategies.

To elaborate on the possibilities, I give here two examples of regime switching models I implemented for interest rates and for FX derivatives. The short rate process has the following form:

$$r_t = \lambda(t)\rho_t \tag{3}$$

where $\lambda(t)$ is a deterministic function of time, assumed positive to avoid negative rates and calculated in such a way to fit the term structure of rates precisely. A stylized description of the process ρ_t in stochastic calculus notations is given as follows:

$$d\rho_t = \mu_{a_t}dt + \kappa(t)(\theta(t) - \rho_t)dt + \sigma_{a_t}\rho^{\beta_{a_t}(t)}dW + \text{small jumps}, \tag{4}$$

$$da_t = k(t)(\bar{a} - a_t)dt + s(t)dW + \text{small jumps}. \tag{5}$$

The regime variable a_t denotes monetary policy and has the effect of shifting the mean reversion level for rates, thus affecting the steepness of the curve. Jumps are added to ensure that whenever there is a change in monetary policy there is also a sizeable change in the short rate.

The short rate process depends on parameters which are a function of the monetary regime variable and of time. Parameters are assumed to be constant over periods of 3 months but are allowed to change otherwise. As we explain below, there are a total of 26 free parameters and the calibration routine is tasked with estimating them all. The regime variable a can take 8 values corresponding to as many rate regimes. Regime switching confers volatility to rate spreads by inducing steepening or flattening of the yield curve.

FX models with deterministic rates are very convenient for calibration purposes. They can be defined in terms of the exchange rate X_t subject to a regime switching dynamics of the form

$$dX_t = \mu(t)dt + \sigma_{a_t}X^{\beta_{a_t}(t)}dW + \text{jumps}, \tag{6}$$

$$da_t = k(t)(\bar{a} - a_t)dt + s(t)dW + \text{small jumps}. \tag{7}$$

The drift term $\mu(t)$ is adjusted in such a way to achieve risk-neutrality, i.e.

$$E_t\left[\frac{dX_t}{X_t}\right] = (f^d(t) - f^f(t))dt. \tag{8}$$

where $f^d(t)$ and $f^f(t)$ are the domestic and foreign overnight forward rates. The jump terms are added in such a way that a higher level of volatility is accompanied by a jump in the underlying.

The FX model above shows both stochastic volatility and stochastic reversal dynamics. To model the FX process when interest rates are stochastic, one can still use the model above but reinterpret the process X_t, i.e. by interpreting as FX rate the process

$$\bar{X}_t = e^{\int_0^t (r_t^d - r_t^f - f_t^d + f_t^f)dt} X_t. \tag{9}$$

This introduces factors in payoffs that need to be accounted for. Since we have both kernels and discounted kernels, this calculation can be reduced to matrix manipulation.

From a pricing viewpoint, one can use new variations on the traditional strategies of backward induction and simulation, with the added benefit of being able to evaluate long step transition probability kernels. Backward induction is particularly efficient for callables and European options. Monte Carlo simulations instead are ideally suited for Target Redemption Notes and similar forward looking path dependent options. Finally, the ability to differentiate transition probability kernels is a very powerful tool combined with moment methods for derivative written on daily averages like volatility derivatives.

Efficient implementations would see portfolios being priced in aggregate (as opposed to pricing individual securities individually), as this strategy would better exploit concurrency on current hardware. This type of orchestration will likely necessitate a reorganization of middle-ware environment which nowadays are largely based on individual derivative pricing, not aggregate valuation.

Transition probability kernels give a way to execute long step Monte Carlo algorithms. To calculate price sensitivities, likelihood ratio methods combined with long-step Monte Carlo is very effective in this context.

Correlation can be modeled by means of dynamic Gaussian copulas while preserving marginal distributions. There is no difficulty to use fully calibrated lattice models for each risk factor, as for instance two interest rates and one foreign exchange rate.

In summary, running Monte Carlo simulations involves evaluating kernels, discounted kernels and discount factors by means of GPU coprocessors. This is the single most demanding task and can require 1-2 Tera Flops. Scenario generation can take place both on the CPU or device side on the GPU. However, current CPUs have an edge at Monte Carlo scenario generations. Two recent Intel Xeon processors are roughly equivalent to three nVidia Tesla 1060. Teslas however have superior performance at kernel calculations, showcasing sustained performance rates of 360 GF/sec as opposed to 20-30 GF/sec obtainable on an iCore7 in single precision.

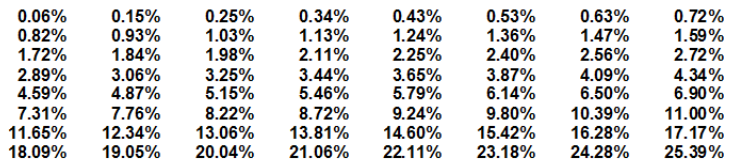
3 An example of Global Calibration

Calibration involves designing models which reproduce all the econometric features of the underlying process as they transpire from historical time series. Next, one needs to optimize parameters.

Our case study consists of an interest rate model in the Japanese yen on August 31, 2008. We take a calibration basket consisting of around 220 European swaptions, 70 flow CMS spread options and 25 callable CMS spread options. One evaluation of a calibration basket requires 1-2 tera-flops for interest rate derivatives, a bit less for FX and equity derivatives.

On multi-GPU hardware, one evaluation takes around 3 seconds. A multi-threaded optimization algorithm arrives to an optimum within about 2 hours on a single 4-GPU unit. To achieve this task, we developed an optimization algorithm for calibration which is

- Suitable to objective functions which are not differentiable due to the large number of input datapoints;
- Amenable to a multi-threaded implementation;



0.06%	0.15%	0.25%	0.34%	0.43%	0.53%	0.63%	0.72%
0.82%	0.93%	1.03%	1.13%	1.24%	1.36%	1.47%	1.59%
1.72%	1.84%	1.98%	2.11%	2.25%	2.40%	2.56%	2.72%
2.89%	3.06%	3.25%	3.44%	3.65%	3.87%	4.09%	4.34%
4.59%	4.87%	5.15%	5.46%	5.79%	6.14%	6.50%	6.90%
7.31%	7.76%	8.22%	8.72%	9.24%	9.80%	10.39%	11.00%
11.65%	12.34%	13.06%	13.81%	14.60%	15.42%	16.28%	17.17%
18.09%	19.05%	20.04%	21.06%	22.11%	23.18%	24.28%	25.39%

- Tuned for situations where a function evaluation is time-expensive.

The particular model I implemented is a short rate model with regime switching of the form in the previous section. The short rate process depends on parameters which are a function of the monetary regime variable and of time. Parameters are assumed to be constant over periods of 3 months but are allowed to change otherwise. As we explain below, there are a total of 26 free parameters and the calibration routine is tasked with estimating them all. The regime variable a can take 8 values corresponding to as many rate regimes. Regime switching confers volatility to rate spreads by inducing steepening or flattening of the yield curve.

rate regime	drift	grid dilation factor
1	-1.25%	40 %
2	-1 %	60 %
3	-0.75%	80%
4	-0.50%	100%
5	-0.25%	120 %
6	0.00 %	140 %
7	0.25 %	160 %

For each regime, one finds a short rate grid containing 64 nodes. The number 64 is chosen to optimize GPU matrix handling. The base grid is given in the following table: This grid corresponds to a grid dilation factor of 100%. The other dilation factors apply to different regimes.

The function $\lambda(t)$ is constrained to be positive, not to allow for negative rates. Also, the calibration algorithm favors solutions where $\lambda(t)$ is a deterministic function of time. If the calibration basket contained only swaptions and no CMS spread options, the function $\lambda(t)$ would be flatter and close to one. In this case, the regime dynamics would be responsible for reproducing a steep yield curve. However, correlations between forward swap rates would be too low to correctly price CMS spread options.

The model parameters are collectively summarized by the tables in Fig. 1 and Fig. 2 giving the initial and optimal values, respectively. This tables describe term structure functions which are either constant or of the form

$$\xi_i(t) = A_i + B_i \exp(-t/\tau_i). \tag{10}$$

where t is time. There are 12 such curves in the model. Some curves are assumed constant and parameterized in terms of the constant A only. Other ones are parameterized in terms of the asymptotic at infinity A, the initial value $A + B$ and the characteristic decay time. This results in a grand total of 26 scalar parameters.

In addition to depending on time, parameters may also depend on the regime. The tables assigns values for the lowest regime and the highest, while values for the other regimes are obtained by linear interpolation.

The model parameters are used by the modeler to form a Markov generator defining the dynamics. The modeler can do so by using a simple high level interface which is invoked at start time and used to fill one buffer on each device for each period covered by the model. To go as far as 50 years in the future on a quarterly basis, one needs 200 buffers. For a model of lattice dimension 512 this takes about 600MB on each device, abundantly below the 4GB memory limit.

Calibration and pricing proceeds from the stipulated user-model interface with model independent algorithms. One of the main properties of the regime switching dynamics for short rates we have selected is that it gives rise to scenarios for the yield curve which look realistic and require small adjustments to reproduce exactly the actual curves. See Figs. 3, 4 and 5. Realism in the underlying scenarios

- Confers robustness when pricing exotics;
- Produces meaningful hedge ratios;
- Allows for a single calibration to be used against a great variety of exotic payoffs.

Econometric realism for the shape of yield curves is not easily achieved unless the model reflects the actual short rate process. See for instance Fig. 6 for typical yield curve scenarios that can be produced by a simple PCA analysis. This shows that fitting two moments is not a guarantee of realism. Current interest rate pricing models often generate unrealistic curves by emphasizing only correlation matching. There is to add that having a realistic evolution for curves is not a strictly necessary condition for valuation to be correct as long as the payoffs is only sensitive to the first and second moments. However, hedge ratios and very exotic payoffs can be problematic. As a rule, when only two moments are sufficient as for Bermuda swaptions and (unlevered) callable CMS spread options, we find a very tight agreement in the valuation provided by our stochastic monetary policy model and moment methods based on PCA analysis as the BGM model.

In the example discussed here below I calibrate with respect to JPY interest rate derivatives including

- At-the-money European swaptions of maturities: 6 months, 1y, 2y, 3y, 4y, 5y, 6y, 7y, 8y, 9y, 10y, 20y and of tenors: 6 months, 1y, 2y, 3y, 4y, 5y, 6y, 7y, 8y, 9y, 10y, 15y, 20y.
- Out-of-the-money European swaptions of maturity-tenor pairs: 5y-2y, 5y-5y 5y-10y, 5y-20y, 5y-30y, 10y-2y, 10y-5y, 10y-10y, 10y-20y, 10y-30y, 20y-2y, 20y-5y, 20y-10y, 20y-20y and strikes as in Fig. 12.

- Callable CMS spread options in the following table:

payoff	maturity	strike
20y - 5y	10y	1.2%
20y - 2y	10y	2.4 %
20y - 2· 2y	10y	1.5%
20y - 2y	15y	1.8%
20y - 5y	15y	3 %
20y - 2· 2y	15y	2 %
20y - 2· 2y	5y	3 %
20y - 2· 2y	20y	1.5 %
20y - 2· 6m	15y	1 %
20y - 2· 6m	10y	2.5 %
20y - 3· 6m	10y	3 %
20y - 3· 6m	15y	0.5 %
20y - 2· 2y	20y	0.5 %
20y - 6m	10y	0.5 %
20y - 6m	15y	0.75 %

Callable CMS spread options

The quality of fit is shown in Fig. 7 for at-the-money swaptions, Fig. 8 for out-of-the-money swaptions, Fig. 9 and Fig. 10 for flow CMS spread options, Fig. 15 for callable CMS spread options. Figures 16 and 17 show two examples of pricing functions for callable CMS spread options which represent very useful information for the structurer. Finally, Fig. 11 shows the term structure of a swap-spread volatility and Fig. 12 a term structure for expected future correlation. Fig. 13 is also interesting as it shows the correlation structure as a function of the short rate and the monetary policy regime. This reflects the real world experience of varying degree of correlation as a function of the steepness and general shape of the curve.

The stochastic monetary policy model compares surprisingly well with BGM, as is shown in the figure below. The average discrepancy observed on a large real portfolio of callable CMS spreads is 14bp of nominal. The bias due to the handling of the call features in BGM using the LS variance reduction method is 8bp and justifies most of the discrepancy. The peak is also particularly pronounced and related to the simulation noise as in this sample only 1000 scenarios were used for the BGM calculation. The systematic discrepancy with respect to a non-parametric, 50 factor BGM model solved using the lower bounds in the Longstaff-Schwarz algorithm is +1bp. The estimated gap between lower and upper bounds for BGM is security dependent and in the range 3-20bp. There is no smile structure. See Fig. 18.

The discrepancy with a 2-factor Hull-White model with local calibration and adjustors is much greater at around 60 bp and there is a noticeable smile effect. See Fig. 19.

This examples shows how global calibration can be useful for risk control analysis. Simpler models such as Hull-White's are often used in front-desk applications because of their speed. A 50 factor BGM model on the other extreme would mostly be used for risk control as a benchmark because of its high quality of fit. The conclusion from out analysis of this case is that a globally calibrated model can play the same role as a benchmark.

Regarding performance: The valuation of an individual calibration basket valuation 2.9 seconds but the operation can be parallelized and scales almost perfectly on multi-GPU equipment. The global calibration run takes about two hours. Pricing a single CMS deal takes 800 milliseconds.

To speed up portfolio pricing and achieve economy of scale with also thanks to global calibration, one needs to synchronize cash flow dates. This operation has only marginal impact on total portfolio value and reduces the compute time down to 28 seconds.

Since the callable CMS spread options are priced by backward induction, as a result of a pricing run one obtains not only the spot value of the portfolio, but also the value at all other possible initial conditions. This allows one to perform a Value-at-risk analysis at nearly no computational cost by using the model probability themselves. Figure 20 shows the result of this analysis. Interestingly, the graph shows that the risk of loss corresponding to the 99% percentile of an un-hedged portfolio is linked to an easing change in monetary policy regime. This is precisely what happened after this dataset was taken, in the aftermath of the September 2008 financial crisis.

4 Conclusion

We find that global calibration is implementable in the difficult case of interest rate derivatives also thanks to recent progress in computer engineering. Past experience in the credit-equity domain makes us conclude that global calibration promises to be a concept of broad applicability across all asset classes.

The organization of labor around the concept of global calibration would see a team dedicated to calibrating models globally across asset classes or, in alternative, global calibration datasets being obtained from a specialized third party provider. The pricing function for exotic derivatives would be divorced from calibration and focus on correct payoff implementation. More than one calibrated model can be produced for any given risk factor, giving a way to the end user to assess model risk.

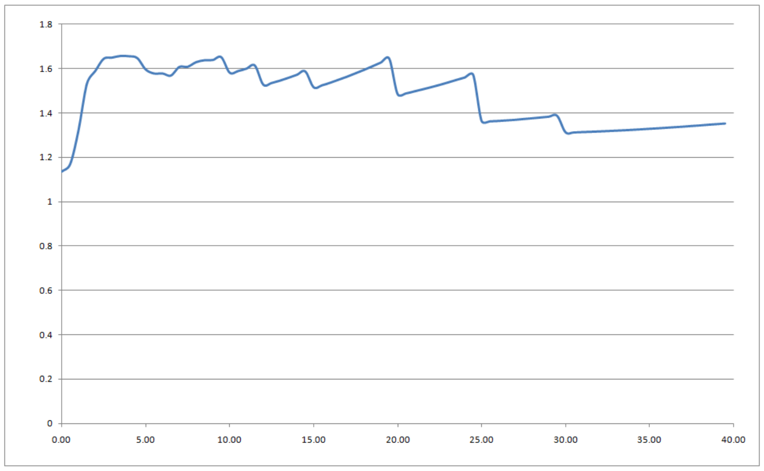
From the implementation standpoint, global calibration involves a team of engineers dedicated to system development, maintenance and optimization and a team of economists devoted to model building. The two would interact through a programming API but the engineers would not impose constraints on the modeling features, except for placing limits to how large the number of state variables can be.

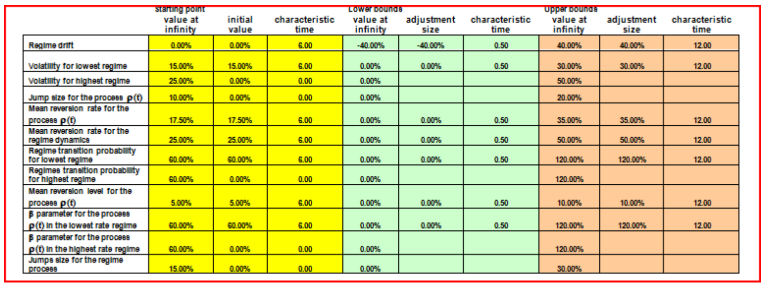
The implementation necessitates multi-GPU hardware, especially for the crucial calibration step. Pricing benefits greatly of GPU coprocessors for kernel calculations, while CPUs are found to be best at scenario generation.

Case studies show that once the constraint of analytic solvability is lifted, a parsimoniously chosen 2 factor lattice model with 512 state variables achieves the same modeling quality of a 50 factor BGM model, which balances the constraint of analytic solvability with the large number of factors. The more parsimonious model choice leads to computing performances that are about 4 order of magnitude better than the more standard alternative.

Global calibration opens the way to new business tools such as real time aggregate valuation of exotic portfolios and real time value-at-risk calculation.

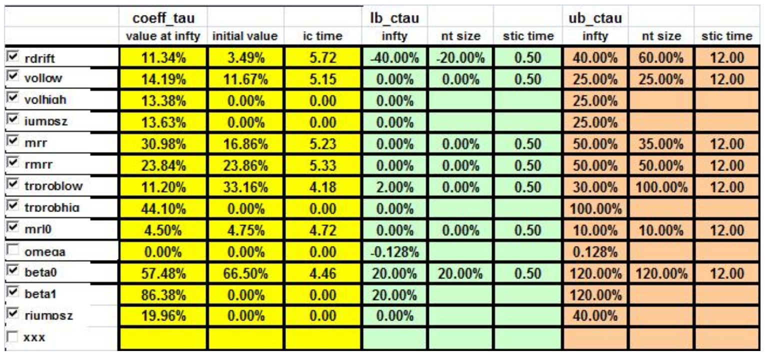
Although global calibration would certainly not supplant local calibration, we conclude that it represents a useful complement to existing methodologies and could be particularly useful for risk control function to detect outliers and systematic pricing biases.





	Starting point			Lower bounds			Upper bounds		
	value at infinity	initial value	characteristic time	value at infinity	adjustment size	characteristic time	value at infinity	adjustment size	characteristic time
Regime drift	0.00%	0.00%	6.00	-40.00%	-40.00%	0.50	40.00%	40.00%	12.00
Volatility for lowest regime	15.00%	15.00%	6.00	0.00%	0.00%	0.50	30.00%	30.00%	12.00
Volatility for highest regime	25.00%	0.00%	0.00	0.00%			50.00%		
Jump size for the process $\rho(t)$	10.00%	0.00%	0.00	0.00%			20.00%		
Mean reversion rate for the process $\rho(t)$	17.50%	17.50%	6.00	0.00%	0.00%	0.50	35.00%	35.00%	12.00
Mean reversion rate for the regime dynamics	25.00%	25.00%	6.00	0.00%	0.00%	0.50	50.00%	50.00%	12.00
Regime transition probability for lowest regime	60.00%	60.00%	6.00	0.00%	0.00%	0.50	120.00%	120.00%	12.00
Regimes transition probability for highest regime	60.00%	0.00%	0.00	0.00%			120.00%		
Mean reversion level for the process $\rho(t)$	5.00%	5.00%	6.00	0.00%	0.00%	0.50	10.00%	10.00%	12.00
β parameter for the process $\rho(t)$ in the lowest rate regime	60.00%	60.00%	6.00	0.00%	0.00%	0.50	120.00%	120.00%	12.00
β parameter for the process $\rho(t)$ in the highest rate regime	60.00%	0.00%	0.00	0.00%			120.00%		
Jumps size for the regime process	15.00%	0.00%	0.00	0.00%			30.00%		

Figure 1: Initial Point in the optimization



	coeff_tau			lb_ctau			ub_ctau		
	value at infty	initial value	ic time	infty	nt size	stic time	infty	nt size	stic time
✓ rdrift	11.34%	3.49%	5.72	-40.00%	-20.00%	0.50	40.00%	60.00%	12.00
✓ vollow	14.19%	11.67%	5.15	0.00%	0.00%	0.50	25.00%	25.00%	12.00
✓ volhiah	13.38%	0.00%	0.00	0.00%			25.00%		
✓ iumpsz	13.63%	0.00%	0.00	0.00%			25.00%		
✓ mrr	30.98%	16.86%	5.23	0.00%	0.00%	0.50	50.00%	35.00%	12.00
✓ rmrr	23.84%	23.86%	5.33	0.00%	0.00%	0.50	50.00%	50.00%	12.00
✓ trproblow	11.20%	33.16%	4.18	2.00%	0.00%	0.50	30.00%	100.00%	12.00
✓ trprobhia	44.10%	0.00%	0.00	0.00%			100.00%		
✓ mrl0	4.50%	4.75%	4.72	0.00%	0.00%	0.50	10.00%	10.00%	12.00
☐ omeaa	0.00%	0.00%	0.00	-0.128%			0.128%		
✓ beta0	57.48%	66.50%	4.46	20.00%	20.00%	0.50	120.00%	120.00%	12.00
✓ beta1	86.38%	0.00%	0.00	20.00%			120.00%		
✓ riumpsz	19.96%	0.00%	0.00	0.00%			40.00%		
☐ xxx									

Figure 2: Optimal parameters

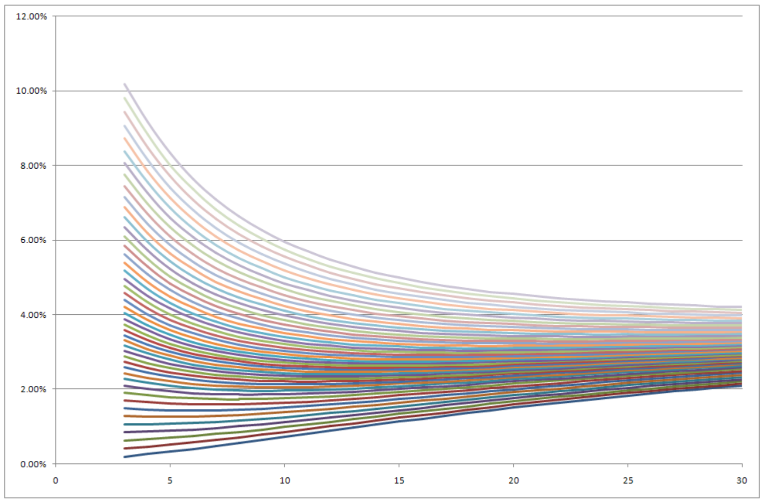


Figure 3: Yield curves in regime 0 with $\lambda(t) \equiv 1$

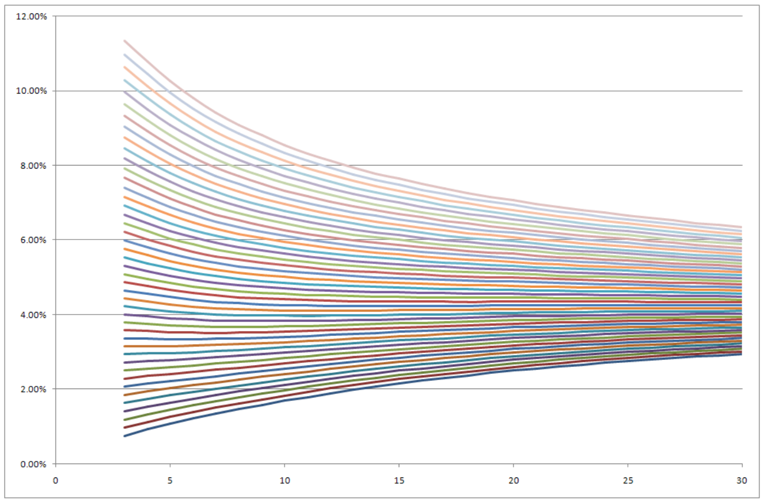


Figure 4: Yield curves in regime 3 with $\lambda(t) \equiv 1$

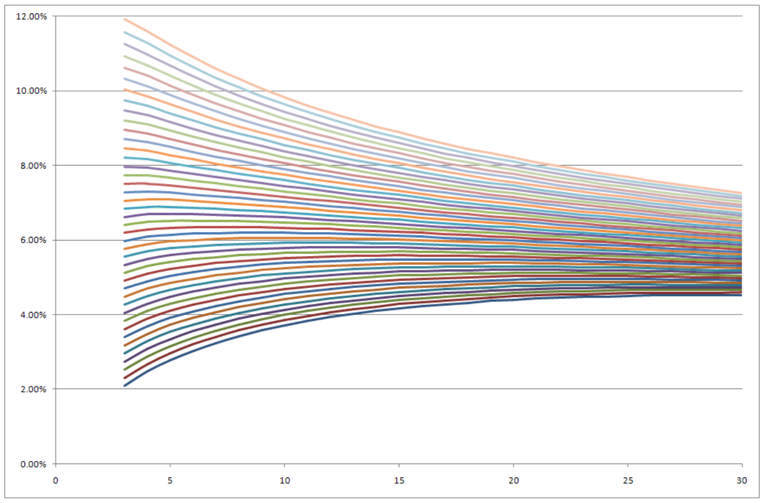


Figure 5: Yield curves in regime 7 with $\lambda(t) \equiv 1$

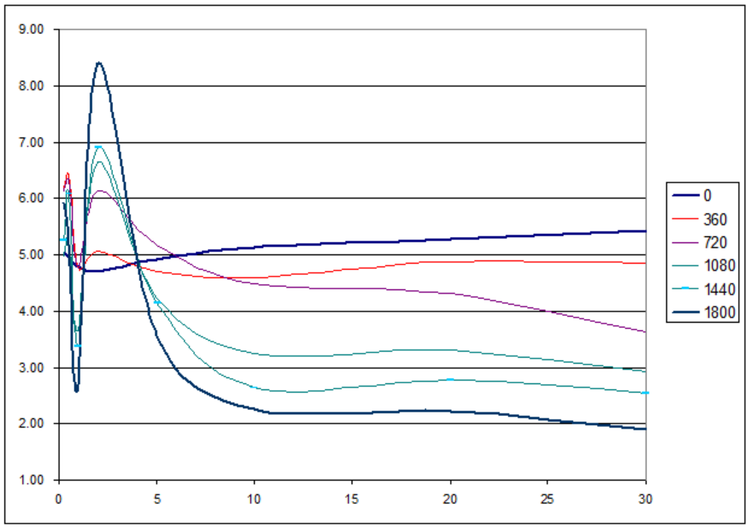
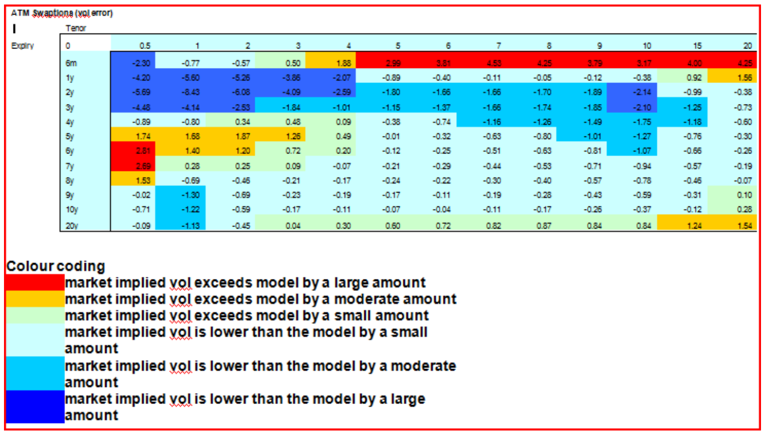


Figure 6: Arbitrage free yield curve scenarios with correct PCA moments



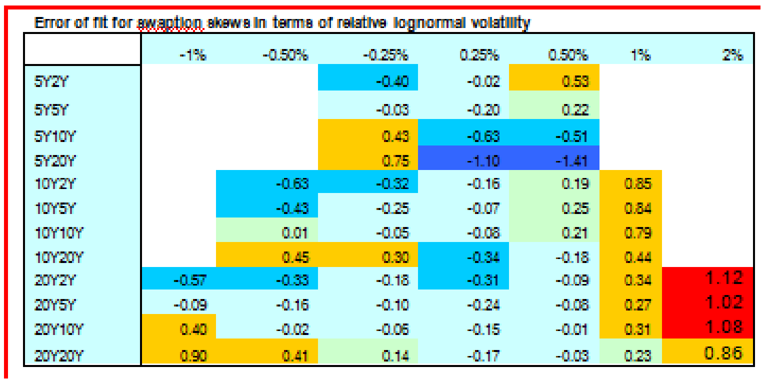
ATM Swaptions (vol error)

Expiry	Tenor 0	0.5	1	2	3	4	5	6	7	8	9	10	15	20
	6m	-2.30	-0.77	-0.57	0.50	1.88	2.99	3.81	4.53	4.25	3.79	3.17	4.00	4.25
	1y	-4.20	-5.60	-5.26	-3.86	-2.07	-0.89	-0.40	-0.11	-0.05	-0.12	-0.38	0.92	1.56
	2y	-5.69	-8.43	-6.08	-4.09	-2.59	-1.80	-1.66	-1.66	-1.70	-1.89	-2.14	-0.99	-0.38
	3y	-4.48	-4.14	-2.53	-1.84	-1.01	-1.15	-1.37	-1.66	-1.74	-1.85	-2.10	-1.25	-0.73
	4y	-0.89	-0.80	0.34	0.48	0.09	-0.38	-0.74	-1.16	-1.26	-1.49	-1.75	-1.18	-0.60
	5y	1.74	1.68	1.87	1.26	0.49	-0.01	-0.32	-0.63	-0.80	-1.01	-1.27	-0.76	-0.30
	6y	2.81	1.40	1.20	0.72	0.20	-0.12	-0.25	-0.51	-0.63	-0.81	-1.07	-0.66	-0.26
	7y	2.69	0.28	0.25	0.09	-0.07	-0.21	-0.29	-0.44	-0.53	-0.71	-0.94	-0.57	-0.19
	8y	1.53	-0.69	-0.46	-0.21	-0.17	-0.24	-0.22	-0.30	-0.40	-0.57	-0.78	-0.46	-0.07
	9y	-0.02	-1.30	-0.69	-0.23	-0.19	-0.17	-0.11	-0.19	-0.28	-0.43	-0.59	-0.31	0.10
	10y	-0.71	-1.22	-0.59	-0.17	-0.11	-0.07	-0.04	-0.11	-0.17	-0.26	-0.37	-0.12	0.28
	20y	-0.09	-1.13	-0.45	0.04	0.30	0.60	0.72	0.82	0.87	0.84	0.84	1.24	1.54

Colour coding

🟥	market implied vol exceeds model by a large amount
🟧	market implied vol exceeds model by a moderate amount
🟩	market implied vol exceeds model by a small amount
⬜	market implied vol is lower than the model by a small amount
🟦	market implied vol is lower than the model by a moderate amount
⬛	market implied vol is lower than the model by a large amount

Figure 7: ATM swaptions fit errors in log-normal vol



Error of fit for swaption skews in terms of relative lognormal volatility

	-1%	-0.50%	-0.25%	0.25%	0.50%	1%	2%
5Y2Y			-0.40	-0.02	0.53		
5Y5Y			-0.03	-0.20	0.22		
5Y10Y			0.43	-0.63	-0.51		
5Y20Y			0.75	-1.10	-1.41		
10Y2Y		-0.63	-0.32	-0.16	0.19	0.85	
10Y5Y		-0.43	-0.25	-0.07	0.25	0.84	
10Y10Y		0.01	-0.05	-0.08	0.21	0.79	
10Y20Y		0.45	0.30	-0.34	-0.18	0.44	
20Y2Y	-0.57	-0.33	-0.18	-0.31	-0.09	0.34	1.12
20Y5Y	-0.09	-0.16	-0.10	-0.24	-0.08	0.27	1.02
20Y10Y	0.40	-0.02	-0.06	-0.15	-0.01	0.31	1.08
20Y20Y	0.90	0.41	0.14	-0.17	-0.03	0.23	0.86

Figure 8: Out of the money swaptions fit errors in log-normal vol

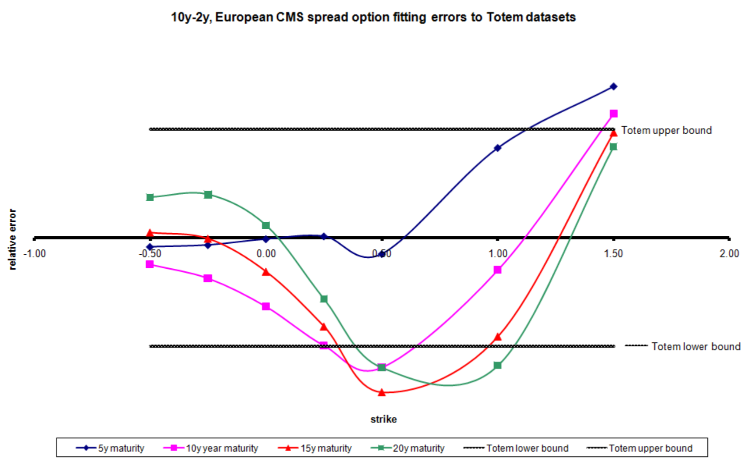


Figure 9: Fit with flow CMS spread options

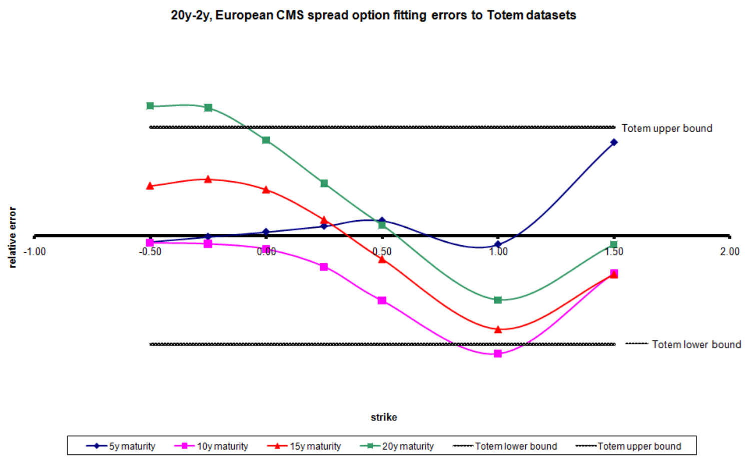


Figure 10: Fit with flow CMS spread options

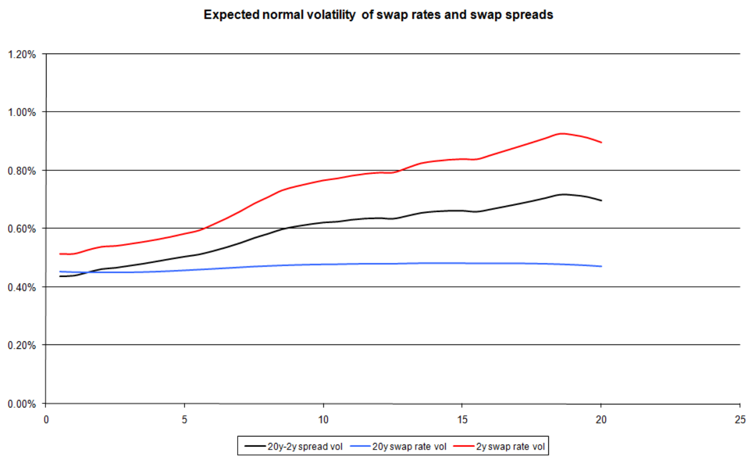


Figure 11: Expected future spread volatility

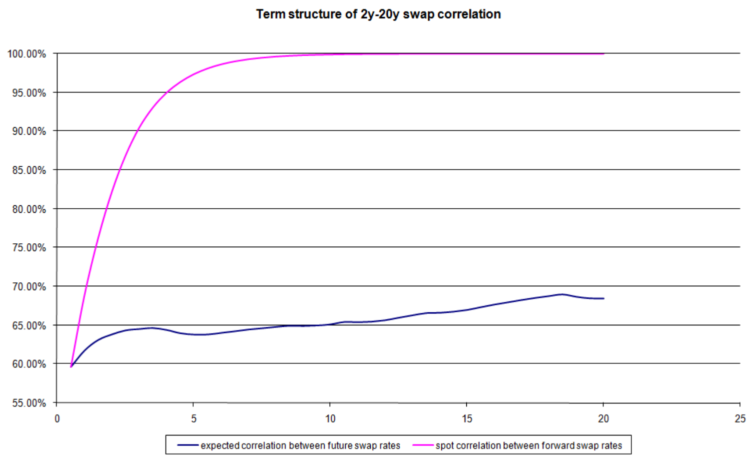


Figure 12: Expected future correlation

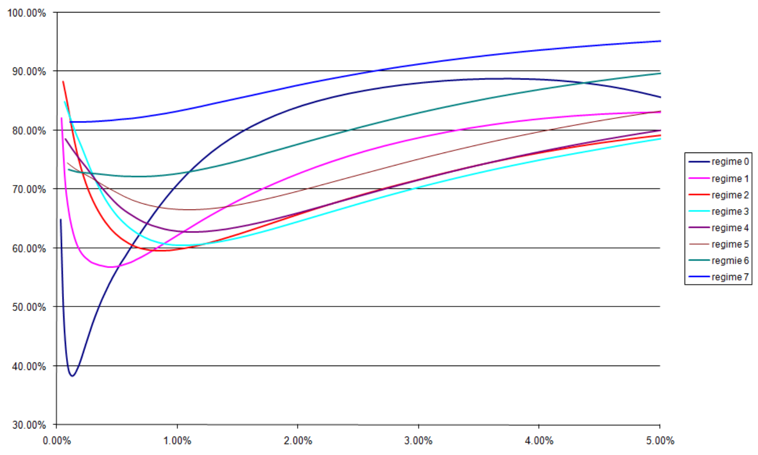


Figure 13: Correlation by regime

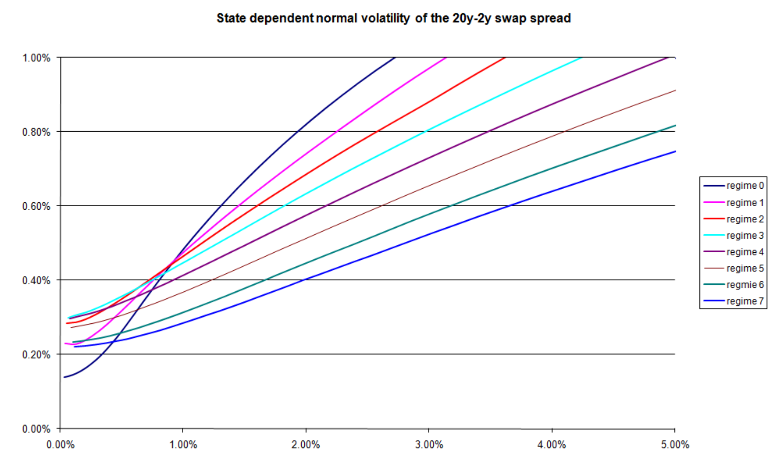


Figure 14: Spread volatility by regime

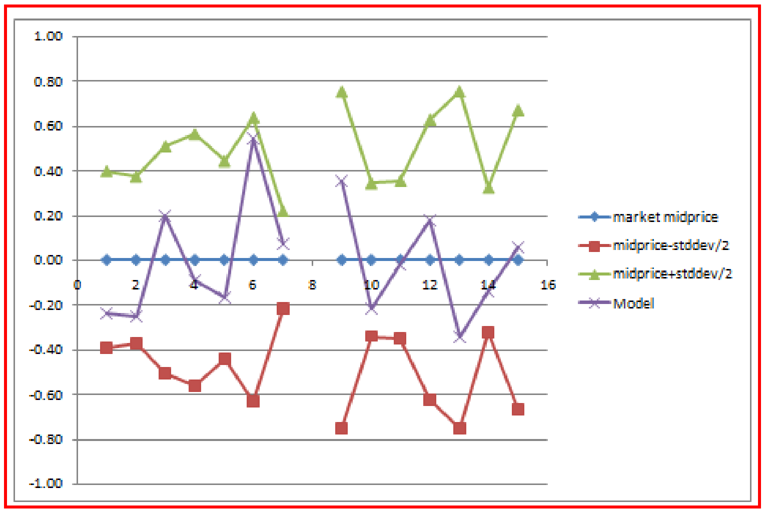


Figure 15: Callable CMS Spread Options Fit Errors in Log-normal Vol

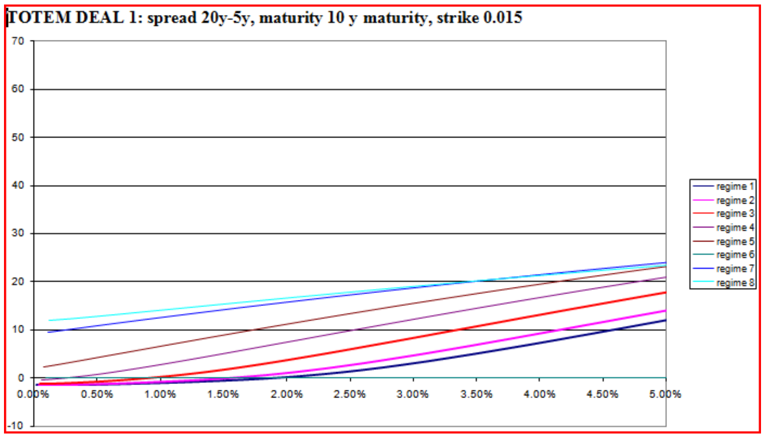


Figure 16: Callable CMS Spread Options, Totem deal 1

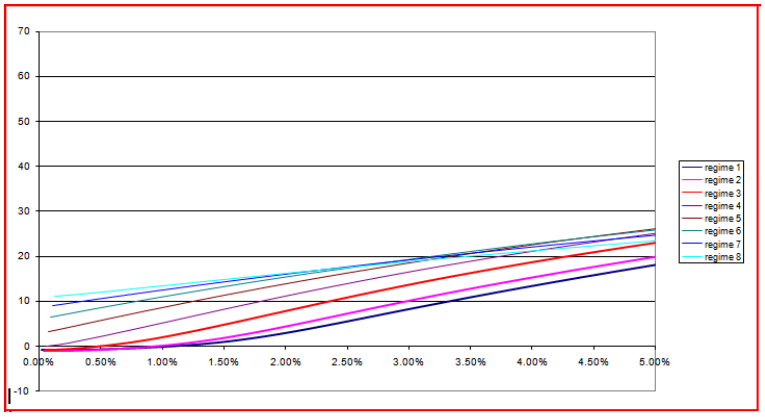


Figure 17: Callable CMS Spread Options, Totem deal 14

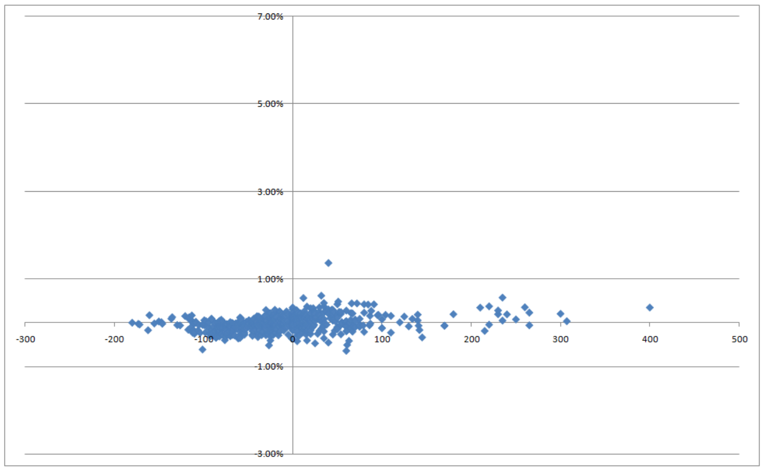


Figure 18: Discrepancies with BGM on a large portfolio

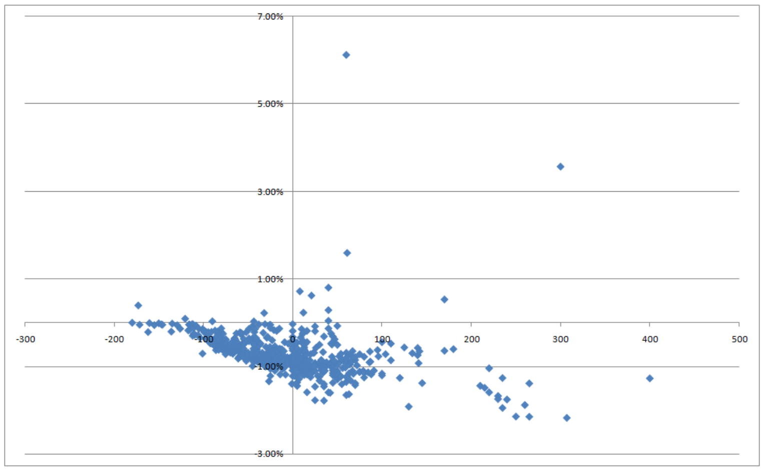


Figure 19: Discrepancies with multi-factor Hull-White with adjustors on a large portfolio

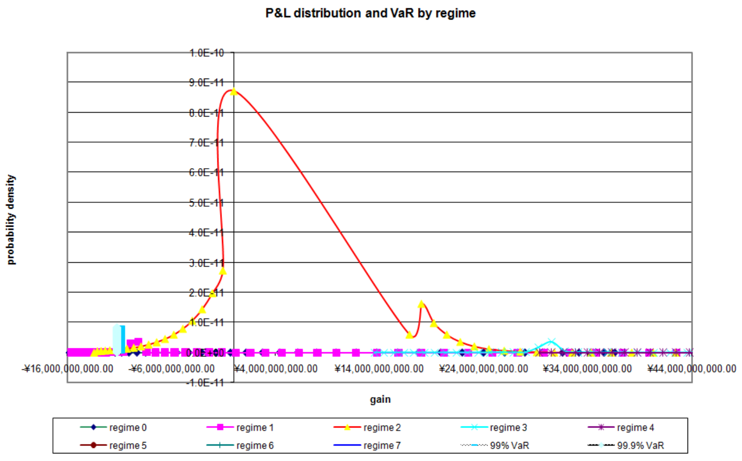


Figure 20: Endogenous profit and loss distribution. Notice the left tail corresponding to VaR. The model identifies it with an event of change of monetary policy toward lower rates.