Calibration and Simulation of Interest Rate Models in MATLAB

Kevin Shea, CFA
Principal Software Engineer
MathWorks
Outline

- Calibration to Market Data
- Calibration to Historical Data
- Counterparty Credit Risk
- Introduction to MATLAB Production Server
- User Story
# Interest Rate Models

<table>
<thead>
<tr>
<th>Short Rate Models</th>
<th>Market Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vasicek</td>
<td>Heath-Jarrow-Morton</td>
</tr>
<tr>
<td>Cox-Ingersoll-Ross</td>
<td>LIBOR Market Model</td>
</tr>
<tr>
<td>Black-Derman-Toy</td>
<td></td>
</tr>
<tr>
<td>Black-Karasinski</td>
<td></td>
</tr>
<tr>
<td>Hull-White</td>
<td></td>
</tr>
<tr>
<td>G2++</td>
<td></td>
</tr>
</tbody>
</table>
Interest Rate Models

<table>
<thead>
<tr>
<th>Short Rate Models</th>
<th>Market Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vasicek</td>
<td>Heath-Jarrow-Morton</td>
</tr>
<tr>
<td>Cox-Ingersoll-Ross</td>
<td>LIBOR Market Model</td>
</tr>
<tr>
<td>Black-Derman-Toy</td>
<td></td>
</tr>
<tr>
<td>Black-Karasinski</td>
<td></td>
</tr>
<tr>
<td>Hull-White</td>
<td></td>
</tr>
<tr>
<td>G2++</td>
<td></td>
</tr>
</tbody>
</table>

Hull-White

\[ dr = (\theta(t) - ar)dt + \sigma dW(t) \]
## Interest Rate Models

<table>
<thead>
<tr>
<th>Short Rate Models</th>
<th>Market Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vasicek</td>
<td>Heath-Jarrow-Morton</td>
</tr>
<tr>
<td>Cox-Ingersoll-Ross</td>
<td>LIBOR Market Model</td>
</tr>
<tr>
<td>Black-Derman-Toy</td>
<td></td>
</tr>
<tr>
<td>Black-Karasinski</td>
<td></td>
</tr>
<tr>
<td>Hull-White</td>
<td></td>
</tr>
<tr>
<td>G2++</td>
<td></td>
</tr>
</tbody>
</table>

**Hull-White**

\[
dr = (\theta(t) - ar) dt + \sigma dW(t)
\]

**G2++**

\[
\begin{align*}
    r(t) &= x(t) + y(t) + \varphi(t) \\
    dx(t) &= -ax(t)dt + \sigma dW_1(t) \\
    dy(t) &= -bx(t)dt + \eta dW_2(t) \\
    dW_1(t)dW_2(t) &= \rho dt
\end{align*}
\]
Calibrate to Market Data

- Choose a set of liquid calibration instruments – typically caps, floors, swaptions.
- Find the set of model parameters that matches as closely as possible the observed prices.

\[
\sum_{k=0}^{n} (P_i - \hat{P}_i(\theta))^2
\]

\(P_i\): Market Price
\(\hat{P}_i\): Model Price
\(\theta\): Model Parameters
Calibrate to Swaption Volatility Matrix

- Solve nonlinear data-fitting problems with \texttt{lsqnonlin}.
- Choose model and target.
- Flexibility with constraints, starting parameters, weights.

\[
\sum_{k=0}^{n} (P_i - \hat{P}_i(\theta))^2
\]

```
>> objfun = @(x) Price - swaptionbyhwcf(RateSpec,x(1),x(2),Strike,Exp,Mat));
>> x0 = [.1 .01];
>> lb = [0 0];
>> ub = [1 1];
>> HWParams = lsqnonlin(objfun,x0,lb,ub,options);
```
More Optimization Capabilities


- Global Optimization Toolbox
  - Simulated Annealing
  - Pattern Search
  - Genetic Algorithm
  - Global Optimization
  - Multistart Framework

```matlab
>> problem = createOptimProblem('lsqnonlin','objective',objfun,'x0',x0,'lb',lb,'ub',ub,'options',options);
>> ms = MultiStart;
>> [x,f] = run(ms,problem,20)
```

Calibrate to Historical Data

- Use MLE to Calibrate to Short Rate Data
- Use Kalman Filter to Calibrate to Historical Yield Curve Data
- Import Data Using Datafeed Toolbox™ and the FRED Datafeed.


Calibrate CIR Model using MLE

\[ dr(t) = a(b - r)dt + \sigma \sqrt{r}dW(t) \]

- \( a \): mean reversion speed
- \( \sigma \): volatility of the short rate
- \( b \): level
- \( W \): Brownian motion

```matlab
>> ShortRateData = fetch(fred,'GS3M', '01-Jan-1995', '31-Dec-2010');
>> [CIR_Param,CIR_CI] = mle(ShortRates, 'pdf', {@cirpdf, dt},
   'start',x0,'lowerbound', [0 0 0], 'optimfun','fmincon')
CIR_Param =
   0.1285    0.0052    0.0523
CIR_CI =
   -0.0286   -0.0111    0.0471
   0.2856    0.0216    0.0576
```
Stochastic Differential Equation Models

- Suite of models including: \( \text{bm, gbm, cir, hwv, heston, cev} \)
- Simulate methods
- Framework for creating custom models

```
>> CIR = cir(a, b, Sigma,'StartState',r0);
>> dt = 1/252;
>> nPeriods = 252*2;
>> nTrials = 10000;
>> Paths = simulate(CIR,nPeriods,'nTrials',nTrials,'DeltaTime',dt);
```
Calibrate using Kalman Filter

- Formulate models as state space systems.
- Use Kalman filter to estimate parameters.
- Estimate parameters from historical yield curves.
State Space formulation for G2++ Model

Transition Equation

\[ x_t = Ax_{t-1} + B\mu \]

Measurement Equation

\[ y_t = Cx_t + D\epsilon + E \]

\[
A = \begin{bmatrix}
e^{-a\Delta t} & 0 \\
0 & e^{-b\Delta t}
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
\sigma \sqrt{\frac{1 - e^{-2a\Delta t}}{2a}} & 0 \\
0 & \eta \sqrt{\frac{1 - e^{-2b\Delta t}}{2b}}
\end{bmatrix}
\]

State Space Model

New state space model, **ssm** in Econometrics Toolbox™.

- Supports time-invariant and time-varying, linear state-space models.
- Perform univariate and multivariate time-series data analysis.
- Functionality to: estimate, filter, smooth, simulate, forecast

```matlab
G2PPSSM = ssm(@(Params)g2ppssm(Params,dT,tenor));
[G2PP,Param] = estimate(G2PPSSM,Data,x0,'lb',lb,'ub',ub);
```
Interest Rate Model Simulation

Specify models and simulate entire term structure

- Support for Hull White, G2++ and LiborMarketModel.
- `simTermStructs` simulates entire term structure.

```matlab
>> G2PP = LinearGaussian2F(RateSpec,a,b,sigma,eta,rho);
>> nDates = 24;
>> DT = 1/12;
>> nTrials = 10000;
>> Paths = simTermStructs(G2PP,nDates,'NTRIALS',nTrials,'DeltaTime',DT);
```
Swap Portfolio

- Store data in a MATLAB Table.
- Easy to read in data.
- Tabular display.

>> SwapPort = readtable('SwapPortfolio.xlsx')

SwapPort =

<table>
<thead>
<tr>
<th>Notional</th>
<th>Maturity</th>
<th>RecType</th>
<th>PayType</th>
<th>RecRate</th>
<th>PayRate</th>
<th>RecReset</th>
<th>PayReset</th>
</tr>
</thead>
<tbody>
<tr>
<td>1e+07</td>
<td>'1/15/2018'</td>
<td>1</td>
<td>0</td>
<td>0.031</td>
<td>10</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>5e+06</td>
<td>'2/15/2018'</td>
<td>0</td>
<td>1</td>
<td>20</td>
<td>0.032</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>1e+06</td>
<td>'3/15/2019'</td>
<td>1</td>
<td>0</td>
<td>0.033</td>
<td>30</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>2e+06</td>
<td>'4/15/2019'</td>
<td>0</td>
<td>1</td>
<td>40</td>
<td>0.034</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>1e+07</td>
<td>'5/15/2020'</td>
<td>1</td>
<td>0</td>
<td>0.036</td>
<td>50</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>7e+06</td>
<td>'6/15/2020'</td>
<td>0</td>
<td>1</td>
<td>65</td>
<td>0.036</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>7.5e+06</td>
<td>'7/15/2021'</td>
<td>1</td>
<td>0</td>
<td>0.0385</td>
<td>70</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>8e+06</td>
<td>'8/15/2021'</td>
<td>0</td>
<td>1</td>
<td>75</td>
<td>0.04</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>3e+06</td>
<td>'9/15/2022'</td>
<td>1</td>
<td>0</td>
<td>0.039</td>
<td>85</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>3.5e+06</td>
<td>'10/15/2022'</td>
<td>0</td>
<td>1</td>
<td>95</td>
<td>0.04</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>
Valuing the Portfolio

- Value portfolio using `swapbyzero`
- Use `parfor` to loop over simulation dates.

```matlab
>> Values = zeros(nDates,nSwaps,nTrials);
>> parfor dateidx=1:nDates
    Values(dateidx,:,:,:) = swapbyzero(...)
end
```
Counterparty Credit Risk Functions

Compute exposures and CCR profiles

- Support for computing credit exposures.
- Support for computing various credit exposure profiles, including potential future exposure and expected exposure.

$$\text{Exposures} = \text{creditexposures}(\text{Values});$$
$$\text{Profiles} = \text{exposureprofiles}(\text{SimDates}, \text{Exposures});$$
Computing Credit Valuation Adjustment

- Compute exposure from exposure profiles
- Compute default probabilities from \texttt{cdsbootstrap}

\[ CVA = (1 - R) \int_0^T DiscExp(t) dPD(t) \]

\begin{verbatim}
>> ProbData = cdsbootstrap(ZeroCurve, CDSData, Settle);
>> Recovery = 0.4;
>> CVA = (1-Recovery)*sum(discEE(2:end,:).*diff(ProbData(:,2)));
CVA = 12244.32
\end{verbatim}
Third Party Interfaces

Access third party analytics

- Support for accessing Numerix® instruments and risk models.
- Support for accessing FinCAD through the F3 Toolbox for MATLAB.
MATLAB Production Server™

- Directly deploy MATLAB programs into production
  - Centrally manage multiple MATLAB programs and runtime versions
  - Automatically deploy updates without server restarts

- Scalable and reliable
  - Service large numbers of concurrent requests
  - Add capacity or redundancy with additional servers

- Use with web, database and application servers
  - Lightweight client library isolates MATLAB processing
  - Access MATLAB programs using native data types
What is MATLAB Production Server?

- Enterprise framework for running packaged MATLAB programs

- Server software
  - Manages packaged MATLAB programs & worker pool

- Runtime libraries
  - MATLAB Compiler Runtime (MCR)

- Lightweight client library (.NET & Java)
  - Request MATLAB programs (functions)
Use Case: Risk Manager

- Web Interface
- Monitor exposure profiles and CVA for each counterparty.
- Different simulation models, calibration methods
- CVA sensitivity analysis
- What-if scenarios, stress scenarios
Client Library

- Request MATLAB programs running on MATLAB Production Server

- Two libraries available
  - .NET: MathWorks.MATLAB.ProductionServer.Client.dll
  - Java: mps_client.jar

- HTTP or HTTPS based communication protocol
- Automatic data marshalling between .NET or Java types and MATLAB types
- Dynamic Invocation API
Use Case: Trader

- Spreadsheet front end
- Analyze a new trade
  - Use different models and calibration methods
  - Compute CVA
  - Compute new exposures
Desktop Application

- Request analytics on MATLAB Production Server from Microsoft® Excel

- MATLAB Builder™ EX
  - Generate CTF and corresponding add-in
  - Support 32 and 64bit Excel environments with same CTF/add-in pair
  - Deploy without admin rights
Integrating with IT systems

MATLAB Compiler™

Web Server

MATLAB Production Server
- Portfolio Optimization
- Pricing
- Risk Analytics

Application Server

Database Server

Web Applications

Desktop Applications

Excel®
Developing a New Interest Rate Model

**Challenge**
Develop a new interest rate model based on work by Deguillaume, Rebonato and Pogudin (2013).

Solution
The model was developed in MATLAB and can be calibrated to multiple currencies and markets.

Deployment
The model was then deployed to customers via the SmartModels Excel Add-In interface.

Summary

- **Calibration Approaches**
  - Market Data: `lsqnonlin`
  - Historical Data: `mle, ssm`

- **Monte Carlo Simulation in MATLAB**
  - `bm, gbm, cir, hwv, heston, cev`
  - `HullWhite1F, LinearGaussian2F, LiborMarketModel`

- **Counterparty Credit Risk**
  - `creditexposures, exposureprofiles`
  - `cdsbootstrap`

- **Enterprise deployment with MATLAB Production Server**