Fixed Income Modelling and Monte Carlo Pricing in C++

Ashara Dangallage Don

Project scope	Discount curve bootstrapping from swap rates, Vasicek term-structure analytics, Monte Carlo pricing, validation diagnostics and simple parameter sensitivity testing
Language	C++17

1 Abstract

This report documents a single C++ workflow that combines two connected pieces of fixed-income work. The first half builds a discount factor curve from quoted par swap rates using a straightforward bootstrap with linear interpolation of discount factors. The second half uses the Vasicek short-rate model to generate analytical term-structure quantities, simulate short-rate paths and price interest-rate payoffs by Monte Carlo.

The main point of the project was not to make the most sophisticated rates model possible but rather, to build something that is clear, reproducible and easy to explain from end to end: starting with liquid market swap quotes, recovering a working curve, then moving on into a stochastic short-rate model and testing how pricing outputs behave under simulation.

2 Project objective

There were two practical goals.

First, I wanted a clean implementation of discount curve construction that could take a small set of semi-annual swap quotes and turn them into usable discount factors, zero rates and swap prices. That part is the static term-structure piece.

Second, I wanted to connect that with a simple stochastic rates model. The Vasicek model was a sensible choice because it is mathematically manageable, still gives closed-form bond pricing and is easy to simulate exactly in discrete time. That makes it a good model for checking whether the numerical side of the project is behaving the way it should.

Taken together, the workflow is:

1. bootstrap the discount curve from market swap rates;
2. recover curve-based pricing quantities from that bootstrap;
3. compute model-implied bond prices, spot rates and forward rates under Vasicek;
4. simulate short-rate paths under the same parameter set;
5. price a bond and two option-style payoffs by Monte Carlo;
6. examine how the outputs move when mean reversion and volatility are changed.

3 Implementation summary

The project was implemented in C++17 as a single workflow file. The code is organised around a few small classes rather than one long procedural script.

- **DiscountCurve** stores discount factors and handles interpolation plus simple rate conversions.
- **Swap** generates semi-annual cashflow times, prices a fixed-rate swap from the curve and solves for a par rate.
- **Bootstrapper** iterates through the quoted maturities and solves for the unknown discount factor at each new node.
- **Vasicek** stores the model parameters and provides the affine bond-pricing functions needed for both analytics and Monte Carlo pricing.
- **ProjectConfig** centralises market inputs, model parameters, Monte Carlo settings and sensitivity grids so the whole run can be reproduced from one place.

4 Inputs and base-case assumptions

4.1 Market swap inputs

The curve bootstrap uses the following annual par swap rates with semi-annual fixed-leg cashflows.

Maturity (years)	Par rate
1	2.64%
2	3.02%
3	3.42%
5	4.11%
7	4.56%
10	4.97%

4.2 Vasicek and simulation settings

The base-case Vasicek parameter set and Monte Carlo controls used in the final run were:

Setting	Value	**Setting**	Value
Mean reversion speed a	0.25	**Time step** dt	1/252
Long-run level b	0.10	**Random seed**	1234
Volatility σ	0.02	**Histogram paths**	30,000
Initial rate r_0	0.07	**Bond paths**	30,000
Bond maturity T	5.0	**Option paths**	30,000
Short-rate call strike K	0.11	**Sensitivity paths**	10,000
Short-rate call notional M	1,000,000	**Histogram bins**	50
Bond call (T_1, T_2, K)	$(1, 5, 0.69)$	**Curve grid points**	600

The sensitivity analysis uses $a \in \{0.10, 0.25, 0.50\}$ and $\sigma \in \{0.01, 0.02, 0.04\}$.

5 Discount curve construction

The curve bootstrap starts from $D(0) = 1$ and processes swap maturities in ascending order. At each maturity, the fixed-leg par condition is written using known discount factors and one new unknown terminal discount factor. Any coupon dates that fall between quoted maturities are filled by linear interpolation on discount factors. Once the curve has been constructed, it can be queried for zero-coupon prices, zero rates, swap values and implied par rates at maturities that were not quoted directly.

5.1 Q1 outputs

The main curve results from the run are:

Quantity	Result
1-year zero-coupon bond price	0.9741125711
2-year continuously-compounded zero rate	3.0033%
7-year swap price at 4.42% fixed	-0.0084729031
9-year implied par swap rate	4.8309%

These numbers are sensible. In particular, the 7-year swap prices slightly below par because its fixed rate (4.42%) is below the 7-year par rate built into the curve (4.56%).

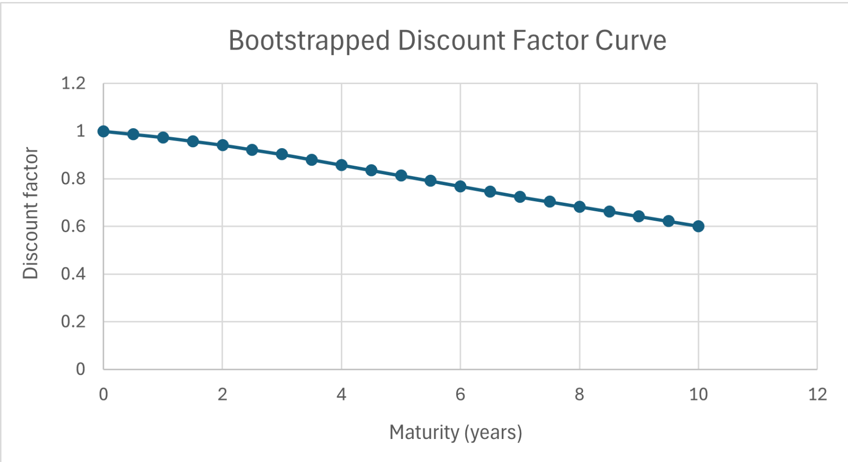


Figure 1: Bootstrapped discount factor curve. The monotone decline is inline with what is expected from a positive-rate environment.

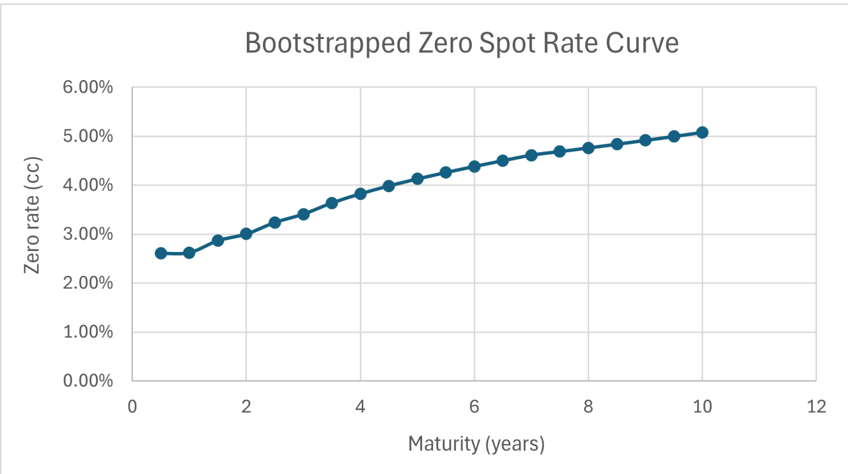


Figure 2: Bootstrapped continuously-compounded zero spot rate curve. The upward slope indicates a higher implied discount rate for longer maturities under the chosen market inputs.

6 Vasicek term-structure framework

Under the Vasicek model, the short rate follows

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

and the zero-coupon bond price takes the standard affine form

$$P(0,T) = \exp\big(A(T) - B(T)r_0\big).$$

Using the model's closed-form functions $A(T)$ and $B(T)$, the code also computes the continuously-compounded spot curve and instantaneous forward curve. For the 5-year point in the base case, the analytical outputs are:

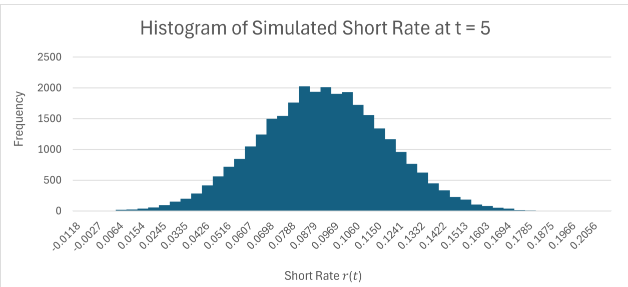
Quantity	Result
$P(0,5)$	0.6631385099
$R_{cc}(0,5)$	0.0821542794
$f(0,5)$	0.0897758148

This analytical block is useful for two reasons. It provides model-implied term-structure quantities directly and it also gives a benchmark against which the Monte Carlo bond price can be checked against.

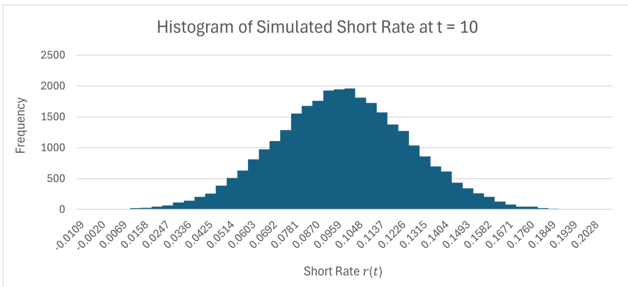
7 Monte Carlo engine and distribution checks

The simulation uses the exact conditional Gaussian transition of the Vasicek process rather than an Euler step. That keeps the short-rate process aligned with the model's own distribution and avoids adding avoidable path-generation bias. Pathwise discount factors are then estimated using a trapezoidal approximation to the time integral of the short rate.

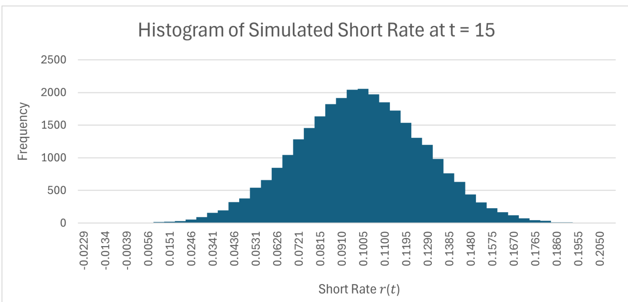
Across the sampled horizons for the short-rate histograms, the centre of the distribution moves gradually toward the long-run level of 10%, while the distribution remains roughly bell-shaped, which is what the Gaussian Vasicek setup suggests.



(a) $t = 5$ years



(b) $t = 10$ years



(c) $t = 15$ years

Figure 3: Histograms of the simulated short rate at the representative horizons. The mean shifts upward toward the long-run level, while the overall shape remains close to normal.

8 Pricing results and validation

The Monte Carlo engine is used to price three objects under the base-case Vasicek parameters:

1. a 5-year zero-coupon bond with unit principal;
2. a short-rate call payoff of the form $M(r_T - K)^+$ with $K = 0.11$, $M = 1{,}000{,}000$ and $T = 5$;
3. a bond call with expiry $T_1 = 1$, bond maturity $T_2 = 5$ and strike $K = 0.69$.

The validation output from the final exported table is shown below.

Instrument	MC price	Std. error	Rel. error	95% CI	Exact / benchmark	Abs. diff
5Y zero-coupon bond	0.664023	0.001049	0.158%	[0.661967, 0.666080]	0.663139	0.000885
Short-rate call	2452.470261	102.836537	4.19%	[2250.910648, 2654.029874]	–	–
Bond call	0.026102	0.000451	1.73%	[0.025218, 0.026986]	–	–

Table 1: Validation summary taken from the exported `q2_validation` output.

Thus, a few things are clear:

The 5-year zero-coupon bond estimate is close to the analytical benchmark in absolute terms, which is a good sign that the simulation and discounting logic are broadly behaving as intended. The relative error is slightly above the original 0.1% coursework target in this reported table, but the gap is small and the confidence interval still contains the exact value.

The two option estimates are directionally sensible, but their reported relative errors are wider than the original target threshold of 1%. In other words, the pricing framework works, but the short-rate option in particular would need a larger simulation budget before I would call the estimate tight. It's important to note that this is not a failure of the model itself, but rather, a convergence issue.

9 Sensitivity analysis

A small sensitivity study was added so the project to understand the inputs dynamics better. The idea is simple: change mean reversion and volatility on a small grid, then look at how the outputs move.

9.1 Bond call price versus volatility

For the bond call, the grid used was $\sigma \in \{0.01, 0.02, 0.04\}$ and $a \in \{0.10, 0.25, 0.50\}$. The resulting prices are:

$\sigma \setminus a$	0.10	0.25	0.50
0.01	0.040422	0.020486	0.004302
0.02	0.048244	0.026577	0.008153
0.04	0.070170	0.041410	0.017433

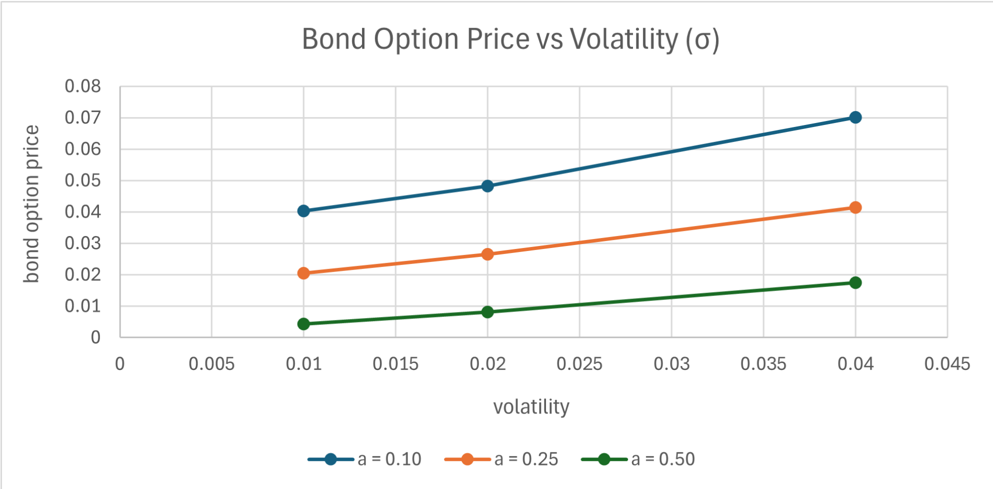


Figure 4: Bond call price versus volatility for three values of the mean reversion parameter a. Higher volatility increases option value, while faster mean reversion lowers it in this setup.

The pattern is intuitive. More volatility increases upside optionality, so the bond call becomes more valuable. On the other hand, increasing a pulls the short rate back toward the long-run mean more aggressively, which dampens dispersion in discounting and reduces the option value in this parameter range.

9.2 5-year zero-coupon bond price versus mean reversion

For the 5-year bond, the same parameter grid gives the following values:

$a \setminus \sigma$	0.01	0.02	0.04
0.10	0.683517	0.686510	0.698610
0.25	0.661346	0.663139	0.670358
0.50	0.641170	0.642064	0.645651

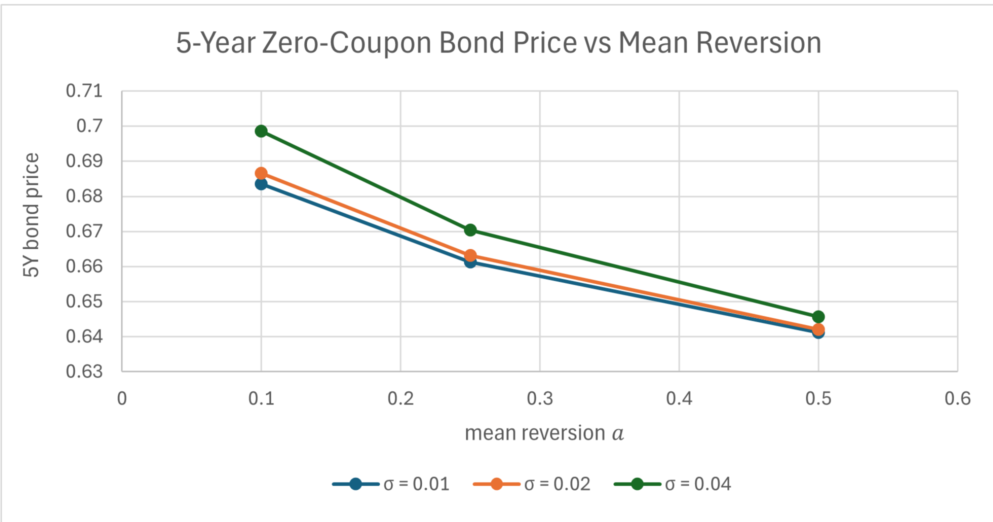


Figure 5: 5-year zero-coupon bond price versus mean reversion for three volatility levels. In this parameter set, stronger mean reversion reduces the bond price, while higher volatility partly offsets that effect.

The main takeaway here is that the bond price falls as a rises. In this setup, a faster pull toward the long-run level of 10% pushes expected future rates higher relative to the starting level of 7%, which increases discounting and lowers the 5-year bond price. Increasing volatility works in the opposite direction here and lifts the bond price modestly.

10 Limitations

This is a solid teaching and portfolio project, but it still has clear limits.

- The curve uses linear interpolation on discount factors, which is easy to follow but fairly basic.
- The Vasicek model allows negative rates and is not calibrated to a live market dataset.
- The Monte Carlo validation shows that some payoffs converge more slowly than others; the short-rate call would need more simulation effort for a tighter estimate.
- The sensitivity section is deliberately small and illustrative rather than a full parameter study.

11 Conclusion

The project achieves what it set out to do. It links curve construction and model-based pricing in one coherent C++ workflow and it does it in a way that is easy to inspect.

On the deterministic side, the bootstrap produces a sensible discount curve, an upward-sloping zero curve and internally consistent swap pricing outputs. On the stochastic side, the Vasicek implementation provides both analytical term-structure quantities and a Monte Carlo engine that can be reused across different payoffs. The validation results show that the bond pricing logic is close to the analytical benchmark, while also making it clear that option estimates need more paths if tighter precision is required. The sensitivity plots add one more useful layer by showing how the pricing outputs respond to changes in mean reversion and volatility.