

Model Validation Report: Down-and-Out Barrier Call (QuantLib BarrierOption)

MatLogica — Automated Tape-Based Validation

October 2026 v15

Contents

1	Introduction	3
1.1	Summary	3
1.2	Regulatory Framework	3
1.3	Methodology	3
1.4	Key Terms	3
2	Model Description	4
2.1	Model Inventory Entry	4
2.2	Feeding Models	4
2.3	Input Classification	4
2.4	Engine Configurations	4
2.5	Pricing Methodology	5
2.5.1	Engine 1: Analytic (Merton–Reiner–Rubinstein)	5
2.5.2	Engine 2: Finite Differences (Douglas Scheme)	6
2.5.3	Engine 3: Binomial Tree (CRR / JR / LR)	7
2.6	Tape Structure Comparison	8
2.7	Scope and Limitations	8
3	Test Configuration	9
4	Testing	9
4.1	Cross-Engine Benchmark	9
4.2	Implementation Accuracy (Tape vs Fresh)	9
4.3	Spot Sensitivity Ladder	9
4.4	Volatility Sensitivity Ladder	10
4.5	Rate Sensitivity Ladder	10
4.6	Frozen Branch Analysis	11
4.7	Convergence Tests	12
4.8	Barrier Parity: DI + DO = European	12
4.9	Materiality Analysis	13
4.10	Stress and Boundary Tests	13
4.11	Back-testing	13
5	Findings	13
6	Validation Outcome	14

1 Introduction

1.1 Summary

Model Name	Down-and-Out Barrier Call (BSM)
Model ID	[TBA — to be assigned by Model Governance]
Model Tier	[TBA — to be assigned by Model Governance]
Lifecycle Stage	Use
Revalidation	[Frequency set by model tier, per PRA SS1/23]
Previous Validation	N/A (initial validation)
Engine Configurations	5: Analytic, FD, Binomial CRR/JR/LR
Validation Date	October 02, 2026
Outcome	Conditionally Approved
Findings	8
Conditions	F1, F4, F5, F8 (Section 5)

1.2 Regulatory Framework

This validation is conducted in accordance with:

- **PRA SS1/23**, *Model risk management principles for banks* (updated April 2026) — Principle 4: independent validation; Principle 3: documentation sufficient for an independent third party.
- **Fed SR 26-2** (April 2026, superseding SR 11-7) — conceptual soundness, effective challenge, on-going monitoring and outcomes analysis.
- **ECB Guide to internal models** (revised 2025–2026) — independent internal validation and quantitative testing.

1.3 Methodology

Separate tapes recorded for each engine configuration. Tape A (source tracing, `trace=True`, with every literal as an input, `trace_inputs_as_random`): per-engine introspection — call structure, mathematical formula extraction. Tape B (analytic engine, production mode): replay with 4 marked inputs (S_0 , r , q , σ) for AAD Greeks, sensitivity ladders, and stress tests.

1.4 Key Terms

Tape Directed acyclic graph of mathematical operations recorded from one execution. Each node carries source file, line, function.

Tape replay Re-evaluation of the recorded graph with modified inputs. Millions of evaluations per second.

AAD Adjoint Algorithmic Differentiation — exact first-order derivatives from a single reverse pass.

Frozen branch A conditional whose predicate was passive at recording time. The branch taken is hard-wired; replay outside that domain may give incorrect results.

Fresh pricing Independent QuantLib evaluation (not tape replay), used as benchmark.

2 Model Description

2.1 Model Inventory Entry

Business Use	Pricing, risk (delta, vega, rho, gamma, theta)
Asset Class	Equity / FX
Product Scope	Single-asset barrier options (down/up, in/out, call/put)
Engine Configurations	Analytic (production), FD, Binomial CRR/JR/LR (benchmark)
Outputs	NPV, delta, gamma, vega, theta, rho
Input Dependencies	Yield curve model, dividend model, volatility surface model

2.2 Feeding Models

Validation reports for the following feeding models have not been provided. This results in Finding F1 (material), which blocks full approval.

Feeding Model	Provides	Model ID	Validation
Yield Curve Model	Risk-free discount factor $D(0, T)$	[TBA]	[Not provided]
Dividend Model	Continuous dividend yield q	[TBA]	[Not provided]
Vol Surface Model	Black-Scholes implied vol σ	[TBA]	[Not provided]

2.3 Input Classification

The table below lists all model inputs as identified from the tape. Classification (observable / calibrated / marked) is a **model governance decision** and must be confirmed by the model owner. The tape shows how each input enters the computation; it does not determine the production data source.

Concrete values used for testing are in §3.

Parameter	Class	Tape evidence / Production source
S_0 (spot price)	[To be confirmed]	Tape: user input. Production: [equity feed — to be specified by model owner]
σ (Black-Scholes implied volatility)	[To be confirmed]	Tape: user input via <code>BlackConstantVol</code> (single flat value). Production: [vol surface model, Model ID: TBA]
r (risk-free rate, continuous)	[To be confirmed]	Tape: <code>FlatForward</code> (single flat value). Production: [yield curve model, Model ID: TBA]
q (continuous dividend yield)	[To be confirmed]	Tape: <code>FlatForward</code> (single flat value). Production: [dividend model, Model ID: TBA]
K (strike price)	Trade parameter	Contract specification
B (barrier level)	Trade parameter	Contract specification (down-and-out)
Day count convention	Internal	Hardcoded in QuantLib library (Actual/365 Fixed). Not configurable without code change

2.4 Engine Configurations

The model supports five engine configurations (Table 1). The production engine is **Analytic** (closed-form Merton–Reiner–Rubinstein), selected by [Model Governance / Model Owner]. Benchmark engines (FD, Binomial CRR/JR/LR) are used for independent validation.

Table 1: Engine configurations with tape statistics

Engine	Role	Tape entries	Leaves	A2P sites
Analytic	Production	473	261	30
FD BlackScholes	Benchmark	479544	127935	71
Binomial CRR 200	Benchmark	441136	145688	48
Binomial JR 200	Benchmark	476681	130241	42
Binomial LR 200	Benchmark	435918	211867	44

2.5 Pricing Methodology

A separate tape was recorded for each engine configuration (Table 1); the mathematical description below is extracted from these tapes. Tape evidence is presented in per-engine Tape Verification Tables, separately from the formulas. The  icon under each formula, every function name in those tables, and every source site in Sections 4.6 and 4.10 link to the lines recorded on the tape in the public QuantLib source (github.com/lballabio/QuantLib, commit 367ce80ae4f2, release 1.41). The traced build is the AADC port of that release, so a line may differ slightly from upstream.

2.5.1 Engine 1: Analytic (Merton–Reiner–Rubinstein)

Under BSM with constant risk-free rate r , continuous dividend yield q , and volatility σ , the down-and-out call with barrier $B < K$ is given by the Merton (1973), Reiner and Rubinstein (1991) formula.

The drift parameter μ and its derived quantity λ :

$$\mu = \frac{r - q}{\sigma^2} - \frac{1}{2}, \quad \lambda = 1 + \mu \quad (1)$$

 `mu()`  `muSigma()`

The total standard deviation over $[0, T]$:

$$\hat{\sigma} = \sigma \sqrt{T} \quad (2)$$

 `stdDeviation()`

The building blocks $A(\phi)$ and $C(\phi, \eta)$:

$$A(\phi) = \phi S_0 e^{-qT} \Phi(\phi x_1) - \phi K e^{-rT} \Phi(\phi x_1 - \phi \hat{\sigma}) \quad (3)$$

$$C(\phi, \eta) = \phi S_0 e^{-qT} \left(\frac{B}{S_0} \right)^{2\lambda} \Phi(\eta y_1) - \phi K e^{-rT} \left(\frac{B}{S_0} \right)^{2\lambda-2} \Phi(\eta y_1 - \eta \hat{\sigma}) \quad (4)$$

 `A()`  `C()`

where

$$x_1 = \frac{\ln(S_0/K)}{\hat{\sigma}} + \lambda \hat{\sigma}, \quad y_1 = \frac{\ln(B^2/(S_0 K))}{\hat{\sigma}} + \lambda \hat{\sigma} \quad (5)$$

 `A()`  `C()`

For a down-and-out call with $B < K$:

$$V_{DO} = A(1) - C(1, 1) \quad (6)$$

 `calculate()`  `triggered()`

If $S_0 \leq B$ at inception, $V_{DO} = 0$ (already knocked out).

Tape Verification. Tape: 473 entries, 261 leaves, 30 conversion sites.

Table 2: Analytic engine: formula $\rightarrow$ tape mapping

Equation	Description	Tape function	Entries
(1)	drift μ, λ	<code>mu(), muSigma()</code>	12, 4
(2)	$\hat{\sigma} = \sigma\sqrt{T}$	<code>stdDeviation()</code>	6
(3)	building block $A(\phi)$	<code>A()</code>	13
(4)	building block $C(\phi, \eta)$	<code>C()</code>	23
(5)	x_1, y_1	<code>inside A(), C()</code>	incl. above
(6)	$V_{DO} = A - C$	<code>calculate()</code>	6
—	rebate term (zero here)	<code>F()</code>	1
—	Φ via error function	<code>operator()</code> in <code>errorfunction.cpp</code>	135
—	inception check $S_0 \leq B$	<code>triggered()</code>	1

Source: `analyticbarrierengine.cpp`, `barrieroption.cpp`.

2.5.2 Engine 2: Finite Differences (Douglas Scheme)

The FD engine solves the BSM PDE on a 1D spatial mesh with an absorbing boundary at B :

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r - q)S \frac{\partial V}{\partial S} - rV = 0, \quad V(B, t) = 0 \quad (7)$$

The spatial domain is discretised in log-spot space $x = \ln S$ on a uniform grid whose lower edge is the barrier (for a down-and-out option):

$$x_i = x_{\min} + i h, \quad h = \frac{x_{\max} - x_{\min}}{N - 1}, \quad x_{\min} = \ln B, \quad x_{\max} = \ln S_0 + \sigma\sqrt{T} \Phi^{-1}(1 - \varepsilon) \quad (8)$$

 `Uniform1dMesher`  `FdmBlackScholesMesher`

The drift-diffusion operator L is:

$$L_{i,j} = \frac{1}{2}\sigma^2(t) D_{i,j}^{(2)} + s D_{i,j}^{(1)} - r \delta_{i,j}, \quad s = (r - q) - \frac{1}{2}\sigma^2(t) \quad (9)$$

 `setTime()`  `axpyb()`

where $D^{(1)}$ and $D^{(2)}$ are derivative stencils (Appendix A, Eqs. (23)–(24)).

Time stepping uses the Douglas (Crank–Nicolson, $\theta = 0.5$) scheme:

$$(I - \theta \Delta t L) V^n = V^{n+1} + (1 - \theta) \Delta t L V^{n+1} \quad (10)$$

 `step()`

The tridiagonal system is solved via the Thomas algorithm:

$$\beta_0 = d_0, \quad y_0 = r_0 / \beta_0 \quad (11)$$

$$\beta_i = d_i - l_i u_{i-1} / \beta_{i-1}, \quad y_i = (r_i - l_i y_{i-1}) / \beta_i \quad (12)$$

$$x_N = y_N, \quad x_i = y_i - u_i x_{i+1} / \beta_i \quad (13)$$

 `solve_splitting()`  `solveFor()`

Barrier: $V(B, t) = 0$ via absorbing mesh node. Solution extracted via cubic spline (14):

$$V(x) = a + b(x - x_i) + c(x - x_i)^2 + d(x - x_i)^3 \quad (14)$$

 `value()`  `derivative()`

Tape Verification. Tape: 479544 entries, 127935 leaves, 71 active-to-passive conversion sites, 41 source files. See Appendix A for stencil details.

Table 3: FD engine: formula $\rightarrow$ tape mapping

Equation	Description	Tape function	Entries
(8)	mesh x_i	<code>Uniform1dMesher, FdmBlackScholesMesher</code>	202, 282
(9)	operator L , drift s	<code>setTime(), axpyb()</code>	606, 70,700
(23)–(24)	$D^{(1)}, D^{(2)}$ stencils	<code>FirstDerivativeOp, SecondDerivativeOp</code>	994, 794
(10)	Douglas time step	<code>step()</code> in <code>douglasscheme.cpp</code>	1,010
(11)–(13)	Thomas algorithm	<code>solve_splitting(), solveFor()</code>	150,591, 1,586
—	$y = Lv$ product	<code>apply()</code>	101,000
(14)	cubic interpolation	<code>value(), derivative()</code>	21, 14
—	barrier $V(B) = 0$	mesh boundary node	—

2.5.3 Engine 3: Binomial Tree (CRR / JR / LR)

The binomial engine constructs a recombining tree for the BSM process. For CRR (Cox–Ross–Rubinstein), the up/down factors and risk-neutral probability are:

$$u = e^{\sigma\sqrt{\Delta t}}, \quad d = 1/u, \quad p = \frac{e^{(r-q)\Delta t} - d}{u - d} \quad (15)$$

 `CoxRossRubinstein`

At each time step i and state j , the spot is:

$$S_{i,j} = S_0 \cdot u^{2j-i}, \quad j = 0, 1, \dots, i \quad (16)$$

 `underlying()`

Payoff at maturity:

$$V_{N,j} = \max(S_{N,j} - K, 0) \quad (17)$$

 `operator()`

Backward induction:

$$V_{i,j} = e^{-r\Delta t} [p V_{i+1,j+1} + (1-p) V_{i+1,j}] \quad (18)$$

 `stepback()`

Barrier adjustment at each time step:

$$V_{i,j} \leftarrow \begin{cases} 0 & \text{if } S_{i,j} \leq B \\ V_{i,j} & \text{otherwise} \end{cases} \quad (19)$$

 `checkBarrier()`  `adjustBarrier()`

with proximity threshold:

$$|S_{i,j} - B| < \varepsilon_{\text{rel}} \cdot B, \quad \varepsilon_{\text{rel}} = 42 \epsilon_{\text{mach}} \approx 10^{-14} \quad (20)$$

 `close_enough()`

Jarrow–Rudd uses drift-adjusted probabilities:

$$u_{\text{JR}} = e^{(r-q-\frac{1}{2}\sigma^2)\Delta t + \sigma\sqrt{\Delta t}}, \quad d_{\text{JR}} = e^{(r-q-\frac{1}{2}\sigma^2)\Delta t - \sigma\sqrt{\Delta t}}, \quad p_{\text{JR}} = \frac{1}{2} \quad (21)$$

 `BinomialTree`

Leisen–Reimer uses the Peizer–Pratt inversion:

$$p_{\text{LR}} = h(d_2), \quad p'_{\text{LR}} = h(d_1), \quad u_{\text{LR}} = \frac{e^{(r-q)\Delta t} p'_{\text{LR}}}{p_{\text{LR}}} \quad (22)$$

 `LeisenReimer`

where h is the Peizer–Pratt function. All three share backward induction (18) and barrier adjustment (19).

Tape Verification. All three variants recorded independently:

Variant	Entries	Leaves	A2P sites	Calibration function
CRR	441136	145688	48	<code>CoxRossRubinstein()</code> (8 entries)
JR	476681	130241	42	<code>BinomialTree()</code> (2 entries, $p = \frac{1}{2}$)
LR	435918	211867	44	<code>LeisenReimer()</code> (21 entries)

Entry counts differ between the variants through their calibration of u , d , p . Backward induction (18), barrier adjustment (19), and proximity check (20) are shared.

Table 4: Binomial engines: formula $\rightarrow$ tape mapping (CRR variant)

Equation	Description	Tape function	Entries
(15)	CRR u , d , p	<code>CoxRossRubinstein()</code>	8
(21)	JR u , d , $p = \frac{1}{2}$	<code>BinomialTree()</code> (JR tape)	2
(22)	LR Peizer–Pratt	<code>LeisenReimer()</code> (LR tape)	21
(16)	node spots $S_{i,j}$	<code>underlying()</code>	137,400
(17)	terminal payoff	<code>operator()</code> in <code>payoffs.cpp</code>	878
(18)	backward induction	<code>stepback()</code>	180,624
(19)	barrier knock-out	<code>checkBarrier()</code> , <code>adjustBarrier()</code>	45,808, 34,278
(20)	proximity threshold	<code>close_enough()</code>	20,399

2.6 Tape Structure Comparison

Engine	Entries	Leaves	A2P sites	Dominant cost
Analytic	473	261	30	<code>operator()</code> in <code>errorfunction.cpp</code> (135)
FD BS	479544	127935	71	<code>solve_splitting</code> in <code>triplebandlinearop.cpp</code> (150,591)
Binomial CRR	441136	145688	48	<code>stepback</code> in <code>bsmlattice.hpp</code> (180,624)

The analytic tape is about $933\times$ smaller than the numerical engines' tapes. The FD and binomial tapes are of similar size but different structure: the dominant recorded function (largest entry count) is shown in the last column.

2.7 Scope and Limitations

- **Barrier type:** Down-and-Out only. Up-and-Out, Down-In, Up-In require separate tapes.
- **Monitoring:** continuous only. Discrete monitoring activates different code paths.
- **Term structures:** flat rates, dividends, volatility. Production term structures require separate validation.
- **MC engine:** not tested.
- **Tape locality:** branches frozen at recording-time inputs; replay is valid only where they do not flip (see §4.6, §4.10).
- **Back-testing:** P&L data not provided (see §5).
- **Tape B:** recorded for analytic engine only; numerical engine Greeks via cross-engine comparison.

3 Test Configuration

The following values are specific to this test date. Production values will differ.

$S_0 = 100$, $K = 100$, $B = 80$ (down-and-out), $r = 4.0\%$, $q = 2.0\%$, $\sigma = 25\%$, $T = 1.0000$ years (evaluation date 24 9 2026, Actual/365 Fixed), continuous monitoring, rebate 0.

Analytic NPV = 10.277694. AAD: $\Delta = 0.607119$, $\mathcal{V} = 30.0775$, $\rho = 45.5119$.

4 Testing

4.1 Cross-Engine Benchmark

Objective. Verify that all five engines produce consistent prices for the same barrier option.

Methodology. Each engine prices the same Down-and-Out call at base parameters. Results compared to the analytic (closed-form) engine.

Pass criteria. Each engine within 0.5% of analytic.

	Engine	NPV	Error vs Analytic	Status
Results.	Analytic	10.277694	0.0000%	Pass
	FD BlackScholes	10.277152	0.0053%	Pass
	Binomial CRR 200	10.271054	0.0646%	Pass
	Binomial JR 200	10.272687	0.0487%	Pass
	Binomial LR 200	10.225445	0.5084%	Fail

Conclusion. Passed for 3 of 4 benchmark engines. Binomial LR 200 is 0.5084% from analytic, outside the 0.5% tolerance. It is a benchmark engine, not the production engine; see Section 5.

4.2 Implementation Accuracy (Tape vs Fresh)

Objective. Verify that tape replay produces identical prices to independent QuantLib evaluation.

Methodology. Replay Tape B with varied inputs; compare to fresh `BarrierOption` construction.

Pass criteria. Relative error $< 10^{-3}$ at all points.

	S_0	r	q	σ	Tape	Fresh	Error
Results.	100	0.04	0.02	0.25	10.277694	10.277694	0.00e+00%
	85	0.04	0.02	0.25	2.339095	2.339095	1.53e-06%
	120	0.04	0.02	0.25	24.623229	24.623229	2.42e-07%
	100	0.02	0.02	0.25	9.389708	9.389708	0.00e+00%
	100	0.06	0.02	0.25	11.209629	11.209629	0.00e+00%
	100	0.04	0.00	0.25	11.436079	11.436079	4.66e-14%
	100	0.04	0.04	0.25	9.203779	9.203779	0.00e+00%
	100	0.04	0.02	0.15	6.817526	6.817536	1.44e-04%
	100	0.04	0.02	0.40	13.713691	13.713691	2.02e-08%
	90	0.04	0.02	0.25	4.761567	4.761567	2.51e-09%
	110	0.04	0.02	0.25	16.927702	16.927702	1.93e-10%

Conclusion. Passed. Maximum relative error 1.44e-04%.

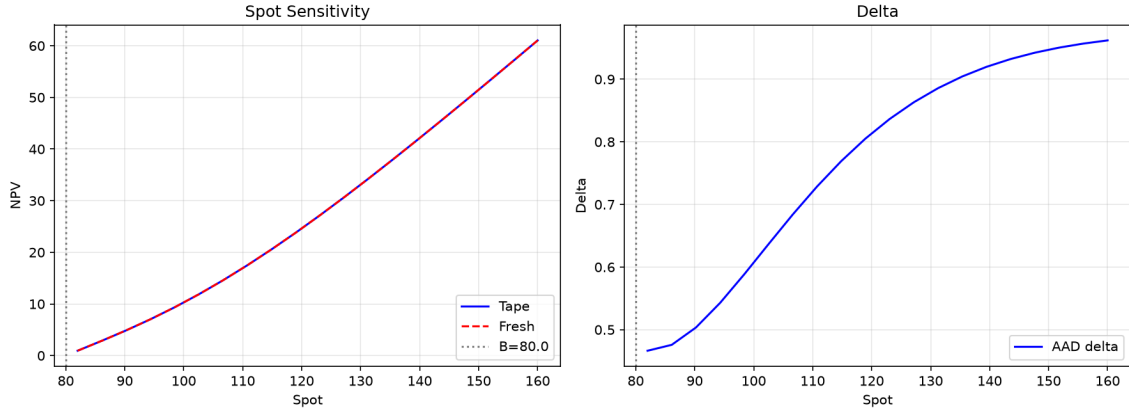
4.3 Spot Sensitivity Ladder

Objective. Verify NPV and delta behaviour across spot range, especially near barrier.

Methodology. Sweep S_0 from 82 to 160. Tape replay with AAD delta vs fresh QuantLib.

Pass criteria. NPV non-negative; $\text{NPV} \rightarrow 0$ as $S \rightarrow B$; delta smooth away from barrier; tape and fresh agree.

Results.



Conclusion. Passed. Tape and fresh agree to 1.2e-03% across the range; NPV is non-negative and falls to 0.9358 at $S_0 = 82$ and to 0.0047 at $S_0 = 80.01$ (Section 4.10) as $S \rightarrow B$. Delta is positive throughout, from 0.466 (minimum, at $S_0 \approx 82$) to 0.962; the largest change between adjacent ladder points is 0.048.

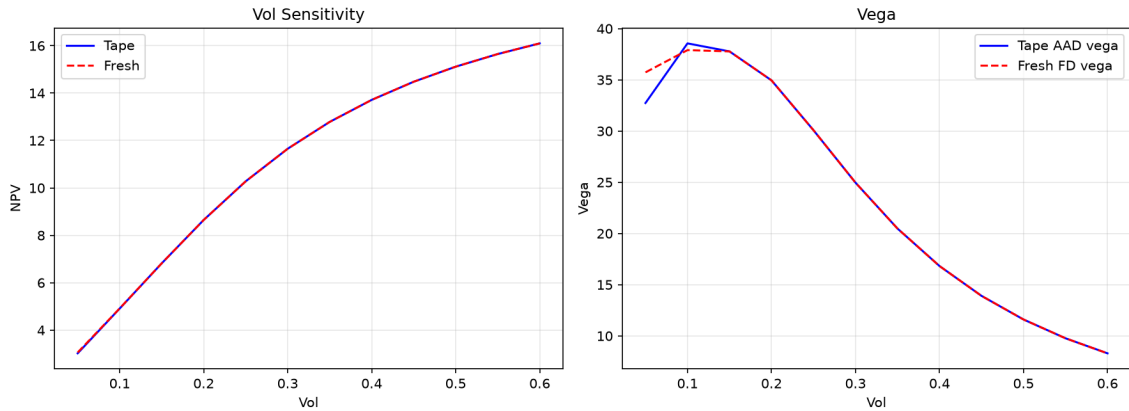
4.4 Volatility Sensitivity Ladder

Objective. Verify the dependence of NPV on σ , including the knock-out effect.

Methodology. Sweep σ from 5% to 60%. Tape replay vs fresh.

Pass criteria. Tape and fresh agree. NPV need not be monotonic in σ (higher vol increases both option value and knock-out probability), so vega may change sign.

Results.



Conclusion. Failed. Tape and fresh differ by more than 0.1% at $\sigma = 5\%$ (1.23%), $\sigma = 10\%$ (0.14%); branches frozen at the recording volatility flip there (Section 4.6); the tape is not valid below about $\sigma = 15\%$. Over the tested range fresh NPV is monotonically increasing in σ (3.06 to 16.10); fresh vega (central difference) is positive throughout (tape AAD vega differs from it by up to 8.4%), from 37.9 (peak, at $\sigma \approx 10\%$) to 8.3. The knock-out probability offsets part of the volatility gain.

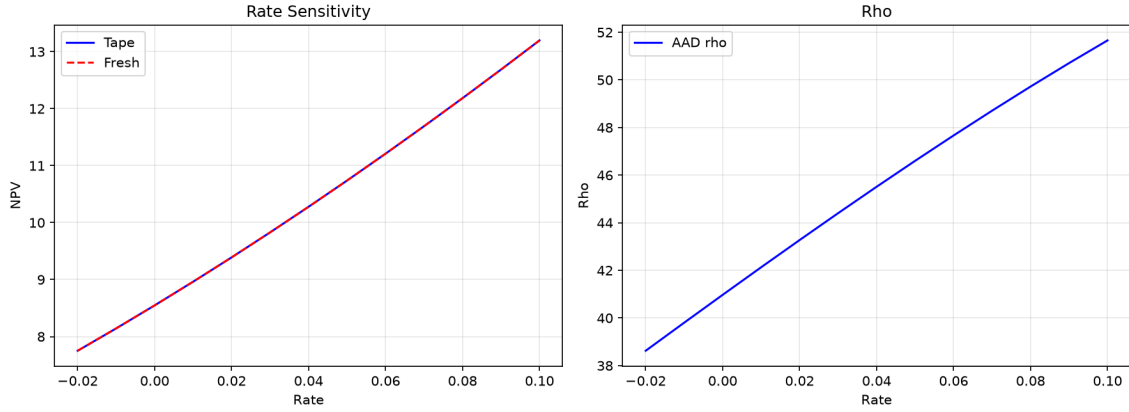
4.5 Rate Sensitivity Ladder

Objective. Verify smooth dependence of NPV on risk-free rate and correctness of rho.

Methodology. Sweep r from -2% to 10% . Tape replay with AAD rho vs fresh QuantLib.

Pass criteria. Smooth, positive rho (higher r increases forward, benefits call). Tape and fresh agree.

Results.



Conclusion. Passed. Tape and fresh agree to $1.7\text{e-}12\%$. NPV is increasing in r and rho is positive throughout (minimum 38.62).

4.6 Frozen Branch Analysis

Objective. Identify the branches frozen at recording time (passive warnings) and establish which of them affect replay within the tested ranges.

Methodology. List every active-to-passive conversion site per engine tape, by source file. Then record the analytic engine a third time with branch-switch registration (Tape C) and replay it at every stress scenario and across the volatility ladder: a registered switch whose sign differs from recording time is a branch the tape has frozen in the other direction.

Pass criteria. No frozen branch flips within the normal operating range.

Results.

Engine	Conversions	Sites	Top source files (conversions)
Analytic	129	30	<code>interestrate.cpp</code> (38), <code>termstructure.cpp</code> (20), <code>voltermstructure.cpp</code> (18), <code>simplequote.hpp</code> (16), <code>errorfunction.cpp</code> (16), <code>yieldtermstructure.cpp</code> (6)
FD	17885	71	<code>triplebandlinearop.cpp</code> (10100), <code>interestrate.cpp</code> (1606), <code>cubicinterpolation.hpp</code> (1564), <code>termstructure.cpp</code> (1204), <code>comparison.hpp</code> (800), <code>simpsonintegral.hpp</code> (589)
Binomial CRR	104925	48	<code>discretizedbarrieroption.cpp</code> (78260), <code>predefined_ops.h</code> (13564), <code>comparison.hpp</code> (11084), <code>timegrid.cpp</code> (1713), <code>lattice.hpp</code> (215), <code>interestrate.cpp</code> (24)

Branches observed to flip on replay within the tested scenarios: `analyticbarrierengine.cpp:210`, `errorfunction.cpp:207`, `errorfunction.cpp:209`, `errorfunction.cpp:210`, `errorfunction.cpp:221`, `errorfunction.cpp:225`, `errorfunction.cpp:227`, `errorfunction.cpp:234`, `interestrate.cpp:79`, `normaldistribution.cpp:36`.

Conclusion. Failed. Within the tested scenarios 10 recorded branch sites flip on replay, in `errorfunction.cpp` (7), `normaldistribution.cpp` (1), `analyticbarrierengine.cpp` (1), `interestrate.cpp` (1). The error-function sites choose the approximation region behind Φ and, at line 225, the sign of the result. A region flip alone gives small errors; a flipped sign branch returns the wrong-signed erf, which is the 25.3% error at $\text{vol}=1.0$. Static classification by function name does not reveal this; replay with switch registration does. See also Section 4.10.

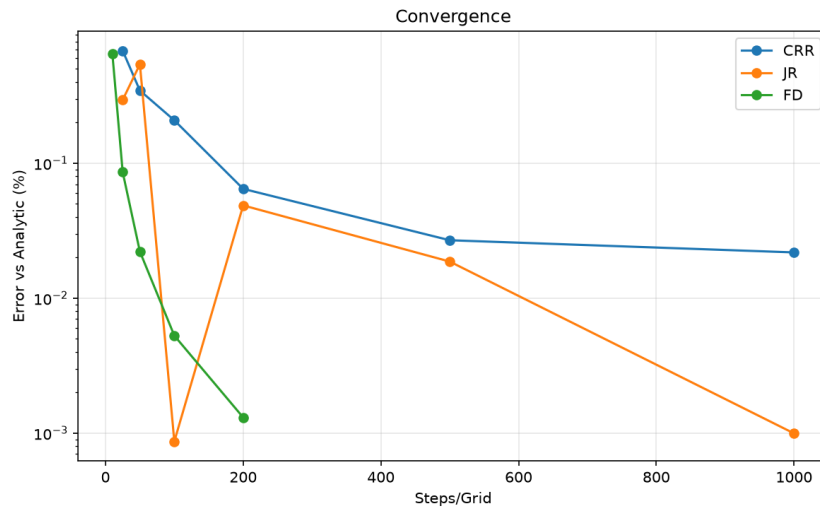
4.7 Convergence Tests

Objective. Verify that numerical engines (FD, Binomial) converge to the analytic solution as resolution increases.

Methodology. Vary resolution (FD: time steps = space grid points; Binomial: tree steps) and compare to analytic NPV = 10.277694.

Pass criteria. Error < 0.1% at production resolution (200 steps / grid points). Monotonic convergence.

	Engine	Steps/Grid	NPV	Error
Results.	CRR	25	10.347873	0.6828%
	CRR	50	10.313253	0.3460%
	CRR	100	10.299080	0.2081%
	CRR	200	10.271054	0.0646%
	CRR	500	10.274932	0.0269%
	CRR	1000	10.279938	0.0218%
	JR	25	10.247402	0.2947%
	JR	50	10.221998	0.5419%
	JR	100	10.277783	0.0009%
	JR	200	10.272687	0.0487%
	JR	500	10.279614	0.0187%
	JR	1000	10.277592	0.0010%
	FD	10	10.210753	0.6513%
	FD	25	10.268854	0.0860%
	FD	50	10.275418	0.0221%
	FD	100	10.277152	0.0053%
	FD	200	10.277560	0.0013%



Conclusion. **CRR:** passed, 0.0646% at 200; **JR:** accuracy met, monotonicity not met, 0.0487% at 200 (error rises 25→50, 100→200); **FD:** passed, 0.0013% at 200. Non-monotonic tree convergence is the known effect of barrier non-alignment with tree nodes.

4.8 Barrier Parity: DI + DO = European

Objective. Verify the model-free identity: Down-In + Down-Out = European.

Methodology. Price DO, DI, and European call with same parameters.

Pass criteria. $|(DO + DI) - EU|/EU < 10^{-10}$.

Results. DO = 10.277694, DI = 0.377613, sum = 10.655307, European = 10.655307. Error: 3.33e-14%.

Conclusion. Passed. Parity holds to machine precision.

4.9 Materiality Analysis

Objective. Identify marked (unobservable) parameters whose variation within plausible ranges produces price changes exceeding the bid/ask spread. For a 100-notional equity barrier option the bid/ask spread is approximately 0.10 in price units. The risk-free rate r (**marked**, hardcoded flat via `FlatForward`) and dividend yield q (**marked**, hardcoded flat) are not calibrated from market instruments and therefore subject to materiality assessment.

Methodology. Perturb each marked parameter by $\pm 1\%$ (100bp) from base and reprice via tape replay. Impact is the total swing $|\text{NPV}(+1\%) - \text{NPV}(-1\%)|$. Compare to the bid/ask threshold.

Pass criteria. All material parameters must have documented valuation controls (IPV, reserve, calibration process). If no control documented, a finding is created.

	Parameter	Base	Range	ΔNPV at -1%	ΔNPV at $+1\%$	Impact	Material?
Results.	r	0.04	$\pm 1\%$	-0.450	+0.461	0.910	Yes
	q	0.02	$\pm 1\%$	+0.569	-0.547	1.116	Yes

Conclusion. Both r and q are material. No valuation controls have been documented for these parameters. See §5.

4.10 Stress and Boundary Tests

Objective. Verify model behaviour under extreme inputs: near barrier, deep ITM, near-zero vol, high vol.

Methodology. Tape replay vs fresh pricing at each stress point.

Pass criteria. Prices ≥ 0 . Tape and fresh agree within 0.1% (the same tolerance as the implementation-accuracy test). A larger divergence indicates a frozen branch and is reported as a finding.

	Scenario	Tape	Fresh	Error	Flipped sites	Status
Results.	S=81	0.4690	0.4690	4.97e-04%	3	OK
	S=80.01	0.0047	0.0047	1.36e-01%	3	FAIL
	S=150	51.4862	51.4864	4.06e-04%	3	OK
	vol=0.01	1.9492	1.9492	1.02e-06%	7	OK
	vol=1.0	22.6503	18.0832	2.53e+01%	4	FAIL
	r=0	8.5472	8.5472	0.00e+00%	1	OK

Conclusion. 4 of 6 scenarios pass. At S=80.01 the tape returns 0.004699 against a fresh price of 0.004705 (0.136%): branches frozen at recording time flip there ([errorfunction.cpp:209](#), [errorfunction.cpp:207](#), [errorfunction.cpp:221](#)), so the tape is not valid at this input. At vol=1.0 the tape returns 22.650255 against a fresh price of 18.083191 (25.3%): branches frozen at recording time flip there ([errorfunction.cpp:209](#), [errorfunction.cpp:207](#), [errorfunction.cpp:225](#), [errorfunction.cpp:221](#)), so the tape is not valid at this input.

4.11 Back-testing

Back-testing has not been performed. P&L data was not provided for this validation. This is noted in Scope and Limitations (§2.7).

5 Findings

#	Severity	Finding	Recommendation
---	----------	---------	----------------

F1	Material	Feeding model validation reports not available	Validate the yield curve, dividend and vol surface models independently.
F2	Low	No valuation control documented for r	Impact 0.910 > bid/ask 0.10: document IPV, reserve or calibration.
F3	Low	No valuation control documented for q	Impact 1.116 > bid/ask 0.10: document IPV, reserve or calibration.
F4	Medium	Tape replay invalid at $S=80.01$: tape 0.00469898 vs fresh 0.00470538 (0.136%)	Branches frozen at recording time flip here: errorfunction.cpp:209 , errorfunction.cpp:207 , errorfunction.cpp:221 . Restrict replay to the validated range, or re-record when inputs leave it.
F5	Medium	Tape replay invalid at $vol=1.0$: tape 22.6503 vs fresh 18.0832 (25.3%)	Branches frozen at recording time flip here: errorfunction.cpp:209 , errorfunction.cpp:207 , errorfunction.cpp:225 , errorfunction.cpp:221 . Restrict replay to the validated range, or re-record when inputs leave it.
F6	Low	Binomial LR 200 is 0.5084% from analytic, outside the 0.5% benchmark tolerance	Benchmark engine only. Use a higher resolution or exclude it from barrier benchmarks.
F7	Low	JR convergence not monotonic (error rises 25→50, 100→200)	Known barrier/node alignment effect. Use at least the production resolution; consider barrier-aligned trees.
F8	Medium	Tape replay invalid at low volatility: $\sigma = 5\%$ (1.23%), $\sigma = 10\%$ (0.14%)	Branches frozen at the recording volatility flip (Section 4.6). Re-record when σ leaves the validated range.

6 Validation Outcome

Decision: Conditionally Approved.

Per-engine tape analysis:

- **Analytic** (473 entries): Merton–Reiner–Rubinstein formula verified from tape ((1)–(6)).
- **FD** (479,544 entries): Douglas scheme with tridiagonal solve, absorbing boundary at B ((7)–(10)).
- **Binomial** (441,136 entries, CRR): CRR/JR/LR trees with barrier adjustment ((15)–(18)).

Test results:

- Cross-engine benchmark: pass for 3 of 4 benchmark engines.
- Implementation accuracy: pass.
- Sensitivity ladders: spot pass, volatility fail, rate pass.
- Convergence: CRR pass, JR accuracy only, FD pass.
- Barrier parity: pass.
- Stress: fail at $S=80.01$, $vol=1.0$.

Conditions for full approval:

1. Feeding model validation reports not available — Finding F1.
2. Tape replay invalid at S=80.01: tape 0.00469898 vs fresh 0.00470538 (0.136%) — Finding F4.
3. Tape replay invalid at vol=1.0: tape 22.6503 vs fresh 18.0832 (25.3%) — Finding F5.
4. Tape replay invalid at low volatility: $\sigma = 5\%$ (1.23%), $\sigma = 10\%$ (0.14%) — Finding F8.

A FD Engine: Supplementary Stencil Details

Non-uniform mesh spacing. Let $h_i = x_i - x_{i-1}$ denote the local spacing (uniform in log-spot: $h_i = h$ for all i). The first derivative stencil (23) reduces to the central difference:

$$\left. \frac{\partial V}{\partial x} \right|_i \approx \frac{V_{i+1} - V_{i-1}}{2h} \quad (23)$$

and the second derivative (24) to:

$$\left. \frac{\partial^2 V}{\partial x^2} \right|_i \approx \frac{V_{i+1} - 2V_i + V_{i-1}}{h^2} \quad (24)$$

Tridiagonal band assembly. The `axpyb` function (70,700 entries) computes $L = a_1 D^{(1)} + a_2 D^{(2)} + b I$ where $a_1 = s = (r - q) - \frac{1}{2}\sigma^2$, $a_2 = \frac{1}{2}\sigma^2$, $b = -r$. The resulting tridiagonal bands for the uniform mesh are:

$$l_i = \frac{\sigma^2}{2h^2} - \frac{s}{2h}, \quad d_i = -\frac{\sigma^2}{h^2} - r, \quad u_i = \frac{\sigma^2}{2h^2} + \frac{s}{2h} \quad (25)$$

Thomas algorithm detail. The forward sweep (11)–(12) eliminates the lower band; the back substitution (13) recovers the solution. Total cost: $O(N)$ per time step. The tape records 150,591 entries for `solve_splitting` across all time steps (default 100 time steps $\times$ 100 grid points, about 15 operations per node and step).

Apply operation. The `apply` function (101,000 entries) computes the matrix-vector product $y = L V$:

$$y_i = l_i V_{i-1} + d_i V_i + u_i V_{i+1} \quad (26)$$

used on the right-hand side of the Douglas step (10).