QUANTIL

Bid Price Optimizer

Validation note

How the calculations were tested, and with what result.

Validation note | for tool version 1.2.2 | 2026-10-27
Quantil is a trading name of Bouman Management B.V. | Barendrecht, the Netherlands | Chamber of Commerce (KvK) 98301764

Contents

1. Purpose and summary 3

2. Approach 4

2.1 Three independent implementations 4
2.2 Test environment and reference case 4

3. Building blocks 5

3.1 Random generator 5
3.2 Normal distribution functions 5
3.3 Beta-PERT sampling 5
3.4 Correlation (Gaussian copula) 5
3.5 Analytic approximation 6

4. Win model 7

4.1 Formula against brute-force simulation 7
4.2 Accuracy of the market-factor integration 7

5. End-to-end tests 8

5.1 Workbook against the Python reference 8
5.2 VBA engine 8
5.3 Formula and error scan 8

6. Statistical stability 9

6.1 Twenty seeds 9
6.2 Number of iterations 9

7. Edge cases and controls 10

7.1 Report generator 10

8. Conclusion 11

1. Purpose and summary

This note documents how the calculations of the Quantil Bid Price Optimizer, version 1.2.2, were verified, and with what result. It complements the Methodology note, which describes the method. The aim is that an estimator, controller, auditor or client can judge for themselves whether the numbers can be relied on.

Every building block was tested against an independent reference (exact library functions, brute-force simulation or high-precision integration), and the complete workbook was tested end to end against a separate Python implementation of the same model. The results are summarized below and detailed in the following chapters.

Test	Criterion	Result	Status
Random generator (1,000,000 draws)	Uniform, no serial correlation	KS p = 0.27; lag-1 r = -0.0012	**Pass**
Normal distribution functions	Error < 10^{-8}	CDF 2.2 x 10^{-16}; inverse 5.6 x 10^{-9}	**Pass**
Beta-PERT sampling table	Quantile error < 0.01% of range	Max 0.0015% of range	**Pass**
Copula correlation (three presets)	Within 0.01 of target	Max deviation 0.003	**Pass**
Win probability vs brute-force simulation	Within 3 standard errors	Max 1.1 standard errors	**Pass**
Market-factor integration (9 points)	Error < 0.1 point	< 0.1 point up to co-movement 0.6	Pass, range limited
Workbook formulas vs Python (67 prices)	Identical to rounding	P(win) 3.0 x 10^{-15}; E[profit] < EUR 0.01	**Pass**
VBA simulation vs Python	Identical to rounding	Max difference EUR 0.0001 per run	**Pass**
Recommended price, independent optimizer	Identical	EUR 301,864,979.50 in both	**Pass**
Stability over 20 seeds	B* within the band	B* EUR 301.86 M to EUR 302.07 M	**Pass**
Edge cases and messages	Correct flag or message	8 of 8 scenarios	**Pass**
Report generator texts	Correct conclusions per scenario	3 of 3 scenarios	**Pass**

Table 1. Summary of the validation results.

> **Two findings and the action taken**
>
> **Market co-movement above 0.6.** The 9-point integration over the common market factor loses accuracy for very strong co-movement (error 0.3 point at 0.7, 3.6 points at 0.9). The input range was limited to 0 to 0.6 in version 1.2.1, where the error is below 0.1 point. The reference case uses 0.30.
>
> **Analytic approximation.** Without a current simulation the normal approximation underestimates the spread when risks are linked to cost categories (about 11% lower standard deviation in the reference case). This was already documented; the status line and the Validate button make clear when the approximation is in use.

2. Approach

2.1 Three independent implementations

The model exists three times, written separately and compared with each other:

- **Workbook formulas.** Win probability, optimization, decision rules, scenarios, sensitivity and cash flow are Excel formulas.
- **VBA engine.** The Monte Carlo cost simulation (modules modQuantilRandom and modBPO_Engine).
- **Python reference engine.** A separate implementation of the cost simulation, the win model and the optimizer, used only for testing. It uses the same random generator and the same numerical constants so that results can be compared run by run.

Building blocks were additionally tested against independent references that share no code with the tool: the SciPy library (normal and Beta distributions, adaptive integration) and brute-force simulation of competitor bids with NumPy.

2.2 Test environment and reference case

Item	Version
Workbook	Quantil Bid Price Optimizer 1.2.2, English edition, reference case Kestrow Link (fictitious)
Microsoft Excel	Microsoft 365 for Windows (simulation, buttons, charts, Word and PowerPoint generator)
LibreOffice	24.2 (headless recalculation of all formulas, VBA engine in compatibility mode)
Python	3.13 with NumPy 2.5 and SciPy 1.18
Settings	Seed 42, 10,000 iterations, correlation Medium, market co-movement 0.30, P80

Table 2. Test environment.

3. Building blocks

3.1 Random generator

MRG32k3a implements the recurrence and constants of L'Ecuyer (1999). One million consecutive draws from seed 42 were tested:

Statistic	Expected	Observed
Mean	0.5	0.49976
Variance	0.08333	0.08348
Kolmogorov-Smirnov test (uniform)	p > 0.05	p = 0.267
Chi-square test, 100 classes	p > 0.05	p = 0.637
Lag-1 autocorrelation	0 (standard error 0.001)	-0.0012
Smallest and largest value	strictly between 0 and 1	2.6 x 10^{-7} and 0.9999998

Table 3. Tests of the random generator.

Because a draw is never exactly 0 or 1, the inverse normal function is never called with an invalid argument. The VBA and Python generators produce identical sequences, so every simulation run can be reproduced exactly.

3.2 Normal distribution functions

The normal CDF (West, 2005) was compared with SciPy at 16,001 points between -8 and 8, and the inverse (Acklam, 2003) at 24,000 probabilities between 10^{-9} and $1 - 10^{-9}$. The largest absolute error is 2.2 x 10^{-16} for the CDF and 5.6 x 10^{-9} for the inverse (relative 1.1 x 10^{-9}). Both are many orders of magnitude below any effect on the results.

3.3 Beta-PERT sampling

The 201-point quantile table was compared with the exact Beta quantile function from SciPy, and 200,000 samples per line were compared with the exact mean and standard deviation:

Three-point estimate (EUR M)	Table error (share of range)	Mean deviation	Std dev deviation
9.0 / 9.8 / 11.5	0.0002%	+0.01%	+0.58%
26.0 / 28.5 / 32.0	0.0001%	+0.01%	+0.47%
2.8 / 3.1 / 3.9	0.0001%	+0.02%	+0.67%
0.5 / 1.0 / 4.0	0.0015%	+0.16%	+1.09%

Table 4. Accuracy of the PERT sampling (200,000 samples per line).

The table itself is accurate to about one millionth of the range. The linear interpolation between the table points makes single lines slightly wider than the exact distribution (standard deviation +0.3% to +0.9%, of which about 0.15% is sampling noise); the mean is unaffected within sampling error. The effect on the total cost is small and conservative: it never understates the uncertainty.

3.4 Correlation (Gaussian copula)

For each preset, 50,000 sets of latent normal scores were generated with the engine's factor structure and the achieved correlations were measured:

Preset	Same category: target	Achieved	Different categories: target	Achieved
Low	0.30	0.298	0.09	0.093

Preset	Same category: target	Achieved	Different categories: target	Achieved
Medium	0.50	0.497	0.20	0.201
High	0.70	0.698	0.42	0.420

Table 5. Target and achieved correlation.

The copula changes the dependence but not the distribution of each line. With 20,000 iterations the mean direct cost is EUR 241.04 M without and EUR 241.19 M with the High preset, against an exact value of EUR 241.03 M (both within sampling error), while the standard deviation of the direct cost rises from EUR 3.48 M to EUR 10.01 M. This is the intended effect: correlation widens the total, it does not move it.

With the preset None the engine reproduces the version 1.0 engine exactly (largest difference over 5,000 runs: EUR 0.00).

3.5 Analytic approximation

With independent cost lines (preset None) the normal approximation was compared with a 50,000-run simulation:

Statistic	Analytic	Simulation	Difference
Mean	EUR 281.98 M	EUR 282.02 M	EUR -0.03 M
Standard deviation	EUR 6.25 M	EUR 6.25 M	EUR -0.00 M
P80	EUR 287.24 M	EUR 287.17 M	EUR 0.07 M
P95	EUR 292.26 M	EUR 292.76 M	EUR -0.51 M

Table 6. Analytic approximation against simulation (no correlation).

Without correlation the approximation is close; it understates the far tail slightly because the risks are skewed. With the Medium preset and risks linked to cost categories the approximation gives a standard deviation of about EUR 10.1 M against EUR 11.4 M in the simulation, and therefore a lower probability of a loss if won. This confirms the guidance in the Methodology note: use the approximation for exploring and the simulation for the decision.

4. Win model

4.1 Formula against brute-force simulation

For the five competitors of the reference case, 400,000 tenders were simulated per price: a common market factor, each competitor's participation and its log-normal bid were drawn, and the share of tenders in which every bidding competitor was more expensive was counted. The win-factor shift was then applied in the same way as in the tool.

Price	Formula (tool)	Simulation	Standard error	Difference / SE
EUR 280.00 M	81.70%	81.74%	0.06%	0.5
EUR 290.00 M	60.47%	60.48%	0.08%	0.1
EUR 301.86 M	30.01%	29.98%	0.07%	0.4
EUR 310.00 M	14.00%	13.95%	0.05%	1.1
EUR 320.00 M	3.88%	3.90%	0.03%	0.7

Table 7. Probability of winning: formula and brute-force simulation (market co-movement 0.30).

All differences are within the statistical noise of the simulation. Two further variants were tested in the same way:

- **MEAT award** (quality bonus 10% of the reference estimate, our score 7.5, competitor scores 5.5 to 8.0): the workbook gives 33.72% at its recommended price, the simulation 33.78% (standard error 0.06 point).
- **Generic bidders** (three additional unknown bidders): formula 42.60% and 16.13%, simulation 42.61% and 16.15% at EUR 290 M and EUR 302 M.

4.2 Accuracy of the market-factor integration

The 9-point Gauss-Hermite rule was compared with adaptive integration (SciPy, tolerance 10^{-13}) over 81 prices on the full grid:

Market co-movement	0.1	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Largest error (points of P(win))	< 0.0001	0.0004	0.003	0.02	0.08	0.31	1.09	3.60

Table 8. Integration error of the 9-point rule.

Up to 0.6 the error stays below 0.1 point of probability, which is far below the uncertainty of the market inputs themselves. Above 0.6 the conditional win curve becomes so steep that nine points are not enough. **The input is therefore limited to 0 to 0.6 from version 1.2.1.** A co-movement above 0.6 would mean that competitors hardly differ from each other beyond the market as a whole, which is rarely realistic.

5. End-to-end tests

5.1 Workbook against the Python reference

The complete workbook was recalculated in LibreOffice and compared with the Python reference engine on all 67 evaluated prices (41 grid points, 21 fine-grid points and the strategy prices):

Quantity	Largest difference
Simulated total cost per run (10,000 runs)	EUR 0.005 (rounding to cents when written)
P(win)	3.0×10^{-15}
Expected profit	EUR 0.000003
P(loss if won)	0 (identical)
Recommended price, band	Identical: EUR 301,864,979.50; band EUR 299.46 M to EUR 303.07 M

Table 9. Workbook against the Python reference.

5.2 VBA engine

The VBA simulation was run in LibreOffice (compatibility mode) and in Microsoft Excel for Windows and compared run by run with the Python engine for the same seed. The largest difference was EUR 0.00004 (LibreOffice) and EUR 0.00008 (Excel) on totals of about EUR 290 M, i.e. floating-point rounding. The sequence of random numbers, the copula and the PERT tables are therefore identical in all three implementations.

5.3 Formula and error scan

After every build all sheets are recalculated and scanned for error values. The only error values present are the intentional #N/A values that keep chart series empty (price curve below break-even, histogram helper series and the peak marker of the cash curve).

6. Statistical stability

6.1 Twenty seeds

The simulation was repeated with seeds 1 to 20 (10,000 iterations each) and the full optimization was run for each:

Result	Seed 42	Lowest	Highest	Std dev over seeds
Recommended price B*	EUR 301.86 M	EUR 301.86 M	EUR 302.07 M	EUR 0.09 M
P(win) at B*	30.0%	29.5%	30.0%	0.22 pt
Expected profit	EUR 2.49 M	EUR 2.43 M	EUR 2.56 M	EUR 0.04 M
P(loss if won)	14.3%	13.3%	14.6%	0.28 pt
Cost at P80	EUR 291.27 M	EUR 290.97 M	EUR 291.64 M	EUR 0.17 M
Expected cost E[C]	EUR 282.03 M	EUR 281.79 M	EUR 282.24 M	EUR 0.12 M

Table 10. Results over 20 seeds (10,000 iterations each).

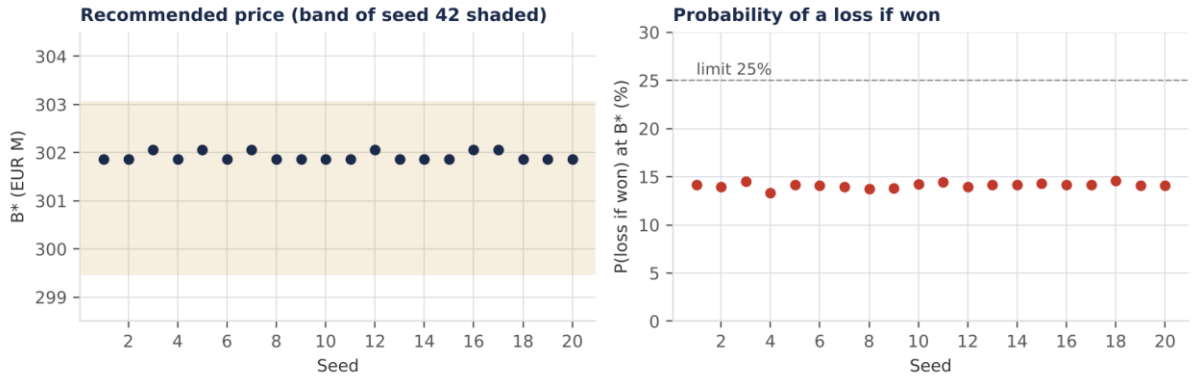


Figure 1. Recommended price and probability of a loss if won for seeds 1 to 20.

Every seed gives a recommended price inside the band of the reference run, and the decision rules lead to the same verdict in all twenty runs. The variation of B* (about EUR 0.1 M) is mainly the step size of the fine grid (EUR 0.2 M).

6.2 Number of iterations

Iterations	Std dev of B*	Std dev of P80 cost	Std dev of P(loss if won)	Std dev of E[C]
2,000	EUR 0.14 M	EUR 0.45 M	0.65 pt	EUR 0.23 M
5,000	EUR 0.11 M	EUR 0.11 M	0.33 pt	EUR 0.08 M
10,000	EUR 0.10 M	EUR 0.16 M	0.33 pt	EUR 0.12 M

Table 11. Spread of the results over 10 seeds per number of iterations.

With 2,000 iterations the probability of a loss if won varies by about 0.65 point between seeds; with 10,000 by about 0.33 point. This supports the guidance: 2,000 iterations to explore, 10,000 for the final decision. The convergence verdict on 10_SIMULATION uses the same statistics for the case at hand.

7. Edge cases and controls

The following scenarios were built from the reference workbook, recalculated and checked for the correct behavior:

Scenario	Expected behavior	Observed
Cost line with most likely above maximum	Line flagged; Validate stops on it	'Check order' on the line; Validate jumps to it
Custom correlation matrix not positive definite	Matrix flagged; no simulation with this matrix	'Not valid'; status switches to the analytic approximation; the engine stops with a message
Cost input changed after the simulation	Results not silently based on old simulation	Status 'Out of date'; analytic approximation in use until the next run
Minimum margin 12% (no feasible price)	Clear message, verdict review	'No price on the grid meets all decision rules'; verdict BID WITH REVIEW
MEAT award	Quality value applied to all bidders	Our quality value EUR 20.0 M; P(win) 33.7%; matches simulation
Chosen price below break-even (EUR 286.0 M)	NO-BID	NO-BID: expected profit is negative
Chosen price far above optimum (EUR 318.0 M)	Review: P(win) below minimum	BID WITH REVIEW: 1 rule not met
Modules of another version	Stop with message	Message 'modules do not match this workbook'; no calculation

Table 12. Edge cases.

7.1 Report generator

The text engine of the Word and PowerPoint generator was run on the three price situations above (recommended price, a price below break-even and a price well above the optimum) and the conclusions and recommendations were checked against the results:

Situation	Key generated text
Recommended price EUR 301.9 M	Submit at EUR 301.9 M; win probability above the neutral share of 20%; loss risk well within the limit; firm up the largest cost lines (direct cost is the largest driver).
Chosen EUR 286.0 M	Do not submit at the current price; deviation of -5.3% costs EUR 5.91 M expected profit; loss probability 58% exceeds the limit; high chance of winning is bought with margin.
Chosen EUR 318.0 M	Submit only with an explicit decision on the rule not met; win probability 5% below the neutral share; competitor price level is the largest driver; invest in market intelligence.

Table 13. Report generator scenarios.

The complete Word report was additionally generated in Microsoft Word from the reference case and checked page by page (structure, figures, tables, numbering and contents).

8. Conclusion

The building blocks of the Quantil Bid Price Optimizer are numerically accurate, the three implementations agree to floating-point precision, and the results are stable across random seeds. The tests found one limitation that matters in practice, the accuracy of the market-factor integration above a co-movement of 0.6, and the input range was limited accordingly in version 1.2.2. The known limitations of the method itself (expert inputs, simplified award formulas, the analytic approximation) are described in the Methodology note.

The validation will be repeated for every release that changes a calculation. The test scripts and results are kept by the publisher and can be made available to clients on request.

All names and figures in the reference case are fictitious.