Compute Market

Model choices and grounding

Rationale and reference map

Contents

Design principle

One stochastic curve system supplies the market state. Contractual settlement turns that state into cash; operating and financing constraints determine which decisions are feasible. Shared factors connect these calculations without assigning the same value to a traded claim, a physical promise and collateral.

The choices below state the construction, its limits and the code that implements it. References establish the mathematical precedents; the compute application and synthetic inputs belong to this project.

Market representation

1. Model delivery dates jointly

Delivery curves make tenor exposure explicit and give strips, options and equipment appraisal a common input. Calendar delivery and time to delivery remain distinct: Musiela transport describes how a curve rolls through tenor coordinates. A structural spot model can also imply a curve, and may be preferable when supply and demand identify its dynamics. Here the curve is the statistical primitive, while operations constrain actual delivery.

HJM and commodity-curve representations provide the precedent [R15–R18]. `curves/model.py` defines the law; `curves/adapter.py` samples it into a finite control program.

2. Choose factor shapes that survive calendar roll

Exponential loadings $\ell_a(x) = e^{-Kx}c_a$ with nonnegative diagonal decay are closed under time shift. The finite state therefore represents the chosen curve family exactly, with fast conditional simulation and analytic sensitivities. Empirical PCA can discover useful modes, but a fitted span need not remain invariant as contracts age. Such a realization needs a projection rule and a measured closure error.

`CurveModel.loading`, `value`, `greeks` and `step` use the same shapes [R16]. Tests check calendar-date invariance and finite-difference Greeks. Richer volatility or kernel specifications require additional identifying observations.

3. Respect each market's price support

Compute references are exponential affine; electricity is additive and can be negative. Clipping simulated prices changes expectations, option values and pricing identities. Shifted logarithms or mixtures are possible alternatives, with additional support parameters to identify.

`CurveSpec.kind` selects the form and its compensation. Negative-power-price tests and log-input validation enforce the distinction [R17].

4. Let a common shock reach prices and collateral

Correlated Brownian factors and shared marked jumps connect level, generation, region, service, power and financing. A long-end compute shock changes both rental margins and the appraisal supported by those margins. Separate resale and execution bases retain risks outside this relationship.

`defaults.py` supplies the loadings; `portfolio.py` maps the state to collateral and operations [R23]. Constant coefficients keep the reference experiment interpretable. Regime or stochastic-volatility models would add time-varying covariance; named stresses instead isolate particular mechanisms.

5. Enforce pricing drift through the cumulant

The log-price compensator integrates the complete Q cumulant, including compound-Poisson jumps. Each clean fixed-delivery futures reference is then a Q-martingale. Removing jump compensation or imposing storage carry would change that law.

`cumulant` and `_compensator` are shared by simulation, claims and observations [R15–R17, R19]. Independent Monte Carlo checks the martingale identity. A general HJM SPDE could impose the restriction directly; the affine family integrates it with deterministic quadrature.

6. Separate forecasting from pricing

P and Q share the factor state, loadings, diffusion covariance and decay. Brownian drift and jump intensity or mark distribution may differ, subject to compatible support. This preserves risk premia while keeping forecasts and financial marks in the same coordinates.

`CurveModel` checks drift span and jump support; `fit_physical_drift` changes only P parameters [R17, R19]. More general affine measure changes can alter mean reversion. The reference study fixes those structural coefficients when fitting physical drift; hidden-state inference is tested separately.

Settlement, claims and observations

7. Discount on the same state

A shifted affine short rate fits the initial discount curve and shares shocks with compute. Joint sampling of factor endpoints and integrated rates retains compute–rate covariance, which separates terminal forwards from continuously margined futures.

`discount` implements the bond transform and `forward` the convexity adjustment [R19]. Discounted Monte Carlo checks both. Deterministic rates are a tested limit; commercial funding spreads enter borrowing terms rather than the clean discount rate.

8. Aggregate the contract's actual deliveries

Quantities weight prices, not logarithms. Periodically settled strips use payment-date discounting and terminal-forward strikes. Splitting an interval and recombining identical rights must preserve value. A single later settlement or a useful-output unit requires a different operator.

`DeliveryStrip` and `continuous_strip` serve observations, fitting and valuation [R16, R18]. Quadrature splits at initial-curve knots; partition tests check the aggregation identity.

9. Normalize rights before fitting prices

Each observation records its instrument form, delivery dates, level, uncertainty and source. A physical quote requires an explicit rights adjustment before it can serve as a clean reference. Otherwise a price difference caused by cancellation or firm delivery becomes a spurious market factor.

`observations.py` implements the operators, Jacobians, curve fit and filter [R18, R21]. Joint estimation of contractual rights and market curves is possible with identifying observations. Regularization can stabilize a sparse fit but cannot supply missing data rank.

10. Match the method to the exercise right

Gaussian European claims use an expiry-forward formula. Jump claims use deterministic discounted-transform inversion, checked by independent Monte Carlo. Swing claims use a least-squares continuation indexed by remaining rights and evaluated on independent paths. An ex-post best exercise date is an information bound, not an admissible policy.

`claims.py` and `option_pricing.py` implement these methods [R19, R22, R23]. Tests cover parity, replay and exercise limits. Monte Carlo remains the independent check for payoffs outside the transform family.

Assets, debt and private value

11. Appraise the remaining life of the equipment

The appraisal integrates forward margins to a fixed calendar end of life, scales to the initial equipment mark and adds an independent resale basis. An aging cohort cannot acquire extra life merely because the valuation date advances. Operating receipts and terminal equipment value cover disjoint periods.

`hardware_marks` supplies the appraisal and its curve sensitivities [R06, R07]. Secondary-market observations can replace or inform its resale basis. The owner's operating-option value does not replace a lender's contractual appraisal.

12. Put debt on its payment dates

Borrowing limits, eligibility, coupons, principal and draw restrictions enter the cash ledger. Fixed-rate debt keeps its coupon. Selling pledged equipment raises cash while shrinking the borrowing base, so repair must solve both effects together.

`simulation._repair_one` checks the physical and financial breakpoints and returns the feasible side of a binding cash boundary. `curves/control.py` pays incoming obligations before new decisions and accrues interest on the actual post-action balance through the final interval [R08, R09].

13. Price a reservation as a change to the book

Compare optimized private value before and after the contract's obligations and payment enter the state. Payment can relax a cash constraint; delivery can consume capacity or offset an existing exposure. A common forecast therefore supports different reservation prices for different books.

`case.py` and `hedge_pricing.py` require the existing book and certify the numerical reservation payment [R07, R08]. A clean strip plus separately valued rights is useful when the effects separate. Financing interactions determine when that decomposition fails.

14. Measure scarcity where the constraint binds

Private capacity value is the change in operating value after adding an eligible block. Market scarcity moves reference prices for everyone; fragmentation, debt maturity and contract concentration alter a particular operator's opportunity cost.

`capacity_shadow` revalues inventory, while placement and workload modules expose nonlinear physical constraints [R05]. A market-equilibrium model could feed aggregate scarcity back into price formation. Valuing a given book does not require solving that equilibrium.

Hedging and control

15. Report the span and the residual

Project portfolio deltas onto instrument deltas in the conditional covariance metric. Whitening and a pseudoinverse handle units and redundant instruments; rank, residual variance and orthogonality reveal what is actually covered.

`risk.project` and `factor_attribution` use the curve engine's coordinates [R20, R23]. Full revaluation handles large moves, options and funding constraints. Complete delta rank still leaves nonlinear jump risk, as a separate test demonstrates.

16. State whether the mandate rewards trading alpha

The default hedge objective centers forecast gains, charges adverse expected carry, pays execution costs and prices margin's cash opportunity cost. The ledger records full realized settlements, premiums and distributions. Centering belongs to the risk objective, not to spendable cash.

`risk_execution.py` also supports centered-risk and total-return mandates. The latter restores signed expected carry. This is a business preference, not a pricing theorem; directions, limits, turnover, depth and charges remain declared inputs.

17. Give operations and hedges one cash budget

After incoming settlement and mandatory repair, one conditional LP chooses current dispatch, borrowing, repayment and financial positions. Sequential discretionary decisions could each consume liquidity needed by the other.

`continuation.py` fits a specified future operating plan on independent paths; `control.py` optimizes current choices against it [R11, R22]. A full multistage Bellman solution would optimize future operating choices as well. The finite-control backend provides a separate sampled-program comparison. Cash reconstruction and failed paths remain part of evaluation.

18. Define risk on a stated decision grid

Finite programs use nested mean/expected shortfall. The path controller applies conditional expected shortfall to its continuation problem; reporting evaluates terminal loss and financing outcomes separately. Changing decision frequency while repeating a fixed tail penalty changes the preference.

`tails.py` reports atom mass and worst-child saturation; fixed-horizon experiments vary branching and seed [R01–R04, R10]. Expected utility, entropic risk or constrained return are alternative mandates, not changes to the clean pricing curve.

Numerical and empirical discipline

19. Separate transition error from policy error

Sample OU endpoints and integrated factors jointly, including explicit Poisson times and marks. The constant-coefficient law avoids Euler step bias; smooth integrals use deterministic quadrature. Continuation error, finite-support error and Monte Carlo error remain separate.

`model.step` generates both paths and conditional innovations [R10, R19]. Nonlinear coefficients may require time discretization. Larger trees alone cannot repair a poor continuation fit or weakly resolved conditional tail.

20. Save the objects that define a result

A result needs its full case, existing book, fitted P law, continuation, tape, policy outputs and solver certificates. A seed cannot identify changed source, data or dependency builds.

`curves replay` compares every recorded numeric output field without refitting [R13, R14]. Dependency records, source identities and independent accounting checks accompany the retained experiments. The release manifest checks integrity, not publisher identity.

21. Stress mechanisms without inventing event probabilities

Interventions separately change curve factors, availability, local/index basis, collateral eligibility, margin or market depth. Paired policies share the same underlying tape. A price shock reaches appraisal through the common state; an execution freeze acts directly on market access.

`stress_economy` records the intervention and start date [R09]. Failures, shortfalls and cash outcomes stay in the sample. An estimated regime model could assign occurrence probabilities with adequate data; these named experiments measure conditional consequences.

Risk, operating options and estimation

22. Keep delivery buckets alongside factors

Calendar-delivery tents express exposure in the dates a commercial desk trades. Log bumps preserve positive compute prices; additive bumps retain electricity and rate units. The bucket-to-factor Jacobian links this detail to the smaller factor state and exposes interpolation error.

`buckets.py` computes delta, gamma, half-bump diagnostics and off-span shocks [R23, R25]. Node-supported claims check the analytic bridge; reoptimized-book callbacks capture policy response. Dense buckets alone hide covariance, while factors alone omit possible curve shapes.

23. Keep operating value, appraisal and exit proceeds separate

Economic value comes from an operating/idle/retire policy. Lender value follows the specified appraisal and update schedule. Liquidation proceeds depend on the quantity the market will absorb. Summing these values would count the same equipment repeatedly.

`fleet_options.py` includes startup, shutdown, standby and minimum-run constraints [R22, R24]. Independent evaluation values the trained policy; a pathwise perfect-information program supplies an upper comparison on the same grid. A forward-margin annuity remains a useful appraisal proxy, with different rights from that operating policy.

24. Preserve the instrument's cash mechanics

The registry separates ex-distribution marks, cash events, ownership, maturity, execution, collateral and recovery. A paid long put consumes premium at purchase. Futures pay variation settlement. A transferable forward retains its strike, and each coupon leaves the mark exactly once.

`instruments.py`, `option_pricing.py`, `settlement.py` and `registry.py` share these conventions across pricing and control [R08, R19]. Tests check fixed strikes, distributions and blocked reopening after default. A terminal payoff matrix is sufficient only when interim trading and funding do not matter.

25. Model liquidity as access to execution

A mean-reverting jump state combines idiosyncratic variation with funding and downside factors. It drives spreads, turnover depth and collateral multipliers, permitting ordinary funding stress to coincide with expensive hedging. Position limits, turnover caps and impact thresholds remain separate inputs.

`liquidity.py` maps the observed state to bounded multipliers and contributes to supplier default hazard [R09]. It has no futures-martingale restriction. Instrument-specific order-book models are an alternative with adequate data; the supplied coefficients are synthetic assumptions.

26. Expose calibration anchors and identification rank

Declare free coordinates, fixed scale/sign/loading anchors, bounds and priors. Fit market repricing errors alongside physical conditional characteristic-function moments. A separate Gaussian quasi-likelihood handles latent quote histories. Identification comes from the data Jacobian before adding prior penalties.

`calibration.py` preserves positive-semidefinite covariance and reports rank, bounds, convergence and source hashes [R19, R21, R26]. A regularized point estimate cannot resolve observationally equivalent factor rotations or risk-premium directions. The synthetic recovery fixes nuisance coordinates and holds out option expiries; the latent example retains a full one-coordinate objective profile. An efficient joint likelihood would require a richer jump-density/filter approximation.

27. Separate process variation from parameter uncertainty

Resample whole independent synthetic histories, perturb quotes at their measurement uncertainty and refit each draw. Report failed fits. Propagate successful models to separate within-model process variation from between-model variation using total variance.

`bootstrap_calibration` and `predictive_decomposition` implement this design. More Monte Carlo paths refine conditional outcomes without identifying uncertain parameters. A single dependent market history needs block resampling or an appropriate parametric observation model; profile likelihood and posterior inference are other options. The reported bootstrap is a declared design, not a finite-sample coverage theorem.

Reading the evidence

Pricing identities establish consistency within the chosen curve family. Recovery experiments test estimation under synthetic observations. Held-out comparisons evaluate a frozen policy under the supplied law. Named stresses measure consequences of specified interventions. The research report keeps those claims separate; the data contract identifies the observations needed for a live book.

Bibliography

[R01] Rockafellar, R. T.; Uryasev, S. (2000)

Optimization of conditional value-at-risk. Journal of Risk, 2(3), 21–41.

Primary source.

Use in this project. Expected-shortfall epigraph and scenario optimization.

[R02] Ruszczyński, A. (2010)

Risk-averse dynamic programming for Markov decision processes. Mathematical Programming, 125, 235–261.

Primary source.

Use in this project. Nested conditional risk and dynamic programming; read with R03.

[R03] Ruszczyński, A. (2014)

Erratum to: Risk-averse dynamic programming for Markov decision processes. Mathematical Programming, 145, 601–604.

Primary source.

Use in this project. Correction to the original general-state result; finite trees avoid measurable-selection issues.

[R04] Iyengar, G. N. (2005)

Robust Dynamic Programming. Mathematics of Operations Research, 30(2), 257–280.

Primary source.

Use in this project. Rectangular transition ambiguity and robust recursion.

[R05] Gallego, G.; van Ryzin, G. (1994)

Optimal Dynamic Pricing of Inventories with Stochastic Demand over Finite Horizons. Management Science, 40(8), 999–1020.

Primary source.

Use in this project. Price-dependent arrivals and the opportunity cost of perishable inventory.

[R06] Carmona, R.; Coulon, M.; Schwarz, D. (2013)

Electricity price modeling and asset valuation: a multi-fuel structural approach. Mathematics and Financial Economics, 7(2), 167–202.

Primary source.

Use in this project. Structural interaction among output prices, demand and input costs. Author manuscript: https://arxiv.org/abs/1205.2299.

[R07] Callegaro, G.; Campi, L.; Giusto, V.; Vargiolu, T. (2016)

Utility indifference pricing and hedging for structured contracts in energy markets. Author manuscript, arXiv:1407.7725, revised 20 February 2016.

Primary source.

Use in this project. Portfolio-dependent prices and exercise/hedge interactions in incomplete energy markets. Its exponential utility differs from this project's mean/ES preference.

[R08] Brigo, D.; Francischello, M.; Pallavicini, A. (2015)

Invariance, existence and uniqueness of solutions of nonlinear valuation PDEs and FBSDEs inclusive of credit risk, collateral and funding costs. Author manuscript, arXiv:1506.00686, revised 27 November 2015.

Primary source.

Use in this project. Cash-flow-based treatment of collateral, credit, treasury funding and overlapping valuation adjustments.

[R09] Brunnermeier, M. K.; Pedersen, L. H. (2009)

Market Liquidity and Funding Liquidity. Review of Financial Studies, 22(6), 2201–2238.

Primary source.

Use in this project. Interaction between funding constraints, margin and market liquidity.

[R10] Kleywegt, A. J.; Shapiro, A.; Homem-de-Mello, T. (2002)

The Sample Average Approximation Method for Stochastic Discrete Optimization. SIAM Journal on Optimization, 12(2), 479–502.

Primary source.

Use in this project. Repeated sampling and optimization-error assessment. Author institutional record: https://pure.uai.cl/en/publications/the-sample-average-approximation-method-for-stochastic-discrete-o/.

[R11] Rawlings, J. B.; Mayne, D. Q.; Diehl, M. M. (2026)

Model Predictive Control: Theory, Computation, and Design. Second edition, sixth printing; Nob Hill Publishing, author resource page.

Primary source.

Use in this project. Receding-horizon control and the distinction between a local controller and a full value function.

[R12] MLCommons (2026)

MLPerf Inference: Datacenter. Benchmark specification and results portal; accessed 18 September 2026.

Primary source.

Use in this project. Workload-specific quality, latency and throughput conventions for useful-work normalization.

[R13] SciPy contributors (2026)

scipy.optimize.milp. SciPy 1.17.0 reference manual; accessed 18 September 2026.

Primary source.

Use in this project. MILP status, dual bound, node count and relative-gap fields exposed by the executed solver interface.

[R14] SciPy contributors (2026)

scipy.optimize.linprog, method highs. SciPy 1.17.0 reference manual; accessed 18 September 2026.

Primary source.

Use in this project. LP tolerances, residuals and marginals used for numerical certificates.

[R15] Heath, D.; Jarrow, R.; Morton, A. (1992)

Bond Pricing and the Term Structure of Interest Rates: A New Methodology for Contingent Claims Valuation. Econometrica, 60(1), 77–105.

Primary source.

Use in this project. The forward curve as the primary stochastic object and drift consistency. The project uses commodity futures restrictions, not the interest-forward HJM drift copied unchanged.

[R16] Benth, F. E.; Krühner, P. (2014)

Representation of Infinite-Dimensional Forward Price Models in Commodity Markets. Communications in Mathematics and Statistics, 2, 47–106. DOI.

Primary source.

Use in this project. Function-valued commodity curves, Lévy drivers, OU realizations and delivery-period operators.

[R17] Benth, F. E.; Piccirilli, M.; Vargiolu, T. (2019)

Mean-reverting additive energy forward curves in a Heath–Jarrow–Morton framework. Mathematics and Financial Economics, 13, 543–577. DOI.

Primary source.

Use in this project. Shared additive multicommodity factors and the distinction between real-world dynamics and martingale pricing dynamics. The linked manuscript has the shorter title Additive energy forward curves in a Heath–Jarrow–Morton framework.

[R18] Benth, F. E.; Koekebakker, S. (2008)

Stochastic modeling of financial electricity contracts. Energy Economics, 30(3), 1116–1157. DOI.

Primary source.

Use in this project. Actual delivery periods and consistent aggregation of financial electricity contracts; transferred here to compute service strips.

[R19] Duffie, D.; Pan, J.; Singleton, K. (2000)

Transform Analysis and Asset Pricing for Affine Jump-Diffusions. Econometrica, 68(6), 1343–1376. DOI.

Primary source.

Use in this project. Affine transform methods. The project specializes to constant-coefficient OU factors with compound-Poisson Gaussian marks and implements its transforms directly.

[R20] Föllmer, H.; Schweizer, M. (1991)

Hedging of Contingent Claims under Incomplete Information. In Davis and Elliott (eds.), Applied Stochastic Analysis, Stochastics Monographs 5, 389–414.

Primary source.

Use in this project. Orthogonal residual risk and incomplete-market hedging. The project's covariance-weighted factor projection is a local diagnostic, distinct from a full dynamic martingale decomposition.

[R21] Kalman, R. E. (1960)

A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering, 82(1), 35–45. DOI.

Primary source.

Use in this project. State-space filtering and covariance recursion. Nonlinear observation linearization and jump moment matching are explicit implementation choices.

[R22] Longstaff, F. A.; Schwartz, E. S. (2001)

Valuing American Options by Simulation: A Simple Least-Squares Approach. Review of Financial Studies, 14(1), 113–147. DOI.

Primary source.

Use in this project. Regression-based continuation for exercise decisions. The project adds a finite remaining-rights state and evaluates the frozen exercise rule on independent paths.

[R23] Benth, F. E.; Krühner, P. (2015)

Derivatives pricing in energy markets: an infinite-dimensional approach. SIAM Journal on Financial Mathematics, 6(1), 825–869. DOI.

Primary source.

Use in this project. Options and Greeks as functionals of an entire forward curve, and cross-commodity covariance.

[R24] Aïd, R.; Campi, L.; Langrené, N.; Pham, H. (2014)

A Probabilistic Numerical Method for Optimal Multiple Switching Problems in High Dimension. SIAM Journal on Financial Mathematics, 5(1), 191–231. DOI.

Primary source.

Use in this project. Regression-based switching control. The linked author manuscript is titled A probabilistic numerical method for optimal multiple switching problem and application to investments in electricity generation. The project uses a finite operating grid, independent policy evaluation and a perfect-information comparison.

[R25] Benth, F. E.; Krühner, P. (2015)

Approximation of forward curve models in commodity markets with arbitrage-free finite dimensional models. Author manuscript, arXiv:1512.05983.

Primary source.

Use in this project. Separates a curve-valued architecture from its finite-dimensional realization. The project uses an exponential loading family exactly closed under delivery-date transport.

[R26] Feuerverger, A.; Mureika, R. A. (1977)

The Empirical Characteristic Function and Its Applications. Annals of Statistics, 5(1), 88–97. DOI.

Primary source.

Use in this project. Empirical characteristic-function estimation. The conditional OU–Lévy moment weights, identification diagnostics and path-cluster bootstrap are specified explicitly in the project.