CCR

M. Hosseini

August 2025

1 CDS Curves to Stochastic Intensities

We need to clear a couple of potential misunderstandings, discussed below.

1. In interest rate modeling:

   - **Zero-Coupon Bonds (ZCBs):** If the full term structure of zero-coupon bond prices $P(t, T)$ is available, the valuation of interest rate derivatives can be done without explicitly modeling the short-rate process $r(t)$.
   - **Short Rate Models:** When ZCBs are not directly observable across all maturities, or when future rate paths are required (e.g., for stress testing, VaR), we simulate $r(t)$ via models like Hull-White or CIR.

   In credit risk modeling:

   - **Market CDS Curves:** If CDS spreads are observable across maturities for a given counterparty or rating class, one can bootstrap implied default probabilities and use them to compute Expected Exposure (EE), Potential Future Exposure (PFE), and Credit Valuation Adjustment(CVA).
   - **CIR-Based Intensity Models:** When assessing **forward-looking risk profiles**, **rating migrations**, or simulating tail events, it is necessary to model the stochastic intensity λ_t dynamically. This produces generator matrices $Q(t)$ and time-evolving transition matrices based on survival probabilities $P(t, T)$.

   Notice, credit risk modeling is likened to interest rate modeling.

2. CIR parameters calibrated to **market CDS quotes** are typically risk-neutral ($\mathbb{Q}$-measure) and suitable for pricing or CVA calculation. For stress testing, and computing real exposure distributions (EE, PFE), we must operate under the **physical measure** $\mathbb{P}$.

   Therefore:

   - If CIR is calibrated to CDS curves then we are working under $\mathbb{Q}$-measure.
   - Otherwise a shift is needed to work under $\mathbb{P}$-measure.

1.1 Simulation Steps

In this simulation study, we model the exposure profile and credit risk of a simplified **payer interest rate swap** over a 5-year horizon. The swap is assumed to be at-the-money (ATM) at inception and is evaluated under stochastic credit migration dynamics.

Rather than explicitly modeling forward rates and swap legs, we adopt a **proxy exposure model**, where the mark-to-market (MtM) of the swap at time t is approximated as:

$$V^S(t) = \exp\left(-\frac{s_{\text{rating}(t)}}{10000} \cdot t\right)$$

Here, $s_{\text{rating}(t)}$ is the credit spread corresponding to the counterparty's simulated credit rating at time t. This formulation captures the degradation in valuation due to deteriorating creditworthiness and is consistent with how the market prices counterparty credit risk into discounting. The simulated exposure paths $V^S(t)$ are used to compute EE, PFE, and CVA profiles across multiple rating paths driven by a CIR-based stochastic intensity model.

Step 1: Simulate Credit Paths Using $Q(t)$ or $P(t, T)$

From the CIR intensity model, derive time-dependent generator matrices $Q(t)$ and compute the transition matrix:

$$P(t, T) = \mathbb{E}\left[\exp\left(\int_t^T Q(u)\,du\right)\right]$$

Simulate multiple rating paths for a counterparty over $[0, T]$ under the physical measure $\mathbb{P}$ using $P(t, T)$ where $P(.)$ is interpreted as probability of survival (credit risk modeling) and not ZCB (interest rate modeling).

Relationship Between Q_A, λ_t, and `rating_paths`

The simulation of credit rating paths is built from two components:

(a) A deterministic rating migration structure: the generator matrix Q_A.

(b) A stochastic intensity process: CIR-modeled λ_t under the physical measure $\mathbb{P}$.

Together, these drive the evolution of rating states in the simulation. The relationship unfolds through the following structured steps:

(c) **Start with a base generator matrix Q_A:** This matrix encodes the baseline instantaneous transition rates between credit states for entities currently rated A. Each row i defines the intensity of transitioning from state i to all others.

(d) **Generate paths of intensity λ_t via the CIR model:** We simulate a time series of credit intensities per path using the CIR process under the physical measure:

$$d\lambda_t = \kappa(\nu - \lambda_t)dt + \sigma\sqrt{\lambda_t}dW_t, \quad \lambda_0 > 0$$

This introduces randomness in transition likelihoods across time and across paths.

(e) **Form a time-stepped transition matrix P_t:** At each time step t, compute the scaled generator matrix:

$$\widetilde{Q}_t = Q_A \cdot \lambda_t \cdot \Delta t \cdot \text{scale\_factor}$$

Then exponentiate it to obtain a transition matrix:

$$P_t = \exp(\widetilde{Q}_t)$$

which is a valid stochastic matrix reflecting the time-dependent migration behavior under the current intensity.

(f) **Simulate rating evolution using P_t:** At each time step and for each path:

- Use the current rating r_t as the row index.
- Draw the next rating r_{t+1} using the row $P_t[r_t, :]$ as a multinomial probability vector.
- Apply absorbing state behavior: once a counterparty defaults (state D), it remains there.

The final result is a matrix `rating_paths` $\in \mathbb{N}^{N \times T}$ that contains the time series of credit ratings for each simulated counterparty under $\mathbb{P}$.

This layered process allows us to translate a fixed Q_A matrix into dynamic, forward-looking rating trajectories that incorporate both structural migration tendencies and stochastic variation, crucial for applications such as EE, PFE, and CVA.

Step 2: Assign Spread Curves per Rating Bucket

For each credit rating (AAA, AA, A, etc.), assign representative credit spreads $\{s_i(t)\}$ or hazard rates $\{\lambda_i(t)\}$ to define market valuation profiles.

These spread curves can be flat, interpolated, or bootstrapped from historical averages by rating class.

Step 3: Compute Exposure Paths $V^S(t)$

Front Office (Trader) vs Back Office (Risk Management)

> **Trader:** "What's the price of this swap today, given the current CDS spread curve for an A-rated counterparty?"
> **Risk Manager:** "What happens if the counterparty becomes BBB in 3 months Or CCC in 9 months? What's the exposure then? What's the PFE at 95% or 99% confidence?"

Both use the same pricing model, but in very different contexts: the trader uses *today's credit state* and therefore related curve from Bloomberg say, while the risk manager explores *possible future credit paths* using a stochastic generator matrix.

In a simulation framework, the risk engine attempts to replicate how the front office would price the derivative *in future scenarios*, each conditioned on a possible evolution of credit ratings. Each simulated path of ratings (e.g., A $\to$ BBB $\to$ CCC) drives a corresponding path of credit spreads, and thus the discounting or default expectations used in the swap valuation.

Therefore, the exposure path $V^S(t)$ is computed using the same pricing logic as the front office, but with inputs (e.g., credit state, spread curve, LGD) determined by the scenario-specific path, thus bridging the modeling gap between the front and back office.

For each simulated path of ratings:

$$V^S(t) = \mathrm{MtM}_{\mathrm{curve}=s_{\mathrm{rating}(t)}}(\mathrm{derivative})$$

Step 4: Compute Exposure Profiles

$$E(t) = \max(V^S(t), 0)$$

Step 5: Expected Exposure (EE)

$$\mathrm{EE}(t) = \mathbb{E}^{\mathbb{P}}[E(t)] \quad \text{(averaged over all paths)}$$

Step 6: Potential Future Exposure (PFE)

$$\mathrm{PFE}_\alpha(t) = \inf\{x : \mathbb{P}(E(t) \leq x) \geq \alpha\}$$

Compute this across time for confidence levels $\alpha = 95\%, 99\%$.

Step 7: Plot or Tabulate Results

- Simulated paths of $V^S(t)$ and $E(t)$,
- EE and PFE envelopes (time series).

1.2 Simulation Results

The following tables compare the calibrated transition matrix P^{init} — often derived from rating agency data or historical frequencies — with the empirically simulated stressed matrix $\widehat{P}^{\text{stress}}$, generated via dynamic Monte Carlo paths driven by a CIR intensity process with $\lambda_0 = 0.15$, mimicking a systemic downturn.

Table 1: Stochastic Credit Migration Matrix (1-Year Horizon)

From / To	AAA	AA	A	BBB	BB	B	CCC	D
AAA	0.891	0.096	0.008	0.002	0.003	0.000	0.000	0.000
AA	0.009	0.901	0.074	0.010	0.003	0.003	0.000	0.000
A	0.001	0.029	0.890	0.064	0.010	0.005	0.000	0.000
BBB	0.001	0.004	0.065	0.844	0.063	0.016	0.002	0.005
BB	0.000	0.002	0.008	0.070	0.780	0.102	0.013	0.024
B	0.000	0.002	0.003	0.007	0.051	0.827	0.042	0.069
CCC	0.000	0.000	0.011	0.011	0.020	0.073	0.657	0.227
D	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000

Table 2: Empirical Transition Matrix $\widehat{P}$ (Stressed Simulation – 5-year horizon)

From / To	AAA	AA	A	BBB	BB	B	CCC	D
AAA	0.867	0.067	0.044	0.000	0.000	0.022	0.000	0.000
AA	0.006	0.822	0.127	0.023	0.011	0.009	0.001	0.000
A	0.003	0.054	0.793	0.103	0.027	0.013	0.001	0.006
BBB	0.001	0.008	0.087	0.745	0.091	0.042	0.006	0.021
BB	0.000	0.009	0.020	0.097	0.654	0.118	0.022	0.079
B	0.000	0.003	0.009	0.025	0.069	0.706	0.042	0.145
CCC	0.000	0.000	0.018	0.009	0.053	0.088	0.504	0.327
D	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000

1. **Increased Downgrade and Default Likelihood:** The empirical matrix $\widehat{P}^{\text{stress}}$ reflects elevated downgrade pressure. For example, A-rated names now have a 10.3% probability of falling to BBB (vs. 6.4% in P^{init}) and a combined

$$0.103 + 0.027 + 0.013 + 0.001 + 0.006 \approx 15.0\%$$

probability of migrating to *BBB, BB, B, CCC, or D* — up from

$$0.064 + 0.010 + 0.005 + 0.000 + 0.000 \approx 7.9\%$$

initially. The probability of default (PD) from A rises from

$$0.000 \quad (\approx 0.0\%) \quad \text{to} \quad 0.006 \quad (\approx 0.6\%).$$

2. **Systematic Stress Realism:** The stressed simulation embeds macroeconomic deterioration via high λ_0 and a scaling factor on Q. This boosts migration intensities and amplifies deterioration risk across the rating spectrum — for instance, CCC-to-D increases from

$$0.227 \quad (\text{Table 1},\ 22.7\%) \quad \text{to} \quad 0.327 \quad (\text{Table 2},\ 32.7\%).$$

   While the apparent *upgrade* probability from CCC increases from

$$0.000 + 0.000 + 0.011 + 0.011 + 0.020 + 0.073 = 0.115 \quad (\approx 11.5\%)$$

   to

$$0.000 + 0.000 + 0.018 + 0.009 + 0.053 + 0.088 = 0.168 \quad (\approx 16.8\%),$$

   this reflects the longer 5-year horizon in $\widehat{P}^{\text{stress}}$, not improved recovery odds. In reality, reversals should be rare under stressed liquidity conditions.

3. **Path Irreversibility for Lower Ratings:** Under P^{init}, CCC entities have limited recovery options — e.g., a $\approx 5.7\%$ chance of improving to BBB, computed as

$$0.011 \quad (\text{Table 1},\ BBB).$$

   Under $\widehat{P}^{\text{stress}}$, CCC firms are almost trapped:

$$0.504 \quad (\approx 50.4\%) \text{ remain CCC}, \quad 0.327 \quad (\approx 32.7\%) \text{ default},$$

   with minimal upward migration, capturing downgrade inertia under systemic stress.

4. **Entropic Dispersion and Migration Breadth:** The stressed matrix reveals entropic dispersion — a broader spread of migration outcomes, including non-zero flows to default from nearly every rating. For example, in $\widehat{P}^{\text{stress}}$, the probability of default from BB is

$$0.079 \quad (\approx 7.9\%),$$

   compared with

$$0.024 \quad (\approx 2.4\%)$$

   in P^{init}. This statistical richness is crucial for tail-sensitive capital metrics such as CVA VaR or Expected Shortfall under FRTB-IMA.

5. **Regulatory Compliance and Conservatism:** Regulators emphasize conservative, forward-looking deterioration estimates. P^{init} is retrospective and optimistic — e.g.,

$$0.000 \quad (\text{PD for AAA})$$

   — while $\widehat{P}^{\text{stress}}$ embeds forward-looking risk, giving

$$0.000 \quad (\text{no observed AAA defaults in stressed simulation but downgrade risk increased}).$$

6. **Credit Spread Dynamics via Structural Model:** The use of a Cox–Ingersoll–Ross (CIR) process to modulate the generator matrix Q ties migration probabilities to underlying credit spread dynamics. This makes the framework compatible with joint credit–market risk modeling, enabling integrated CVA and FRTB-IMA capital simulations.

In summary, $\widehat{P}^{\text{stress}}$ provides a more granular, deteriorated, and structurally consistent view of credit risk — better aligned with CVA capital requirements under stress.

Table 3 shows the evolution of EE/PFE over a 5-year horizon, based on Monte Carlo simulations incorporating both market dynamics and credit deterioration, with a high credit spread volatility ($\sigma = 55\%$).

Table 3: Expected and Potential Future Exposure (Volatility $\sigma = 55\%$)

	0.000	0.500	1.000	1.500	2.000	2.500	3.000	3.500	4.000	4.500	5.000
Expected Exposure (EE)	1.000	0.983	0.916	0.850	0.771	0.722	0.663	0.614	0.573	0.529	0.496
PFE 95%	1.000	1.251	1.311	1.368	1.430	1.505	1.459	1.520	1.485	1.474	1.477
PFE 99%	1.000	1.381	1.516	1.660	1.688	1.845	1.809	1.771	1.903	1.957	2.202

EE represents the average positive mark-to-market value of the derivative across all simulated paths. EE declines gradually from 1.000 at inception to approximately 0.496 by year 5, reflecting both the amortization of market exposure and the increasing effect of credit spread discounting as counterparties deteriorate in credit quality.

PFE at the 95% and 99% confidence levels reflects the distributional tail of exposure outcomes. These values remain elevated well above the EE across all maturities, signaling persistent tail risk. For example, by year 5, the 99% PFE reaches 2.202, more than 4 times the average EE. This highlights the asymmetry between average and stressed scenarios under credit stress.

The growing divergence between EE and PFE over time illustrates the nonlinear impact of credit spread volatility. The CIR-driven intensity process introduces strong dispersion across paths, amplifying the risk of downgrade or default in a subset of scenarios. This makes tail exposure a dominant concern, underscoring the importance of using high-volatility stress parameters in regulatory capital assessments.

Finally, Table 4 summarizes the distribution of simulated counterparty credit ratings at the end of the 5-year simulation horizon.

Table 4: Final Rating Distribution at 5-Year Horizon

	AAA	AA	A	BBB	BB	B	CCC	D
Frequency	4	39	82	70	29	33	2	241
Percentage	0.80%	7.80%	16.40%	14.00%	5.80%	6.60%	0.40%	48.20%

This table provides a snapshot of rating migration outcomes across all simulation paths. Key observations include:

– Nearly half of all counterparties (48.2%) end up in default ('D') by year 5, underscoring the severe credit stress embedded in the simulation, particularly under the high-intensity CIR model.

– Only 16.4% of entities retain their original rating ('A'), indicating moderate credit inertia in the face of systemic stress.

– Downgrades are widespread: 14.0% migrate to 'BBB', 5.8% to 'BB', and 6.6% to 'B'. These results align with empirical credit cycle behavior during market downturns.

– Upgrades to 'AA' (7.8%) or 'AAA' (0.8%) are rare, reflecting the asymmetric nature of rating transitions — with deterioration being more likely than improvement, especially under stressed scenarios.

Overall, this distribution supports key objectives of credit risk simulation frameworks — providing realistic, forward-looking migration patterns that inform exposure-at-default (EAD), CVA, and default risk charge (DRC) capital calculations under FRTB/ICAAP frameworks.

Next, we compute a proxy for CVA using the simulation results for expected exposure (EE), calibrated under the physical measure. Assuming a constant discount factor and recovery rate, CVA is estimated via:

$$\mathrm{CVA}(t) = (1 - R) \times \mathbb{E}[\mathrm{EE}(t) \cdot \mathrm{PD}(t) \cdot D(t)]$$

where $D(t) = e^{-rt}$ is the discount factor and $PD(t)$ is the empirical cumulative default probability up to time t.

Total CVA Proxy: **0.481717**

This represents the discounted expected credit loss over the life of the derivative exposure, under a base-case scenario calibrated to historical credit dynamics.

The DRC aims to capture losses that may arise from default events within the trading book. As a simplified proxy, we extract the 95% quantile of exposure at the first time of default across all simulation paths:

$$\mathrm{DRC} \approx 95\%\text{-quantile}\left(V^S(\tau_{\text{default}})\right)$$

DRC Proxy (95%): **0.596453**

This figure reflects a tail-risk measure consistent with regulatory expectations for capital charges, based on simulated default scenarios.

To explore credit deterioration, we simulate exposures under stressed CIR parameters. The results demonstrate sensitivity of the CVA proxy to changes in the mean-reversion rate κ and initial intensity λ_0.

Table 5: CVA Proxy Under Stress Scenarios

$\kappa \backslash \lambda_0$	0.01	0.03	0.05
0.3	0.289253	0.344241	0.383228
0.5	0.274063	0.329484	0.361337
0.7	0.258693	0.308909	0.328015

We observe that CVA increases notably as λ_0 rises, indicating the expected exposure is more sensitive to the initial credit deterioration level than to mean-reversion speed. This aligns with intuition: higher λ_0 leads to more rapid migration into lower credit states, increasing default likelihood.

The results provide insights into the potential magnitude of CVA losses and associated capital for regulatory reporting purposed.

Appendice: FRTB

The stochastic CIR-driven migration model introduced above not only supports regulatory FRTB calculations such as DRC and CVA-ES, but also forms the basis for simulating forward-looking exposure profiles. In what follows, we detail the procedure to compute Expected Exposure (EE) and Potential Future Exposure (PFE), using the same model engine under the real-world measure.

The Internal Models Approach (IMA) of the Fundamental Review of the Trading Book (FRTB) requires computing regulatory capital for market risk using simulation-based methods.

The stochastic generator matrix driven by CIR-calibrated credit intensities, discussed previously under Corporate Bond Pricing, is fed into the required FRTB IMA metrics.

Let:

λ_t: CIR intensity process calibrated to market CDS spreads,

$Q(t)$: generator matrix of rating transitions as a function of λ_t,

$P(t,T) = \mathbb{E}\left[\exp\left(\int_t^T Q(u)\,du\right)\middle|\mathcal{F}_t\right]$: transition matrix over horizon $[t,T]$,

$\pi_{ij}(T)$: transition probability from rating i to j over time T,

LGD_i: loss given default,

L: loss variable; $\mathcal{P}_t = L_{t+\Delta} - L_t$: P&L.

Expected Shortfall (ES) is the key market risk measure under FRTB with confidence interval designated at 97.5%:

$$\mathrm{ES}_{97.5\%} = \frac{1}{0.025}\int_0^{0.025} \mathrm{VaR}_u(L)\,du$$

Using simulated paths of λ_t and rating transitions, we generate credit spread movements and compute portfolio mark-to-market changes:

$$\mathrm{MTM}_t = f(\mathrm{spread}_t, \mathrm{rating}_t) \Rightarrow \mathcal{P}_t = \mathrm{MTM}_{t+\Delta} - \mathrm{MTM}_t$$

Then compute empirical ES from $\mathcal{P}_t$.

The Stressed Expected Shortfall (SES) is similar to the above but under stressed parameters $(\kappa^*, \theta^*, \sigma^*)$ fitted to a crisis period (say 2008 GFC):

$$d\lambda_t^{\mathrm{stress}} = \kappa^*(\theta^* - \lambda_t)\,dt + \sigma^*\sqrt{\lambda_t}dW_t$$

Simulate under this new measure and recompute ES.

Default Risk Charges (DRC) accounts for losses due to rating migrations and defaults:

$$\mathrm{DRC} = \sum_i \sum_j \pi_{ij}(1\mathrm{y}) \cdot \Delta\mathrm{EAD}_{ij} + \pi_{iD}(1\mathrm{y}) \cdot \mathrm{LGD}_i$$

Use:

$$P(0,1) = \mathbb{E}\left[\exp\left(\int_0^1 Q(u)\,du\right)\right]$$

driven by CIR paths, to compute the 1-year migration probabilities π_{ij}, including default, and apply the above DRC formula.

Risk Not In VaR (RNIV) FRTB requires add-ons for risks not captured in VaR models. This includes:

- Wrong-way risk (correlation breakdown),
- Model specification uncertainty,
- Illiquid risk factors.

The CIR generator framework may not capture non-linear co-movements or jump-to-default risk correlations. These must be quantified separately.

Credit Value Adjustment (CVA) Risk Capital The CVA VaR/ES component measures the volatility of CVA exposure. CVA itself is given by:

$$\mathrm{CVA} = \int_0^T \mathbb{E}[\mathrm{LGD}_t \cdot \mathrm{EE}_t \cdot \lambda_t \cdot e^{-rt}]\, dt$$

Use CIR-calibrated λ_t and stochastic $Q(t)$ to simulate counterparty credit events. Combine with simulated exposures to compute CVA P&L, and apply ES methodology.

Potential Future Exposure (PFE) Defined as:

$$\mathrm{PFE}_\alpha(t) = \inf\{x : \mathbb{P}(\mathrm{Exposure}_t \leq x) \geq \alpha\}$$

CIR-based λ_t drives rating transitions and credit spreads, which in turn affect exposure profiles. Simulate distribution of exposure at each future time and extract percentiles.

Listing 1: GeneratorMatrix

```
class GeneratorMatrix:

    def __init__(self, transition_matrix):
        self.P = transition_matrix
        self.ratings = ['AAA', 'AA', 'A', 'BBB', 'BB', 'B', 'CCC', 'D']
        self.tilde_Q = None
        self.Q = None

    def compute(self):
        N = self.P.shape[0]
        tilde_Q = self.P - np.eye(N)
        term = tilde_Q.copy()
        for k in range(2, N + 1):
            term = -np.dot(term, (self.P - np.eye(N))) / k
            tilde_Q += term

        Q = tilde_Q.copy()
        for i in range(N):
            for j in range(N):
                if i != j and Q[i, j] < 0:
                    Q[i, j] = 0
            Q[i, i] = -np.sum(Q[i, :]) + Q[i, i]

        self.tilde_Q, self.Q = tilde_Q, Q
        return self.tilde_Q, self.Q

    def display(self):
        print("Original Transition Matrix:")
        print(pd.DataFrame(self.P, index=self.ratings, columns=self.ratings).round(4))
        print("\nUnadjusted Generator Matrix (tilde_Q):")
        print(pd.DataFrame(self.tilde_Q, index=self.ratings, columns=self.ratings).round(5))
        print("\nAdjusted Generator Matrix (Q):")
        print(pd.DataFrame(self.Q, index=self.ratings, columns=self.ratings).round(5))
```

Listing 2: CIRSimulator

```
class CIRSimulator:

    def __init__(self, lambda0, kappa, nu, sigma, dt, n_paths, n_steps):
        self.lambda0 = lambda0
        self.kappa = kappa
        self.nu = nu
        self.sigma = sigma
        self.dt = dt
        self.n_paths = n_paths
        self.n_steps = n_steps

    def simulate(self):
        paths = np.zeros((self.n_paths, self.n_steps + 1))
        paths[:, 0] = self.lambda0
        for i in range(self.n_paths):
            for t in range(self.n_steps):
                dW = np.random.normal(0, np.sqrt(self.dt))
                lam = paths[i, t]
                paths[i, t + 1] = np.abs(lam + self.kappa * (self.nu - lam) * self.dt
                + self.sigma * np.sqrt(lam) * dW)
        return paths
```

Listing 3: RatingSimulator Class

```
class RatingSimulator:

    def __init__(self, Q, ratings, scale_factor=100):
        self.Q = Q
        self.ratings = ratings
        self.scale_factor = scale_factor

    def simulate(self, lambda_paths, dt, initial_index):
        n_paths, n_steps = lambda_paths.shape
        ratings_matrix = np.zeros((n_paths, n_steps), dtype=int)
        ratings_matrix[:, 0] = initial_index
        cache = {}

        for i in range(n_paths):
            for t in range(n_steps - 1):
                lam = round(lambda_paths[i, t], 4)
                if lam not in cache:
                    cache[lam] = expm(self.Q * lam * dt * self.scale_factor)
                P = cache[lam]
                current = ratings_matrix[i, t]
                if current == len(self.ratings) - 1:
                    ratings_matrix[i, t + 1] = current
                else:
                    probs = P[current] / P[current].sum()
                    ratings_matrix[i, t + 1] = np.random.choice(len(self.ratings), p=probs)
        return ratings_matrix
```

Listing 4: MarketSimulator Class

```
class MarketSimulator:
    def __init__(self, n_paths, n_steps, T, mu=0.0, sigma_mkt=0.2):
        self.n_paths = n_paths
        self.n_steps = n_steps
        self.T = T
        self.mu = mu
        self.sigma_mkt = sigma_mkt

    def simulate(self):
        dt = self.T / self.n_steps
        grid = np.linspace(0, self.T, self.n_steps + 1)
        paths = np.zeros((self.n_paths, self.n_steps + 1))
        paths[:, 0] = 1.0
        for i in range(self.n_paths):
            for t in range(self.n_steps):
                dW = np.random.normal(0, np.sqrt(dt))
                paths[i, t + 1] = paths[i, t] * np.exp((self.mu - 0.5 * self.sigma_mkt**2) * dt + self.sigma_mkt * dW)
        return np.maximum(paths, 0), grid
```

Listing 5: ExposureCalculator Class

```
class ExposureCalculator:
    def __init__(self, ratings, rating_spreads):
        self.ratings = ratings
        self.spread_dict = rating_spreads

    def map_ratings_to_spreads(self, rating_paths):
        return np.vectorize(lambda x: self.spread_dict[self.ratings[x]])(rating_paths)

    def compute_discount_paths(self, spread_paths, time_grid):
        return np.exp(-spread_paths / 10000 * time_grid)

    def compute_adjusted_exposure(self, market_paths, discount_paths):
        return market_paths * discount_paths

    def compute_stats(self, adjusted_exposure):
        EE = np.mean(adjusted_exposure, axis=0)
        PFE_95 = np.percentile(adjusted_exposure, 95, axis=0)
        PFE_99 = np.percentile(adjusted_exposure, 99, axis=0)
        return EE, PFE_95, PFE_99
```

Listing 6: Main

```
P = np.array([
    [0.8910, 0.0963, 0.0078, 0.0019, 0.0030, 0.0000, 0.0000, 0.0000],
    [0.0086, 0.9010, 0.0747, 0.0099, 0.0029, 0.0029, 0.0000, 0.0000],
    [0.0009, 0.0291, 0.8894, 0.0649, 0.0101, 0.0045, 0.0000, 0.0000],
    [0.0006, 0.0043, 0.0656, 0.8427, 0.0644, 0.0160, 0.0018, 0.0045],
    [0.0004, 0.0022, 0.0079, 0.0719, 0.7764, 0.1043, 0.0127, 0.0241],
    [0.0000, 0.0019, 0.0031, 0.0066, 0.0517, 0.8246, 0.0435, 0.0685],
    [0.0000, 0.0000, 0.0116, 0.0116, 0.0203, 0.0754, 0.6493, 0.2319],
    [0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 1.0000]
])

ratings = ['AAA', 'AA', 'A', 'BBB', 'BB', 'B', 'CCC', 'D']
```

```python
rating_spreads = {'AAA': 20, 'AA': 30, 'A': 50, 'BBB': 80, 'BB': 200, 'B': 400, 'CCC': 800, 'D': 10000}

# Parameters
lambda0, kappa, nu, sigma = 0.15, 0.7, 0.02, 0.55
n_paths, n_steps, T = 500, 20, 5.0
dt = T / n_steps
initial_index = 2  # 'A'

# Run the pipeline
gen = GeneratorMatrix(P)
_, Q = gen.compute()

cir = CIRSimulator(lambda0, kappa, nu, sigma, dt, n_paths, n_steps)
lambda_paths = cir.simulate()

rater = RatingSimulator(Q, ratings)
rating_paths = rater.simulate(lambda_paths, dt, initial_index)

exp_calc = ExposureCalculator(ratings, rating_spreads)
spread_paths = exp_calc.map_ratings_to_spreads(rating_paths)

market_sim = MarketSimulator(n_paths, n_steps, T)
market_paths, time_grid = market_sim.simulate()

discount_paths = exp_calc.compute_discount_paths(spread_paths, time_grid)
adjusted_exposure = exp_calc.compute_adjusted_exposure(market_paths, discount_paths)
EE, PFE_95, PFE_99 = exp_calc.compute_stats(adjusted_exposure)
```