POLITECNICO
MILANO 1863

Financial Engineering 2023/24

Assignment 5 - Group 20

Margherita Bencini, Alessandro Torazzi, Matteo Torba, Giovanni Urso

Contents

1 Certificate Pricing

1.a Value Upfront using NIG model

We were asked to price the upfront for a swap contract, in which a commercial bank wanted to enter to hedge a stuctured product.

Working with a NIG model, we started by calibrating its parameters with `NMMCalibration`, which minimizes the mean squared difference between the model's and the respective market's prices and we obtained the following optimal parameters.

σ	η	k
0.104934	12.737181	1.316103

Table 1: NIG hyperparameters

In order to calculate the NPV of the structured bond, we needed to compute the probability of the stock being over a certain strike (3200 €) and weight the discounted cash flows for each scenario by the respective probability. We knew that we could compute this probability of avoiding Early Redemption as done in Lewis (2001), as it would be the expected payoff of a digital call option:

$$^1P_{notER} = \frac{1}{2} + \frac{1}{\pi}\int_0^\infty Re[\frac{e^{iuk}\Phi_T(u)}{iu}]du$$

Being Φ_T the characteristic function for the NIG model:

$$^2\Phi_T(u) = e^{-iu\cdot ln(L(\eta))+ln(L(\frac{u^2+i(1+2\eta)u}{2}))}$$

To implement those calculations, we needed the log-moneyness $k = log(\frac{F0_{rd}}{K})$ computed with the Forward at reset date, which was actually two business days before the coupon payment date.

Having thus obtained the Probability of Early Reset P_{ER}, we were then able to compute the NPV for party A (Commercial Bank), as the sum of the NPVs of the two cases (with/without Early Redemption) weighted with the corresponding probabilities:

$$^3NPV = P_{ER}NPV_{ER} + (P_{notER})NPV_{notER}$$

$$^4NPV_{ER} = 1 - B(0,1) + \Sigma_{i=1}^4(B(0,\frac{i}{4})\delta(\frac{i}{4},\frac{i+1}{4}))1.3\% - 6\%\delta(0,1)B(0,1)$$

[1] P_{notER} : Probability of no early redemption, k : log-moneyness, Φ_T : characteristic function

[2] ln(L): Laplace transform logarithm

[3] P_{ER} : Probability of Early Redemption, NPV_F : NPV without early redemption

[4] $\delta(\frac{i}{4},\frac{i+1}{4})$: time interval among $\frac{i}{4}$ and $\frac{i+1}{4}$, $B(t_0,T)$: deterministic discount

$$^{5}NPV_{notER} = 1 - B(0,2) + \Sigma_{i=1}^{8}(B(0,\frac{i}{4})\delta(\frac{i}{4},\frac{i+1}{4}))1.3\% - 2\%\delta(1,2)B(0,2)$$

Having computed the NPV neglecting the upfront, we could finally value said Upfront X as

$$^{6}X = NPV \cdot N = 2\,489\,120.52€$$

To check our result, we also calculated the Upfront via Monte-Carlo method: having computed 10^5 asset simulations using NIG dynamics in `SimulateNIG`, we could easily obtain the result and 95% confidence interval via Matlab **`normfit`**

$$X = 2\,474\,950.59€$$

$$X_{95\%} = [2456870.0€, 2493031.13€]$$

1.b Value Upfront with Black model

Asked to value the Upfront with a different model, we chose Black's one, as using Black we had the possibility to easily compute the probability of no early redemption, again exploiting the digital call value not discounted:

$$^{7}P_{notER} = N(d_2) = N(\frac{log(\frac{F0_{rd}}{K})}{\sqrt{T_{rd}}\sigma_{bl}} - \frac{1}{2}\sqrt{T_{rd}} \cdot \sigma_{bl})$$

We were then able to calculate the NPV^{BL} similarly as before:

$$NPV^{bl} = P^{bl}_{ER}NPV^{bl}_{ER} + (P^{bl}_{notER})NPV^{bl}_{notER}$$

Therefore obtaining the Upfront Value as well:

$$X^{bl} = 1\,538\,756.70€$$

Computing the difference between the Black's Upfront and the NIG one, we can see that that the error is considerable. This is due to the presence of Digital Risk, that pricing with the Black model doesn't allow to consider in the computation.

$$E = X^{NIG} - X^{bl} = 950\,363.8€$$

[5] $\delta(\frac{i}{4},\frac{i+1}{4})$: time interval among $\frac{i}{4}$ and $\frac{i+1}{4}$, $B(t_0,T)$: deterministic discount
[6] N: Notional
[7] rd: reset date

In order to check the impact of the digital risk on the Black model's price error, we calculated it using `Digital`, and then added this value to the Probability of avoiding early redemption, obtaining the corrected probability and therefore the corrected NPV^{bl-c} and the corrected Black's upfront X^{bl-c}:

$$NPV^{bl-c} = P_{ER}^{bl-c} NPV_{ER}^{bl-c} + (P_{notER}^{bl-c}) NPV_{notER}^{bl-c}$$

$$X^{bl-c} = NPV^{bl-c} N = 2\,341\,378.05€$$

The corrected Black's upfront isn't that different from the value obtained with the NIG model, as the addition of the digital risk gives a more realistic result.

1.c Three-year expiry case model choice

After we have priced the upfront for the swap involving the structured bond, we are asked to evaluate the same quantity for a more exotic product. The new contract features consist in a longer expiry and a set of multiple reset dates which determine whether the early redemption happens before maturity. The new condition states that the product continues until expiry only if the price of the underlying is above the threshold for all the reset dates, otherwise the respective coupon for each year is payed up to the last reset date, when the contract immediately ends. We noticed that such exotic payoff has a strong path dependency, and finding a closed formula for early redemption probabilities would result in conditioning on the middle reset dates, solving multivariate integrals. Considering a Monte-Carlo approach would be the best method to compute the upfront value, as this will be the sample mean of the realised exotic payoff for each simulation.

1.d Value upfront for three-year expiry case

Asked to value the upfront of the three years contract, we simulated again the Asset evolution using `SimulateNIG`, this time in a two years' window, as we needed the asset values at both the redemption dates.

As there were more possible scenarios than we had in the previous contract, we decided to compute the NPV considering the probabilities to pay each coupon, and the ones to pay the two possible fixed legs:

$$^{8}NPV = -C_{1y} P[S(t_1) < K] - C_{2y} P[S(t_2) < K] + (LF_{3y} - C_{3y}) P[S(t_1) > K \cap S(t_2) > K] +$$

[8] C_{iy} : Coupon at i-th year, $P[X]$: Probability of X happening, $S(t_i)$ = Stock price at i-th reset time, LF_{iy} : Fixed Leg for the i-th year

$$+LF_{2y}P[S(t_1) < K \cup S(t_2) < K]$$

We were then able to obtain the upfront value and, using `normfit`, the 95% confidence interval:

$$X = 3\,648\,330.77€$$

$$X_{95\%} = [3\,612\,153.63€, 3\,684\,507.91€]$$

2 Bermudian Swaption Pricing via Hull-White

We were asked to price on the 15th of February 2008 a 10y Bermudan yearly Payer Swaption Strike 5% non-call 2, considering a single curve framework and a 1-factor Hull-White model with $\alpha = 11\%$, $\sigma = 0.8\%$.

2.a Price via Tree

The first request for this exercise was to price the Bermudan Swaption, via a tree for the underlying Ornstein-Uhlenbeck process. As we were in a Gaussian HJM framework, we knew that the expression for r_t was given by:

$$r_t = x_t + \Phi_t$$

$$\begin{cases} dx_t = -ax_t + \sigma dW_t \\ x_{t_0} = 0 \end{cases}$$

We were considering an option written on a Swap, that gave us the right to enter into a Swap Contract (maturity 10 years) every year from the 2^{nd} to the 9^{th}. The expression of the price of a "European" Swaption Payer, with only one reset date T_α and expiry of the Swap T_ω, would be:

$$SP_{\alpha\omega} = \mathbb{E}[D(t_0, T_\alpha)BPV_{\alpha\omega}(S_{\alpha\omega}(T_\alpha) - K)^+]$$

With

$$S_{\alpha\omega}(T_\alpha) = \frac{1 - B_{\alpha\omega}(T_\alpha)}{BPV_{\alpha\omega}(T_\alpha)}$$

and

$$BPV_{\alpha\omega}(T_\alpha) = \sum_{i=\alpha+1}^{\omega} \delta(t_{i-1}, t_i)B(T_\alpha)$$

In the case of a Bermudan Option instead, in each of the Early Exercise Dates we would have to evaluate the Swap Price $S_{\alpha\omega}$ and then decide whether to exercise or not. As we did not know a Closed Formula for this option, in order to compare the expected payoffs in the situations and price with the most convenient one, a numerical technique was mandatory: the Trinomial Tree.

We then started our computation for the forward discounts to obtain the $S_{\alpha\omega}(T_\alpha)$.

According to the Hull White model theory, we knew the following expression for the forward ZC held:

$$B(t_i, t_i, t_i + \tau) = B(t_0, t_i, t_i + \tau) e^{-x_i \frac{\sigma(0,\tau)}{\sigma} - \frac{1}{2}\int_{t_0}^{t_i} [\sigma(u, t_i+\tau)^2 - \sigma(u, t_i)^2]\, dx} \tag{1}$$

$$\sigma(s,t) = \sigma \frac{1 - e^{-a(t-s)}}{a} \tag{2}$$

From the bootstrap we obtained the discounts: after interpolating them for each date of a possible reset t_i = 2,3...10, we computed the matrix of forwards. The matrix was built in order to we have in every column the forward rates with respect to the same reset date.

As the expression of $B(t_i, t_i, t_i + \tau)$ depended only on x_i, we simulated his value via a Trinomial Tree. In the Trinomial Tree we discretized time with a time step dt. We computed dt as $dt = \frac{1}{N}$, with N therefore being the number of steps in each year and thus the control variable for precision. We then computed the following quantitites, according to theory:

$$\hat{\mu} = 1 - e^{-adt} \qquad \hat{\sigma} = \sigma\sqrt{\frac{1 - e^{2adt}}{2a}}$$

$\Delta x = \sqrt{3}\hat{\sigma}$ $\quad$ $x = l$ with l from - l_{max} to l_{max}, with l_{max} being: $\frac{1-\sqrt{\frac{2}{3}}}{\hat{\mu}} < l_{max} < \frac{\sqrt{\frac{2}{3}}}{\hat{\mu}}$
We selected l_{max} = 1669, the smallest possible, as its value did not impact the precision of the method and we wanted to avoid a big tree for computational simplicity.
In the Trinomial Tree from a given node the underlying had only three possibilities of evolution, going up of a quantity Δx, down of the same quantity Δx or staying the same. The probabilities of these evolutions depended whether we were in the middle of the tree or at the extremes with $x = l_{max}$ or l_{min}: we computed those following the theoretic formulas.

We then started our evaluation from the last date in which we have the right to exercise, at the 9^{th} year. At this time, we computed the payoff

$$(S_{\alpha\omega}(T_\alpha) - K)^+$$

with T_α = 9y. The idea is to move backwardly along the tree discounting until reaching t_0.

The procedure used was, summarized:

- Computation for each realization x_i of the step the forward discount via equation (1)
- Computation of the stochastic discounts. From theory, we knew the following approximation of the stochastic discounts for a Trinomial Tree:

$$D(t_i, t_{i+1}) = B(t_i, t_{i+1})[exp(-\frac{1}{2}(\hat{\sigma}^*)^2) - \frac{\hat{\sigma}^*}{\hat{\sigma}}[e_{i+1}^{-adt} + \hat{\mu}x_i]]$$

with

$$\hat{\sigma}^* = \frac{\sigma}{a}\sqrt{dt - 2\frac{1-e^{adt}}{a} + \frac{1-e^{-2adt}}{2a}}$$

The most important thing at this point was to consider the Δx correctly, distinguishing three cases whether the previous point was one Δx upper, lower or at the same level.

- Moving backwards. We considered each possible node at t_{i-1} that could have led to the nodes in t_i, dividing the cases in going up, down or staying the same from t_{i-1}. From the value of the nodes at t_i we move backwardly multiplying each stochastic discount for its probability and assigning to each node in t_{i-1} its possible future patterns. With this computation we then managed to discount the payoff for each value of x in t_{i-1}.

- Checking for Early Reset Dates. At this point we had to check whether the time step t_{i-1} for which we had computed the stochastic discounts was an Early Reset Date. If it was, we would have computed the payoff of the Swaption $(S_{t_{i-1}\omega}(t_{i-1}) - K)^+$ with t_{i-1}, the given Early Reset Date, and we would have compared it with the Continuation Value computed in the previous point, taking the maximum between the two values.

- Updating. We then updated the previously computed values, "moving" ourself in the tree in order to reach earlier nodes.

The procedure had to be repeated until reaching t_0. The price obtained for the option was, with N = 1000:

$$PRICE = 2.17867634\%$$

The price is expressed as percentage over the Notional of the Swaption.

We also took note of the number of Early Exercises at each year from the 2^{nd} to the 9^{th}, represented in the table below:

Expiry	Number of Early Exercise
9	1677
8	1670
7	1664
6	1659
5	1652
4	1646
3	1640
2	1633

This table shows that at least for some values of x_i it is not convenient to exercise at the first occasion, at year 2, but it's better to wait, as the payoff is given by the Continuation Value. As the number of early exercises continues to increase over time, we could infer that it's convenient to wait for higher rates even when in the money, hoping to have a better payoff at later dates.

2.b Tree check

To check that the tree had been correctly implemented, we priced the European Swaptions with Expiries ranging form 2 years to 9 years and Maturities 10 years both via the trinomial tree and and via the Jamshidian formula.
In order to compute the prices via the trinomial tree, we navigated the tree backwardly, we computed the Payoff of the European Swaption at each Expiry and discounted them back "step-by-step" along the tree to the initial time t_0, with the same procedure described in the section above.
In the other case, we started by using the Jamshidian formula to compute the prices of the Call options written on a coupon bond, where the coupons were imposed to be equal to the Strike of the Swaption ($K = 5\%$) and the Strike of the Call (K_{call}) equal to 1: to price the Call options for each Expiry t_α, the value of $X^*_{t_\alpha}$ was computed by matching the price of the coupon bond with K_{call} and solving numerically via the Matlab built-in function `fsolve` the following equation:

$$K_{call} = P(t_\alpha; t_\alpha, t_\omega) = \sum_{i=\alpha+1}^{\omega} c_i B(t_\alpha; t_\alpha, t_i; X_{t_\alpha}) =$$

$$= \sum_{i=\alpha+1}^{\omega} c_i B(t_0; t_\alpha, t_i) e^{-\frac{X_{t_\alpha}}{\sigma}(\sigma(t_\alpha,t_i)-\sigma(t_\alpha,t_\alpha)) \int_{t_0}^{t_\alpha}[\sigma(u,t_i)^2-\sigma(u,t_\alpha)^2]\,dx} \tag{3}$$

where:

- $B(t_0; t_\alpha, t_i)$ is the forward discount obtained from the bootstrap;
- $\sigma(u, t)$ was already defined in equation (2)
- c_i is the coupon defined as

$$\begin{cases} c\delta_i & \text{for } i < \omega \\ 1 + c\delta_i & \text{for } i = \omega \end{cases}$$

A graphical solution of the equation (3) for the Expiry $t_i = 9$ can be observed in Figure 1.

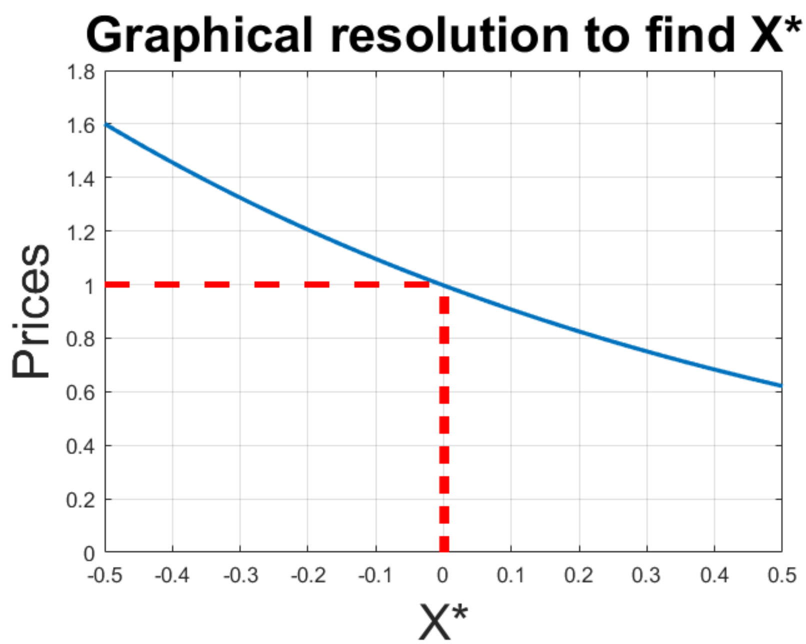

Figure 1: Graphical solution of the equation to find X*

Once found the values of $X^*_{t_\alpha}$, we wrote the coupon bond Call options as the weighted sum of Call options written on a zero coupon bond with Expiry t_α and Maturity t_i for $t_i = t_{alpha} + 1, ..., 10$.

$$C_{t_\alpha}(t_0) = \sum_{i=\alpha+1}^{10} c_i C_{t_i}(t_0)$$

We computed the prices of the ZCB plain vanillas via the function `ZC_Call_Gaussian_HJM` using the formula given in the Gaussian HJM framework:

$$C_{t_i}(t_0) = B(t_0, t_\alpha)[B(t_0, t_\alpha, t_i)\mathrm{N}(d_1) - B(t_\alpha; t_\alpha, t_i; X^*_{t_\alpha})\mathrm{N}(d_2)]$$

where:

$$d_{1,2} = \frac{ln\left(\frac{B(t_0;t_\alpha,t_i)}{B(t_\alpha;t_\alpha,t_i;X^*_{t_\alpha})}\right)}{V\sqrt{t_\alpha - t_0}} \pm V\sqrt{t_\alpha - t_0}$$

$$V^2 = \frac{1}{t_\alpha - t_0}\int_{t_0}^{t_\alpha} (\sigma(t, t_i) - \sigma(t, t_\alpha))^2 dt$$

By weighting the results for the coupons c_i and summing them up, we finally retrieved the prices of the coupon bond Call options and we used the Put-Call parity where the underlying is the coupon bond price, to obtain the coupon bond Put options prices equivalent to the European Swaption Prices.

For the value of yearly time steps N=1000, the results obtained, both for the European Swaptions and the coupon bond Put options, for different Expiries, and their respective absolute errors, are reported in Table 2.

Expiry	European Swaption Value via tree	Put price via Jamshidian formula	Error
2	0.81811775%	0.81799165%	1.260998e-06
3	1.20324147%	1.20275223%	4.892352e-06
4	1.42359969%	1.42313980%	4.598853e-06
5	1.48071882%	1.48028846%	4.303600e-06
6	1.40115156%	1.40068487%	4.666854e-06
7	1.19237736%	1.19197740%	3.999557e-06
8	0.87526846%	0.87500303%	2.654277e-06
9	0.47406803%	0.47377627%	2.917565e-06

Table 2: European Swaption price computed via the trinomial tree for N=100 time steps each year and Put option price via Jamshidian formula for different Expiries.

We can observe that the two methods give back very close results for all the Expiries considered, since the errors are of the magnitude of $o(10^{-6})$.
It can also be noticed that for any chosen Expiry the price of the European Swaption is lower than the Bermudan one: this result is expected, as Bermudan contracts add the right to enter in the Swap earlier or later with respect to the European options.

2.c Swaptions bounds via Jamshidian formula

For what stated before regarding the comparison between the European and the Bermudan swaptions, a natural lower bound for the Bermudan one is the maximum value of the European ones, having as maturities the early exercise dates. Regarding the upper bound, we have reasoned in term of rights, trying to find a strategy made of vanillas options which for sure dominates the Bermudan payoff: having the possibility to enter in the swap, if convenient, in all early exercise opportunity dates without having to decide if is more convenient to exercise or wait. The previous situation in terms of payoff could be obtained with a portfolio of European swaptions, so a possible upper bound for the Bermudan is the sum of the European ones. The two bounds obtained as described above are in Table 3 and we recognize that the actual value of the Bermudan swaption is closer to the lower limit, as the upper one is a loose estimate.

Lower bound	**Upper bound**
1.48028846%	8.86561372%

Table 3: Bermudan swaption bounds

2.d Evaluation of the error for different values of time steps N

In the sections above, we arbitrarily chose the number of yearly time steps $N = 100$ in the solution of the exercise. To better analyze the behaviour of the trinomial tree for different numbers of time steps, we varied N as the powers of 2 from 2^1 to 2^{10}.
In the following table, for these different values of N, we reported the Bermudan Swaption's price, the computational time, and the error between the Eruopean swaption prices, via the tree and via the Jamshidian formula taken as the norm infinity of the difference between the two prices.

N	Bermudan Swaption price	Computational time (s)	Error
2	2.22149536%	3.014830e-02	7.640079e-04
4	2.19945864%	3.128370e-02	3.610839e-04
8	2.18884445%	4.531670e-02	1.536373e-04
16	2.18392677%	7.305610e-02	9.709789e-05
32	2.18001084%	1.311704e-01	4.266934e-05
64	2.18049554%	2.057961e-01	2.759933e-05
128	2.17924063%	2.930145e-01	1.384819e-05
256	2.17918620%	7.414808e-01	7.568612e-06
512	2.17891462%	2.744593e+00	5.930288e-06
1024	2.17872516%	8.465992e+00	5.147493e-06

Table 4: Bermudan Swaption price, Computational time and Error for different values of N

From this Table we can immediately notice that the tree already computes a good value of the Bermudan Swaption price for low numbers of yearly time steps; obviously, as the number of nodes of the tree increases, the computational time increases and the error decreases.
To have a better understanding of these trends, we decided to plot the Bermudan Swaption price for different values of N in Figure 2 and the errors' behaviour in Figure 3.

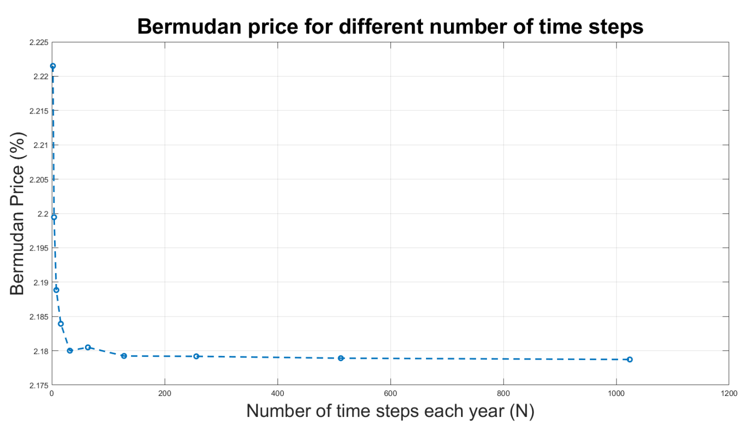

Figure 2: Bermudan Swaption price computed via the trinomial tree for different numbers of yearly time steps N

As we had noticed by observing Table 4, the Bermudan price obtained for N=32 is already an acceptable value: indeed, the Bermudan price plummets to a value around 2.18% as N reaches the value of 32 and then continues to decrease significantly slower. From this graph can also infer that our tree seems to converge from above to the "correct" price as N goes to infinity.

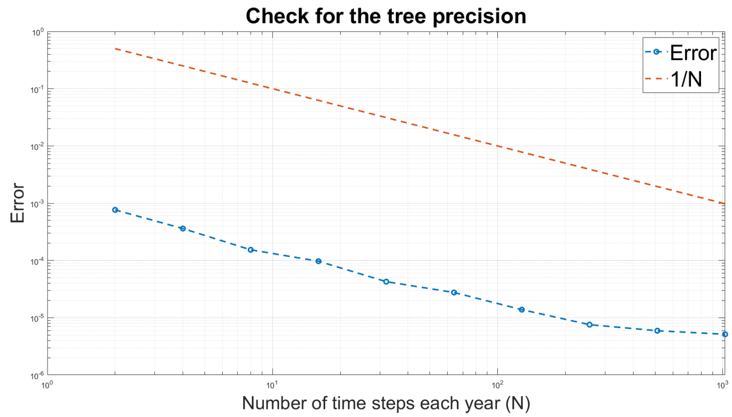

Figure 3: Maximum error for European swaptions w.r.t. number of yearly intervals, using the Jamshidian's prices as benchmark

As we can observe from Figure 3 the L^{∞} norm of the errors for the European swaption prices computed using the tree decreases linearly with the number of time interval for each year ensuring a precision of $10e{-}5$ from approximateltey 128 steps. As this error is acceptable for the estimate of the price, it is not necessary to increase the number of steps, avoiding an increase in computational time and possible storage issues.

References

Lewis, Alan L. (2001). "A simple option formula for general jump-diffusion and other exponential Lévy processes". In.