Analysis of Firm Default Probability and Credit Spread through Structural Models: A simulation-based approach

Simone Perfetto

January 31, 2021

Introduction

The objective of this essay is to model the expected default probability and credit spread of a firm that is financing itself partly through debt capital. In particular, I want to explore the impact of different modelling assumptions the to expected default probability and credit spread, using a simulation-based approach.

Default probability

In the context of credit risk, Structural Models link firm's default probability to the economics of the firm itself, i.e., its capital structure. Let me explore two famous models from the literature: Merton's and Black-Cox's.

Merton model

Merton model settings are loosely summarized as follows:

- Firm's value of assets is modeled as a stochastic process $\{Vt\}_{t\geq 0}$ that follows a geometric brownian motion
- Firm finances its assets through a mix of debt and equity. Specifically, the face value of debt to be extinguished at maturity T is D. Thus, at any point in time $t \leq T$, we have $V_t = B_t + E_t$ (where B_t stands for "bond" and E_t for "equity")
- Firm defaults only if at maturity $V_T < D$. If $V_T \geq D$, the firm is able to repay its debt.

Thus, the residual equity claims on firm's assets entitled to shareholders can be thought of as a European call option with strike price equal to the face value of debt: shareholders can keep everything in excess of such debt at maturity; if $V_T < D$, they do not "exercise" the call and opt for default. For a more rigorous discussion of Merton model, see Merton (1974).

Given the above setting, a practical approach to the estimation of default probability consists in the following:

- Perform a Monte-Carlo simulation of many "paths" (i.e., realized stochastic processes) for the value of the firm over a certain period of time T with sample frequency dt. The initial firm value for each path is V_0

- Estimate the default probability as the fraction of paths for which at maturity we have $V_T < D$, i.e., default as per above definition.

The following input parameters are used for the simulation of firm value's stochastic process according to Merton's model:

- $r = 3\%$
- $D = 100$ (face value of debt)
- $T = 20$ (years), $dt = 1/12$ (frequency expressed in years)
- $\sigma_v = 25\%$
- $V_0 = 200$

250,000 paths are simulated. The following code snippet implements the simulation.

```
import numpy as np
import pandas as pd
import numpy.random as rnd
import cufflinks as cf  # library to use plotly plots along with pandas
import plotly.graph_objs as go
from tqdm import trange
from scipy.stats import norm
cf.go_offline(connected=True)

def gbm_generator(v0: float, T: int, sigma: float,
                  r: float, dt=1/12, n_paths = 250_000):
    """
    :param v0: initial firm value
    :param T: number of years
    :param sigma: diffusion of GBM
    :param r: drift of GBM (risk-free rate)
    :param dt: inverse of frequency of observations per year (in years)
    :param n_paths: number of different paths to be simulated
    :return: a matrix of n=n_paths GBMs over T years sampled every dt year
    """
    n_periods = int(T/dt)       # total number of observations
    # create empty matrix with dimension (n_periods+1)*n_paths
    v = np.zeros((n_periods + 1, n_paths))
    v[0, :] = v0 * np.ones(n_paths)
    # simulate firm value path according to GBM model
    for i in trange(n_periods):
        rand_vec = np.random.normal(0, 1, n_paths)
        v[i+1, :] = v[i, :] * np.exp((r - sigma ** 2 / 2) * dt +
                                     sigma * (dt ** 0.5) * rand_vec)
    return v

np.random.seed(0)
v0, D, T, sigma, r,  = 200, 100, 20, 0.25, 0.03
dt = 1 / 12
n_paths, n_periods = 250_000, int(T/dt)

paths_data = gbm_generator(v0, T, sigma, r, dt, n_paths)
```

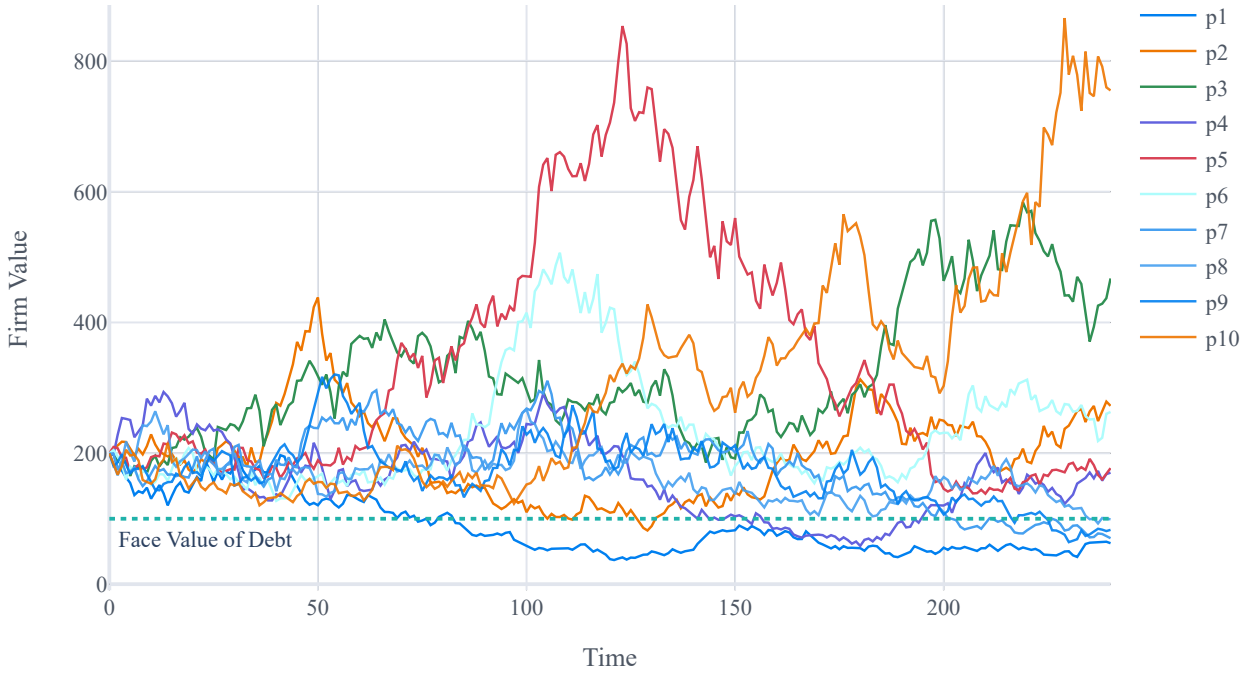
The following code snippet plots 10 randomly-chosen paths among the 250000.

```
# plot 10 randomly-chosen paths
rand_idx_plot = rnd.randint(low=0, high=n_paths-1, size=10)
paths_for_plot = paths_data[:, rand_idx_plot]
paths_for_plot_df = pd.DataFrame(paths_for_plot,
                                 columns=[f"p{i}" for i in range(1, 11)])

fig = paths_for_plot_df.iplot(colorscale="polar", xTitle="Time",
                              yTitle="Firm Value",
                              theme="white", asFigure=True)
fig.add_shape(type="line", line_color="LightSeaGreen", line_width=2,
              opacity=1, line_dash="dot", x0=0, x1=1, xref="paper", y0=D, y1=D)
fig.add_trace(go.Scatter(x=[23], y=[65], text=["Face Value of Debt"],
                         mode="text", showlegend=False)) # debt threshold label

fig.update_layout(font=dict(family="Computer Modern"))
fig.write_image("images/firm_value_simulated_paths.pdf", format="pdf")
fig.show()
```

Figure 1: Ten simulated paths for Firm Value process over 20 years (240 months)



Now we estimate the probability of default as the fraction of paths experiencing default at maturity (i.e., whose firm value V_T is below the threshold D). Thus:

```
res_v_maturity = paths_data[-1, :] - D
pd_merton = len(res_v_maturity[res_v_maturity < 0]) / len(res_v_maturity)
print("Estimated probability of default "
      "under Merton model settings:", round(pd_merton, 4))
>>> Estimated probability of default under Merton model settings: 0.2753
```

Black-Cox model

Merton model allows only for default at maturity, which is a non-realistic assumption. Indeed, bonds are usually issued with safety covenants and subordination arrangements as further protection tools for bondholders. Any breach of such agreements can trigger early default. This results in a more disciplined management of financial resources by shareholders, limiting their incentive in engaging in riskier projects. Black-Cox model (Black and Cox 1976) allows for a closer representation of reality, as it incorporate the possibility of early default at any point in time before maturity in case a certain threshold is hit by the value of firm's assets.

Differently from previous case, default threshold is now a deterministic function of time:

$$D(t) = De^{-r(T-t)}$$

Once clarified the different definition of "default", the probability of default can still be estimated as the fraction of defaulting paths. We use the paths simulated above and check their (possibly early) default status with the following code snippet, and then compute Black-Cox default probability:

```
# define default threshold considering progressive discount factor
discount_vec = np.array([np.exp(-dt * (i - 1) * r)
                         for i in range(n_periods + 1, 0, -1)])
Dt = D * discount_vec
# build array to check any default status in at least one period,  for each path
default_status = np.any(paths_data < Dt.reshape(-1, 1), axis=0)
pd_bc = len(default_status[default_status]) / len(default_status)
print("estimated probability of default "
      "under Black-Cox model settings is:", round(pd_bc, 4))
>>> Estimated probability of default under Black-Cox model settings: 0.4145
```

Probability of default in Black-Cox model is significantly higher than probability of default in Merton model. In fact, the allowance of default at any period up to maturity increases the chances that the firm value's stochastic process moves below the threshold and trigger default covenants. Thus, ceteris paribus, default is more likely in Black-Cox model than in Merton model.

Comparison of models

We might wonder what would be the capital structure for Black-Cox model that equalizes the probability of default in Merton model with D=100, ceteris paribus. The following code snippet implements a routine to find the answer:

```
def merton_minus_bc_squared(Dt_: np.array, pd_mert: float, paths: np.array):
    """
    :param Dt_: vector of discounted Face Value of Debt D
    :param pd_mert: reference default probability under Merton
    :param paths: matrix of simulated paths
    :return: squared error (p_mert - p_bc)^2
    """
    default_stat = np.any(paths < Dt_.reshape(-1, 1), axis=0)
    pd_bcox = len(default_stat[default_stat]) / len(default_stat)
    return (pd_mert - pd_bcox) ** 2
```

```
# GridSearch D between [0.25, 100] with 0.25 jumps to find
# min(merton_minus_bc_squared)
merton_minus_bc_squared_array = np.zeros(400)
for i in trange(400):
    D = (i + 1)/4
    Dt = D * discount_vec
    merton_minus_bc_squared_array[i] = \
        merton_minus_bc_squared(Dt_=Dt, pd_mert=pd_merton, paths=paths_data)

# show Face Value of Debt for which (p_merton - p_bc)^2 is minimum
merton_minus_bc_squared_df = pd.DataFrame(merton_minus_bc_squared_array,
                                          index=[(i+1) / 4 for i in range(400)])
print("Black-Cox - Face Value of Debt allowing equal"
      " default probabilities:", float(merton_minus_bc_squared_df.idxmin()))
>>> Black-Cox - Face Value of Debt allowing equal default probabilities: 70.25
```

Thus, a zero-coupon bond with face value equal to 70.25 would be enough in Black-Cox setting to achieve the same probability of default you would get with a 100-Face Value bond in Merton setting. We demonstrated how Black-Cox is much more conservative. As discussed above, Black-Cox is a closer representation of reality.

Credit spread

In the context of credit risk, the credit spread of a defaultable bond is the difference between the bond yield and the yield of an equivalent risk-free bond. It is a very important metric as it incorporates the premium required for bearing the additional default risk when investing in the defaultable bond.

For a bond with maturity T and constant risk-free rate, we have in symbols:

$$s_{t,T} = y_{t,T} - r$$

Now, using the same notation as above, we have:

$$V_t = B_t + E_t = B_t + Call_{BS}(V_t, D, r, T, \sigma_v)$$

where $Call_{BS}(.)$ represent the Black-Scholes option-like feature of shareholders' equity claims, with strike price equal to the face value of debt D. By definition, we know also that a zero-coupon bond yield is as follows:

$$B_t = De^{-y_{t,T} * (T-t)}, \quad \text{equivalent to:} \quad y_{t,T} = \frac{1}{T-t} \log\left(\frac{D}{B_t}\right) = \frac{1}{T-t} \log\left(\frac{D}{V_t - Call_{BS}(.)}\right)$$

Where we used the definition of firm's value of assets as equity plus debt and the call-option-like feature of shareholders equity on firm's asset value with strike K. Furthermore, we know that:

$$B_t = De^{-y_{t,T} * (T-t)}, \qquad \text{which is equivalent to:} \qquad y_{t,T} = \frac{1}{T-t} \log\left(\frac{K}{B_t}\right)$$

using the three above, we can easily derive the credit spread for various input parameters as:

$$s_{t,T} = y_{t,T} - r$$

We now explore credit spread within Merton model, using same baseline inputs as above, except for face value of debt:

- r = 3%
- σ_v = 25%
- V_0 = 200
- D = 150 (face value of debt)

However, we want to see how varying any of the first three baseline inputs (i.e., r, σ_v, V_0 affects the credit spread itself at different maturities.

The following code snippet creates a function to compute credit spreads by taking a matrix of simulated paths and other parameters as inputs. It also define baseline inputs to be used later.

```python
def credit_spread_simulator(paths, r: float, D: float, T, dt):
    """
    :param paths: matrix of simulated paths
    :param r: risk-free rate
    :param D: Face value of debt
    :param T: maturity
    :param dt:  inverse of frequency of observations per year (in years)
    :return: credit_spread
    """
    maturity_row = int(T / dt)
    payoffs_at_maturity = np.where(paths[maturity_row, :] - D >= 0,
                                   paths[maturity_row, :] - D, 0)
    # discounted expected payoff
    call = np.exp(-r * T) * np.mean(payoffs_at_maturity)
    v0 = paths[0, 0]
    b0 = v0 - call
    y0 = (1 / T) * np.log(D / b0)
    return y0 - r

# define base inputs
np.random.seed(0)
v0_base, D, T_base, sigma_base, r_base = 200, 150, 20, 0.25, 0.03
dt = 1 / 12
n_paths = 250_000
n_periods = int(T_base / dt)
maturities = np.linspace(start=dt, stop=20, num=240, dtype=np.float64)
```

Varying initial firm value

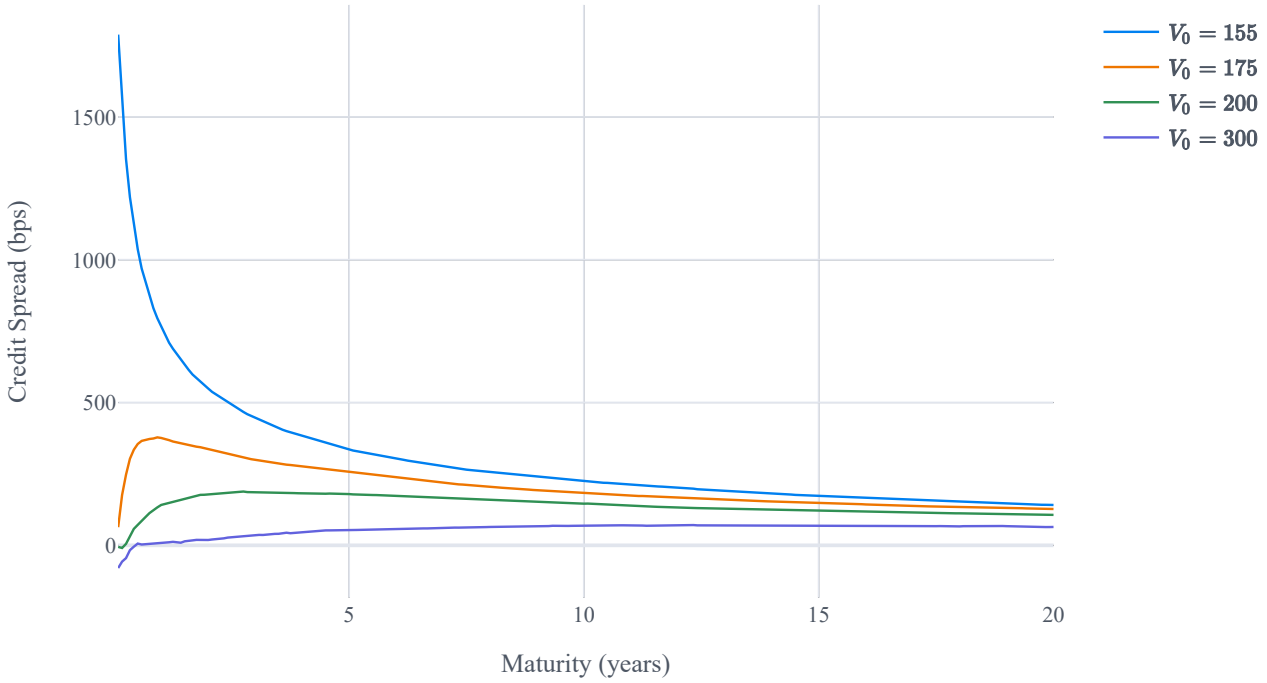
Now we focus on the different credit spreads for several maturities and varying initial firm value. The following code snippet computes credit spreads and show them in a plot.

```python
def spread_plotter(spread_data: pd.DataFrame, image_name: str):
    figure = spread_data.iplot(colorscale="polar", theme="white", asFigure=True,
                               yTitle="Credit Spread (bps)",
                               xTitle="Maturity (years)",)
    figure.update_layout(font=dict(family="Computer Modern"))
    figure.write_image("images/{}.pdf".format(image_name), format="pdf")
    figure.show()

firm_vals = [155, 175, 200, 300]
spread_matrix_varying_vals = np.zeros([len(maturities), len(firm_vals)])
for j, val in enumerate(firm_vals):
    paths_data = gbm_generator(v0=val, T=T_base, sigma=sigma_base,
                               r=r_base, dt=dt, n_paths=n_paths)
    for i, tau in enumerate(maturities):
        spread_matrix_varying_vals[i][j] = credit_spread_simulator(
            paths=paths_data, r=r_base, D=D, T=tau, dt=dt) * 10000

spread_df_varying_vals = pd.DataFrame(spread_matrix_varying_vals, index=maturities,
                                      columns=[f"$V_0 = {v}$" for v in firm_vals])
spread_plotter(spread_df_varying_vals, image_name="spreads_varying_value")
```

Figure 2: Credit Spread Curve for different starting Firm Values



Merton model allows both a monotonically decreasing and increasing spread curve (as shown in the plot for $V_0 = 300$ and $V_0 = 155$, respectively). Merton model allows also a humped-shape credit spread curve (for example, when $V_0 = 175$).

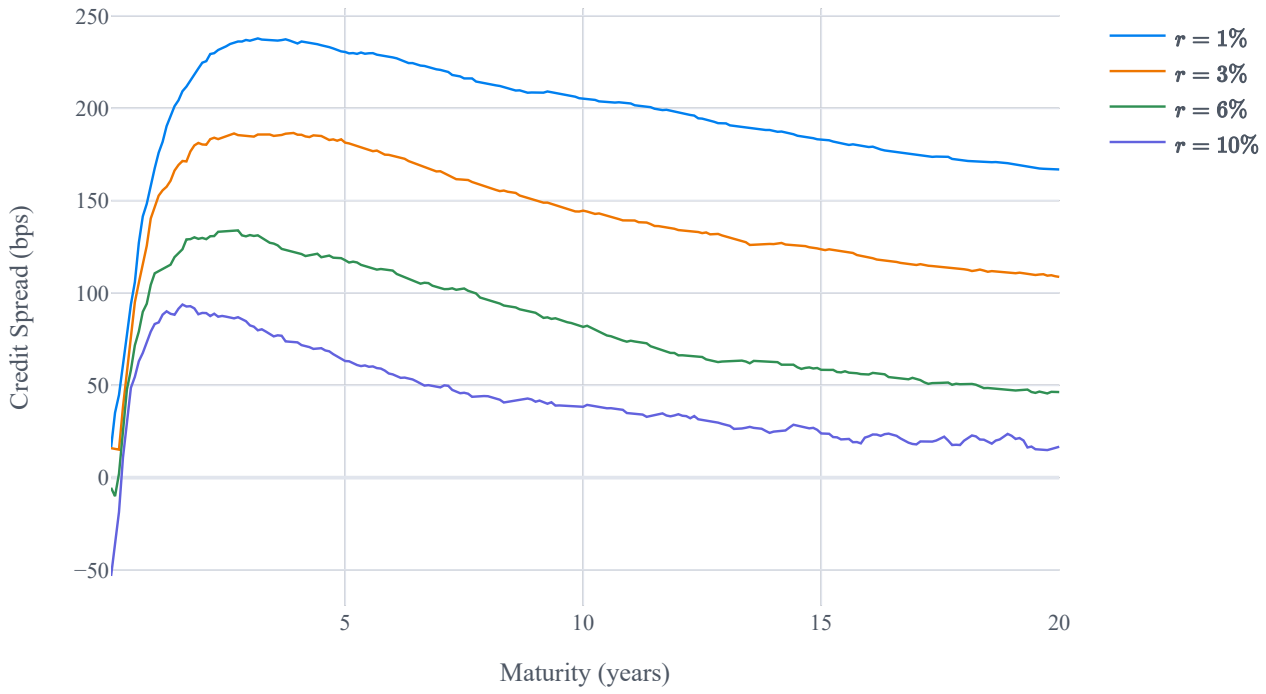
- The spread curve is monotonically decreasing when the firm is in extreme financial distress, i.e, when the current value of assets is very close or below the face value of debt. In that case, the probability of default for shorter maturities is very high and the large credit spread is a direct consequence. For longer maturities, however, credit spread is progressively decreasing. In fact, conditional on surviving, the firm might be able to bring the value of assets above the face value of debt before payment is due. The clear example is given in the plot for $V_0 = 155$.
- The spread curve is monotonically increasing when the firm has a solid financial structure, such that the probability of default for shorter maturities is 0 de facto. In other words, it is extremely unlikely that the value of assets will go below the face value of debt, which makes the credit spread approximately 0. However, for longer maturities, there is a higher chance for the value of the firm to end up in bad states of the world, thus the credit spread progressively increase with longer maturities. For instance, when $V_0 = 300$, the firm is very unlikely to default in the short term as the value of assets is significantly above the face value of debt.
- The spread curve has a humped shape when the firm is relatively healthy now, but might experience some financial distress in the medium term. The spread curve then decreases because, conditional to surviving until that point in time, firm's prospects might subsequently improve.

Varying risk-free rate

Now we focus on the different credit spreads for several maturities and varying risk-free rate. The following code snippet computes credit spreads and show them in a plot.

```
rates = [0.01, 0.03, 0.06, 0.1]
spread_matrix_varying_rates = np.zeros([len(maturities), len(rates)])
for j, rate in enumerate(rates):
    paths_data = gbm_generator(v0=v0_base, T=T_base, sigma=sigma_base,
                               r=rate, dt=dt, n_paths=n_paths)
    for i, tau in enumerate(maturities):
        spread_matrix_varying_rates[i][j] = credit_spread_simulator(
            paths=paths_data, r=rate, D=D, T=tau, dt=dt) * 10000
spread_varying_rates = pd.DataFrame(spread_matrix_varying_rates, index=maturities,
                                    columns=[f"$r = {int(rt * 100)} \%$"
                                             for rt in rates])
spread_plotter(spread_varying_rates, image_name="spreads_varying_rates")
```

Figure 3: Credit Spread Curve for different levels of Interest Rates



As shown in the plot and the table, low levels of interest rates shift the whole credit spread curve up. The interpretation of risk-free rate impact on spread is tricky, as there are two competing factors.

- On the one hand, when rates are high, the call-option-like shareholders' residual claim on assets is more valuable, as the expected strike price payment (i.e., face value of debt) in the future is discounted more heavily. This implies that the value of the bond B_t is less and bondholders require a higher yield $y_{t,T}$.
- On the other hand, the higher interest rate you subtract from the yield to obtain the spread

more than compensates the effect above, resulting in higher credit spreads when interest rates are lower.

For shorter maturities, the "discount effect" is limited though, so we observe relatively smaller spreads all levels of interest rates considered.

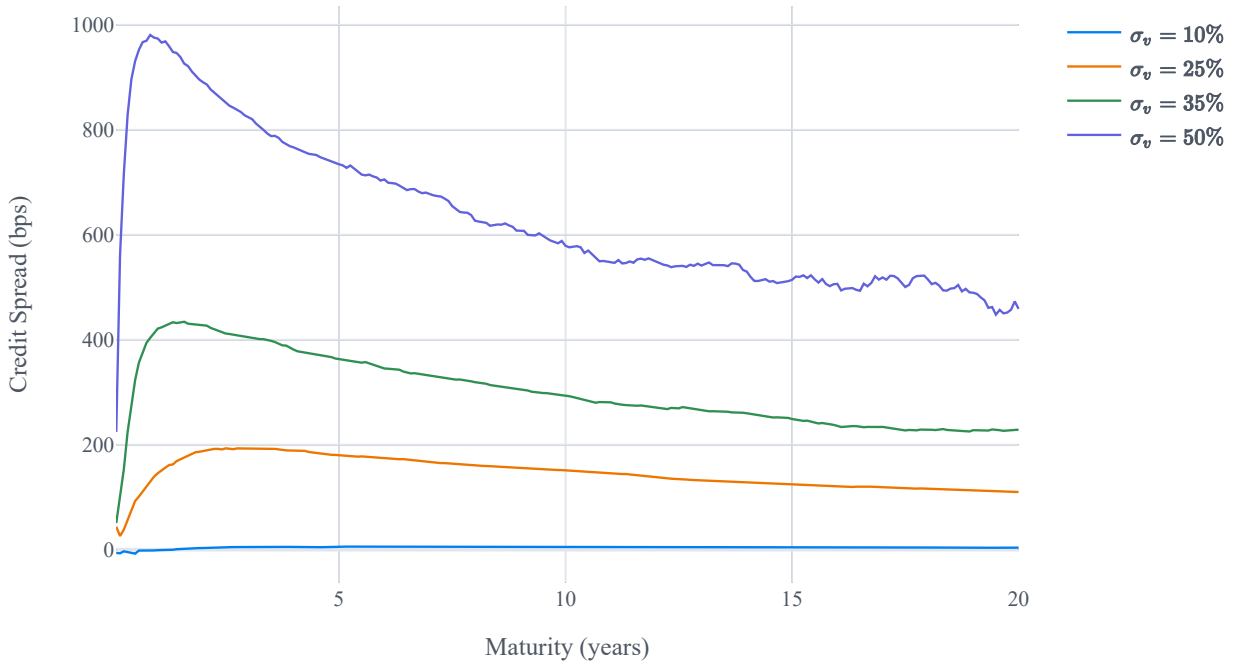
Varying volatility of firm value

Now we focus on the different credit spreads for several maturities and varying firm value volatility. The following code snippet computes credit spreads and show them in a plot.

```
sigmas = [.10, .25, .35, .50]
spread_matrix_varying_sigmas = np.zeros([len(maturities), len(sigmas)])

for j, sig in enumerate(sigmas):
    paths_data = gbm_generator(v0=v0_base, T=T_base, sigma=sig,
                               r=r_base, dt=dt, n_paths=n_paths)
    for i, tau in enumerate(maturities):
        spread_matrix_varying_sigmas[i][j] = credit_spread_simulator(
            paths=paths_data, r=r_base, D=D, T=tau, dt=dt) * 10000
spread_varying_sigmas = pd.DataFrame(spread_matrix_varying_sigmas, index=maturities,
                                     columns=[f"$\sigma_v = {int(sig * 100)}\%$"
                                              for sig in sigmas])
spread_plotter(spread_varying_sigmas, image_name="spreads_varying_vol")
```

Figure 4: Credit Spread Curve for different Firm's Asset Volatilities



As shown in the plot, higher volatility of firm's assets shifts the whole credit spread curve up. Moreover, the humped shape of the curve at shorter term maturities is more pronounced for more

volatile firms. Both effects can be explained from the call-option-like feature of equity.If the underlying is more volatile, the option is worth more due to its asymmetric pay-off structure. However, we also have $V_t = B_t + E_t$. Thus, if E_t is worth more because of higher volatility and V_t is kept constant, then the value of the bond B_t is less. For this reason, the bond is riskier and bondholders require a higher premium, which determines the higher credit spread, ceteris paribus.

Conclusion

Structural models constitute a fascinating tool to analyse firm's capital structure and the credit risk associated with the level of indebtedness. They provide a clear and understandable economic intuition on probabilities of default and credit spread. The simulation approach just shows how powerful and economically-coherent conclusions can be drawn by using these models. Such models can be enhanced by relaxing some strict assumptions and should represented the cornerstone for any business involved in corporate lending and credit risk assessment.

References

Black, Fischer and John Cox (1976). "Valuing Corporate Securities: Some Effects of Bond Indenture Provisions". In: *The Journal of Finance* 31.2. ISSN: 00221082. DOI: 10.2307/2326607.

Merton, Robert C. (1974). "On the Pricing of Corporate Debt: The Risk Structure of Interest Rates". In: *The Journal of Finance* 29.2, pp. 449–470. ISSN: 00221082. DOI: 10.2307/2978814.