Credit curve construction

Othmane ZARHALI

The aim of the present is to challenge several credit spread curve construction methods. The reason behind those methods is that credit market data quotes are discrete as a result we need to learn efficiently the spread curves from those quotes.

1 Problem setting

Let's consider the following toy context : Be a sample $(X_1, X_2, ..., X_n)$ of a continuous random variable with a density function with respect to the lebesgues measure used to explain another random variable Y.
It's obvious that that best $\mathbb{L}^2$ approximation of Y given X is the conditional expectation function $f(x) = \mathbb{E}(Y/X = x)$ also called the regression function.

In fact, we need to estimate f with $\hat{f}$ such that the RMSE error is minimal in the sample set.

$$RMSE := \mathbb{E}\left(\left(f(X) - \hat{f}(X)\right)^2\right) \tag{1}$$

Different approximation methods will be performed based on the interpolation of the CDS spread quotes.

2 Interpolation methods

2.1 Polynomial spline interpolation

Indeed, this is the most intuitive approach to estimate the latter regression function. We choose the following spline polynomial basis as interpolating functions :

— **Constant piece wise spline** :

$$\begin{cases} h_1(x) = 1_{x \leq k_1} \\ \forall k \in [\![1; M]\!], h_i(x) = 1_{k_i < x \leq k_{i+1}} \end{cases} \tag{2}$$

Where M is a fixed integer and $(k_1, ..., k_M)$ a fixed M-uplet called the real line partition node.

— **Piecewise polynomial spline** :

$$\begin{cases} h_i(x) = x^i, \forall i \in [\![1; \tilde{M}]\!] \\ h_j(x) = \left((x-k_j)_+\right)^{\tilde{M}-1}, \forall j \in [\![\tilde{M}+1, L]\!] \end{cases} \tag{3}$$

Where $\tilde{M}$ and L are fixed integers and $\left(k_{\tilde{M}+1}, ..., k_L\right)$ a kernel.

— **Basic spline polynomials** :

$$\begin{cases} h_1(x) = 1 \\ h_2(x) = x \\ h_{j+2}(x) = \frac{\left((x-k_l)_+\right)^3 - \left((x-k_L)_+\right)^3}{k_L - k_l} - \frac{\left((x-k_L)_+\right)^3 - \left((x-k_{L-1})_+\right)^3}{k_L - k_{L-1}}, \forall j \in [\![1, L]\!] \end{cases} \tag{4}$$

The kernel choice is not obvious but one way to guess it is to mimick the random variable sample or its empirical quantile.

2.2 Kernel method

Définition 1 *Let K be a unidimensional function. We claim that K is a kernel if the three conditions are verified :*

— *K is a positive bounded continuous function over $\mathbb{R}$.*
— *K is compactly supported and $\int_{\mathbb{R}} K(t)dt = 1$.*
— *$\int_{\mathbb{R}} tK(t)dt = 0$ et $\int_{\mathbb{R}} t^2K(t)dt$ is not vanishing.*

Depending on the model, we can choose between several kernel types :

— The gaussian kernel :

$$K(t) = \frac{1}{\sqrt{2\pi}} exp(-\frac{t^2}{2}) \tag{5}$$

— The Epanechnikov kernel :

$$K(t) = (1 - \frac{t^2}{h^2})1_{\{\frac{t}{h}<1\}} \tag{6}$$

— The triweight kernel :

$$K(t) = \frac{105}{48}\left((1-t^2)^3\right)_+ 1_{\{t<1\}} \tag{7}$$

— The normalised triweight kernel :

$$K(t) = \frac{105}{48}\left(\left(1 - \left(\frac{t}{h}\right)^2\right)^3\right)_+ 1_{\{\frac{t}{h}<1\}} \tag{8}$$

h is bandwidth parameter that can be estimated afterwards.

2.2.1 Intuition behind the kernel regression function estimation

The insight comes from the bayes formula :

$$f(x) = \mathbb{E}(Y/X = x) = \frac{\mathbb{E}(Y\delta_X(x))}{\mathbb{E}(\delta_X(x))}$$

Where $\delta_X(x)$ is the Dirac impulsion on x. If X is an absolute continous random variable then the expected Dirac impulsion on x is nothing but the density function evaluated on x.
The reason is that we seek a suitable regularisation of the Dirac impulsion by a kernel function :

$$\mathbb{E}(\hat{\delta}_X(x)) = \sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right) \tag{9}$$

Thus, we obtain :

$$\mathbb{E}(Y\hat{\delta_X}(x)) = \sum_{i=1}^{n} Y_i K\left(\frac{X_i - x}{h}\right) \tag{10}$$

The latter is the so called **Nadaraya Watson** estimator :

$$\hat{f}(x) = \sum_{i=1}^{n} \frac{K\left(\frac{X_i - x}{h}\right)}{\sum_{j=1}^{n} K\left(\frac{X_j - x}{h}\right)} Y_i \tag{11}$$

2.3 Local polynomial estimators

If we assume additional regularity conditions on the regression function (ie C^p with a sufficiently higher order p).
Thus, we have the following expansion of order p around any $x^{'}$:

$$f(x) \approx \sum_{i=0}^{p} \frac{(x - x')^i}{i!} f^{(i)}(x') \tag{12}$$

By considering $\beta_i = \frac{f^{(i)}(x')}{i!}$ with $i = 0, ..., p$, we have :

$$f(x) \approx \sum_{i=0}^{p} \beta_i (x - x')^i =: \hat{f}(x) \tag{13}$$

We then estimate the regression function with a polynomial of order p called the **local polynomial estimator of order** p .
The coefficient β_i are estimated using a least square minimisation approach.

2.3.1 Construction of the local polynomial estimators

The aim of this subsection is to estimate $(\beta_i)_{i \in \{1,...,p\}}$.
Given a well chosen kernel K, we have :

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \sum_{i=1}^{n} \left(\left(Y_i - \sum_{i=0}^{p} \beta_i (X_i - x)^i \right)^2 K\left(\frac{X_i - x}{h}\right) \right) \tag{14}$$

We claim the following property as well :

Proposition 1 *The local polynomial estimator of order 0 is the Nadaraya Watson estimator.*

Preuve 1 *The first step is to transform the optimisation problem (14) into a standard quadratic one.*

Let's introduce the notations :

- $D = diag\left(K\left(\frac{X_1-x}{h}\right), K\left(\frac{X_2-x}{h}\right), ..., K\left(\frac{X_n-x}{h}\right)\right)$
  the diagonal matrix with coefficients : $K\left(\frac{X_1-x}{h}\right), K\left(\frac{X_2-x}{h}\right), ..., K\left(\frac{X_n-x}{h}\right)$.

- *The matrix* X *:*

$$\begin{pmatrix} 1 & (X_1-x) & (X_1-x)^2 & ... & (X_1-x)^p \\ 1 & (X_2-x) & (X_2-x)^2 & ... & (X_2-x)^p \\ . & . & . & ... & . \\ . & . & . & ... & . \\ . & . & . & ... & . \\ 1 & (X_n-x) & (X_n-x)^2 & ... & (X_n-x)^p \end{pmatrix}$$

- *Soit* Y *:*

$$\begin{pmatrix} Y_1 \\ Y_2 \\ . \\ . \\ . \\ Y_n \end{pmatrix}$$

Thus, the optimisation problem is equivalent to :

$$\underset{\beta \in \mathbb{R}^{p+1}}{min} (Y - X\beta)^t D(Y - X\beta) \tag{15}$$

The latter has a closed form solution :

$$\hat{\beta} = \left(X^t DX\right)^{-1} X^t DY \tag{16}$$

Now, we should compute $\hat{\beta}_0$ *:*

The first component of $X^t DY$ *is* $\sum_{i=1}^n Y_i K\left(\frac{X_i-x}{h}\right)$.
We denote :

- $\lambda_1 = (X_1 - x)$, $\lambda_2 = (X_2 - x)$,..., $\lambda_p = (X_p - x)$
- $\alpha_1 = K\left(\frac{X_1-x}{h}\right)$, $\alpha_2 = K\left(\frac{X_2-x}{h}\right)$,..., $\alpha_p = K\left(\frac{X_p-x}{h}\right)$

After computations, we get :

$$X^t DX = \begin{pmatrix} \sum_{i=1}^n \alpha_i & \sum_{i=1}^n \alpha_i\lambda_i & \sum_{i=1}^n \alpha_i\lambda_i^2 & ... & \sum_{i=1}^n \alpha_i\lambda_i^p \\ \sum_{i=1}^n \alpha_i\lambda_i & \sum_{i=1}^n \alpha_i\lambda_i^2 & \sum_{i=1}^n \alpha_i\lambda_i^3 & ... & \sum_{i=1}^n \alpha_i\lambda_i^p \\ . & . & . & ... & . \\ . & . & . & ... & . \\ . & . & . & ... & . \\ \sum_{i=1}^n \alpha_i\lambda_i^p & \sum_{i=1}^n \alpha_i\lambda_i^{p+1} & \sum_{i=1}^n \alpha_i\lambda_i^{p+2} & ... & \sum_{i=1}^n \alpha_i\lambda_i^{2p} \end{pmatrix}$$

The case $p = 0$ corresponds to $\left(X^t DX\right)^{-1}$ as the scalar $\frac{1}{tr(D)}$ where $tr(D)$ is the trace matrix operator. We conclude that $\hat{\beta}_0$ is the Nadaraya Watson estimator.

2.4 Bandwidth estimation

We claim the following limit theorem :

Théorème 1 *Given a kernel K.*
We suppose that the regression function f is at least C^3 regular.
Be p the marginal distribution of X and n the sample size.
For a given bandwidth h, we denote the estimator $\hat{f}_{h,n}$ of f.
for any fixed x we have the convergence in distribution :

$$\sqrt{nh}\left(\hat{f}_{h,n}(x) - f(x) - \frac{1}{2}h^2\sigma_0^2 f''(x)\right) \longrightarrow N\left(0, p^{-1}(x)\sigma^2(x)\int K^2(t)dt\right)$$

Where :

- $\sigma_0 = \int t^2 K^2(t)dt$
- $\sigma^2(x) = Var(Y/X = x)$

We will extract the right h by minimising the MSE :

Définition 2 *We define the MSE (Minimum Squared Error) between f et $\hat{f}_{h,n}$ as follows :*

$$MSE = \mathbb{E}\left(\left(\hat{f}_{h,n}(x) - f(x)\right)^2\right) \tag{17}$$

Preuve 2 *The Nadaraya Watson estimator has the form :*

$$\hat{f}(x) = \frac{1}{nhp(x)}\sum_{i=1}^{n} Y_i K\left(\frac{X_i - x}{h}\right) \tag{18}$$

Using the second order expansion of the regression function :

$\hat{f}_{h,n}(x) - f(x) = \frac{1}{nhp(x)}\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)\left[Y_i - f(x) - f'(x)(X_i - x) - \frac{1}{2}f''(x)(X_i - x)^2\right]$

Thus, the asymptotic variance of the latter quantity is :

$\left(\frac{1}{2}h^2\sigma_0^2 f''(x)\right)^2 + \frac{1}{nhp(x)}\sigma^2(x)\int K^2(u)du$

By minimising the MSE as a h functional, we obtain the asociated normal equation :

$\frac{3}{4}h^3\left(\sigma_0^2 f''(x)\right)^2 - \frac{1}{nh^2p(x)}\sigma^2(x)\int K^2(t)dt = 0$
Which is equivalent to :
$\frac{3}{4}h^5\left(\sigma_0^2 f''(x)\right)^2 = \frac{1}{np(x)}\sigma^2(x)\int K^2(t)dt$

Thus, the optimal bandwidth is of the form :

$$h_{opt} = \frac{1}{n^{\frac{1}{5}}}\left(\frac{\sigma^2(x)\int K^2(t)dt}{\frac{3}{4}p(x)\sigma_0^4\left(f''(x)\right)^2}\right)^{\frac{1}{5}} \tag{19}$$

Indeed, h is a function of x. For simulation purposes, we consider $h_{opt} = \frac{1}{n^{\frac{1}{5}}}$.

3 CDS spread curve construction by interpolation

The idea here is to apply what has been presented before to construct the theoretical CDS spread curves of a floating CDS leg with a defaut time assumed to be ecponentially distributed with parameterλ. Furthermore, we suppose the interest rates to be constants.

The CDS spread of maturity T has the following closed formula :

$$s_T = \frac{\mathbb{E}\left((1-R)\,1_{\tau \le T}\right)}{\mathbb{E}\left(\sum_{T_i} D(0,T_i)\delta_{T_i} 1_{T_i \le \tau}\right)} \tag{20}$$

Where δ_{T_i} is the time increment : $T_{i+1} - T_i$.

We assume the recovery rate R to be deterministic, we have :

$$s_T = \frac{(1-R)\left(1 - e^{-\lambda T}\right)}{\sum_{T_i} D(0,T_i)\delta_{T_i} e^{-\lambda T_i}} \tag{21}$$

As a result, the CDS spread curve has a term structure and a closed formule.
We fix the following parameters :

- $\lambda = 0.4$
- $r = 1\%$ (taux sans risque)
- $R = 70\%$

Now it's time for the numerical experiments.

3.1 Spline interpolation

The toy spline interpolation is indeed the linear one.

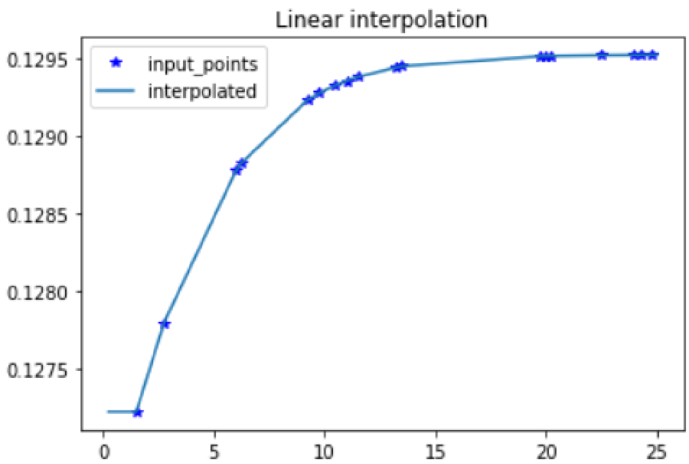

Figure 1 – Linear interplation

We test here with higher order splines :

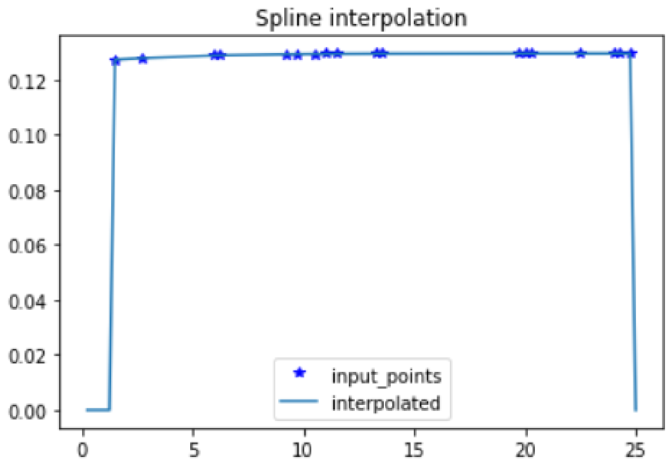

Figure 2 – Order 1 spline interpolation

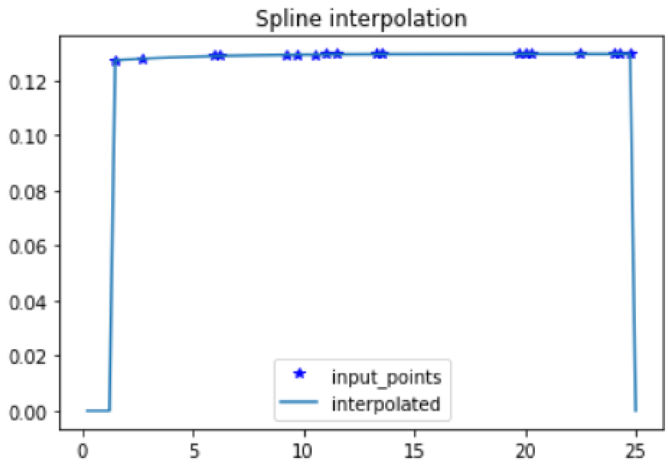

Figure 3 – Order 2 spline interpolation

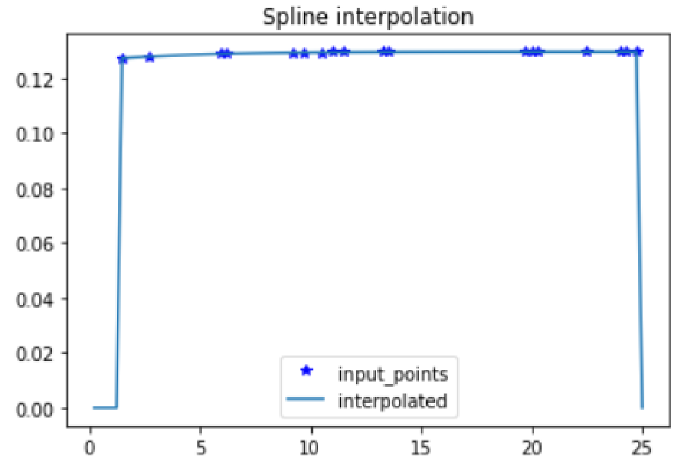

FIGURE 4 – Order 3 spline interpolation

3.2 Newton polynomial interpolation

The Newton polynomials are sufficiently smooth.
We recall their definitions :

Définition 3 *The newton polynomial basis is of the form :*

$$n_j(x) = \Pi_{i<j}\left(x - x_i\right), j \in \{0, ..., k\} \tag{22}$$

Thus, the interpolated function is :

$$N(x) = \sum_{j=1}^{k} a_j n_j(x), a := (a_1, ..., a_k) \in \mathbb{R}^k \tag{23}$$

Where k is the interpolation order.

Here is the numerical construction :

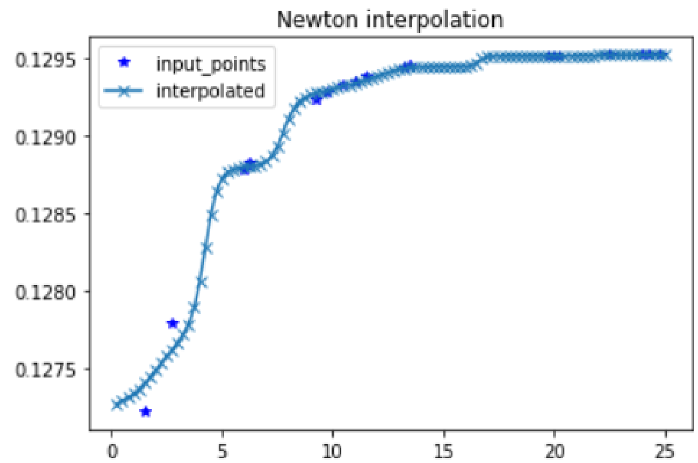

FIGURE 5 – Newton interpolation

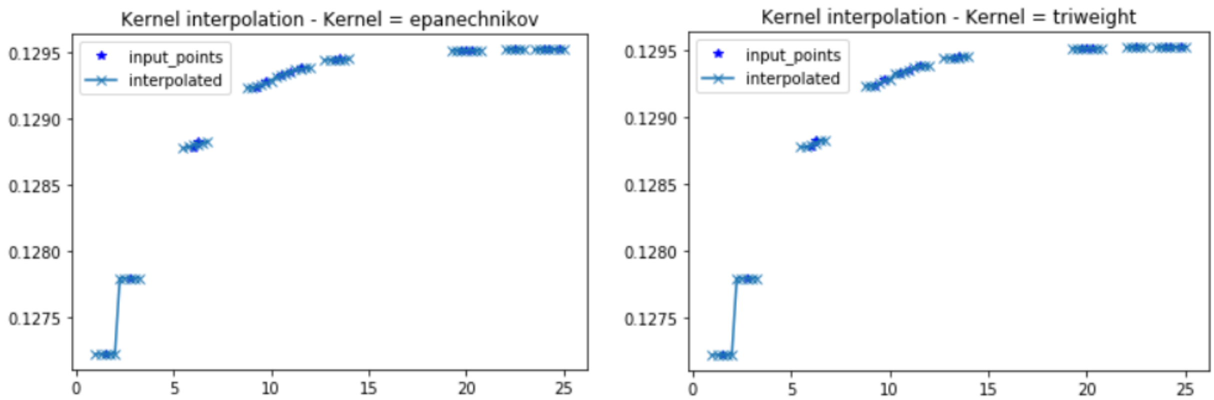

Figure 6 – Non regular kernel interpolation

3.3 MSE curves

The MSE profil is one of the vanilla interpolation criterion :

Type d'interpolation	MSE
Linéaire	1.0853554376500203e-06
Spline d'ordre 2	0.0009997485544122494
Spline d'ordre 3	0.0009997483383308015
Newton	1.0803816153744061e-06

Consequently, we deduce that the Newton interpolation is smoother thanks to the smoothness of the newton basis.

3.4 Kernel interpolation

We consider the following kernels for the numericla experiments :

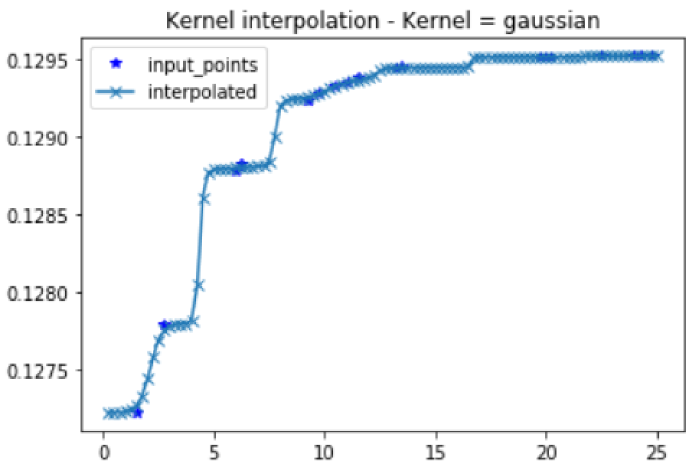

Figure 7 – Gaussain kernel interpolation

The importance of the smoothness criterion appears also in the kernel interpolation. Moreover, the gaussian kernel has the best MSE interpolation value of 9.72634202162033e-07 which is better than the previous polynomial interpolations.

In addition to that, the interpolaton quality depends also on the density of the spread cloud. If we consider a weak spread term structure with few cotations the construction will underfit too much the underlying curve.

3.5 Local polynomial interpolation

Before exposing the numerical results, we ecpose some preliminary remarks :

— This interpolation approach depend also on the chosen kernel.
— Depending on the chosen kernel, the matrix X^tDX can be singular because of too much vanishing values in its diagonal.
— The matrix X^tDX may not be invertible depending on the expansion order and the kernel.
— The key underlying assumption of this interplolation is the local expansion. Thus, if the spread cotations are too close to each others we have an overfitting risk.

Here are the numerical results with the gaussian kernel.

The MSE profile was computed as well :

Locla polynomial degree	**MSE**
0	9.938801005265504e-07
1	1.3277375272289652e-06
2	8.484160080421e-07
3	0.0003265450721439385
4	80645577.7844475

We see that above the degree 3 we start to overfit the CDS spread curve. The Nadaraya-Watson estimator is reproduced with a 0 degree local polynomial interpolation.

Here are the graphs :

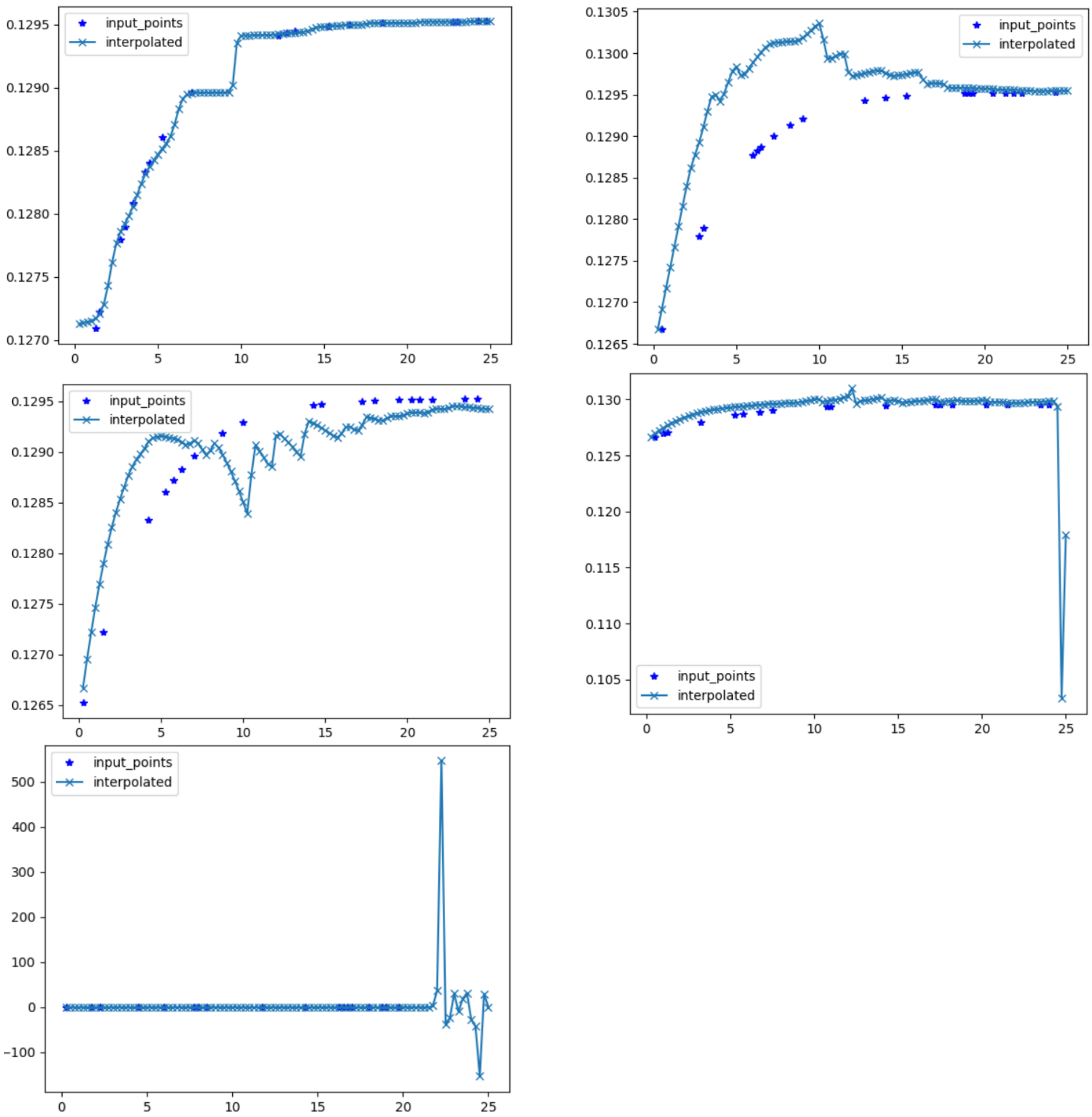

FIGURE 8 – Local polynomial interpolation - degree 0 to 4

4 Bootstrapping the issuer survival curve

We aim this time to construct the survival term structure $(Q(T_i))$ given the CDS spread term structure.

Here is the associated bootstrapping algorithm :

1. CDS spread cotation interpolation.
2. Survival vector initialisation.
3. **Iteration n** : Compute $Q(T_n)$ given $(Q(T_1), Q(T_2), .., Q(T_{n-1}))$ by inverting the following formula :
$$s_{T_n} = \frac{(1-R)\,(1-Q(T_n))}{\sum_{T_i \le T_n} D(0,T_i)\delta_{T_i} Q(T_i)} \tag{24}$$
4. Stop when all the CDS spread cotations are used.

We obtain the survival curve :

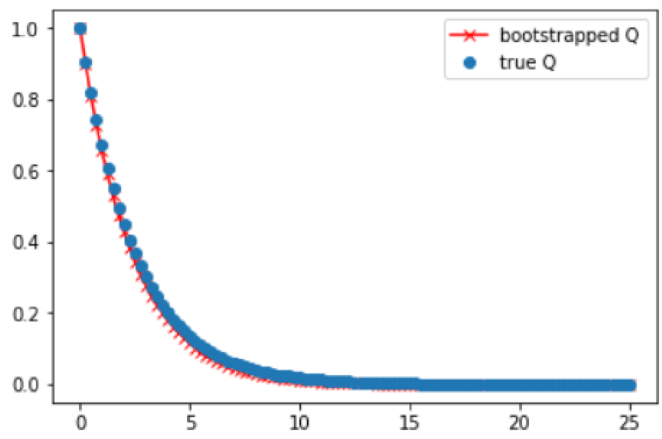

FIGURE 9 – Survival curve construction

Thus, this method is arbitrage free (non increasing constructed survival curve). Furthermore, with only a linear CDS spread interpolation we perform an MSE of 0.00011876208283351839.

The key interest of the survival curve construction is to be able to compare the default of different issuers. The CDS is a suitable credit hedging instrument for exotic non linear products.

5 Conclusion

Several approaches exist to learn the CDS spread curve. One way to benefit from it is to input it to deduce the survival probability curve of the issuer outputed from the bootstrap.
A linear interpolation of the CDS spreads can be considered for the latter purpose. However, more regular interpolation approaches should be used to avoid the error propagation such as Nadaraya Watson or local polynomial ones with a more accurate calibrated bandwidth parameter.

Références

[1] O'Kane, *Modelling single-name and Mutli-name Credit Derivatives*, 2008