

TODO

1. $PDE()$: boundary values of B at lower-right corner, when $i=200$. not sure why $X(100)$ (mid value on X grid) is used, not $X(200)$ (final value on X grid)
2. not sure why $\ln(\text{deriv})$ is part of the likelihood function. deriv measures the slope, $\frac{\text{change in CDS spread}}{\text{change in default intensity value}}$. Since default intensity is evenly spaced out on the grid, deriv essentially measures the increase in CDS spread as default probability increases. If this is part of the likelihood function, then it seeks to find parameters that yields model CDS spread within the interval with the largest increase in CDS spread over each additional increase in default intensity (?).

CDS

CDS spread is how much the protection buyer pays periodically (assuming \$1 face value) in exchange for the recovery of losses from the protection seller in the event of a default.

Protection leg

The present value of receiving the cash amount C at time τ is given the expectation:

$$\mathcal{C}(0, T) = E \left[C(\tau) e^{-\int_0^\tau r(t) dt} \mathbf{1}_{\tau \leq T} \right]$$

We wish to express $\mathcal{C}(0, T)$ in terms of the credit intensity process. Consider first a contract that pays $C(s)$ if the default takes place in time small interval $[s, s + ds]$. The value of this cash flow is

$$E \left[C(s) e^{-\int_0^s r(t) dt} \mathbf{1}_{\tau \in [s, s+ds]} \right]$$

We can rewrite it as

$$E \left[C(s) e^{-\int_0^s r(t) dt} \lambda(s) e^{-\int_0^s \lambda(t) dt} \right] = E \left[C(s) \lambda(s) e^{-\int_0^s (r(t) + \lambda(t)) dt} \right]$$

Integrating over s from 0 to T we find that

$$\mathcal{C}(0, T) = E \left[\int_0^T C(s) \lambda(s) e^{-\int_0^s (r(t) + \lambda(t)) dt} ds \right].$$

In Pan-Singleton's notation:

Protection leg:

$$(1 - R^Q) \int_t^{t+M} E_t^Q \left[\lambda_u^Q e^{-\int_t^u (r_s + \lambda_s^Q) ds} \right] du$$

Premium leg

Consider now the premium leg of a CDS maturing on T , with the premium consisting of the periodic coupon payments only (no upfront fee). Assume we are on a coupon payment date. The PV of a \$1 paid on T_j is given by the risky discount factor

$$\begin{aligned}\mathcal{P}(0, T_j) &= \mathbb{E} \left[e^{-\int_0^{T_j} r(s) ds} \mathbf{1}_{\tau > T_n} \right] \\ &= \mathbb{E} \left[e^{-\int_0^{T_j} (r(s) + \lambda(s)) ds} \right]\end{aligned}$$

In Pan-Singleton's notation:

The premium leg of a CDS contract with semi-annual premium payments is

$$\frac{1}{2} CDS_t(M) \sum_{j=1}^{2M} E_t^{\mathbb{Q}} \left[e^{-\int_t^{t+.5j} (r_s + \lambda_s^Q) ds} \right]$$

Intuitively, the periodic payments are further discounted by the default intensity λ .

CDS spread equation

Setting equal the present values of the premium leg and the protection leg, CDS spread satisfies the following equation:

$$CDS_t(M) = \frac{2(1 - R^Q) \int_t^{t+M} E_t^Q \left[\lambda_u e^{-\int_t^u (r_s + \lambda_s) ds} \right] du}{\sum_{j=1}^{2M} E_t^Q \left[e^{-\int_t^{t+j/2} (r_s + \lambda_s) ds} \right]}$$

Assume independent r_t and λ_t , we have

$$CDS_t(M) = \frac{2(1 - R^Q) \int_t^{t+M} D(t, u) E_t^Q \left[\lambda_u e^{-\int_t^u \lambda_s ds} \right] du}{\sum_{j=1}^{2M} D(t, t + j/2) E_t^Q \left[e^{-\int_t^{t+j/2} \lambda_s ds} \right]}$$

Cast above equation into

$$CDS_t(M) := (1 - R^Q) \frac{\text{SUM2}}{\text{SUM1}}$$

where

$$\begin{aligned}\text{SUM2} &= 2 \int_t^{t+M} D(t, u) E_t^Q \left[\lambda_u e^{-\int_t^u \lambda_s ds} \right] du \\ \text{SUM1} &= \sum_{j=1}^{2M} D(t, t + j/2) E_t^Q \left[e^{-\int_t^{t+j/2} \lambda_s ds} \right]\end{aligned}$$

Claim: SUM2 and SUM1 only differ in terms of $g(\cdot)$ in expectations (λ in SUM2, 1 in SUM1), under suitable choice of discretizing the integral in SUM2.

Explanation: consider SUM2, let $du = \frac{1}{2}$, then $u = t + \frac{1}{2}j$, where j goes from 1 to $2M$. Then we have

$$\begin{aligned}
\text{SUM2} &= 2 \int_t^{t+M} D(t, u) E_t^Q \left[\lambda_u e^{-\int_t^u \lambda_s ds} \right] du \\
&\approx 2 \left(\sum_{j=1}^{2M} D(t, t + j/2) E_t^Q \left[\lambda_u e^{-\int_t^{t+j/2} \lambda_s ds} \right] \right) \frac{1}{2} \\
&= \sum_{j=1}^{2M} D(t, t + j/2) E_t^Q \left[\lambda_u e^{-\int_t^{t+j/2} \lambda_s ds} \right]
\end{aligned}$$

See conditional statement in line 74-80 in `PDE()` that takes care of different $g(\cdot)$ in $E(\cdot)$

Derive PDE for $E(\cdot)$ via Feynman-Kac

Feynman-Kac:

Let X be an Ito process such that

$$dX_t = b(X_t) dt + \sigma(X_t) dW_t$$

Let $v : \mathbb{R}^K \times [t_0, t] \rightarrow \mathbb{R}$ by

$$v(x, t) = E \left[e^{-\int_t^t \varphi(X_s) ds} g(X_t) \mid X_0 = x \right]$$

where g and φ are a twice continuously differentiable and continuous lower bounded functions respectively. Under technically suitable conditions in (Feynman-Kac) Theorem 8.2.1 in Oksendal (2000), v uniquely solves

$$v_t(x_i, t_j) - v_x(x_i, t_j) b(x_i) - \frac{1}{2} \left[\sigma(x_i)^2 v_{xx}(x_i, t_j) \right] + \varphi(x_i) v(x_i, t_j) = 0$$

with initial condition $v(x, t_0) = g(x)$.

Finite difference:

With the following choice of finite difference schemes, we can reproduce the same scheme used in the fortran code:

$$\begin{aligned}
v_t(x_i, t_j) &\approx \frac{v_{i,j} - v_{i,j-1}}{\Delta t} \\
v_x(x_i, t_j) &\approx \frac{v_{i+1,j} - v_{i-1,j}}{2\Delta x} \\
v_{xx}(x_i, t_j) &\approx \frac{v_{i+1,j} - 2v_{i,j} + v_{i-1,j}}{(\Delta x)^2}
\end{aligned}$$

Plugging these approximations back to the PDE right above, we have

$$\frac{v_{i,j} - v_{i,j-1}}{\Delta t} - \frac{v_{i+1,j} - v_{i-1,j}}{2\Delta x} b(x_i) - \frac{1}{2} \sigma(x_i)^2 \left(\frac{v_{i+1,j} - 2v_{i,j} + v_{i-1,j}}{(\Delta x)^2} \right) + \varphi(x_i) v_{i,j} = 0$$

Rearrange to have $v_{i,j-1}$ term on the right, and the rest on the left, so that it's clear to see how the update depends on the value at previous point in time:

$$\left(\frac{\sigma^2(x_i)}{2(\Delta x)^2} - \frac{b(x_i)}{2\Delta x} \right) v_{i-1,j} + \left(-\frac{1}{\Delta t} - \frac{\sigma^2(x_i)}{2(\Delta x)^2} - \varphi(x_i) \right) v_{i,j} + \left(\frac{\sigma^2(x_i)}{2(\Delta x)^2} + \frac{b(x_i)}{2\Delta x} \right) v_{i+1,j} = -\frac{1}{\Delta t} v_{i,j-1}$$

In Pan-Singleton's notation:

Recall the dynamics of λ is

$$\begin{aligned} d \ln \lambda_t &= \kappa (\theta - \ln \lambda_t) dt + \sigma_\lambda dB_t \\ &= (\kappa \theta - \kappa \ln \lambda_t) dt + \sigma_\lambda dB_t \\ &:= (\text{ALP} - \text{BET} \ln \lambda_t) dt + \text{SIG} dB_t \end{aligned}$$

the last equality comes from the variable name defined in the fortran code. The mapping between two set of notations is

$$\begin{aligned} b(x_i) &= (\text{ALP} - \text{BET} \ln x_t) \\ \sigma^2(x_i) &= \text{SIG}^2 \end{aligned}$$

Plugging the mapping into the finite difference scheme, one can check this is equivalent to fortran/python implementation. See line 94-96 in `PDE()` .

Note: Boundary value. At first and last grid points, only two values of v are available, which cannot form an approximation of second order derivative. Therefore, as in line 88-91 in `PDE()` , second order derivative terms are dropped. However, in specifying end values of `B` , it's not clear why `X[99]` is used, as opposed to `X[199]` . For more, see TODO at front of page.

Discounting function $D(\cdot, \cdot)$

Discounting values are given in `PFILE` , which is a 85×260 matrix, with each column corresponding to one of the maturities from 1 week to 260 weeks and with each row corresponding to month-end data from 1-2003 to 2-1010. Recall, the time grid in the finite difference scheme is discretized into monthly value. Then, to correctly discount CDS spreads given by the PDE, we would compute, in the case of 1 year CDS:

```
for i in range(200):
    # 1 YR CDS
    sum1, sum2 = 0, 0
    for j in range(1, 4+1):
        sum1 += PFILE[ICAR_2, 13*j - 1] * F[i, 1+j*3 - 1]
        sum2 += PFILE[ICAR_2, 13*j - 1] * G[i, 1+j*3 - 1]
```

`F` and `G` are both matrices of size $(200, 61)$ i.e. (default intensity grid, time grid), which stores the value of the expectation computed using finite difference, subject to different boundary conditions.

`i` iterates through default intensity grid, which is x , and `j` iterates through time grid, which is t . `ICAR_2` is a hyper-parameter in this case, where it iterates through each observation in the past. But since the likelihood function is later computed by taking the sum of likelihood at each observation in past time. The specific value of `ICAR_2` is not of relevance here.

We can do a sanity check and verify that the value of $PFILE[,]$ and $F[,]$ are consistent in terms of point in time. For 1 year CDS, the second index in $PFILE$ are $\{13, 26, 39, 52\}$ which corresponds to end of the 3, 6, 9, and 12 months. The second index in F is $\{4, 7, 10, 13\}$ with the first time value being t_0 , so this is essentially $\{3, 6, 9, 12\}$ months. The time is consistent. Same thing with 3 and 5 year CDS.

Interpolation

$XINT()$ and $XINTP()$ interpolates CDS spreads on a discrete grid to match market data. It assumes 3 year CDS are priced correctly, and then computes implied default intensity from CDS spreads. The following code finds which interval the market CDS spread falls into, then computes $xlam$ which is implied default intensity as well as 'calibrated' 1 and 5 year CDS spread. $deriv$ computes $\frac{\text{change in CDS spread}}{\text{change in default intensity value}}$, the slope of the interval that market CDS value falls into.

```
for i in range(199):
    if ((CDS3 >= CDS3T[i]) and (CDS3 <= CDS3T[i+1])):
        weight = (CDS3 - CDS3T[i]) / (CDS3T[i+1] - CDS3T[i])
        xlam = (1 - weight) * X[i] + weight * X[i+1]
        cds1x = (1 - weight) * CDS1T[i] + weight * CDS1T[i+1]
        cds3x = (1 - weight) * CDS3T[i] + weight * CDS3T[i+1]
        cds5x = (1 - weight) * CDS5T[i] + weight * CDS5T[i+1]

        deriv = (CDS3T[i+1] - CDS3T[i]) / (np.log(X[i+1]) - np.log(X[i]))
        break
```

Likelihood function

There are (mainly) three components in the likelihood function, each having a normal distribution:

1. interpolated 1 year CDS spread under Q measure

```
# mean
ER1 = cds1 - cds1x
# log-likelihood
TOTAL = TOTAL - 0.5 * np.log(2 * np.pi * SIG1**2) - ER1 * ER1 / (2 * SIG1**2)
```

2. interpolated 5 year CDS spread under Q measure

```
# mean
ER2 = cds5 - cds5x
# log-likelihood
TOTAL = TOTAL - 0.5 * np.log(2 * np.pi * SIG2**2) - ER2 * ER2 / (2 * SIG2**2)
```

3. implied default intensity under P measure

```

# mean
XM = np.log(XLAMB) * np.exp(-1 * BETH / 12.) + ALPH/BETH * (1 - np.exp(-1 * BETH / 12.))
# variance
V = SIG * SIG / (2 * BETH) * (1 - np.exp(-2 * BETH / 12.))
#log likelihood
TOTAL = TOTAL -0.5 * np.log(2 * np.pi * V) -1 * (np.log(xlam) - XM)**2 / (2 * V)

```

Since the default intensity process is a mean-reversion Ornstein-Uhlenbeck (OU) process, its distribution can be given in the following standard notation (see, [for instance](#)):

Suppose

$$dY_t = \alpha (m - Y_t) dt + \beta d\hat{Z}_t$$

then

$$Y_t - Y_0 \cdot e^{-\alpha t} \sim \mathcal{N} \left(m (1 - e^{-\alpha t}), \frac{\beta^2}{2\alpha} (1 - e^{-2\alpha t}) \right)$$

Recall

$$\begin{aligned} d \ln \lambda_t &:= (\text{ALP} - \text{BET} \ln \lambda_t) dt + \text{SIG} dB_t \\ &= \text{BET} \left(\frac{\text{ALP}}{\text{BET}} - \ln \lambda_t \right) dt + \text{SIG} dB_t \end{aligned}$$

then,

$$\ln \lambda_t \sim \mathcal{N} \left(\ln \lambda_t e^{-\frac{\text{BET}}{12}} + \frac{\text{ALP}}{\text{BET}} (1 - e^{-\frac{\text{BET}}{12}}), \frac{\text{SIG}^2}{2\text{BET}} (1 - e^{-2\frac{\text{BET}}{12}}) \right)$$

Note: the time step is 1/12 monthly, and the mapping between two notations is $\alpha = \text{BET}$, $m = \frac{\text{ALP}}{\text{BET}}$, $\beta = \text{SIG}$.

4. derive

```

TOTAL = TOTAL - np.log(deriv)

```

Not entirely sure yet why this is part of the likelihood function. See TODO #2 at front of page.