

PYTHON 金融工程

美式期权和百慕大期权 偏微分方程有限差分

American and Bermudan Option, PDE FD

王的机器出品



1.1. 欧式期权 PDE FD 回顾

在**布莱尔-斯科尔斯** (Black-Scholes, BS) 模型下, 股票价格 $S(t)$ 的 SDE 为

$$dS(t) = [r - q]S(t)dt + \sigma S(t)dB(t)$$

其中

$S(t)$ = t 时点的股票价格

r = 常数型连续利率

q = 常数型连续红利率

σ = 常数型瞬时波动率

根据费曼卡兹定理 (Feynman-Kac Theorem) 对出任意一个金融衍生品 $V(t, S)$ 的 PDE 为

$$\underbrace{\frac{\partial V}{\partial t}}_{\text{时间维度}} + (r - q)S \underbrace{\frac{\partial V}{\partial S}}_{\text{空间维度}} + \frac{1}{2}\sigma^2 S^2 \underbrace{\frac{\partial^2 V}{\partial S^2}}_{\text{空间维度}} - rV = 0$$

这个金融衍生品可以是欧式、美式期权、百慕大、障碍和奇异期权等。只要标的资产 $S(t)$ 的 SDE 一样, 那么衍生品 $V(t, S)$ 的 PDE 都一样, 不同的是**终值条件**和**边界条件**。回顾用 PDE FD 求解衍生品步骤, 我们来看哪些需要调整。

步骤		方程
1	方程解域	边界条件会变, 因此解域会变。向上敲出 (up-and-out) 障碍期权不需要把 S_{\max} 设得很大, 设置为 $S_{\max} = L$ 即可 (L 期权障碍水平), 因为碰到障碍支付为零或者返还。
2	网格打点	该步骤完全不变。
3	终边条件	终值和边界条件会变, 因此根据不同产品的支付来确定终边条件。
4	时空离散	该步骤完全不变。
5	差分格式	该步骤基本不变, 对于有可提前执行 (early exercise) 性质的衍生品会有小小改变, 包括美式期权和百慕大期权。

从上表可看出, 只要弄懂了欧式期权 PDE FD 这五步, 对于其他种类的期权, 这五部其实变化不大, 因此不需要花同样的时间和精力去学习它。本贴就来讲美式期权和百慕大期权 PDE FD, 在此之前先回顾下欧式期权 PDE FD。

步骤		方程
1	方程解域	确定解域, S 的上下界为 $[0, 4 \times \max(S, K)]$, t 的上下界为 $[0, T]$
2	网格打点	<p>在 S-轴和 t-轴上等分成 $m+1$ 和 n 份, 即在 S-轴和 t-轴上有 $m+2$ 和 $n+1$ 点 (分成 $m+1$ 份而不是 m 份的原因是为了符号简便)。</p> $S \in \{S_j\}_{j=0}^{m+1} \Rightarrow S_j = S_{\min} + j\Delta_S, \quad \Delta_S = \frac{S_{\max} - S_{\min}}{m+1}$ $t \in \{t_i\}_{i=0}^{n+1} \Rightarrow t_i = i\Delta_t, \quad \Delta_t = \frac{T}{n}$ <p>建立网格 (t_i, S_j), 在每个点上的期权值表示为 $V_j(t_i) = V(t_i, S_j) = V(i\Delta_t, j\Delta_S)$</p>

3	终边条件	<p>用 ω 来区分欧式看涨看跌期权, $\omega = 1$ 时为看涨, $\omega = -1$ 时为看跌。</p> <p>终值条件</p> $V_j(t_n) = \max(\omega \cdot (j\Delta_S - K), 0)$ <p>边界条件</p> $V_0(t_i) = \max(\omega \cdot (S_0 e^{-q(T-t_i)} - K e^{-r(T-t_i)}), 0)$ $V_{m+1}(t_i) = \max(\omega \cdot (S_{m+1} e^{-q(T-t_i)} - K e^{-r(T-t_i)}), 0)$ <p>得到上面两式都是因为很小或很大时, 欧式期权已经近似于远期。进一步来看, 欧式期权中设置 $S_0 = 0$, 因此 $V_0(t_i) = \max(-\omega K e^{-r(T-t_i)}, 0)$。对于看涨期权 $\omega = 1$ 有 $V_0(t_i) = 0$; 对于看跌期权 $\omega = -1$ 有 $V_0(t_i) = K e^{-r(T-t_i)}$。</p>
4	时空离散	<p>在 $(t_{i,i+1}^\theta, S_j)$ 点上做离散, 其中 $t_{i,i+1}^\theta = \theta t_i + (1 - \theta)t_{i+1}, \theta \in [0, 1]$。</p> <p>时间维度</p> $\frac{\partial V_j(t_{i,i+1}^\theta)}{\partial t} = \frac{V_j(t_{i+1}) - V_j(t_i)}{\Delta_t} + \mathbf{1}_{\{\theta \neq 0.5\}} \mathbf{0}(\Delta_t) + \mathbf{0}(\Delta_t^2)$ <p>空间维度</p> $\frac{V_j(t_{i,i+1}^\theta)}{\partial S} = \theta \frac{V_{j+1}(t_i) - V_{j-1}(t_i)}{2\Delta_S} + (1 - \theta) \frac{V_{j+1}(t_{i+1}) - V_{j-1}(t_{i+1})}{2\Delta_S} + \mathbf{0}(\Delta_S^2)$ $\frac{\partial V_j^2(t_{i,i+1}^\theta)}{\partial S^2} = \theta \frac{V_{j+1}(t_i) - 2V_j(t_i) + V_{j-1}(t_i)}{\Delta_S^2} + (1 - \theta) \frac{V_{j+1}(t_{i+1}) - 2V_j(t_{i+1}) + V_{j-1}(t_{i+1})}{\Delta_S^2} + \mathbf{0}(\Delta_S^2)$
5	差分格式	<p>差分方程为</p> $\underbrace{[I - \theta \Delta_t A]}_{\text{三对角稀疏矩阵}} \times V(t_i) = \underbrace{[I + (1 - \theta) \Delta_t A]}_{\text{三对角稀疏矩阵}} \times V(t_{i+1}) + \underbrace{(1 - \theta) \Delta_t \Omega(t_i) + \theta \Delta_t \Omega(t_{i+1})}_{\text{边界条件}}$ <p>其中</p> <p>I 是 $m \times m$ 的单位矩阵</p> $A = \begin{bmatrix} c_1 & u_1 & \cdots & 0 & 0 \\ l_2 & c_2 & u_2 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & l_{m-1} & c_{m-1} & u_{m-1} \\ 0 & 0 & \cdots & l_m & c_m \end{bmatrix}$ <p>是 $m \times m$ 的三对角线矩阵</p> $l_j = 0.5[\sigma^2 S_j^2 \Delta_S^{-2} - (r - q) S_j \Delta_S^{-1}]$ <p>对于 $j = 1, 2, \dots, m$</p> $c_j = -\sigma^2 S_j^2 \Delta_S^{-2} - r$ $u_j = 0.5[\sigma^2 S_j^2 \Delta_S^{-2} + (r - q) S_j \Delta_S^{-1}]$ $\Omega(t) = \left[\underbrace{l_1 \cdot V_0(t)}_{\text{下边界条件}}, 0, \dots, 0, \underbrace{u_m \cdot V_{m+1}(t)}_{\text{上边界条件}} \right]^T$ <p>是 $m \times 1$ 向量 (存储边界条件值)</p>

PDE FD 最终简化成连续求解下面方程 n 次。

$$\mathbf{M} \times \mathbf{V}(t_i) = \mathbf{z}(t_{i+1})$$

其中

$$\mathbf{M} = \mathbf{I} - \theta \Delta_t \mathbf{A}$$

$$\mathbf{z}(t_{i+1}) = [\mathbf{I} + (1 - \theta) \Delta_t \mathbf{A}] \times \mathbf{V}(t_{i+1}) + (1 - \theta) \Delta_t \boldsymbol{\Omega}(t_i) + \theta \Delta_t \boldsymbol{\Omega}(t_{i+1})$$

由于 \mathbf{M} 也是一个三对角矩阵，我们可用 LU-分解得到 $\mathbf{M} = \mathbf{L} \cdot \mathbf{U}$ ，其中 \mathbf{L} 是一个下对角线矩阵， \mathbf{U} 是一个上对角线矩阵。当 $i = n - 1, n - 2, \dots, 0$ 时

$$\mathbf{M} \times \mathbf{V}(t_i) = \mathbf{z}(t_{i+1})$$

\Leftrightarrow

$$(\mathbf{L} \cdot \mathbf{U}) \times \mathbf{V}(t_i) = \mathbf{z}(t_{i+1})$$

\Leftrightarrow

$$\mathbf{V}(t_i) = \mathbf{U}^{-1} \cdot (\mathbf{L}^{-1} \cdot \mathbf{z}(t_{i+1}))$$

由于 \mathbf{L} 和 \mathbf{U} 对角线上下半的元素都是零，求它们的逆矩阵的复杂度只是 $\mathbf{O}(m)$ 而不是 $\mathbf{O}(m^2)$ 。

美式期权的 PDE 和欧式期权的 PDE 有一些区别，因为前者需要考虑美式期权买方可以在到期日前任何时间行使其权利来买 (看涨) 卖 (看跌) 标的。美式期权 $V(t, S)$ 的 PDE 如下：

$$\min \left\{ -\frac{\partial V}{\partial t} - (r - q)S \frac{\partial V}{\partial S} - \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rV, V - \omega(S - K) \right\} = 0$$

美式期权的终值条件和欧式期权的一样，都是 $V(T, S(T)) = \max(\omega(S(T) - K), 0)$ 。上面 PDE 推导过程如下：

📖 美式期权 PDE 推导

对于美式期权买方，

第一步 - 构建投资组合 Π ，买入 1 个单位衍生生物 V ，卖出 Δ 个单位原生产资产 S

$$\Pi = V - \Delta S$$

第二步 - 利用伊藤公式写出 $d\Pi$ 的表达式

$$d\Pi = \frac{\partial V}{\partial t} dt + \frac{\partial V}{\partial S} dS + \frac{1}{2} \frac{\partial^2 V}{\partial S^2} (dS)^2 - \Delta dS = \frac{\partial V}{\partial t} dt + \frac{1}{2} \frac{\partial^2 V}{\partial S^2} (dS)^2 + \left(\frac{\partial V}{\partial S} - \Delta \right) dS$$

第三步 - 选取适当的 Δ 值将 $d\Pi$ 里的随机项 dS 去掉，即设置 $\left(\frac{\partial V}{\partial S} - \Delta \right) = 0$ 解得 $\Delta = \frac{\partial V}{\partial S}$

$$d\Pi = \frac{\partial V}{\partial t} dt + \frac{1}{2} \frac{\partial^2 V}{\partial S^2} (dS)^2 = \frac{\partial V}{\partial t} dt + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} dt$$

第四步 - 由于组合没有随机项无风险，在 dt 期间其回报也是无风险的 $r\Pi dt$ ，其中 r 是无风险利率。由于卖出股票是有红利的，那么期权投资者还需要额外支付 $q\Delta S dt$ 红利。这时组合变化包含两部分：

美式期权 PDE 推导

$$\underbrace{r\Pi dt}_{\text{组合无风险回报}} + \underbrace{q\Delta S dt}_{\text{额外支出红利}} = r\Pi dt + q \frac{\partial V}{\partial S} S dt$$

第五步 - 美式期权买方有执行期权的权利，一旦择时不是最优，其组合回报会少于“无风险回报和红利”总和。

$$\begin{aligned} & \left(\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} \right) dt \leq r\Pi dt + q \frac{\partial V}{\partial S} S dt \\ \Leftrightarrow & \left(\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} \right) dt \leq r \left(V - \frac{\partial V}{\partial S} S \right) dt + q \frac{\partial V}{\partial S} S dt \\ \Leftrightarrow & \frac{\partial V}{\partial t} - (r - q)S \frac{\partial V}{\partial S} - \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rV \leq 0 \end{aligned}$$

对于美式期权，当 $V > \omega(S - K)$ 时，期权价值比执行价值高，因此期权买方不会行使权利，那么这时美式期权相当于一个欧式期权，其期权价值 V 满足欧式期权的 PDE。

$$\frac{\partial V}{\partial t} - (r - q)S \frac{\partial V}{\partial S} - \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rV = 0, \text{ 当 } V > \omega(S - K)$$

当 $V = \omega(S - K)$ 时，上面的不等式还是成立，因此

$$\frac{\partial V}{\partial t} - (r - q)S \frac{\partial V}{\partial S} - \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rV \leq 0, \text{ 当 } V = \omega(S - K)$$

结合两种情况得出一种更简洁的表达形式

$$\min \left\{ -\frac{\partial V}{\partial t} - (r - q)S \frac{\partial V}{\partial S} - \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rV, V - \omega(S - K) \right\} = 0$$

求解美式期权有两种方法：**迭代法** (iterative method) 和**惩罚法** (penalty method)。

1.2. 美式期权

1.2.1. 迭代法

具体而言，在用 FD 求解 PDE 时，在任一点 (t_i, S_j) 上期权值不能小于内在价值 (intrinsic value)，因此在每次计算 $V_j(t_i)$ 后，还需要和其内在价值 $\omega(S_j - K)$ 作比较取最大值。

$$V_j(t_i) = \max(V_j(t_i), \omega(S_j - K))$$

对于完全显式法 (FE)，我们可以先求解 $V_j(t_i)$ ，再与 $\omega(S_j - K)$ 比较选最大值并更新 $V_j(t_i)$ 。但对于完全隐式法 (FI) 和克兰克尼克尔森法 (CN)，求解 $V_j(t_i)$ 同时改变上式左右其值。

这时我们不能直接像解欧式期权那种用逆矩阵法来求解，而需要采用迭代法，比如**雅克比** (Jacobi) 法、**高斯赛尔德** (Gauss-Seidel, GS) 法和**逐次超松弛** (successive over-relaxation, SOR) 法。



迭代法

在求解线性方程 $\mathbf{Ax} = \mathbf{b}$ 时

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1n} \\ a_{21} & a_{22} & \ddots & \ddots & a_{2n} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & \cdots & a_{nn} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_n \end{bmatrix}$$

当矩阵 \mathbf{A} 很大时，直接求逆 $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ 的计算效率和存储空间会是问题，因此会采用迭代法来求解 \mathbf{x} 。具体操作是分解 $\mathbf{A} = \mathbf{C} + \mathbf{D}$ ，带入方程得到

$$\begin{aligned} &(\mathbf{C} + \mathbf{D})\mathbf{x} = \mathbf{b} \\ \Leftrightarrow &\mathbf{D}\mathbf{x} = -\mathbf{C}\mathbf{x} + \mathbf{b} \\ \Leftrightarrow &\mathbf{x} = -\mathbf{D}^{-1}\mathbf{C}\mathbf{x} + \mathbf{D}^{-1}\mathbf{b} \end{aligned}$$

上面求解过程也有对矩阵求逆，但是我们可以选择合适的矩阵 \mathbf{D} 来提高求逆的效率。最理想的 \mathbf{D} 是单位矩阵 \mathbf{I} 。

定义 $\mathbf{x}^{(k)}$ 是第 k 次迭代计算的结果，带入上式得到从第 $k-1$ 次到第 k 次的关系式。

$$\mathbf{x}^{(k)} = -\mathbf{D}^{-1}\mathbf{C}\mathbf{x}^{(k-1)} + \mathbf{D}^{-1}\mathbf{b}$$

假设该方程的精确解为 \mathbf{x}^* ，那么第 k 次迭代计算之后的误差为

$$\begin{aligned} \mathbf{e}^{(k)} &= \mathbf{x}^* - \mathbf{x}^{(k)} = -\mathbf{D}^{-1}\mathbf{C}\mathbf{x}^* + \mathbf{D}^{-1}\mathbf{b} - (-\mathbf{D}^{-1}\mathbf{C}\mathbf{x}^{(k-1)} + \mathbf{D}^{-1}\mathbf{b}) = -\mathbf{D}^{-1}\mathbf{C}(\mathbf{x}^* - \mathbf{x}^{(k-1)}) \\ &= -\mathbf{D}^{-1}\mathbf{C}\mathbf{e}^{(k-1)} \end{aligned}$$

设定 $\mathbf{B} = -\mathbf{D}^{-1}\mathbf{C} = \mathbf{D}^{-1}(\mathbf{D} - \mathbf{A}) = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$ ，可得到

$$\lim_{k \rightarrow \infty} \mathbf{e}^{(k)} = \lim_{k \rightarrow \infty} \mathbf{B}\mathbf{e}^{(k-1)} = \lim_{k \rightarrow \infty} \mathbf{B}^2\mathbf{e}^{(k-2)} = \cdots = \lim_{k \rightarrow \infty} \mathbf{B}^k\mathbf{e}^{(0)}$$

当 $\lim_{k \rightarrow \infty} \mathbf{B}^k \rightarrow 0$ 那么 $\lim_{k \rightarrow \infty} \mathbf{e}^{(k)} \rightarrow 0$ ，那么第 k 次迭代误差趋近零，迭代数值解收敛于精确解。要使得

$$\begin{aligned} \lim_{k \rightarrow \infty} \mathbf{B}^k &\rightarrow 0 \\ \Leftrightarrow \rho(\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}) &< 1 \\ \Leftrightarrow \rho(\mathbf{B}) &< \|\mathbf{B}\| < 1 \end{aligned}$$

因此只要选择合适的矩阵 \mathbf{D} ，使得 $\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$ 的范数小于等于 1，用迭代法就一定能收敛。

三种迭代法的矩阵 \mathbf{D} 选取以及迭代方程总结在下表。

	\mathbf{D}	迭代方程
Jacobi	$\begin{bmatrix} a_{11} & 0 & \cdots & 0 & 0 \\ 0 & a_{22} & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & 0 & a_{nn} \end{bmatrix}$	$x_i^{(k+1)} = \frac{1}{a_{i,i}} \left(b_i - \sum_{j=1, j \neq i}^n a_{i,j} x_j^{(k)} \right)$
GS	$\begin{bmatrix} a_{11} & 0 & \cdots & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ a_{n1} & a_{n2} & \cdots & \cdots & a_{nn} \end{bmatrix}$	$x_i^{(k+1)} = \frac{1}{a_{i,i}} \left(b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{(k+1)} - \sum_{j=i+1}^n a_{i,j} x_j^{(k)} \right)$
SOR	$\begin{bmatrix} a_{11} & 0 & \cdots & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ a_{n1} & a_{n2} & \cdots & \cdots & a_{nn} \end{bmatrix}$	$x_i^{(k+1)} = \omega \cdot \frac{1}{a_{i,i}} \left(b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{(k+1)} - \sum_{j=i+1}^n a_{i,j} x_j^{(k)} \right) + (1 - \omega) \cdot x_i^{(k)}$

回顾

$$\mathbf{M} \times \mathbf{V}(t_i) = \mathbf{z}(t_{i+1})$$

其中

$$\begin{aligned} \mathbf{M} &= \mathbf{I} - \theta \Delta_t \mathbf{A} \\ \mathbf{z}(t_{i+1}) &= [\mathbf{I} + (1 - \theta) \Delta_t \mathbf{A}] \times \mathbf{V}(t_{i+1}) + (1 - \theta) \Delta_t \mathbf{Q}(t_i) + \theta \Delta_t \mathbf{Q}(t_{i+1}) \end{aligned}$$

从 (t_{i+1}, S_j) 到 (t_i, S_j) ，设置在 t_i 时要求解的初始值 $V_{i,j}^{(0)}$ 为在 t_{i+1} 时已求解的值 $V_{i+1,j}$ ，即 $V_{i,j}^{(0)} = V_{i+1,j}$ 。根据 SOR 迭代法的表达式

$$\begin{aligned} \text{第 } k+1 \text{ 迭代 } \mathbf{V} \text{ 中第 } j \text{ 个值 } \widehat{V_{i,j}^{(k+1)}} &= \frac{\alpha}{M_{j,j}} \left(z_{i+1,j} - \sum_{p=1}^{j-1} M_{p,j} \widehat{V_{i,p}^{(k+1)}} - \sum_{p=j+1}^m M_{p,j} \widehat{V_{i,p}^{(k)}} \right) + (1 - \alpha) V_{i,j}^{(k)} \\ &= V_{i,j}^{(k)} + \frac{\alpha}{M_{j,j}} \left(z_{i+1,j} - \sum_{p=1}^{j-1} M_{p,j} V_{i,p}^{(k+1)} - \sum_{p=j}^m M_{p,j} V_{i,p}^{(k)} \right) \end{aligned}$$

其中 α 是 SOR 的参数，通常在 0 到 2 之间，本书里设置为 1.5。

矩阵 $\mathbf{M} = \mathbf{I} - \theta \Delta_t \mathbf{A}$ 的展开形式如下：

$$\mathbf{I} - \theta \Delta_t \mathbf{A} = \begin{bmatrix} 1 - \theta \Delta_t c_1 & -\theta \Delta_t u_1 & \cdots & 0 & 0 \\ -\theta \Delta_t l_2 & 1 - \theta \Delta_t c_2 & -\theta \Delta_t u_2 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & -\theta \Delta_t l_{m-1} & 1 - \theta \Delta_t c_{m-1} & -\theta \Delta_t u_{m-1} \\ 0 & 0 & \cdots & -\theta \Delta_t l_m & 1 - \theta \Delta_t c_m \end{bmatrix}$$

定义 $G_j = G(S_j) = \omega(S_j - K)$ 为期权内在价值，考虑具体的三对角矩阵 \mathbf{M} 的元素，带入迭代表达式 $V_j(t_i) = \max(V_j(t_i), \omega(S_j - K))$ 得到

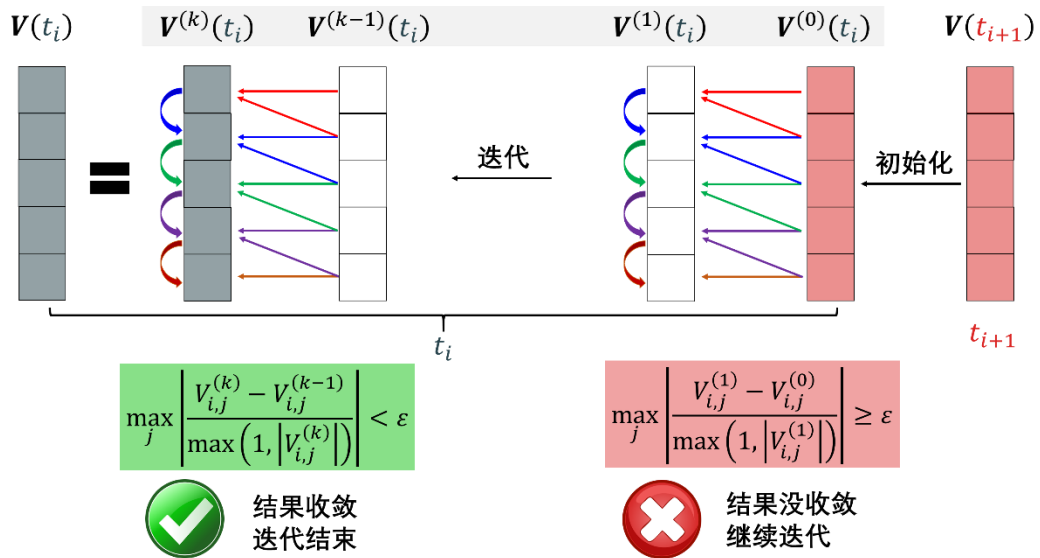
$$V_{i,j}^{(k+1)} = \begin{cases} \max \left(G_j, V_{i,j}^{(k)} + \frac{\alpha}{1 - \theta \Delta_t c_j} (z_{i+1,j} - [1 - \theta \Delta_t c_j] V_{i,j}^{(k)} + \theta \Delta_t u_j V_{i,j+1}^{(k)}) \right), & j = 1 \\ \max \left(G_j, V_{i,j}^{(k)} + \frac{\alpha}{1 - \theta \Delta_t c_j} (z_{i+1,j} + \theta \Delta_t l_j V_{i,j-1}^{(k+1)} - [1 - \theta \Delta_t c_j] V_{i,j}^{(k)} + \theta \Delta_t u_j V_{i,j+1}^{(k)}) \right), & j = 2, \dots, m-1 \\ \max \left(G_j, V_{i,j}^{(k)} + \frac{\alpha}{1 - \theta \Delta_t c_j} (z_{i+1,j} + \theta \Delta_t l_j V_{i,j-1}^{(k+1)} - [1 - \theta \Delta_t c_j] V_{i,j}^{(k)}) \right), & j = m \end{cases}$$

每轮迭代的终止条件为

$$\max_j \left| \frac{V_{i,j}^{(k+1)} - V_{i,j}^{(k)}}{\max(1, |V_{i,j}^{(k+1)}|)} \right| < \varepsilon$$

其中 ε 是 SOR 的收敛参数，本书里设置为 10^{-6} 。

从 t_{i+1} 到 t_i 迭代过程的可视图如下，不同颜色的箭头代表用哪些 $V_{i,j}^{(k)}, V_{i,j+1}^{(k)}$ 和 $V_{i,j-1}^{(k+1)}$ 来更新 $V_{i,j}^{(k+1)}$ 。



在时点 t_i 一旦结果收敛，我们继续从 (t_i, S_j) 到 (t_{i-1}, S_j) 按上面的过程求解方程直到得到 V_0 。整个流程总结在下表，其中 $V_i^{(k)} = [V_{i,1}^{(k)}, V_{i,2}^{(k)}, \dots, V_{i,m}^{(k)}]$ 。

步骤	迭代过程	迭代次数
$t_n \rightarrow t_{n-1}$	$V_n = V_{n-1}^{(0)} \rightarrow V_{n-1}^{(1)} \rightarrow \dots \rightarrow V_{n-1}^{(k_n)} = V_{n-1}$	k_n
$t_{n-1} \rightarrow t_{n-2}$	$V_{n-1} = V_{n-2}^{(0)} \rightarrow V_{n-2}^{(1)} \rightarrow \dots \rightarrow V_{n-2}^{(k_{n-1})} = V_{n-2}$	k_{n-1}
\vdots		
$t_2 \rightarrow t_1$	$V_2 = V_1^{(0)} \rightarrow V_1^{(1)} \rightarrow \dots \rightarrow V_1^{(k_2)} = V_1$	k_2
$t_1 \rightarrow t_0$	$V_1 = V_0^{(0)} \rightarrow V_0^{(1)} \rightarrow \dots \rightarrow V_0^{(k_1)} = V_0$	k_1

1.2.2. 惩罚法

美式期权 PDE 模型还可以写成一个**线性互补问题** (linear complementarity problem, LCP)，证明忽略直接给出结果。

$$\min \left\{ -\frac{\partial V}{\partial t} - (r - q)S \frac{\partial V}{\partial S} - \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rV, V - \omega(S - K) \right\} = 0$$

$$\Leftrightarrow \frac{\partial V}{\partial t} + (r - q)S \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV + \lambda(G - V)^+ = 0, \quad G = \max(\omega(S - K), 0)$$

其中 λ 是一个惩罚系数。上面最终结果是一个非线性的 PDE。首先类比两种方法的 PDE：

方法	偏微分方程
普通法	$\frac{\partial V}{\partial t} + (r - q)S \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV = 0$
惩罚法	$\frac{\partial V}{\partial t} + (r - q)S \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV + \lambda(G - V)^+ = 0$

两种方法的 PDE 唯一区别就是惩罚项 $\lambda(G - V)^+$ ，在 (t_{i+1}^θ, S_j) 点上对此项做离散

$$\lambda(G_j - V_j(t_{i+1}^\theta))^+ = \lambda(G_j - \theta V_j(t_i) - (1 - \theta)V_j(t_{i+1}))^+ = \lambda \theta \left(\frac{G_j}{\theta} - \underbrace{V_j(t_i)}_{\text{未知}} - \frac{1 - \theta}{\theta} \underbrace{V_j(t_{i+1})}_{\text{已知}} \right)^+$$

用同样的迭代思路，设置在 t_i 时要求解的初始值 $V_{i,j}^{(0)}$ 为在 t_{i+1} 时已求解的值 $V_{i+1,j}$ ，即 $V_{i,j}^{(0)} = V_{i+1,j}$ 。在第 $k+1$ 次迭代计算 $V_{i,j}^{(k+1)}$ 时， $V_{i,j}^{(k)}$ 是已知的。

$$\begin{aligned}
\lambda\theta\left(\frac{G_j}{\theta} - V_j(t_i) - \frac{1-\theta}{\theta}V_j(t_{i+1})\right)^+ &= \lambda\theta\left(\frac{G_j}{\theta} - V_{i,j}^{(k+1)} - \frac{1-\theta}{\theta}V_{i+1,j}\right)^+ \\
&= \lambda\theta\left(\underbrace{\frac{G_j}{\theta} - \frac{1-\theta}{\theta}V_{i+1,j} - V_{i,j}^{(k)}}_x + \underbrace{V_{i,j}^{(k)} - V_{i,j}^{(k+1)}}_{\Delta x}\right)^+ \\
&= \lambda\theta \cdot f(x + \Delta x) \\
&= \lambda\theta \cdot [f(x) + f'(x)\Delta x] \\
&= \lambda\theta \cdot [\max(x, 0) + \mathbf{1}_{\{x>0\}}\Delta x] \\
&= \lambda\theta \cdot \begin{cases} x + \Delta x, & x > 0 \\ 0, & x \leq 0 \end{cases} \\
&= \lambda\theta \cdot (x + \Delta x) \cdot \mathbf{1}_{\{x>0\}} \\
&= \lambda\theta\left[\frac{G_j}{\theta} - \frac{1-\theta}{\theta}V_{i+1,j} - V_{i,j}^{(k)} + V_{i,j}^{(k)} - V_{i,j}^{(k+1)}\right] \cdot \mathbf{1}_{\left\{\frac{G_j}{\theta} - \frac{1-\theta}{\theta}V_{i+1,j} - V_{i,j}^{(k)} > 0\right\}} \\
&= \lambda\left[G_j - (1-\theta)V_{i+1,j} - \theta V_{i,j}^{(k+1)}\right] \cdot \underbrace{\mathbf{1}_{\left\{G_j > (1-\theta)V_{i+1,j} + \theta V_{i,j}^{(k)}\right\}}}_{d_j(t_{i,i+1}^\theta)} \\
&= \lambda\left[G_j - (1-\theta)V_{i+1,j} - \theta V_{i,j}^{(k+1)}\right] \cdot d_j(t_{i,i+1}^\theta) \\
&= \lambda G_j d_j(t_{i,i+1}^\theta) - \lambda(1-\theta)d_j(t_{i,i+1}^\theta)V_{i+1,j} - \lambda\theta d_j(t_{i,i+1}^\theta)V_{i,j}^{(k+1)}
\end{aligned}$$

以普通法的差分方程为起点，加上上面惩罚项 $\lambda(G - V)^+$ 的离散表达式，我们推导出惩罚法的差分方程。

普通法		
偏微分方程	$\frac{\partial V}{\partial t} + (r - q)S \frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV = 0$	
差分方程	元素形式	$V_{i+1,j} + (1 - \theta)\Delta_t \cdot [l_j \times V_{i+1,j-1} + c_j \times V_{i+1,j} + u_j \times V_{i+1,j+1}] = V_{i,j} - \theta\Delta_t \cdot [l_j \times V_{i,j-1} + c_j \times V_{i,j} + u_j \times V_{i,j+1}]$
	矩阵形式	其中 $\mathbf{M} \times \mathbf{V}(t_i) = \mathbf{z}(t_{i+1})$ $\mathbf{M} = \mathbf{I} - \theta\Delta_t \mathbf{A}$ $\mathbf{z}(t_{i+1}) = [\mathbf{I} + (1 - \theta)\Delta_t \mathbf{A}] \times \mathbf{V}(t_{i+1}) + (1 - \theta)\Delta_t \mathbf{\Omega}(t_i) + \theta\Delta_t \mathbf{\Omega}(t_{i+1})$

惩罚法		
偏微分方程	$\frac{\partial V}{\partial t} + (r - q)S \frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV + \lambda(G - V)^+ = 0$	
差分方程	元素形式	$V_{i+1,j} + (1 - \theta)\Delta_t \cdot [l_j \times V_{i+1,j-1} + (c_j - \lambda\Delta_t d_j(t_{i,i+1}^\theta)) \times V_{i+1,j} + u_j \times V_{i+1,j+1}] + \lambda G_j d_j(t_{i,i+1}^\theta)\Delta_t = V_{i,j}^{(k+1)} - \theta\Delta_t \cdot [l_j \times V_{i,j-1}^{(k+1)} + (c_j - \lambda\Delta_t d_j(t_{i,i+1}^\theta)) \times V_{i,j}^{(k+1)} + u_j \times V_{i,j+1}^{(k+1)}]$
	矩阵形式	其中 $\mathbf{M} \times \mathbf{V}^{(k+1)}(t_i) = \mathbf{z}(t_{i+1})$ $\mathbf{M} = \mathbf{I} - \theta\Delta_t (\mathbf{A} - \lambda\Delta_t \mathbf{D})$ $\mathbf{z}(t_{i+1}) = [\mathbf{I} + (1 - \theta)\Delta_t (\mathbf{A} - \lambda\Delta_t \mathbf{D})] \times \mathbf{V}(t_{i+1}) + (1 - \theta)\Delta_t \mathbf{\Omega}(t_i) + \theta\Delta_t \mathbf{\Omega}(t_{i+1}) + \lambda \mathbf{D} \mathbf{G} \Delta_t$

$$D = \begin{bmatrix} d_1 & 0 & \cdots & 0 & 0 \\ 0 & d_2 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & d_{m-1} & 0 \\ 0 & 0 & \cdots & 0 & d_m \end{bmatrix}, \quad d_j = \mathbf{1}_{\{G_j > (1-\theta)V_{i+1,j} + \theta V_{i,j}^{(k)}\}}$$

同 SOR 法一样，每轮迭代的终止条件为

$$\max_j \left| \frac{V_{i,j}^{(k+1)} - V_{i,j}^{(k)}}{\max\left(1, |V_{i,j}^{(k+1)}|\right)} \right| < \varepsilon$$

其中 ε 是 SOR 的收敛参数，本书里设置为 10^{-6} 。

1.2.3. 具体例子

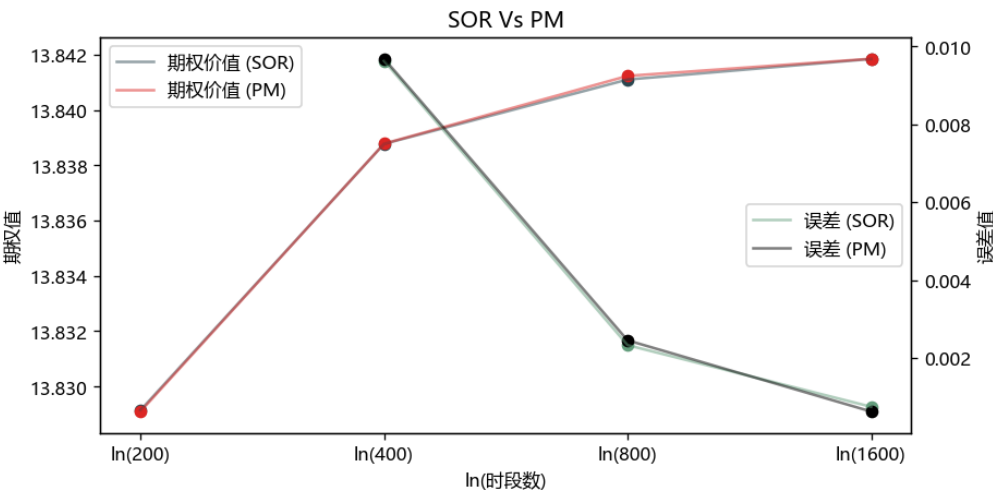
考虑一个美式看跌期权，它的特征和欧式看跌期权一样，如下表所示：

属性	股票 价格	行权 价格	连续 利率	连续 红利率	波动率	到期 年限
符号	S	K	r	q	σ	T
值	50	60	3%	1%	40%	1

用迭代法 (SOR) 和惩罚法 (PM) 计算的美式期权结果如下：

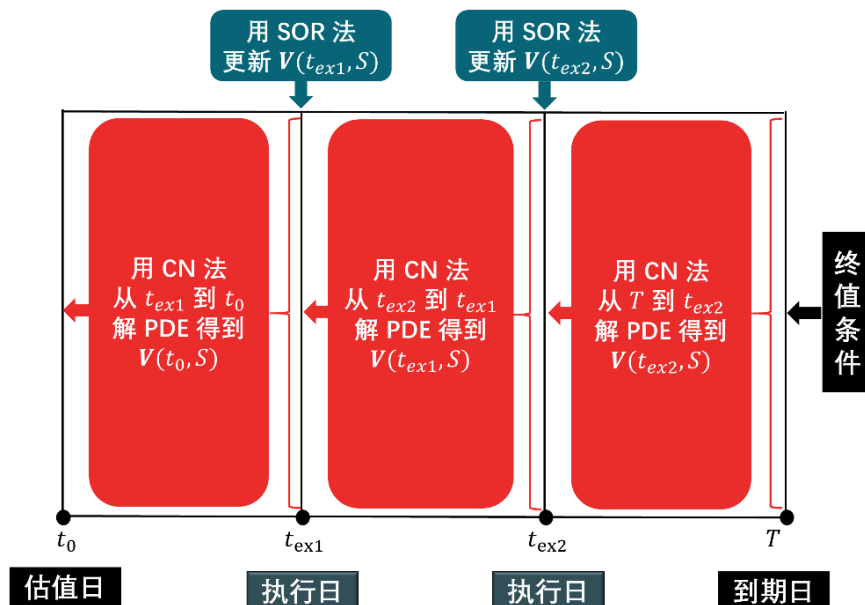
方法	美式期权	方法	欧式期权
SOR	13.8387	Closed-Form	13.5832
PM	13.8387	CN	13.5806

从上表结果发现 SOR 和 PM 得到的结果高度一致，而且美式期权比欧式期权价值更大。检查 SOR 和 PM 的计算结果，发现高度一致 (下图左 y 轴是期权值，红色和深青色几乎重合) 而且收敛速率快 (下图右 y 轴是误差值，分别衡量 N_t 从 200 到 400，从 400 到 800，从 800 到 1600 计算的期权值的差异)。



1.3. 百慕大期权

百慕大期权 (Bermuda option) 是一种可以在到期日前所规定的**一系列时间**行权的期权。比如，期权 3 年后到期，但只有在最后一年的每个月才能被执行。由于美式期权是可以在任意时间行权，因此美式期权可看成是一种特殊的百慕大期权。用 PDE FD 来求解百慕大期权也非常简单，在所有执行日上用美式期权那一套 (SOR 或 PM) 解 PDE，在其他时间上用欧式期权那一套 (FE, FI 或 CN) 解 PDE。



考虑三个百慕大看跌期权，它的特征和欧式和美式看跌期权一样，如下表所示：

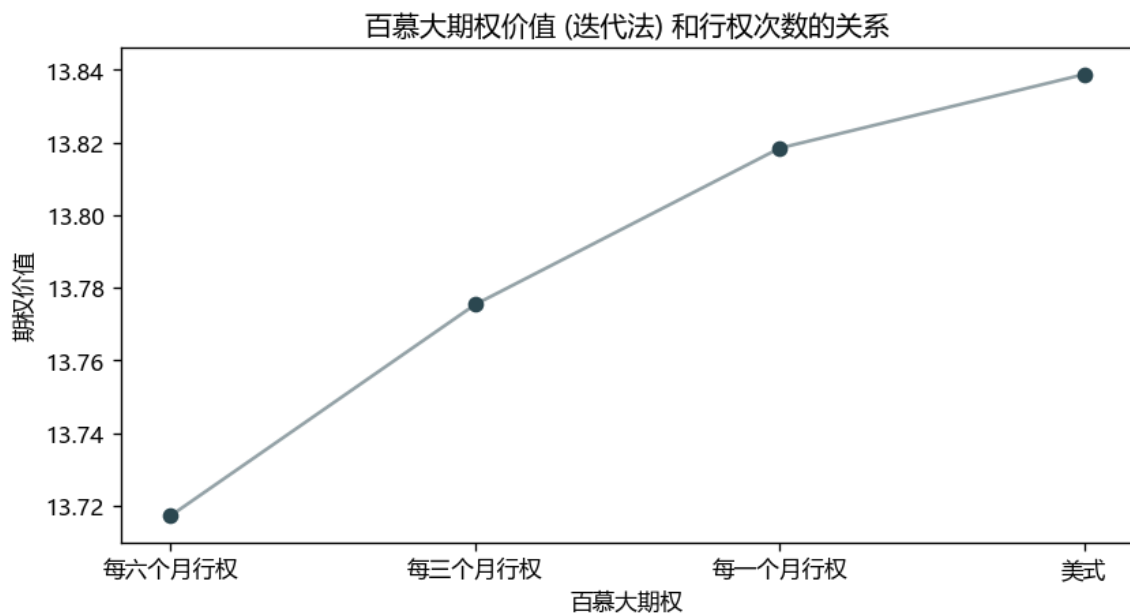
属性	股票价格	行权价格	连续利率	连续红利率	波动率	到期年限
符号	S	K	r	q	σ	T
值	50	60	3%	1%	40%	1

不同的是它们分别是 (执行日不包括最后到期日)

- 每半年执行权利，执行日的年限为 $[0.5]$
- 每三个月执行权利，执行日的年限为 $[0.25, 0.5, 0.75]$
- 每个月执行权利，执行日的年限为 $[\frac{1}{12}, \frac{2}{12}, \frac{3}{12}, \frac{4}{12}, \frac{5}{12}, \frac{6}{12}, \frac{7}{12}, \frac{8}{12}, \frac{9}{12}, \frac{11}{12}]$

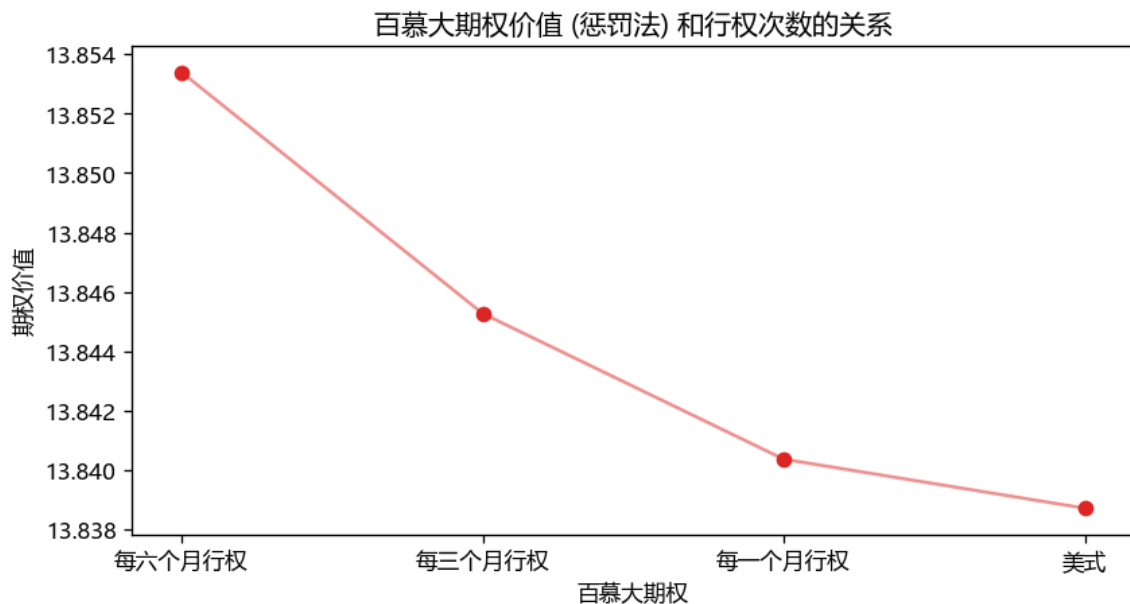
用 SOR 法计算的结果和图如下：

期权	价值
每半年执行	13.7172
每三个月执行	13.7753
每个月执行	13.8184
任意时间执行 (美式)	13.8387



用 PM 法计算的结果和图如下：

期权	价值
每半年执行	13.8534
每三个月执行	13.8453
每个月执行	13.8404
任意时间执行 (美式)	13.8387



随着可执行日的增加，百慕大期权价值越小，结果是不合理的。

接着我们用两个极端情况：

- 没有执行日，这时百慕大期权是个欧式期权
- 可连续执行，这时百慕大期权是个美式期权

比较结果如下表所示

PM 方法	百慕大期权价值	极限情况	价值
没有执行日	13.580616848408729	欧式期权	13.580616848408729
可连续执行	13.838720062454613	美式期权	13.838720062454613

极限情况的结果完全正确，侧面证明 PM 法在单独处理美式和欧式期权定价部分都没有错，但为了保险起见，我们还是用 SOR 法来对百慕大期权进行定价。