一个带有幂收益结构的复合期权定价研究

摘　要

本文旨在对一个以奇异看跌期权 V_1 为标的资产的复合期权 V_2（在本项目中称为衍生品 X)进行定价和敏感性分析。内层期权 V_1 的收益结构为 $\max\{K_2 - (S(t_{ex}))^{3/2}, 0\}$，它具有幂函数的奇异特征。我们选择腾讯控股有限公司（Tencent Holdings Ltd.）的股票作为 V_1 的标的资产，并从公开市场数据中估计了历史波动率、股息收益率和无风险收益率等关键输入参数。在定价模型方面，我们分别采用了二项式定价模型和蒙特卡洛模拟（Monte Carlo Simulation, MCS）方法。通过比较两种模型的定价结果，并对关键参数（如波动率、时间、行权价）进行敏感性分析，我们深入理解了这些因素对衍生品 X 价值的影响。特别地，针对内层美式期权 V_1 的定价，MCS 模型采用了 Longstaff-Schwartz（LSM）方法进行改进。报告最后还对两种模型的收敛性进行了对比分析。

关键词：复合期权定价；幂收益期权；二项式模型；蒙特卡洛模拟；Longstaff-Schwart

目 录

一、引　言

1.1 项目背景与衍生品结构

期权定价是金融衍生品领域的核心问题。本文关注一个具有两层复杂性的衍生品 $X(V_2)$：

(1) 内层期权 (V_1)

一个以标的资产 S 为基础的奇异看跌期权。其到期日为 T_{V1}，行权价为 K_2，收益结构为 $V_1(S(T_{V1})) = \max\{K_2 - (S(t_{ex}))^{3/2}, 0\}$，其中 $t_{ex} = T_{V1}$（欧式）或 $t_{ex} \le T_{V1}$（美式）。收益中的幂函数 $S^{3/2}$ 引入了奇异特征。

(2) 外层期权 (V_2)

一个以 V_1 作为标的资产的欧式期权。其到期日为 T_{V2}，行权价为 K_1，收益为 $V_2 = \max\{N_1(V_1(T_{V2}, \cdot, \cdot) - K_1), 0\}$。本项目旨在为这一复杂的复合奇异期权 V_2 在当前时间 t 进行定价。

1.2 定价方法概述

由于衍生品 X 涉及奇异收益和复合结构，传统的 Black-Scholes 模型不再适用。本项目采用两种数值方法进行定价：

(1) 二项式定价模型 (Binomial Model)

基于离散时间步长，采用风险中性定价原理，通过回溯计算期权价值。

(2) 蒙特卡洛模拟 (Monte Carlo Simulation, MCS)

通过大量路径模拟标的资产价格，计算期权收益的期望值，并折现。对于内层美式期权 V_1，结合了 Longstaff-Schwartz (LSM) 方法处理提前行权的优化问题。

1.3 本文定价前提假设

(1) 标的资产 S 的价格服从几何布朗运动 (Geometric Brownian Motion, GBM)；

(2) 市场是无摩擦的（无交易成本、税收），且连续交易；

(3) 无套利机会存在。

二、标的资产

2.1 标的资产介绍

本文选取腾讯控股有限公司（Tencent Holdings Limited, 股票代码：0700.HK）作为期权定价研究的标的资产。在市场特性方面，港股市场进行 T+0 回转交易，不设涨跌幅限制，日内价格波动更能反映投资者对新信息的即时反应，这一特点使其适合作为期权定价与波动率研究的标的资产。在分红政策方面，腾讯实行定期派息，其股息率相对温和但具有稳定性。因此，在构建欧式期权定价模型时，需将股息收益率纳入贴现与漂移项的处理之中，以保证定价的准确性。

2.2 历史波动率

股票的波动率σ用于度量股票收益产生的不确定性，在本文中定义为股价年化连续复利收益率的标准差，股票的波动率通常介于 15%-60%之间(John Haul, 1988)。对于历史波动率的估计，通常是用固定区间内的股票价格数据进行标准差计算。

具体的，对于 n 个观测日的 n 个股票收盘价观测值S_i，我们定义股票的对数收益率u_i为：

$$u_i = \ln\left(\frac{S_i}{S_{i-1}}\right), \quad i = 1,2,\dots,n \tag{1}$$

收益率的标准差估计值 s 为：

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(u_i - \bar{u})^2} \tag{2}$$

我们定义以年为单位的时间长度为τ，本文取腾讯股票收盘价 2024 年 11 月 5 日到 2025 年 11 月 5 日 250 个收盘价数据作为观测值，则$\tau = \frac{1}{250}$，则σ的估计值可以由下式求得：

$$\hat{\sigma} = \frac{s}{\sqrt{\tau}} \tag{3}$$

该估计值的标准误差大约为$\frac{\hat{\sigma}}{\sqrt{2n}}$。经计算，腾讯股价在本文研究区间的历史波动率约为 0.35, 该估计值的标准误大约为 0.0157，即 1.57%。

2.3 隐含波动率

期权的隐含波动率，简单说就是市场根据期权的价格倒推出来的波动率。它反映了市场对未来股票价格波动程度的预期，从而判断市场情绪是乐观还是悲观。隐含波动率的取决于定价模型的不同，在 Black -Scholes 模型中，对于连续股息

率为 q 的期权，有:

$$C = S_0 e^{-qT} N(d_1) - K e^{-rT} N(d_2) \tag{4}$$

其中：

$$d_1 = \frac{\ln(S_0/K) + (r - q + 1/2\sigma^2)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T} \tag{5}$$

由于本文中 X 的内层期权是一个奇异的幂期权，属于场外期权，市场数据难以获得。因此本文认为其一的一个替代方法是使用交易所标准期权的隐含波动率替代。由于标的资产相同，本文认为这是一个合理的替代。

在具体求解上，首先需要收集到期日与执行价格相同或相似的腾讯股票期权市场数据，然后使用数值方法进行求解。

2.4 股息收益率

本文定义一年内股息收益率为:

$$r_i = \frac{dividend\_per\_share_i}{S_t}$$

$$R_i = 1 + r_i \tag{6}$$

本文取腾讯公司 2023 年至 2025 年的派息数据估算年化连续股息收益率 q，q 可由下面的公式计算得出：

$$q = \frac{ln\left(\prod_{i=1}^{3} R_i\right)}{3} \tag{7}$$

经计算，q 的值约为 0.008，即 0.8%。

2.5 无风险收益率

本文取研究区间内 12 个十年期美债的到期收益率观测值的均值作为无风险收益率的估计值，经计算，本文的无风险收益率估计值 rf 为 0.0428，即 4.28%。

三、二项式定价

3.1 二项式定价原理

二项式定价方法(M^{CRR})是 Cox 与 Ross 于 1979 年提出的一种期权定价方法。M^{CRR}将时间范围[0,T]分为 M 个长度为Δt 的等长区间，则在 t 内有 M+1 个点。M^{CRR}假设模型中存在两种资产，一种为有风险的资产，另一种为无风险的债券。模型无风险短期利率存在，且服从$B_t = B_0 e^{rt}$，$B_0 > 0$。

模型假设风险资产价格初期为正且变化服从如下规律：

$$S_{t+\Delta t} = S_t * m \tag{7}$$

其中，m 随机等于{u, d}，其中，$0 < d < e^{r\Delta t} < u \equiv e^{\sigma\sqrt{\Delta t}}$ 。为了简化，模型有 $u = \frac{1}{d}$ 。

假设无风险中性定价成立，风险资产价格服从下面的关系：

$$S_t = e^{-r\Delta t} \cdot \mathbf{E}^{\mathbf{Q}}[S_{t+\Delta t}] \tag{8}$$

其中，风险中性概率 q 为：

$$q = \frac{e^{r\Delta t} - d}{u - d} \tag{9}$$

则欧式期权的价值可以将 t = T 时刻的最终收益折现至 t = 0 时刻得到：

$$C_0 = e^{-rT}\mathbf{E}^{\mathbf{Q}}[C_T] \tag{10}$$

折现的过程从 $t = T - \Delta t$ 开始，一步一步、一个节点一个节点倒推完成。

3.1.1 美式期权的二项式定价

美式期权的二项式定价与欧式期权相似。不同的点在于美式期权可提前执行，因此在每一个节点需要比较提前行权的价值(执行价值)与继续持有的期望价值(持有价值)，则节点处期权价值为 max(执行价值,持有价值)。在每一个节点进行比较，一个一个节点倒推折现回期初即为美式期权的二项式定价价格。

3.1.2 复合期权 X 的二项式定价模型实现

对于复合期权 V2，定价需要一个双重回溯过程，下文称为双重二叉树。首先构建价格树: 构建从 t 到 T_{V1} 的完整二叉树，总时间步 M=Mouter+Minner。

(1) 定价内层期权 (V1)

从T_{V1}（内层期权到期日）开始，计算 V1 的到期收益 $\max\{K_2-(S(T_{V1}))^{3/2},0\}$。回溯至 T_{V2}（外层期权到期日/内层期权起始日），在每个节点上，根据风险中性概率和折现因子计算 V1 的期望价值。

对于美式 V1，在回溯的每一步（$j=T_{V1}-1,\ldots,T_{V2}$），必须比较继续持有价值和即时行权价值 $\max\{K_2-\left(S\left(t_j\right)\right)^{3/2},0\}$，并取两者中较大者。

(2) 定价外层期权 (V2):

以 T_{V2} 时刻树上所有 V1 的价值作为 V2 在 T_{V2} 时刻的标的资产价格。

计算 V2 在 T_{V2} 的收益$V_2=\max\{N_1(V_1(T_{V2})-K_1),0\}$。将此收益从 T_{V2} 时刻回溯到 t 时刻，得到最终的 V2 价格。

3.2 二项式定价结果

3.2.1 期权合约及二项式参数

(1) 外层期权执行价 K1

由于期权 X 为虚拟期权，外层期权的执行价无可参考价格，故本文取内层期权在 T 时刻价格的中位数作为外层期权 X 的基准执行价格，经计算为 70$

(2) 内层期权执行价 K2

参考香港交易所腾讯股票标准欧式期权的常见执行价，基准值设为 630HK$，经汇率转化后，为 81.31$。

(3) 二叉树总步数 N

基准值设为 1000。

(4) 当前时间 t

基准值为 2025 年 11 月 5 日。

(5) 内层期权起始日 TV2

参考香港交易所腾讯标准欧式期权交易日，基准值设为 2026 年 2 月 25 日。

(6) 内层期权到期日 TV1

基准期限为 3 个月，到期时间为 2026 年 5 月 26 日。

本文的二项式定价表格如下：

表 1　二项式定价表

外层期权类型	外层期权执行价 K1	外层期权时间步 N_outer	内层期权类型	内层期权期限 C_inner（年）	内层期权执行价 K2	内层期权时间步 N_inner	衍生品 X 价格
看涨	70	555	欧式看跌	0.25	81.31	445	48.43
看涨	70	555	美式看跌	0.25	81.31	445	51.18

四、蒙特卡洛模拟定价

4.1 蒙特卡洛模拟基础

4.1.1 定义

蒙特卡洛模拟是一种基于随机抽样的数值计算方法，通过生成大量随机路径来估计金融衍生品的期望价值。在期权定价中，其核心思想是在风险中性测度下模拟标的资产价格路径，然后计算每条路径的 payoff 的贴现值，最后取平均值作为期权价格的估计。简而言之，其核心思想是用大量的随机抽样来估计期望值，用“平均值”逼近“期望值”。在风险中性测度下，期权价格可以表示为贴现期望收益：

$$V = e^{-rT} E^{Q}[payoff] \tag{11}$$

4.1.2 奇异期权的结构分析

(1) 内层期权（幂期权）

$$Payoff_1 = \max(K_2 - S(t_{ex})^{1.5}, 0) \tag{12}$$

(3) 外层期权（复合期权）

$$Payoff_2 = \max(N \times (V_1(T) - K_1), 0) \tag{13}$$

4.2 蒙特卡洛模拟结果

基于以上基础和参数设定（二叉树部分），设置内外层模拟路径数 $N_{inner} = N_{outer} = 10000$，我们计算了内层期权为欧式和美式的情况，结果如下：

表 2　蒙特卡洛计算结果

内层期权类型	Derivative X 价格	二叉树的结果
欧式	48.03183752	48.4324002
美式	51.75982308	51.1804167

可以看出，二叉树和蒙特卡洛的结果接近。为了更直观地对比二叉树和蒙特卡洛模拟的结果，我们绘制了内层为欧式期权的蒙特卡洛收敛图以及和二叉树对比图：

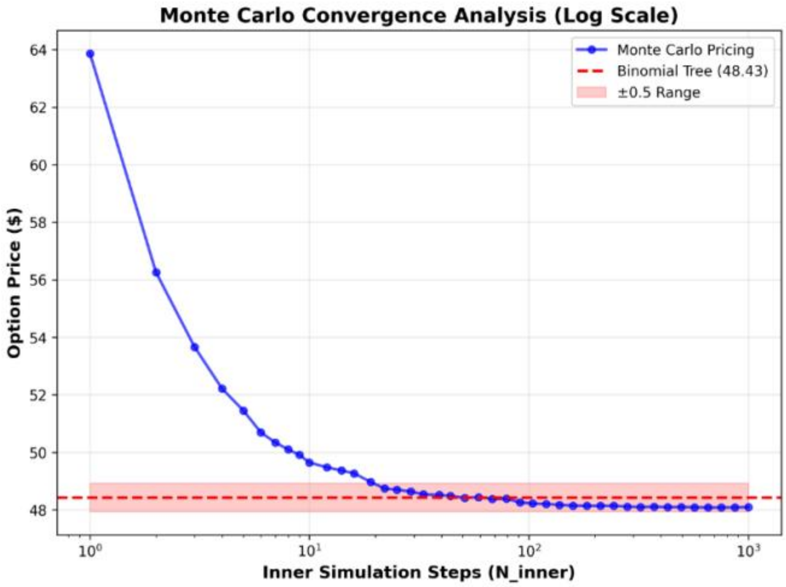

图 1　内层为欧式期权的蒙特卡洛收敛

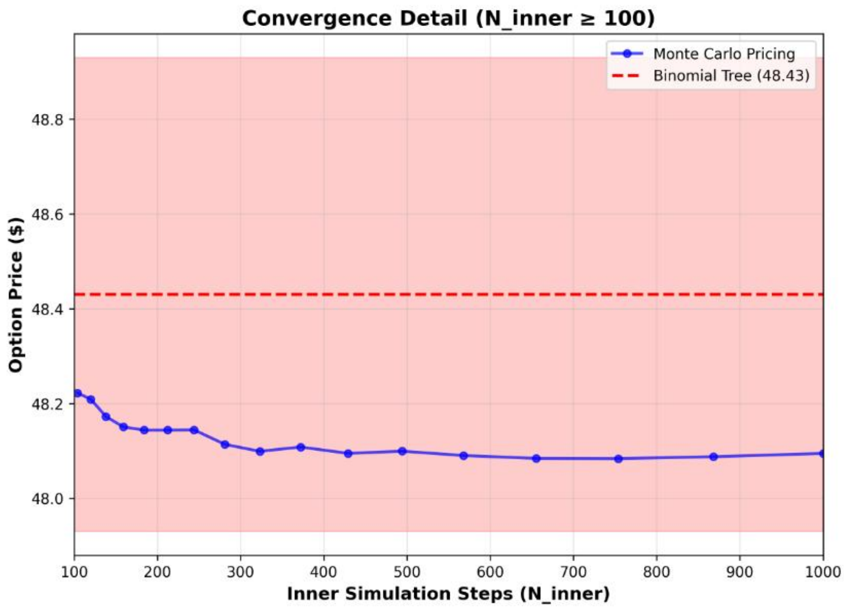

图 2　层为欧式期权的蒙特卡洛和二叉树结果对比

五、敏感性分析

5.1 Delta 和 Gamma 定义

5.1.1 Delta(Δ)

希腊字母 $delta$ 表示标的资产价格变化引起的期权价格变化，计算方式为：

$$delta(\Delta) = \frac{\partial V}{\partial S} \tag{14}$$

一般而言，看涨期权的 $delta$ 为正值，看跌期权的 $delta$ 为负值。在二叉树中，

$$delta = \frac{\partial f}{\partial S} = \frac{f_{s+\Delta s} - f_{s-\Delta s}}{2\Delta S} \tag{15}$$

5.1.2 Gamma(γ)

希腊字母 $Gamma$ 表示标的资产价格变化引起的 $delta$ 值变化，一般的计算方式为：

$$Gamma(\Gamma) = \frac{\partial \Delta}{\partial S} \tag{16}$$

看涨期权与看跌期权的多头 $Gamma$ 值均大于 0，看涨期权与看跌期权的 $Gamma$ 值均小于 0，平值期权的 $Gamma$ 最大，此时 $delta$ 的变化率最大。深度虚值和深度实值期权的 $Gamma$ 接近于 0。在二叉树中：

$$Gamma = \frac{\partial^2 f}{\partial S^2} = \frac{f_{S+\Delta S} + f_{S-\Delta S} - 2f_S}{\Delta S^2} \tag{17}$$

5.2 不同情形 X 的 delta 和 Gamma 结果

表 3　二叉树基本敏感性分析

希腊字母	内层期权为欧式	内层期权为美式
Delta	-3.919311	-4.125624
Gamma	0.142785	0.226336

具体而言，当内层期权为欧式期权时，腾讯股价每上涨 1 美元，期权价格变化为-3.92 美元；当内层期权为美式期权时，腾讯股价每上涨 1 美元，期权价格变化为-4.13 美元。

5.3 其他希腊字母分析

5.3.1 Vega(V)

$Vega$ 表示的时期权隐含波动率变化引起的期权价格变化，一般计算方式为：

$$Vega = \frac{\partial V}{\partial \sigma} \tag{18}$$

平值期权的 $Vega$ 最高，深度实值和深度虚值的期权 $Vega$ 接近 0。

5.3.2 Theta(θ)

$Theta$ 表示期权的时间价值随时间流逝损耗的速度，一般的计算方式是：

$$Theta = \frac{\partial V}{\partial T} \tag{19}$$

时间价值接近到期日时逐渐趋近于 0。但对于期权买方来说，由于期权价值随着到期日临近、时间价值减少而损耗。所以无论期权的买方持有的是看涨期权还是看跌期权，距离到期日剩余时间越长，期权的价值越高，所以一般 $Theta$ 都是负数。

5.3.3 其他希腊字母敏感性分析结果

表 4　其他希腊字母分析

希腊字母	内层期权为欧式	内层期权为美式
Vega	230.722	236.121750
Theta	-82.380068	-86.243505

由上表可得，当内层期权为欧式时，波动率增加 1%，期权价格变化 2.3072 美元；当内层期权是美式时，波动率增加 1%，期权价格变化 2.3612 美元.。不论内层为美式还是欧式期权，随着到期日的接近，期权价值以同样的速率在减少。

六、参考文献

[1] Hull, J. C. (2022). *Options, Futures, and Other Derivatives (11th ed.). Pearson Education.*

[2] Cox, J. C., Ross, S. A., & Rubinstein, M. (1979). *Option pricing: A simplified approach. Journal of Financial Economics,* 7(3), 229-263.

[3] Longstaff, F. A., & Schwartz, E. S. (2001). *Valuing American Options by Simulation: A Simple Least-Squares Approach. The Review of Financial Studies,* 14(1), 113-147.

七、附 录

7.1 二项式定价代码

```
import pandas as pd
import numpy as np

from parameters import *

# MFE5130 2025 奇异期权定价 二项式模型
def binominal_priceX(style):
    #===生成二叉树列索引矩阵===
    mu = np.arange(M + 1)
    mu = np.resize(mu, (M + 1, M + 1))
    md = np.transpose(mu)
    #print("二叉树列索引矩阵 md:\n", md)
    mu = u ** (mu - md)
    md = d ** md
    S = S0 * mu * md

    iv = np.zeros_like(S)
    iv[:,M] = np.maximum(K2[5] - S[:,M]**1.5, 0)
    #===计算内层期权价值矩阵===
    #iv = np.maximum(K2[5] - S**1.5, 0)  # 计算内层期权价值，欧式看涨期权
    z = 0
    for j in range(M - 1, M-M_inner-1, -1):
        holding_value = (p * iv[0:M - z, j + 1] + (1 - p) * iv[1:M - z + 1, j + 1]) * df
        if style == 'American':
            exercise_value = np.maximum(K2[5] - S[0:M-z,j]**1.5, 0)
            iv[0:M-z, j] = np.maximum(holding_value,exercise_value)
        elif style == 'European':
            iv[0:M-z, j] = holding_value
        z += 1
    #print("内层期权在 T 日的价格 C(T):\n", iv[M - M_inner])
    #计算内层期权在 T 时刻的价格平均值和中位数以分析得出合理的 K1
    # V1_T_mean = np.mean(iv[:,M_outer])
    # V1_T_median = np.median(iv[:,M_outer])
    # print(f"内层期权在 T 时刻的价格平均值为:{V1_T_mean},价格中位数为
{V1_T_median}")
    # V1_t_mean = V1_T_mean * np.exp(-rf*C_outer)
    # V1_t_median = V1_T_median * np.exp(-rf*C_outer)
    # print(f"内层期权在 t 时刻的价格平均值为:{V1_t_mean},价格中位数为
{V1_t_median}")
```

```
    #===计算内层期权价值矩阵===
    #===计算第 M_outer 步后的内层期权价格列===
    iv[0:M_outer+1,M_outer] = np.maximum(N1*(iv[0:M_outer+1,M_outer] -
K1),0)
    #===计算第 M_outer 步后的内层期权价格列===



    #===计算外层看涨期权价值矩阵===
    z = 0
    for i in range(M_outer-1,-1,-1):
        iv[0:M_outer-z,i] = (iv[0:M_outer-z,i+1]*p + iv[1:M_outer-z+1,i+1]*(1-p)) *
df
        z +=1
    #print("外层期权在 t 日的价格 C(t):\n", iv[0,0])
    priceX = iv[0,0]
    #===计算外层看涨期权价值矩阵===
    return priceX

a = binominal_priceX('European')
b = binominal_priceX('American')
print(f"内层为欧式期权的 X 的二项式定价为:{a}")
print(f"内层为美式期权的 X 的二项式定价为:{b}")

print(f"欧式期权的价格变化率为{100*(a/48.43 - 1):.2f}%")
print(f"美式期权的价格变化率为{100*(b/51.18 - 1):.2f}%")
```

7.2 蒙特卡洛模拟定价代码

```
import pandas as pd
import numpy as np

from tqdm import tqdm
from parameters import *
from utils import *
# MFE5130 2025  奇异期权定价  蒙特卡洛模型

#重新命名参数


def monte_carlo_priceX(style):
    #==生成 N_outer 个外层期权价格 S_T==
    #创建一个随机数生成器 rng
    rng = np.random.default_rng(seed)
```

```
    Z = rng.standard_normal(N_outer)
    muT = rf - q - 0.5 * volatility**2
    sigT = volatility*np.sqrt(C_outer)
    S_T = S0*np.exp(muT*C_outer + sigT*Z) #形状 (N_outer,)

    # muC = rf - q - 0.5 * volatility**2
    # sigC = volatility*np.sqrt(C_inner)
    # discount_C = np.exp(-rf*C_inner)
    discount_T = np.exp(-rf*C_outer)
    discounted_payoffs = np.zeros(N_outer)
    #V1_Ts = []
    #遍历每一个外层价格 ST 求这个 ST 对应的 Px
    for j in tqdm(range(N_outer), desc='计算中'):
        s_t = S_T[j]
        #z_inner = rng.standard_normal(N_inner)
        if style=='European':
            # S_TC = s_t * np.exp(muC*C_inner + sigC * z_inner) #得到 N_inner 个
T+C 时刻的股价
            # #计算 T+C 时刻的 N_inner 个内层期权收益
            # payoff_inner = np.maximum(K2[5] - S_TC ** 1.5, 0)
            # #计算 T 时刻内层期权的价格
            # V1_T = payoff_inner.mean() * discount_C
            #V1_Ts.append(V1_T)
            params = {
                "S0": s_t,          # 初始股价 (Initial Stock Level)
                "K": K2[5],          # 行权价格 (Strike Price)
                "T": C_inner,          # 到期时间 (Time-to-Maturity)
                "r": rf,          # 无风险利率 (Short Rate)
                "sigma": volatility,       # 波动率 (Volatility)
                "q": q, # 股息收益率

                # --- 模拟参数 ---
                "I": N_inner,          # 模拟路径数量
            }
            V1_T = European_MC_calculation(params)
        elif style=='American':
            params = {
                # --- 模型参数 ---
                "S0": s_t,          # 初始股价 (Initial Stock Level)
                "K": K2[5],          # 行权价格 (Strike Price)
                "T": C_inner,          # 到期时间 (Time-to-Maturity)
                "r": rf,          # 无风险利率 (Short Rate)
                "sigma": volatility,       # 波动率 (Volatility)
                "q": q, # 股息收益率
```

```
            # --- 模拟参数 ---
            "I": N_inner,          # 模拟路径数量
            "M": 100,          # 时间步数量
        }
        V1_T = American_MC_calculation(params)
    else:
        print(f"style 必须是 European 和 American 中的一个!")

    #计算 T 时刻外层期权的收益
    payoff_outer_T = np.maximum(N1*(V1_T - K1),0)
    #计算 0 时刻外层期权的价格
    discounted_payoffs[j] = discount_T * payoff_outer_T

  #分析内层期权 T 时刻的价格，以正确模拟 K1
  # V1_Ts = np.array(V1_Ts)
  # V1_T_mean = np.mean(V1_Ts)
  # V1_T_median = np.median(V1_Ts)

  # print(f"内层期权在 T 时刻的价格平均值为:{V1_T_mean},价格中位数为
{V1_T_median}")

  price_X = discounted_payoffs.mean()
  #print(f"外层{N_outer}步内层{N_inner}步的蒙特卡洛定价法得到期权 X 的
价格为:{price_X}")
  return price_X


print(f"内层为欧式期权的 X 的蒙特卡洛模拟定价
为:{monte_carlo_priceX('European')}")
print(f"内层为美式期权的 X 的蒙特卡洛模拟定价
为:{monte_carlo_priceX('American')}")
```