

10 显式有限差分法计算期权价格

使用申明*

2021 年 2 月 23 日

目录

1 简介	1
1.1 有限差分法	1
1.2 显式差分法计算期权价格	1
2 计算美式看跌期权价格步骤	2
3 步骤 Python 代码实现	2
4 参考资料	3

1 简介

1.1 有限差分法

将微分方程表示为定义在离散的格点上的差分方程，由给定的边界条件通过相近格点间的差分关系迭代计算出未知边界上的数值。显式和隐式差分法的区别主要在于替代计算微分方程的差分方程中对微分项的离散化近似表示不同（主要是 $\frac{\partial f}{\partial t}$ 项的表示不同）。使得显式差分法可以直接由已知边界条件递推出目标边界上的数值，而隐式差分法中则不断需要解方程组进行递推计算。但显式差分法计算出的结果不一定稳定，小的误差可能被放大使得计算出的结果不对。

1.2 显式差分法计算期权价格

我们知道期权价格满足微分方程：

$$\frac{\partial f}{\partial t} + rS \frac{\partial f}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = rf. \quad (1)$$

将其写成差分形式：

$$rf(i-1, j) - \frac{f(i, j) - f(i-1, j)}{\Delta t} = rS(i, j) \frac{f(i, j+1) - f(i, j-1)}{2\Delta S} + \frac{1}{2} \sigma^2 S^2(i, j) \frac{f(i, j+1) + f(i, j-1) - 2f(i, j)}{\Delta S^2}. \quad (2)$$

其中 $f(i, j) = f(i\Delta t, j\Delta S)$, $S(i, j) = j\Delta S$, $\Delta t = T/N$, $\Delta S = S_{max}/M$ 。

然后我们可以将 $i-1$ 时刻的 $f(i-1, j)$ 用 i 时刻的数值表示：

$$f(i-1, j) = a_j f(i, j-1) + b_j f(i, j) + c_j f(i, j+1). \quad (3)$$

$$a_j = \frac{1}{1+r\Delta t} \left(-\frac{1}{2} j \Delta t + \frac{1}{2} \sigma^2 j^2 \Delta t \right), \quad b_j = \frac{1}{1+r\Delta t} (1 - \sigma^2 j^2 \Delta t), \quad (4)$$

$$c_j = \frac{1}{1+r\Delta t} \left(\frac{1}{2} j \Delta t + \frac{1}{2} \sigma^2 j^2 \Delta t \right). \quad (5)$$

*作者不对内容正确性负责。如果您希望使用部分内容作为报告、文章内容等，请您注明内容来源为“金融工程资料小站”网站。

这样我们就可以从期权执行时刻开始一列一列往回递推期权价格。需要说明的是，对于股价的上下边界 S_{max} 和 0，如果我们考虑欧式看跌期权，则 S_{max} 只要足够大（比如 $3S_0$ ），则上边界出期权价格都可以近似为 0，下边界 $S = 0$ 处期权价格都为执行价格 K 。欧式看涨期权的价格可以由看跌-看涨平价关系间接得出。对于美式看涨期权，由于其总是不会被提前执行，所以其价格等于欧式看涨，对于美式看跌期权，则只需要在价格递推时额外考虑应不应该在当前格点行使期权。

2 计算美式看跌期权价格步骤

1. 确定时间和股价变化范围， $0 \leq t \leq T$, $0 \leq S \leq S_{max}$, $S_{max} = 3S_0$ ，并将区间离散化为二维等间距网格点， $\Delta t = T/N$, $\Delta S = S_{max}/M$ 。
2. 确定三个边界上的期权价格， $t = T$ 时 $f(N, j) = \max(0, K - j\Delta S)$ ， $S = 0$ 时 $f(i, 0) = K$ ， $S = S_{max}$ 时 $f(i, M) = 0$ 。
3. 从 $t = T$ 的边界开始，往回递推出上一个时刻所有格点上的期权价格。递推公式使用上面简介中给出的表达式。
4. 考虑应不应该在新计算出的格点处执行美式看跌期权，即更新 $f(i, j) = \max(f(i, j), K - j\Delta S)$ 。
5. 重复步骤 3，4，直到初始时刻，然后使用线性插值法得出 $S = S_0$ 处期权价格。

3 步骤 Python 代码实现

```
def American_put_explicit(r, sigma, S_0, K, T, M, N):
    # S_max = 3*S_0, 分为M段, T分为N段。
    dS = 3*S_0/M
    dt = T/N

    # 使用两列格点期权价格不断往回迭代计算, f1为左边列, f2为右边列。
    f1 = [max(0.0, K-i*dS) for i in range(M+1)]
    f2 = list(f1)

    for i in range(N-1, -1, -1):
        f2 = list(f1)
        # 上下边界条件。
        f1[0] = K
        f1[-1] = 0
        for j in range(1, M, 1):
            # M, N的选取应该最好使得 sigma*sigma*j*dt 小于1。
            f1[j] = f2[j-1]*(-0.5*r*j*dt+0.5*sigma*sigma*j*dt)/(1+r*dt)
            f1[j] += f2[j]*(1-sigma*sigma*j*dt)/(1+r*dt)
            f1[j] += f2[j+1]*(0.5*r*j*dt+0.5*sigma*sigma*j*dt)/(1+r*dt)
        # 判断是否应该在格点执行美式期权。
        for j in range(M+1):
            f1[j] = max(f1[j], K-j*dS)
    # 线性插值得期权初始时刻在S_0处的价格。
    pos = int(S_0/dS)
    put_price = f1[pos]+(f1[pos+1]-f1[pos])/dS*(S_0-dS*pos)

    return put_price

if __name__ == "__main__":
```

4 参考资料	3
put_price = American_put_explicit(0.1, 0.4, 50, 60, 1.0, 150, 50000)	30
print("American put price: {0:0.5f}".format(put_price))	31
	32
Output:	33
American put price: 12.07188	34

4 参考资料

参考文献

[1] 《期权、期货及其他衍生产品》（原书第 9 版）第 21 章，John C. Hull 著，王勇、索吾林译，机械工业出版社，2014.11 。