网格变换理论

华泰长城资管实习生吕麦多

2025 年 3 月 7 日

主要参考:Pricing Financial Instruments: The Finite Difference Method (Domingo Tavella, Curt Randall) Chapter 5 pdf P217-230。

第一部分先讨论简单情形，只加密特定点网格，不考虑经过加密点。第二部分建立在第一部分基础上讨论如何经过加密点。

Quantlib 中给出了任意非均匀网格的差分格式，将在第三部分推导。

目录

1 聚焦特定点的网格

以累计向下敲入转线性结构为例。

1.1 变换的选取

该理论较适用于敲出、敲入点，即在离散观察的障碍或雪球期权支付不连续的情况。我们希望在障碍点附近加密。考虑对股价 S 做逆变换 S=S(u)，u 满足一定的性质，使得 v=v(S(u),t,T,r,q,B,K) u 下的均匀网格即为 S 下特定点附近加密的不均匀网格，假设 J(u)=$\frac{\partial S}{\partial u}$，

为了方便考虑边界条件的一致性，我们要求

$$S(u):[S_{min},S_{max}]\to[S_{min},S_{max}]$$

由此转化后的方程依然满足边界条件

$$v(S_{min}=ki\ barrier,t)=Forward(kibarrier,t,T,r,q),v(S_{max},t)=0$$

为实现加密效果，规定

$$J(u)=A*\Pi_{i=1}^{n}\sqrt{\alpha_i^2+(S(u)-S_i)^2}$$

其中 S_i 是需要加密的股价点，在 S_i 附近，J(u) 小，因此 u 的均匀网格对应的 S(u) 存在更多 S_i 附近的点，实现网格加密。J(u)=$\frac{\partial S}{\partial u}$，$\alpha_i$ 是可以自行调整的参数，A 可以由边界条件程序自动解出，S=S(u) 可以由 ODE 求解器求出。

1.2 参数对网格选取的影响

注意到

$$dS=J(u)du$$

若 J(u) 越小，dS 越小，u 下的均匀网格对应的 S 网格点越多，S 的网格越密集。因此 α_i 越小，网格点越密集，α_i 增大到无穷，则网格收敛到均匀网格。

注意本算法计算效果对于参数较为敏感。仅加密一个点可能导致计算效果不好，为了平衡总体的效果，建议在计算的定义域内放两个加密点让网格数量分布不要太集中。

1.3 参数影响的一个例子

barrier=95,ki barrier=90, strike=105, ki strike=98, 在 ki barrier=90 和 ki strike=98 两个点加密

均匀网格结果如下：

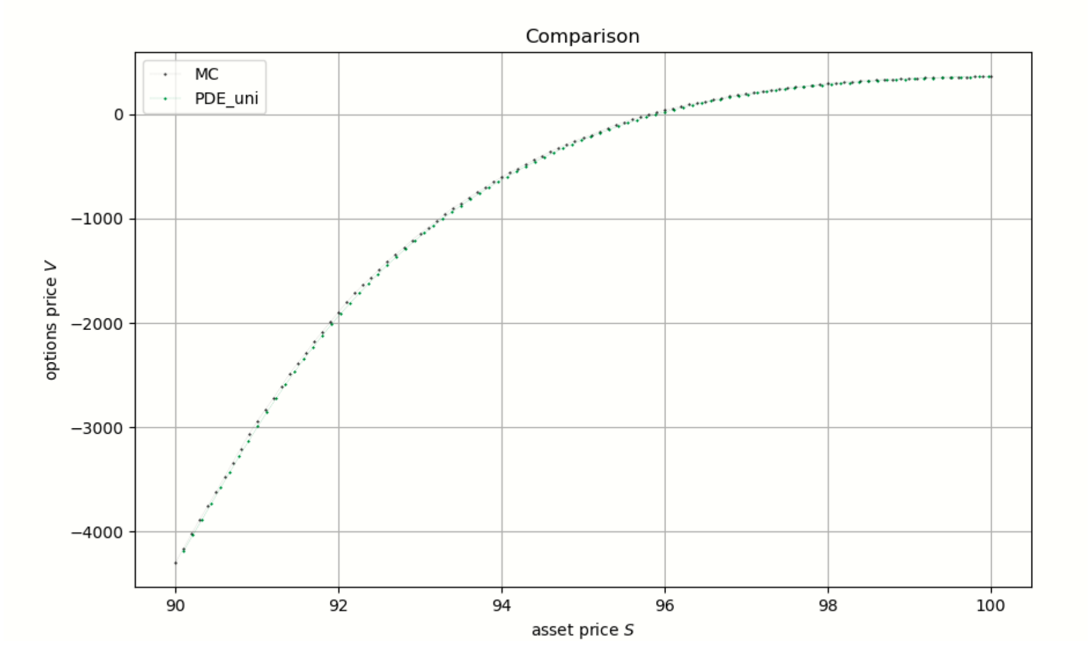

图 1: 均匀网格

取 $\alpha_i = 0.3 * S_i$ 得到不均匀网格结果如下：

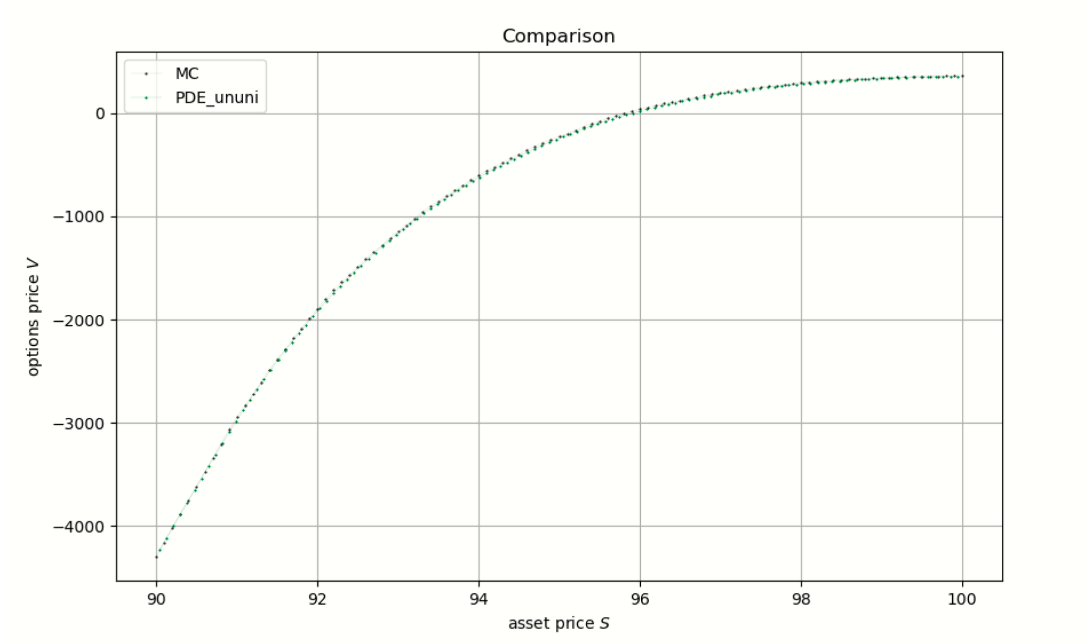

图 2: 0.3, 不均匀网格

在加密点附近不均匀网格贴合 MC 曲线更好

取 $\alpha_i = 0.1 * S_i$ 得到不均匀网格结果如下：

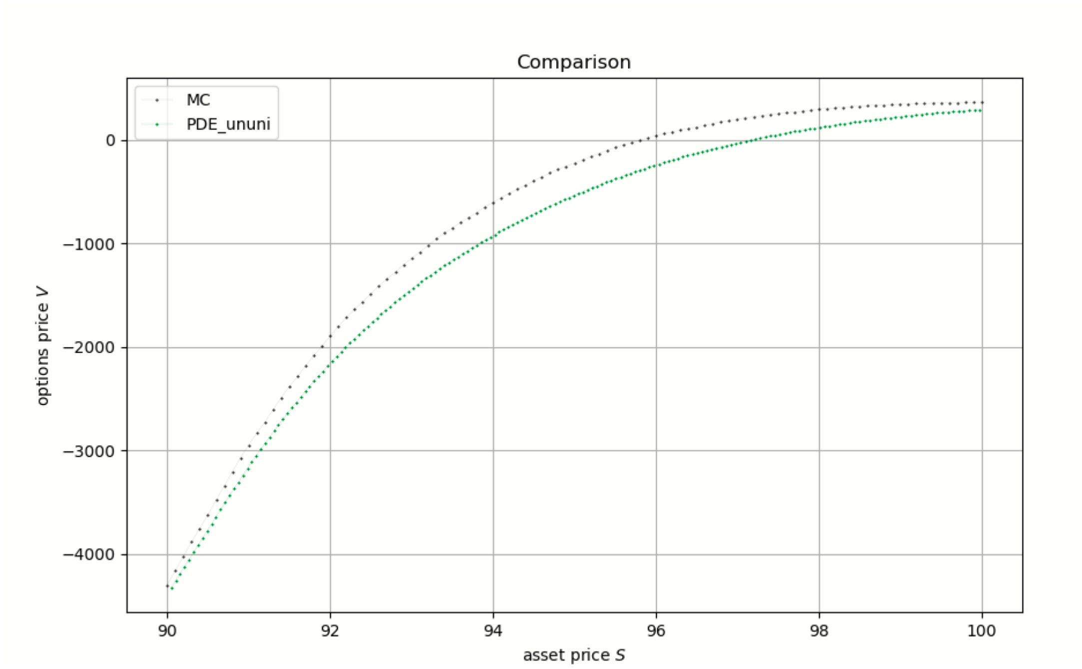

图 3: 0.1，不均匀网格

效果特别差，经验上一般取 $\alpha_i = 0.3 - 0.4 * S_i$ 效果好一些

注意，加密部分的效果好的代价是不加密部分网格稀疏，导数计算误差大。因此在远离非加密点的价格点处建议用均匀网格。此外，加密点 α_i 小，网格密并不是加密点计算准确的充分条件，还要经过调参平衡其他点网格疏密。

2 加密的同时将加密点放在网格点上

2.1 基本思路

在在第一部分中，我们得到 S 的不均匀网格 $\{S_i\}$，然而不均衡网格并未实现每个加密点都在网格上，假设股价 S 需要的加密点集为 $\{B_i\}$, 如果能将 $\{S_i\}$ 中距离 $\{B_i\}$ 中的点最近的 Spot 点略微修改一下，变为 $\{B_i\}$，就可以实现这样的效果。

2.2 具体实现

在不均匀网格点集 S_i 中找到最接近加密点 B_i 的点对应的网格点原象，对其原象加以修改变为加密点的原象 $\{b_i\}$，即伸缩网格

$$S_i = \begin{cases} B_i & \text{如果存在 } B_i\text{，使得 } |S_i - B_i| \text{ 最小} \\ S_i & \text{其他} \end{cases}$$

由此，我们完成了对网格 S 的构建。

3　非均匀网格下的价格数值解

3.1　任意非均匀网格的一阶导差分格式

本部分讨论为了记号简便舍去时间记号 t，只讨论价格关于股价的导数差分格式。假设网格点集为 $\{S_i\} \quad i \in \{0, 1...n\}$，对应的价格为 $v(s_i)$，先考虑一阶导数的计算 $\frac{\partial v}{\partial S}$。

对于网格的左端点，只能用向后差分，即

$$\frac{\partial v}{\partial S}|_{s=s_0} = \frac{v_1 - v_0}{s_1 - s_0}$$

对于网格的右端点，只能用向前差分，即

$$\frac{\partial v}{\partial S}|_{s=s_{-1}} = \frac{v_{-1} - v_{-2}}{s_{-1} - s_{-2}}$$

对于其他点，考虑中心差分：

记 $\Delta S_i = S_i - S_{i-1}, \Delta S_{i+1} = S_{i+1} - S_i$, 由泰勒展开有

$$v(S_{i+1}) = v(S_i) + \Delta S_{i+1} v'(S_i) + \frac{1}{2}(\Delta S_{i+1})^2 v''(S_i) + o((\Delta S_{i+1})^2)$$

$$v(S_{i-1}) = v(S_i) - \Delta S_i v'(S_i) + \frac{1}{2}(\Delta S_i)^2 v''(S_i) + o((\Delta S_i)^2)$$

消去 $v''(S_i)$ 得到：

$$v'(S_i) = -\frac{\Delta S_{i+1}}{(\Delta S_i + \Delta S_{i+1})\Delta S_i} v(S_{i-1}) + \frac{\Delta S_{i+1} - \Delta S_i}{\Delta S_i \Delta S_{i+1}} v(S_i) + \frac{\Delta S_i}{(\Delta S_i + \Delta S_{i+1})\Delta S_{i+1}} v(S_{i+1}) \; 1 \le i \le n-1$$

记

$$a_i = -\frac{\Delta S_{i+1}}{(\Delta S_i + \Delta S_{i+1})\Delta S_i}, b_i = \frac{\Delta S_{i+1} - \Delta S_i}{\Delta S_i \Delta S_{i+1}}, c_i = \frac{\Delta S_i}{(\Delta S_i + \Delta S_{i+1})\Delta S_{i+1}}$$

记

$$\vec{v} = (v(S_i))_{i\in\{0,1...n\}}, \vec{v'} = (v'(S_i))_{i\in\{0,1...n\}}$$

$$\vec{v'} = A\vec{v}$$

其中

$$A = \begin{pmatrix} -\frac{1}{\Delta S_1} & \frac{1}{\Delta S_1} & 0 & ... & ... & ... & ... & 0 \\ a_1 & b_1 & c_1 & 0 & ... & ... & ... & 0 \\ 0 & a_2 & b_2 & c_2 & 0 & ... & ... & 0 \\ \vdots & ... & ... & ... & ... & ... & ... & \vdots \\ 0 & ... & ... & ... & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & ... & ... & ... & ... & 0 & -\frac{1}{\Delta S_{-1}} & \frac{1}{\Delta S_{-1}} \end{pmatrix}$$

实际参与计算时不一定会用到第一个和最后一个的导数。

3.2 任意非均匀网格的二阶导差分格式

第一个和最后一个不存在二阶导，只考虑可以中心差分的部分。

$$v(S_{i+1}) = v(S_i) + \Delta S_{i+1} v'(S_i) + \frac{1}{2}(\Delta S_{i+1})^2 v''(S_i) + o((\Delta S_{i+1})^2)$$

$$v(S_{i-1}) = v(S_i) - \Delta S_i v'(S_i) + \frac{1}{2}(\Delta S_i)^2 v''(S_i) + o((\Delta S_i)^2)$$

消去 $v'(S_i)$ 得到：

$$v''(S_i) = \frac{2}{\Delta S_i(\Delta S_i + \Delta S_{i+1})} v_{i-1} - \frac{2}{\Delta S_i \Delta S_{i+1}} v_i + \frac{2}{(\Delta S_i + \Delta S_{i+1})\Delta S_{i+1}} v_{i+1}, 1 \le i \le n-1$$

记

$$e_i = \frac{2}{\Delta S_i(\Delta S_i + \Delta S_{i+1})}, f_i = -\frac{2}{\Delta S_i \Delta S_{i+1}}, g_i = \frac{2}{(\Delta S_i + \Delta S_{i+1})\Delta S_{i+1}}$$

为了完整起见，记

$$\vec{v} = (v(S_i))_{i\in\{0,1\dots n\}}, \vec{v''} = (v''(S_i))_{i\in\{0,1\dots n\}}$$

$$\vec{v''} = B\vec{v}$$

其中

$$B = \begin{pmatrix} 0 & 0 & 0 & \dots & \dots & \dots & \dots & 0 \\ e_1 & f_1 & g_1 & 0 & \dots & \dots & \dots & 0 \\ 0 & e_2 & f_2 & g_2 & 0 & \dots & \dots & 0 \\ \vdots & \dots & \dots & \dots & \dots & \dots & \dots & \vdots \\ 0 & \dots & \dots & \dots & 0 & e_{n-1} & f_{n-1} & g_{n-1} \\ 0 & \dots & \dots & \dots & \dots & 0 & 0 & 0 \end{pmatrix}$$

3.3 非均匀网格的隐式差分方程

由隐式条件, 可以写成以下向量方程:

$$\frac{\vec{V}_{t+1} - \vec{V}_t}{\Delta t} + \frac{1}{2}\sigma^2 S^2 B\vec{V}_t + (r-q)SA\vec{V}_t - r\vec{V}_t = 0$$

即

$$\vec{V}_{t+1} = [E + r\Delta t E - \frac{1}{2}\sigma^2 S^2 \Delta t B - (r-q)S\Delta t A]\vec{V}_t$$

该方法适用于任意的网格，可以证明：均匀网格时的方程为该方程特殊形式。

4 Quantlib 中的实现代码链接

一阶导数实现的代码链接：

`https://rkapl123.github.io/QLAnnotatedSource/d4/d95/firstderivativeop_8cpp_source.html`

二阶导数实现的代码链接：

`https://rkapl123.github.io/QLAnnotatedSource/d9/db7/secondderivativeop_8cpp_source.html`

加密卡点网格实现的代码链接: `https://rkapl123.github.io/QLAnnotatedSource/da/d12/concentrating1dmesher_8cpp_source.html`