文章编号:1000-5862(2013)04-0416-05

求解美式期权定价问题的两类新的迭代算法

刘　哲,孙　哲*,黄晓梅
(江西师范大学数学与信息科学学院,江西 南昌　330022)

摘要:提出了 2 类改进的局部策略迭代算法求解一类美式期权定价模型离散得到的优化控制差分方程组,证明了算法的收敛性.数值实验表明了算法的有效性.

关键词:美式期权;转换模型;策略迭代;局部策略迭代
中图分类号:O 211.6;O 221.1　　**文献标志码**:A

0　引言

期权是一类重要的金融衍生工具,它的定价模型取决于标的资产价格的演化模型.在连续时间情形下,标的资产的价格演化可以通过一个随机微分方程来描述,从而作为它的衍生物,期权价格可以用一个偏微分方程来刻画[1].美式期权可以在到期日或之前的任一交易日执行,而欧式期权只能在到期日执行.因此,美式期权定价问题的数学模型一般可归结为一个自由边界问题或线性互补问题.自由边界又称为最佳执行边界.在美式期权定价问题中,最佳执行边界必须作为定价模型解的一部分来求解,这使得美式期权定价问题的求解具有相当的难度.

本文考虑基于 Black-Scholes 模型和马尔科夫状态转换模型的美式期权定价模型[2-3]的数值解法.对于各类美式期权定价模型离散后得到的线性互补问题,现在已经有许多有效的算法可以对其进行求解[4-8].最近,P. A. Forsyth 等[9]提出了罚方法来解美式期权定价问题,他们先把相应的线性互补问题转化为 1 个罚方程,然后采用半光滑 Newton 法求解罚方程. Y. Huang 等利用隐格式的有限差分法对罚方程进行离散,得到 1 个优化控制差分方程组,该方程组是 1 个非线性方程组.策略迭代算法是解此类方程组的有效方法之一.文献[2]提出了策略迭代、不动点策略迭代和局部策略迭代等算法来解该方程组,其中前 2 个算法在每次迭代需要解 1 个线性方程组,而局部策略迭代算法在每次迭代需要解 1 个非线性方程组,这都是比较耗时的.

本文将采用罚方法来求解美式期权定价问题,并提出 2 类改进的局部策略迭代算法来求解离散罚方程,在一定的条件下证明了算法的收敛性.数值实验表明这 2 个改进算法比不动点策略迭代算法更加有效.

1　模型的建立与离散

1.1　模型的建立

设 S 为标的资产的市场价格,T 为期权的到期日,E 为期权的执行价格,$\tau = T - t$,$\sigma_j(j = 1,2,\cdots,K)$ 为标的资产价格的波动率,$V_j(S,\tau)$ 为在状态 j 时期权的无套利价格.为了建立期权定价问题的数学模型,定义下列微分算子

$$L_jV_j = \frac{1}{2}\sigma_j^2S^2\frac{\partial^2V_j}{\partial S^2} + (r - \rho_j)S\frac{\partial V_j}{\partial S} - (r + \lambda_j)V_j, \tag{1}$$

$$J_jV = \sum_{k=1,k\neq j}^{K}\frac{\lambda_{jk}}{\lambda_j}V_k(\xi_{jk}S,\tau), \tag{2}$$

其中 r 为无风险利率(假定为常数),$(\xi_{jk}) \in \mathbf{R}^{K\times K}$ 为状态转换矩阵,当状态 $j \to k$ 时,标的资产的价格 $S \to \xi_{jk}S$,$\boldsymbol{\lambda} \in \mathbf{R}^{K\times K}$ 为状态转移概率矩阵,且 $\lambda_{jk} \geqslant 0$ $(j \neq k)$,

$$\lambda_{jj} = -\sum_{k=1,k\neq j}^{K}\lambda_{jk}, \rho_j = \sum_{k=1,k\neq j}^{K}\lambda_{jk}(\xi_{jk} - 1),$$

$$\lambda_j = \sum_{k=1,k\neq j}^{K}\lambda_{jk}.$$

美式期权的价格满足线性互补微分方程

$$\min\left(V_{j,\tau} - L_jV_j - \lambda_jJ_jV, V_j - V^*\right) = 0, j = 1,2,$$

收稿日期:2013-03-05
基金项目:国家自然科学基金(11126147,11201197)资助项目.
通信作者:孙　哲(1983-),男,湖南邵阳人,副教授,博士,主要从事偏微分方程数值解法的研究.

$\cdots, K$，　(3)

其中$(S,\tau) \in [0,\infty) \times [0,T]$，$V^*(S)$为期权的收益函数. 本文考虑美式看跌期权，则

$$V^*(S) = \max(E - S, 0),$$

为了进行数值计算，在区域$[0, S_{\max}] \times [0, T]$上对方程(3)进行求解. 当$S = 0$时，期权价格满足方程(3)；当$S = S_{\max}$与$\tau = 0$时，期权价格分别为

$$V_j(S_{\max}, \tau) = 0, V_j(S, 0) = V^*(S), \quad (4)$$
$$S \in [0, S_{\max}], \tau \in [0, T], j = 1, 2, \cdots, K.$$

在数值计算中，如果$\xi_{jk} S > S_{\max}$，则用$V_k(S_{\max}, \tau)$来替换公式(2)中$V_k(\xi_{jk} S, \tau)$. 文献[10]指出了当$S_{\max}$足够大时，误差会比较小.

1.2　离散化

令$h = S_{\max} / i_{\max}$，$S_i = ih\ (i = 0, 1, \cdots, i_{\max})$，$\tau^n\ (n = 0, 1, \cdots, p)$表示第$n$个时间步，$\Delta\tau$为时间间隔，$i_{\max}$和$p$为正整数. 设$L_j^h, J_j^h$分别为算子$L_j, J_j$的离散形式，$V_{i,j}^n$为期权在状态$j$点$(S_i, \tau^n)$处的价格，定义$N = K \times i_{\max}$维列向量

$$\boldsymbol{V}^n = [V_{0,1}^n, \cdots, V_{i_{\max}-1,1}^n, \cdots, V_{0,K}^n, \cdots, V_{i_{\max}-1,K}^n]^{\mathrm{T}}.$$

若记$l = (j-1) \times i_{\max} + i$，则$V_{i,j}^n = V_l^n$.

为了确保微分算子L_j离散后得到的矩阵为M矩阵，本文采用文献[2]中的正系数离散化方法来离散(1)，即

$$[L_j^h \boldsymbol{V}^n]_{i,j} = \alpha_{i,j} V_{i-1,j}^n - (\alpha_{i,j} + \beta_{i,j} + r + \lambda_j) V_{i,j}^n + \beta_{i,j} V_{i+1,j}^n, \quad (5)$$

其中$\alpha_{i,j} \geqslant 0$，$\beta_{i,j} \geqslant 0$. 用线性插值的方法离散算子(2)，即

$$[J_j^h \boldsymbol{V}^n]_{i,j} = \sum_{k=1, k \neq j}^{K} \frac{\lambda_{jk}}{\lambda_j} I_{i,j,k}^h \boldsymbol{V}^n, \quad (6)$$

$$I_{i,j,k}^h \boldsymbol{V}^n = \omega V_{m,k}^n + (1 - \omega) V_{m+1,k}^n \cong V_k(\min(S_{\max}, \xi_{jk} S_i), \tau^n), \omega \in [0, 1]. \quad (7)$$

由文献[2]可知方程(3)的罚方程为

$$V_{j,\tau} - L_j V_j - \lambda_j J_j V + \min_{\varphi \in \{0,1\}} \left[\frac{\varphi(V_j - V^*)}{\varepsilon} \right] = 0, \quad (8)$$

其中罚参数$0 < \varepsilon \ll 1$. 用正系数离散化方法对上式离散后可得

$$\frac{V_{i,j}^{n+1}}{\Delta\tau} - \theta L_j^h V_{i,j}^{n+1} + \frac{\varphi_{i,j}^{n+1}}{\varepsilon} V_{i,j}^{n+1} = \frac{V_{i,j}^n}{\Delta\tau} + \frac{\varphi_{i,j}^{n+1}}{\varepsilon} V_i^* + \lambda_j \theta [J_j^h V^{n+1}]_{i,j} + (1 - \theta) [L_j^h V_{i,j}^n + \lambda_j [J_j^h \boldsymbol{V}^n]_{i,j}],$$
$$j = 1, 2, \cdots, K, n = 0, 1, 2, \cdots, p - 1, \quad (9)$$

其中$\varphi_{i,j}^{n+1} \in \arg \min_{\varphi \in \{0,1\}} \{\varphi(V_{i,j}^{n+1} - V_i^*) / \varepsilon\}$. 当$\theta = 1/2$与$\theta = 1$时，上述方程分别对应于C-N格式与完全隐格式. 初边值条件(4)的离散形式为

$$V_{i_{\max},j}^n = 0, V_{i,j}^0 = V_i^*.$$

由(9)式可知，美式期权定价问题在每个时间步需要解1个优化控制差分方程组. 该方程组可表述为

$$\boldsymbol{A}^*(Q) \boldsymbol{U} = \boldsymbol{C}(Q), \quad (10)$$
$$Q = \arg \min_{Q \in Z} [\boldsymbol{A}^*(Q) - \boldsymbol{C}(Q)],$$

其中$\boldsymbol{A}^*$为$N \times N$的矩阵，$\boldsymbol{U}, \boldsymbol{C}$是$N \times 1$的向量，$Z = \{0, 1\}^N$为控制集，并且$A_l^*$，$C_l$仅与$Q_l$有关. 若记$\boldsymbol{A}(Q)$为$V_{j,\tau} - L_j V_j$离散后对应的矩阵，$\boldsymbol{B}(Q)$为$J_j\ (j = 1, 2, \cdots, K)$离散后对应的矩阵，则

$$\boldsymbol{A}^*(Q) = \boldsymbol{A}(Q) - \boldsymbol{B}(Q).$$

下面给出2个重要的命题.

命题 1[11]　假设$\boldsymbol{A}(Q)$由正系数离散方法(5)得到，$\boldsymbol{B}(Q)$由线性插值(6)和(7)得到，那么，下面2个结论成立：

(ⅰ) $\boldsymbol{B}(Q) \geqslant 0$;

(ⅱ) 矩阵$\boldsymbol{A}^*(Q)$和$\boldsymbol{A}(Q)$均为严格对角占优的M矩阵.

命题 2　假设$\boldsymbol{A}, \boldsymbol{B}$为$N \times N$的矩阵，且$\boldsymbol{A}$严格对角占优，$\boldsymbol{B} \geqslant 0$，则

$$\| \boldsymbol{A}^{-1} \boldsymbol{B} \|_\infty \leqslant \max_l \left\{ \sum_u B_{l,u} / \sum_u A_{l,u} \right\}.$$

2　算法及其收敛性

下面提出2个新的迭代算法来解(10)式. 由于新算法在每次迭代只需要执行一些算术运算，因此，与不动点策略迭代算法相比，新算法减少了计算量.

考虑矩阵分裂

$$\boldsymbol{A}^*(Q) = \boldsymbol{D}(Q) - \boldsymbol{F}(Q), \quad (11)$$

其中$\boldsymbol{D}(Q)$是$\boldsymbol{A}^*(Q)$的对角矩阵. 则由命题1知$\boldsymbol{D}(Q)$与$\boldsymbol{F}(Q)$均为非负矩阵，且$\boldsymbol{D}(Q)$的对角元素大于0. 下面给出改进的局部策略迭代算法.

算法 1

步骤 1　给定初始迭代值$\boldsymbol{U}^0$，正常数tolerance，令$k := 0$;

步骤 2　解$\min_{Q \in Z} [\boldsymbol{D}(Q) \boldsymbol{U}^{k+1} - \boldsymbol{F}(Q) \boldsymbol{U}^k - \boldsymbol{C}(Q)] = 0$;

步骤 3　若$k > 0$且$\max_{1 \leqslant l \leqslant N} \frac{|U_l^{k+1} - U_l^k|}{\max[\text{scale}, |U_l^{k+1}|]} <$ tolerance，则算法终止；否则，令$k := k + 1$，转至步骤2.

下面将给出算法1的收敛性定理.

定理 1　如果$\boldsymbol{A}^*(Q)$为M矩阵，并且存在正常数$C_1 < 1$使得

$$\max_{Q \in Z} \| \boldsymbol{D}(Q)^{-1} \boldsymbol{F}(Q) \|_\infty \leqslant C_1,$$

那么算法 1 收敛. 此外,若 U^* 为方程组(10)的解,则

$$\|E^{k+1}\|_\infty \leqslant C_1 \|E^k\|_\infty,$$

其中 $E^k = U^k - U^*$.

证 设 U^* 为方程组(10)的解,则 $\exists Q' \in Z$ 使得

$$D(Q')U^* - F(Q')U^* - C(Q') = 0, \quad (12)$$

而由算法 1 知,

$$\min_{Q\in Z}[D(Q)U^{k+1} - F(Q)U^k - C(Q)] = 0, \quad (13)$$

那么

$$D(Q')U^{k+1} - F(Q')U^k - C(Q') \geqslant 0. \quad (14)$$

由(14)式减去(12)式可得

$$D(Q')(U^{k+1} - U^*) - F(Q')(U^k - U^*) \geqslant 0,$$

即

$$D(Q')E^{k+1} - F(Q')E^k \geqslant 0.$$

由于 $\forall Q \in Z$,$D(Q)^{-1} > 0$. 于是有

$$E^{k+1} \geqslant D(Q')^{-1}F(Q')E^k \geqslant -C_1\|E^k\|_\infty e, \quad (15)$$

其中 $e = (1,1,\cdots,1)^{\mathrm{T}}$. 另一方面,由(13)式可知,$\exists \bar{Q} \in Z$ 使得

$$D(\bar{Q})U^{k+1} - F(\bar{Q})U^k - C(\bar{Q}) = 0.$$

从而

$$0 = D(\bar{Q})U^{k+1} - F(\bar{Q})U^k - C(\bar{Q}) - \min_{Q\in Z}[D(Q)U^* - F(Q)U^* - C(Q)] \geqslant D(\bar{Q})U^{k+1} - F(\bar{Q})U^k - C(\bar{Q}) - [D(\bar{Q})U^* - F(\bar{Q})U^* - C(\bar{Q})] = D(\bar{Q})E^{k+1} - F(\bar{Q})E^k.$$

故

$$E^{k+1} \leqslant D(\bar{Q})^{-1}F(\bar{Q})E^k \leqslant C_1\|E^k\|_\infty e. \quad (16)$$

结合(15)与(16)两式可知

$$\|E^{k+1}\|_\infty \leqslant C_1\|E^k\|_\infty.$$

从而算法 1 收敛. 定理 1 得证.

对于离散方程(9),将验证定理 1 中的条件成立.

定理 2 对于离散方程(9),矩阵分裂(11)式满足定理 1 的条件,即存在正常数 $C_1 < 1$ 使得

$$\max_{Q\in Z}\|D(Q)^{-1}F(Q)\|_\infty \leqslant C_1.$$

证 由(9)式可知,$\forall Q \in Z$,有矩阵 $D(Q)$ 与 $F(Q)$ 的第 l 行的行和分别为

$$Row\_Sum_l(D(Q)) = \frac{1}{\Delta\tau} + \theta(\alpha_l + \beta_l + r + \lambda_j) + \frac{\varphi_l}{\varepsilon}, i < i_{\max},$$

$$Row\_Sum_l(F(Q)) = \theta(\alpha_l + \beta_l + \lambda_j), i < i_{\max}.$$

从而,根据命题 2 得

$$\|D(Q)^{-1}F(Q)\|_\infty \leqslant \max_{1\leqslant l\leqslant N}\left\{\frac{\sum_u F_{l,u}}{\sum_u D_{l,u}}\right\} = \max_{\substack{1\leqslant l\leqslant N\\1\leqslant j\leqslant K}}\left\{\frac{\theta(\alpha_l + \beta_l + \lambda_j)}{\frac{1}{\Delta\tau} + \theta(\alpha_l + \beta_l + r + \lambda_j) + \frac{\varphi_l}{\varepsilon}}\right\} \leqslant \frac{\theta\eta\Delta\tau}{1 + \theta(r+\eta)\Delta\tau} < 1.$$

因此,令 $C_1 = \theta\eta\Delta\tau / [1 + \theta(r+\eta)\Delta\tau]$,其中 $\eta = \max_{\substack{1\leqslant l\leqslant N\\1\leqslant j\leqslant K}}(\alpha_l + \beta_l + \lambda_j)$,则定理 2 得证.

下面再考虑矩阵分裂

$$A^*(Q) = M(Q) - N(Q), \quad (17)$$

其中 $M(Q)$ 为 $A(Q)$ 的下三角矩阵. 由命题 1 知 $N(Q) \geqslant 0$,且 $M(Q)$ 为对角占优的 M 矩阵. 下面给出另一个改进的局部策略迭代算法.

算法 2

步骤 1 给定初始迭代值 U^0,正常数 tolerance,令 $k := 0$;

步骤 2 解 $\min_{Q\in Z}[M(Q)U^{k+1} - N(Q)U^k - C(Q)] = 0$;

步骤 3 若 $k > 0$ 且 $\max_{1\leqslant l\leqslant N}\frac{|U_l^{k+1} - U_l^k|}{\max[\text{scale}, |U_l^{k+1}|]} <$ tolerance,则算法终止; 否则,令 $k := k+1$,转至步骤 2.

类似于定理 1 和定理 2 的证明,可以得到算法 2 的收敛性定理 3 及定理 4.

定理 3 如果 $A^*(Q)$ 为 M 矩阵,并且存在正常数 $C_2 < 1$ 使得

$$\max_{Q\in Z}\|M(Q)^{-1}N(Q)\|_\infty \leqslant C_2,$$

那么算法 2 收敛. 此外,若 U^* 为方程组(10)的解,则

$$\|E^{k+1}\|_\infty \leqslant C_2\|E^k\|_\infty,$$

其中 $E^k = U^k - U^*$.

定理 4 对于离散方程(9),矩阵分裂(17)式满足定理 3 的条件,即存在正常数 $C_2 < 1$ 使得

$$\max_{Q\in Z}\|M(Q)^{-1}N(Q)\|_\infty \leqslant C_2.$$

此处取 $C_2 = \theta\zeta\Delta\tau / [1 + \theta(r+\zeta)\Delta\tau]$,其中 $\zeta = \max_{1\leqslant l,j\leqslant N}(\beta_l + \lambda_j)$ 即可.

3 数值实验

接下来通过 2 个数值实验来测试算法 1 和算法 2 的有效性. 同时,对算法 1、算法 2 和不动点策略迭代算法的计算结果进行比较. 考虑美式看跌期权(3),其相应的参数见表 1.

表 1　数值实验参数

参数	取值
K	3
T	0.5
E	100
S_{max}	500
r	0.02
ε	$10^{-6}\Delta\tau$
tolerance	10^{-8}

$$\boldsymbol{\lambda}=\begin{bmatrix}-3.20 & 0.20 & 3.00\\ 1.00 & -1.08 & 0.08\\ 3.00 & 0.20 & -3.20\end{bmatrix},$$

$$\boldsymbol{\xi}=\begin{bmatrix}1.00 & 0.90 & 1.10\\ 1.20 & 1.00 & 1.30\\ 0.95 & 0.80 & 1.00\end{bmatrix},\boldsymbol{\sigma}=\begin{bmatrix}0.20\\ 0.15\\ 0.30\end{bmatrix}.$$

本文的数值实验采用文献 [9] 中的方法生成变时间步长. 由于 $V^*(S)$ 是非光滑的,因此,为了使 C-N 格式计算结果稳定,在实验中对离散方程(9) 的前 2 个时间步采用完全隐格式,以后的时间步均采用 C-N 格式[12].

实验 1　采用 C-N 格式离散(8) ,用算法 1、算法 2 和不动点策略迭代算法分别计算期权的价格,并对计算结果中的期权价格 $V_1(100,T)$ 进行比较. 表 2 列出了计算结果. 在表 2 中,grid(m,p) 表示求解区域的划分情况,其中 m 为需要求解的格点数目,p 为时间步数,iter 表示算法在每个时间步的平均迭代次数,value 表示期权价格 $V_1(100,T)$,change 表示随着划分的细化期权价格的变化量,ratio 表示期权价格变化量的比率,cpu 表示计算所用的总时间,单位为 s.

表 2　实验 1 的有关结果

	grid(m,p)	iter	value	change	ratio	cpu(s)
算法 1	(600,121)	5.73	7.614 959			15.60
	(1 200,245)	6.55	7.617 779	0.002 820		61.16
	(2 400,490)	7.87	7.618 205	0.000 426	6.62	205.48
	(4 800,978)	9.93	7.618 293	0.000 088	4.84	988.68
算法 2	(600,121)	3.92	7.614 661			16.16
	(1 200,245)	4.14	7.617 633	0.002 972		70.75
	(2 400,490)	4.38	7.618 133	0.000 500	5.94	224.85
	(4 800,978)	5.19	7.618 259	0.000 126	3.97	1 077.81
不动点策略迭代算法	(600,121)	4.27	7.614 959			40.30
	(1 200,245)	3.78	7.617 779	0.002 820		200.57
	(2 400,490)	3.53	7.618 205	0.000 426	6.62	883.26
	(4 800,978)	3.19	7.618 293	0.000 088	4.84	2 832.63

实验 2　采用完全隐格式对(8) 进行离散,用算法 1、算法 2 和不动点策略迭代算法分别计算期权的价格,并对计算结果中的期权价格 $V_1(100,T)$ 进行比较. 其计算结果见表 3.

表 3　实验 2 的有关结果

	grid(m,p)	iter	value	change	ratio	cpu(s)
算法 1	(600,121)	7.41	7.598 406			16.42
	(1 200,245)	8.83	7.609 696	0.011 290		74.82
	(2 400,490)	11.32	7.614 209	0.004 513	2.50	223.63
	(4 800,978)	15.02	7.616 306	0.002 097	2.15	1 114.98
算法 2	(600,121)	4.62	7.597 839			17.83
	(1 200,245)	5.08	7.609 417	0.011 578		80.44
	(2 400,490)	6.00	7.614 075	0.004 658	2.49	252.70
	(4 800,978)	7.56	7.616 247	0.002 172	2.14	1 282.67
不动点策略迭代算法	(600,121)	4.70	7.598 406			44.16
	(1 200,245)	4.17	7.609 696	0.011 290		200.70
	(2 400,490)	3.68	7.614 209	0.004 513	2.50	797.08
	(4 800,978)	3.43	7.616 306	0.002 097	2.15	2 848.08

从表 2 和表 3 不难看到,对于 C-N 格式和完全隐格式,算法 1 与不动点策略迭代算法的计算结果都相同. 随着划分的细化,算法 1 和算法 2 的平均迭代次数都在增加,但这 2 个算法迭代所用的时间比不动点策略迭代所用的时间要少很多,并且算法 1 所用的时间最少. 因此,算法 1 和算法 2 比不动点策略迭代算法更高效.

对于 C-N 格式 ratio > 4 表明,实验 1 中的 3 个算法都达到 2 阶收敛,而对于完全隐格式 ratio < 4 表明,实验 2 中的 3 个算法都是 1 阶收敛的. 因此,

算法 1 和算法 2 在实验 2 中所用的时间要比实验 1 的多.

此外,随着划分的细化,算法 1 的平均迭代次数比算法 2 的多,这与 $C_2 < C_1$ 是一致的. 虽然算法 1 和算法 2 的平均迭代次数都比不动点策略迭代算法的多,但由于这 2 个算法在每步迭代只需要执行一些算术运算,而不动点策略迭代算法在每步迭代需要解 1 个线性方程组,因此,与不动点策略迭代算法相比算法 1 和算法 2 减少了计算量. 从而,算法 1 和算法 2 迭代所用时间比不动点策略迭代算法要少.

4 小结

本文考虑了基于 Black-Scholes 模型和马尔科夫状态转换模型的美式期权定价模型的数值解法. 首先采用罚方法得到线性互补问题的罚方程,然后利用隐格式的有限差分法离散罚方程,得到一个优化控制差分方程组,最后提出了 2 个改进的局部策略迭代算法来解离散方程,证明了算法的收敛性. 数值实验表明,算法 1 和算法 2 比不动点策略迭代算法的平均迭代次数多,但由于算法 1 和算法 2 在每步迭代只需要执行一些算术运算,与不动点策略迭代算法相比这 2 个算法减少了计算量,从而,算法 1 和算法 2 迭代所用的时间比不动点策略迭代算法要少很多,并且算法 1 和算法 2 在 C-N 格式下所用时间比完全隐格式少. 因此,算法 1 和算法 2 比不动点策略迭代算法更高效.

5 参考文献

[1] 姜礼尚. 期权定价的数学模型和方法 [M]. 北京: 高等教育出版社,2003.

[2] Huang Y,Forsyth P A,Labahn G. Methods for pricing American options under regime switching [J]. J Sci Comput,2011,33(5):2144-2168.

[3] Boyarchenko S,Levendorskii S. American options in regime-switching models [J]. J Control Optim,2009,48(3):1353-1376.

[4] Cryer C W. The solution of a quadratic programming problem using systematic overrelaxation [J]. SIAM J Contr,1971,9(3):385-392.

[5] Ikonen S,Toivanen J. Operator splitting methods for American option pricing [J]. Appl Math Lett,2004,17(7):809-814.

[6] Ikonen S,Toivanen J. Component wise splitting methods for pricing American options under stochastic volatility [J]. Int J Theor Appl Finance,2007,10(2):331-361.

[7] Ikonen S,Toivanen J. Efficient numberical methods for pricing American options under stochastic volatility [J]. Numer Methods Partial Differential Equations,2008,24(1):104-126.

[8] 段班详,邓洁. 线性互补问题的 SSOR 多分裂算法 [J]. 江西师范大学学报: 自然科学版,2011,35(5):459-463.

[9] Forsyth P A,Vetzal K R. Quadratic convergence for valuing American options using a penalty method [J]. J Sci Comput,2002,23(6):2095-2122.

[10] Kennedy J S. Hedging contingent claims in markets with jumps [D]. Ontario: University of Waterloo,2007.

[11] Huang Y,Forsyth P A,Labahn G. Combined fixed point and policy iteration for Hamilton-Jacobi-Bellman equations in finance [J]. J Numer Anal,2012,50(4):1861-1882.

[12] Giles M B,Carter R. Convergence analysis of Crank-Nicolson and Rannacher time-marching [J]. J Comput Finance,2006,9(4):89-112.

Two New Iterative Methods for Solving American Option Pricing Problems

LIU Zhe,SUN Zhe*,HUANG Xiao-mei

(College of Mathematics and Informatics,Jiangxi Normal University,Nanchang Jiangxi 330022,China)

Abstract: Two modified local policy iterative algorithms have been proposed for solving the optimal control difference equations arising from the American option pricing problems. The convergence of the algorithms have been also proved. Numerical experiments show the effectiveness of the algorithms.

Key words: American option; regime switching model; policy iteration; local policy iteration

(责任编辑: 曾剑锋)