9

Free-Boundary Problems

As we know, a problem of pricing an American-style derivative can be formulated as a linear complementarity problem, and for most cases, it can also be written as a free-boundary problem. In Chap. 8, we have discussed how to solve a linear complementarity problem. Here, we study how to solve a free-boundary problem numerically. Many derivative security problems have a final condition with discontinuous derivatives at some point. In this case, their solutions are not very smooth in the domain near this point, and their numerical solutions will have relatively large error. In Chap. 8, we have suggested to deal with this problem in the following way: instead of calculating the price of the derivative security, a difference between the price and an expression with the same or almost the same weak singularity is solved numerically. Because the difference is smooth, the error of numerical solution will be smaller. This method can still be used for free-boundary problems. For them there is another problem. On one side of the free boundary, the price of an American-style derivative satisfies a partial differential equation, and on the other side, it is equal to a given function. Because of this, the second derivative of the price is usually discontinuous on the free boundary. If we can follow the free boundary and use the partial differential equation only on the domain where the equation holds, then we can have less error. Hence, in Sect. 9.1 we not only discuss how to separate the weak singularity caused by the discontinuous first derivative at expiry but also describe how to convert a free-boundary problem into a problem defined on a rectangular domain so that we can easily use the partial differential equation only on the domain where the equation holds. The method described in Sect. 9.1 is referred to as the singularity-separating method (SSM) for free-boundary problems. The next two sections are devoted to discussing how to solve this problem using implicit schemes and pseudo-spectral methods for one-dimensional and two-dimensional cases. There, we also give some results on American vanilla, barrier, Asian, and lookback options, two-factor American vanilla options, and two-factor convertible bonds.

Y.-l. Zhu et al., *Derivative Securities and Difference Methods*, Springer Finance, DOI 10.1007/978-1-4614-7306-0_9, © Springer Science+Business Media New York 2013

9.1 SSM for Free-Boundary Problems

9.1.1 One-Dimensional Cases

From Chaps. 3–5, we know that there are many American-style derivatives. Their major features are the same, but there are some differences among them. In this subsection, first taking an American vanilla call option as an example, we give the details of the singularity-separating method for free-boundary problems. Then, we briefly point out what modifications are needed in order to apply the method to other American-style derivatives.

From Sect. 3.3, we know that on the domain $[0, S_f(t)] \times [0, T]$, the price of an American call option, $C(S, t)$, is the solution of the free-boundary problem

$$\begin{cases} \dfrac{\partial C}{\partial t} + \dfrac{1}{2}\sigma^2 S^2 \dfrac{\partial^2 C}{\partial S^2} + (r - D_0)S\dfrac{\partial C}{\partial S} - rC = 0, & \\ \qquad\qquad\qquad\qquad\qquad\qquad 0 \le S \le S_f(t), \quad 0 \le t \le T, & \\ C(S, T) = \max(S - E, 0), & 0 \le S \le S_f(T), \\ C(S_f(t), t) = S_f(t) - E, & 0 \le t \le T, \\ \dfrac{\partial C}{\partial S}(S_f(t), t) = 1, & 0 \le t \le T, \\ S_f(T) = \max(E, rE/D_0); & \end{cases} \tag{9.1}$$

whereas on the domain $(S_f(t), \infty) \times [0, T]$, $C(S, t) = S - E$. Here, we assume $D_0 \ne 0$. Therefore, as long as we have the solution of the free-boundary problem, we can determine $C(S, t)$ for any $S \ge 0$ and any $t \in [0, T]$. The function $C(S, T) = \max(S - E, 0)$ has a discontinuous derivative at $S = E$. Therefore, $C(S, t)$ is not very smooth in the region where $S \approx E$ and $t \approx T$. Because the second derivative of $C(S, T)$ at $S = E$ goes to infinity, the truncation error of numerical methods near $S = E$ and $t = T$ is relatively large. In order to avoid such a relatively large error, we first find the numerical result of the difference between the prices of the American call option and the European call option, and then add the difference and the price of the European call option together to get the price of the American call option. Similar to those cases given in Sect. 8.3, the function representing the difference is very smooth, so numerical solution can be obtained efficiently.

Now we give the details of the method. Let $c(S, t)$ represent the price of the European call option, whose closed-form expression is given by the formula (2.90). As we know, $c(S, t)$ is the solution of the problem

$$\begin{cases} \dfrac{\partial c}{\partial t} + \dfrac{1}{2}\sigma^2 S^2 \dfrac{\partial^2 c}{\partial S^2} + (r - D_0)S\dfrac{\partial c}{\partial S} - rc = 0, & 0 \le S, \quad 0 \le t \le T \\ c(S, T) = \max(S - E, 0), & 0 \le S. \end{cases}$$

Define

$$\overline{C}(S, t) = C(S, t) - c(S, t)$$

on the domain $[0, S_f(t)] \times [0, T]$. Both $C(S,T)$ and $c(S,T)$ are equal to $\max(S-E,0)$, so $\overline{C}(S,T) = 0$. The functions $C(S,t)$ and $c(S,t)$ satisfy the same linear homogeneous partial differential equation, so the difference between them does the same. At the free boundary $S = S_f(t)$, we have

$$\overline{C}\left(S_f(t),t\right) = C\left(S_f(t),t\right) - c\left(S_f(t),t\right) = S_f(t) - E - c\left(S_f(t),t\right)$$

and

$$\frac{\partial \overline{C}}{\partial S}\left(S_f(t),t\right) = \frac{\partial C}{\partial S}\left(S_f(t),t\right) - \frac{\partial c}{\partial S}\left(S_f(t),t\right) = 1 - \frac{\partial c}{\partial S}\left(S_f(t),t\right).$$

Therefore, $\overline{C}(S,t)$ is the solution of the following free-boundary problem

$$\begin{cases} \dfrac{\partial \overline{C}}{\partial t} + \dfrac{1}{2}\sigma^2 S^2 \dfrac{\partial^2 \overline{C}}{\partial S^2} + (r-D_0)S\dfrac{\partial \overline{C}}{\partial S} - r\overline{C} = 0, & 0 \le S \le S_f(t) \\ & 0 \le t \le T, \\ \overline{C}(S,T) = 0, & 0 \le S \le S_f(T), \\ \overline{C}\left(S_f(t),t\right) = S_f(t) - E - c\left(S_f(t),t\right), & 0 \le t \le T, \\ \dfrac{\partial \overline{C}}{\partial S}\left(S_f(t),t\right) = 1 - \dfrac{\partial c}{\partial S}\left(S_f(t),t\right), & 0 \le t \le T, \\ S_f(T) = \max(E, rE/D_0). \end{cases} \tag{9.2}$$

In the problem above, we need to determine $\overline{C}(S,t)$ on a non-rectangular domain, and one of its boundaries, $S = S_f(t)$, is also unknown.

In order to make discretization of the boundary conditions on the free boundary simple and convert the final-boundary value problem into an initial-boundary value problem, we introduce a new coordinate system $\{\xi, \tau\}$ through a transformation defined by

$$\begin{cases} \xi = \dfrac{S}{S_f(t)}, \\ \tau = T - t. \end{cases}$$

This transformation converts the four boundaries of the domain of the problem (9.2), $S = 0$, $S = S_f(t)$, $t = T$, and $t = 0$, into $\xi = 0$, $\xi = 1$, $\tau = 0$, and $\tau = T$, respectively (see Fig. 9.1). Now the problem is defined on a rectangular domain, and the value of the solution at $\tau = 0$ is given, that is, the problem now is an initial-boundary value problem on a rectangular domain.

Let

$$s_f(\tau) = \frac{1}{E} S_f(T-\tau)$$

and

$$u(\xi,\tau) = \frac{1}{E}\overline{C}(S,t) = \frac{1}{E}\overline{C}\left(\xi E s_f(\tau),\ T-\tau\right),$$

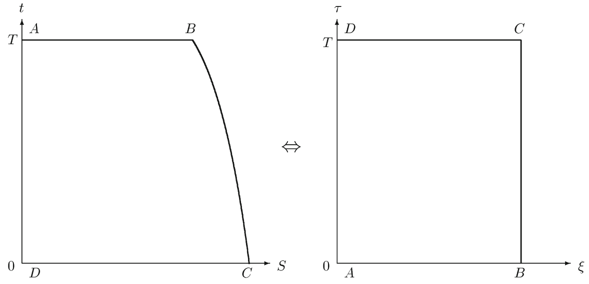

Fig. 9.1. Transforming a non-rectangular domain to a rectangular domain

that is,

$$S_f(t) = Es_f(T-t)$$

and

$$\overline{C}(S,t) = Eu\left(\frac{S}{Es_f(T-t)},\ T-t\right).$$

Since

$$\begin{aligned}
\frac{\partial \overline{C}}{\partial t} &= E\left[\frac{\xi}{s_f(\tau)}\frac{ds_f(\tau)}{d\tau}\frac{\partial u}{\partial \xi} - \frac{\partial u}{\partial \tau}\right],\\
\frac{\partial \overline{C}}{\partial S} &= \frac{\partial u}{\partial \xi}\frac{1}{s_f(\tau)},\\
\frac{\partial^2 \overline{C}}{\partial S^2} &= \frac{1}{E}\frac{\partial^2 u}{\partial \xi^2}\left[\frac{1}{s_f(\tau)}\right]^2,
\end{aligned}$$

the problem (9.2) can be rewritten as

$$\begin{cases}
\dfrac{\partial u}{\partial \tau} = k_2\xi^2\dfrac{\partial^2 u}{\partial \xi^2} + \left(k_1 + \dfrac{1}{s_f}\dfrac{ds_f}{d\tau}\right)\xi\dfrac{\partial u}{\partial \xi} - k_0 u, & 0 \le \xi \le 1,\\
 & 0 \le \tau \le T,\\
u(\xi,0) = 0, & 0 \le \xi \le 1,\\
u(1,\tau) = g\left(s_f(\tau),\tau\right), & 0 \le \tau \le T,\\
\dfrac{\partial u}{\partial \xi}(1,\tau) = h\left(s_f(\tau),\tau\right), & 0 \le \tau \le T,\\
s_f(0) = \max(1, r/D_0),
\end{cases} \tag{9.3}$$

where $k_0 = r$, $k_1 = r - D_0$, $k_2 = \sigma^2/2$,

$$g\left(s_f(\tau),\tau\right) = s_f(\tau) - 1 - \frac{1}{E}c\left(Es_f(\tau),\ T-\tau\right)$$

and

$$h\left(s_f(\tau), \tau\right) = s_f(\tau)\left[1 - \frac{\partial c\left(E s_f(\tau),\ T-\tau\right)}{\partial S}\right].$$

The differential equation in the problem (9.3) is a partial differential equation for u and can be understood as an ordinary differential equation for $s_f(\tau)$. This problem is a combination of an initial-boundary value problem for $u(\xi, \tau)$ on the domain $[0,1] \times [0,T]$ and an initial value problem for $s_f(\tau)$ on the interval $[0,T]$. It can be solved using explicit schemes, implicit schemes, or pseudo-spectral methods. After we obtain $u(\xi,\tau)$, we can get the price of the American call option on the domain $[0, S_f(t)] \times [0,T]$ by

$$C(S,t) = Eu\left(\frac{S}{Es_f(T-t)}, T-t\right) + c(S,t).$$

From the expression of $C(S,t)$, we know that in order to computing $C(S,t)$, we need to write a code for computing $u(\xi,\tau)$ and also need to have a code for calculating $c(S,t)$. When the projects of Chap. 6 have been finished, the function **double** BS can be used for such a purpose.

The method described here is referred to as the singularity-separating method for American call options. The solution of the original American call option satisfies different equations in the two regions divided by the free boundary $S = S_f(t)$, and its solution has a discontinuous second derivative—a type of weak singularity—on the free boundary. In this method, the position of the free boundary is tracked accurately, so that we can use the different equations in each region exactly. Because the solution in the domain $(S_f(t), \infty) \times [0,T]$ is given by a known function, we only need to determine the solution in the region $[0, S_f(t)] \times [0,T]$. In this region, the second derivative near the free boundary is continuous, so the solution we want to get numerically is smoother than the original solution. Here, we also suggest to compute the difference between the American call option and the European call option numerically in the domain $[0, S_f(t)] \times [0,T]$, instead of directly computing the price of the American call option numerically. The derivative of solution of the American call option with respect to S at the point (E,T) is discontinuous if $S_f(T) \neq E$. The difference is much smoother than the solution of the American call option in the domain $[0, S_f(t)] \times [0,T]$, which make the truncation error smaller. Therefore, in the method described above, we use some techniques such that the solution we need to get numerically is much smoother than the original solution, which makes numerical methods more efficient. We refer to this as singularity-separating as we did in Sect. 8.3, because the solution becomes smoother than the original one after some singularities on the free boundary and at the point (E,T) have been "separated". Here, the singularity that has been "separated" is the discontinuity of the derivatives of the solution, which is weak. The idea of the method was originally developed for dealing with shock problems in fluid mechanics (see [97]) and the Stefan problem (see [86]), the solutions of which had, for most of the cases, stronger discontinuities than we have here. It might be more precise if we use "weak-singularity-separating" instead of singularity-separating. However, for simplicity we just keep the name of the method.

As pointed in Sect. 3.3.3, between American call and put options there exists the put–call symmetry relations. Using these relations, pricing a put option can be reduced to pricing a call option. There, the symmetry relations have been derived when American option problems are formulated as linear complementarity problems. Here, let us derive this conclusion when the problems are written as free-boundary problems. Let $P(S,t)$ stand for the price of an American put option. $P(S,t)$ should be the solution of the problem (3.16) on the domain $[S_f(t),\infty)\times[0,T]$ and equal $E-S$ on the domain $[0,S_f(t))\times[0,T]$. Let

$$\begin{cases} \eta = \dfrac{E^2}{S}, \\ u(\eta,t) = \dfrac{EP(S,t)}{S}, \end{cases}$$

then it is easy to see that $u(\eta,t)$ is the solution of the free-boundary problem

$$\begin{cases} \dfrac{\partial u}{\partial t}+\dfrac{1}{2}\sigma^2\eta^2\dfrac{\partial^2 u}{\partial \eta^2}+(D_0-r)\eta\dfrac{\partial u}{\partial \eta}-D_0u=0, & 0\le\eta\le\eta_f(t), \\ & 0\le t\le T, \\ u(\eta,T)=\max(\eta-E,0), & 0\le\eta\le\eta_f(T), \\ u\left(\eta_f(t),t\right)=\eta_f(t)-E, & 0\le t\le T, \\ \dfrac{\partial u}{\partial \eta}\left(\eta_f(t),t\right)=1, & 0\le t\le T, \\ \eta_f(T)=\max\left(E,D_0E/r\right) \end{cases} \tag{9.4}$$

on the domain $[0,\eta_f(t)]\times[0,T]$; whereas on the domain $(\eta_f(t),\infty)\times[0,T]$,

$$u(\eta,t)=\eta-E.$$

As we can see, if the parameter r and the parameter D_0 in the problem (9.1) exchange their positions, then the problem (9.1) almost becomes the problem (9.4), except for the state variable. Therefore, $P(S,t)$ can be determined in the following way. First, understanding D_0 as r and r as D_0, we solve the problem (9.1) with the state variable η, instead of S, and get $u(\eta,t)$. Then, $P(S,t)$ is obtained by

$$P(S,t)=\frac{S}{E}u\left(\frac{E^2}{S},t\right).$$

That is, we find $P(S,t)$ by using one of the symmetry relations.

It is not always reasonable to assume the volatility to be a constant. If the volatility is thought as a function of S, namely, $\sigma=\sigma(S)$, then the formulation

(9.1) is still true after changing σ to $\sigma(S)$. Is the formulation (9.2) still true? The answer is no because in this case we do not have analytic solutions for European option. However, we can define

$$\overline{C}(S,t) = C(S,t) - c_{E}(S,t;\sigma(E))$$

on the domain $[0, S_f(t)] \times [0, T]$, where $c_E(S,t;\sigma(E))$ denotes the price of the European call option with $\sigma = \sigma(E)$. In this case, $\overline{C}(S,t)$ does not satisfy the Black–Scholes equation. Instead, it satisfies the following nonhomogeneous equation:

$$\frac{\partial \overline{C}}{\partial t} + \frac{1}{2}\sigma^2(S)S^2\frac{\partial^2 \overline{C}}{\partial S^2} + (r - D_0)S\frac{\partial \overline{C}}{\partial S} - r\overline{C} = f(S,t), \tag{9.5}$$

where $f(S,t)$ is given by the expression (8.79) in Sect. 8.3.2. For this case, the formulation is almost the same as the problem (9.2) except that the partial differential equation in problem (9.2) should be replaced by problem (9.5). Therefore, the singularity-separating method still works for American options with variable volatilities because the singularity is weakened.

The same idea still works for American barrier, Asian, and lookback options. In order to remove the weak singularity at $S = E$ and $t = T$, we can use the solutions of vanilla European options for American barrier, Asian, and lookback options. However, it will be better to compute numerically the differences between American and European barrier options and between American and European lookback options because the differences are smaller in these cases. Just like the vanilla option case, the partial differential equation that the differences satisfy in these cases is still the partial differential equation in the problem (9.2). For European Asian options, explicit solutions have not been found, and the partial differential equation for Asian options is different from vanilla options. Thus, when we apply the SSM, the resulting equation for Asian options differs slightly from barrier and lookback options. For average strike options with $\alpha = 1$, the singularity-separating method will still work, and the difference will be a solution of a nonhomogeneous partial differential equation problem with a weaker singularity. It is not difficult to derive the problem in this case, and we leave this as a problem for readers.

Consider put options on stocks paying dividends discretely. Suppose that the last dividend is paid at time t_K. This method can still be used from $t = T$ to $t = t_K$. From $t = t_K$ to $t = 0$, the solution is already smooth, so we can just compute the price of the American option directly. It is clear that in this way a quite good result still can be obtained on a coarse mesh.

9.1.2 Two-Dimensional Cases

Two-Factor Options. In the above, we have discussed the formulation of American options if the volatility is a constant or a function of S. Now let us look at the case both the price of asset and the volatility of the asset price are random variables. As we have done in Sect. 8.3.6, we call such an option a two-factor option. Here, we discuss how to formulate the American two-factor vanilla call option as a free boundary problem if $D_0 \neq 0$.

We still assume the asset price S and the stochastic volatility σ to follow the set of equations (8.98) and require the conditions (8.99) and (8.100) or the conditions (8.101) and (8.102) to hold.

Consider an American two-factor vanilla call option problem and let its value be $C(S, \sigma, t)$. As an American call option, it satisfies the condition:

$$C(S, \sigma, t) \geq \max(S - E, 0).$$

Because a European two-factor call option is a solution of the problem (8.105), the value of a two-factor vanilla American call option is a solution of the following linear complementarity problem:

$$\begin{cases} \min\left(-\dfrac{\partial C}{\partial t} - \mathbf{L_{S,\sigma}} C,\ C - G_c\right) = 0, & 0 \leq S,\ \sigma_l \leq \sigma \leq \sigma_u,\ t \leq T, \\ C(S, \sigma, T) = G_c(S, T), & 0 \leq S,\ \sigma_l \leq \sigma \leq \sigma_u, \end{cases} \tag{9.6}$$

where $\mathbf{L_{S,\sigma}}$ is given by the expression (8.104):

$$\begin{aligned} \mathbf{L_{S,\sigma}} = & \frac{1}{2}\sigma^2 S^2 \frac{\partial^2}{\partial S^2} + \rho\sigma S q \frac{\partial^2}{\partial S \partial \sigma} + \frac{1}{2} q^2 \frac{\partial^2}{\partial \sigma^2} \\ & + (r - D_0) S \frac{\partial}{\partial S} + (p - \lambda q) \frac{\partial}{\partial \sigma} - r, \end{aligned}$$

and

$$G_c(S, t) = \max(S - E, 0).$$

Consider the case $D_0 > 0$. Because

$$\frac{\partial G_c}{\partial t} + \mathbf{L_{S,\sigma}} G_c < 0 \quad \text{for} \quad S > \max(E, rE/D_0)$$

and

$$\frac{\partial G_c}{\partial t} + \mathbf{L_{S,\sigma}} G_c \geq 0 \quad \text{for} \quad S \leq \max(E, rE/D_0),$$

there exists a free boundary $S = S_f(\sigma, t)$ starting from the straight line $S = \max(E, rE/D_0)$ at $t = T$ in the (S, σ, t)-space, and the entire domain is divided into two regions by the free boundary. On the domain $(S_f(\sigma, t), \infty) \times [\sigma_l, \sigma_u] \times [0, T]$,

$$C(S, \sigma, t) = \max(S - E, 0);$$

whereas on $[0, S_f(\sigma, t)] \times [\sigma_l, \sigma_u] \times [0, T]$, $C(S, \sigma, t)$ is the solution of the following free-boundary problem:

$$
\begin{cases}
\dfrac{\partial C}{\partial t} + \mathbf{L}_{\mathbf{s},\sigma} C = 0, & 0 \le S \le S_f(\sigma, t), \\
& \sigma_l \le \sigma \le \sigma_u, \quad 0 \le t \le T, \\
C(S, \sigma, T) = \max(S - E, 0), & 0 \le S \le S_f(\sigma, T), \\
& \sigma_l \le \sigma \le \sigma_u, \\
C\left(S_f(\sigma, t), \sigma, t\right) = S_f(\sigma, t) - E, & \sigma_l \le \sigma \le \sigma_u, \quad 0 \le t \le T, \\
\dfrac{\partial C\left(S_f(\sigma, t), \sigma, t\right)}{\partial S} = 1, & \sigma_l \le \sigma \le \sigma_u, \quad 0 \le t \le T, \\
S_f(\sigma, T) = \max(E, rE/D_0), & \sigma_l \le \sigma \le \sigma_u.
\end{cases}
\tag{9.7}
$$

Just like the European two-factor option case, we let

$$
\overline{C}(S, \sigma, t) = C(S, \sigma, t) - c_1(S, \sigma, t) \tag{9.8}
$$

on the domain $[0, S_f(\sigma, t)] \times [\sigma_l, \sigma_u] \times [0, T]$. Here, $c_1(S, \sigma, t)$ is the same as the function $c(S, t)$ given by the formula (2.90) in Sect. 2.6.5, namely, the price of the vanilla European call option when σ is a constant. Thus, the difference $\overline{C}$ is the solution of the following free-boundary problem:

$$
\begin{cases}
\dfrac{\partial \overline{C}}{\partial t} + \mathbf{L}_{\mathbf{s},\sigma} \overline{C} = f(S, \sigma, t), & 0 \le S \le S_f(\sigma, t), & \sigma_l \le \sigma \le \sigma_u, \\
& & 0 \le t \le T, \\
\overline{C}(S, \sigma, T) = 0, & 0 \le S \le S_f(\sigma, T), & \sigma_l \le \sigma \le \sigma_u, \\
\overline{C}\left(S_f(\sigma, t), \sigma, t\right) = S_f(\sigma, t) - E - c_1\left(S_f(\sigma, t), \sigma, t\right), & & \sigma_l \le \sigma \le \sigma_u, \\
& & 0 \le t \le T, \\
\dfrac{\partial \overline{C}\left(S_f(\sigma, t), \sigma, t\right)}{\partial S} = 1 - \dfrac{\partial c_1\left(S_f(\sigma, t), \sigma, t\right)}{\partial S}, & & \sigma_l \le \sigma \le \sigma_u, \\
& & 0 \le t \le T, \\
S_f(\sigma, T) = \max(E, rE/D_0), & & \sigma_l \le \sigma \le \sigma_u,
\end{cases}
\tag{9.9}
$$

where

$$
f(S, \sigma, t) = -\rho\sigma S q \frac{\partial^2 c_1}{\partial S \partial \sigma} - \frac{1}{2} q^2 \frac{\partial^2 c_1}{\partial \sigma^2} - (p - \lambda q) \frac{\partial c_1}{\partial \sigma},
$$

$\dfrac{\partial c_1}{\partial \sigma}$, $\dfrac{\partial^2 c_1}{\partial \sigma^2}$, and $\dfrac{\partial^2 c}{\partial S \partial \sigma}$ being given by the set of expressions (8.108).

As we see from the problems (9.7) and (9.9), the derivative of $C(S, \sigma, t)$ with respect to S is discontinuous at the point $t = T$ and $S = E$, and the derivative of $\overline{C}(S, \sigma, t)$ with respect to S at $t = T$ is identically equal to zero. It is expected that $\overline{C}(S, \sigma, t)$ is smoother than $C(S, \sigma, t)$ even though in this case the singularity only becomes weaker but is not completely removed because of the term $\dfrac{\partial^2 c_1}{\partial S \partial \sigma}$ in $f(S,\sigma,t)$. Therefore, when a numerical method is used, the truncation error for the problem (9.9) will be smaller than the problem (9.7). This is why we consider the formulation (9.9) instead of the formulation (9.7).

The free-boundary problem (9.9) is defined on the domain $[0, S_f(\sigma,t)] \times [\sigma_l, \sigma_u] \times [0,T]$ and the free boundary $S_f(\sigma,t)$ is a moving and unknown boundary. In order to make the discretization simple, we introduce the following transformation

$$\begin{cases} \xi = \dfrac{S}{S_f(\sigma,t)}, \\ \sigma = \sigma, \\ \tau = T - t. \end{cases} \tag{9.10}$$

This transformation maps the domain

$$[0, S_f(\sigma,t)] \times [\sigma_l, \sigma_u] \times [0,T]$$

in the (S,σ,t)-space onto a new domain

$$[0,1] \times [\sigma_l, \sigma_u] \times [0,T]$$

in the (ξ,σ,τ)-space and the moving boundary onto a plane under the new coordinate system. In the (ξ,σ,τ)-space, it is easy to construct numerical methods to solve the problem. Define

$$s_f(\sigma,\tau) = S_f(\sigma,t) = S_f(\sigma, T-\tau)$$

and

$$u(\xi,\sigma,\tau) = \overline{C}(S,\sigma,t) = \overline{C}\left(\xi s_f(\sigma,\tau), \sigma, T-\tau\right).$$

Among the derivatives of $\overline{C}$ and u, there are the following relations:

$$\begin{aligned}
\frac{\partial \overline{C}}{\partial t} &= \frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial t} + \frac{\partial u}{\partial \tau}\frac{\partial \tau}{\partial t} = \frac{\xi}{s_f}\frac{\partial s_f}{\partial \tau}\frac{\partial u}{\partial \xi} - \frac{\partial u}{\partial \tau}, \\
\frac{\partial \overline{C}}{\partial S} &= \frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial S} = \frac{1}{s_f}\frac{\partial u}{\partial \xi}, \\
\frac{\partial \overline{C}}{\partial \sigma} &= \frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial \sigma} + \frac{\partial u}{\partial \sigma} = -\left(\frac{\xi}{s_f}\frac{\partial s_f}{\partial \sigma}\frac{\partial u}{\partial \xi} - \frac{\partial u}{\partial \sigma}\right), \\
\frac{\partial^2 \overline{C}}{\partial S^2} &= \frac{1}{s_f^2}\frac{\partial^2 u}{\partial \xi^2},
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 \overline{C}}{\partial S \partial \sigma} &= \frac{\partial}{\partial \sigma}\left(\frac{1}{s_f}\frac{\partial u}{\partial \xi}\right) = -\frac{1}{s_f^2}\frac{\partial s_f}{\partial \sigma}\frac{\partial u}{\partial \xi} + \frac{1}{s_f}\left(\frac{\partial^2 u}{\partial \xi^2}\frac{\partial \xi}{\partial \sigma} + \frac{\partial^2 u}{\partial \xi \partial \sigma}\right) \\
&= -\frac{1}{s_f^2}\frac{\partial s_f}{\partial \sigma}\frac{\partial u}{\partial \xi} - \frac{\xi}{s_f^2}\frac{\partial s_f}{\partial \sigma}\frac{\partial^2 u}{\partial \xi^2} + \frac{1}{s_f}\frac{\partial^2 u}{\partial \xi \partial \sigma}, \\
\frac{\partial^2 \overline{C}}{\partial \sigma^2} &= -\left[\frac{\partial}{\partial \sigma}\left(\frac{\xi}{s_f}\frac{\partial s_f}{\partial \sigma}\right)\frac{\partial u}{\partial \xi} + \frac{\xi}{s_f}\frac{\partial s_f}{\partial \sigma}\left(\frac{\partial^2 u}{\partial \xi^2}\frac{\partial \xi}{\partial \sigma} + \frac{\partial^2 u}{\partial \xi \partial \sigma}\right)\right. \\
&\qquad \left. - \left(\frac{\partial^2 u}{\partial \sigma \partial \xi}\frac{\partial \xi}{\partial \sigma} + \frac{\partial^2 u}{\partial \sigma^2}\right)\right] \\
&= \left\{\left(\frac{\xi}{s_f}\right)^2 \left(\frac{\partial s_f}{\partial \sigma}\right)^2 \frac{\partial^2 u}{\partial \xi^2} - 2\frac{\xi}{s_f}\frac{\partial s_f}{\partial \sigma}\frac{\partial^2 u}{\partial \xi \partial \sigma} + \frac{\partial^2 u}{\partial \sigma^2}\right. \\
&\qquad \left. + \left[2\frac{\xi}{s_f^2}\left(\frac{\partial s_f}{\partial \sigma}\right)^2 - \frac{\xi}{s_f}\frac{\partial^2 s_f}{\partial \sigma^2}\right]\frac{\partial u}{\partial \xi}\right\}.
\end{aligned}$$

Substituting them into the partial differential equation in the problem (9.9) yields

$$\frac{\partial u}{\partial \tau} = a_1 \xi^2 \frac{\partial^2 u}{\partial \xi^2} + a_2 \xi q \frac{\partial^2 u}{\partial \xi \partial \sigma} + a_3 q^2 \frac{\partial^2 u}{\partial \sigma^2} + a_4 \xi \frac{\partial u}{\partial \xi} + a_5 \frac{\partial u}{\partial \sigma} + a_6 u + a_7,$$

where

$$\begin{aligned}
a_1 &= \frac{1}{2}\sigma^2 - \frac{\rho \sigma q}{s_f}\frac{\partial s_f}{\partial \sigma} + \frac{1}{2}\left(\frac{q}{s_f}\frac{\partial s_f}{\partial \sigma}\right)^2, \\
a_2 &= \rho \sigma - \frac{q}{s_f}\frac{\partial s_f}{\partial \sigma}, \\
a_3 &= \frac{1}{2}, \\
a_4 &= \frac{1}{s_f}\frac{\partial s_f}{\partial \tau} + r - D_0 - (\rho \sigma q + p - \lambda q)\frac{1}{s_f}\frac{\partial s_f}{\partial \sigma} \\
&\qquad + \left(\frac{q}{s_f}\frac{\partial s_f}{\partial \sigma}\right)^2 - \frac{1}{2}q^2 \frac{1}{s_f}\frac{\partial^2 s_f}{\partial \sigma^2}, \\
a_5 &= p - \lambda q, \\
a_6 &= -r, \\
a_7 &= -f(S, \sigma, t) = -f\left(\xi s_f(\sigma, \tau), \sigma, T - \tau\right).
\end{aligned}$$

Therefore, noticing

$$\begin{cases}
c_1(S_f, \sigma, t) = S_f \mathrm{e}^{-D_0(T-t)} N(d_1) - E\mathrm{e}^{-r(T-t)} N(d_2), \\
\dfrac{\partial c_1(S_f, \sigma, t)}{\partial S} = \mathrm{e}^{-D_0(T-t)} N(d_1),
\end{cases}$$

we can rewrite the problem (9.9) as

$$
\begin{cases}
\dfrac{\partial u}{\partial \tau} = a_1 \xi^2 \dfrac{\partial^2 u}{\partial \xi^2} + a_2 \xi q \dfrac{\partial^2 u}{\partial \xi \partial \sigma} + a_3 q^2 \dfrac{\partial^2 u}{\partial \sigma^2} + a_4 \xi \dfrac{\partial u}{\partial \xi} + a_5 \dfrac{\partial u}{\partial \sigma} \\
\qquad + a_6 u + a_7, \qquad 0 \le \xi \le 1, \quad \sigma_l \le \sigma \le \sigma_u, \quad 0 \le \tau \le T, \\
u(\xi, \sigma, 0) = 0, \qquad 0 \le \xi \le 1, \quad \sigma_l \le \sigma \le \sigma_u, \\
u(1, \sigma, \tau) = s_f(\sigma, \tau) \left[1 - \mathrm{e}^{-D_0 \tau} N(d_1)\right] - E \left[1 - \mathrm{e}^{-r\tau} N(d_2)\right], \\
\qquad \sigma_l \le \sigma \le \sigma_u, \quad 0 \le \tau \le T, \\
\dfrac{\partial u(1, \sigma, \tau)}{\partial \xi} = s_f(\sigma, \tau) \left[1 - \mathrm{e}^{-D_0 \tau} N(d_1)\right], \\
\qquad \sigma_l \le \sigma \le \sigma_u, \quad 0 \le \tau \le T, \\
s_f(\sigma, 0) = \max\left(E, \dfrac{rE}{D_0}\right), \qquad \sigma_l \le \sigma \le \sigma_u,
\end{cases} \tag{9.11}
$$

where

$$
d_1 = \left[\ln \frac{s_f \mathrm{e}^{-D_0 \tau}}{E \mathrm{e}^{-r\tau}} + \frac{1}{2}\sigma^2 \tau\right] \Big/ \left(\sigma\sqrt{\tau}\right) \quad \text{and} \quad d_2 = d_1 - \sigma\sqrt{\tau}.
$$

Once we have the solution of the problem (9.11), $u(\xi, \sigma, \tau)$, we can get the value of the original American call option by

$$
\begin{aligned}
C(S, \sigma, t) &= \overline{C}(S,\ \sigma,\ t) + c_1(S,\ \sigma,\ t) \\
&= u\left(\frac{S}{s_f(\sigma, T-t)},\ \sigma,\ T-t\right) + c_1(S,\ \sigma,\ t).
\end{aligned} \tag{9.12}
$$

This method is called the singularity-separating method for American two-factor call options.

For two-factor vanilla American put options, the linear complementarity problem is

$$
\begin{cases}
\min\left(-\dfrac{\partial P}{\partial t} - \mathbf{L}_{\mathbf{S},\sigma} P,\ P - G_p\right) = 0, \quad 0 \le S,\ \sigma_l \le \sigma \le \sigma_u,\ t \le T, \\
P(S, \sigma, T) = G_p(S, T), \quad 0 \le S,\ \sigma_l \le \sigma \le \sigma_u,
\end{cases}
$$

where

$$
G_p(S, t) = \max(E - S, 0).
$$

Introducing the transformation

$$
\begin{cases}
\eta = \dfrac{E^2}{S}, \\
\sigma = \sigma, \\
t = t, \\
u(\eta, \sigma, t) = \dfrac{EP(S, \sigma, t)}{S}
\end{cases} \tag{9.13}
$$

and noticing the following relations

$$
\begin{aligned}
\frac{\partial \eta}{\partial S} &= -\frac{E^2}{S^2}, & \frac{\partial P}{\partial t} &= \frac{S}{E}\frac{\partial u}{\partial t}, \\
\frac{\partial P}{\partial S} &= \frac{u}{E} - \frac{E}{S}\frac{\partial u}{\partial \eta}, & \frac{\partial P}{\partial \sigma} &= \frac{S}{E}\frac{\partial u}{\partial \sigma}, \\
\frac{\partial^2 P}{\partial S^2} &= \frac{E^3}{S^3}\frac{\partial^2 u}{\partial \eta^2}, \\
\frac{\partial^2 P}{\partial S \partial \sigma} &= \frac{1}{E}\frac{\partial u}{\partial \sigma} - \frac{E}{S}\frac{\partial^2 u}{\partial \eta \partial \sigma}, \\
\frac{\partial^2 P}{\partial \sigma^2} &= \frac{S}{E}\frac{\partial^2 u}{\partial \sigma^2},
\end{aligned}
$$

we can convert the linear complementarity problem above into another linear complementarity problem

$$
\begin{cases}
\min\left(-\dfrac{\partial u}{\partial t} - \mathbf{L}_{\eta,\sigma} u,\ u - G_u\right) = 0, & 0 \le \eta,\ \sigma_l \le \sigma \le \sigma_u,\ t \le T, \\
u(\eta, \sigma, T) = G_u(\eta, T), & 0 \le \eta,\ \sigma_l \le \sigma \le \sigma_u,
\end{cases}
$$

where

$$G_u(\eta, t) = \max(\eta - E, 0)$$

and

$$
\begin{aligned}
\mathbf{L}_{\eta,\sigma} = {} & \frac{1}{2}\sigma^2\eta^2\frac{\partial^2}{\partial \eta^2} - \rho\sigma q\eta\frac{\partial^2}{\partial \eta \partial \sigma} + \frac{1}{2}q^2\frac{\partial^2}{\partial \sigma^2} \\
& + (D_0 - r)\eta\frac{\partial}{\partial \eta} + [p - (\lambda - \rho\sigma)q]\frac{\partial}{\partial \sigma} - D_0.
\end{aligned}
$$

This problem has the same form as the problem (9.6). The only difference is that r and D_0 are switched, and ρ and λ in the problem (9.6) are replaced by $-\rho$ and $\lambda - \rho\sigma$ here. Therefore, a put problem can be written as a call problem.

Let $C(S, \sigma, t; a, b, c, d)$ and $P(S, \sigma, t; a, b, c, d)$ denote the prices of American call and put options and $S_{cf}(\sigma, t; a, b, c, d)$ and $S_{pf}(\sigma, t; a, b, c, d)$ be their optimal exercise prices. Here, a, b, c, and d are parameters (or parameter functions) for the risk-free interest rate r, dividend yield rate D_0, correlation coefficient ρ, and market price of volatility risk λ, respectively. Then, what we have described above can be written as a relation between the American two-factor vanilla put and call options:

$$
\begin{cases}
P(S, \sigma, t; a, b, c, d) = \dfrac{S}{E} C\left(\dfrac{E^2}{S}, \sigma, t; b, a, -c, d - c\sigma\right), \\
S_{pf}(\sigma, t; a, b, c, d) = E^2 / S_{cf}(\sigma, t; b, a, -c, d - c\sigma).
\end{cases}
$$

If we let

$$\eta = E^2/S, \quad \bar{c} = -c$$

and

$$\bar{d} = d - c\sigma = d + \bar{c}\sigma,$$

then the first relation above can be written as

$$P\left(\frac{E^2}{\eta}, \sigma, t; a, b, -\bar{c}, \bar{d} - \bar{c}\sigma\right) = \frac{E}{\eta} C\left(\eta, \sigma, t; b, a, \bar{c}, \bar{d}\right)$$

or

$$C\left(S, \sigma, t; a, b, c, d\right) = \frac{S}{E} P\left(\frac{E^2}{S}, \sigma, t; b, a, -c, d - c\sigma\right).$$

The second relation can be written in a symmetric form

$$S_{pf}(\sigma, t; a, b, c, d) \times S_{cf}(\sigma, t; b, a, -c, d - c\sigma) = E^2.$$

Therefore, we can have the following relations:

$$\begin{cases} P(S, \sigma, t; a, b, c, d) = \dfrac{S}{E} C\left(\dfrac{E^2}{S}, \sigma, t; b, a, -c, d - c\sigma\right), \\ C\left(S, \sigma, t; a, b, c, d\right) = \dfrac{S}{E} P\left(\dfrac{E^2}{S}, \sigma, t; b, a, -c, d - c\sigma\right), \\ S_{pf}(\sigma, t; a, b, c, d) \times S_{cf}(\sigma, t; b, a, -c, d - c\sigma) = E^2, \end{cases} \tag{9.14}$$

which in this book are referred to as the call–put symmetry relations between American two-factor vanilla call and put options. Thus, if we have a code for one type of option, call or put, then in order to calculate another type of option, we only need to make a little change.

The free-boundary problem for a call option is defined on a finite domain and that for a put option is on an infinite domain. Consequently, it will be natural to write a code for call options and calculate a put option as a call option.

Two-Factor Convertible Bonds. Another example of American-style derivatives depending on two random variables is two-factor convertible bonds. Let $B_c(S, r, t)$ be the price of such a bond. As was pointed out in Sect. 5.7, the computational domain of a two-factor convertible bond problem can be divided into two parts. On the domain $(S_f(r, t), \infty) \times [r_l, r_u] \times [0, T]$,

$$B_c(S, r, t) = \max(Z, nS);$$

whereas on the domain $[0, S_f(r, t)] \times [r_l, r_u] \times [0, T]$, $B_c(S, r, t)$ is the solution of the free-boundary problem:

$$
\begin{cases}
\dfrac{\partial B_c}{\partial t} + \mathbf{L_{s,r}} B_c + kZ = 0, & 0 \le S \le S_f(r,t), \quad r_l \le r \le r_u, \\
 & 0 \le t \le T, \\
B_c(S,r,T) = \max(Z, nS), & 0 \le S \le S_f(r,T), \quad r_l \le r \le r_u, \\
B_c\left(S_f(r,t), r, t\right) = nS_f(r,t), & r_l \le r \le r_u, \quad 0 \le t \le T, \\
\dfrac{\partial B_c}{\partial S}\left(S_f(r,t), r, t\right) = n, & r_l \le r \le r_u, \;\; 0 \le t \le T, \\
S_f(r,T) = \max\left(\dfrac{Z}{n}, \dfrac{kZ}{D_0 n}\right), & r_l \le r \le r_u.
\end{cases}
\tag{9.15}
$$

where

$$
\mathbf{L_{s,r}} = \frac{1}{2}\sigma^2 S^2 \frac{\partial^2}{\partial S^2} + \rho\sigma S w \frac{\partial^2}{\partial S \partial r} + \frac{1}{2} w^2 \frac{\partial^2}{\partial r^2} + (r - D_o) S \frac{\partial}{\partial S} + (u - \lambda w)\frac{\partial}{\partial r} - r.
$$

Let $b_c(S,r,t)$ be the solution of the problem

$$
\begin{cases}
\dfrac{\partial b_c}{\partial t} + \dfrac{1}{2}\sigma^2 S^2 \dfrac{\partial^2 b_c}{\partial S^2} + (r - D_0) S \dfrac{\partial b_c}{\partial S} - r b_c = 0, & 0 \le S, \\
 & 0 \le t \le T, \\
b_c(S,T) = \max(Z, nS) = n\max(S - Z/n, 0) + Z, & 0 \le S,
\end{cases}
\tag{9.16}
$$

where σ, r, and D_0 are constants. This problem has the following solution:

$$
b_c(S,r,t) = nc(S,t;Z/n) + \mathrm{e}^{-r(T-t)} Z,
$$

where $c(S,t;Z/n)$ is the price of a European call option with an exercise price $E = Z/n$. Define

$$
\overline{B}_c(S,r,t) = B_c(S,r,t) - b_c(S,r,t). \tag{9.17}
$$

For $\overline{B}_c(S,r,t)$, the free boundary problem is

$$
\begin{cases}
\dfrac{\partial \overline{B}_c}{\partial t} + \mathbf{L_{s,r}} \overline{B}_c + kZ = f(S,r,t), & 0 \le S \le S_f(r,t), \\
 & r_l \le r \le r_u, \;\; 0 \le t \le T, \\
\overline{B}_c(S,r,T) = 0, & 0 \le S \le S_f(r,T), \;\; r_l \le r \le r_u, \\
\overline{B}_c\left(S_f(r,t), r, t\right) = nS_f(r,t) - b_c\left(S_f(r,t), r, t\right), & \\
 & r_l \le r \le r_u, \;\; 0 \le t \le T, \\
\dfrac{\partial \overline{B}_c}{\partial S}\left(S_f(r,t), r, t\right) = n - \dfrac{\partial b_c\left(S_f(r,t), r, t\right)}{\partial S}, & \\
 & r_l \le r \le r_u, \;\; 0 \le t \le T, \\
S_f(r,T) = \max\left(\dfrac{Z}{n}, \dfrac{kZ}{D_0 n}\right), & r_l \le r \le r_u,
\end{cases}
\tag{9.18}
$$

where

$$f(S,r,t) = -\rho\sigma S w\frac{\partial^2 b_c}{\partial S \partial r} - \frac{1}{2}w^2\frac{\partial^2 b_c}{\partial r^2} - (u-\lambda w)\frac{\partial b_c}{\partial r}.$$

In order to make the discretization easy, we introduce the following transformation

$$\begin{cases} \xi = \dfrac{S}{S_f(r,t)}, \\ \bar{r} = \dfrac{r-r_l}{r_u-r_l}, \\ \tau = T-t. \end{cases} \tag{9.19}$$

This transformation maps the domain

$$[0, S_f(r,t)] \times [r_l, r_u] \times [0,T]$$

in the (S, r, t)-space onto the domain

$$[0,1] \times [0,1] \times [0,T]$$

in the $(\xi, \bar{r}, \tau)$-space. We also introduce two new variables u and s_f defined by

$$\begin{cases} u(\xi,\bar{r},\tau) = \dfrac{\overline{B}_c(S,r,t)}{Z}, \\ s_f(\bar{r},\tau) = \dfrac{S_f(r,t)}{Z/n} \end{cases} \tag{9.20}$$

and let

$$v(\xi,\bar{r},\tau) = b_c(S,r,t)/Z.$$

For v we have

$$\begin{aligned} v(\xi,\bar{r},\tau) &= nc(S,t;Z/n)/Z + \mathrm{e}^{-r(T-t)} \\ &= (nS/Z)\mathrm{e}^{-D_0(T-t)}N(d_1) - \mathrm{e}^{-r(T-t)}N(d_2) + \mathrm{e}^{-r(T-t)} \\ &= \xi s_f(\bar{r},\tau)\mathrm{e}^{-D_0\tau}N(d_1) + \mathrm{e}^{-r\tau}N(-d_2), \end{aligned}$$

where

$$\begin{aligned} d_1 &= \left[\ln\frac{S\mathrm{e}^{(r-D_0)(T-t)}}{Z/n} + \frac{1}{2}\sigma^2(T-t)\right] \Bigg/ \left(\sigma\sqrt{T-t}\right) \\ &= \left[\ln\left(\xi s_f(\bar{r},\tau)\mathrm{e}^{(r-D_0)\tau}\right) + \frac{1}{2}\sigma^2\tau\right] \Bigg/ \left(\sigma\sqrt{\tau}\right), \\ d_2 &= d_1 - \sigma\sqrt{\tau}. \end{aligned}$$

Thus, v can be expressed as a function of $\xi s_f(\bar{r},\tau)$ and τ. Because

$$\overline{B}_c(S,r,t) = Zu(\xi,\bar{r},\tau) = Zu\left(\frac{nS}{Zs_f\left(\dfrac{r-r_l}{r_u-r_l},T-t\right)},\ \frac{r-r_l}{r_u-r_l},\ T-t\right),$$

we have

$$\begin{aligned}
\frac{\partial\overline{B}_c}{\partial t} &= Z\left(-\frac{\partial u}{\partial\tau}+\frac{\partial u}{\partial\xi}\frac{\xi}{s_f}\frac{\partial s_f}{\partial\tau}\right),\\
\frac{\partial\overline{B}_c}{\partial S} &= \frac{\partial u}{\partial\xi}\frac{n}{s_f},\\
\frac{\partial\overline{B}_c}{\partial r} &= Z\left(-\frac{\partial u}{\partial\xi}\frac{\xi}{s_f}\frac{\partial s_f}{\partial\bar{r}}+\frac{\partial u}{\partial\bar{r}}\right)\frac{1}{r_u-r_l},\\
\frac{\partial^2\overline{B}_c}{\partial S^2} &= \frac{1}{Z}\frac{\partial^2 u}{\partial\xi^2}\left(\frac{n}{s_f}\right)^2,\\
\frac{\partial^2\overline{B}_c}{\partial S\partial r} &= \left(-\frac{\partial^2 u}{\partial\xi^2}\frac{n\xi}{s_f^2}\frac{\partial s_f}{\partial\bar{r}}+\frac{\partial^2 u}{\partial\xi\partial\bar{r}}\frac{n}{s_f}-\frac{\partial u}{\partial\xi}\frac{n}{s_f^2}\frac{\partial s_f}{\partial\bar{r}}\right)\frac{1}{r_u-r_l},\\
\frac{\partial^2\overline{B}_c}{\partial r^2} &= Z\left\{\frac{\partial^2 u}{\partial\xi^2}\left(\frac{\xi}{s_f}\frac{\partial s_f}{\partial\bar{r}}\right)^2-2\frac{\partial^2 u}{\partial\xi\partial\bar{r}}\frac{\xi}{s_f}\frac{\partial s_f}{\partial\bar{r}}\right.\\
&\qquad\left.+\frac{\partial u}{\partial\xi}\left[2\frac{\xi}{s_f^2}\left(\frac{\partial s_f}{\partial\bar{r}}\right)^2-\frac{\xi}{s_f}\frac{\partial^2 s_f}{\partial\bar{r}^2}\right]+\frac{\partial^2 u}{\partial\bar{r}^2}\right\}\left(\frac{1}{r_u-r_l}\right)^2.
\end{aligned}$$

Substituting these expressions into the problem (9.18) yields

$$\begin{cases}
\dfrac{\partial u}{\partial\overline{\tau}} = \mathbf{L}_{\xi,\bar{\mathbf{r}}}u+a_7, & 0\le\xi\le1,\quad 0\le\bar{r}\le1,\quad 0\le\tau\le T,\\
u(\xi,\bar{r},0)=0, & 0\le\xi\le1,\quad 0\le\bar{r}\le1,\\
u(1,\bar{r},\tau)=s_f(\bar{r},\tau)-v(1,\bar{r},\tau), & 0\le\bar{r}\le1,\quad 0\le\tau\le T,\\
\dfrac{\partial u}{\partial\xi}(1,\bar{r},\tau)=s_f(\bar{r},\tau)-\dfrac{\partial v}{\partial\xi}(1,\bar{r},\tau), & 0\le\bar{r}\le1,\quad 0\le\tau\le T,\\
s_f(\bar{r},0)=\max(1,k/D_0), & 0\le\bar{r}\le1,
\end{cases}\tag{9.21}$$

where

$$\begin{aligned}
\mathbf{L}_{\xi,\bar{\mathbf{r}}} &= a_1\xi^2\frac{\partial^2}{\partial\xi^2}+a_2\xi w\frac{\partial^2}{\partial\xi\partial\bar{r}}+a_3w^2\frac{\partial^2}{\partial\bar{r}^2}+\left(a_4+\frac{1}{s_f}\frac{\partial s_f}{\partial\tau}\right)\xi\frac{\partial}{\partial\xi}\\
&\quad+a_5\frac{\partial}{\partial\bar{r}}+a_6,
\end{aligned}$$

$$\begin{aligned}
a_1 &= \frac{1}{2}\sigma^2 - \rho\sigma w \frac{1}{s_f(r_u-r_l)}\frac{\partial s_f}{\partial \bar{r}} + \frac{1}{2}w^2\left[\frac{1}{s_f(r_u-r_l)}\frac{\partial s_f}{\partial \bar{r}}\right]^2, \\
a_2 &= \frac{1}{r_u-r_l}\left[\rho\sigma - \frac{w}{s_f(r_u-r_l)}\frac{\partial s_f}{\partial \bar{r}}\right], \\
a_3 &= \frac{1}{2(r_u-r_l)^2}, \\
a_4 &= r - D_0 - \frac{1}{s_f(r_u-r_l)}\frac{\partial s_f}{\partial \bar{r}}(\rho\sigma w + u - \lambda w) \\
&\quad + \frac{1}{2}w^2\left\{2\left[\frac{1}{s_f(r_u-r_l)}\frac{\partial s_f}{\partial \bar{r}}\right]^2 - \frac{1}{s_f(r_u-r_l)^2}\frac{\partial^2 s_f}{\partial \bar{r}^2}\right\}, \\
a_5 &= \frac{u-\lambda w}{r_u-r_l}, \\
a_6 &= -r, \\
a_7 &= k + \rho\sigma S w\frac{\partial^2 v}{\partial S \partial r} + \frac{1}{2}w^2\frac{\partial^2 v}{\partial r^2} + (u-\lambda w)\frac{\partial v}{\partial r}.
\end{aligned}$$

We will refer to this method as the singularity-separating method for two-factor convertible bonds.

In the problem (9.21), Z and n are not involved. That is, the solution of the problem, $u(\xi, \bar{r}, \tau)$ and $s_f(\bar{r}, \tau)$, does not depend on Z or n. The problem (9.21) is called the problem for a standard convertible bond.

If the asset price S, the asset price volatility σ and the interest rate r are all considered as random variables, then we have American three-factor option problems and three-factor convertible bond problems. It is not difficult to generalize the method here to such three-dimensional problems.

9.2 Implicit Finite-Difference Methods

9.2.1 Solution of One-Dimensional Problems

The problem (9.3) can be solved by different numerical methods, for example, explicit finite-difference methods, implicit finite-difference methods, pseudo-spectral methods, and so forth. In this book, we only discuss the implicit finite-difference methods and the pseudo-spectral methods. In this subsection, we discuss how to use implicit finite-difference methods to solve free-boundary problem (9.3).

As we have pointed out, the problem we are going to solve is defined on $[0,1]\times[0,T]$ on the (ξ,τ)-plane. For simplicity, we assume that we still use the equidistant mesh given by the set of expressions (8.2). Let u_m^n stand for the value of u at the points $\xi = \xi_m \equiv m\Delta\xi$ and $\tau = \tau^n \equiv n\Delta\tau$, and s_f^n represent the value of s_f at $\tau = \tau^n$. At time $t = 0$, the function u and s_f are known, i.e., $u_m^0, m = 0, 1, \cdots, M$ and s_f^0 are known. We need to find $u_m^n, m = 0, 1, \cdots, M$ and s_f^n, $n = 1, 2, \cdots, N$.

The partial differential equation in the problem (9.3) can be discretized by

$$\begin{aligned}
&\frac{u_m^{n+1}-u_m^n}{\Delta\tau}\\
&=\frac{1}{2}\left[k_2m^2\left(u_{m+1}^{n+1}-2u_m^{n+1}+u_{m-1}^{n+1}\right)+\frac{k_1m}{2}\left(u_{m+1}^{n+1}-u_{m-1}^{n+1}\right)-k_0u_m^{n+1}\right]\\
&\quad+\frac{1}{2}\left[k_2m^2\left(u_{m+1}^{n}-2u_m^{n}+u_{m-1}^{n}\right)+\frac{k_1m}{2}\left(u_{m+1}^{n}-u_{m-1}^{n}\right)-k_0u_m^{n}\right]\\
&\quad+\frac{s_f^{n+1}-s_f^n}{\left(s_f^{n+1}+s_f^n\right)\Delta\tau}\left[\frac{m}{2}\left(u_{m+1}^{n+1}-u_{m-1}^{n+1}\right)+\frac{m}{2}\left(u_{m+1}^{n}-u_{m-1}^{n}\right)\right]
\end{aligned} \tag{9.22}$$

at $m=0,1,2,\cdots,M-1$, for $n=0,1,\cdots,N-1$. Here, in all coefficients, $\xi=m\Delta\xi$ and $\tau=(n+1/2)\Delta\tau$, so from Sect. 6.1, we know that the scheme has a truncation error of $O(\Delta\tau^2,\Delta\xi^2)$. At $m=0$, the equation actually becomes

$$\frac{u_0^{n+1}-u_0^n}{\Delta\tau}=\frac{-k_0}{2}\left(u_0^{n+1}+u_0^n\right),$$

therefore, u_{-1}^n and u_{-1}^{n+1} do not appear in the equations. The boundary conditions at $\xi=1$ in the problem (9.3) can be replaced by

$$u_M^{n+1}=g(s_f^{n+1},\tau^{n+1}), \tag{9.23}$$

and

$$\frac{3u_M^{n+1}-4u_{M-1}^{n+1}+u_{M-2}^{n+1}}{2\Delta\xi}=h\left(s_f^{n+1},\tau^{n+1}\right). \tag{9.24}$$

Here, the condition (9.23) is exact, and the truncation error of the approximate boundary condition (9.24) is $O(\Delta\xi^2)$ because the first derivative is approximated by a one-sided second-order difference scheme. In the system (9.22)–(9.24), if u_m^n, $m=0,1,\cdots,M$ and s_f^n are given, then there are $M+2$ unknowns: $u_m^{n+1}, m=0,1,\cdots,M$ and s_f^{n+1}. The number of equations in the system is also $M+2$. Therefore, we can determine $u_m^{n+1}, m=0,1,\cdots,M$ and s_f^{n+1} from this system. From the initial conditions in problem (9.3), the second and the fifth equations there, we can obtain

$$u_m^0=0,\quad m=0,1,\cdots,M$$

and

$$s_f^0=\max(1,r/D_0).$$

Consequently, starting from $n=0$, we can find the solution at τ^{n+1} from the solution at τ^n successively.

However, the system is a nonlinear one, so we cannot find the solution directly. In order to find the solution of the system, we use iteration methods. For example, Eqs. (9.22)–(9.24) can be written as

$$\begin{aligned}
&\frac{u_m^{(j)} - u_m^n}{\Delta\tau}\\
&= \frac{1}{2}\left[k_2 m^2\left(u_{m+1}^{(j)} - 2u_m^{(j)} + u_{m-1}^{(j)}\right) + \frac{k_1 m}{2}\left(u_{m+1}^{(j)} - u_{m-1}^{(j)}\right) - k_0 u_m^{(j)}\right]\\
&\quad + \frac{1}{2}\left[k_2 m^2\left(u_{m+1}^n - 2u_m^n + u_{m-1}^n\right) + \frac{k_1 m}{2}\left(u_{m+1}^n - u_{m-1}^n\right) - k_0 u_m^n\right]\\
&\quad + \frac{s_f^{(j)} - s_f^n}{\left(s_f^{(j-1)} + s_f^n\right)\Delta\tau}\left[\frac{m}{2}\left(u_{m+1}^{(j-1)} - u_{m-1}^{(j-1)}\right) + \frac{m}{2}\left(u_{m+1}^n - u_{m-1}^n\right)\right],\\
&\qquad\qquad m = 0, 1, \cdots, M-1,
\end{aligned} \tag{9.25}$$

$$u_M^{(j)} = g\left(s_f^{(j)}, \tau^{n+1}\right), \tag{9.26}$$

and

$$\frac{3u_M^{(j)} - 4u_{M-1}^{(j)} + u_{M-2}^{(j)}}{2\Delta\xi} = h\left(s_f^{(j)}, \tau^{n+1}\right), \tag{9.27}$$

where $u_m^{(j)}, s_f^{(j)}$ are the j-th iteration values of u_m^{n+1}, s_f^{n+1} respectively. In order to start an iteration, we set $u_m^{(0)} = u_m^n, m = 0, 1, \cdots, M$ and $s_f^{(0)} = s_f^n$. The system consisting of Eqs. (9.25)–(9.27) is linear for $u_m^{(j)}, m = 0, 1, \cdots, M$, and nonlinear for $s_f^{(j)}$. This system can be solved by a modified LU decomposition method described below.

The system of equations (9.25) can be rewritten as

$$\begin{aligned}
&-\frac{1}{2}\left(k_2 m^2 + \frac{k_1 m}{2}\right)\Delta\tau u_{m+1}^{(j)} + \left[1 + \left(k_2 m^2 + \frac{k_0}{2}\right)\Delta\tau\right]u_m^{(j)}\\
&-\frac{1}{2}\left(k_2 m^2 - \frac{k_1 m}{2}\right)\Delta\tau u_{m-1}^{(j)}\\
&-\frac{1}{\left(s_f^{(j-1)} + s_f^n\right)}\left[\frac{m}{2}\left(u_{m+1}^{(j-1)} - u_{m-1}^{(j-1)}\right) + \frac{m}{2}\left(u_{m+1}^n - u_{m-1}^n\right)\right]s_f^{(j)}\\
= &\frac{1}{2}\left(k_2 m^2 + \frac{k_1 m}{2}\right)\Delta\tau u_{m+1}^n + \left[1 - \left(k_2 m^2 + \frac{k_0}{2}\right)\Delta\tau\right]u_m^n\\
&+\frac{1}{2}\left(k_2 m^2 - \frac{k_1 m}{2}\right)\Delta\tau u_{m-1}^n\\
&-\frac{1}{\left(s_f^{(j-1)} + s_f^n\right)}\left[\frac{m}{2}\left(u_{m+1}^{(j-1)} - u_{m-1}^{(j-1)}\right) + \frac{m}{2}\left(u_{m+1}^n - u_{m-1}^n\right)\right]s_f^n,\\
&\qquad\qquad m = 0, 1, \cdots, M-1.
\end{aligned} \tag{9.28}$$

When $m = 0$, the equation simply becomes:

$$\left(1 + \frac{k_0}{2}\varDelta\tau\right) u_0^{(j)} = \left(1 - \frac{k_0}{2}\varDelta\tau\right) u_0^n.$$

Thus, no iteration for u_0^{n+1} is needed, and

$$u_0^{n+1} = \frac{1 - \dfrac{k_0}{2}\varDelta\tau}{1 + \dfrac{k_0}{2}\varDelta\tau} u_0^n.$$

Furthermore, noticing $u_0^0 = 0$, we have $u_0^{n+1} = 0, n = 0, 1, \cdots, N-1$. Therefore, u_0^n can be understood as a given quantity, i.e., for each iteration, there are $M+1$ unknowns: $u_m^{(j)}, m = 1, 2, \cdots, M$, and $s_f^{(j)}$. The $M+1$ unknowns satisfy a system in the following form:

$$\begin{cases} b_1 u_1^{(j)} + c_1 u_2^{(j)} + e_1 s_f^{(j)} = f_1, \\ a_m u_{m-1}^{(j)} + b_m u_m^{(j)} + c_m u_{m+1}^{(j)} + e_m s_f^{(j)} = f_m, \; m{=}2, 3, \cdots, M-1, \\ u_{\scriptscriptstyle M}^{(j)} = g\left(s_f^{(j)}, \tau^{n+1}\right), \\ d_{\scriptscriptstyle M} u_{\scriptscriptstyle M-2}^{(j)} + a_{\scriptscriptstyle M} u_{\scriptscriptstyle M-1}^{(j)} + b_{\scriptscriptstyle M} u_{\scriptscriptstyle M}^{(j)} = h\left(s_f^{(j)}, \tau^{n+1}\right). \end{cases} \tag{9.29}$$

The top $M-1$ equations of this system are linear equations for $u_m^{(j)}, m = 1, 2, \cdots, M$ and $s_f^{(j)}$. Let us rewrite the first equation as

$$u_1^{(j)} = \alpha_1 u_2^{(j)} + \beta_1 s_f^{(j)} + \gamma_1,$$

where

$$\alpha_1 = -c_1/b_1, \quad \beta_1 = -e_1/b_1, \quad \text{and} \quad \gamma_1 = f_1/b_1.$$

Suppose we have a relation in the form

$$u_{m-1}^{(j)} = \alpha_{m-1} u_m^{(j)} + \beta_{m-1} s_f^{(j)} + \gamma_{m-1}.$$

Substituting this relation into the second equation in the system (9.29) and solving the equation for $u_m^{(j)}$, we have

$$u_m^{(j)} = \alpha_m u_{m+1}^{(j)} + \beta_m s_f^{(j)} + \gamma_m,$$

where

$$\alpha_m = \frac{-c_m}{b_m + a_m\alpha_{m-1}}, \quad \beta_m = -\frac{e_m + a_m\beta_{m-1}}{b_m + a_m\alpha_{m-1}}, \quad \text{and} \quad \gamma_m = \frac{f_m - a_m\gamma_{m-1}}{b_m + a_m\alpha_{m-1}}.$$

This procedure can be done for $m = 2, 3, \cdots, M-1$ successively. Therefore, the first and second equations in the system (9.29) are equivalent to the following relation

$$u_m^{(j)} = \alpha_m u_{m+1}^{(j)} + \beta_m s_f^{(j)} + \gamma_m, \quad m = 1, 2, \cdots, M-1, \tag{9.30}$$

where

$$\begin{cases} \alpha_m = \dfrac{-c_m}{b_m + a_m \alpha_{m-1}}, \\ \beta_m = -\dfrac{e_m + a_m \beta_{m-1}}{b_m + a_m \alpha_{m-1}}, \\ \gamma_m = \dfrac{f_m - a_m \gamma_{m-1}}{b_m + a_m \alpha_{m-1}}. \end{cases} \tag{9.31}$$

Here, we define $a_1 = 0$. Using the two relations in the system (9.30) with $m = M-2$ and $M-1$, we can eliminate $u_{M-2}^{(j)}$ and $u_{M-1}^{(j)}$ in the last equation of the system (9.29) and obtain

$$\begin{aligned} & d_M \left[\alpha_{M-2}\alpha_{M-1} u_M^{(j)} + (\alpha_{M-2}\beta_{M-1} + \beta_{M-2}) s_f^{(j)} + \alpha_{M-2}\gamma_{M-1} + \gamma_{M-2}\right] \\ & + a_M \left(\alpha_{M-1} u_M^{(j)} + \beta_{M-1} s_f^{(j)} + \gamma_{M-1}\right) + b_M u_M^{(j)} \\ = {} & h\left(s_f^{(j)}, \tau^{n+1}\right). \end{aligned}$$

Substituting the third equation in the system (9.29) into this equation yields

$$\begin{aligned} & [(d_M \alpha_{M-2} + a_M)\alpha_{M-1} + b_M]\, g\left(s_f^{(j)}, \tau^{n+1}\right) \\ & + [d_M(\alpha_{M-2}\beta_{M-1} + \beta_{M-2}) + a_M \beta_{M-1}]\, s_f^{(j)} \\ & + d_M(\alpha_{M-2}\gamma_{M-1} + \gamma_{M-2}) + a_M \gamma_{M-1} \\ = {} & h\left(s_f^{(j)}, \tau^{n+1}\right). \end{aligned}$$

This is an equation for $s_f^{(j)}$, and we can use the secant method to get its solution. In order to start the secant method, we need two approximate values of $s_f^{(j)}$. For s_f^1, we can take $s_f^{(0)} = s_f^0$ and $s_f^{(1)} = s_f^0 + \varepsilon$ as the two initial values. Here, ε is a proper positive number because $s_f(t)$ is an increasing function in τ for an American call option. For $s_f^j, j = 2, 3, \cdots, N$, we can take

$$s_f^{(0)} = s_f^{j-1} + 0.75 \cdot \frac{s_f^{j-1} - s_f^{j-2}}{\tau^{j-1} - \tau^{j-2}} (\tau^j - \tau^{j-1})$$

and

$$s_f^{(1)} = s_f^{j-1} + 1.5 \cdot \frac{s_f^{j-1} - s_f^{j-2}}{\tau^{j-1} - \tau^{j-2}} (\tau^j - \tau^{j-1})$$

as the two initial values for s_f^j.

After $s_f^{(j)}$ is found, we can obtain $u_M^{(j)}$ from the third equation in the system (9.29) and $u_m^{(j)}$ from the system (9.30), $m = M-1, M-2, \cdots, 1$, successively. From the system (9.28), we know that a_m, b_m and c_m do not depend on $u_m^{(j-1)}$ and $s_f^{(j-1)}$. Thus, a_m, b_m, and c_m remain unchanged during the iteration. Furthermore, from the expression of α_m in the set of expressions (9.31), we know that α_m and $b_m + a_m \alpha_{m-1}$ also remain unchanged. f_m in the system (9.29) is a sum of two parts:

$$\frac{1}{2}\left(k_2 m^2 + \frac{k_1 m}{2}\right) \Delta\tau u_{m+1}^n + \left[1 - \left(k_2 m^2 + \frac{k_0}{2}\right) \Delta\tau\right] u_m^n$$
$$+ \frac{1}{2}\left(k_2 m^2 - \frac{k_1 m}{2}\right) \Delta\tau u_{m-1}^n$$

and

$$\frac{-1}{\left(s_f^{(j-1)} + s_f^n\right)} \left[\frac{m}{2}\left(u_{m+1}^{(j-1)} - u_{m-1}^{(j-1)}\right) + \frac{m}{2}\left(u_{m+1}^n - u_{m-1}^n\right)\right] s_f^n.$$

The first part also does not depend on $u_m^{(j-1)}$ and $s_f^{(j-1)}$. In order to make the computation efficient, all these unchanged quantities during the iteration should be computed once and stored for future use.

The iteration (9.25)–(9.27) will give a second-order accuracy if two iterations are performed. In fact, $u_m^{(1)}$ and $s_f^{(1)}$ are solutions of a first-order scheme, and $u_m^{(2)}$ and $s_f^{(2)}$ are solutions of an improved Euler method in the τ-direction, which gives second-order accuracy in the τ-direction (see any book on numerical methods for ordinary differential equations). This scheme is always second order in the ξ-direction, so the results have an accuracy of $O\left(\Delta\xi^2, \Delta\tau^2\right)$.

The way of solving the system (9.22)–(9.24) is not unique. If s_f^{n+1} is given, then the system consisting of Eqs. (9.22) and (9.23) is a system with $M+1$ linear equations and $M+1$ unknowns $u_m^{(n+1)}$, $m = 0, 1, \cdots, M$. Therefore, this system determines the dependence of u_m^{n+1} on s_f^{n+1}, i.e., the functions $u_m^{n+1}(s_f^{n+1})$, $m = 0, 1, \cdots, M$. Substituting the three functions $u_{M-2}^{n+1}(s_f^{n+1})$, $u_{M-1}^{n+1}(s_f^{n+1})$, $u_M^{n+1}(s_f^{n+1})$ into Eq. (9.24), we have an equation for s_f^{n+1}:

$$f(s_f^{n+1}) \equiv \frac{3u_M^{n+1}(s_f^{n+1}) - 4u_{M-1}^{n+1}(s_f^{n+1}) + u_{M-2}^{n+1}(s_f^{n+1})}{2\Delta\xi} - h\left(s_f^{n+1}, \tau^{n+1}\right) = 0. \tag{9.32}$$

This equation can be solved by the secant method. When using the secant method, we need to evaluate $f(s_f^{n+1})$ for a given s_f^{n+1}. This can be done as follows. Let s_f^{n+1} in Eqs. (9.22) and (9.23) take the given value, then solve the linear system consisting of Eqs. (9.22) and (9.23) by the LU decomposition method described in Sect. 6.2.1. Substituting the value of $u_M^{n+1}, u_{M-1}^{n+1}, u_{M-2}^{n+1}$ into Eq. (9.32) yields the value $f(s_f^{n+1})$. As long as we have $f(s_f^{n+1})$ for two different s_f^{n+1}, we can start the iteration. When $f(s_f^{n+1})$ is very close to zero for some given s_f^{n+1}, we obtain the solution for s_f^{n+1}, and the solution of the linear system corresponding to this s_f^{n+1} gives the values for $u_m^{n+1}, m = 0, 1, \cdots, M$. This is another way to solve the system (9.22)–(9.24).

Wu and Kwok (see [85]) suggested a similar scheme to system (9.22)–(9.24). The main difference is that they computed the option price directly.

9.2.2 Solution of Greeks

In practice, we usually need to know not only the price of the derivative security but also the sensitivities of the price to the parameters, i.e., the derivatives of the price with respect to parameters. As mentioned in Sect. 3.3.4, these derivatives are usually denoted by Greeks on the market. For example, $\dfrac{\partial V}{\partial S}, \dfrac{\partial^2 V}{\partial S^2}, \dfrac{\partial V}{\partial t}, \dfrac{\partial V}{\partial \sigma}, \dfrac{\partial V}{\partial r}$ are usually called Delta ($\varDelta$), Gamma ($\varGamma$), Theta ($\varTheta$), Vega ($\mathcal{V}$), and Rho (ρ), respectively. When we know the price of the derivative security for all S and for all $t \in [0, T]$, it is easy to get Delta, Gamma, and Theta. Here, we discuss how to get the other Greeks.

Let $V(S, t; \sigma, r, D_0)$ be the price of a derivative security. Here, we explicitly indicate that V depends on σ, r, and D_0. Thus, the sensitivities of the option price to them can be described by $\mathcal{V} = \dfrac{\partial V}{\partial \sigma}$, $\rho = \dfrac{\partial V}{\partial r}$, and $\rho_d = \dfrac{\partial V}{\partial D_0}$. In order to get $\dfrac{\partial V}{\partial \sigma}$, we can have $V(S, t; \sigma_1, r, D_0)$ and $V(S, t; \sigma_1 + \varDelta\sigma, r, D_0)$, then get $\dfrac{\partial V}{\partial \sigma}$ for a σ near σ_1 by

$$\frac{V(S, t; \sigma_1 + \varDelta\sigma, r, D_0) - V(S, t; \sigma_1, r, D_0)}{\varDelta\sigma}.$$

We also can solve the problem derived in Sect. 3.3.4 to get $\dfrac{\partial V}{\partial \sigma}$.

Let us take $\dfrac{\partial C}{\partial \sigma}$ as an example to explain how to get such a Greek. Set $\overline{C}(S, t) = C(S, t) - c(S, t)$ and suppose $\overline{C}(S, t)$ and $S_f(t)$ have been obtained. Instead of $\dfrac{\partial C}{\partial \sigma}$, let us discuss how to obtain $\dfrac{\partial \overline{C}}{\partial \sigma}$, which will be denoted by $\overline{C}_\sigma$ in this subsection. As pointed out in Sect. 3.3.4, $\dfrac{\partial C}{\partial \sigma}$ is the solution of problem (3.27). Thus, $\overline{C}_\sigma$ should satisfy

$$\begin{cases} \dfrac{\partial \overline{C}_\sigma}{\partial t} + \dfrac{1}{2}\sigma^2 S^2 \dfrac{\partial^2 \overline{C}_\sigma}{\partial S^2} + (r - D_0) S \dfrac{\partial \overline{C}_\sigma}{\partial S} - r\overline{C}_\sigma + \sigma S^2 \dfrac{\partial^2 \overline{C}}{\partial S^2} = 0, \\ \qquad\qquad\qquad\qquad\qquad\qquad 0 \le S \le S_f(t), \quad 0 \le t \le T, \\ \overline{C}_\sigma(S,T) = 0, \qquad\qquad\qquad 0 \le S \le S_f(T), \\ \overline{C}_\sigma(S_f(t),t) = -\dfrac{\partial c\,(S,t)}{\partial \sigma}, \quad 0 \le t \le T. \end{cases}$$

This is a problem with a known moving boundary. By using the transformation

$$\begin{cases} \xi = \dfrac{S}{S_f(t)}, \\ \tau = T - t \end{cases}$$

and letting

$$s_f(\tau) = \frac{1}{E} S_f(T - \tau)$$

and

$$W(\xi,\tau) = \frac{1}{E}\overline{C}_\sigma(S,t) = \frac{1}{E}\overline{C}_\sigma\left(\xi E s_f(\tau), T-\tau\right),$$

the problem above can be written as an initial-boundary value problem on a rectangular domain:

$$\begin{cases} \dfrac{\partial W}{\partial \tau} = k_2 \xi^2 \dfrac{\partial^2 W}{\partial \xi^2} + \left(k_1 + \dfrac{1}{s_f}\dfrac{ds_f}{d\tau}\right)\xi \dfrac{\partial W}{\partial \xi} - k_0 W + \sigma \xi^2 \dfrac{\partial^2 u}{\partial \xi^2}, \\ \qquad\qquad\qquad\qquad\qquad\qquad 0 \le \xi \le 1, \quad 0 \le \tau \le T, \\ W(\xi,0) = 0, \qquad\qquad\qquad\qquad 0 \le \xi \le 1, \\ W(1,\tau) = -\dfrac{1}{E}\dfrac{\partial c\,(E s_f(\tau), T-\tau)}{\partial \sigma}, \qquad 0 \le \tau \le T, \end{cases} \tag{9.33}$$

where $u(\xi,\tau)$ and $s_f(\tau)$ are the solution of the problem (9.3), and $c(S,t)$ is the price of the European call given in Sect. 2.6.5. The equation in the problem (9.33) can be discretized by

$$\begin{aligned} &\frac{W_m^{n+1} - W_m^n}{\Delta\tau} \\ =& \frac{1}{2}k_2 m^2 \left(W_{m+1}^{n+1} - 2W_m^{n+1} + W_{m-1}^{n+1} + W_{m+1}^n - 2W_m^n + W_{m-1}^n\right) \\ &+ \frac{1}{2}\left\{\left[\frac{k_1}{2} + \frac{s_f^{n+1} - s_f^n}{\left(s_f^{n+1} + s_f^n\right)\Delta\tau}\right] m \left(W_{m+1}^{n+1} - W_{m-1}^{n+1}\right) - k_0 W_m^{n+1}\right. \\ &\left. + \left[\frac{k_1}{2} + \frac{s_f^{n+1} - s_f^n}{\left(s_f^{n+1} + s_f^n\right)\Delta\tau}\right] m \left(W_{m+1}^n - W_{m-1}^n\right) - k_0 W_m^n\right\} \\ &+ \left(d_m^{n+1} + d_m^n\right)/2, \quad m = 0, 1, \cdots, M-1, \end{aligned} \tag{9.34}$$

where

$$d = \sigma\xi^2 \frac{\partial^2 u}{\partial \xi^2}.$$

The boundary condition in the problem (9.33) can be written as

$$W_{M}^{n+1} = -\frac{1}{E}\frac{\partial c\left(Es_f(\tau^{n+1}), T-\tau^{n+1}\right)}{\partial \sigma}. \tag{9.35}$$

The system (9.34) and (9.35) is a linear system for W_m^{n+1}, $m = 0, 1, \cdots, M$ and we can get W_m^{n+1} by the LU decomposition method if W_m^n, $m = 0, 1, \cdots, M$, and s_f^n, s_f^{n+1}, $\dfrac{\partial^2 u_m^n}{\partial \xi^2}$, and $\dfrac{\partial^2 u_m^{n+1}}{\partial \xi^2}$ are given. As soon as we obtain W, $\dfrac{\partial C}{\partial \sigma}$ can be found by

$$\frac{\partial C}{\partial \sigma}(S,t) = EW\left(\frac{S}{Es_f\,(T-t)}, T-t\right) + \frac{\partial c}{\partial \sigma}(S,t).$$

When u and s_f are obtained, we need to solve an initial-boundary value problem in order to get $\dfrac{\partial C}{\partial \sigma}$ if the method above is adopted. If we obtain $\dfrac{\partial C}{\partial \sigma}$ by using

$$\frac{V(S,t;\sigma_1+\Delta\sigma,r,D_0) - V(S,t;\sigma_1,r,D_0)}{\Delta\sigma},$$

then we need to solve another free-boundary problem in order to have $V(S,t;\sigma_1+\Delta\sigma,r,D_0)$ when $V(S,t;\sigma_1,r,D_0)$ has been found. The amount of work to solve a free-boundary problem by the method described in Sect. 9.2.1 is more than twice of the amount of the work to solve an initial-boundary value problem by the method given here. This is why we formulate a problem for $\overline{C}_\sigma$ and obtain $\dfrac{\partial C}{\partial \sigma}$ by solving the problem (9.33).

9.2.3 Numerical Results of Vanilla Options and Comparison

In this subsection, we will discuss some issues on the efficiency of the numerical method described in Sect. 9.2.1 and the performance of the method combined with the extrapolation technique. Here, a method combined with the extrapolation technique means that the computation is first done on a mesh by the method, then reduce the mesh sizes in the both directions by a factor of 1/2 (or other numbers) and do the computation on the second mesh again, and finally get the results by the formula (7.30) in Sect. 7.3 (or other similar formulae). The method in Sect. 9.2.1 is an implicit finite-difference version of the SSM and, for simplicity, is referred to as the SSM in this subsection. Here, we also compare the results obtained by the SSM and the combination of the SSM and the extrapolation technique with the results by other methods for two options. Finally, through the shape of the free boundaries, we point out that adopting nonuniform time steps can make the method more accurate.

Table 9.1. Parameters

Interest rates r	$0.05 \sim 0.20$ with $\Delta r = 0.025$
Volatilities σ	$0.1 \sim 0.5$ with $\Delta\sigma = 0.1$
Dividend yields D_0	$0.00 \sim 0.15$ with $\Delta D_0 = 0.025$
Expiries T	3 days, 15 days, 1$\sim$ 12 months with $\Delta T = 1$ month

Table 9.2. American call options with $r = 0.1$ and $T = 1$ year

$D_0\backslash\sigma$	0.1	0.2	0.3	0.4	0.5
0.000	–	–	–	–	–
0.025	–	–	–	–	–
0.050	–	12×6	12×6	12×6	12×6
0.075	16×8	12×8	12×8	12×8	12×8
0.100	28×14	18×10	16×10	14×8	14×8
0.125	44×16	30×12	24×10	18×8	14×8
0.150	48×18	32×12	26×10	20×8	16×8

The SSM combined with the extrapolation technique has been tested for American vanilla call and put options with various parameters. The parameters tested are given in Table 9.1. Consider the standard American call problem, i.e., the problem with $E = 1$. Suppose $r = 0.1$, $T = 1$, and require the maximum error of C for $S \in [0.9, 1.1]$ to be less than or equal to 10^{-4}. Table 9.2 lists the numbers of mesh intervals needed for different D_0 and σ in order to get such results. There, $M \times N$ means that for the second mesh, M subintervals in the ξ-direction and N time-steps in the τ-direction are taken. In Table 9.2, "–" means that for this set of parameters, and for $S \in [0.9, 1.1]$, the difference between the American call option and the European call option is less than or only a slightly greater than 10^{-4}, so no numerical method is needed. From here, we know that if the method described in Sect. 9.2.1 is used, then a coarse mesh is enough for obtaining a result with error about 10^{-4} for $S \in [0.9, 1.1]$.

Table 9.3. American put option with $r = 0.05$ and $T = 1$ year

$D_0\backslash\sigma$	0.1	0.2	0.3	0.4	0.5
0.000	40×12	24×8	18×6	14×4	12×4
0.025	36×12	26×8	16×4	14×4	12×4
0.050	32×10	22×6	16×4	12×4	12×4
0.075	–	22×6	16×4	12×4	12×4
0.100	–	–	16×4	12×4	12×4
0.125	–	–	–	12×4	12×4
0.150	–	–	–	–	12×4

As pointed out in Chap. 3, using the symmetry relations, we can have the value of an American put option from an American call option with interchanging the interest rate and dividend yield. However, we can also solve the put option problem directly. In Table 9.3, we list the numbers of mesh inter-

Table 9.4. Optimal prices for American call options

($\sigma = 0.2$, $T = 1$ and $E = 100$)

$D_0 \backslash r$	0.050	0.075	0.100	0.125	0.150
0.050	141.540893	170.943495	223.764096	277.831844	331.285054
0.075	128.372144	137.454215	155.027353	186.574326	222.166283
0.100	122.069175	127.037558	134.599182	147.295598	168.445693
0.125	118.119037	121.403431	125.903014	132.417054	142.448401
0.150	115.346132	117.723481	120.800277	124.918028	130.659131

vals needed in order to have an accuracy of about 10^{-4} for $S \in [0.9, 1.1]$ and $r = 0.05$. Thus, for both American call and put options, only a coarse mesh is needed in order to get the accuracy usually needed. From the price of the call option with $r = 0.1$ and $D_0 = 0.05$, we can have the value of the put option with $r = 0.05$ and $D_0 = 0.1$. From Tables 9.2 and 9.3, we know that in order to get the price of the put option with $r = 0.05$, $D_0 = 0.1$, and $\sigma = 0.3$, we can take a 16×4 mesh if we solve a put problem directly or we can take a 12×6 mesh if we solve a corresponding call problem and get the solution using the symmetry relations. For these two meshes, the CPU times needed are very close, so we can choose either way. However, if we already have a code to compute American call option prices, then using the second way would be a better choice since only very little code needs to be added.

With this method, it is not difficult to get results with a high accuracy. In Table 9.4, the optimal price for American call options with various r and D_0 are listed. Analysis shows these results to be exact to at least seven digits (see [98]).

Table 9.5. American call option

($r = 0.1$, $\sigma = 0.2$, $D_0 = 0.05$, $T = 1\,\text{year}$, $S = E = 100$,
and the exact value $= 9.94092345\cdots$)

	Without extrapolation			With extrapolation		
Meshes	Results	Errors	CPU(s)	Results	|Errors|	CPU(s)
32×2	9.941663	−0.000739	0.00025	9.940902	0.000021	0.00045
64×4	9.941097	−0.000174	0.00083	9.940908	0.000015	0.0012
128×8	9.940962	−0.000038	0.0027	9.940917	0.000006	0.0038
256×16	9.940932	−0.000009	0.0099	9.9409225	0.000001	0.0125

Now let us discuss the convergence rate of the SSM. Let $r = 0.1, \sigma = 0.2, D_0 = 0.05, T = 1\,\text{year}$, and $S = E = 100$. In order to study the convergence rate, we have to know the exact solution. We do not have the exact solution, but we can get a solution with a very high accuracy and obtain the first few digits of the exact solution. For the parameters given above, our computation shows the exact call option price $C = 9.94092345\cdots$ and the exact put option price $P = 5.92827717\cdots$. As long as we have such a solution, we can find the error of any solution up to the eighth decimal. In Table 9.5,

Table 9.6. American put options

($r = 0.1, \sigma = 0.2, D_0 = 0.05, T = 1\,\text{year}, S = E = 100$,
and the exact value $= 5.92827717\cdots$)

Meshes	Without extrapolation			With extrapolation		
	Results	Errors	CPU(s)	Results	|Errors|	CPU(s)
12×4	5.968338	−0.040060	0.00035	5.925575	0.002702	0.00065
24×8	5.937883	−0.009606	0.00084	5.927732	0.000545	0.0014
48×16	5.930477	−0.002200	0.0025	5.928008	0.000269	0.0035
96×32	5.928819	−0.000542	0.0078	5.928266	0.000011	0.0108
192×64	5.928409	−0.000132	0.0300	5.928272	0.000005	0.0387
384×128	5.928310	−0.000033	0.1200	5.9282767	0.0000005	0.1400

the results without using the extrapolation technique for four meshes and the errors up to the sixth decimal are listed on the second and third columns from the left. When the numbers of intervals in the both directions is doubled, the error is reduced by a factor about 1/4. This means that the error is $O\left(\varDelta\xi^2, \varDelta\tau^2\right)$. Therefore, it has a second-order convergence rate. In Table 9.6, the results and errors for the put option are given. From there, we see that the convergence rate is also second order for the put option.

A method with a high convergence rate has a better performance if the mesh size is small enough. However, if the mesh size is not small enough, it might not be true. For a fixed mesh, the computational amount of work is different for different methods. Thus, from a practical point of view, a method should be judged by its performance. Therefore, we also list the CPU time needed to perform such a computation on a Space Ultra 10 computer for each mesh in Tables 9.5 and 9.6.

Using these data on errors and CPU times in Tables 9.5 and 9.6, the data given for PEFDII, Binomial, PSOR, and PIFDII in Chap. 8, the graphs of $\log_{10}$(CPU time in second) versus $\log_{10}$(error) for call and put options are plotted in Figs. 9.2 and 9.3, respectively. On these two figures, the lower the point, the better the performance because a lower point means that for a fixed error, it needs less CPU time. From there, we can see that the singularity-separating method (SSM) has the best performance for these two cases if the error required is less than 10^{-2}. Moreover, the higher the accuracy required, the greater the advantage of the SSM.

If the SSM is combined with the extrapolation technique, then the performance is even better. In order to explain this, the results, errors, and CPU times when the SSM is combined with the extrapolation technique are listed in the right three columns of Tables 9.5 and 9.6, and the corresponding graphs of $\log_{10}$(CPU time in second) versus $\log_{10}$(error) are also plotted in Figs. 9.2 and 9.3. There, SSME stands for the singularity-separating method with the extrapolation technique. From here, we can see that the extrapolation technique is very useful. At the beginning of this subsection, we showed that for various parameters, the SSM with the extrapolation technique could give very

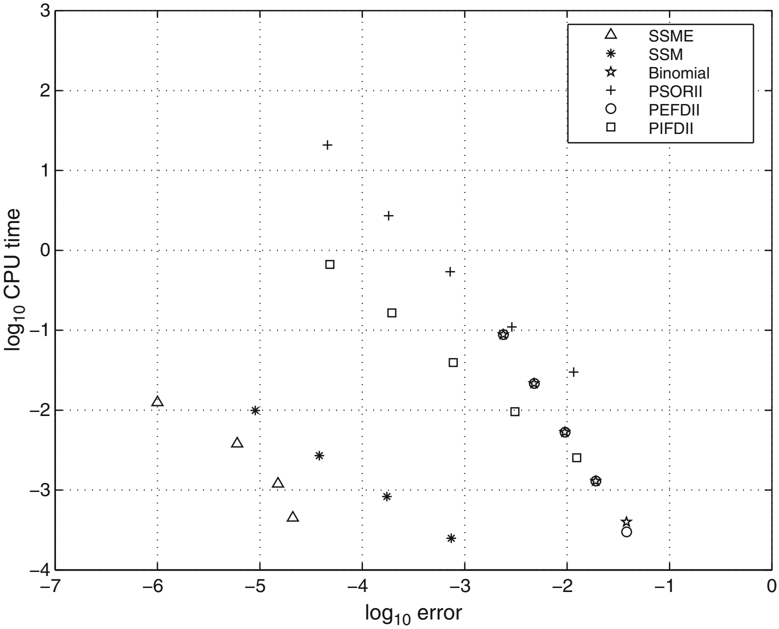

Fig. 9.2. Graphs of CPU time versus error for a call option, $S = 100$

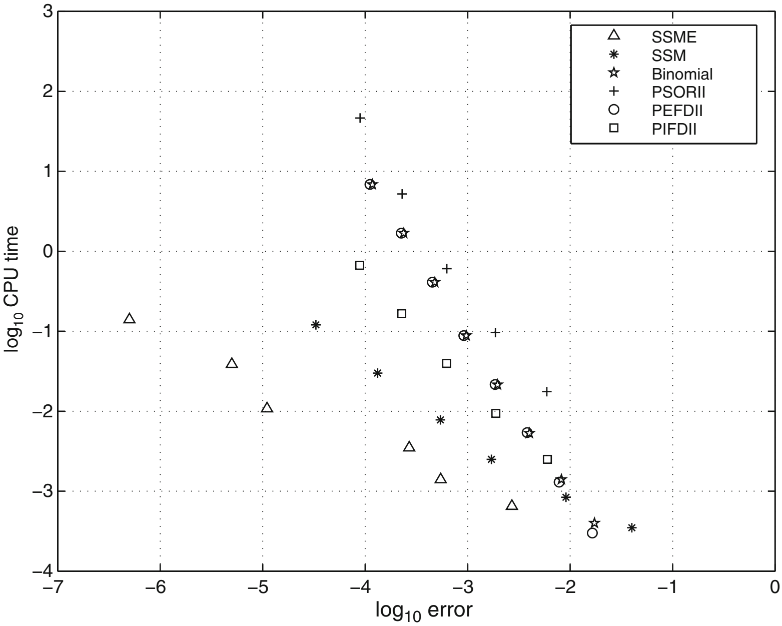

Fig. 9.3. Graphs of CPU time versus error for a put option, $S = 100$

good results on quite coarse meshes. This is because due to the error function being quite smooth, the extrapolation technique is always helpful when combined with SSM.

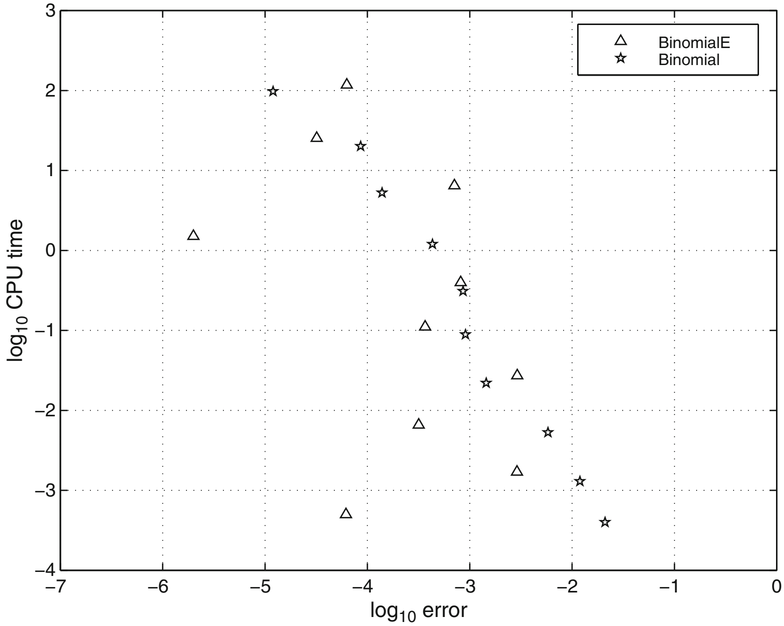

Fig. 9.4. Graphs of CPU time versus error for a call option, $S = 110$

Here, we would like to point out that the extrapolation technique is not always helpful. Let us find out if the performance is improved when the binomial method is combined with the extrapolation technique. Consider a call option with $r = 0.1$, $\sigma = 0.2$, $D_0 = 0.05$, $T = 1$, and $E = 100$. In Fig. 9.4 for $S = 110$ we plot the graphs of $\log_{10}$(CPU time in second) versus $\log_{10}$(error) for the binomial method with and without extrapolation. There, "Binomial" and "BinomialE" mean the binomial method and the binomial method with extrapolation technique. From there, we can see that on some meshes, the extrapolation technique improves the results, but on other meshes, it makes the results worse. In order to have some details about why this happens, the data of the errors and the CPU times are listed for the two cases in Table 9.7. As a first-order method, the error should be reduced by a factor about 1/2 when the number of time steps is doubled. Because the error function is not smooth due to the non-smoothness of the solution, from the table we see that from one mesh size to another, the error before extrapolation does not always show such a property and sometimes the sign of the error even changes. Thus, when the extrapolation technique is used, the error increases for some cases if the sign is unchanged and always increases if the sign changes. This phenomena occurs even if the mesh size is very small. Therefore, the extrapolation technique is not always helpful for the binomial method. However, Broadie and Detemple in [14] suggested an improved binomial method called the binomial Black and Scholes method (BBS). Examples show that the error of BBS decreases and does not change its sign when the mesh size decreases. As long as it is true, the extrapolation technique is helpful for the BBS method.

Table 9.7. American call option (binomial method)

($r = 0.1, \sigma = 0.2, D_0 = 0.05, T = 1\,$year, $E = 100$, $S = 110$, and the exact value $= 16.8016638\cdots$)

Numbers of	Without extrapolation			With extrapolation		
time steps	Results	Errors	CPU(s)	Results	|Errors|	CPU(s)
50	16.822670	−0.021006	0.0004	16.801602	0.000062	0.0005
100	16.813618	−0.011954	0.0013	16.804566	0.002902	0.0017
200	16.807482	−0.005818	0.0053	16.801346	0.000318	0.0066
400	16.803114	−0.001450	0.0220	16.798746	0.002918	0.0273
800	16.802573	−0.000909	0.0880	16.802032	0.000370	0.1110
1,600	16.802526	−0.000862	0.3100	16.802479	0.000817	0.3980
3,200	16.802096	−0.000432	1.2000	16.801666	0.000002	1.5100
6,400	16.801525	+0.000139	5.2700	16.800953	0.000710	6.4700
12,800	16.801578	+0.000086	20.100	16.801632	0.000032	25.370
25,600	16.801652	−0.000012	97.600	16.801727	0.000063	117.70

Finally, in this subsection we give two graphs on the location of the free boundaries. In Figs. 9.5 and 9.6, the location of the free boundaries is plotted for three call options and three put options, respectively. There, $E = 100, \sigma = 0.24$, and $t = 0 \sim 10$. The other parameters for the three call options are $(r = 0, D_0 = 0.06)$, $(r = 0.06, D_0 = 0.06)$, and $(r = 0.06, D_0 = 0.03)$, and for the three put options they are $(r = 0.06, D_0 = 0)$, $(r = 0.06, D_0 = 0.06)$, and $(r = 0.03, D_0 = 0.06)$. For all the cases, the location of the free boundary moves quite fast at $t \approx T$. Therefore, the time step at $t \approx T$ should be smaller than the time step at $t << T$. In order to make computation more efficient, the time step used for all the numerical results in this subsection is not constant. When we need to find the solution for $\tau \in [0, T]$ and the total number of time step is N, then τ^n is determined by the formula

$$\tau^n = \frac{n^2}{N^2}T, \quad n = 0, 1, \cdots, N$$

and from τ^n to τ^{n+1}, the time step is $\tau^{n+1} - \tau^n = \dfrac{2n+1}{N^2}T$. When such variable time steps are used, the extrapolation technique can still be used, which is left as an exercise problem for readers to show.

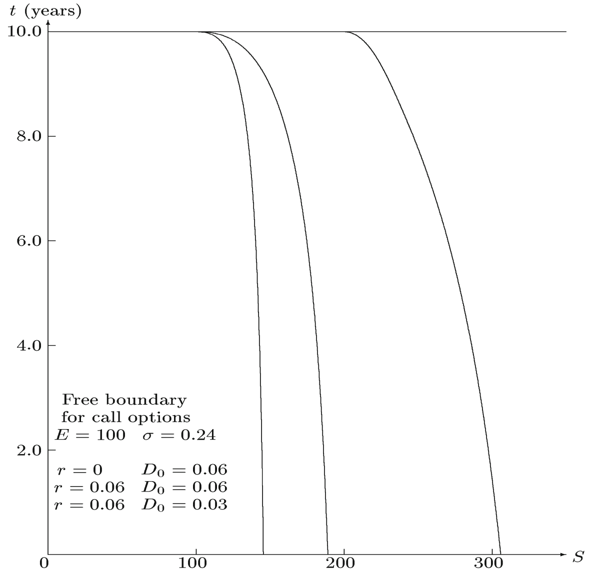

Fig. 9.5. Locations of free boundaries of call options in the (S, t)-plane [The parameters for these curves from the left to the right are (r=0, D_0=0.06), (r=0.06, D_0=0.06), and (r=0.06, D_0=0.03)]

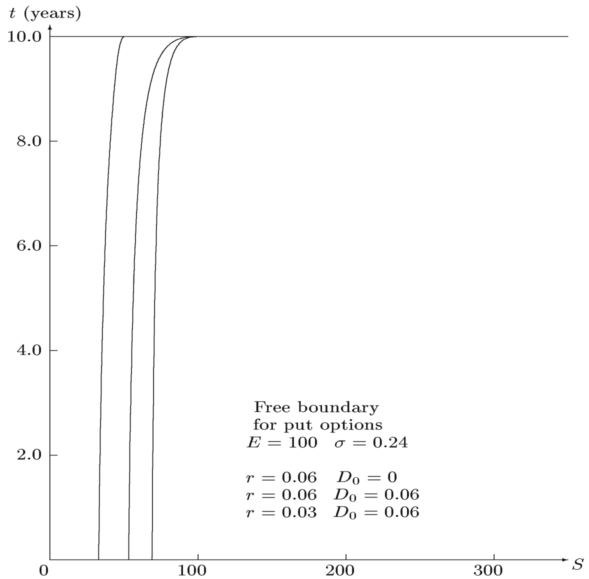

Fig. 9.6. Locations of free boundaries of put options in the (S, t)-plane. [The parameters for these curves from the right to the left are (r=0.06, D_0=0), (r=0.06, D_0=0.06), and (r=0.03, D_0=0.06)]

9.2.4 Solution and Numerical Results of Exotic Options

After making a slight change, the implicit finite-difference method described in Sect. 9.2.1 still can be used for computing free-boundary problems for American-style barrier, Asian, and lookback options. Here, we first show some results for American barrier and lookback options. Then, we discuss some modifications we have used when we compute the prices of American-style Asian options and give some results on Asian options.

For $S = 60 \sim 160$, the prices of American down-and-out call options with $B_l = 80$, 85, 90, 95 and the price of American down-and-out call option with $B_l = 0$—the price of the American vanilla call option—have been shown in Fig. 4.1. There, the parameters are $r = 0.1$, $D_0 = 0.05$, $\sigma = 0.2$, $T = 1$ year, and $E = 100$. Here, for $S = 60 \sim 160$ and for the same parameters, the prices of American up-and-out put options with $B_u = 105$, 110, 115, 120 and the price of American up-and-out put option with $B_u = \infty$—the price of the American vanilla put option—are represented in Fig. 9.7. From these curves, we see again that the price of a barrier option is less than a vanilla option. The reason is still that the holder of a barrier option has less rights than a holder of a vanilla option. In Sect. 4.2.3, we have pointed out that for call options, the higher the lower barrier B_l, the less the rights and the cheaper the option. Here, we give some data to show how big the difference between the barrier options and the vanilla options is. In Table 9.8, the prices of the American down-and-out and vanilla call options for S= 80, 85, 90, 95, 100, 105, 110, 115, and 120 are listed. From the data, we can see that the difference is significant for most of the cases.

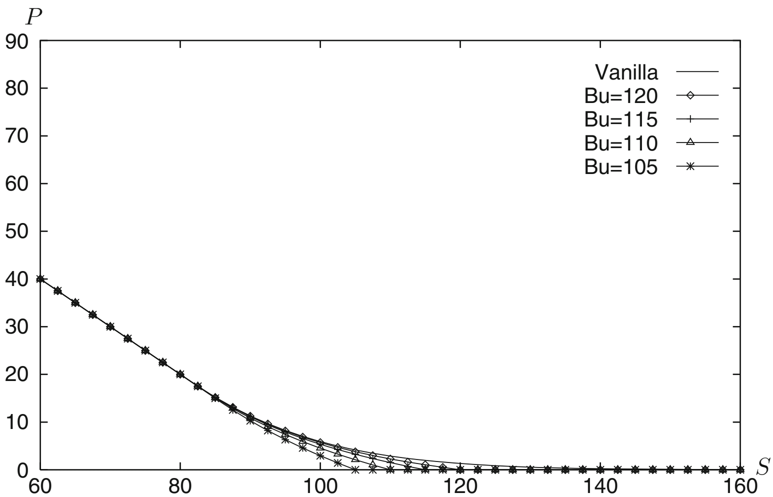

Fig. 9.7. Values of American vanilla put option and American up-and-out put options with $B_u = 105$, 100, 115, 120 ($r = 0.1$, $D_0 = 0.05$, $\sigma = 0.2$, $T = 1$ year, and $E = 100$)

Table 9.8. American down-and-out call option

($r = 0.1$, $\sigma = 0.20$, $D_0 = 0.05$, $T = 1$, and $E = 100$)

S	Vanilla	$B_l = 80$	$B_l = 85$	$B_l = 90$	$B_l = 95$
80	1.769	0	0	0	0
85	3.057	2.181	0	0	0
90	4.843	4.418	3.165	0	0
95	7.145	6.943	6.242	4.251	0
100	9.941	9.846	9.464	8.243	5.361
105	13.182	13.138	12.934	12.202	10.292
110	16.802	16.782	16.674	16.244	15.005
115	20.728	20.719	20.663	20.415	19.626
120	24.893	24.889	24.861	24.720	24.226

In Sect. 4.4, for an American lookback strike call option, the values $W(\eta, t)$ as functions of η for $t = 0, 0.2, 0.4, 0.6, 0.8$ are shown in Fig. 4.7. Here, for an American lookback strike put option, similar curves are represented in Fig. 9.8. From this figure, we know that $W(\eta, t) = V(S, H, t)/S$ is an increasing function in $\eta = H/S$. That is, if S is fixed, then $V(S, H, t)$ is an increasing function in H. This is because the payoff $\max(H - S, 0)$ increases for $S \leq H$ as H increases. The highest price up to time t is of course greater than or equal to the price at time t. Thus, $\eta = H/S$ must be greater than or equal to 1. Consequently, $W(\eta, t)$ is defined only for $\eta \geq 1$ and for a fixed t, the price of the option has a minimum at $\eta = 1$. In Fig. 9.8, we can observe this being true and the value of $W(\eta, t)$ at $\eta = 1$ and $t = 0$ being about 0.16. This means that the minimum price at $t = 0$ is about 16 % (the actual value is 16.37 %) of S. From the last subsection, we know that the value of the vanilla put option with $S = E$ is 5.93 % of S. Hence, the price of an American lookback strike put option is much higher than the price of an American vanilla put option. The reason is that the holder of an American lookback strike put option can sell a stock at any time t for the maximum price during the time interval $[0, t]$, whereas a holder of an American vanilla put option can sell a stock at any time t for the price at time t that is always less than or equal to the maximum price during the time interval $[0, t]$.

In Fig. 9.9, the location of the free boundary of the American lookback strike put option is given. In Fig. 4.8, a similar result for a call is represented. In Sect. 3.3.1, it has been shown that the locations of free boundaries for vanilla options are monotone functions in t. In fact, this is also true for American lookback strike options. Figures 4.8 and 9.9 show this fact. In Sect. 3.3.1, we also have pointed out that the monotonicities of the free boundary and of the price with respect to t are related. This reflects that $W(\eta, t)$ should be monotone functions of t for any fixed η. Figures 4.7 and 9.8 show this feature.

For details on how to compute American barrier and lookback options by using SSM, see [18, 99].

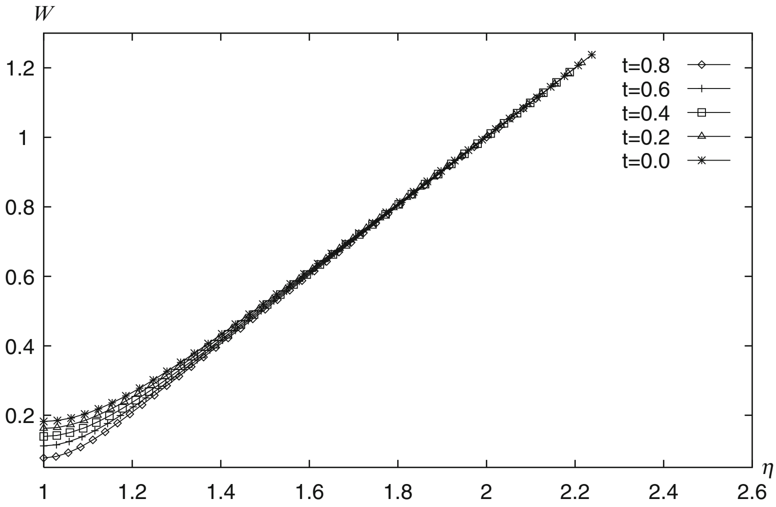

Fig. 9.8. $W(\eta, t)$ of an American lookback strike put option ($r = 0.05$, $D_0 = 0.1$, and $\sigma = 0.2$)

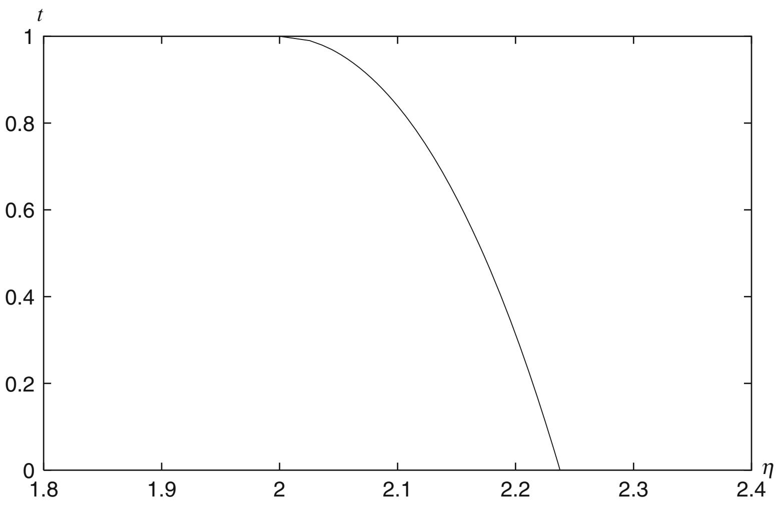

Fig. 9.9. The free boundary of an American lookback strike put option ($r = 0.05$, $D_0 = 0.1$, and $\sigma = 0.2$)

Now let us look at average options. In Fig. 9.10, the line with $*$ gives the solution $W(\eta, 0)$ for an American average strike call option by the singularity-separating method with the implicit finite-difference method (SSMIMP), similar to that described in Sect. 9.2.1. The result of a put option with the same parameters is given in Fig. 9.11 also by a curve with $*$. These two curves are almost horizontal straight lines except near one of the boundaries because

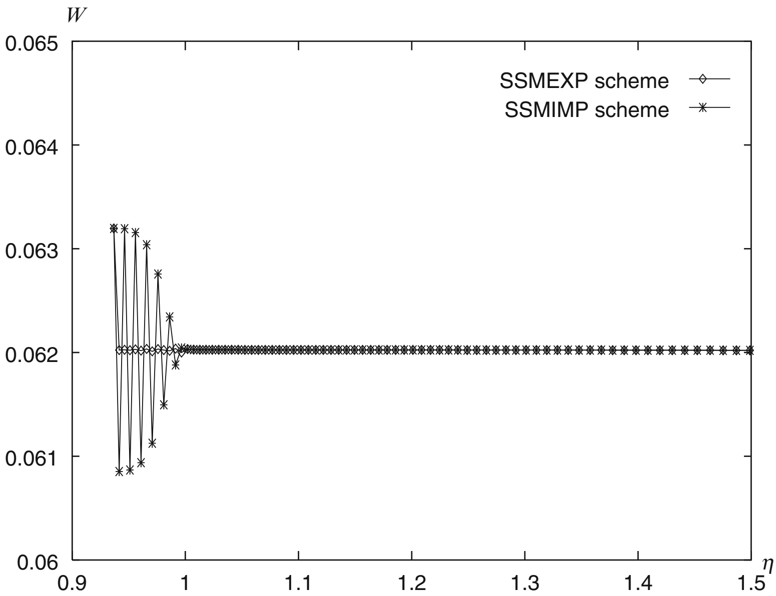

Fig. 9.10. $W(\eta, t)$ of an American average strike call option ($r = 0.1$, $D_0 = 0.1$, $\sigma = 0.2$, and $t = 0$)

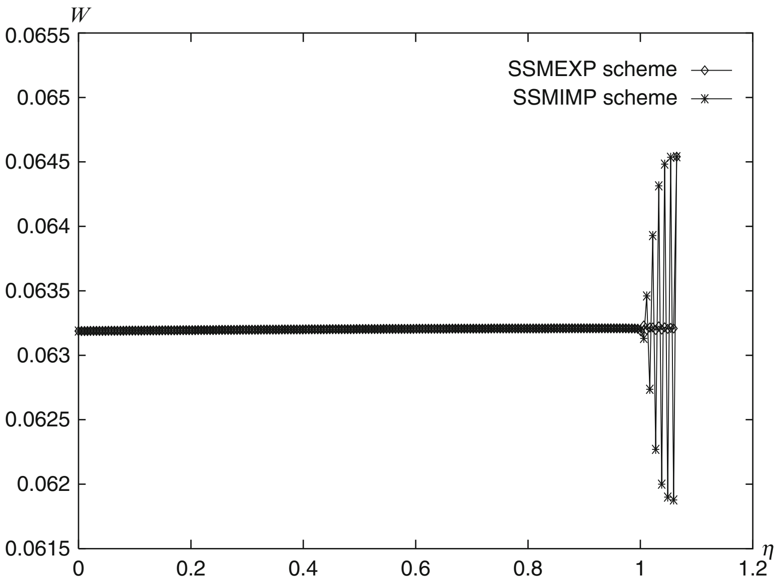

Fig. 9.11. $W(\eta, t)$ of an American average strike put option ($r = 0.1$, $D_0 = 0.1$, $\sigma = 0.2$, and $t = 0$)

there is a term $(1-\eta)/t$ in the partial differential equation of Asian options. Actually, this boundary is the free boundary. Near the free boundary, the exact solution changes very rapidly, and the numerical solution has oscillations. At time $t=0$, the average price of the stock is always equal to the price of the stock, so we actually only need the value of $W(\eta, t)$ at $\eta = A/S = 1$, which is the level of the horizontal straight line. Therefore, we can still have a good result for $W(1,t)$ by finding the level of the horizontal straight line. However, in order to get rid of the oscillations and make the entire result nicer, we use the following scheme to approximate the first and second derivatives with respect to ξ in the partial differential equation.

Let us consider the equation:

$$a_i^{n+1/2}\frac{\partial^2 U}{\partial \xi^2} + b_i^{n+1/2}\frac{\partial U}{\partial \xi} + c_i^{n+1/2}U = 0.$$

Its characteristic equation is

$$a_i^{n+1/2}\lambda^2 + b_i^{n+1/2}\lambda + c_i^{n+1/2} = 0. \tag{9.36}$$

When $a_i^{n+1/2} > 0$ and $c_i^{n+1/2} < 0$, it has two distinct real roots:

$$\begin{cases} \lambda_{1,i} = \dfrac{-b_i^{n+1/2} + \sqrt{\left(b_i^{n+1/2}\right)^2 - 4a_i^{n+1/2}c_i^{n+1/2}}}{2a_i^{n+1/2}}, \\ \lambda_{2,i} = \dfrac{-b_i^{n+1/2} - \sqrt{\left(b_i^{n+1/2}\right)^2 - 4a_i^{n+1/2}c_i^{n+1/2}}}{2a_i^{n+1/2}}. \end{cases}$$

Let

$$\varphi(\xi) = \mathrm{e}^{\lambda_{1,i}(\xi-\xi_i)}, \quad \psi(\xi) = \mathrm{e}^{\lambda_{2,i}(\xi-\xi_i)} \tag{9.37}$$

be the local basis functions. Then, on a subinterval $[\xi_{i-1}, \xi_{i+1}]$ near ξ_i, a function $W(\xi, \tau^{n+1/2})$ can be approximated by

$$\alpha_i\varphi(\xi) + \beta_i\psi(\xi) + \gamma_i, \tag{9.38}$$

where α_i, β_i, and γ_i are determined by the following conditions:

$$\begin{cases} \alpha_i\varphi(\xi_{i-1}) + \beta_i\psi(\xi_{i-1}) + \gamma_i = W_{i-1}^{n+1/2}, \\ \alpha_i\varphi(\xi_i) + \beta_i\psi(\xi_i) + \gamma_i = W_i^{n+1/2}, \\ \alpha_i\varphi(\xi_{i+1}) + \beta_i\psi(\xi_{i+1}) + \gamma_i = W_{i+1}^{n+1/2}. \end{cases}$$

From these conditions, we have

$$\begin{cases} \alpha_i = \alpha_{1,i} W_{i-1}^{n+1/2} + \alpha_{2,i} W_i^{n+1/2} + \alpha_{3,i} W_{i+1}^{n+1/2}, \\ \beta_i = \beta_{1,i} W_{i-1}^{n+1/2} + \beta_{2,i} W_i^{n+1/2} + \beta_{3,i} W_{i+1}^{n+1/2}, \\ \gamma_i = W_i^{n+1/2} - \alpha_i - \beta_i, \end{cases} \tag{9.39}$$

where

$$\begin{aligned} \alpha_{1,i} &= \left[\psi(\xi_{i+1}) - \psi(\xi_i)\right]/G_i, \\ \alpha_{2,i} &= \left[\psi(\xi_{i-1}) - \psi(\xi_{i+1})\right]/G_i, \\ \alpha_{3,i} &= \left[\psi(\xi_i) - \psi(\xi_{i-1})\right]/G_i, \\ \beta_{1,i} &= \left[\varphi(\xi_i) - \varphi(\xi_{i+1})\right]/G_i, \\ \beta_{2,i} &= \left[\varphi(\xi_{i+1}) - \varphi(\xi_{i-1})\right]/G_i, \\ \beta_{3,i} &= \left[\varphi(\xi_{i-1}) - \varphi(\xi_i)\right]/G_i, \\ G_i &= \left[\varphi(\xi_{i-1}) - \varphi(\xi_i)\right]\left[\psi(\xi_{i+1}) - \psi(\xi_i)\right] \\ &\quad - \left[\varphi(\xi_{i+1}) - \varphi(\xi_i)\right]\left[\psi(\xi_{i-1}) - \psi(\xi_i)\right]. \end{aligned}$$

If $b_i^{n+1/2}$ is a very large positive number, then $|\lambda_{2,i}|$ is very large and the exponential function $\psi(\xi)$ changes very rapidly. Therefore, even if $W(\xi, \tau^{n+1/2})$ changes very rapidly, as long as its behavior is close to an exponential function, (9.38) can still give a very good approximation not only for the function itself but also for its derivatives. Differentiating function (9.38) with respect to ξ yields

$$\frac{\partial W}{\partial \xi} \approx \alpha_i \lambda_{1,i} \varphi(\xi) + \beta_i \lambda_{2,i} \psi(\xi), \quad \frac{\partial^2 W}{\partial \xi^2} \approx \alpha_i \lambda_{1,i}^2 \varphi(\xi) + \beta_i \lambda_{2,i}^2 \psi(\xi). \tag{9.40}$$

Therefore, we can have the following approximation:

$$\begin{aligned} & a_i^{n+1/2} \frac{\partial^2 W_i^{n+1/2}}{\partial \xi^2} + b_i^{n+1/2} \frac{\partial W_i^{n+1/2}}{\partial \xi} + c_i^{n+1/2} W_i^{n+1/2} \\ = {}& a_i^{n+1/2} \left[\alpha_i \lambda_{1,i}^2 \varphi(\xi_i) + \beta_i \lambda_{2,i}^2 \psi(\xi_i)\right] \\ & + b_i^{n+1/2} \left[\alpha_i \lambda_{1,i} \varphi(\xi_i) + \beta_i \lambda_{2,i} \psi(\xi_i)\right] + c_i^{n+1/2} W_i^{n+1/2} \\ = {}& -c_i^{n+1/2} \left[\alpha_i + \beta_i - W_i^{n+1/2}\right] \\ = {}& -c_i^{n+1/2} \Big[(\alpha_{1,i} + \beta_{1,i}) W_{i-1}^{n+1/2} + (\alpha_{2,i} + \beta_{2,i}) W_i^{n+1/2} \\ & \qquad + (\alpha_{3,i} + \beta_{3,i}) W_{i+1}^{n+1/2} - W_i^{n+1/2}\Big], \end{aligned}$$

where we have used the facts that $\lambda_{1,i}$ and $\lambda_{2,i}$ are roots of Eq. (9.36) and that the expressions of α_i and β_i are given by the set of expressions (9.39).

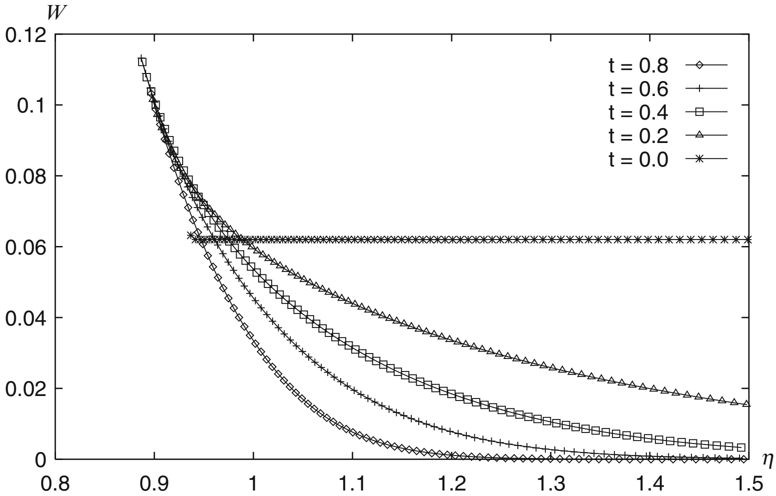

Fig. 9.12. $W(\eta, t)$ of an American average strike call option ($r = 0.1$, $D_0 = 0.1$, $\sigma = 0.2$, and $\alpha = 1$)

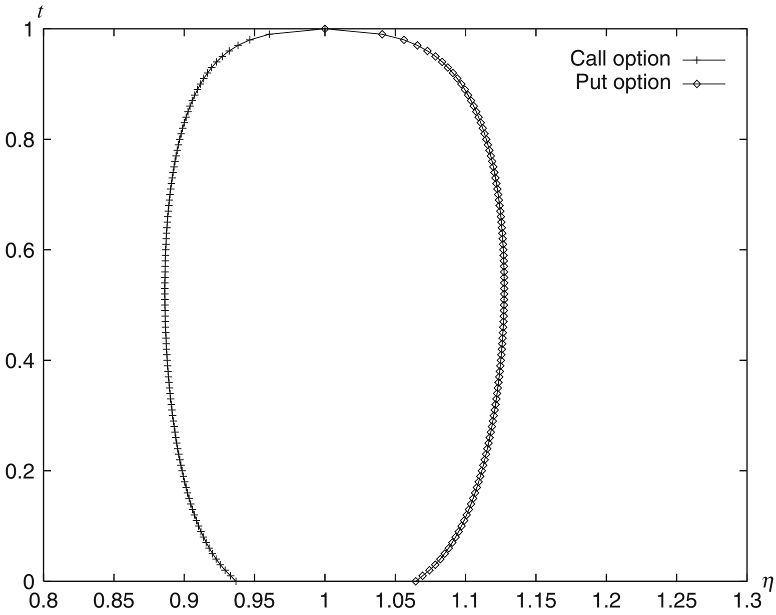

Fig. 9.13. Free boundaries of American average strike options ($r = 0.1$, $D_0 = 0.1$, $\sigma = 0.2$, and $\alpha = 1$)

If the first and second derivatives with respect to ξ in the partial differential equation for an Asian option are not discretized by central schemes but by the set of expressions (9.40), then we have a new scheme, which is called the exponential scheme, and the method is referred to as the singularity-separating method with an exponential scheme (see [17]) and abbreviated as SSMEXP. The results of the exponential scheme are also given in Figs. 9.10 and 9.11 by the curves with $\diamond$, which have no oscillations. From these curves, we see that this scheme improves the results. Therefore, in order to get the price of Asian options, we use this scheme. In Fig. 9.12, for an American average strike call option, the values of $W(\eta, t)$ as functions of η for $t = 0$, 0.2, 0.4, 0.6, 0.8 are given. The price of the option is $V(S, A, t) = AW(S/A, t)$. Because $A = S$ at $t = 0$, in order to find the value of the option at $S = \$100$ and $t = 0$, we need to find $\$100W(1, 0)$. From Fig. 9.12, we see that it is a little higher than $\$100 \times 0.06 = \$6.00$ (from the data we have it is \$6.20). In Fig. 4.3, the values of the American average strike put option with the same parameters are represented. From there, we see that the price for an American average strike put option with the same parameters at $t = 0$ is also a little higher than $\$100 \times 0.06 = \$6.00$ (from the data we have it is \$6.32). Thus, the difference between the call and put prices for the average options is much smaller than for the vanilla options. In Fig. 9.13, the free boundaries of the average strike call and put options are given, which shows that the locations of free boundaries are not monotone functions in t for the average strike options. This indicates that $W(\eta, t)$ is not a monotone function of t for a fixed η, which can be seen in Figs. 4.3 and 9.12.

9.2.5 Solution of Two-Dimensional Problems

In this subsection, we will discuss how to price two-factor vanilla American call options numerically. Here, "two-factor" means that both S and σ are random variables. If D_0 is not equal to zero, then pricing two-factor vanilla American options involves solving two-dimensional free-boundary problems. In what follows, we will give some details on implicit finite-difference methods for two-dimensional free-boundary problems. For the American call, the corresponding free-boundary problem is given by the problem (9.7) or the problem (9.9). Those problems can be converted into a problem on a rectangular domain, for example, the problem (9.9) can be converted into the problem (9.11). Therefore, determining the price on the domain $[0, S_f(\sigma, t)] \times [\sigma_l, \sigma_u] \times [0, T]$ can be reduced to solving the problem (9.11) on a rectangular domain $[0, 1] \times [\sigma_l, \sigma_u] \times [0, T]$ in the (ξ, σ, τ)-space.

We use equidistant grid points on the rectangular domain. Let $\Delta\xi = 1/M$, $\Delta\sigma = (\sigma_u - \sigma_l)/I$, and $\Delta\tau = T/N$ be the mesh sizes in the ξ-, σ-, and τ-directions, respectively, where M, I and N are positive integers. We thus have $M+1$, $I+1$, and $N+1$ nodes in the ξ-, σ-, and τ-directions, respectively. The $M + 1$ nodes in the ξ-direction are $\xi_m = m\Delta\xi$, $m = 0, 1, \cdots, M$, the $I + 1$ nodes in the σ-direction are $\sigma_i = \sigma_l + i\Delta\sigma$, $i = 0, 1, 2, \cdots, I$, and the $N + 1$ nodes in the τ-direction are $\tau^n = n\Delta\tau$, $n = 0, 1, \cdots, N$. In what follows, we also define $\tau^{n+1/2} = (n + 1/2)\Delta\tau$. Let $u^n_{m,i}$ stand for the approximate value of u at $\xi = \xi_m$, $\sigma = \sigma_i$, and $\tau = \tau^n$ and $s^n_{f,i}$ denote the approximate value of s_f at $\sigma = \sigma_i$, and $\tau = \tau^n$.

If $\xi \neq 1$, $\sigma \neq \sigma_l$, and $\sigma \neq \sigma_u$, then at a point $(\xi_m, \sigma_i, \tau^{n+1/2})$, the partial differential equation in the problem (9.11) can be discretized by the following second-order approximation:

$$
\begin{aligned}
&\frac{u_{m,i}^{n+1} - u_{m,i}^{n}}{\Delta\tau} \\
= &\frac{a_1 m^2}{2}\left(u_{m+1,i}^{n+1} - 2u_{m,i}^{n+1} + u_{m-1,i}^{n+1} + u_{m+1,i}^{n} - 2u_{m,i}^{n} + u_{m-1,i}^{n}\right) \\
&+\frac{a_2 qm}{8\Delta\sigma}(u_{m+1,i+1}^{n+1} - u_{m+1,i-1}^{n+1} - u_{m-1,i+1}^{n+1} + u_{m-1,i-1}^{n+1} \\
&\qquad +u_{m+1,i+1}^{n} - u_{m+1,i-1}^{n} - u_{m-1,i+1}^{n} + u_{m-1,i-1}^{n}) \\
&+\frac{a_3 q^2}{2\Delta\sigma^2}(u_{m,i+1}^{n+1} - 2u_{m,i}^{n+1} + u_{m,i-1}^{n+1} + u_{m,i+1}^{n} - 2u_{m,i}^{n} + u_{m,i-1}^{n}) \\
&+\frac{a_4 m}{4}(u_{m+1,i}^{n+1} - u_{m-1,i}^{n+1} + u_{m+1,i}^{n} - u_{m-1,i}^{n}) \\
&+\frac{a_5}{4\Delta\sigma}(u_{m,i+1}^{n+1} - u_{m,i-1}^{n+1} + u_{m,i+1}^{n} - u_{m,i-1}^{n}) \\
&+\frac{a_6}{2}(u_{m,i}^{n+1} + u_{m,i}^{n}) + a_7, \\
&\qquad m = 0, 1, \cdots, M-1, \quad i = 1, 2, \cdots, I-1.
\end{aligned}
\tag{9.41}
$$

Here, q and all the coefficients a_1–a_7 should be evaluated at ξ_m, σ_i and $\tau^{n+1/2}$ in order to guarantee second-order accuracy. For a_1–a_7, the expressions are

$$
\begin{aligned}
a_{1,m,i}^{n+1/2} &= \frac{1}{2}(\sigma_l + i\Delta\sigma)^2 \\
&\quad -\frac{\rho_{m,i}^{n+1/2}(\sigma_l + i\Delta\sigma)q_{m,i}^{n+1/2}}{2\Delta\sigma}\frac{(s_{f,i+1}^{n+1} - s_{f,i-1}^{n+1} + s_{f,i+1}^{n} - s_{f,i-1}^{n})}{(s_{f,i}^{n+1} + s_{f,i}^{n})} \\
&\quad +\frac{1}{2}\left[\frac{q_{m,i}^{n+1/2}}{2\Delta\sigma}\frac{(s_{f,i+1}^{n+1} - s_{f,i-1}^{n+1} + s_{f,i+1}^{n} - s_{f,i-1}^{n})}{(s_{f,i}^{n+1} + s_{f,i}^{n})}\right]^2, \\
a_{2,m,i}^{n+1/2} &= \rho_{m,i}^{n+1/2}(\sigma_l + i\Delta\sigma) - \frac{q_{m,i}^{n+1/2}}{2\Delta\sigma}\frac{(s_{f,i+1}^{n+1} - s_{f,i-1}^{n+1} + s_{f,i+1}^{n} - s_{f,i-1}^{n})}{(s_{f,i}^{n+1} + s_{f,i}^{n})}, \\
a_{3,m,i}^{n+1/2} &= \frac{1}{2}, \\
a_{4,m,i}^{n+1/2} &= \frac{2}{(s_{f,i}^{n+1} + s_{f,i}^{n})}\frac{s_{f,i}^{n+1} - s_{f,i}^{n}}{\Delta\tau} + r - D_0 \\
&\quad -\left[\rho_{m,i}^{n+1/2}(\sigma_l + i\Delta\sigma)q_{m,i}^{n+1/2} + p_{m,i}^{n+1/2} - \lambda_{m,i}^{n+1/2}q_{m,i}^{n+1/2}\right] \\
&\qquad \times\frac{s_{f,i+1}^{n+1} - s_{f,i-1}^{n+1} + s_{f,i+1}^{n} - s_{f,i-1}^{n}}{2\Delta\sigma(s_{f,i}^{n+1} + s_{f,i}^{n})}
\end{aligned}
$$

$$+\left[\frac{q_{m,i}^{n+1/2}}{2\Delta\sigma}\frac{(s_{f,i+1}^{n+1}-s_{f,i-1}^{n+1}+s_{f,i+1}^{n}-s_{f,i-1}^{n})}{(s_{f,i}^{n+1}+s_{f,i}^{n})}\right]^2$$

$$-\left[\frac{q_{m,i}^{n+1/2}}{\Delta\sigma}\right]^2\frac{s_{f,i+1}^{n+1}-2s_{f,i}^{n+1}+s_{f,i-1}^{n+1}+s_{f,i+1}^{n}-2s_{f,i}^{n}+s_{f,i-1}^{n}}{2\left(s_{f,i}^{n+1}+s_{f,i}^{n}\right)},$$

$$\begin{aligned}
a_{5,m,i}^{n+1/2} &= p_{m,i}^{n+1/2}-\lambda_{m,i}^{n+1/2}q_{m,i}^{n+1/2},\\
a_{6,m,i}^{n+1/2} &= -r,\\
a_{7,m,i}^{n+1/2} &= -f(m\Delta\xi(s_{f,i}^{n+1}+s_{f,i}^{n})/2,\ \sigma_l+i\Delta\sigma,\ T-(n+1/2)\Delta\tau).
\end{aligned}$$

At the boundaries $\sigma=\sigma_l$ and $\sigma=\sigma_u$, due to $q=0$, the partial differential equation in the problem (9.11) becomes

$$\frac{\partial u}{\partial\tau}=a_1\xi^2\frac{\partial^2 u}{\partial\xi^2}+a_4\xi\frac{\partial u}{\partial\xi}+a_5\frac{\partial u}{\partial\sigma}+a_6u+a_7.$$

Just like the European case, this equation possesses hyperbolic properties in the σ-direction. Hence, we can approximate the partial differential equation in problem (9.11) at the boundary $\sigma=\sigma_l$ by

$$\begin{aligned}
&\frac{u_{m,0}^{n+1}-u_{m,0}^{n}}{\Delta\tau}\\
&=\frac{a_1m^2}{2}(u_{m+1,0}^{n+1}-2u_{m,0}^{n+1}+u_{m-1,0}^{n+1}+u_{m+1,0}^{n}-2u_{m,0}^{n}+u_{m-1,0}^{n})\\
&\quad+\frac{a_4m}{4}(u_{m+1,0}^{n+1}-u_{m-1,0}^{n+1}+u_{m+1,0}^{n}-u_{m-1,0}^{n})\\
&\quad+\frac{a_5}{4\Delta\sigma}(-u_{m,2}^{n+1}+4u_{m,1}^{n+1}-3u_{m,0}^{n+1}-u_{m,2}^{n}+4u_{m,1}^{n}-3u_{m,0}^{n})\\
&\quad+\frac{a_6}{2}(u_{m,0}^{n+1}+u_{m,0}^{n})+a_7,\qquad m=0,1,\cdots,M-1
\end{aligned}\tag{9.42}$$

and at the boundary $\sigma=\sigma_u$ by

$$\begin{aligned}
&\frac{u_{m,I}^{n+1}-u_{m,I}^{n}}{\Delta\tau}\\
&=\frac{a_1m^2}{2}(u_{m+1,I}^{n+1}-2u_{m,I}^{n+1}+u_{m-1,I}^{n+1}+u_{m+1,I}^{n}-2u_{m,I}^{n}+u_{m-1,I}^{n})\\
&\quad+\frac{a_4m}{4}(u_{m+1,I}^{n+1}-u_{m-1,I}^{n+1}+u_{m+1,I}^{n}-u_{m-1,I}^{n})\\
&\quad+\frac{a_5}{4\Delta\sigma}(3u_{m,I}^{n+1}-4u_{m,I-1}^{n+1}+u_{m,I-2}^{n+1}\\
&\qquad+3u_{m,I}^{n}-4u_{m,I-1}^{n}+u_{m,I-2}^{n})\\
&\quad+\frac{a_6}{2}(u_{m,I}^{n+1}+u_{m,I}^{n})+a_7,\qquad m=0,1,\cdots,M-1.
\end{aligned}\tag{9.43}$$

Here, $\dfrac{\partial u}{\partial \sigma}$ is discretized by a one-sided second-order scheme in order for all the node points involved to be in the computational domain. Here, a_1 and a_4–a_7 are also evaluated at ξ_m, σ_i and $\tau^{n+1/2}$. The formulae for a_1 and a_4–a_7 are almost the same as those given above, except that the partial derivative $\dfrac{\partial s_f}{\partial \sigma}$ is discretized in the same way as $\dfrac{\partial u}{\partial \sigma}$. That is, $\dfrac{\partial s_f}{\partial \sigma}$ in the difference scheme (9.42) is approximated by

$$\frac{-s_{f,2}^{n+1} + 4s_{f,1}^{n+1} - 3s_{f,0}^{n+1} - s_{f,2}^{n} + 4s_{f,1}^{n} - 3s_{f,0}^{n}}{4\Delta\sigma}$$

and in scheme (9.43) by

$$\frac{3s_{f,I}^{n+1} - 4s_{f,I-1}^{n+1} + s_{f,I-2}^{n+1} + 3s_{f,I}^{n} - 4s_{f,I-1}^{n} + s_{f,I-2}^{n}}{4\Delta\sigma}.$$

From the expression for a_4, we see that because $q = 0$ at $\sigma = \sigma_l$ and $\sigma = \sigma_u$, we do not need one-sided second-order finite-difference schemes for $\dfrac{\partial^2 s_f}{\partial \sigma^2}$.

Noticing that the coefficients of $\dfrac{\partial^2 u}{\partial \xi^2}, \dfrac{\partial u}{\partial \xi}$ in the problem (9.11) at $\xi = 0$ are zero, $u_{-1,i}^{n}$ does not appear in Eqs. (9.41)–(9.43) with $m = 0$.

At $\xi = 1$, there are two boundary conditions in the problem (9.11). One can be written as

$$u_{M,i}^{n+1} = g(s_{f,i}^{n+1}, \tau^{n+1}), \quad i = 0, 1, 2, \cdots, I, \tag{9.44}$$

where

$$g(s_f, \tau) = s_f \left[1 - \mathrm{e}^{-D_0\tau} N(d_1)\right] - E\left[1 - \mathrm{e}^{-r\tau} N(d_2)\right].$$

The other can be approximated by

$$3u_{M,i}^{n+1} - 4u_{M-1,i}^{n+1} + u_{M-2,i}^{n+1} = 2\Delta\xi h(s_{f,i}^{n+1}, \tau^{n+1}), \quad i = 0, 1, \cdots, I,$$

or

$$\begin{aligned} 3g(s_{f,i}^{n+1}, \tau^{n+1}) - 4u_{M-1,i}^{n+1} + u_{M-2,i}^{n+1} = 2\Delta\xi h(s_{f,i}^{n+1}, \tau^{n+1}), \\ i = 0, 1, \cdots, I, \end{aligned} \tag{9.45}$$

where

$$h(s_f, \tau) = s_f \left[1 - \mathrm{e}^{-D_0\tau} N(d_1)\right].$$

At $\tau = 0$, from

$$u(\xi, \sigma, 0) = 0 \quad \text{and} \quad s_f(\sigma, 0) = \max(E, rE/D_0),$$

we have

$$\begin{cases} u_{m,i}^0 = 0, \qquad m = 0,1,\cdots,M, \quad i = 0,1,\cdots,I, \\ s_{f,i}^0 = \max(E, rE/D_0), \qquad\qquad i = 0,1,\cdots,I. \end{cases} \tag{9.46}$$

For a fixed n, the system (9.41)–(9.45) consists of $(M+2)(I+1)$ equations. If $u_{m,i}^n$, $m = 0,1,\cdots,M$, $i = 0,1,\cdots,I$ and $s_{f,i}^n$, $i = 0,1,\cdots,I$ are known, then in the system there are $(M+2)(I+1)$ unknowns, namely, $u_{m,i}^{n+1}$, $m = 0,1,\cdots,M$, $i = 0,1,\cdots,I$ and $s_{f,i}^{n+1}$, $i = 0,1,\cdots,I$, and these unknowns can be obtained from solving the system. Because the set of initial conditions (9.46) gives $u_{m,i}^0$ for all m and i and $s_{f,i}^0$ for all i, we can have $u_{m,i}^{n+1}$, $i = 0,1,\cdots,I$, $m = 0,1,\cdots,M$ and $s_{f,i}^{n+1}$, $i = 0,1,\cdots,I$ for $n = 0,1,\cdots,N-1$ successively.

There are many ways to solve the above nonlinear system. If $s_{f,i}^{n+1}$, $i = 0,1,\cdots,I$ are given, then the system consisting of Eqs. (9.41)–(9.44) is a linear system for $u_{m,i}^{n+1}$, $m = 0,1,\cdots,M$ and $i = 0,1,\cdots,I$. One way to solve the system is as follows. Guessing $s_{f,i}^{n+1}$, $i = 0,1,\cdots,I$ and solving the system (9.41)–(9.44), we get all the approximate $u_{m,i}^{n+1}$, $m = 0,1,\cdots,M$, and $i = 0,1,\cdots,I$. Then check if Eq. (9.45) holds. If it does, we get our solution; if not, we determine new $s_{f,i}^{n+1}$, $i = 0,1,\cdots,I$, in the following way.

For each i, Eq. (9.45) is a nonlinear equation for $s_{f,i}^{n+1}$ when $u_{M-1,i}^{n+1}$ and $u_{M-2,i}^{n+1}$ are given. We take the root of the nonlinear equation as the new value of $s_{f,i}^{n+1}$. This root can be determined by Newton's method based on Eq. (9.45):

$$s_{f,i}^{(k+1)} = s_{f,i}^{(k)} - \frac{\theta(s_{f,i}^{(k)})}{\theta'(s_{f,i}^{(k)})},$$

where $s_{f,i}^{(k)}$ is the k-th iterative value of $s_{f,i}^{n+1}$ and

$$\begin{aligned} \theta(s_{f,i},\tau^{n+1}) &= 3g(s_{f,i},\tau^{n+1}) - 4u_{M-1,i}^{n+1} + u_{M-2,i}^{n+1} - 2\Delta\xi h(s_{f,i},\tau^{n+1}), \\ \theta'(s_{f,i},\tau^{n+1}) &= 3\frac{\partial g}{\partial s_{f,i}}(s_{f,i},\tau^{n+1}) - 2\Delta\xi\frac{\partial h}{\partial s_{f,i}}(s_{f,i},\tau^{n+1}) \\ &= (3-2\Delta\xi)\left[1 - \mathrm{e}^{-D_0\tau^{n+1}}N(d_1)\right] + \frac{2\Delta\xi}{\sigma\sqrt{2\pi\tau^{n+1}}}\mathrm{e}^{-D_0\tau^{n+1}-d_1^2/2} \end{aligned}$$

with $d_1 = \dfrac{\ln(s_{f,i}/E) + (r - D_0 + \sigma^2/2)\tau^{n+1}}{\sigma\sqrt{\tau^{n+1}}}$. As the starting value $s_{f,i}^{(0)}$ of this procedure, we take the value of $s_{f,i}^{n+1}$ used when the system (9.41)–(9.44) is solved previously.

9.2.6 Numerical Results of Two-Factor Options

Now let us show some results obtained by the numerical method above. We use the following two stochastic volatility models:

$$d\sigma = a(b-\sigma)dt + c\frac{1-\left(1-2\dfrac{\sigma-\sigma_l}{\sigma_u-\sigma_l}\right)^2}{1-0.975\left(1-2\dfrac{\sigma-\sigma_l}{\sigma_u-\sigma_l}\right)^2}\sigma dX_2, \quad \sigma_l \le \sigma \le \sigma_u \quad (9.47)$$

and

$$d\sigma = a(b-\sigma)dt + c\left[\frac{(\sigma-\sigma_l)(\sigma_u-\sigma)}{(\sigma_u-\sigma_l)^2}\right]^{1/2}\sigma dX_2, \quad \sigma_l \le \sigma \le \sigma_u, \quad (9.48)$$

where a, b, and c are positive parameters. The models (9.47) and (9.48) are referred to as Model I and Model II, respectively, in what follows. Both models are in the form (8.98). There is only a little difference between them. In Model I, $\dfrac{\partial q(\sigma,t)}{\partial\sigma}$ is bounded on $[\sigma_l,\sigma_u]$, and the reversion conditions are reduced to the conditions (8.101) and (8.102). Clearly, $q(\sigma_l) = q(\sigma_u) = 0$, so the equality conditions in the conditions (8.101) and (8.102) hold. In this case, the inequality conditions are $a(b-\sigma_l) \ge 0$ and $a(b-\sigma_u) \le 0$, which can be combined into

$$\sigma_l \le b \le \sigma_u. \quad (9.49)$$

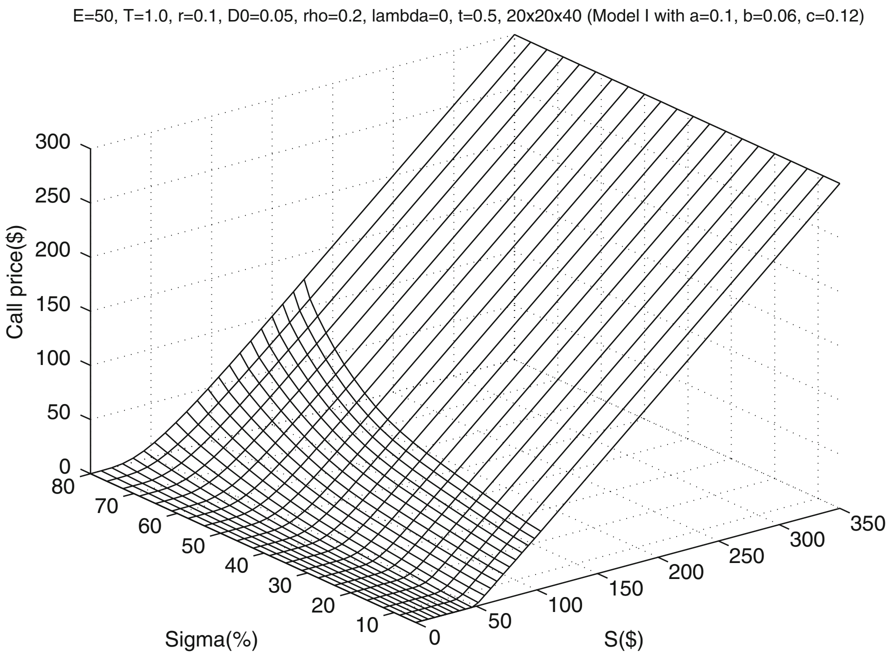

Fig. 9.14. The American call price ($t = 0.5$, $T = 1.0$, $\rho = 0.2$, and $\lambda = 0$)

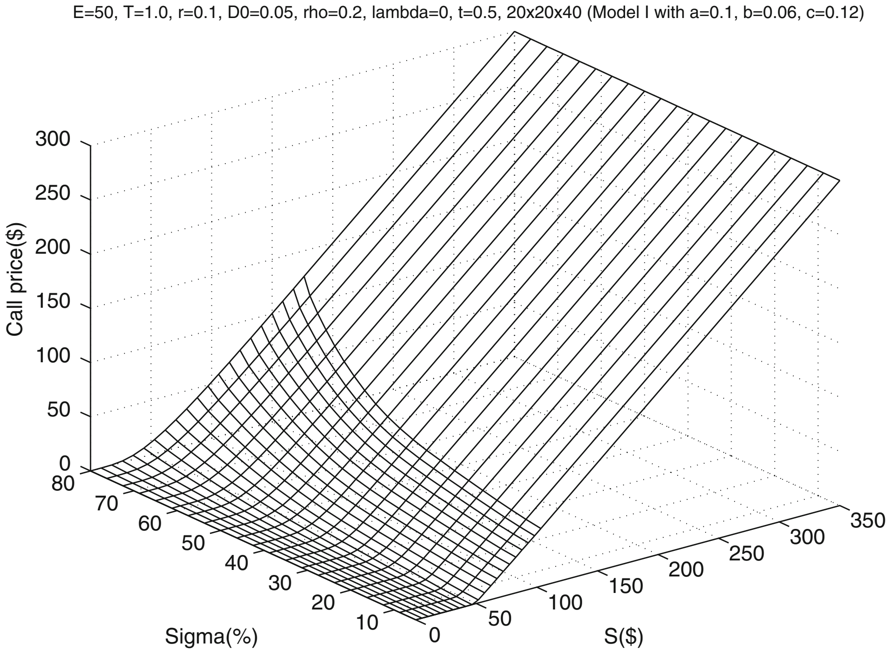

Fig. 9.15. The American call price ($t = 0.5$, $T = 1.0$, $\rho = 0.2$, and $\lambda = 0$)

Consequently, when the relation (9.49) holds, Model I satisfies the reversion conditions. For Model II, the equality conditions of the conditions (8.99) and (8.100) always hold, and the inequality conditions become

$$\begin{cases} p(\sigma_l, t) - q(\sigma_l, t)\dfrac{\partial q(\sigma_l, t)}{\partial \sigma} = a(b - \sigma_l) - 0.5c^2\sigma_l^2/(\sigma_u - \sigma_l) \geq 0, \\ p(\sigma_u, t) - q(\sigma_u, t)\dfrac{\partial q(\sigma_u, t)}{\partial \sigma} = a(b - \sigma_u) + 0.5c^2\sigma_u^2/(\sigma_u - \sigma_l) \leq 0. \end{cases} \tag{9.50}$$

Therefore, in order for Model II to satisfy the reversion conditions (8.99) and (8.100), we require that the set of conditions (9.50) holds. In the following examples, we take $\sigma_l = 0.05$ and $\sigma_u = 0.8$.

Example 1. Here, we calculate a 1-year American call option with Model I. We choose $a = 0.1$, $b = 0.06$, $c = 0.12$, $\rho = 0.2$, and $\lambda = 0$. We take 20 grid points in the ξ-direction and 20 grid points in the σ-direction and 40 time steps in the τ-direction, namely, the mesh is $20 \times 20 \times 40$. The other parameters are $E = 50$, $r = 0.1$, and $D_0 = 0.05$.

Figures 9.14 and 9.15 show the values of the American call option with $T = 1$ at time $t = 0.5$ and $t = 0$. Because those parameters a, b, c, ρ, and λ do not depend on time, Fig. 9.14 also shows the value of an option with $T = 0.5$ at time $t = 0$. Here, the strips represent the plane $C = S - E$, the solution for

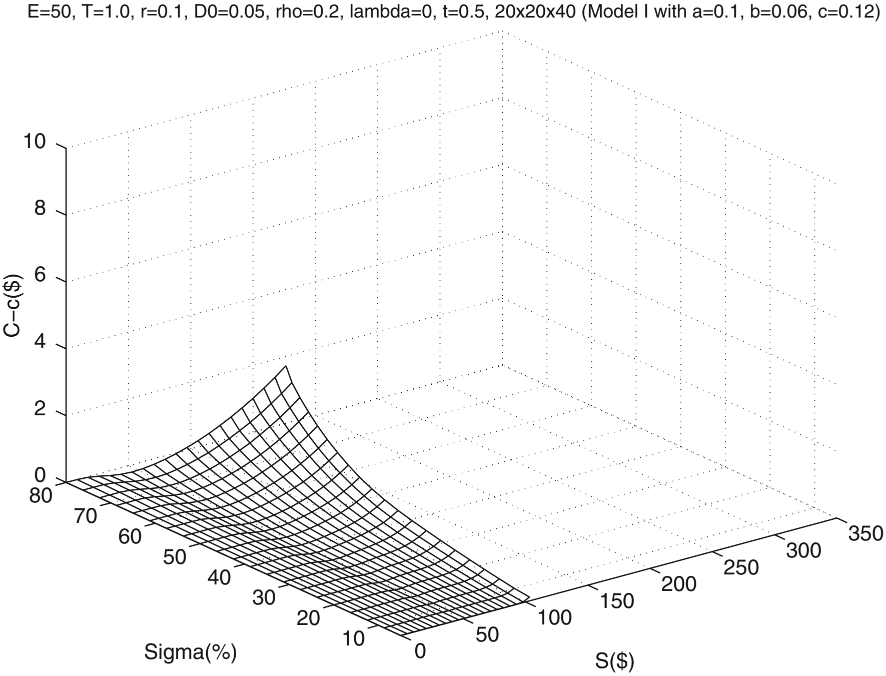

Fig. 9.16. The difference function $\overline{C} = C - c$
($t = 0.5$, $T = 1.0$, $\rho = 0.2$, and $\lambda = 0$)

Table 9.9. Numerical solutions with extrapolation

($E = 50$, $T = 3.0$, $r = 0.1$, $D_0 = 0.05$,
$a = 0.1$, $b = 0.06$, $c = 0.12$, $\rho = 0.2$, and $\lambda = 0$)

σ	S	u_1	u_2	u_3	U_1^*	U_2^*
0.125	45	3.93255	3.93246	3.93148	3.93115	3.93097
0.125	50	7.01873	7.02191	7.02241	7.02257	7.02253
0.125	55	10.7219	10.7224	10.7225	10.7225	10.7226
0.200	45	5.58000	5.57170	5.57137	5.57126	5.57160
0.200	50	8.49808	8.49254	8.49207	8.49191	8.49209
0.200	55	11.8781	11.8756	11.8742	11.8737	11.8736
0.350	45	8.93697	8.92810	8.92576	8.92498	8.92496
0.350	50	11.8615	11.8610	11.8607	11.8606	11.8605
0.350	55	15.1021	15.0953	15.0925	15.0916	15.0914

$S > S_f(\sigma, t)$, and the meshed surface shows the solution for $S \le S_f(\sigma, t)$. In Figs. 9.16 and 9.17, the difference $\overline{C}$ is shown for $t = 0.5$ and 0, respectively. There, only the solution of the free-boundary problem has been shown. As we know, the derivative of C with respect to S at $t = T$ is discontinuous at $S = E$. Comparing Figs. 9.14 and 9.15, we see that the value of C becomes smoother as t decreases. However, we know from Fig. 9.15 that even at $t = 0$,

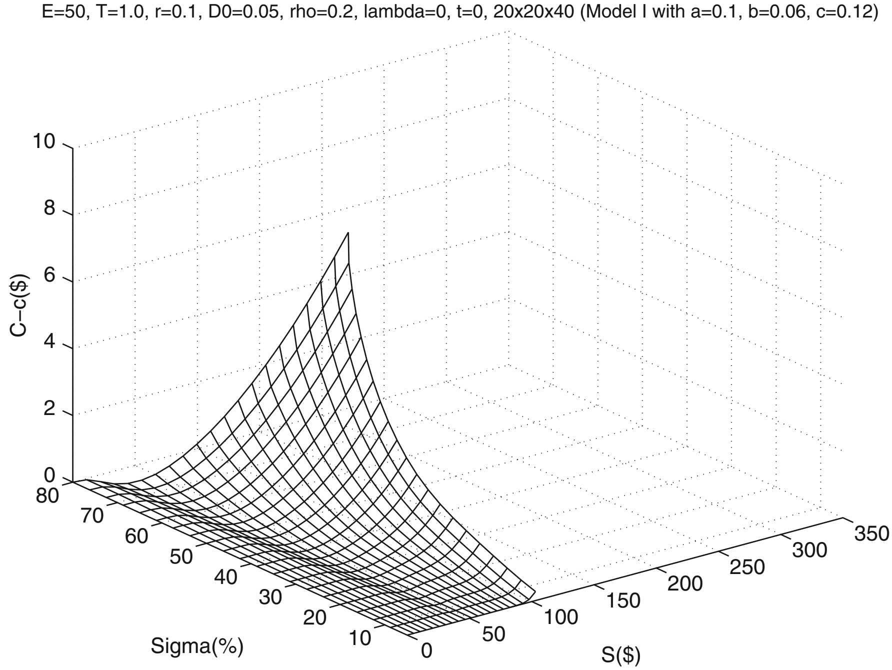

Fig. 9.17. The difference function $\overline{C} = C - c$ ($t = 0.0$, $T = 1.0$, $\rho = 0.2$, and $\lambda = 0$)

for smaller σ, C still changes rapidly with respect to S near $S = E$. The difference $\overline{C}$ at $t = T$ is identically equal to zero and remains smooth as t decreases, which can be seen from Figs. 9.16 and 9.17. Because $\overline{C}$ is much smoother than C, we can have much better numerical results if we use the partial differential equation for $\overline{C}$ instead of C when we do the computation.

Example 2. In this example, we calculate a 3-year American call option for Model II. All the other parameters, except ρ and λ, are the same as those in Example 1. In this case

$$a(b - \sigma_l) - 0.5c^2\sigma_l^2/(\sigma_u - \sigma_l) = 0.000976 > 0$$

and

$$a(b - \sigma_u) + 0.5c^2\sigma_u^2/(\sigma_u - \sigma_l) = -0.067856 < 0.$$

Thus, the set of conditions (9.50) holds, and no boundary condition is needed in order to determine the price.

First, we take $\rho = 0.2$ and $\lambda = 0$ and do the computation on different meshes. In Table 9.9, u_1 is the numerical solution using a mesh of $10 \times 10 \times 20$, u_2 is the value using a mesh of $20 \times 20 \times 40$, and u_3 is the value using a mesh

Table 9.10. Comparison of the results with various λ

$(E = 50,\ T = 3.0,\ r = 0.1,\ D_0 = 0.05,\ a = 0.1,\ b = 0.06,\ c = 0.12,\ \text{and}\ \rho = 0.2)$

σ	S	$\lambda = -1.0$	$\lambda = -0.5$	$\lambda = 0$	$\lambda = 0.5$	$\lambda = 1.0$	One-factor
0.125	45	4.1031	4.0170	3.9310	3.8449	3.7589	4.1789
0.125	50	7.1681	7.0953	7.0225	6.9497	6.8770	7.2432
0.125	55	10.825	10.774	10.723	10.672	10.620	10.885
0.200	45	5.9394	5.7555	5.5716	5.3877	5.2037	6.1134
0.200	50	8.8482	8.6701	8.4921	8.3141	8.1360	9.0216
0.200	55	12.193	12.033	11.874	11.714	11.554	12.361
0.350	45	9.6937	9.3085	8.9250	8.5417	8.1576	9.9913
0.350	50	12.647	12.254	11.861	11.467	11.072	12.964
0.350	55	15.876	15.485	15.091	14.701	14.309	16.209

Table 9.11. Comparison of the results with various ρ

$(E = 50,\ T = 3.0,\ r = 0.1,\ D_0 = 0.05,\ a = 0.1,\ b = 0.06,\ c = 0.12,\ \text{and}\ \lambda = 0)$

σ	S	$\rho = 0.4$	$\rho = 0.2$	$\rho = 0$	$\rho = -0.2$	$\rho = -0.4$	One-factor
0.125	45	3.9290	3.9310	3.9329	3.9349	3.9369	4.1789
0.125	50	7.0127	7.0225	7.0323	7.0422	7.0520	7.2432
0.125	55	10.711	10.723	10.735	10.747	10.759	10.885
0.200	45	5.5735	5.5716	5.5697	5.5678	5.5659	6.1134
0.200	50	8.4815	8.4921	8.5027	8.5134	8.5240	9.0216
0.200	55	11.854	11.874	11.893	11.913	11.933	12.361
0.350	45	8.9460	8.9250	8.9038	8.8825	8.8616	9.9913
0.350	50	11.866	11.861	11.855	11.849	11.844	12.964
0.350	55	15.083	15.091	15.100	15.109	15.118	16.209

of $40 \times 40 \times 80$. There, we also give results when the extrapolation technique is used. U_1^* is the extrapolation value obtained by

$$U_1^* = \frac{1}{3}(4u_3 - u_2)$$

and U_2^* is the extrapolation value generated by

$$U_2^* = \frac{1}{21}(32u_3 - 12u_2 + u_1).$$

From the table, we see that the errors of u_1 are on the second decimal place, those of u_2 and u_3 are on the third decimal place, and for the extrapolation values U_1^* and U_2^*, they are on the fourth decimal place. This shows that the extrapolation technique increases accuracy.

Then, we take $\rho = 0.2$ and try different λ to see how the results vary. The mesh used is $40 \times 40 \times 80$. In Table 9.10, we compare the values of the options with different parameters λ. The columns with $\lambda = -1,\ -0.5,\ 0,\ 0.5,\ 1.0$ at the top contain the values of the options when $\lambda = -1,\ -0.5,\ 0,\ 0.5,\ 1.0,$

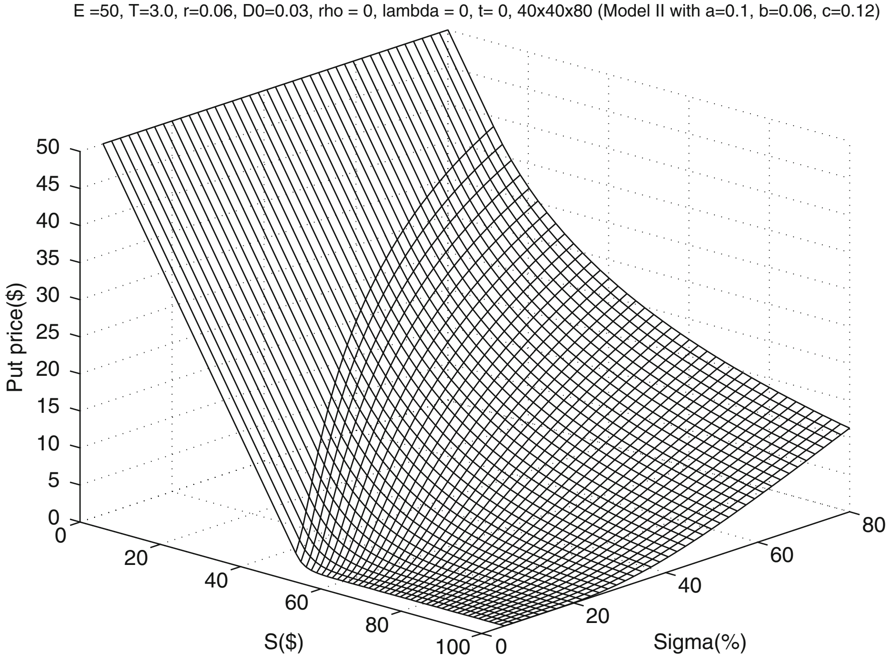

Fig. 9.18. The American put value at $t = 0.0$ ($E = 50$, $T = 3.0$, $\rho = 0$, and $\lambda = 0$)

respectively. For this case, the smaller the λ, the higher the call option price. The difference among the results for $\lambda \in [-1, 1]$ is about 10 %–20 %. This shows that we can calibrate the model to some extent even if we choose a constant λ. We also list the values of the one-factor model with a constant volatility. From Table 9.10, we see that the one-factor model overprices the American call options.

In Table 9.11, we compare the values of the options with a different correlation factor ρ and $\lambda = 0$, while the other parameters are kept unchanged. The notation is similar to Table 9.10. The results show that the option price varies a little when the correlation factor changes. Here, we again see that one factor model overprices the American call option.

An American two-factor put option problem can also be reduced to solving a free-boundary problem. However, the free-boundary problem is defined on an infinite domain. As we have pointed out, a vanilla two-factor put option can be converted into a vanilla two-factor call option with the same parameters except for r, D_0, ρ, and λ. Therefore, as long as we have a code for call options, we can also obtain the price of any put option.

Example 3. We want to have the price of a put option with $r = 0.06$, $D_0 = 0.03$, $\rho = 0$, and $\lambda = 0$ for Model II. The other parameters are the same as those in Example 2. We use a mesh $40 \times 40 \times 80$. In order to do this, we can

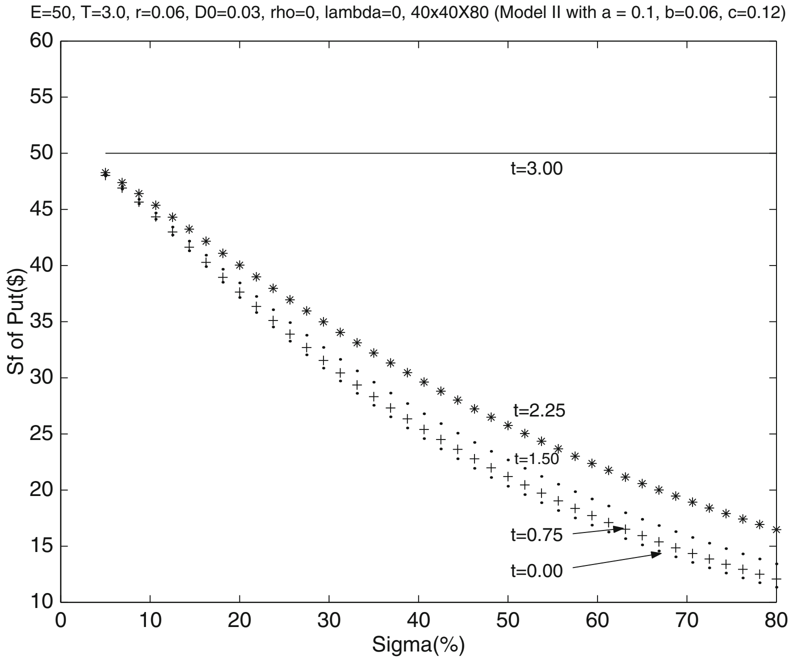

Fig. 9.19. The free boundaries of a put option for different times ($E = 50$, $T = 3.0$, $\rho = 0$, and $\lambda = 0$)

first calculate a call option with $r = 0.03$, $D_0 = 0.06$, $\rho = 0$, and $\lambda = 0$. Then, using the set of relations (9.14), we can have the price and the optimal exercise price of the put option. In Figs. 9.18 and 9.19, the price of the put option at $t = 0$ for $S \in [0, 100]$ and the optimal exercise price at $t = 0, 0.75, 1.5, 2.25,$ and 3.0 are shown.

For more results and details on two-factor options, see [56, 93, 94]. Finally, we point out that the models given here are assumptions. In order to use such a computation in practice, the models should be found from the market data.

9.3 Pseudo-Spectral Methods

9.3.1 The Description of the Pseudo-Spectral Methods for Two-Factor Convertible Bonds

A free-boundary problem can also be solved by pseudo-spectral methods. If the solution is smooth, then the pseudo-spectral method as a high-order difference method may be more efficient. Thus, when we compute $\overline{C} = C - c$, the pseudo-spectral method might be another good tool. Also, a parabolic operator always smoothes the solution. Thus, even if the initial value is not smooth, the solution becomes smooth after a while. The life span of a convertible bond is quite long. If the time to the expiry is more than 2 years, the solution is already quite smooth. Thus, if expiry is not soon, then the solution of a convertible bond is quite smooth and for that time, a pseudo-spectral method might be a good choice. In the last section, we already took the American call option as an example to give the details of the implicit difference methods. In this section, we will describe the details of the pseudo-spectral method for the two-factor convertible bond problem.

In Sect. 9.1.2, a two-factor convertible bond problem with $D_0 > 0$ was reduced to the problem (9.15) or the problem (9.21). Suppose that we do not take the difference and want to solve $V(s, r, t)$ directly, that is, we solve the problem (9.15). Using the transformation (9.19) and defining $u(\xi, \bar{r}, \tau)$ and $s_f(\bar{r}, \tau)$ by the set of formulae (9.20), we can rewrite the problem (9.15) as the following problem on $u(\xi, \bar{r}, \tau)$ and $s_f(\bar{r}, \tau)$:

$$\begin{cases} \dfrac{\partial u}{\partial \tau} = \mathbf{L}_{\xi,\bar{\mathbf{r}}} u + a_7, & 0 \le \xi \le 1, \quad 0 \le \bar{r} \le 1, \quad 0 \le \tau \le T, \\ u(\xi, \bar{r}, 0) = \max\left(1, \xi s_f(\bar{r}, 0)\right), & 0 \le \xi \le 1, \quad 0 \le \bar{r} \le 1, \\ u(1, \bar{r}, \tau) = s_f(\bar{r}, \tau), & 0 \le \bar{r} \le 1, \quad 0 \le \tau \le T, \\ \dfrac{\partial u}{\partial \xi}(1, \bar{r}, \tau) = s_f(\bar{r}, \tau), & 0 \le \bar{r} \le 1, \quad 0 \le \tau \le T, \\ s_f(\bar{r}, 0) = \max\left(1, k/D_0\right), & 0 \le \bar{r} \le 1, \end{cases} \tag{9.51}$$

where $\mathbf{L}_{\xi,\bar{\mathbf{r}}}$ is the same as given in the problem (9.21):

$$\begin{aligned} \mathbf{L}_{\xi,\bar{\mathbf{r}}} = {} & a_1 \xi^2 \frac{\partial^2}{\partial \xi^2} + a_2 \xi w \frac{\partial^2}{\partial \xi \partial \bar{r}} + a_3 w^2 \frac{\partial^2}{\partial \bar{r}^2} + \left(a_4 + \frac{1}{s_f}\frac{\partial s_f}{\partial \tau}\right) \xi \frac{\partial}{\partial \xi} \\ & + a_5 \frac{\partial}{\partial \bar{r}} + a_6 \end{aligned}$$

and

$$a_7 = k.$$

Therefore, finding the value of a convertible bond is now reduced to solving a problem on a rectangular domain. Suppose that we take $M+1$ nodes in the ξ-direction: $\xi_0, \xi_1, \cdots, \xi_M$, $L+1$ nodes in the $\bar{r}$-direction: $\bar{r}_0, \bar{r}_1, \cdots, \bar{r}_L$, and $N+1$ nodes in the τ-direction: $\tau^0, \tau^1, \cdots, \tau^N$, where $\xi_0 = 0$, $\xi_M = 1$, $\bar{r}_0 = 0$, $\bar{r}_L = 1$, $\tau^0 = 0$, and $\tau^N = T$. Furthermore, we assume the nodes in the τ-direction to be equidistant with $\Delta\tau = T/N$. Let $u^n_{m,l}$ denote $u(\xi_m, \bar{r}_l, \tau^n)$ and $s^n_{f,l}$ stand for $s_f(\bar{r}_l, \tau^n)$. For a fixed n, we need to determine $u^n_{m,l}, m = 0, 1, \cdots, M$ and $l = 0, 1, \cdots, L$, and $s^n_{f,l}, l = 0, 1, \cdots, L$. In what follows, let $\left\{u^n_{m,l}\right\}$ and $\left\{s^n_{f,l}\right\}$ denote the sets

$$\left\{u^n_{m,l}, m = 0, 1, \cdots, M \text{ and } l = 0, 1, \cdots, L\right\}$$

and

$$\left\{s_{f,l}^n, l = 0, 1, \cdots, L\right\}$$

respectively. For $n = 0$, $\left\{u_{m,l}^n\right\}$ and $\left\{s_{f,l}^n\right\}$ are determined by the initial conditions of the problem, which gives

$$\begin{aligned}
u_{m,l}^0 &= \max\left(1, \xi_m s_{f,l}^0\right), \quad m = 0, 1, \cdots, M, \quad l = 0, 1, \cdots, L, \\
s_{f,l}^0 &= \max\left(1, k/D_0\right), \quad l = 0, 1, \cdots, L.
\end{aligned}$$

What we need to do is to find $\left\{u_{m,l}^n\right\}$ and $\left\{s_{f,l}^n\right\}$ for $n = 1, 2, \cdots, N$.

According to Sect. 6.1.2, we may assume the solution on the domain $[0,1] \times [0,1]$ to be polynomials in each direction. Under such an assumption, for a fixed n, $\dfrac{\partial u}{\partial \xi}$, $\dfrac{\partial u}{\partial \bar{r}}$, $\dfrac{\partial^2 u}{\partial \xi^2}$, $\dfrac{\partial^2 u}{\partial \xi \partial \bar{r}}$, and $\dfrac{\partial^2 u}{\partial \bar{r}^2}$ at any point are linear combinations of $u_{m,l}^n$, $m = 0, 1, \cdots, M$ and $l = 0, 1. \cdots, L$, and $\dfrac{\partial s_f}{\partial \bar{r}}$ at any $\bar{r}$ is a linear combination of $s_{f,l}^n$, $l = 0, 1. \cdots, L$. Therefore, the partial differential equation and the boundary conditions in the problem (9.51) can be discretized into algebraic equations, and solving the equations yields $\left\{u_{m,l}^n\right\}$ and $\left\{s_{f,l}^n\right\}$.

Now we describe the details. Suppose that for a fixed pair of n and l, $u_{m,l}^n$, $m = 0, 1, \cdots, M$ are known. According to these values, we can establish a polynomial in ξ with degree M. From this polynomial, we can determine $\dfrac{\partial u}{\partial \xi}$, $\dfrac{\partial^2 u}{\partial \xi^2}$ at any point for $\bar{r} = \bar{r}_l$ and $\tau = \tau^n$. If ξ_m is defined as follows:

$$\xi_m = \frac{1}{2}\left(1 - \cos\frac{m\pi}{M}\right), \quad m = 0, 1, \cdots, M,$$

then

$$\begin{aligned}
\frac{\partial u}{\partial \xi}(\xi_m, \bar{r}_l, \tau^n) &= \sum_{i=0}^{M} D_{\xi,m,i}\, u(\xi_i, \bar{r}_l, \tau^n), \\
\frac{\partial^2 u}{\partial \xi^2}(\xi_m, \bar{r}_l, \tau^n) &= \sum_{i=0}^{M} D_{\xi\xi,m,i}\, u(\xi_i, \bar{r}_l, \tau^n)
\end{aligned}$$

and according to Sect. 6.1.2,

$$D_{\xi,m,i} = \begin{cases}
\dfrac{c_m(-1)^{m+i}}{c_i(\xi_m - \xi_i)}, & m \neq i, \\
-\dfrac{2M^2+1}{3}, & m = i = 0, \\
\dfrac{1 - 2\xi_i}{4\xi_i(1 - \xi_i)}, & m = i = 1, 2, \cdots, M-1, \\
\dfrac{2M^2+1}{3}, & m = i = M,
\end{cases}$$

where $c_0 = c_M = 2$ and $c_i = 1$, $i = 1, 2, \cdots, M-1$, and

$$D_{\xi\xi,m,i} = \sum_{k=0}^{M} D_{\xi,m,k} D_{\xi,k,i}.$$

For brevity, we define

$$\mathbf{D}_{\xi,\mathbf{m}} u_{m,l}^n = \sum_{i=0}^{M} D_{\xi,m,i} u(\xi_i, \bar{r}_l, \tau^n),$$
$$\mathbf{D}_{\xi\xi,\mathbf{m}} u_{m,l}^n = \sum_{i=0}^{M} D_{\xi\xi,m,i} u(\xi_i, \bar{r}_l, \tau^n)$$

and write the two approximations in difference operator form:

$$\frac{\partial u}{\partial \xi}(\xi_m, \bar{r}_l, \tau^n) = \mathbf{D}_{\xi,\mathbf{m}} u_{m,l}^n,$$
$$\frac{\partial^2 u}{\partial \xi^2}(\xi_m, \bar{r}_l, \tau^n) = \mathbf{D}_{\xi\xi,\mathbf{m}} u_{m,l}^n$$

Similarly, if $\bar{r}_l$ is defined by

$$\bar{r}_l = \frac{1}{2}\left(1 - \cos\frac{l\pi}{L}\right), \quad l = 0, 1, \cdots, L,$$

then

$$\frac{\partial u}{\partial \bar{r}}(\xi_m, \bar{r}_l, \tau^n) = \mathbf{D}_{\bar{\mathbf{r}},\mathbf{l}} u_{m,l}^n,$$
$$\frac{\partial^2 u}{\partial \bar{r}^2}(\xi_m, \bar{r}_l, \tau^n) = \mathbf{D}_{\bar{\mathbf{r}}\bar{\mathbf{r}},\mathbf{l}} u_{m,l}^n,$$

and

$$\frac{\partial^2 u}{\partial \xi \partial \bar{r}}(\xi_m, \bar{r}_l, \tau^n) = \mathbf{D}_{\bar{\mathbf{r}},\mathbf{l}} \mathbf{D}_{\xi,\mathbf{m}} u_{m,l}^n.$$

These difference operators are defined by

$$\mathbf{D}_{\bar{\mathbf{r}},\mathbf{l}} u_{m,l}^n = \sum_{j=0}^{L} D_{\bar{r},l,j} u(\xi_m, \bar{r}_j, \tau^n),$$
$$\mathbf{D}_{\bar{\mathbf{r}}\bar{\mathbf{r}},\mathbf{l}} u_{m,l}^n = \sum_{j=0}^{L} D_{\bar{r}\bar{r},l,j} u(\xi_m, \bar{r}_j, \tau^n),$$
$$\mathbf{D}_{\bar{\mathbf{r}},\mathbf{l}} \mathbf{D}_{\xi,\mathbf{m}} u_{m,l}^n = \sum_{j=0}^{L} D_{\bar{r},l,j} \sum_{i=0}^{M} D_{\xi,m,i} u(\xi_i, \bar{r}_j, \tau^n),$$

where

$$D_{\bar{r},l,j} = \begin{cases} \dfrac{c_l(-1)^{l+j}}{c_j(\bar{r}_l - \bar{r}_j)}, & l \neq j, \\ -\dfrac{2L^2+1}{3}, & l = j = 0, \\ \dfrac{1-2\bar{r}_j}{4\bar{r}_j(1-\bar{r}_j)}, & l = j = 1, 2, \cdots, L-1, \\ \dfrac{2L^2+1}{3}, & l = j = L, \end{cases}$$

with $c_0 = c_L = 2$ and $c_j = 1$, $j = 1, 2, \cdots, L-1$, and

$$D_{\bar{r}\bar{r},l,j} = \sum_{k=0}^{L} D_{\bar{r},l,k} D_{\bar{r},k,j.}$$

In the τ-direction, we can approximate $\dfrac{\partial u}{\partial \tau}$ and $\dfrac{\partial s_f}{\partial \tau}$ by central differences:

$$\begin{aligned} \frac{\partial u}{\partial \tau}(\xi_m, \bar{r}_l, \tau^{n+1/2}) &= \frac{u(\xi_m, \bar{r}_l, \tau^{n+1}) - u(\xi_m, \bar{r}_l, \tau^n)}{\Delta\tau}, \\ \frac{\partial s_f}{\partial \tau}(\bar{r}_l, \tau^{n+1/2}) &= \frac{s_f(\bar{r}_l, \tau^{n+1}) - s_f(\bar{r}_l, \tau^n)}{\Delta\tau}. \end{aligned}$$

Therefore, the first equation in the problem (9.51) can be approximated by

$$\begin{aligned} &\frac{u_{m,l}^{n+1} - u_{m,l}^n}{\Delta\tau} \\ &= \frac{1}{2}\mathbf{L}_{\mathbf{m,l}}^{\mathbf{n+1/2}} \left(u_{m,l}^{n+1} + u_{m,l}^n\right) \\ &\quad + \left(\frac{1}{s_{f,l}^{n+1} + s_{f,l}^n} \frac{s_{f,l}^{n+1} - s_{f,l}^n}{\Delta\tau}\right) \xi_m \mathbf{D}_{\xi,\mathbf{m}} \left(u_{m,l}^{n+1} + u_{m,l}^n\right) + a_{7,m,l}^{n+1/2}, \\ &\quad m = 0, 1, \cdots, M-1, \quad l = 0, 1, \cdots, L. \end{aligned} \tag{9.52}$$

Here, the operator $\mathbf{L}_{\mathbf{m,l}}^{\mathbf{n+1/2}}$ and the scalar $a_{7,m,l}^{n+1/2}$ are defined by

$$\mathbf{L}_{\mathbf{m,l}}^{\mathbf{n+1/2}} = \frac{1}{2}\left(\mathbf{L}_{\mathbf{m,l}}^{\mathbf{n+1}} + \mathbf{L}_{\mathbf{m,l}}^{\mathbf{n}}\right),$$

and

$$a_{7,m,l}^{n+1/2} = \frac{1}{2}\left(a_{7,m,l}^{n+1} + a_{7,m,l}^n\right),$$

where

$$\begin{aligned} \mathbf{L}_{\mathbf{m,l}}^{\mathbf{n}} &= a_{1,m,l}^n \xi_m^2 \mathbf{D}_{\xi\xi,\mathbf{m}} + a_{2,m,l}^n \xi_m w_{m,l}^n \mathbf{D}_{\bar{\mathbf{r}},\mathbf{l}} \mathbf{D}_{\xi,\mathbf{m}} + a_{3,m,l}^n \left(w_{m,l}^n\right)^2 \mathbf{D}_{\bar{\mathbf{r}}\bar{\mathbf{r}},\mathbf{l}} \\ &\quad + a_{4,m,l}^n \xi_m \mathbf{D}_{\xi,\mathbf{m}} + a_{5,m,l}^n \mathbf{D}_{\bar{\mathbf{r}},\mathbf{l}} + a_{6,m,l}^n, \\ a_{i,m,l}^n &= a_i(\xi_m, \bar{r}_l, \tau^n), \quad i = 1, 2, \cdots, 7, \end{aligned}$$

$$w^n_{m,l} = w(\xi_m, \bar{r}_l, \tau^n),$$

and the derivatives $\dfrac{\partial s_f}{\partial \bar{r}}, \dfrac{\partial^2 s_f}{\partial \bar{r}^2}$ appearing in a_1, a_2, and a_4 are approximated by

$$\frac{\partial s_f}{\partial \bar{r}}(\bar{r}_l, \tau^n) = \sum_{j=0}^{L} D_{\bar{r},l,j} s_f(\bar{r}_j, \tau^n),$$
$$\frac{\partial^2 s_f}{\partial \bar{r}^2}(\bar{r}_l, \tau^n) = \sum_{j=0}^{L} D_{\bar{r}\bar{r},l,j} s_f(\bar{r}_j, \tau^n).$$

The boundary conditions, the third and fourth relations in the problem (9.51), can be discretized as follows:

$$u^{n+1}_{M,l} = s^{n+1}_{f,l}, \qquad l = 0, 1, \cdots, L, \tag{9.53}$$

$$\mathbf{D}_{\xi,\mathbf{M}} u^{n+1}_{M,l} = s^{n+1}_{f,l}, \quad l = 0, 1, \cdots, L. \tag{9.54}$$

The system (9.52)–(9.54) has a truncation error of $O(\Delta\tau^2)$ in the τ-direction and is an M-th order scheme in the ξ-direction and an L-th order scheme in the $\bar{r}$-direction.

In the system (9.52)–(9.54), there are $(M+2)(L+1)$ equations. When $\left\{u^n_{m,l}\right\}$ and $\left\{s^n_{f,l}\right\}$ are given, the unknowns are $u^{n+1}_{m,l}$, $m = 0, 1, \cdots, M$, $l = 0, 1, \cdots, L$, $s^{n+1}_{f,l}$, $l = 0, 1, \cdots, L$, the total of which is also $(M+2)(L+1)$. Therefore, it is a closed system. Unfortunately, it is a nonlinear system, and we have to use iteration. Let $u^{(k)}_{m,l}, s^{(k)}_{f,l}$ represent the k-th iteration value of $u^{n+1}_{m,l}, s^{n+1}_{f,l}$, and we rewrite Eq. (9.52) in the form

$$\begin{aligned}
&u^{(k)}_{m,l} - \frac{\Delta\tau}{2} \bar{\mathbf{L}}^{(\mathbf{k-1})}_{\mathbf{m,l}} u^{(k)}_{m,l} - \frac{s^{(k)}_{f,l}}{s^{(k-1)}_{f,l} + s^n_{f,l}} \xi_m \mathbf{D}_{\xi,\mathbf{m}} \left(u^{(k-1)}_{m,l} + u^n_{m,l}\right) \\
&= u^n_{m,l} + \frac{\Delta\tau}{2} \bar{\mathbf{L}}^{(\mathbf{k-1})}_{\mathbf{m,l}} u^n_{m,l} - \frac{s^n_{f,l}}{s^{(k-1)}_{f,l} + s^n_{f,l}} \xi_m \mathbf{D}_{\xi,\mathbf{m}} \left(u^{(k-1)}_{m,l} + u^n_{m,l}\right) + \Delta\tau a_{7,m,l}, \\
&\qquad m = 0, 1, \cdots, M-1, \quad l = 0, 1, \cdots, L,
\end{aligned} \tag{9.55}$$

where

$$\bar{\mathbf{L}}^{(\mathbf{k-1})}_{\mathbf{m,l}} = \frac{1}{2}\left(\mathbf{L}^{(\mathbf{k-1})}_{\mathbf{m,l}} + \mathbf{L}^{\mathbf{n}}_{\mathbf{m,l}}\right).$$

Equations (9.53) and (9.54) can be written as

$$u^{(k)}_{M,l} = s^{(k)}_{f,l}, \qquad l = 0, 1, \cdots, L, \tag{9.56}$$

$$\mathbf{D}_{\xi,\mathbf{M}} u_{M,l}^{(k)} = s_{f,l}^{(k)}, \qquad l = 0, 1, \cdots, L. \tag{9.57}$$

The system (9.55)–(9.57) is a linear one for $u_{m,l}^{(k)}$, $m = 0, 1, \cdots, M$, $l = 0, 1, \cdots, L$ and $s_{f,l}^{(k)}$, $l = 0, 1, \cdots, L$. It can be solved by a direct or iteration method. We can let $u_{m,l}^{(0)} = u_{m,l}^{n}$, $m = 0, 1, \cdots, M$, $l = 0, 1, \cdots, L$ and $s_{f,l}^{(0)} = s_{f,l}^{n}$, $l = 0, 1, \cdots, L$. When $\left\{u_{m,l}^{(k-1)}\right\}$ and $\left\{s_{f,l}^{(k-1)}\right\}$ are known, we can find $\left\{u_{m,l}^{(k)}\right\}$ and $\left\{s_{f,l}^{(k)}\right\}$ by solving the system (9.55)–(9.57). When all $u_{m,l}^{(k)} - u_{m,l}^{(k-1)}$ and $s_{f,l}^{(k)} - s_{f,l}^{(k-1)}$ become very small, we can stop the iteration. Just like the case of one-dimensional finite-difference methods, we can stop at $k = 2$, and the result should be second-order accurate in the τ-direction. This is because $\left\{u_{m,l}^{(1)}\right\}$ and $\left\{s_{f,l}^{(1)}\right\}$ can be understood as a result of a first-order scheme in τ. The results $\left\{u_{m,l}^{(2)}\right\}$ and $\left\{s_{f,l}^{(2)}\right\}$ actually are the results of a scheme in which the improved Euler method is used in the τ-direction. Therefore, if $\left\{u_{m,l}^{n}\right\}$ and $\left\{s_{f,l}^{n}\right\}$ are given, we can obtain $\left\{u_{m,l}^{n+1}\right\}$ and $\left\{s_{f,l}^{n+1}\right\}$ by solving the system (9.55)–(9.57). Because $\left\{u_{m,l}^{0}\right\}$ and $\left\{s_{f,l}^{0}\right\}$ are given by the initial conditions, we can repeat the procedure described above for $n = 0, 1, \cdots, N-1$, and finally get $\left\{u_{m,l}^{N}\right\}$ and $\left\{s_{f,l}^{N}\right\}$.

As long as we find $\left\{u_{m,l}^{N}\right\}$ and $\left\{s_{f,l}^{N}\right\}$, for any S, r we can have the price of the convertible bond at $t = 0$ in the following way. If

$$S > Zs_f\left(\frac{r - r_l}{r_u - r_l}, T\right) \Big/ n,$$

then

$$V = \max(Z, nS);$$

while

$$S < Zs_f\left(\frac{r - r_l}{r_u - r_l}, T\right) \Big/ n,$$

then

$$V(S, r, 0) = Zu\left(\frac{nS}{Zs_f\left(\dfrac{r - r_l}{r_u - r_l}, T\right)}, \frac{r - r_l}{r_u - r_l}, T\right).$$

Usually, $\dfrac{r - r_l}{r_u - r_l} \neq \bar{r}_l$ for any l and $\dfrac{nS}{Zs_f\left(\dfrac{r - r_l}{r_u - r_l}, T\right)} \neq \xi_m$ for any m. In order to find $V(S, r, 0)$, we therefore need to use interpolation.

When $t \approx T$ and $S \approx \max\left(\dfrac{Z}{n}, \dfrac{KZ}{D_0 n}\right)$, the solution in the S-direction is not smooth. In order to overcome this problem, we need to solve the problem

(9.21) instead of the problem (9.51). The method for the problem (9.21) is almost the same as the method for the problem (9.51). The only difference is the boundary conditions and a_7. In this case, $a_{7,m,l}$ in the system (9.55) should be replaced by

$$\frac{1}{2}\left(a_{7,m,l}^{(k-1)} + a_{7,m,l}^{n}\right)$$

because a_7 involves the location of the free boundary. Here, $a_{7,m,l}^{(k-1)}$ is the $(k-1)$-th iteration value of $a_{7,m,l}^{n+1}$. If we still want to solve the problem (9.51), then at $t \approx T$, using the finite-difference methods or using the pseudo-spectral methods in the r-direction and using the finite-difference methods in the S-direction might be better. Readers can find the details about how to solve the two-factor convertible bond problems using the implicit finite-difference method in [95] and using the mixture of the pseudo-spectral methods and the finite-difference methods in [77]. In what follows, for brevity, we will refer to the mixture of the pseudo-spectral method and the finite-difference method as the pseudo-spectral method because in the entire computation, the main method is the pseudo-spectral method. It is clear that this problem can also be solved as a linear complementarity problem using an explicit or an implicit finite-difference scheme.

9.3.2 Numerical Results of Two-Factor Convertible Bonds

Here, we show some numerical results of a two-factor convertible bond by the pseudo-spectral method and compare the results by the pseudo-spectral method with the results obtained by the finite-difference method, by the projected explicit and projected implicit finite-difference methods.

The interest rate model we adopted for the example is based on the model used by Brennan and Schwartz (see [12]) and Druskin et al. (see [27]) even though in practice in order to get the interest rate model, we should solve an inverse problem by using the data on the market. Their model is

$$dr = u(r,t)dt + w(r,t)dX_2, \quad 0 \le r,$$

where

$$\begin{cases} u(r,t) = -0.13r + 0.008 + \lambda(r,t)w(r,t), \\ w(r,t) = \sqrt{0.26r}. \end{cases}$$

We made the following modifications. We assume

$$0 \le r \le 0.3$$

and instead of $0.26r$, use

$$0.26r\phi^2(r),$$

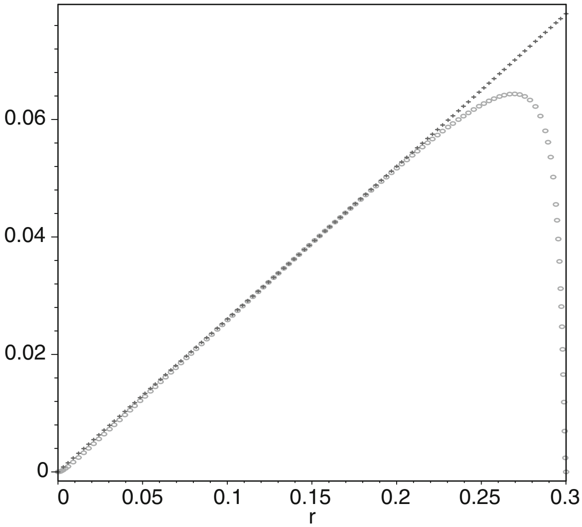

Fig. 9.20. The functions $0.26r$ and $0.26r\phi^2(r)$

where

$$\phi(r) = \frac{1-(1-2r/0.3)^2}{1-0.975\,(1-2r/0.3)^2}.$$

Thus, our model for the example here is

$$dr = u(r,t)dt + w(r,t)dX_2, \quad 0 \le r \le 0.3,$$

where

$$\begin{cases} u(r,t) = -0.13r + 0.008 + \lambda(r,t)w(r,t), \\ w(r,t) = \sqrt{0.26r}\phi(r). \end{cases}$$

The functions $0.26r$ and $0.26r\phi^2(r)$ are shown in Fig. 9.20, and we can see that for $r \in [0, 0.2]$, the difference is very small. Because

$$\phi(0) = \phi(0.3) = 0$$

and $d\phi(r)/dr$ is bounded on $[0, 0.3]$, we have

$$w(0,t)\frac{\partial w(0,t)}{\partial r} = w(0.3,t)\frac{\partial w(0.3,t)}{\partial r} = 0.$$

Therefore, the reversion conditions can be written as:

$$\begin{cases} u(0,t) \ge 0, \\ w(0,t) = 0 \end{cases}$$

and

$$\begin{cases} u(0.3, t) \le 0, \\ w(0.3, t) = 0. \end{cases}$$

Because of

$$u(0, t) = 0.008 > 0$$

and

$$u(0.3, t) = -0.13 \times 0.3 + 0.008 = -0.031 < 0,$$

we do not need any boundary conditions at $r = 0$ and $r = 0.3$.

We still assume the volatility and the dividend yield of the underlying stock to be

$$\sigma(S, t) = 0.20$$

and

$$D_0 = 0.05,$$

respectively, and the correlation of the two random variables dX_1 and dX_2 to be

$$\rho(S, r, t)dt = -0.01dt.$$

Let us consider a standard convertible bond with $k = 0.06$ and $T = 30$. First, we give the result obtained by the pseudo-spectral methods. Concretely, for $\tau \in [0, 2]$, in the r-direction the pseudo-spectral method described in Sect. 8.4 is adopted, and in the S-direction the implicit finite-difference method discussed in Sect. 9.2.1 is used, and we take $M = 60$, $L = 10$; for $\tau \in [2, 30]$ in both directions, the pseudo-spectral method is used and $M = L = 10$. In the τ-direction, a nonuniform time step is used and $N = 50$. In Fig. 9.21, the values of the two-factor convertible bond at $t = 1$ month, 6 months, 1 year, 5 years, 10 years, and 30 years are plotted. In Fig. 9.22, the location curves of the free boundary at various times are given.

Besides the method mentioned in this section, the implicit finite-difference method similar to the method in Sect. 9.2.5, the projected explicit finite-difference method and the projected implicit finite-difference method have been used to compute the same problem on various meshes. For the implicit finite-difference method, the value of the convertible bond at $r = 0.05, S = 1, t = 30$ years on a very fine mesh is $1.3116835\cdots$ [1] and these eight digits are unchanged as the mesh size further decreases. Therefore, this value is accurate to at least seven digits. After we have a highly accurate result, we can obtain the first few digits of the error of the results on different meshes. For each computation, we also record the CPU time. Thus, for each error, we can have the corresponding CPU time. Figure 9.23 is a $\log_{10}$(error) versus $\log_{10}$(CPU time in second) graph, and each point in the figure represents a performance of the method. Because the ranges of errors and CPU times are very large, we adopt $\log_{10}$(Error) and $\log_{10}$(CPU time in second) as variables. There, a "$\times$"

[1] When this figure was obtained, the function $\phi(r)$ used was $\left[4r(0.3 - r)/0.3^2\right]^{1/8}$.

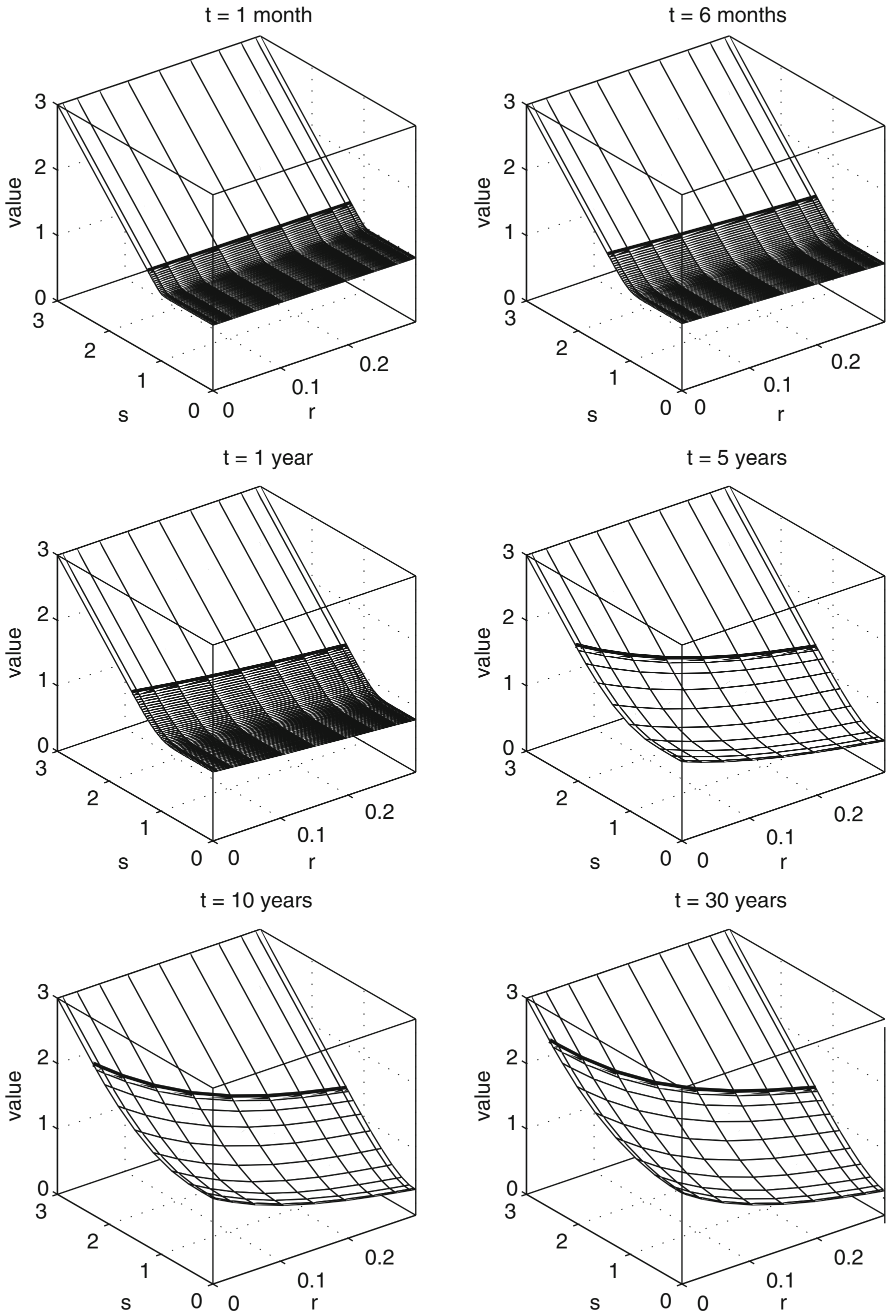

Fig. 9.21. Prices of a two-factor convertible bond at six different times

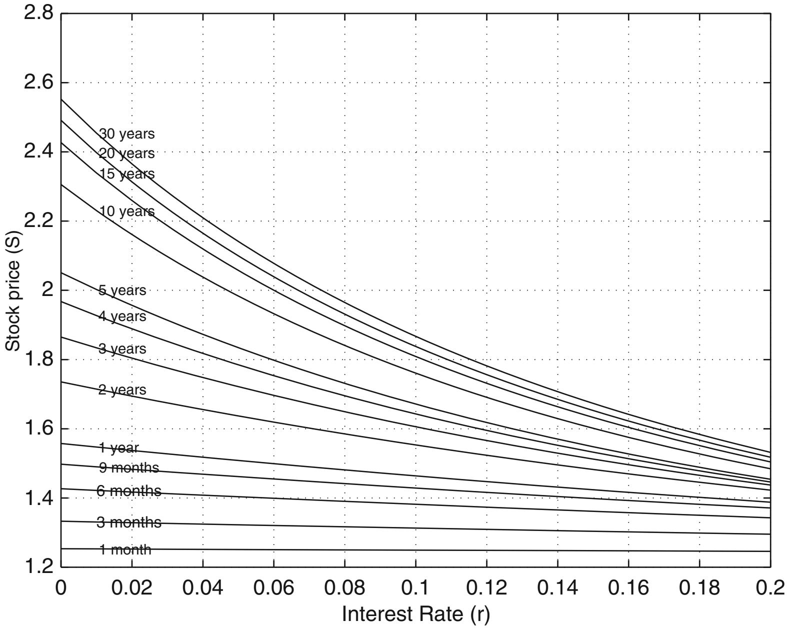

Fig. 9.22. Locations of a free boundary at various times

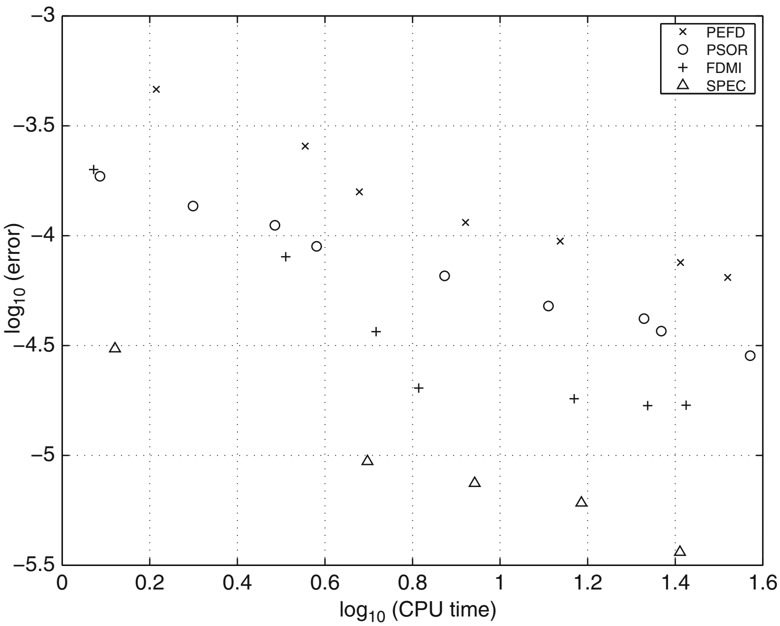

Fig. 9.23. $\log_{10}$(error) versus $\log_{10}$(CPU time in second)

represents the performance of the projected explicit finite-difference method, which is referred to as PEFD in the figure. A "∘" indicates the performance of the projected implicit finite-difference method. The successive over relaxation method is used to get the solution. Therefore, this method is referred to as PSOR in the figure. A "+" stands for the performance of the implicit finite-difference method. In order to get the solution of the nonlinear algebraic equations, the alternating-direction iteration method is used (see [77]). In the figure, it is referred to as FDMI. In the figure, a "Δ" represents the performance of the pseudo-spectral method, which is referred to as SPEC there. Clearly, the lower the point, the better the performance. From Fig. 9.23, we see that the pseudo-spectral method has the best performance for this example.

Problems

Table 9.12. Problems and sections

Problems	Sections	Problems	Sections	Problems	Sections
1–3	9.1	4–10(a, b)	9.2	10(c)–12	9.3

1. Consider the following free-boundary problem that is related to American lookback strike put options:

$$\begin{cases} \dfrac{\partial W}{\partial t}+\dfrac{1}{2}\sigma^2\eta^2\dfrac{\partial^2 W}{\partial \eta^2}+(D_0-r)\,\eta\dfrac{\partial W}{\partial \eta}-D_0W=0, & \\ \qquad\qquad\qquad\qquad\qquad 1\le\eta\le\eta_f(t), \quad 0\le t\le T, & \\ W(\eta,T)=\max(\eta-\beta,0), & 1\le\eta\le\eta_f(T), \\ \dfrac{\partial W}{\partial \eta}(1,t)=0, & 0\le t\le T, \\ W(\eta_f,t)=\eta_f-\beta, & 0\le t\le T, \\ \dfrac{\partial W}{\partial \eta}(\eta_f,t)=1, & 0\le t\le T, \\ \eta_f(T)=\beta\max(1,D_0/r). & \end{cases}$$

By using the closed-form solution of the problem

$$\begin{cases} \dfrac{\partial W_1}{\partial t}+\dfrac{1}{2}\sigma^2\eta^2\dfrac{\partial^2 W_1}{\partial \eta^2}+(D_0-r)\eta\dfrac{\partial W_1}{\partial \eta}-D_0W_1=0, & \eta\ge 1, \quad 0\le t\le T, \\ W_1(\eta,T)=\max(\eta-\beta,0), & \eta\ge 1, \\ \dfrac{\partial W_1}{\partial \eta}(1,t)=0, & 0\le t\le T, \end{cases}$$

convert this problem into a problem whose solution has a continuous derivative everywhere. Here we also require that the problem is defined on a rectangular domain: $[0,1] \times [0,T]$, has an initial condition, and the free boundary is the right boundary. (Assume $1 < \beta$).

2. Consider the following free-boundary problem that is related to American average strike call options:

$$\begin{cases} \dfrac{\partial W}{\partial t} + \dfrac{1}{2}\sigma^2\eta^2\dfrac{\partial^2 W}{\partial \eta^2} + \left[(D_0 - r)\eta + \dfrac{1-\eta}{t}\right]\dfrac{\partial W}{\partial \eta} - D_0 W = 0, & \\ & \eta_f(t) \le \eta, \quad t \le T, \\ W(\eta, T) = \max(1-\eta, 0), & \eta_f(T) \le \eta, \\ W(\eta_f(t), t) = 1 - \eta_f(t), & t \le T, \\ \dfrac{\partial W}{\partial \eta}(\eta_f(t), t) = -1, & t \le T, \\ \eta_f(T) = \min\left(1, \dfrac{1 + D_0 T}{1 + rT}\right). & \end{cases}$$

Convert this problem into a problem with a singularity weaker than the singularity here for $t > 0$. Also require that the new problem is defined on a rectangular domain, has an initial condition and the right boundary corresponds to the free boundary.

3. *Let $C(S, \sigma, t;\ a, b, c, d)$ and $P(S, \sigma, t;\ a, b, c, d)$ denote the prices of American two-factor call and put options and $S_{cf}(\sigma, t;\ a, b, c, d)$ and $S_{pf}(\sigma, t;\ a, b, c, d)$ be their optimal exercise prices. Here, a, b, c, and d are parameters (or parameter functions) for the risk-free interest rate r, dividend yield rate D_0, correlation coefficient ρ, and market price of volatility risk λ, respectively. Show that between American two-factor put and call options there is the following put–call symmetry relation:

$$\begin{cases} P(S, \sigma, t; a, b, c, d) = \dfrac{S}{E} C\left(\dfrac{E^2}{S}, \sigma, t; b, a, -c, d - c\sigma\right), \\ C(S, \sigma, t; a, b, c, d) = \dfrac{S}{E} P\left(\dfrac{E^2}{S}, \sigma, t; b, a, -c, d - c\sigma\right), \\ S_{pf}(\sigma, t; a, b, c, d) \times S_{cf}(\sigma, t; b, a, -c, d - c\sigma) = E^2. \end{cases}$$

4. Consider the following free-boundary problem:

$$\begin{cases} \dfrac{\partial V}{\partial \tau} = \dfrac{1}{2}\sigma^2\xi^2(1-\xi)^2\dfrac{\partial^2 V}{\partial \xi^2} + (r-D_0)\xi(1-\xi)\dfrac{\partial V}{\partial \xi} \\ \qquad -[r(1-\xi)+D_0\xi]V, & 0 \le \xi < \xi_f(\tau), \quad 0 \le \tau, \\ V(\xi,0) = \max(2\xi-1,0), & 0 \le \xi < \xi_f(0), \\ V\left(\xi_f(\tau),\tau\right) = 2\xi_f(\tau)-1, & 0 \le \tau, \\ \dfrac{\partial V}{\partial \xi}\left(\xi_f(\tau),\tau\right) = 2, & 0 \le \tau, \\ \xi_f(0) = \max\left(\dfrac{1}{2}, \dfrac{r}{r+D_0}\right). \end{cases}$$

It can be easily seen that the free-boundary problem for the American call options under the (S,t)-space can be rewritten as this form if let $\xi = S/(S+E)$ and $\tau = T-t$.

(a) Convert this problem into a problem whose solution has a continuous derivative everywhere. Here we also require that the problem is defined on a rectangular domain and with an initial condition.

(b) Design a second-order implicit method to solve the new problem. (Need to check whether or not the number of equations which can be used is equal to the number of unknowns.)

5. Consider the following the free-boundary problem:

$$\begin{cases} \dfrac{\partial u}{\partial \tau} = \dfrac{1}{2}\sigma^2\left[\xi + \dfrac{1}{\bar\eta_f - 1}\right]^2\dfrac{\partial^2 u}{\partial \xi^2} \\ \qquad + \left[(D_0-r)\left(\xi + \dfrac{1}{\bar\eta_f-1}\right) + \dfrac{\xi}{\bar\eta_f-1}\dfrac{d\bar\eta_f}{d\tau}\right]\dfrac{\partial u}{\partial \xi} - D_0 u, & 0 \le \xi \le 1, \\ & 0 \le \tau \le T, \\ u(\xi,0) = 0, & 0 \le \xi \le 1, \\ \dfrac{\partial u}{\partial \xi}(0,\tau) = 0, & 0 \le \tau \le T, \\ u(1,\tau) = \bar\eta_f(\tau) - \beta - W_1(\bar\eta_f(\tau), T-\tau), & 0 \le \tau \le T, \\ \dfrac{\partial u}{\partial \xi}(1,\tau) = (\bar\eta_f(\tau)-1)\left[1 - \dfrac{\partial W_1(\bar\eta_f(\tau),T-\tau)}{\partial \eta}\right], & 0 \le \tau \le T, \\ \bar\eta_f(0) = \beta\max\left(1, D_0/r\right), \end{cases}$$

where $W_1(\eta, T-\tau)$ is a given function. Design a second-order implicit method to solve this problem which is the new problem obtained in Problem 1. (Need to check whether or not the number of equations which can be used is equal to the number of unknowns.)

6. *Consider the nonlinear system consisting of the following equations

$$\begin{aligned}
&\frac{u_m^{n+1}-u_m^n}{\Delta\tau}\\
&=\frac{1}{2}\left[k_2 m^2\left(u_{m+1}^{n+1}-2u_m^{n+1}+u_{m-1}^{n+1}\right)+\frac{k_1 m}{2}\left(u_{m+1}^{n+1}-u_{m-1}^{n+1}\right)-k_0 u_m^{n+1}\right]\\
&\quad+\frac{1}{2}\left[k_2 m^2\left(u_{m+1}^{n}-2u_m^{n}+u_{m-1}^{n}\right)+\frac{k_1 m}{2}\left(u_{m+1}^{n}-u_{m-1}^{n}\right)-k_0 u_m^{n}\right]\\
&\quad+\frac{s_f^{n+1}-s_f^n}{\left(s_f^{n+1}+s_f^n\right)\Delta\tau}\left[\frac{m}{2}\left(u_{m+1}^{n+1}-u_{m-1}^{n+1}\right)+\frac{m}{2}\left(u_{m+1}^{n}-u_{m-1}^{n}\right)\right],\\
&\qquad m=0,1,2,\cdots,M-1,
\end{aligned}$$

and

$$u_M^{n+1}=g(s_f^{n+1},\tau^{n+1}),$$

$$\frac{3u_M^{n+1}-4u_{M-1}^{n+1}+u_{M-2}^{n+1}}{2\Delta\xi}=h\left(s_f^{n+1},\tau^{n+1}\right),$$

where u_m^n are known, τ^{n+1} is given, k_0, k_1, and k_2 are constants, and $g(s,\tau)$ and $h(s,\tau)$ are given functions. Discuss how to solve this system, provide at least two methods that you think are simple and effective, and give the details for one of the methods.

7. *Is the extrapolation technique always helpful and why?

8. Consider the scheme given in Problem 4. Why the extrapolation technique can still be used when a non-uniform mesh in τ with

$$\tau^n=n^2T/N^2,\quad n=0,1,\cdots,N,$$

is adopted? (Hint: Define $\tau_1=\sqrt{\tau T}$. Solving a problem with a variable step in τ is the same as solving a problem with a constant step in τ_1.)

9. *Design an exponential scheme to approximate

$$a(\xi)\frac{d^2U}{d\xi^2}+b(\xi)\frac{dU}{d\xi}+c(\xi)U,$$

where $a(\xi)>0$ and $c(\xi)<0$.

10. Assume σ to be a random variable satisfying

$$d\sigma=p(\sigma,t)dt+q(\sigma,t)dX,$$

where dX is a Wiener process. In this case, evaluating American call options can be reduced to solving the following free-boundary problem:

$$\begin{cases} \dfrac{\partial C}{\partial t} + \mathbf{L}_{\mathbf{S},\sigma} C = 0, & 0 \le S \le S_f(\sigma, t), \\ & \sigma_l \le \sigma \le \sigma_u, \quad 0 \le t \le T, \\ C(S, \sigma, T) = \max(S - E, 0), & 0 \le S \le S_f(\sigma, T), \\ & \sigma_l \le \sigma \le \sigma_u, \\ C\left(S_f(\sigma, t), \sigma, t\right) = S_f(\sigma, t) - E, & \sigma_l \le \sigma \le \sigma_u, \quad 0 \le t \le T, \\ \dfrac{\partial C\left(S_f(\sigma, t), \sigma, t\right)}{\partial S} = 1, & \sigma_l \le \sigma \le \sigma_u, \quad 0 \le t \le T, \\ S_f(\sigma, T) = \max(E, rE/D_0), & \sigma_l \le \sigma \le \sigma_u, \end{cases}$$

where

$$\begin{aligned} \mathbf{L}_{\mathbf{S},\sigma} = & \frac{1}{2}\sigma^2 S^2 \frac{\partial^2}{\partial S^2} + \rho\sigma S q \frac{\partial^2}{\partial S \partial \sigma} + \frac{1}{2} q^2 \frac{\partial^2}{\partial \sigma^2} \\ & + (r - D_0) S \frac{\partial}{\partial S} + (p - \lambda q) \frac{\partial}{\partial \sigma} - r. \end{aligned}$$

(a) *Convert this problem into a problem defined on a rectangular domain and whose solution has a singularity weaker than the singularity here.

(b) *Design a second-order implicit method to solve the new problem. (Here and also for part (c), do not require to discuss the solution of the nonlinear system.)

(c) Design a pseudo-spectral method to solve the new problem.

11. Consider the following free-boundary problem related to one-factor convertible bonds:

$$\begin{cases} \dfrac{\partial B_c}{\partial t} + \dfrac{1}{2}\sigma^2 S^2 \dfrac{\partial^2 B_c}{\partial S^2} + (r - D_0) S \dfrac{\partial B_c}{\partial S} - r B_c + kZ = 0, & \\ & 0 \le S \le S_f(t), \;\; 0 \le t \le T, \\ B_c(S, T) = \max(Z, nS), & 0 \le S \le S_f(T), \\ B_c\left(S_f(t), t\right) = n S_f(t), & 0 \le t \le T, \\ \dfrac{\partial B_c}{\partial S}\left(S_f(t), t\right) = n, & 0 \le t \le T, \\ S_f(T) = \max\left(\dfrac{Z}{n}, \dfrac{kZ}{D_0 n}\right). & \end{cases}$$

(a) Convert this problem into a problem whose solution has a continuous derivative everywhere, and which is defined on a rectangular domain and has an initial condition.

(b) Design a pseudo-spectral method to solve the new problem. (Do not require to discuss the solution of the nonlinear system.)

12. *Consider the nonlinear system consisting of the following equations:

$$\begin{aligned}
&\frac{u_{m,l}^{n+1} - u_{m,l}^{n}}{\Delta\tau} \\
&= \frac{1}{2}\mathbf{L}_{\mathbf{m,l}}^{\mathbf{n+1/2}}\left(u_{m,l}^{n+1} + u_{m,l}^{n}\right) \\
&\quad + \left(\frac{1}{s_{f,l}^{n+1} + s_{f,l}^{n}} \frac{s_{f,l}^{n+1} - s_{f,l}^{n}}{\Delta\tau}\right)\xi_m \mathbf{D}_{\xi,\mathbf{m}}\left(u_{m,l}^{n+1} + u_{m,l}^{n}\right) + a_{7,m,l}, \\
&\qquad m = 0, 1, \cdots, M-1, \quad l = 0, 1, \cdots, L, \\
&\qquad u_{M,l}^{n+1} = s_{f,l}^{n+1}, \qquad l = 0, 1, \cdots, L,
\end{aligned}$$

and

$$\mathbf{D}_{\xi,\mathbf{M}} u_{M,l}^{n+1} = s_{f,l}^{n+1}, \quad l = 0, 1, \cdots, L,$$

where

$$\mathbf{L}_{\mathbf{m,l}}^{\mathbf{n+1/2}} = \frac{1}{2}\left(\mathbf{L}_{\mathbf{m,l}}^{\mathbf{n+1}} + \mathbf{L}_{\mathbf{m,l}}^{\mathbf{n}}\right).$$

Here, $u_{m,l}^{n}, m = 0, 1, \cdots, M,\ l = 0, 1, \cdots, L$ and $s_{f,l}^{n}, l = 0, 1, \cdots, L$ are given and $u_{m,l}^{n+1}, m = 0, 1, \cdots, M,\ l = 0, 1, \cdots, L$ and $s_{f,l}^{n+1}, l = 0, 1, \cdots, L$ are unknown. In the system, $\mathbf{D}_{\xi,\mathbf{m}}$ and $\mathbf{L}_{\mathbf{m,l}}^{\mathbf{n}}$ are difference operators with variable coefficients. $\mathbf{L}_{\mathbf{m,l}}^{\mathbf{n+1}}$ is another difference operator whose coefficients depend on $s_{f,l}^{n+1}, l = 0, 1, \cdots, L$. Discuss how to solve this system and give an outline of a method that you think is simple and effective.

Projects

General Requirements

(A) Submit a code or codes in C or C^{++} that will work on a computer the instructor can get access to. At the beginning of the code, write down the name of the student and indicate on which computer it works and the procedure to make it work.

(B) Each code should use an input file to specify all the problem parameters and the computational parameters for each computation and an output file to store all the results. In an output file, the name of the student, all the problem parameters, and the computational parameters should be given, so that one can know what the results are and how they were obtained. The input file should be submitted with the code.

(C) If not specified, for each case two results are required. For the first result, a 50×10 mesh should be used. For the second result, the accuracy required is 0.001, and the mesh used should be as coarse as possible.

(D) Submit results in form of tables. When a result is given, always provide the problem parameters and the computational parameters.

1. **Implicit Scheme (9.22)–(9.24).** Suppose σ, r, D_0 are constant. Write a code performing implicit singularity-separating method for American calls and puts. In the code, a result of an American call option should be obtained by the implicit scheme (9.22)–(9.24), whereas a result of an American put option should be obtained through solving a corresponding call problem numerically and then using the symmetry relation.
   - For American call and put options, give the results for the case: $S = 100$, $E = 100$, $T = 1$, $r = 0.1$, $D_0 = 0.05$, $\sigma = 0.2$.
   - For American call and put options, give the results for the case: $S = 100$, $E = 100$, $T = 1$, $r = 0.05$, $D_0 = 0.1$, $\sigma = 0.2$.
   - For American call and put options, find the results with an accuracy of 0.00001 under the help of the extrapolation technique. The problem parameters are $S = 90, 100, 110$, $E = 100$, $T = 1.00$, $r = 0.1$, $D_0 = 0.05$, and $\sigma = 0.2$.
2. Using the binomial method (8.28) with the formulae (8.25)–(8.27) try to find the values of American call and put options with an accuracy of 0.00001. The problem parameters are $S = 90, 100, 110$, $E = 100$, $T = 1.00$, $r = 0.10$, $D_0 = 0.05$, and $\sigma = 0.2$.