

Adaptive and high-order methods for valuing American options

Christina C. Christara and Duy Minh Dang

Department of Computer Science, University of Toronto

Toronto, Ontario M5S 3G4, Canada

{ccc, dmdang}@cs.toronto.edu

April 29, 2010

Abstract

We develop space-time adaptive and high-order methods for valuing American options using a partial differential equation (PDE) approach. The linear complementarity problem arising due to the free boundary is handled by a penalty method. Both finite difference and finite element methods are considered for the space discretization of the PDE, while classical finite differences, such as Crank-Nicolson, are used for the time discretization. The high-order discretization in space is based on an optimal finite element collocation method, the main computational requirements of which are the solution of one tridiagonal linear system at each time step, while the resulting errors at the gridpoints and midpoints of the space partition are fourth-order. To control the space error, we use adaptive gridpoint distribution based on an error equidistribution principle. A time stepsize selector is used to further increase the efficiency of the methods. Numerical examples show that our methods converge fast and provide highly accurate options prices, Greeks, and early exercise boundaries.

Key words: adaptive mesh selection, error equidistribution, quadratic splines, collocation, finite differences, European option, American option, penalty method

1 Introduction

The pricing of an American option is a difficult task, mainly due to the early exercise feature of the option [34, 38]. Typically, at any time, there is a specific value of the asset price that divides the asset domain into the early exercise region, where the option should be exercised immediately, and the continuation region, where the option should be held. Hence, the early exercise feature leads to an additional constraint which stipulates that the value of an American option must be greater than or equal to its payoff. This constraint requires special treatment, a fact that makes an explicit closed form solution for an American option intractable for most cases. Consequently, numerical methods must be used.

Although several approaches, such as Monte Carlo simulations [16, 25], lattice (tree) methods [21, 22], or integral equations [2, 4, 27], can be used for pricing an American option, for problems in low dimensions, i.e. less than five dimensions, the partial differential equation (PDE) approach is very popular, due to its efficiency and global character. In addition, accurate hedging parameters, such as delta and gamma, which are essential for risk-managing financial derivatives, are generally much easier to compute via a PDE approach than via other methods. Using a PDE approach, the American option pricing problem can be formulated as a time-dependent linear complementarity problem (LCP) with the inequalities involving the Black-Scholes PDE and some additional constraints [37]. Recently, several approaches for handling the LCP have been developed. In particular, various penalty methods were discussed in [14, 29, 39]. In this paper, we adopt the penalty method of [14] to solve the LCP. According to this approach, a penalty term is introduced in the discretized equations in order to enforce the early exercise constraint. Although this method was originally built upon a finite volume (FV) discretization method for the space dimension, the idea of this method could be extended to other discretization techniques, such as finite difference (FD) and finite element.

The popularity of finite differences in option pricing is mainly due to their intuition and easiness to implement. Finite elements can also be used as an alternative. These discretization methods offer several advantages over finite

differences such as: (i) the solution is a piecewise polynomial approximation to the entire domain, while the method of finite differences supplies an approximate solution only to distinct points in the domain, thus interpolation may become necessary; (ii) there are several finite element methods, for instance, spline collocation, that supply hedging parameters, such as delta and gamma, as a by-product, and, in addition, allow other hedging parameters to be computed in a slightly easier manner than with finite differences. In particular, certain spline collocation¹ methods have been shown to be effective on uniform and non-uniform grids for the solution of boundary value problems [7, 8] and parabolic initial value problems [6].

Using spline collocation in its standard formulation gives only second-order, thus suboptimal, accuracy. In the context of parabolic PDEs, this suboptimal spline collocation method requires the solution of one tridiagonal linear system at each timestep. In general, high-order methods in space usually require larger discretization stencils, and hence the systems to be solved at each timestep are not tridiagonal. In [6], several optimal and efficient methods based on quadratic spline collocation (QSC) are developed for one-dimensional linear parabolic PDEs. These methods give fourth-order (optimal) convergence on the knots and midpoints with the main computational requirements of the methods being the solution of only one tridiagonal linear system at each timestep. Extensions of such efficient high-order spline collocation methods to option pricing, especially to pricing American options, have not been previously discussed in literature. This shortcoming motivated our work.

Adaptive methods aim at dynamically adjusting the location of the gridpoints in order to control the error in the approximate solution. Although adaptive techniques have been extensively developed for numerical solutions of parabolic PDEs (e.g. [7, 12, 13, 36]), they are not so frequent in the option pricing literature. Example of algorithms for adaptivity in space and time can be found in [1, 26, 30]. In [1], a space adaptive mesh refinement based on a posteriori estimates of the finite element discretization errors of the Black-Scholes equation computed using a Hilbert sum is proposed. The authors of [30] proposed a space-time adaptive FD technique for pricing multi-asset European options. The adaptivity in space is based on first solving the problem on a coarse grid with large timesteps for an estimation of the errors, and then resolving the problem with more gridpoints redistributed in such a way that the estimated local error is below a certain level. In [26], an error equation is derived for the global error in the solution, and the grid and timestep sizes are chosen such that a tolerance on the final global error is satisfied by the solution. A popular technique for mesh generation and adaptation is based on De Boor's equidistribution principle [11]. The underlying idea of this principle is to relocate the nodes to equidistribute the error in some chosen norm (or semi-norm) among the subintervals of the partition. Although adaptive techniques based on an equidistribution principle are widely used in the numerical solution of PDEs, to the best of our knowledge, these techniques have not been successfully extended to option pricing, in general, and American option pricing, in particular. This deficiency further motivated our work.

In this paper, we develop highly accurate and efficient numerical methods for pricing American options on a single asset. Although we focus primarily on the one-dimensional case, some of the results in this paper can be naturally extended to two or more dimensions. The high-order methods in the spatial dimension are built upon the efficient and high-order QSC methods of [6]. Second-order FD discretization for the spatial variable is also considered. Adaptive techniques based on the equidistribution principle of [11] are introduced into the space dimension. A time stepsize selector [14, 23] is used to further increase the performance of the methods. Numerical results show that our methods provide highly accurate options prices and Greeks, and capture well the moving behavior of the free boundary. In this paper, we do not include gridsize estimators and changes of the gridsize from timestep to timestep, such as those in [36]. We plan to incorporate the gridsize estimators presented in [7] into American option pricing problems in a near future.

¹Note that, in this paper, we use the term "spline" to refer to maximum smoothness piecewise polynomials.

The remainder of the paper is organized as follows. Section 2 presents a PDE formulation of the pricing problem for an American option. We restrict our attention to the American put options. In Section 3, we describe discretization methods with strong emphasis on the efficient and high-order QSC methods, and discuss the selection of an appropriate form for the discrete penalty term. A penalty iteration for the discretized American put option is discussed in Section 4. Section 5 introduces an adaptive mesh algorithm for American option pricing and a simple, but effective, time stepsize selector. Numerical results that demonstrate the efficiency and accuracy of the methods are presented in Section 6. Section 7 concludes the paper.

2 Formulation

The Black-Scholes model for American put options takes the form of a free boundary problem [34, 38]. The disadvantage of the free boundary formulation is that there is an explicit mention of the free boundary. To avoid this, we write the American put option valuation problem in an LCP form, and the optimal free boundary can then be determined a posteriori. More specifically, denoting by S the value at time t of the underlying asset, by T the expiry time of the option, and by $\tau = T - t$ the backward time variable, the early exercise constraint leads to the following LCP for the value $V(S, \tau)$ of an American put option [37]

$$\left\{ \begin{array}{l} \frac{\partial V}{\partial \tau} - \mathcal{L}V = 0 \\ V - V^* \geq 0 \end{array} \right\} \text{ and } \left\{ \begin{array}{l} \frac{\partial V}{\partial \tau} - \mathcal{L}V > 0 \\ V - V^* = 0 \end{array} \right\}, \quad (2.1)$$

subject to the payoff

$$V^*(S) = V(S, 0) = \max(E - S, 0), \quad (2.2)$$

and the boundary conditions

$$\begin{aligned} V(0, \tau) &= E, \\ V(S, \tau) &\sim 0 \quad \text{as } S \rightarrow S_\infty, \end{aligned} \quad (2.3)$$

where

$$\mathcal{L}V \equiv \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV. \quad (2.4)$$

Here, S_∞ is the right boundary of the semi-truncated spatial domain, E is the strike, r and σ are the positive constant risk-free interest rate and constant asset volatility, respectively.

Following [14], we replace the LCP (2.1) by a non-linear PDE obtained by adding a penalty term to the right side of the Black-Scholes equation. More specifically, with a penalty parameter ζ , $\zeta \rightarrow \infty$, we consider the non-linear PDE for an American put option

$$\frac{\partial V}{\partial \tau} - \mathcal{L}V = \zeta \max(V^* - V, 0), \quad S \in \Omega \equiv (0, S_\infty), \tau \in [0, T], \quad (2.5)$$

subject to the initial and boundary conditions (2.2) and (2.3). The penalty term $\zeta \max(V^* - V, 0)$ effectively ensures that the solution satisfies $V - V^* \geq -\epsilon$ for $0 < \epsilon \ll 1$. Essentially, in the region where $V \geq V^*$, the PDE (2.5) resembles the Black-Scholes equation. On the other hand, when $-\epsilon \leq V - V^* < 0$, the Black-Scholes inequality is satisfied, assuring that the early exercise rule is not violated.

3 Discretization

We now discuss the discretization of (2.5) and the selection of appropriate forms for the discrete penalty term. For the rest of the paper, we adopt the following notation. Let $\Delta^j \equiv \{S_0^j \equiv 0 < S_1^j < S_2^j < \dots < S_{n-1}^j < S_n^j \equiv$

$S_\infty\}$ be a partition of $\bar{\Omega} \equiv \Omega \cup \partial\Omega$ at time τ_j , with spatial stepsizes, not necessarily uniform, $h_i^j = S_i^j - S_{i-1}^j, i = 1, 2, \dots, n$. In general, the superscript j applied to an operator or a function of τ and/or S denotes evaluation of the operator or function at time τ_j . Denote by $h_\tau^j = \tau_j - \tau_{j-1}, j = 1, 2, \dots$, the j th timestep size with $\tau_0 = 0$. Let $\bar{V}^j(S) \equiv \bar{V}(S, \tau_j)$ be the approximation to the true solution $V(S, \tau_j)$. Furthermore, let $\bar{V}^j = [\bar{V}_1^j, \bar{V}_2^j, \dots, \bar{V}_{n-1}^j]^T$ be the vector of values $\bar{V}_i^j \equiv \bar{V}(S_i^j, \tau_j), i = 1, \dots, n-1$. Similarly, denote by $V^{*,j} = [V_1^{*,j}, V_2^{*,j}, \dots, V_{n-1}^{*,j}]^T$ the vector of the payoff values $V_i^{*,j} = V^*(S_i^j), i = 1, \dots, n-1$.

To proceed from time τ_{j-1} to time τ_j , we apply the standard θ -timestepping discretization scheme to (2.5)

$$(\mathcal{I} - \theta h_\tau^j \mathcal{L}^j) \bar{V}^j(S) = (\mathcal{I} + (1 - \theta) h_\tau^j \mathcal{L}^{j-1}) \bar{V}^{j-1}(S) + \mathcal{P}^j(\bar{V}^j(S)), \quad S \in \Omega, \quad (3.1)$$

where $0 \leq \theta \leq 1$, and incorporate the boundary conditions (2.3) by setting

$$\bar{V}^j(0) = E, \quad \bar{V}^j(S_\infty) = 0. \quad (3.2)$$

Here, \mathcal{I} and \mathcal{P}^j denote the identity and penalty operators, respectively, where \mathcal{P}^j is defined by $\mathcal{P}^j(\bar{V}^j(S)) = \zeta \max(V^*(S) - \bar{V}^j(S), 0)$. The above timestepping technique, together with the boundary conditions, can be viewed as equivalent to solving a nonlinear BVP at each timestep. In (3.1), the values $\theta = 1/2$ and $\theta = 1$ give rise to the standard Crank-Nicolson (CN) and the fully-implicit methods, respectively. It is known that the CN method is second-order accurate, but prone to producing spurious oscillations, while the implicit method is first-order accurate, but maintains strong stability properties (e.g. [31, 40]). To maintain the accuracy of CN as well as smoothness of the solution, we use the Rannacher smoothing technique [32], which applies the fully-implicit timestepping in the first few (usually two) timesteps.

3.1 Finite differences

Applying the standard centered FD discretization for the space variable in (3.1) gives rise to an $(n-1) \times (n-1)$ algebraic system of the form

$$(I + \theta h_\tau^j M^j + \bar{P}^j) \bar{V}^j = (I - (1 - \theta) h_\tau^j M^{j-1}) \bar{V}^{j-1} + \bar{P}^j V^{*,j} + \bar{g}^j, \quad (3.3)$$

where M^j is a tridiagonal matrix that arises from discretizing \mathcal{L}^j by FD on Δ^j , I is the identity matrix, \bar{P}^j is a diagonal penalty matrix, and \bar{g}^j is a vector containing certain values arising from the boundary conditions. The explicit formula for M^j is

$$\begin{aligned} M^j &\equiv \text{trid}\{(M^j)_{i,i-1}, (M^j)_{i,i}, (M^j)_{i,i+1}\} \\ &= \text{trid}\left\{-\frac{1}{2}\sigma^2(S_i^j)^2\beta_{1i}^j - rS_i^j\alpha_{1i}^j, -\frac{1}{2}\sigma^2(S_i^j)^2\beta_{2i}^j - rS_i^j\alpha_{2i}^j + r, -\frac{1}{2}\sigma^2(S_i^j)^2\beta_{3i}^j - rS_i^j\alpha_{3i}^j\right\}, \end{aligned} \quad (3.4)$$

where

$$\begin{aligned} \alpha_{1i}^j &= -\frac{h_{i+1}^j}{h_i^j(h_i^j + h_{i+1}^j)}, \alpha_{2i}^j = \frac{(h_{i+1}^j - h_i^j)}{h_i^j h_{i+1}^j}, \alpha_{3i}^j = \frac{h_i^j}{h_{i+1}^j(h_i^j + h_{i+1}^j)}, \\ \beta_{1i}^j &= \frac{2}{h_i^j(h_i^j + h_{i+1}^j)}, \beta_{2i}^j = -\frac{2}{h_i^j h_{i+1}^j}, \beta_{3i}^j = \frac{2}{h_{i+1}^j(h_i^j + h_{i+1}^j)}, \end{aligned}$$

and where $\text{trid}\{., ., .\}$ denotes a tridiagonal matrix with the sub-, main and super-diagonal elements listed in the brackets, and with the first and last rows modified to take into account the boundary conditions. The penalty matrix \bar{P}^j is defined by

$$(\bar{P}^j)_{i,l} \equiv \begin{cases} \zeta & \text{if } \bar{V}_i^j < V_i^{*,j} \text{ and } i = l, \\ 0 & \text{otherwise.} \end{cases} \quad (3.5)$$

3.2 Finite element collocation methods

For high-order discretization in space, we apply collocation based on quadratic splines. We remind the reader that the space of quadratic splines with respect to partition Δ^j with n subintervals has dimension $n + 2$, and thus we need $n + 2$ conditions. Two of the $n + 2$ conditions are obtained from the boundary conditions (3.2), and the rest come from collocation conditions, as explained further in this section. Let $\bar{V}^j(S) = \sum_i c_i^j \phi_i^j(S)$ be the spline approximation to $V(S, \tau_j)$ expressed in terms of appropriate quadratic spline basis functions $\phi_i^j(S)$ and coefficients or degrees of freedom (DOFs) c_i^j . Let $\mathcal{D}^j \equiv \{D_i^j\}_{i=1}^n$ be the set of collocation points on the partition Δ^j . It is important to emphasize that the choice of collocation points may affect the order of convergence of the resulting methods [8], especially on a non-uniform grid. In the case of a uniform partition, i.e. $\Delta^j \equiv \{S_i^j = ih, i = 0, \dots, n, h = \frac{S_\infty}{n}\}$, and quadratic splines, it is natural to have $\mathcal{D}^j \equiv \{D_i^j = (S_{i-1}^j + S_i^j)/2, i = 1, \dots, n\}$. That is, for a uniform partition, the set of collocation points for a QSC method is chosen to be the set of the midpoints of the partition. For a non-uniform partition, the set of collocation points is defined slightly differently. In Section 5, we describe in detail how the set of collocation points for a QSC method can be constructed on an adaptive grid. For convenience, let $D_0^j \equiv S_0^j = 0$ and $D_{n+1}^j \equiv S_n^j = S_\infty$. Also, let $\bar{V}_I^j(S)$ be the quadratic-spline interpolant of $V^j(S)$ satisfying

$$\bar{V}_I^j(S) = \bar{V}^j(S), \quad S = 0, \quad S \in \mathcal{D}^j, \quad S = S_\infty. \quad (3.6)$$

For the convenience of the reader, we first briefly review spline collocation methods for linear parabolic PDEs. It is known that applying the standard spline collocation discretization to linear parabolic PDEs results in sub-optimal approximations, i.e. the order of convergence of the spline collocation approximation is less than that of the interpolant in the same polynomial space. To obtain optimal (fourth-order) QSC methods for linear parabolic PDEs, appropriate perturbations of the differential operator \mathcal{L} and of the boundary operator, similar to those used to obtain optimal spline collocation methods for BVPs (e.g. [18]), are developed in [6]. An optimal fourth-order spline collocation method can be obtained via either deferred-correction (two-step method), using the perturbation operator in the right-hand side of the collocation equations of the correction step and requiring the solution of two tridiagonal linear systems per timestep, or extrapolation (one-step method), using the perturbation operator in the left-hand side of the collocation equations and requiring the solution of an almost pentadiagonal linear system at each timestep.²

Several optimal (fourth-order) and efficient QSC methods for solving one-dimensional linear parabolic PDEs with general boundary conditions have recently been introduced and studied in [6]. These methods can be viewed as combining the two steps of the deferred-correction method into one, and treating the perturbation term for \mathcal{L} explicitly, thus maintaining the fourth-order accuracy, while requiring the solution of only one tridiagonal linear system per timestep. The QSC discretization for the space variable considered in this paper can be viewed as an extension of the efficient and optimal method named QSC-CN for linear one-dimensional parabolic PDEs in [6] to the context of one-dimensional non-linear PDEs of the form (2.5). More specifically, $\bar{V}^j(S)$ is computed by

$$\begin{aligned} (\mathcal{I} - \theta h_\tau^j \mathcal{L}^j) \bar{V}^j(S) &= (\mathcal{I} + (1 - \theta) h_\tau^j \mathcal{L}^{j-1} + h_\tau^j \mathcal{P}_\mathcal{L}^{j-1}) \bar{V}^{j-1}(S) + \mathcal{P}^j(\bar{V}^j(S)), \quad S \in \mathcal{D}^j, \\ \bar{V}^j(0) &= E, \quad \bar{V}^j(S_\infty) = 0, \end{aligned} \quad (3.7)$$

with $\bar{V}^0(S) = \bar{V}_I^0(S)$, where $\mathcal{P}_\mathcal{L}^j$ is an appropriate perturbation of \mathcal{L}^j . The definition of $\mathcal{P}_\mathcal{L}^j$ on a general grid can be found in [8], and is omitted here for brevity. The reader is referred to [6] for detailed discussions of the relevant methods.

²By ‘‘almost pentadiagonal’’, we mean that all rows of the matrix, except the first two and last two rows, follow a pentadiagonal pattern.

As discussed in [6], for one-dimensional linear parabolic PDEs, the perturbation terms $\mathcal{P}_{\mathcal{L}}^{j-1}(\bar{V}^{j-1}(S))$ corresponding to the first and last collocation points, namely D_1^j and D_n^j , are responsible for potential instability. Experiments show that similar conclusion holds for the PDE (2.5). Among several remedies proposed in [6], we find that the method QSC-CN0, which completely omits the perturbation terms on the first and last collocation points, is simple and works well for pricing American options. Hence we adopt this choice for $\mathcal{P}_{\mathcal{L}}^j$ in (3.7).

Let $c^j \equiv \{c_i^j\}_{i=0}^{n+1}$ and $c^{*,j} \equiv \{c_i^{*,j}\}_{i=0}^{n+1}$ be the vector of the unknown DOFs for the quadratic spline approximation and the vector of the DOFs for the spline interpolant of the payoff on the partition Δ^j , respectively. Method (3.7) gives rise to a $(n+2) \times (n+2)$ algebraic system of the form

$$(Q_0^j + \theta h_\tau^j Q^j + P^j)c^j = (Q_0^{j-1} - (1-\theta)h_\tau^j Q^{j-1} + h_\tau^j Q_{\mathcal{P}}^{j-1})c^{j-1} + P^j c^{*,j} + g^j, \quad (3.8)$$

where Q_0^j is the quadratic spline interpolation matrix for the partition Δ^j , Q^j arises from discretizing \mathcal{L}^j using QSC, and the matrix $Q_{\mathcal{P}}^{j-1}$ arises from $\mathcal{P}_{\mathcal{L}}^{j-1}$. We refer the reader to [6] and [8] for the explicit definitions of these matrices. It is important to note that Q_0^j is a tridiagonal matrix, as opposed to the identity matrix in the FD case. The penalty matrix P^j is also a tridiagonal matrix, as opposed to a diagonal one in the FD case, and is defined by

$$(P^j)_{i,l} \equiv \begin{cases} \zeta(Q_0^j)_{i,l} & \text{if } \bar{V}^j(S) < V^{*,j}(S), \quad S = D_{i-1}^j \in \mathcal{D}^j \cup \partial\Omega, \\ 0 & \text{otherwise,} \end{cases} \quad (3.9)$$

or, equivalently, $P^j = \bar{P}^j Q_0^j$, with

$$(\bar{P}^j)_{i,l} \equiv \begin{cases} \zeta & \text{if } \bar{V}^j(S) < V^{*,j}(S), \quad S = D_{i-1}^j \in \mathcal{D}^j \cup \partial\Omega, \quad \text{and } i = l, \\ 0 & \text{otherwise.} \end{cases} \quad (3.10)$$

4 Penalty iteration

In [14], a penalty iteration algorithm for American put options in the context of FV discretization methods is presented. The penalty iteration algorithms for (3.3) and (3.8) are essentially the same with that in [14], and hence, for brevity, we only present the penalty algorithm for the QSC methods. Let k be the index of the non-linear penalty iteration. Let $c^{j,(k)}$ be the k th estimate of c^j , and $P^{j,(k)}$ be the k th penalty matrix constructed at the j th timestep. The vector of initial guess $c^{j,(0)}$ is usually chosen to be c^{j-1} , which is the vector of the DOFs of the quadratic spline approximation at the previous timestep. A QSC penalty iteration is presented in Algorithm 1. In general,

Algorithm 1: QSC penalty iteration for American options

- 1: Initialize $c^{j,(0)}$;
 - 2: Construct $P^{j,(0)}$ using (3.9);
 - 3: **for** $k = 0, \dots$, until convergence **do**
 - 4: Solve (3.8) for $c^{j,(k+1)}$;
 - 5: Construct $P^{j,(k+1)}$ using (3.9);
 - 6: **if** $\left[\max_i \left\{ \frac{|\bar{V}^{j,(k+1)}(S) - \bar{V}^{j,(k)}(S)|}{\max(1, |\bar{V}^{j,(k+1)}(S)|)} \text{ for } S = D_i^j \in \mathcal{D}^j \right\} < tol \right]$ or $\left[P^{j,(k)} = P^{j,(k+1)} \right]$ **then**
 - 7: **break**;
 - 8: **end if**
 - 9: **end for**
 - 10: $c^j = c^{j,(k+1)}$;
-

for points at which $\bar{V}^j(S) < V^{*,j}(S)$, where $S \in \Delta^j$ for FD methods or $S \in \mathcal{D}^j$ for QSC methods, if we want the

LCP (2.1) to be computed with a relative precision tol we should have $\zeta \simeq \frac{1}{tol}$. So ζ is well-defined, and cannot be arbitrarily large. It is worth noting that, in practice, a small number (one or two) of penalty iterations usually suffices to obtain convergence. Note that, in case of FD methods, the initial guess $\bar{V}^{j,(0)}$, for $j > 1$, is chosen based on linear extrapolation of the numerical solution from the two previous timesteps, that is,

$$\bar{V}^{j,(0)} = \frac{(h_\tau^j + h_\tau^{j-1})}{h_\tau^{j-1}} \bar{V}^{j-1} - \frac{h_\tau^j}{h_\tau^{j-1}} \bar{V}^{j-2}.$$

Extensive experiments have shown that this choice of initial guess is more efficient than the standard choice of the numerical solution at the previous timestep [10]. For $j = 1$, we set $\bar{V}^{1,(0)} = \bar{V}^0$.

4.1 Solution of the LCP

We now investigate the discrete solution of the LCP (2.1). We first consider the FD case. In this case, at each timestep, the solution of (3.3) is required. We define

$$\mathcal{F}\bar{V}_i^j \equiv \left[(I + \theta h_\tau^j M^j) \bar{V}^j - (I - (1 - \theta) h_\tau^j M^{j-1}) \bar{V}^{j-1} - g^j \right]_i, \quad (4.1)$$

where $[\cdot]_l$ denotes the l th component of a vector. In order to obtain an FD approximate solution of (2.1) with an arbitrary level of precision, we need to show that the solution of (3.3) satisfies $\bar{V}_i^j - V_i^* \rightarrow 0$ as $\zeta \rightarrow \infty$ for gridpoints where $\mathcal{F}\bar{V}_i^j > 0$. For FD methods, similarly to FV methods in [14], this follows if we can show that the term

$$[\bar{P}^j(V^{*,j} - \bar{V}^j)]_i \quad (4.2)$$

is bounded independently of ζ . It is also desirable that the bound be independent of the timestep and the spatial mesh spacing, so that ζ can be chosen without regard to the grid and the timestep sizes. We follow the lines of Theorem 4.1 of [14], which essentially gives sufficient conditions that allow us to bound (4.2). For the FD methods, these sufficient conditions are: (i) the matrix M^j arising from discretizing the differential operator \mathcal{L}^j be an \mathcal{M} -matrix, i.e. a matrix with non-positive off-diagonals, and non-singular with the inverse being non-negative, and (ii) $1 - (1 - \theta) h_\tau^j ((M^j)_{i,i-1} + (M^j)_{i,i+1} + r) \geq 0$, where $(M^j)_{i,i-1}$ and $(M^j)_{i,i+1}$ are given in (3.4). Note that condition (ii) arises since we require that $(I - (1 - \theta) h_\tau^j M^{j-1}) \bar{V}^{j-1}$ be bounded (see Appendix A of [14] for a similar proof in the context of FV methods). When a fully implicit method is used, condition (ii) is trivially satisfied, but, when the CN timestepping is used, this condition essentially requires the boundedness of $\frac{h_\tau^j}{\min_i (h_\tau^j)^2}$. In our experiments, this boundedness condition is not always satisfied. However, we observed that, as long as Crank-Nicolson is preceded by a finite number of fully implicit steps (Rannacher smoothing [32]), (4.2) is bounded independently of ζ and ζ can be chosen without regard to the timestep and mesh spacing. Similar observations were also reported in [14], where, in fact, an open question is posed on whether or not we can remove condition (ii) if the Crank-Nicolson timestepping is preceded by a finite number of fully implicit steps.

We now consider condition (i). For the FD methods, it is more convenient to study the property of the matrix M^j based on the following sufficient conditions for the \mathcal{M} -matrix structure: strict diagonal dominance with positive diagonals and non-positive off-diagonals [17]. Note that, if the matrix M^j satisfies these sufficient conditions, then so does the matrix $I + \theta h_\tau^j M^j$, taking into account that θ and h_τ^j are positive.

LEMMA 4.1. Assume that the partition $\Delta^j \equiv \{S_i^j\}_{i=0}^n$ satisfies the conditions

$$h_i^j \leq \frac{\sigma^2 S_{i-1}^j}{r}, \quad i = 3, 4, \dots, n, \quad (4.3)$$

on the spatial stepsizes, where $h_i^j = S_i^j - S_{i-1}^j$, $i = 3, 4, \dots, n$. Then, the matrix M^j defined in (3.4) is a strictly diagonally dominant matrix with positive diagonal and non-positive off-diagonal entries.

Proof: The explicit formula for super-diagonals of M^j is $-\frac{\sigma^2(S_i^j)^2}{h_{i+1}^j(h_i^j+h_{i+1}^j)} - \frac{rS_i^jh_i^j}{h_{i+1}^j(h_i^j+h_{i+1}^j)}$, $i = 1, 2, \dots, n-2$, and hence the super-diagonal elements are always non-positive. The sub-diagonal entries of M^j are $-\frac{\sigma^2(S_i^j)^2}{h_i^j(h_i^j+h_{i+1}^j)} + \frac{rS_i^jh_{i+1}^j}{h_i^j(h_i^j+h_{i+1}^j)}$, $i = 2, 3, \dots, n-1$. Under the given condition (4.3) on the spatial step length, the sub-diagonal entries are non-positive. Thus, under (4.3), all rows have non-positive off-diagonals. Also, all but the first and last rows of M^j have row sums equal to the positive interest rate ($r > 0$), thus, these rows are strictly diagonally dominant, with positive diagonal elements. Taking into account that $S_1^j = h_1^j$, the first row has elements $M_{1,1}^j = -\frac{1}{2}\sigma^2(S_1^j)^2\beta_{21}^j - rS_1^j\alpha_{21}^j + r = (\sigma^2 + r)h_1^j/h_2^j$, and $M_{1,2}^j = -\frac{1}{2}\sigma^2(S_1^j)^2\beta_{31}^j - rS_1^j\alpha_{31}^j = -(\sigma^2 + r)\frac{(h_1^j)^2}{h_2^j(h_1^j+h_2^j)}$, from which we get $M_{1,1}^j > 0$, $M_{1,2}^j < 0$, and $|M_{1,1}^j| > |M_{1,2}^j|$. Similarly, for the last row, we have $M_{n-1,n-2}^j < 0$ under condition (4.3), and the row sum is greater than r . Thus we also have $M_{n-1,n-1}^j > 0$ and $|M_{n-1,n-1}^j| > |M_{n-1,n-2}^j|$. This concludes the proof. \square

For QSC methods, we have not been able to obtain a rigorous proof of the boundedness of term

$$[P^j(c^{*,j} - c^j)]_i. \quad (4.4)$$

However, as a numerical test, we monitor the quantity

$$\max_{i,j} \left[\frac{\max[0, V^{*,j}(S) - \bar{V}^j(S)]}{\max(1, V^{*,j}(S))} \right], \quad S = D_i^j \in \mathcal{D}^j, \quad (4.5)$$

which is a measure of the maximum relative error in enforcing the American constraint using the penalty method. This quantity will be small if the quantity (4.4) is bounded, and ζ is sufficiently large. During experiments, we noticed that, as long as the Rannacher smoothing technique [32] is used, the a posteriori error quantity (4.5) is indeed of the order of *tol*. In Section 6, we report selected statistics of this measure for all the experiments that we run.

4.2 Convergence of the penalty iteration

The convergence study of the penalty iteration in Theorem 6.1 of [14] essentially consists of the following three results: (i) the nonlinear penalty iteration converges to the unique solution of the discretized equation for any initial iterate; (ii) the iterates converge monotonically; (iii) the iteration has finite termination.

For FD methods, the proof of convergence of the penalty iteration is based on the following two conditions: (a) the matrix $I + \theta h_\tau^j M^j + \bar{P}^{j,(k)}$ be non-singular, and (b) the inverse of the matrix $I + \theta h_\tau^j M^j + \bar{P}^{j,(k)}$ be non-negative, where $\bar{P}^{j,(k)}$ is the k th penalty matrix constructed at the j th timestep for FD methods using (3.5). Under the sufficient condition (4.3) for matrix M^j to be an \mathcal{M} -matrix, both (a) and (b) are satisfied, since $I + \theta h_\tau^j M^j + \bar{P}^{j,(k)}$ is a diagonally dominant \mathcal{M} -matrix.

For QSC methods, since the unknowns are the degrees of freedom, we consider an equivalent transformed discretized problem with the unknowns being values instead of degrees of freedoms. To this end, instead of (3.8), we consider the transformed problem

$$(I + \theta h_\tau^j Q^j (Q_0^j)^{-1} + \bar{P}^j) \bar{V}^j = (I - (1 - \theta) h_\tau^j Q^j (Q_0^j)^{-1} + h_\tau^j Q_P^j (Q_0^j)^{-1}) \bar{V}^{j-1} + \bar{P}^j \bar{V}^{*,j} + \bar{g}^j, \quad (4.6)$$

taking into account (3.9) and (3.10). In (4.6), \bar{V}^j and $\bar{V}^{*,j}$ are vectors of option values and payoff values, respectively, on $\mathcal{D}^j \cup \partial\Omega$, and $\bar{g}^j = g^j(Q_0^j)^{-1}$. Similarly to the convergence proof in the FD case, two conditions must be satisfied at the k th iteration of the j th timestep: (a) the matrix $I + \theta h_\tau^j Q^j (Q_0^j)^{-1} + \bar{P}^{j,(k)}$ be non-singular, i.e. Equation (4.6) should have a unique solution, and (b) the inverse of $I + \theta h_\tau^j Q^j (Q_0^j)^{-1} + \bar{P}^{j,(k)}$ be non-negative, where $\bar{P}^{j,(k)}$ is the k th penalty matrix constructed at the j th timestep for QSC methods using (3.10). Consider the matrix $Q^j (Q_0^j)^{-1}$. Unfortunately, since, in general, this is a dense matrix with alternating signs at the off-diagonal entries, it cannot be an \mathcal{M} -matrix, hence we cannot use the same technique that was employed for the convergence proof of FD methods. Rather, we study numerically whether the matrix $I + \theta h_\tau^j Q^j (Q_0^j)^{-1} + \bar{P}^{j,(k)}$ satisfies (a) and (b). Our numerical results show that, at each timestep and for all gridsizes considered, $I + \theta h_\tau^j Q^j (Q_0^j)^{-1} + \bar{P}^{j,(k)}$ is non-singular and its entries are non-negative within a tolerance of size $tol \simeq \frac{1}{\zeta}$. It is also worth noting that the inverse of $Q^j (Q_0^j)^{-1}$ can be proved to be positive. As shown by the numerical results, the fact that $Q^j (Q_0^j)^{-1}$ does not satisfy the sufficient condition of being an \mathcal{M} -matrix does not seem to have ill-effects on the fast convergence of the penalty methods applied to QSC methods. Related observations were reported in [14], where a conjecture was made that the penalty iteration converges rapidly under much weaker conditions than the sufficient condition that the discretized differential operator be an \mathcal{M} -matrix.

5 Adaptive mesh methods

To construct the adaptive grids, we use *monitor* and respective *grading* functions, and the error equidistribution principle [3, 11]. According to the error equidistribution principle, the partition points are distributed in such a way, that the error in some chosen norm (or semi-norm) is equidistributed among the subintervals of the partition. Depending on the norm chosen, a different monitor and a respective grading function arises. Generally, a grading function has the form $\xi(S, \tau) = \int_0^S \tilde{V} dS / \int_0^{S_\infty} \tilde{V} dS$, for some appropriate *monitor* function $\tilde{V}(S, \tau)$. In this formula, the value $\xi(S, \tau)$ at S represents the portion of the approximate error at time τ from the left endpoint of the spatial domain up to point S . To approximate the value of a grading function, the integrals involved in the formula are approximated using appropriate quadrature rules. Usually, the monitor functions involve high-order derivatives of V , which, in a practical situation, are not known. Therefore, approximate values are used to obtain the respective approximate values of the grading functions.

According to [11], for a discretization method with error proportional to $h^p V^{(q)}$, where h is a spatial stepsize and $V^{(q)}$ is the q th derivative of V with respect to S , a good grading function is $\xi(S, \tau) = \int_0^S |V^{(q)}|^{1/p} dS / \int_0^{S_\infty} |V^{(q)}|^{1/p} dS$. For different spatial discretization methods, we possibly obtain different grading functions. We first consider the monitor functions for the FD method. Ignoring higher-order terms, the truncation errors of the FD approximations for the first and second spatial derivatives can be bounded in terms of $\max(h_i^j)^2$ and $(h_{i+1}^j - h_i^j) + \max(h_i^j)^2$, respectively. This means that the FD method is formally first-order. However, through numerical experiments, we found that $h_{i+1}^j - h_i^j$ is small enough so that the error is dominated by $\max(h_i^j)^2$. Then, the (spatial) discretization error of $\mathcal{L}V$ in the FD method is considered second-order with respect to the stepsizes, and involves the third derivative $V^{(3)}$, resulting in the grading function $\xi_f(S, \tau) = \int_0^S |V^{(3)}|^{2/4} dS / \int_0^{S_\infty} |V^{(3)}|^{2/4} dS$. For the QSC methods, we take $q = 3$ (as the error formula for the interpolant suggests) and $p = 3$ (the global order) [8], resulting in the grading function $\xi_q(S, \tau) = \int_0^S |V^{(3)}|^{2/6} dS / \int_0^{S_\infty} |V^{(3)}|^{2/6} dS$.

Given a grading function $\xi(S, \tau_j)$ for a fixed time τ_j and a fixed number of subintervals n , the adaptive algorithm computes points $S_i^j, i = 0, \dots, n$, with $\xi(S_0^j, \tau_j) \equiv \xi(0, \tau_j) = 0$ and $\xi(S_n^j, \tau_j) \equiv \xi(S_\infty, \tau_j) = 1$, such that $\xi(S_i^j, \tau_j) - \xi(S_{i-1}^j, \tau_j) \approx 1/n, i = 1, \dots, n$, or equivalently, $\xi(S_i^j, \tau_j) \approx i/n$. To do this, we apply an iterative

scheme based on Newton's method

$$\begin{aligned} S_i^{j,(k+1)} &= S_i^{j,(k)} - \frac{\xi(S_i^{j,(k)}) - \frac{i}{n}}{\xi'(S_i^{j,(k)})}, \\ &\approx S_i^{j,(k)} - \frac{\mathcal{Q}(\int_0^{S_i^{j,(k)}} \tilde{V}^{j,(k)} dS) - \frac{i}{n} \mathcal{Q}(\int_0^{S_\infty} \tilde{V}^{j,(k)} dS)}{\tilde{V}_i^{j,(k)}}, \end{aligned} \quad (5.1)$$

where $\tilde{V}_i^{j,(k)}$ denotes the approximate value to the monitor function evaluated at $S_i^{j,(k)}$, and $\mathcal{Q}(\cdot)$ is a quadrature rule approximation to an integral. Several quadrature rules may be used, but, in our experiments, we used the trapezoidal rule for the FD method and the midpoint rule for the QSC method. The fact that, for the QSC method, the midpoints are points of high accuracy and no discontinuities motivates the midpoint rule. For FD method, since the gridpoint values are computed, the trapezoidal rule is a natural choice. Furthermore, we found that the variations between those quadrature rules have negligible effect in the final results. In our experiments, we applied only one iteration of (5.1). That is, at most one re-distribution of the spatial points takes place in one timestep, and thus the placement of the spatial points evolves as the timesteps proceed. Experiments show that this choice works well for the American option pricing problem and is attractive due to its efficiency.

To decide whether one re-distribution of the spatial points is needed at a certain timestep, we use the criterion

$$rdrift \equiv \frac{\max_i \{r_i^j\}}{r^j} \leq \alpha, \quad \text{where } r_i^j = \int_{S_{i-1}^j}^{S_i^j} \tilde{V} dS, \quad i = 1, \dots, n, \quad r^j = \frac{\int_0^{S_\infty} \tilde{V} dS}{n}. \quad (5.2)$$

The ratio *rdrift* gives an indication of how well-distributed the partition is. If this ratio is too large, it follows that the maximum error estimate over all subintervals is considerably larger than the average estimate, and thus, the current partition is not well-distributed. That is, for a partition to be well-distributed, the maximum value of r_i^j must be roughly at most α times as large as the average value r^j . Typical choices for α are small numbers, such as $\alpha = 2$ [36]. We have used $\alpha = 5$ in all our experiments, and the results show that this criterion works well for American option pricing.

We next discuss in more detail how our adaptive mesh techniques work. For the purpose of our discussions, denote by $\bar{V}_{\Delta^l}^j$ the approximate solution on the partition Δ^l at timestep j . A generic algorithm for timestepping from time τ_{j-1} to time τ_j using an adaptive mesh technique is summarized in Algorithm 2. Note that we always start with a uniform grid as if we do not know how the solutions behave, and subsequent partitions are fully determined by the adaptive technique.

We now briefly describe the algorithm. In Lines 1 and 2, we apply a timestepping method, usually Crank-Nicolson, with the exception of the Rannacher smoothing technique for the first few timesteps, using the same spatial points for time τ_j as for τ_{j-1} . This computes approximate values $\bar{V}_{\Delta^j}^j \equiv \bar{V}_{\Delta^{j-1}}^j$. We calculate all needed quantities and check the criterion in (5.2). If the points are well-distributed, we proceed to the next timestep (Lines 3 and 4). If not, the new location of the spatial points Δ^j is computed using (5.1) (Lines 5 and 6). See Remark 5.1 about an adjustment to Δ^j for QSC. Next, we need to calculate values of the approximation at the new spatial points at time τ_j , i.e. $\bar{V}_{\Delta^j}^j$. There are two ways to do this. The first is to interpolate $\bar{V}_{\Delta^{j-1}}^{j-1}$ from the old partition Δ^{j-1} to the new partition Δ^j to obtain $\bar{V}_{\Delta^j}^{j-1}$, and then apply the same timestepping procedure that was previously used at Line 2 to compute values of the approximation at the new partition points, i.e. $\bar{V}_{\Delta^j}^j$. The second way is to simply interpolate $\bar{V}_{\Delta^{j-1}}^j$ from Δ^{j-1} to Δ^j to obtain $\bar{V}_{\Delta^j}^j$. The first technique is used in the first few (β) timesteps (Lines 7, 8 and 9), while the second is used in all subsequent timesteps where a remeshing is invoked (Lines 10 and 11). Note that using the first technique for all timesteps is only desirable for functions that are fast-evolving with time. For functions that evolve slowly with time, such as the value function of an American option, this approach is

Algorithm 2: An adaptive algorithm for timestepping from τ_{j-1} to τ_j

- 1: Let $\Delta^j = \Delta^{j-1}$
 - 2: Compute $\bar{V}_{\Delta^j}^j$ by solving either (3.3) or (3.8) with a penalty iteration.
 - 3: **if** (5.2) is satisfied **then**
 - 4: Proceed to Line 2
 - 5: **else**
 - 6: Apply (5.1) (one iteration) to obtain a new Δ^j .
 (For QSC, some adjustment to Δ^j is made, as described in Remark 5.1.)
 - 7: **if** $j < \beta$ **then**
 - 8: Interpolate $\bar{V}_{\Delta^{j-1}}^{j-1}$ to obtain $\bar{V}_{\Delta^j}^{j-1}$.
 - 9: Compute $\bar{V}_{\Delta^j}^j$ by solving either (3.3) or (3.8) with a penalty iteration.
 - 10: **else**
 - 11: Interpolate $\bar{V}_{\Delta^{j-1}}^j$ to obtain $\bar{V}_{\Delta^j}^j$.
 - 12: **end if**
 - 13: **end if**
 - 14: Proceed to step j .
-

unnecessarily inefficient. During the experiments, we observed that (i) remeshings are always required for the first few timesteps, due to the discontinuity of the initial data; and (ii) using the first technique for these initial timesteps is absolutely crucial for the accuracy of the numerical methods. In the experiments, β is chosen to be 4. Below, we elaborate on several fine points of Algorithm 2 in the form of remarks.

REMARK 5.1. We give some details about how the new partition Δ^j computed at Line 2 of the algorithm is adjusted for QSC. As mentioned earlier, the space of quadratic splines with respect to a partition with n subintervals has dimension $n + 2$. For a uniform grid, the natural choice for collocation points is the set of midpoints and the two boundary points. However, for a non-uniform partition, it is not obvious how these points can be chosen, so that the optimal convergence of the resulting methods is preserved. We follow the technique in [7], to construct a non-uniform grid and a set of collocation points for QSC via a certain mapping function.

Denote by $\Delta \equiv \{S_i = ih, i = 0, \dots, n, h = \frac{S_\infty}{n}\}$ the uniform partition of $\bar{\Omega}$ with n subintervals. Assume that we are timestepping from τ_{j-1} to τ_j , and (5.1) has been applied at Line 2 of Algorithm 2 to give a non-uniform partition $\Delta^j = \{S_i^j\}_{i=0}^n$. For adaptive QSC methods, we define a mapping $\psi^j : \bar{\Omega} \rightarrow \bar{\Omega}$ with ψ^j being a bijective strictly increasing function such that

$$\psi^j(0) = 0, \quad \psi^j(S_\infty) = S_\infty, \quad \psi^j((S_{i-1} + S_i)/2) = (S_{i-1}^j + S_i^j)/2, \quad i = 1, \dots, n. \quad (5.3)$$

We then re-define Δ^j and define \mathcal{D}^j by

$$\Delta^j \equiv \{S_i^j = \psi^j(S_i), \quad i = 0, 1, \dots, n\}, \quad \mathcal{D}^j \equiv \{D_i^j = \psi^j((S_{i-1} + S_i)/2), \quad i = 1, \dots, n\}. \quad (5.4)$$

The above adjustment of Δ^j was used in [7] in the context of BVPs, and gave improved accuracy results, therefore, we adopt it here as well.

The mapping ψ^j is generated using the algorithm for monotone piecewise cubic interpolation from [15]. It is important to point out that the basis functions $\phi_i^j(S)$ are defined with respect to the adjusted Δ^j . With this discussion, for the adaptive mesh QSC methods, Line 2 of Algorithm 2 can be broken down into the following substeps:

2.1 Apply (5.1) (one iteration) to obtain $\Delta^j \equiv \{S_i^j\}_{i=0}^n$

2.2 Construct the mapping $\psi^j : \bar{\Omega} \rightarrow \bar{\Omega}$ using (5.3)

2.3 Adjust Δ^j and define \mathcal{D}^j using (5.4)

REMARK 5.2. In Algorithm 2, interpolation is needed at Lines 2 and 2. For adaptive mesh FD methods, interpolation at these steps takes place on the option values, while for adaptive mesh QSC methods, interpolation takes place on other quantities, as explained later. We explain in detail below how interpolation at these steps of the algorithm is done.

First, consider adaptive mesh FD methods. When $j = 1$, i.e. we are timestepping from $\tau_0 = 0$ to τ_1 , one can take an advantage of the initial boundary condition, hence no interpolation is needed at Line 2. When $j > 1$ and a remeshing is required, interpolation needs to be tailored properly to ensure that certain properties of the problem related to the free boundary are not violated. For an American put, at each timestep, the free boundary point separates the spatial domain into the stopping region, where the option value is equal to the payoff, and the continuation region, where the option value is greater than the payoff. Since, for adaptive FD methods, interpolation takes place on the option values, interpolated option values must be (i) equal (within some tolerance) to the payoff in the stopping region, and (ii) larger than the payoff and decreasing in the continuation region. Several possible choices for interpolation include cubic spline interpolation or piecewise cubic Hermite interpolation. It is known that piecewise cubic Hermite interpolation is less oscillating, but also less accurate than cubic spline interpolation. We chose to use cubic spline interpolation, to obtain higher accuracy. With cubic spline interpolation techniques, we observed that (ii) is always satisfied; however, (i) is not always met. To resolve this problem, we adjust the interpolated values to the left of the free boundary point to be equal to the payoff. It should be noted that the free boundary point at each timestep is approximated using the partition points and option values available, namely $(\{S_i^{j-1}\}_{i=0}^n, \bar{V}_{\Delta^{j-1}}^{j-1})$ (Line 2) or $(\{S_i^{j-1}\}_{i=0}^n, \bar{V}_{\Delta^{j-1}}^j)$ (Line 2).

For adaptive mesh QSC methods, it is important to point out that interpolation at Lines 2 and 2 takes place neither on the option values nor on DOFs corresponding to $\bar{V}_{\Delta^{j-1}}^{j-1}$ and $\bar{V}_{\Delta^{j-1}}^j$, but on the values of $(\mathcal{I} + (1 - \theta)h_\tau^j \mathcal{L}^{j-1} + h_\tau^j \mathcal{P}_{\mathcal{L}}^{j-1})\bar{V}_{\Delta^{j-1}}^{j-1}$ or $(\mathcal{I} + (1 - \theta)h_\tau^j \mathcal{L}^j + h_\tau^j \mathcal{P}_{\mathcal{L}}^j)\bar{V}_{\Delta^{j-1}}^j$. We observed that the standard cubic spline interpolation worked well in this case, and no specific tailoring was needed.

REMARK 5.3. In addition to the Rannacher smoothing technique [32], we also adopt another smoothing technique suggested in [31]. That is, at each timestep, for FD methods, we choose to position a gridpoint at the strike E (the initial kink point). We extend this technique to QSC methods, by positioning a collocation point at the strike E . A combination of the Rannacher smoothing and this technique helps preclude large oscillations in the estimation of the hedging parameters [14, 31]. In addition, when we are interested in the option value and its hedging parameters at $S = E$, this technique provides the benefits of (i) avoiding interpolation in case of an FD method; and (ii) obtaining more precise option values and hedging parameters when a QSC method is used, since collocation points are points of high accuracy. However, in the case of adaptive methods, the location of the gridpoints or collocation points is computed dynamically by the adaptive technique, and thus the strike E may fall between gridpoints or collocation points. In these cases, we need to adjust the grid, so that the strike E falls at a gridpoint (FD) or collocation point (QSC). To apply this adjustment, we use the observation that the option values behave linearly in the area towards the left boundary of the domain, i.e. in the stopping region, and, therefore, in that region, few points are needed. Thus, we propose to move one gridpoint from this area to line up with the strike price, if an FD method is used, or to make the strike a collocation point, if a QSC method is used. More specifically, for FD methods, Δ^j contains E as a gridpoint, while for QSC methods, E is one of the midpoints of Δ^j , before the adjustment (5.4). Then, under the mapping (5.3), the set of collocation points \mathcal{D}^j contains E .

It is important to note that, although it would also be desirable to have a partition with a gridpoint (for FD

methods) or a collocation point (for QSC methods) at the free boundary, it is impossible to construct a partition with that property, since we do not know beforehand the exact location of the free boundary at each timestep. This issue appears with any method, adaptive or not. However, adaptive methods partly resolve this issue. As shown by numerical results, the adaptive technique concentrates a lot of points around the free boundary. As a result, the approximation of the free boundary at each timestep is highly accurate. Note that, the free boundary locations approximated by adaptive mesh QSC methods are based on the set of collocation points, rather than the set of gridpoints as in the case of adaptive mesh FD methods.

REMARK 5.4. We now discuss whether the \mathcal{M} -matrix stepsize restriction (4.3) for FD methods may interfere with the stepsizes chosen by the adaptive technique. We emphasize that condition (4.3) does not impose any restriction on h_2^j and h_1^j , since the first row of M^j does not have a sub-diagonal element and the super-diagonal elements are (unconditionally) non-positive. For the rest of the stepsizes, it is possible that the new partition Δ_j constructed using (5.1) may not satisfy condition (4.3). If we wish to enforce condition (4.3), we can do so by monitoring whether (4.3) holds, while computing the points S_i^j , $i = 1, \dots, n-1$, using (5.1), and adjusting the points accordingly, if needed. For example, if a point S_i^j computed by (5.1) is such that h_i^j violates condition (4.3), that is, if $h_i^j > \frac{\sigma^2 S_{i-1}^j}{r}$, we can set $h_i^j = \frac{\sigma^2 S_{i-1}^j}{r}$, and $S_i^j = S_{i-1}^j + h_i^j$. This means that we may need to introduce some extra spatial points. During the experiments, we monitored carefully whether condition (4.3) holds, and noticed that, for all the cases we ran, the points generated by the adaptive FD procedure never violated this condition. Hence, we never had to introduce more spatial points.

REMARK 5.5. The adaptive technique for FD and QSC used in this paper involves an approximation to the third spatial derivative, $V^{(3)}$, of the solution at various points chosen by the adaptive technique itself. Similarly, the high-order QSC method involves an approximation to $V^{(3)}$ at certain points. Since the exact location of the free boundary of the American option pricing problem is unknown, we cannot rule out the possibility that the methods attempt to compute an approximation to $V^{(3)}$ at the free boundary, at which point the solution is only C^1 . While this seems to be a problem from the mathematical point of view, the numerical results suggest that this may not be the case from the practical point of view. As discussed in Section 6, we apply the methods to the European and American option pricing problems. As it is known, the solution of the former is smooth, while that of the latter has limited smoothness on the free boundary. The fact that equally good results (approximately same order errors for same discretizations) are obtained for both problems gives an indication that the lack of smoothness at the free boundary of the solution of the American option pricing problem may not reduce the effectiveness of the adaptive technique, or the high-order method. Detailed results are given in Tables 6.1 and 6.2, with a discussion in Section 6.

5.1 Timestep selector

In [23], a simple, but effective, time stepsize selector is proposed. The idea is to predict a suitable time stepsize for the next timestep, using only information from the current and previous timesteps. A modified version of this scheme, given in [14], was shown to work well on both uniform [10] and fixed non-uniform grids [14]. It would be interesting to examine whether this time stepsize selector works well in the context of adaptive mesh methods. In addition, since the benchmark solutions, to which our numerical option values are compared, are based on option values from [14], and those values were obtained using this time stepsize selector, incorporating it into our methods enables consistent and fair comparisons. According to [14], given a stepsize h_τ^{j+1} , the new stepsize is selected so that

$$h_\tau^{j+2} = \left(\min_i \left[\frac{\text{dnorm}}{\frac{|\bar{V}(S_i^j, \tau_j + h_\tau^{j+1}) - \bar{V}(S_i^j, \tau_j)|}{\max(N, |\bar{V}(S_i^j, \tau_j + h_\tau^{j+1})|, |\bar{V}(S_i^j, \tau_j)|)}} \right] \right) h_\tau^{j+1}. \quad (5.5)$$

Here, dnorm is a user-defined target relative change, and the scale N is chosen so that the method does not take an excessively large stepsize in the area where the value of the option is small. Normally, for option values in dollars, $N = 1$ is used. The reader is referred to [14] for a detailed discussion of this stepsize selector. In all experiments, we used $h_\tau^0 = 10^{-3}$ and $\text{dnorm} = 0.15$ on the coarsest grids. The value of dnorm is reduced by two at each refinement, while h_τ^0 is reduced by four.

6 Numerical results

We first present selected numerical results to demonstrate the high-order convergence rate of the QSC methods applied to the European option pricing problem. We also present results to demonstrate the efficiency of the adaptive mesh technique presented in Section 5 when combined with QSC or FD discretization methods, applied to the same problem. We then consider the American put option pricing problem, and present results that demonstrate the quality of the QSC approximation to the value and the Greeks for this problem, as well as results that indicate the effectiveness of the adaptive mesh techniques, especially in the accurate tracking of the exercise boundary.

All computations in this section were carried out in MATLAB (in double precision). The QSC and FD methods were programmed by us. The linear systems arising from either the European or the American option pricing problem were solved using the backslash operator in MATLAB. In the case of an American put option, the arising non-linear systems were solved by the penalty method described in Section 4. We used the MATLAB functions *pchip* and *ppval* to construct and evaluate a monotone Hermite piecewise cubic interpolant in (5.3) and (5.4), respectively. We also used the MATLAB function *spline* to construct and evaluate a cubic spline interpolant at Lines 2 and 2 of Algorithm 2.

In our implementation, as basis functions for the quadratic spline space defined on partition Δ^j , we choose the functions $\phi_i^j(S)$, $i = 0, \dots, n + 1$, where

$$\phi_i^j(S) \equiv \begin{cases} \frac{(S - S_{i-2}^j)^2}{(S_i^j - S_{i-2}^j)(S_{i-1}^j - S_{i-2}^j)} & \text{for } S_{i-2}^j \leq S \leq S_{i-1}^j, \\ \frac{(S - S_{i-2}^j)(S_i^j - x)}{(S_i^j - S_{i-2}^j)(S_i^j - S_{i-1}^j)} + \frac{(S_{i+1}^j - S)(x - S_{i-1}^j)}{(S_{i+1}^j - S_{i-1}^j)(S_i^j - S_{i-1}^j)} & \text{for } S_{i-1}^j \leq S \leq S_i^j, \\ \frac{(S_{i+1}^j - S)^2}{(S_{i+1}^j - S_{i-1}^j)(S_{i+1}^j - S_i^j)} & \text{for } S_i^j \leq S \leq S_{i+1}^j, \\ 0 & \text{elsewhere.} \end{cases} \quad (6.1)$$

Note that $\phi_i^j(S)$ and $(\phi_i^j)'(S)$ are well-defined at the nodes of the partition. Whenever we need to compute $(\phi_i^j)''(S)$ at a node, we define it by right (without loss of generality) continuity for S_i^j , $i = 0, \dots, n - 1$, and by left continuity for S_n^j .

The set of parameters for the option pricing problems is $E = 100$, $\sigma = 0.80$, $r = 0.10$, $T = 0.25$, for both European and American, while $\zeta = 10^7$ was used as the penalty parameter for American options. These parameters are taken from [14], where the penalty approach adopted in this paper was proposed. The computational domain is truncated to $[0, S_\infty] \equiv [0, 500]$, where the condition (3.2) is applied.

In addition to adaptive mesh methods, we also consider a certain pre-defined mapping function that produces non-uniform, but fixed, partitions with finer points near the strike E , where the discontinuity in the first derivative of the initial data is. The mapping function considered in this paper is based on a sinh function and was suggested in [9]. Variations of it appear frequently in the literature (e.g. [35]). According to this mapping, non-uniform partitions are defined as images of uniform partitions via the function

$$\omega(s_i) = \left(1 + \frac{\sinh(b(s_i - a))}{\sinh(ba)}\right)E, \quad (6.2)$$

where $s_i = S_i/S_\infty = i/n$, $i = 0, 1, \dots, n$. For QSC methods, following (5.4), the set of collocation points under the mapping (6.2) can be determined with $s_i = (S_{i-1} + S_i)/(2S_\infty) = \frac{i-1/2}{n}$, $i = 1, \dots, n$. In (6.2), the parameter a controls the degree of refinement around the strike E . Larger values of a indicate finer partitions near the strike. It is important to note that by choosing $a = i/n$, the gridpoint S_{i+1} falls exactly at E , and by choosing $a = (i + \frac{1}{2})/n$, the midpoint $(S_{i+1} + S_i)/2$ falls exactly at E . The purpose of the parameter b is to ensure that the last gridpoint falls exactly at S_∞ , the right boundary point. The value for b can be obtained by numerically solving the equation $\omega(1) = S_\infty$ with $a = i/n$ for some i , $1 \leq i \leq n$. Figure 6.1 shows the refinement around the strike price $E = 100$ for various values of a on a truncated spatial domain $[0, 500]$ with $n = 80$ subintervals. For $n = 80$, with $a = 0.4$ and $a = 0.45$, the strike is at the 33rd and 37th gridpoint, respectively, while choosing $a = 0.40625$ positions the strike at the 33rd midpoint.

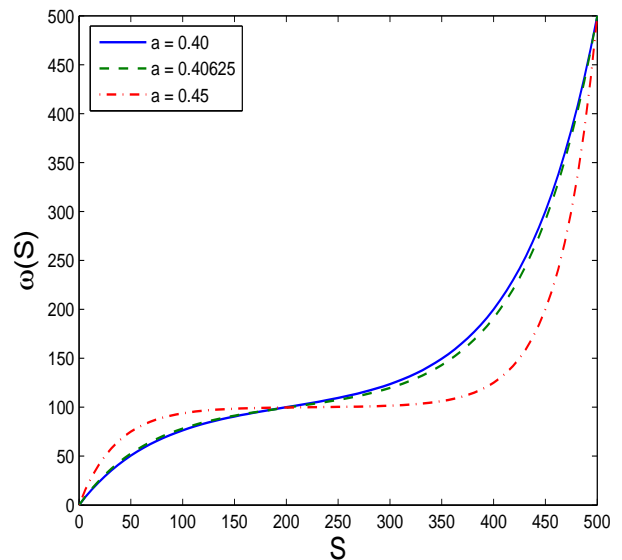


Figure 6.1: Spatial mapping function (6.2) with various values of a on domain $[0, 500]$ and $E = 100$.

Several methods are considered, namely, the adaptive mesh FD method (adaptive FD), the adaptive mesh QSC method (adaptive QSC), the non-uniform mesh FD method (non-uniform FD) and the non-uniform mesh QSC method (non-uniform QSC). The latter two use a predetermined distribution of the spatial points obtained by the mapping function (6.2). As explained earlier, it is beneficial to have the strike as one of the gridpoints (for FD methods) or as one of the collocation points (for QSC methods). The adaptive mesh algorithm, adjusted as in Remark 5.3, automatically generates such a grid for the adaptive FD and the adaptive QSC methods. However, for the non-uniform FD and the non-uniform QSC methods, in order to keep E in the position of a gridpoint and collocation point, respectively, slightly different a 's need to be used. In the experiments, we chose $a = 0.40625$ for the non-uniform QSC methods, while for the non-uniform FD, $a = 0.4$ was used. The gridpoint distributions obtained from these two values of a are virtually the same (see Figure 6.1), so the comparisons are fair. We emphasize that the initial spatial grid for the adaptive FD and QSC methods is uniform, as if we do not know the behaviour of the solution. In this way, we demonstrate more clearly the capabilities of the adaptive techniques.

Since we compare the efficiency of various methods applied to option pricing, it is important to determine the computational cost of each method. The complexity of a method considered in this paper, when applied to pricing either European or American options, at each timestep (or penalty iteration, if any), consists of solving a tridiagonal linear system of size $n \times n$, hence its total cost is proportional to n . For adaptive methods (QSC or FD), for each adaptive timestep in which a remeshing takes place, the extra costs of either interpolating at Lines 2 and solving a tridiagonal linear system of size $n \times n$ at Line 2 or interpolating at Line 2 of Algorithm 2 should also be included. It can be shown that the computation required to construct a cubic spline interpolant is equivalent to solving an $n \times n$ tridiagonal linear system. Since the cost for the evaluation of the cubic spline interpolant at an arbitrary point is constant, the complexity of interpolation in Line 2 or Line 2 of the Algorithm 2 is also proportional to n . We model the total computation cost of each method by the formula

$$\text{total cost} = \text{total} \times n. \quad (6.3)$$

In (6.3), “total” is the number of timesteps (“timesteps #”) or the total number of penalty iterations (“penalty

iterations #”) the method requires for pricing a European or an American option, respectively, including the number of timesteps or penalty iterations required by the adaptive technique, if any, plus the total number of interpolations in the adaptive steps, if any. Note that the total number of interpolations in the adaptive steps is the total number of steps in which a remeshing by the adaptive technique takes place (“adapt #”).

Problem 1: European options

The value of a European option satisfies the Black-Scholes equation and its exact solution as well as hedging parameters (the Greeks), such as delta and gamma, can be found in the literature (e.g. [21]). The adaptive techniques for European options resemble those presented in Algorithm 2 for American options, except that we do not have a constraint on the option values as we do in the case of an American put option. More specifically, in the case of a European option, no penalty iteration is needed in Lines 2 and 2 of Algorithm 2. Table 6.1 shows the numerical results for an at-the-money European put option ($S = E$) obtained by various methods, with variable timesteps chosen by the time stepsize selector (5.5).

Table 6.1: Observed errors for an at-the-money European put option and respective orders of convergence by various methods; variable timesteps are used; Rannacher smoothing is applied in the first few timesteps.

n	timesteps #	value (V)		delta ($\frac{\partial V}{\partial S}$)		gamma ($\frac{\partial^2 V}{\partial S^2}$)		adapt #	total cost
		error	order	error	order	error	order		
adaptive QSC									
80	38	6.67e-04		5.16e-06		5.91e-06		7	3.60e+03
160	138	2.94e-05	4.5	2.20e-06	1.2	1.12e-06	2.4	9	2.35e+04
320	541	2.64e-07	6.8	1.04e-07	4.4	3.36e-07	1.7	15	1.78e+05
non-uniform QSC									
80	35	1.38e-03		6.03e-06		5.79e-06			2.80e+03
240	135	5.64e-05	2.9	4.50e-08	4.4	3.54e-07	2.5		3.24e+04
720	538	2.57e-07	4.9	9.59e-09	1.4	1.92e-08	2.7		3.87e+05
adaptive FD									
80	38	3.94e-03		2.16e-04		7.69e-05		8	3.68e+03
160	73	3.59e-04	3.5	2.23e-05	3.3	1.66e-05	2.3	10	1.33e+04
320	142	4.30e-05	3.1	8.12e-06	1.5	4.13e-06	2.0	15	5.02e+04
640	280	4.56e-07	6.6	1.73e-06	2.3	9.57e-07	2.1	16	1.89e+05
non-uniform FD									
80	35	6.20e-03		9.47e-05		7.45e-06			2.80e+03
160	70	1.56e-03	2.0	2.38e-05	2.0	2.09e-06	1.8		1.12e+04
320	139	3.90e-04	2.0	5.94e-06	2.0	6.00e-07	1.8		4.45e+04
640	276	9.70e-05	2.0	1.48e-06	2.0	1.37e-07	2.1		1.77e+05
1280	551	2.43e-05	2.0	3.71e-07	2.0	3.10e-08	2.2		7.05e+05

Problem 2: American put options

Since no analytic solution for the value of an American put option is available, it is important to establish a highly accurate benchmark solution to which we compare our numerical results. The “exact” option value was computed using the data in [14] (Table 4, implicit constraint, volatility = 0.80) and extrapolation, assuming quadratic conver-

gence, as the methods in [14] are supposed to achieve. With an accuracy requirement of 10^{-7} , the “exact” option value is 14.6788866. Tables 6.2 and 6.3 present selected numerical results for an at-the-money American put option obtained by various methods considered in this paper. In Table 6.2, the values of the options and relevant statistics are presented. We denote by “penalty iter #” the total number of iterations required by the penalty method over all timesteps, including the iterations required in the adaptive mesh generation, if any. In Table 6.2, the quantity “error” is computed as the difference between the “exact” and the numerical solutions; and “adapt #” denotes the total number of remeshings over all timesteps. In Table 6.3, the delta and gamma values are presented. Since we do not have reference values for delta and gamma, to show convergence, we compute the “change” as the difference in delta and gamma values from the coarser grid and the “ratio” as the ratio of changes between successive grids.

Table 6.2: Value of an at-the-money American put option obtained by various methods; variable timesteps are used; Rannacher smoothing is applied in the first few timesteps.

n	timesteps #	value	error	order	penalty iter #	adapt #	total cost
adaptive QSC							
80	57	14.6782991	5.87e-04		141	16	1.26e+04
160	115	14.6788303	5.63e-05	3.4	266	20	4.58e+04
320	230	14.6788877	1.10e-06	5.8	555	23	1.85e+05
non-uniform QSC							
80	54	14.6778209	1.07e-03		99		7.92e+03
240	112	14.6788044	8.26e-05	2.3	247		5.93e+04
720	228	14.6788826	4.40e-06	2.7	597		4.30e+05
adaptive FD							
80	58	14.6730564	5.83e-03		86	11	7.76e+03
160	115	14.6785531	3.33e-04	4.1	169	12	2.90e+04
320	230	14.6788759	1.07e-05	4.9	368	20	1.24e+05
640	460	14.6788877	1.10e-06	3.3	779	39	5.24e+05
non-uniform FD							
80	54	14.6697752	9.11e-03		63		4.32e+03
160	112	14.6766413	2.25e-03	2.0	125		1.79e+04
320	227	14.6783263	5.60e-04	2.0	251		7.26e+04
640	456	14.6787417	1.45e-04	2.0	507		2.92e+05
1280	914	14.6788445	4.21e-05	1.8	1001		1.17e+06
2560	1829	14.6788739	1.27e-05	1.8	1990		4.68e+06

We next compile an efficiency comparison between various methods for solving the two option pricing problems. In Figures 6.2 and 6.3, we plot errors versus computation costs required by each of the methods for the European and American option pricing problems, respectively. Besides methods described in this paper, we also consider the uniform grid control variate method of [21] (control variate) for the pricing of an American option. Among the four methods described in this paper, namely adaptive QSC, adaptive FD, non-uniform QSC, and non-uniform FD methods, it is evident that the adaptive mesh methods significantly outperform the non-uniform methods in general, with the adaptive QSC method being the most efficient method, and the non-uniform FD being the least

Table 6.3: Observed delta and gamma of an at-the-money American option obtained by various methods. These results correspond to option values in Table 6.2.

n	delta ($\frac{\partial V}{\partial S}$)			gamma ($\frac{\partial^2 V}{\partial S^2}$)		
	value	change	ratio	value	change	ratio
adaptive QSC						
80	-0.4056185			0.0100048		
160	-0.4056275	9.02e-06		0.0100209	1.62e-05	
320	-0.4056285	9.87e-07	9.3	0.0100224	1.53e-06	9.1
non-uniform QSC						
80	-0.4056154			0.0100213		
240	-0.4056260	1.05e-05		0.0100224	1.07e-06	
720	-0.4056278	1.86e-06	5.7	0.0100227	3.79e-07	2.8
adaptive FD						
80	-0.3940293			0.0098209		
160	-0.3979309	3.90e-03		0.0098831	6.22e-05	
320	-0.4012858	3.35e-03	1.2	0.0099521	6.89e-05	1.2
640	-0.4030181	1.73e-03	1.9	0.0099779	2.59e-05	1.9
1280	-0.4052580	2.24e-03	0.8	0.0100228	4.49e-05	0.8
non-uniform FD						
80	-0.3948632			0.0098251		
160	-0.4002006	5.34e-03		0.0099258	1.01e-04	
320	-0.4029027	2.70e-03	2.0	0.0099752	4.94e-05	2.0
640	-0.4042625	1.36e-03	2.0	0.0099996	2.44e-05	2.0
1280	-0.4049447	6.82e-04	2.0	0.0100118	1.21e-05	2.0
2560	-0.4052864	3.42e-04	2.0	0.0100178	6.06e-06	2.0

efficient. Below, we discuss in more detail the efficiency comparison between the methods.

More specifically, for low accuracy (about 10^{-3}), the non-uniform QSC and the adaptive QSC are the best methods (and about equivalent between each other), with the adaptive FD being the next best method. For high accuracy (about 10^{-6}), the methods from best to worst are ordered as: adaptive QSC, adaptive FD, non-uniform QSC, non-uniform FD.

Between the non-uniform QSC method and the adaptive FD method, it is interesting to observe that, for low accuracy, the high-order method (non-uniform QSC) wins, while, when high accuracy is required, the efficiency of the adaptive mesh technique becomes more and more pronounced, making the adaptive FD asymptotically more efficient than the non-uniform QSC. This is true for both European and American pricing problems, a fact that confirms the superior efficiency of the adaptive mesh methods over predetermined non-uniform mesh methods.

Furthermore, by comparing the results of the European and American pricing problems, we notice that the accuracies of the adaptive FD, adaptive QSC and non-uniform QSC methods for the American pricing problem are not significantly affected by the lack of smoothness of the solution at the free boundary; see also Remark 5.5.

To quantify the improvement in efficiency between adaptive FD and adaptive QSC for American option pricing with a requirement for high accuracy, we compare the results of adaptive QSC for $n = 320$ with those of adaptive

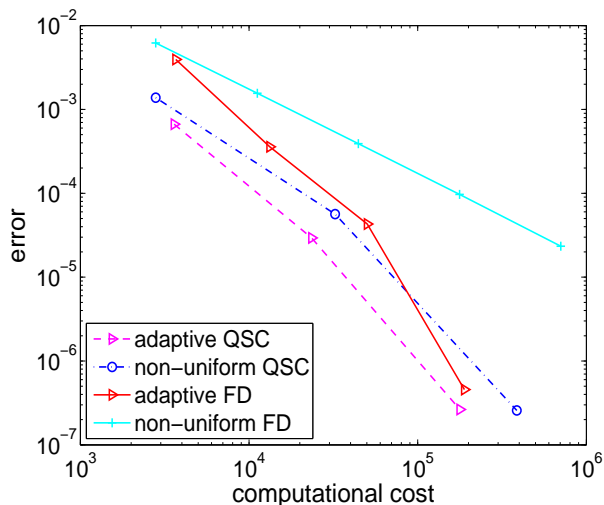


Figure 6.2: Efficiency comparison of various methods applied to the European put option pricing problem.

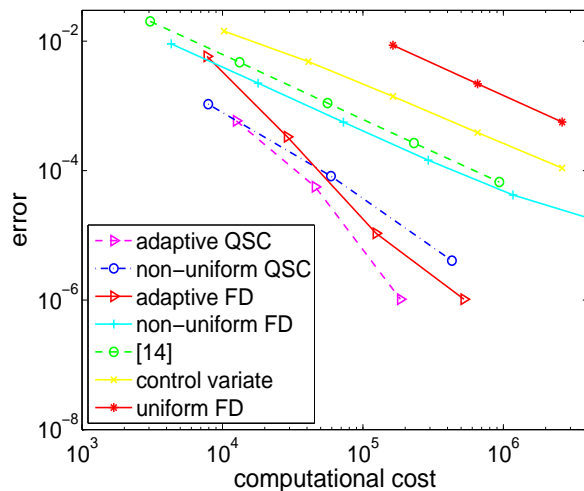


Figure 6.3: Efficiency comparison of various methods applied to the American put option pricing problem.

FD for $n = 640$; see Table 6.2 (error 1.1×10^{-6}). For this case, the adaptive QSC cost is about 25% of the cost of adaptive FD. For low accuracy, we can compare the results of adaptive QSC for $n = 80$ with those of adaptive FD for $n = 160$; see Table 6.2 (errors 5.87×10^{-4} and 3.33×10^{-4} , respectively). For this case, the adaptive QSC cost seems to be less than 50% (but more than 25%) of the cost of adaptive FD.

Regarding other methods for American option pricing, the non-uniform FV method of [14] is even less efficient than the non-uniform FD. These two discretization methods have the same order, but probably the partitions produced by the mapping (6.2) are more favorable than the non-uniform partitions used in [14]. The two remaining methods, namely the control variate and the uniform FD are the least efficient for pricing an American option, with the control variate being slightly more efficient than uniform FD. Certainly, one may introduce adaptive techniques to FV methods or to the control variate method, and possibly improve their efficiency.

Regarding values of delta and gamma of an American option presented on Table 6.3, it is evident that (i) all delta and gamma values indicate convergence, with the delta and the gamma converging to -0.40562 and 0.01002 (with an accuracy of 10^{-5}), respectively; and (ii) the high-order QSC methods (adaptive or non-uniform) provide much more accurate delta and gamma and indicate faster convergence than those provided by the low-order FD methods. However, between an adaptive method and its non-uniform counterpart, it does not seem that the adaptive mesh techniques give substantial benefits in computing these Greeks. These results indicate that high-order methods are particularly effective when the accurate computation of the Greeks is required.

In Figure 6.4, we show the location of the partition points at selected timesteps, as computed by the adaptive QSC method, for the case $n = 80$ and when the number of timesteps is 57. We start with a uniform grid (as if we do not know how the solution behaves). At the first time step, the points are concentrated around the strike. As the time evolves, the points spread to cover the interval between the free boundary (which moves from E to the left and is marked in black) and E , with concentration around the free boundary. Almost no points are needed to the left of the free boundary (where the solution is linear) and few points are needed towards the right end of the interval, where the solution is almost linear. These results indicate that the adaptive technique faithfully captures the behavior of the solution, and places more points around the discontinuity points.

6.1 Penalty iteration for QSC methods

As a posterior check, we report the quantity (4.5), which is a measure of the maximum relative error in enforcing the American constraint using the penalty method with QSC. Also, as an additional check, we monitor the size of the relative residual of Equation (3.8) for all collocation points with $(P^j)_{i,l} = 0$, where $(P^j)_{i,l}$ is defined in (3.9). Since these collocation points are in the continuation region, we expect that, at those points, the Black-Scholes PDE is satisfied, and the residual to be zero within machine epsilon. In Table 6.4, we present observed values of the quantity (4.5) and the residual of (3.8) at the continuation region collocation points from experiments with QSC methods when $\zeta = 10^7$, selected results of which are reported in Table 6.2 and Table 6.3. Note that this choice of the penalty parameter results in a maximum relative error (4.5) in enforcing the American constraint of magnitude of about 10^{-9} , or less, that is, very well below the time and space discretization errors. At the continuation region collocation points, the relative residual of (3.8) is of the size 10^{-14} , i.e. the size of machine epsilon. In addition, in Table 6.5, we give statistics for test cases with varying the penalty parameter ζ . It is interesting to observe that the number of iterations fluctuates insignificantly with changes in value of ζ . This can be viewed as a result of the finite termination property of the penalty iteration.

Table 6.4: Observed values of the quantity (4.5) and the residual of (3.8) in the continuation region when QSC methods are employed. These values are collected from experiments whose results are reported in Table 6.2 and Table 6.3.

adaptive QSC			non-uniform QSC		
n	error (4.5)	residual of (3.8)	n	error (4.5)	residual of (3.8)
80	6e-10	5e-14	80	2e-09	6e-14
160	2e-10	3e-14	240	6e-09	8e-14
320	3e-10	2e-14	720	8e-10	7e-14

Table 6.5: Test of varying the penalty parameter ζ with non-uniform QSC. The tolerance tol in Algorithm 2 is chosen to be $\frac{1}{\zeta}$. The “***” indicates that iterations fail to converge, most likely due to machine epsilon limitations.

ζ	penalty iter #	value	error (4.5)	residual of (3.8)
10^4	212	14.6788016	8e-06	7e-14
10^6	229	14.6788044	6e-09	5e-14
10^8	221	14.6788044	7e-10	4e-14
10^{10}	228	14.6788044	1e-13	4e-14
10^{12}	***	***	***	***

We now study the non-singularity of the matrix $I + \theta h_\tau^j Q^j (Q_0^j)^{-1} + \bar{P}^{j,(k)}$ in (4.6) and the non-negativeness of its inverse. We experimented with different values for gridsizes n and time stepsizes h_τ^j . The non-singularity of the matrix is implied from the numerical results. Regarding the non-negativeness of its inverse, in Table 6.6, we report the most negative entry of the inverse matrix among all penalty iterations and the respective value of the time stepsize h_τ^j with varying values of the penalty parameter ζ . It is evident from the numerical results in Table

6.6 that the magnitude of the most negative entry of the inverse is indeed of order $tol \simeq \frac{1}{\zeta}$, and does not decrease as the grid sizes n increase and time stepsizes h_τ^j decrease. The combination of $h_\tau^j = 2.5 \times 10^{-4}, 5.0 \times 10^{-5}$ and $n = 80$ results in a slightly larger magnitude of the most negative entry, but such a combination is an extreme case (too small time stepsize compared to space stepsize), and should not be chosen, since it would result in a serious imbalance between the time and space discretization errors.

Table 6.6: Non-negativeness of the inverse of $I + \theta h_\tau^j Q^j (Q_0^j)^{-1} + \bar{P}^{j,(k)}$ with non-uniform QSC methods with varying penalty parameter ζ .

ζ	h_τ^j	$n = 80$	$n = 160$	$n = 320$	$n = 640$	$n = 1280$
10^7	5.0e-05	-6.4704e-03	-6.4981e-09	-6.5259e-09	-5.0510e-09	-5.1324e-09
	2.5e-04	-6.6615e-03	-6.8037e-09	-3.7908e-09	-5.1415e-09	-9.6309e-09
	1.0e-03	-7.3388e-09	-1.3990e-09	-5.3757e-09	-9.5919e-09	-1.3699e-08
	5.0e-03	-2.6086e-09	-6.2358e-09	-1.0501e-08	-1.4175e-08	-1.6833e-08
	1.0e-02	-4.1441e-09	-8.4235e-09	-1.2526e-08	-1.5623e-08	-1.7696e-08
	5.0e-02	-8.8121e-09	-1.3083e-08	-1.6013e-08	-1.7837e-08	-1.8924e-08
	1.0e-01	-1.0813e-08	-1.4614e-08	-1.7000e-08	-1.8412e-08	-1.9225e-08
	5.0e-01	-1.4420e-08	-1.6993e-08	-1.8422e-08	-1.9201e-08	-1.9593e-08
10^9	5.0e-05	-6.4704e-03	-6.4981e-11	-6.5259e-11	-5.0510e-11	-5.1324e-11
	2.5e-04	-6.6615e-03	-6.8037e-11	-3.7908e-11	-5.1415e-11	-9.6310e-11
	1.0e-03	-7.3388e-11	-1.3990e-11	-5.3757e-11	-9.5919e-11	-1.3699e-10
	5.0e-03	-2.6086e-11	-6.2358e-11	-1.0501e-10	-1.4175e-10	-1.6834e-10
	1.0e-02	-4.1441e-11	-8.4235e-11	-1.2526e-10	-1.5624e-10	-1.7697e-10
	5.0e-02	-8.8121e-11	-1.3083e-10	-1.6013e-10	-1.7838e-10	-1.8930e-10
	1.0e-01	-1.0813e-10	-1.4614e-10	-1.7001e-10	-1.8415e-10	-1.9236e-10
	5.0e-01	-1.4420e-10	-1.6994e-10	-1.8425e-10	-1.9215e-10	-1.9652e-10

6.2 Early exercise boundary

An interesting problem associated with American option pricing is the analysis of the early exercise boundary and the optimal stopping time. This problem has attracted a lot of attention due to its theoretical and practical importance. The accuracy with which we locate the free boundary has strong effects on the quality of the numerical value of the option computed. As mentioned earlier, the exact analytical expression for the free boundary is not known and locating it accurately and efficiently is a challenging problem. Many researchers have investigated various models, such as integral equations or asymptotic solutions, leading to approximations for the free boundary (e.g. [2, 5, 24, 27, 33]). The purpose of this subsection is to demonstrate the accuracy of the adaptive mesh methods in locating the free boundary for an American put option at each timestep. We carry out a comparison between the free boundary values obtained by FD methods and those obtained by several other methods, namely binomial methods, trinomial methods, integral equation and asymptotic approximation methods of [33], and a method was first proposed by Barone-Adesi and Whaley in [2] and later augmented by MacMillan [27] (the ‘‘BWM’’ method). The set of parameters is $E = 10$, $\sigma = 0.25$, $r = 0.10$, $T = 0.05$ from [33]. The semi-truncated domain in our experiment is $[0, 50]$ and the penalty parameter is $\zeta = 10^7$. We used a grid with 200 mesh points and 200 constant timesteps with non-uniform and adaptive FD methods. In Figure 6.5, we plot the profiles of the free boundary

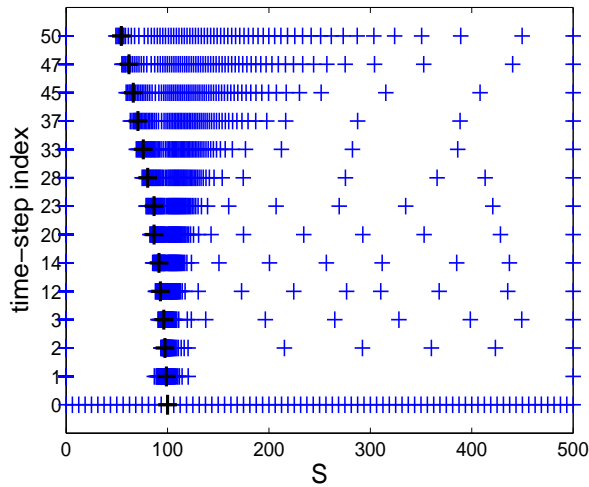


Figure 6.4: The location of the gridpoints chosen by the adaptive QSC method applied to the American put option pricing problem ($n = 80$).

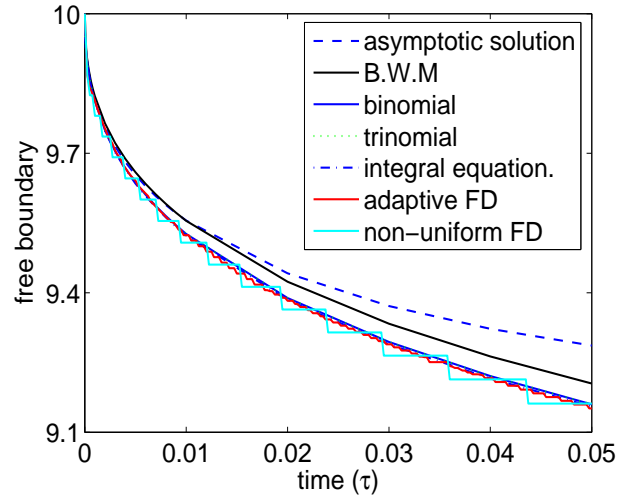


Figure 6.5: Profile of the free boundary points obtained by various methods.

at each timestep obtained by various methods versus time. It is evident that the adaptive technique (adaptive FD) captures the free boundary locations quite well as time evolves, and certainly much better than the non-uniform methods (non-uniform FD). Note that, for both the binomial and trinomial methods, a depth of 1000 subdivisions was used, and these results are considered reference solutions in [33]. The free boundary locations captured by the adaptive technique follow closely those obtained by the tree methods and the integral equation method. The profiles generated by adaptive mesh methods are smooth, while those generated by non-uniform methods are highly non-smooth and look like a step function, and hence do not capture the movement of the moving free boundary properly. The data used for the plot are taken from Table 2 of [33], except the adaptive FD and non-uniform FD methods, which are implemented by us.

7 Conclusions and extensions

We have considered a PDE approach to price American options written on a single asset. We have formulated several highly accurate and efficient methods for pricing American options. These methods are built upon second-order centered FD or optimal fourth-order QSC methods for the spatial discretization, and are integrated with adaptive mesh PDE methods, which rely on grading and monitor functions to determine the distribution of the error along the spatial dimension, and, from that, the location of the spatial gridpoints. At certain timesteps, the adaptive techniques relocate the nodes to equidistribute the error in some chosen norm among the subintervals of the partition. For the solution of the LCP at each timestep, we considered a discrete penalty method. The results show that adaptive PDE methods are effective on the American option pricing problem, and, in particular, they have better ratio of accuracy over computational cost compared to their non-adaptive (still non-uniform) counterparts, and allow for more accurate tracking of the moving boundary. Furthermore, high-order spatial discretization methods have better ratio of accuracy over computational cost compared to their standard second-order counterparts, and provide highly accurate option values and the Greeks delta and gamma of the options. The combination of high-order and adaptive mesh methods gives the best results regarding ratio of accuracy over computational cost.

We conclude by mentioning some extensions of this work. It would be desirable to have a theoretical analysis of the boundedness of (4.4) and the convergence of the penalty iteration in the context of the QSC methods that we have observed in the experiments. It would also be interesting to extend the pricing methods considered in this paper to other American-style options, such as American-style Asian options, or the pricing of convertible bonds with early exercise features. In addition, an application of adaptive techniques to other exotic options, such as barrier options, is of much interest. The fact that we obtained high accuracy over cost by using a uniform grid to start the adaptive technique and letting the adaptive technique take over the “study” of the solution of the American option pricing problem is an indication that the adaptive mesh methods have potential to be used as a “black-box” to determine the behaviour of the solution of other related financial problems as well. Extending the adaptive techniques to multidimensional problems is certainly challenging. In this regard, possible approaches include moving mesh FD methods, such as [19, 20], moving mesh spline collocation methods, and skipped grid spline collocation methods [28]. However, such approaches involve considerable overhead, and their effectiveness has not yet been studied extensively, even for simple PDE problems. There is a lot more to be done to make the adaptive and/or high-order methods effective and practical for the solution of multidimensional financial problems.

References

- [1] Y. ACHDOU AND O. PIRONNEAU, *Computational Methods for Option Pricing*, SIAM, Philadelphia, 2005.
- [2] G. BARONE-ADESI AND R. WHALEY, *Efficient analytic approximations of American option values*, *Journal of Finance*, 42 (1987), pp. 301–320.
- [3] G. F. CAREY AND H. T. DINH, *Grading functions and mesh redistribution*, *SIAM J. Numer. Anal.*, 22 (1985), pp. 1028–1040.
- [4] P. CARR, R. JARROW, AND R. MYNENI, *Alternative characterizations of American put options*, *Mathematical Finance*, 2 (1992), pp. 87–106.
- [5] X. CHEN AND J. CHADAM, *A mathematical analysis of the optimal exercise boundary for American put options*, *SIAM J. Math. Anal.*, 38 (1998), pp. 1613–1641.
- [6] C. CHRISTARA, T. CHEN, AND D. M. DANG, *Quadratic spline collocation for one-dimensional linear parabolic partial differential equations*, *Numerical Algorithms*, 53 (2010), pp. 511–553. doi:10.1007/s11075-009-9317-9, <http://www.springerlink.com/content/v1g8565702373446/>.
- [7] C. C. CHRISTARA AND K. S. NG, *Adaptive techniques for spline collocation*, *Computing*, 76 (2006), pp. 259 – 277.
- [8] ———, *Optimal quadratic and cubic spline collocation on nonuniform partitions*, *Computing*, 76 (2006), pp. 227 – 257.
- [9] N. CLARKE AND K. PARROTT, *Multigrid for American option pricing with stochastic volatility*, *Applied Mathematical Finance*, 6 (1999), pp. 177–195.
- [10] D. M. DANG, *Adaptive finite difference methods for valuing American options*, Master’s thesis, University of Toronto, Toronto, Ontario, Canada, 2007.
- [11] C. DE BOOR, *Good approximation by splines with variable knots II*. Lecture notes in Mathematics, 1974.

- [12] K. ERIKSSON AND C. JOHNSON, *Adaptive finite element methods for parabolic problem. I. A linear model problem*, SIAM J. Numer. Anal., 28 (1991), pp. 43–77.
- [13] ———, *Adaptive finite element methods for parabolic problem. II. Optimal error estimates in $L_\infty L_2$ and $L_\infty L_\infty$* , SIAM J. Numer. Anal., 32 (1995), pp. 706–740.
- [14] P. A. FORSYTH AND K. VETZAL, *Quadratic convergence for valuing American options using a penalty method*, SIAM J. Sci. Comput., 23 (2002), pp. 2095–2122.
- [15] F. N. FRITSCH AND R. E. CARLSON, *Monotone piecewise cubic interpolation*, SIAM J. Numer. Anal., 17 (1990), pp. 238–246.
- [16] M. C. FU, S. B. LAPRISE, D. B. MADAN, Y. SU, AND R. WU, *Pricing American options: A comparison of Monte Carlo simulation approaches*, Journal of Computational Finance, 4 (2001), pp. 39–88.
- [17] W. HACKBUSH, *Iterative Solution of Large Sparse Systems of Equations*, Springer, 1993.
- [18] E. N. HOUSTIS, C. C. CHRISTARA, AND J. R. RICE, *Quadratic-spline collocation methods for two-point boundary value problems*, Internat. J. Numer. Methods Engrg., 26 (1988), pp. 935–952.
- [19] W. HUANG AND R. D. RUSSELL, *A high dimensional moving mesh strategy*, Appl. Numer. Math, 26 (1997), pp. 63–76.
- [20] ———, *Moving mesh strategy based on a gradient flow equation for two-dimensional problems*, SIAM J. Sci. Comput., 20 (1999), pp. 998–1015.
- [21] J. C. HULL, *Options, Futures, and Other Derivatives*, Prentice Hall, seventh ed., 2008.
- [22] L. JIANG AND M. DAI, *Convergence of binomial tree methods for European and American path-dependent options*, SIAM J. Numer. Anal., 42 (2004), pp. 1094–1109.
- [23] C. JOHNSON, *Numerical solution of partial differential equations by the finite element method*, Cambridge University Press, 1987.
- [24] R. KUSKE AND J. KELLER, *Optimal exercise boundary for an American put option*, Appl. Math. Finance, 5 (2007), pp. 107–116.
- [25] F. A. LONGSTAFF AND E. S. SCHWARTZ, *Valuing American options by simulation: A simple least-squares approach*, The Review of Financial Study, 14 (2001), pp. 113–149.
- [26] P. LÖTSTEDT, J. PERSSON, L. V. SYDOW, AND J. TYSK, *Space-time adaptive finite difference method for European multi-asset options*, Computers & Mathematics with Applications, 53 (2007), pp. 1159–1180.
- [27] L. MACMILLAN, *Analytic approximation for the American put option*, Advanced Futures Options Research, 1 (1986), pp. 119–139.
- [28] K. S. NG, *Spline Collocation on Adaptive Grids and Non-Rectangular Domains*, PhD thesis, Department of Computer Science, University of Toronto, June 2005.
- [29] B. NIELSEN, O. SKAVHAUG, AND A. TVEITO, *Penalty and front-fixing methods for the numerical solution of American option problems*, Journal of Computational Finance, 5 (2002), pp. 69–97.

- [30] J. PERSSON AND L. V. SYDOW, *Pricing European multi-asset options using a space-time adaptive FD-method*, *Comput. Visual Sci.*, 10 (2007), pp. 173–183.
- [31] D. M. POOLEY, K. R. VERZAL, AND P. A. FORSYTH, *Convergence remedies for non-smooth payoffs in option pricing*, *Journal of Computational Finance*, 6 (2003), pp. 25–40.
- [32] R. RANNACHER, *Finite element solution of diffusion problems with irregular data*, *Numerische Mathematik*, 43 (1984), pp. 309–327.
- [33] R. STAMICAR, D. ŠEVČOVIČ, AND J. CHADAM, *The early exercise boundary for the American put near expiry: Numerical approximation*, *Canada Applied Mathematics*, 7 (1999), pp. 427–444.
- [34] D. TAVELLA AND C. RANDALL, *Pricing financial instruments: The finite difference method*, John Wiley & Sons, Chichester, 2000.
- [35] J. TOIVANEN, *Numerical valuation of European and American options under Kou’s jump-diffusion model*, *SIAM J. Sci. Comput.*, 30 (2008), pp. 1949–1970.
- [36] R. WANG, P. KEAST, AND P. MUIR, *A high-order global spatially adaptive collocation method for 1-D parabolic PDEs*, *Applied Numerical Analysis*, 50 (2004), pp. 239–260.
- [37] P. WILMOTT, J. DEWYNNE, AND S. HOWISON, *Option pricing: Mathematical Models and Computation*, Oxford Financial Press, 1993.
- [38] P. WILMOTT, S. HOWISON, AND J. DEWYNNE, *Mathematics of Financial Derivatives*, Cambridge University Press, 1995.
- [39] R. ZVAN, P. A. FORSYTH, AND K. VETZAL, *Penalty methods for American options with stochastic volatility*, *Journal of Computational and Applied Mathematics*, 91 (1998), pp. 199–218.
- [40] ———, *Robust numerical methods for PDE models of Asian options*, *Journal of Computational Finance*, 2 (1998), pp. 39–78.