Penalty Methods for Bilateral XVA Pricing in European and American Contingent Claims by a PDE Model

Yuwei Chen[1] and Christina C. Christara[2]
Department of Computer Science
University of Toronto
Toronto, Ontario M5S 3G4, Canada
{ywchen,ccc}@cs.toronto.edu

Abstract

Taking into account default risk in the valuation of financial derivatives has become increasingly important, especially after the 2007-2008 financial crisis. Under some assumptions, the valuation of financial derivatives, including a value adjustment to account for default risk (the so-called XVA), gives rise to a nonlinear PDE [9]. We propose numerical methods for handling the nonlinearity in this PDE, the most efficient of which are discrete penalty iteration methods. We first formulate a penalty iteration method for the case of European contingent claims, and study its convergence. We then extend the method to the case of American contingent claims, resulting in a double-penalty iteration. We also propose boundary conditions and their discretization for the XVA PDE problem in the cases of call and put options, as well as the case of a forward contract. Numerical results demonstrate the effectiveness of our methods.

Key words: Partial Differential Equations, Black-Scholes, credit risk, default risk, credit valuation adjustment, call option, put option, forward contract, nonlinear iteration, finite differences, Crank-Nicolson, control problem, Hamilton-Jacobi-Bellman (HJB) equation.

1 Introduction

Counterparty credit risk [19, 16], also known as default risk, is the risk that, during the duration of an agreement, one party fails to make payments to another party. Derivatives also expose counterparties to credit risk. Since the 2007-2008 financial crisis, it has become a standard practice for derivatives' sellers to adjust the value of the derivatives transactions to reflect the possible losses from counterparty default. This adjustment is usually called credit valuation adjustment (CVA) [19, 16]. However, a more complete valuation adjustment includes debt valuation adjustment, funding valuation adjustment, etc.

Debt valuation adjustment (DVA) [19, 16] is an estimate of the costs to the counterparty of the seller's default risk. It can also be seen as the CVA from the point of view of the counterparty. Funding valuation adjustment (FVA) is an adjustment to the value of a derivative or a portfolio which is designed to make sure that a dealer recovers its average funding costs when he or she trades and hedges derivatives.

Although it is still controversial whether or not a seller should charge DVA or FVA to value derivatives [18, 10, 17], in this work, we leave the debates behind and mainly focus on the mathematical model of valuation adjustment and the numerical method to solve the arising partial differential equation (PDE). We now define the total valuation adjustment [16, 1] as

$$XVA = DVA - CVA + FVA.$$

[1]This research was partly supported by the Ontario Graduate Scholarship (OGS)
[2]This research was partly supported by the Natural Sciences and Engineering Research Council of Canada (NSERC)

If V is the unadjusted value of the derivative, e.g. calculated by Black-Scholes-Merton model, and $\hat{V}$ is the value after taking credit risk into account, then the valuation adjustment is

$$XVA = \hat{V} - V. \tag{1}$$

Note that, the CVA (or XVA) is normally calculated at a counterparty level and not the trade level. This means the CVA can be different to different counterparties even if they have the same derivatives.

The Basel III regulation framework was developed in response to the 2007-2008 financial crisis. The committee set rules for OTC market that bilateral trading needs to take default risk and funding costs into account. Once the rules were set, several methods and frameworks were developed for valuation of derivatives under counterparty risk. Piterbarg [22] derived valuation formulas for the price of derivatives, which incorporate funding cost and collateral agreements. The new model forces adjustment to the discounting term of Black-Scholes PDE. Burgard and Kjaer [8, 9, 11] generalized Piterbarg's model to include sellers and buyers' default risk. They applied the replication portfolio approach to derive the PDE representations of derivatives' values with bilateral risk and funding costs. Moreover, via the Feymann-Kac theorem, the solution to the resulting PDE can also be written in terms of expectation. Brigo and his coauthors [5, 6, 7] introduced a general valuation framework for calculating the bilateral CVA by an expectation representation of risk valuation including CVA, funding spread and collateralization. Simulation approaches such as Monte Carlo are used to approximate the arbitrage-free valuation of bilateral counterparty risk under collateralization. In [5, 6, 7], the analysis focuses on credit default swap (CDS) as underlying portfolios and claimed CVA can be viewed as an option, so-called contingent credit default swap, on the clean value of the contract.

Another popular approach to adjustment valuation is based on *backward stochastic differential equation* (BSDE) analysis. Crépey [13] introduced a BSDE approach to valuation and hedging of bilateral counterparty risk on OTC derivatives under funding constraints. Bichuch, Capponi and Strurm [4] also developed an arbitrage-free valuation framework with a BSDE for the price of European claim taking into account funding spread, the repo market, collateral servicing costs and counterparty credit risks. They show that the XVA is unique in the absence of rate asymmetries. A PDE representation of the BSDE was also shown and studied.

Regarding the PDE arising from the XVA problem, depending on how the derivative is valued in the case of either party of the contract defaults, a nonlinear term may be included [9]. Hence, appropriate numerical methods are needed to handle the nonlinear term, as well as the discretization of the PDE. Arregui, Salvador, and Vázquez consider the PDE models in [9] and price European options with XVA [1], as well as American options with XVA [2]. In both cases, they use the characteristics method for timestepping, a finite element method for the space discretization, and a fixed-point iteration scheme to handle the nonlinearity. The experimental results indicate that their method is first order convergent. Arregui, Salvador, and Vázquez [3] also developed a Monte Carlo approach to American options pricing including XVA. They present the adaptation of Monte Carlo to numerically solve the nonlinear term in the XVA problem.

In our work, we use second-order finite differences in space and Crank-Nicolson discretization in time, with Rannacher smoothing when needed. We impose boundary conditions that are appropriate for call and put options, as well as for the forward contract. The main contributions of this work are

- a fast iterative method for handling the nonlinearity in the XVA PDE problem,
- its convergence analysis, as well as
- its extension to American style derivatives' XVA pricing.

The paper is organized as follows. In Section 2, we present the formulation of the (European option) XVA pricing problem as a PDE. In Section 3, we describe the numerical methods used for the discretization of the PDE and of the boundary conditions, and introduce two types of iteration methods for handling the nonlinearity. For the most efficient of these, we present a theorem about its convergence with the proof found in the Appendix. Section 4 includes the extension of the above methods to the American option XVA pricing problem. In Section 5, we present numerical experiments to study the behaviour of the proposed methods, for XVA pricing in European and American options. Section 6 presents the conclusions and possible extensions of this work.

2 Formulation

In this section, we show the PDE representation for the value of a financial derivative, if the effects of bilateral default risk and the funding costs are considered [9]. The PDE model is extended from Black-Scholes PDE, driven by using multiple hedging arguments, including underlying assets, the zero-coupon bonds of the two parties and risk-free zero-coupon bonds. Note that, in [9], the two parties are refereed to as party B (seller, bank), and party C (counterparty, investor), and we adopt the same naming/notation.

According to the International Swaps and Derivatives Association (ISDA) 2002 Master Agreement, in case either party of a contract defaults, the value of the derivative is determined by a Mark-to-Market rule M, which is chosen to be either $\hat{V}$ or V. Let S be the underlying stock value, σ the volatility in S, r the risk-free interest rate, γ the dividend yield of S, q the stock repo rate, r_B the seller's bond yield, r_C the counterparty's bond yield, $\lambda_B = r_B - r$ (bank hazard rate, instantaneous probability of bank default), $\lambda_C = r_C - r$ (counterparty hazard rate), R_B the recovery percentage on M if seller defaults, R_C the recovery percentage on M if counterparty defaults, r_F the seller's funding rate for borrowed cash, where $r_F = r$ if derivative can be used as collateral, and $r_F = r + (1 - R_B)\lambda_B$ if derivative cannot be used as collateral, $s_F = r_F - r$. The positive and negative values of any security W are denoted by $W^+ \equiv \max\{W, 0\}$ and $W^- \equiv \min\{W, 0\}$.
Define also the differential operator

$$\mathcal{L} \equiv \frac{\sigma^2 S^2}{2}\frac{\partial^2}{\partial S^2} + (q - \gamma)S\frac{\partial}{\partial S} - r\mathcal{I}.$$

In [9], under certain assumptions, and using replicating portfolio arguments, the following PDE model for $\hat{V}$ is obtained

$$\begin{cases} \frac{\partial \hat{V}}{\partial t} + \mathcal{L}\hat{V} = (\lambda_B + \lambda_C)\hat{V} - \lambda_B(R_B M^- + M^+) - \lambda_C(R_C M^+ + M^-) + s_F M^+, \\ \hat{V}(T, S) = H(S). \end{cases} \tag{2}$$

where t is the (forward) time variable and T the maturity time of the derivative.
Note also that riskless [3] option value V satisfies the Black-Scholes PDE

$$\begin{cases} \frac{\partial V}{\partial t} + \mathcal{L}V = 0, \\ V(T, S) = H(S). \end{cases} \tag{3}$$

The Mark-to-Market value M of $\hat{V}$ at default is often chosen to be V or $\hat{V}$. In this paper, we assume $M = \hat{V}$. Then, PDE (2) becomes

$$\begin{cases} \frac{\partial \hat{V}}{\partial t} + \mathcal{L}\hat{V} = \lambda_B(1 - R_B)\hat{V}^- + \lambda_C(1 - R_C)\hat{V}^+ + s_F\hat{V}^+, \\ \hat{V}(T, S) = H(S). \end{cases} \tag{4}$$

[3] "Riskless" derivative in this paper means a financial derivative without considering counterparties' default risk.

Let U denote the XVA. Using $\hat{V} = V + U$ and PDE (3), PDE (4) can be written for U as

$$\begin{cases} \frac{\partial U}{\partial t} + \mathcal{L}U = \lambda_B(1 - R_B)(V + U)^- + \lambda_C(1 - R_C)(V + U)^+ + s_F(V + U)^+, \\ U(T, S) = 0. \end{cases} \tag{5}$$

Note that, in (5), V is considered known. It can be computed by the Black-Scholes formula, or numerically. Note also that $\hat{V}$ in (4) and U in (5) are unknown, and, since these PDEs involve unknowns in the $\max$ or $\min$ notation, they are nonlinear.

3 Numerical Methods

In order to make PDE (5) convenient for the application of numerical PDE methods, we apply the variable transformation $\tau = T - t$ in the time dimension. Then (5) becomes the initial value problem

$$\begin{cases} \frac{\partial U}{\partial \tau} = \mathcal{L}U + f(U, V), \\ U(0, S) = 0 \end{cases} \tag{6}$$

where

$$f(U, V) \equiv -(\lambda_B(1 - R_B)(U + V)^- + \lambda_C(1 - R_C)(U + V)^+ + s_F(U + V)^+). \tag{7}$$

3.1 Discretization

In this subsection, we present the discretization of (6). The semi-infinite space domain of spot price S is truncated into $[0, S_{max}]$, for sufficiently large S_{max}. Then, $[0, S_{max}]$ is divided into N sub-intervals, with the gridpoints positioned uniformly or non-uniformly. Standard second-order finite differences are used for the space discretization of (6).

We assume that the timestepping is handled by the θ-method, which, for $\theta = \frac{1}{2}$ and $\theta = 1$ becomes the Crank-Nicolson (CN) and Backward Euler methods, respectively.

3.2 Boundary conditions

As we indicated before, we need to truncate the semi-infinite domain $[0, +\infty)$ into $[0, S_{max}]$. We also need to formulate boundary conditions when $S = 0$ and $S = S_{max}$. To obtain the left boundary condition, we substitute $S = 0$ into (6) and get

$$\frac{\partial U}{\partial \tau} = -rU + f(U, V) \tag{8}$$

which can be seen as a first-order ordinary differential equation (ODE), computing approximate values of $U(\tau, 0)$ used as Dirichlet conditions.

Regarding the far-side boundary condition, when S_{max} is large enough, the behavior of derivative price $\hat{V}$ after value adjustment is expected to be similar to the Black-Scholes derivative price V. A choice of far-side boundary condition often used in Black-Scholes PDE is the *linear boundary condition*

$$\lim_{S \to \infty} \frac{\partial^2 U}{\partial S^2} = 0. \tag{9}$$

If we apply the far-side boundary condition (9) directly to (6) as in [24], the PDE at $S = S_{max}$ becomes

$$\frac{\partial U}{\partial \tau} = (q - \gamma)S\frac{\partial U}{\partial S} - rU + f(U, V), \tag{10}$$

where the first derivative term can be discretized by first-order or second-order one-sided difference scheme. The implementation of the linear boundary condition by discretizing (10) applies to all derivatives considered in this paper.

Alternatively, we can implement the linear boundary condition (9) by considering the PDE problem

$$\begin{cases} \frac{\partial \hat{V}}{\partial \tau} = \mathcal{L}\hat{V} - \lambda_B(1 - R_B)\hat{V}^- - \lambda_C(1 - R_C)\hat{V}^+ - s_F\hat{V}^+, \\ \hat{V}(0, S) = H(S). \end{cases} \tag{11}$$

arising from (4) with $\tau = T - t$. As explained again in [24], we assume that, close to S_{max} and beyond, we have

$$\hat{V}(\tau, S) = \alpha(\tau)S + \beta(\tau). \tag{12}$$

For some financial derivatives, taking into account the asymptotic behaviour of $\hat{V}$ as $S \to S_{max}$, allows us to know the sign of $\hat{V}(\tau, S_{max})$, which simplifies the right hand side $-\lambda_B(1 - R_B)\hat{V}^- - \lambda_C(1 - R_C)\hat{V}^+ - s_F\hat{V}^+$ to either the positive or the negative terms. Substituting $\hat{V}$ of (12) into the PDE (11) with the simplified right hand side, results in two ODEs for $\alpha(\tau)$ and $\beta(\tau)$, respectively, which can be solved to give rise to Dirichlet boundary conditions for $\hat{V}(\tau, S_{max})$, and thus for $U(\tau, S_{max})$. Clearly, the boundary conditions depend on the type of financial derivative studied. More specifically, for a European Call or Long Forward, we obtain

$$U(\tau, S_{max}) = (e^{(-\lambda_C(1-R_C)-s_F)\tau} - 1)(e^{(q-\gamma-r)\tau}S_{max} - e^{-r\tau}K), \tag{13}$$

while for a European Put and any financial derivative whose value decays to 0 as $S \to S_{max}$, we obtain

$$U(\tau, S_{max}) = 0. \tag{14}$$

3.3 Iteration methods for nonlinear PDE

In this subsection, we introduce two iteration methods, with emphasis on the second one, to handle the nonlinearity in PDE (6).

Let $\tau_j, j = 0, \ldots, N_t$, be the timesteps at which the solution is computed, with $\tau_0 = 0 < \tau_1 < \cdots < \tau_{N_t} = T$, and $\Delta\tau^j = \tau_j - \tau_{j-1}$ be the jth time stepsize. Let $u^j, j = 0, \ldots, N_t$, denote the computed solution vector arising from the approximate values of U at the spatial gridpoints at time τ_j, with u^0 being the initial condition vector. Since we use an iteration method to handle the nonlinearity, let $u^{j,k}, k = 0, \ldots, maxit$, denote the computed solution vector at iteration k of timestep j, with $maxit$ the maximum number of iterations allowed. Let $v^j, j = 0, \ldots, N_t$, be the vector of values of V at the spatial gridpoints, and at timestep j. Note that the values of V can be computed using the Black-Scholes formula directly. For generic vectors u and v arising from (approximate) values of U and V, respectively, at the spatial gridpoints, let $f(u, v)$ denote the vector arising from evaluating f at the components of u and v. Let also A be the matrix arising from the space discretization of $\mathcal{L}U$, and $\mathbb{I}$ be the identity matrix of same order. For simplicity, we assume the spatial gridpoints remain the same at all timesteps.

The first iteration method is a fixed-point iteration method similar to the one in [1]. We remark that the fixed-point iteration method in [1] assumes that the timestepping is handled by the method of

characteristics and the space discretization by a finite element method. In our paper, we consider CN timestepping and centered differences in space. Thus, at each timestep we need to solve

$$(\mathbb{I} - \theta\Delta\tau^j A)u^j = (\mathbb{I} + (1-\theta)\Delta\tau^j A)u^{j-1} + \theta\Delta\tau^j f(u^j, v^j) + (1-\theta)\Delta\tau^j f(u^{j-1}, v^{j-1}). \tag{15}$$

Then the fixed-point iteration method at time τ^j can be described as Algorithm 1.

Algorithm 1 Fixed-point iteration for (6) at step j, with θ-timestepping

Require: Solve $(\mathbb{I} - \theta\Delta\tau^j A)u^j = g^j + \theta\Delta\tau^j f(u^j, v^j)$
where $g^j = (\mathbb{I} + (1-\theta)\Delta\tau^j A)u^{j-1} + (1-\theta)\Delta\tau^j f(u^{j-1}, v^{j-1})$
1: Initialize $u^{j,0} = u^{j-1}$
2: **for** $k = 1, \ldots, maxit$ **do**
3: Solve $(\mathbb{I} - \theta\Delta\tau^j A)u^{j,k} = g^j + \theta\Delta\tau^j f(u^{j,k-1}, v^j)$
4: **if** stopping criterion satisfied **then**
5: Break
6: **end if**
7: **end for**
8: Set $u^j = u^{j,k}$

The stopping criterion in Algorithm 1, motivated by [15], is

$$\max_i \frac{|u_i^{j,k} - u_i^{j,k-1}|}{\max(1, |u_i^{j,k}|)} \leq tol. \tag{16}$$

where tol is a user-chosen tolerance.

The above iteration method is considered to be *explicit*, in the sense that the matrix solved at each iteration is the same and only the right-hand side vector changes.

Regarding the convergence of the fixed-point iteration, we can write it as $u^{j,k} = \mathbf{G}(u^{j,k-1})$, where $\mathbf{G}(u^{j,k-1}) = (\mathbb{I} - \theta\Delta\tau^j A)^{-1}(g^j + \theta\Delta\tau^j f(u^{j,k-1}, v^j))$, and show that $\mathbf{G}$ is a contraction, for sufficiently small $\Delta\tau^j$. We do not go into details, as the focus of this paper is not to study the convergence of the fixed-point iteration.

We next present another iteration method for handing the nonlinearity in (6). We refer to it as *discrete penalty-like iteration*, or, simply, *penalty iteration*, as it is motivated by the same-name method in [15], designed to handle the nonlinear PDE arising from the linear complementarity problem (LCP) in American option pricing. For this reason, with generic vectors u and v, we define the diagonal penalty matrix $P = P(u, v)$ by

$$[P(u,v)]_{ii} \equiv \begin{cases} -\lambda_B(1 - R_B) & \text{if } u_i + v_i < 0 \\ -\lambda_C(1 - R_C) - s_F & \text{if } u_i + v_i \geq 0 \end{cases} \tag{17}$$

Note that, with the above matrix P, the vector arising from the discretized form of $f(U, V)$ can be written as

$$f(u, v) = P(u, v)(u + v) \tag{18}$$

where, since $P(u, v)$ depends on u, there is nonlinearity between $P(u, v)$ and u. Note also that, if $\lambda_B \geq 0$, $\lambda_C \geq 0$, and $s_F \geq 0$, we have $P_{i,j}(u, v) \leq 0$. With the help of the matrix P, we can also write (15) as

$$(\mathbb{I} - \theta\Delta\tau^j(A + P(u^j, v^j)))u^j = (\mathbb{I} + (1-\theta)\Delta\tau^j A)u^{j-1} + \theta\Delta\tau^j P(u^j, v^j)v^j + (1-\theta)\Delta\tau^j f(u^{j-1}, v^{j-1}). \tag{19}$$

We can view the penalty iteration method as generalized Newton's method applied to the nonsmooth nonlinear problem (15) with the nonlinear term given by (18), and the generalized Jacobian defined by

$$\frac{\partial [f(u,v)]_i}{\partial u_j} \equiv \begin{cases} [P(u,v)]_{ii} & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

Let now $P^k = P(u^{j,k}, v^j)$, where for brevity, in the notation P^k, we omitted the superscript j. Using the penalty matrix P^k, the proposed discrete penalty iteration for (6) is described in Algorithm 2.

Algorithm 2 Discrete penalty-like iteration for (6) at step j, with θ-timestepping

Require: Solve $(\mathbb{I} - \theta\Delta\tau^j(A + P(u^j, v^j)))u^j = g^j + \theta\Delta\tau^j P(u^j, v^j)v^j$
where $g^j = (\mathbb{I} + (1-\theta)\Delta\tau^j A)u^{j-1} + (1-\theta)\Delta\tau^j f(u^{j-1}, v^{j-1})$

1: Initialize $u^{j,0} = u^{j-1}$ and $P^0 = P(u^{j,0}, v^j)$
2: **for** $k = 1, \ldots, maxit$ **do**
3: Solve $(\mathbb{I} - \theta\Delta\tau^j(A + P^{k-1}))u^{j,k} = g^j + \theta\Delta\tau^j P^{k-1}v^j$
4: Compute $P^k = P(u^{j,k}, v^j)$
5: **if** stopping criterion satisfied **then**
6: Break
7: **end if**
8: **end for**
9: Set $u^j = u^{j,k}$

The stopping criterion in Algorithm 2, motivated again by [15] is

$$(P^k = P^{k-1}) \text{ or } (\max_i \frac{|u_i^{j,k} - u_i^{j,k-1}|}{\max(1, |u_i^{j,k}|)} \leq tol). \tag{21}$$

Note that the second iteration method is considered to be *implicit*, in the sense that the matrix solved at each iteration depends on the iteration index k. However, since the matrix solved at each iteration is adjusted by only a diagonal matrix, the sparsity structure of the matrix remains the same, which is also the same structure of the matrix corresponding to a linear PDE with same differential operator as in (6). Note also that, if $\lambda_B \geq 0$, $\lambda_C \geq 0$, the matrix P enhances the diagonal dominance of A.

In order to study the convergence of the discrete penalty-like iteration, we make use of a monotonicity argument for the matrix A. In [12], sufficient conditions are derived under which the matrix arising from the discretization of $\mathcal{L}$ by finite differences and with Dirichlet boundary conditions on both ends is strictly diagonally dominant with negative diagonal entries and non-negative off-diagonal entries (Lemma 4.1 in [12]), therefore, monotone. To facilitate the monotonicity argument for the proof of Theorem 1, we consider that A is formed with Dirichlet conditions at both ends, such as those described in Section 3.2. We also assume that the conditions of Lemma 4.1 in [12] hold and, therefore, A is non-singular, monotone and an M-matrix. It is easy to see that, under the same conditions, $(\mathbb{I} - \theta\Delta\tau^j(A + P^{k-1}))$ is also non-singular, monotone and an M-matrix.

THEOREM 1 *(Convergence of discrete penalty-like iteration). Under the assumption that the matrix of the linear system at each iteration in Algorithm 2, i.e.* $(\mathbb{I} - \theta\Delta\tau^j(A + P^{k-1}))$*, is monotone, then*
(i) The discrete penalty-like iteration converges to the unique solution.

(ii) The iteration converges monotonically.
(iii) The iteration has finite termination.

Proof. See Appendix A.

REMARK 1 *The PDE problem (6), with the nonlinear term in (7), can be written as a control problem*

$$\frac{\partial U}{\partial \tau} = \mathcal{L}U + \max_{\theta_1 \in \{0,1\}} \min_{\theta_2 \in \{0,1\}} \left[-\theta_1 \lambda_B (1 - R_B)(U + V) - \theta_2 (\lambda_C (1 - R_C) + s_F)(U + V) \right], \tag{22}$$

where θ_1 and θ_2 are two controls corresponding to the cases that $U+V$ is negative or positive, respectively. Since the problem involves both max and min, it can be viewed as a Hamilton-Jacobi-Bellman-Isaacs (HJBI) problem. For such problems, in general, the Newton-type iteration scheme is not guaranteed to converge, unless the initial guess is close enough to the solution, or a more complex treatment is considered [14, 20]. However, in our case, the control problem (22) is such that it can be reduced to a single control with only max or min terms. More specifically, if $-\lambda_C(1 - R_C) - s_F \leq -\lambda_B(1 - R_B) \leq 0$, problem (22) can be written as

$$\frac{\partial U}{\partial \tau} = \mathcal{L}U + \min_{\theta \in \{0,1\}} \left[-(1 - \theta)\lambda_B (1 - R_B)(U + V) - \theta (\lambda_C (1 - R_C) + s_F)(U + V) \right] \tag{23}$$

while, if $-\lambda_B(1 - R_B) < -\lambda_C(1 - R_C) - s_F \leq 0$, problem (22) can be written as

$$\frac{\partial U}{\partial \tau} = \mathcal{L}U + \max_{\theta \in \{0,1\}} \left[-(1 - \theta)\lambda_B (1 - R_B)(U + V) - \theta (\lambda_C (1 - R_C) + s_F)(U + V) \right]. \tag{24}$$

both of which can be viewed as Hamilton-Jacobi-Bellman (HJB) problems. For problems (23) and (24), the monotonicity argument can be used to show global convergence [15, 14, 20].

4 XVA in American Derivatives

Just as the formulation of the American derivative pricing problem can be presented as a linear complementarity problem (LCP) [15, 23], the formulation of the XVA pricing problem in American derivatives can also be presented as an LCP [2].

We consider the case $M = \hat{V}$, and, as in [2], build the formulation of XVA pricing in American derivatives starting from Equation (5) and adding appropriate constraints that force the adjusted price $\hat{V}(t, S)$ to be above the pay-off $V^*(t, S)$.

Then, the American derivative price $\hat{V}$, if taking risk into account, satisfies

$$\left\{ \begin{array}{r} \frac{\partial \hat{V}}{\partial t} + \mathcal{L}\hat{V} + f(\hat{V}) = 0 \\ \hat{V} - V^* \geq 0 \end{array} \right\} \vee \left\{ \begin{array}{r} \frac{\partial \hat{V}}{\partial t} + \mathcal{L}\hat{V} + f(\hat{V}) \geq 0 \\ \hat{V} - V^* = 0 \end{array} \right\} \tag{25}$$

where

$$f(\hat{V}) = -(\lambda_B (1 - R_B)\hat{V}^- + \lambda_C (1 - R_C)\hat{V}^+ + s_F \hat{V}^+) \tag{26}$$

and the notation $\vee$ means "or". Typically at each time t, there is a particular asset price S^* which divides the asset price domain into two regions: one side that corresponds to the left condition of (25), where it

is suggested to hold the option, and another side that corresponds to the right condition of (25), where it is suggested to early exercise the option. This particular value of S^* is unknown before the differential equation is solved, and is usually called *free boundary* of LCP.
Note that the value $V(t,S)$ of American derivatives without valuation adjustment satisfies the LCP [15]

$$\left\{ \begin{array}{c} \frac{\partial V}{\partial t} + \mathcal{L}V = 0 \\ V - V^* \geq 0 \end{array} \right\} \vee \left\{ \begin{array}{c} \frac{\partial V}{\partial t} + \mathcal{L}V \geq 0 \\ V - V^* = 0 \end{array} \right\} \tag{27}$$

Furthermore, note that equations (25) and (27) have the same terminal condition $\hat{V}(T,S) = V(T,S) = H(S)$ where $H(S)$ is pay-off function at maturity.

The LCP problem (27) also has a free boundary. However it is not guaranteed that the two *free boundaries* of equations (25) and (27) are the same. Hence, it is impossible to find an individual PDE for the XVA value $U(t,S)$. In this section, we mainly focus on numerically solving (25).

4.1 Reformulation to penalty form

Forsyth and Vetzal [15] proposed the discrete penalty method to numerically solve the LCP arising from one-asset American option without XVA. They replace (27) by a nonlinear PDE via adding a large positive penalty term to the right hand side of the Black-Scholes equation.

In the American XVA problem, a similar penalty term can be added. If backward time $\tau = T - t$ is applied, the penalty form of (25) is written as

$$\begin{cases} \frac{\partial \hat{V}}{\partial \tau} = \mathcal{L}\hat{V} + f(\hat{V}) + p\max(V^* - \hat{V}, 0), \\ \hat{V}(0,S) = H(S), \end{cases} \tag{28}$$

where p is a large positive penalty factor. The penalty term forces the solution of (25) to approximately satisfy the obstacle condition $\hat{V} - V^* \geq 0$.
As in Section 3.2, the boundary conditions for $S = 0$ are formed by solving

$$\frac{\partial \hat{V}}{\partial \tau} = -r\hat{V} + f(\hat{V}) + p\max(V^* - \hat{V}, 0), \tag{29}$$

while for $S = S_{max}$, the linear boundary condition

$$\lim_{S \to \infty} \frac{\partial^2 \hat{V}}{\partial S^2} = 0,$$

is implemented either by discretizing

$$\frac{\partial \hat{V}}{\partial \tau} = (q - \gamma)S\frac{\partial \hat{V}}{\partial S} - r\hat{V} + f(\hat{V}) + p\max(V^* - \hat{V}, 0), \tag{30}$$

or by the alternative that gives rise to Dirichlet conditions that depend on the type of financial derivative considered. For an American Call or Long Forward, we obtain

$$\hat{V}(\tau, S_{max}) = \max\{e^{(-\lambda_C(1-R_C)-s_F)\tau}(e^{(q-\gamma-r)\tau}S_{max} - e^{-r\tau}K), S_{max} - K\}, \tag{31}$$

while, for an American Put,

$$\hat{V}(\tau, S_{max}) = 0. \tag{32}$$

4.2 Double-penalty iteration method

Penalty term discretization

The timestepping method and the space discretization for $\mathcal{L}\hat{V}$ in (28) are similar to those for the European case. We now discuss the treatment of the penalty term $p\max(V^* - \hat{V}, 0)$ and the nonlinear term $f(\hat{V})$. Let $\hat{v}^j, j = 0, \ldots, N_t$, be the computed solution vector arising from the approximate values of $\hat{V}$ at the spatial gridpoints at time τ_j. Also let v^* be the vector of values of pay-off V^* at the spatial gridpoints.

When computing the numerical solution $\hat{v}^j$ at step j, the penalty term $p\max(V^* - \hat{V}, 0)$ (arising from the American feature) is discretized as $P_A(\hat{v}^j)(v^* - \hat{v}^j)$, where $P_A(\hat{v}^j)$ is a diagonal matrix defined by

$$[P_A(\hat{v}^j)]_{ii} \equiv \begin{cases} p & \text{if } (\hat{v}^j)_i < (v^*)_i, \\ 0 & \text{otherwise.} \end{cases} \tag{33}$$

The nonlinear term $f(\hat{V})$ (arising from the XVA) is discretized as $P_X(\hat{v}^j)(\hat{v}^j)$, where $P_X(\hat{v}^j)$ is also a diagonal matrix defined by

$$[P_X(\hat{v}^j)]_{ii} \equiv \begin{cases} -\lambda_B(1 - R_B) & \text{if } (\hat{v}^j)_i < 0, \\ -\lambda_C(1 - R_C) - s_F & \text{if } (\hat{v}^j)_i \geq 0. \end{cases} \tag{34}$$

Therefore, to compute $\hat{v}^j$, given $\hat{v}^{j-1}$, the following system of algebraic equations needs to be solved:

$$\begin{aligned}[\mathbb{I} - \theta\Delta\tau^j(A + P_X(\hat{v}^j))]\hat{v}^j + P_A(\hat{v}^j)\hat{v}^j \\ = (\mathbb{I} + (1-\theta)\Delta\tau^j A)\hat{v}^{j-1} + (1-\theta)\Delta\tau^j P_X(\hat{v}^{j-1})\hat{v}^{j-1} + P_A(\hat{v}^j)v^*. \end{aligned} \tag{35}$$

Note that there are two sources of nonlinearity, namely $P_A(\hat{v}^j)$ and $P_X(\hat{v}^j)$, with respect to $\hat{v}^j$. Algorithm 2 and the penalty iteration method to value American option [15] are combined into one iteration method, which can be viewed as generalized Newton's method, to solve (35). In the construction of P_A, the large penalty factor p is chosen as

$$p = \frac{1}{tol}$$

where tol is the tolerance for the stopping criterion of the iteration method.

Double-penalty iteration

If we view the nonlinear term $f(\hat{V})$ or its discrete form $P_X(\hat{v}^j)(\hat{v}^j)$ as a second penalty term, this generalized Newton's iteration method to solve (35) at each timestep can be called *discrete double-penalty iteration* for American XVA pricing.

The stopping criterion in Algorithm 3, modified from (21), is

$$\big[(P_A^k = P_A^{k-1}) \text{ and } (P_X^k = P_X^{k-1})\big] \text{ or } \big[\max_i \frac{|\hat{v}_i^{j,k} - \hat{v}_i^{j,k-1}|}{\max(1, |\hat{v}_i^{j,k}|)} \leq tol\big]. \tag{36}$$

Since the matrix solved at each iteration is adjusted by only two diagonal matrices, the sparsity structure of the matrix remains the same. Also since the penalty parameter p is positive, the matrix P_A enhances the diagonal dominance of $\mathbb{I} - \theta\Delta\tau^j(A + P_X^{k-1})$. It is easy to see that the matrix $[\mathbb{I} - \theta\Delta\tau^j(A + P_X^{k-1}) + P_A^{k-1}]$ is also non-singular, monotone and an M-matrix, under the conditions of Lemma 4.1 in [12], assuming Dirichlet conditions are used.

Algorithm 3 Discrete double-penalty iteration for (28) at step j, with θ-timestepping

Require: Solve $[\mathbb{I} - \theta\Delta\tau^j(A + P_X(\hat{v}^j))]\hat{v}^j + P_A(\hat{v}^j)\hat{v}^j = g^j + P_A(\hat{v}^j)v^*$
where $g^j = (\mathbb{I} + (1-\theta)\Delta\tau^j A)\hat{v}^{j-1} + (1-\theta)\Delta\tau^j P_X(\hat{v}^{j-1})\hat{v}^{j-1}$.
Initialize $\hat{v}^{j,0} = \hat{v}^{j-1}$, $P_A^0 = P_A(\hat{v}^{j,0})$ and $P_X^0 = P_X(\hat{v}^{j,0})$
for $k = 1, \ldots, maxit$ **do**
  Solve $[\mathbb{I} - \theta\Delta\tau^j(A + P_X^{k-1}) + P_A^{k-1}]\hat{v}^{j,k} = g^j + P_A^{k-1}v^*$
  Compute $P_A^k = P_A(\hat{v}^{j,k})$ and $P_X^k = P_X(\hat{v}^{j,k})$
  if stopping criterion satisfied **then**
    Break
  end if
end for
Set $\hat{v}^j = \hat{v}^{j,k}$

REMARK 2 *The penalty form of the American XVA problem (28) can also be written in a control problem form as,*

$$\frac{\partial \hat{V}}{\partial \tau} = \mathcal{L}\hat{V} + \max_{\theta_1,\theta_3\in\{0,1\}} \min_{\theta_2\in\{0,1\}} \left[-\theta_1\lambda_B(1-R_B)\hat{V} - \theta_2(\lambda_C(1-R_C)+s_F)\hat{V} + \theta_3 p(V^* - \hat{V})\right], \quad (37)$$

which involves three controls and both max and min.

When $-\lambda_C(1-R_C) - s_F \le -\lambda_B(1-R_B) \le 0$, problem (37) cannot be reduced into a single max or min control problem, and the monotonicity argument fails in the convergence study. Note that θ_1 and θ_2 depend on the same value $\hat{V}$, but θ_3 depends on another, namely $(V^* - \hat{V})$, arising from American constraints.

When $-\lambda_B(1-R_B) < -\lambda_C(1-R_C) - s_F \le 0$, which is a less frequent and less interesting case, problem (37) will end up with a single max

$$\frac{\partial \hat{V}}{\partial \tau} = \mathcal{L}\hat{V} + \max_{\theta_1,\theta_3\in\{0,1\}} \left[-(1-\theta_1)\lambda_B(1-R_B)\hat{V} - \theta_1(\lambda_C(1-R_C)+s_F)\hat{V} + \theta_3 p(V^* - \hat{V})\right]. \quad (38)$$

To address the convergence problem in HJBI problem, the authors in [20], with appropriate assumptions and more complex arguments, show global convergence for a certain class of HJBI boundary value problems. In [14], other treatments, e.g. a relaxation scheme, or a piecewise constant policy timestepping, are proposed that guarantee global convergence. For our double-penalty method, we did not show global convergence, but, as demonstrated in the numerical experiments, the algorithm exhibits fast convergence. It is worth noting, that, in an initial value problem, a close enough to the solution initial guess is relatively easy to obtain for each timestep.

5 Numerical Results

We present numerical results from applying Algorithms 1 and 2 for the pricing of XVA in European Put, Call and Long Forward, as well as Algorithm 3 for the pricing of American Put, Call and Long Forward including XVA.

5.1 Examples of XVA in European derivatives

Table 1 presents the values of parameters used in the experiments in this section. These parameters have the same values as those used in the examples in [1, 9], except the S_{max} value, which, in [1, 9], is $4K$. A discussion on localization is found later in the section.

Parameter	Value
Domain of S	$[0, 12K]$
Strike Price, K	15
Time to maturity, T	5
Volatility, σ	0.25
Repo rate minus dividend, $q - \gamma$	0.015
Interest rate, r	0.03
Default intensity of party B, λ_B	0.02
Default intensity of party C, λ_C	0.05
Recovery rate of party B, R_B	0.4
Recovery rate of party C, R_C	0.4
Funding spread, s_F	$(1 - R_B)\lambda_B$

Table 1: Model parameters for bilateral XVA in European derivatives.

We discretize the spatial domain into N subintervals, so that the (nonuniform) gridpoints are concentrated around the strike K. We discuss later how we generate nonuniform gridpoints and what the effect of nonuniform grids is compared to uniform ones. The spatial derivatives are discretized by standard second-order centered differences, except the first derivative in the far-side condition (10), which is discretized by backward differences. It is worth mentioning that the implementation of the linear boundary condition through (10) is simple and does not require explicit knowledge of the sign of the solution at S_{max}, so we obtained all the displayed results using (10). However, we also experimented with the alternative implementation of the linear boundary condition, which gives rise to Dirichlet conditions (13) or (14), and the numerical results obtained were identical across all points (within about 7 digits of precision) with those obtained by discretizing (10). The number of timesteps is denoted by N_t, and $\Delta\tau = T/N_t$. Note that Rannacher smoothing is not needed (and therefore not applied) for the numerical solution of (5), since the initial condition is smooth.

In all tables below, "iter tot 1" and "iter avg 1" are total and average (per timestep) number of iterations for Algorithm 1, and "iter tot 2" and "iter avg 2" are total and average (per timestep) number of iterations for Algorithm 2. The tolerance for both iteration schemes was set to 10^{-7}.

5.1.1 Call and put options

We present results from pricing the XVA of European options with the parameter settings in Table 1. For European call and put options, we have an exact solution formula as given in [9]

$$U(\tau, S) = -(1 - \exp(-((1 - R_B)\lambda_B + (1 - R_C)\lambda_C)\tau))V(\tau, S), \tag{39}$$

where the $V(\tau, S)$ can be calculated in closed-form. Thus we can calculate exact errors and the associated convergence orders.

In Tables 2 and 3, we report, for several discretization sizes, the discrete L^∞-norm error over all gridpoints at time $\tau = T$ ($t = 0$), the associated convergence orders, and the total and average per

N	N_t	error in U	order	iter tot 1	iter avg 1	iter tot 2	iter avg 2
50	100	1.41e-03	–	300	3.00	119	1.19
100	200	3.54e-04	2.00	600	3.00	242	1.21
200	400	8.86e-05	2.00	1200	3.00	415	1.04
400	800	2.21e-05	2.00	2400	3.00	814	1.02
800	1600	5.54e-06	2.00	3934	2.46	1614	1.01

Table 2: Results from solving (6) for a European Put using Algorithms 1 and 2 with the parameters in Table 1. Nonuniform grids are used.

N	N_t	error in U	order	iter tot 1	iter avg 1	iter tot 2	iter avg 2
50	100	1.41e-03	–	400	4.00	105	1.05
100	200	3.54e-04	2.00	600	3.00	214	1.07
200	400	8.86e-05	2.00	1200	3.00	430	1.08
400	800	2.22e-05	2.00	2400	3.00	842	1.05
800	1600	5.54e-06	2.00	4800	3.00	1635	1.02

Table 3: Results from solving (6) for a European Call using Algorithms 1 and 2 with the parameters in Table 1. Nonuniform grids are used.

timestep number of iterations. We notice that the average number of penalty-like iterations is very close to 1, irrespectively of the grid size, while the fixed-point iteration algorithm needs 2 to 3 average number of iterations to reach the same tolerance, where, again, the number of iteration seems independent of grid size. Note that the number of penalty iterations is close to optimal. Essentially, in each of the first few timesteps two or three iterations are needed, and in each of the remaining timesteps just one iteration. We also mention that the two iteration schemes produced exactly the same errors for each of the discretization sizes, therefore, we only report one column of errors. The order of convergence is stably 2.

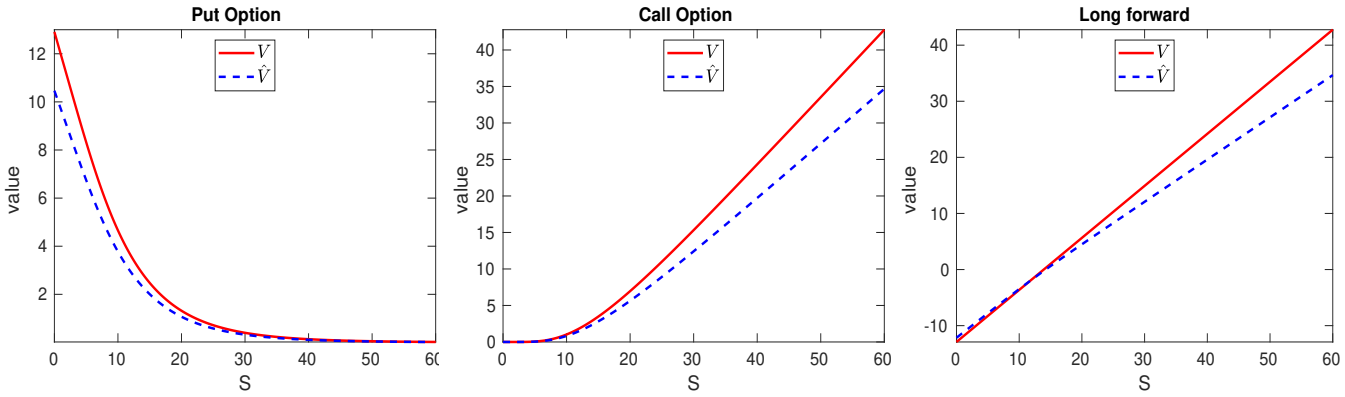

Figure 1: A visualization of various financial derivative values with ($\hat{V}$) or without (V) XVA with the parameters in Table 1. The difference between these two curves is XVA.

In Figure 1, left and centre plots, we visualize V and $\hat{V} = V + U$ for European call and put options. For these options, the XVA is negative and V is above $\hat{V}$.

Discussion on localization

The choice of S_{max} is crucial for an accurate solution. In [21], for European vanilla option, the far-side boundary is chosen so that

$$S_{max} > K \exp(\sqrt{2\sigma^2 T |\ln(tol)|}), \tag{40}$$

where tol is the computing tolerance. An alternative far-side boundary is suggested in [24]

$$S_{max} > K \exp((r - q - \frac{\sigma^2}{2})T + \sigma\nu\sqrt{T}), \tag{41}$$

where $\nu = 3$ is the usual choice for a European option to have small error around $S = K$. If smaller error along the whole price domain is required, ν can be set as 5. In our experiments, we have used $S_{max} = 12K$ for the 5-yr long European options, which seems to agree more with (41), with $\nu = 4.6$. Our experiments indicate that this choice of S_{max} balances the domain truncation and discretization errors for the finest grid resolution.

To show the sensitivity of the accuracy of the solution to the choice of S_{max}, in Table 4, we present results from the XVA valuation of a European call option with the parameters of Table 1, except that we set $S_{max} = 10K$ in one experiment and $S_{max} = 8K$ in another. We only report results from Algorithm 2, as the focus of this discussion is on the error of the computed approximations. These results should be compared to those in Table 3, which has $S_{max} = 12K$.

N	N_t	error in U	order	iter tot 2	iter avg 2
$S_{max} = 10K$					
50	100	1.19e-03	–	106	1.06
100	200	2.98e-04	2.00	217	1.09
200	400	7.45e-05	2.00	431	1.08
400	800	1.86e-05	2.00	842	1.05
800	1600	9.84e-06	0.92	1631	1.02
$S_{max} = 8K$					
50	100	9.30e-04	–	106	1.06
100	200	2.34e-04	1.99	219	1.10
200	400	5.85e-05	2.00	442	1.11
400	800	5.24e-05	0.16	846	1.06
800	1600	5.26e-05	-0.00	1635	1.02

Table 4: Results from solving (6) for a European Call using Algorithm 2 with the parameters in Table 1, except that S_{max} is as indicated. Nonuniform grids are used.

In Table 4, we notice that, although the penalty iteration with different choices of S_{max} converges as fast as with $S_{max} = 12K$, the error with $S_{max} = 10K$ and $N = 800$ does not exhibit second order convergence. This phenomenon is more obvious for $S_{max} = 8K$ and $N \geq 400$. It is worth noting, that the errors for $S_{max} = 10K$ are smaller than the respective errors for $S_{max} = 12K$. This is because the localization error for $S_{max} = 10K$ is smaller than the discretization one, and the latter one is smaller than the discretization error for $S_{max} = 12K$.

Discussion on nonuniform space discretizations

In the derivative pricing problem, nonuniform discretization, with denser points around the strike, is often preferred, as the area around the strike exhibits more nonlinearity, and is more "interesting" to

practitioners.

Smooth mappings of equally spaced gridpoints are often used to generate appropriate nonuniform discretizations. Suppose the domain is $[0, S_{max}]$ and it is equally divided into N subintervals, with x_i being the gridpoints, i.e. $x_i = ih, h = \frac{S_{max}}{N}$, where $i = 0, 1, \ldots, N$. The smooth mapping $w(x)$ generates the nonuniform grid as

$$S_i \equiv w(x_i) = (1 + \frac{\sinh(\beta - (x_i/x_N - \alpha))}{\sinh(\beta\alpha)})K \tag{42}$$

where K is the strike. This mapping produces denser gridpoints around K. Larger parameter α increases the density of the points. The purpose of parameter β is to ensure the last gridpoint is S_{max}. In practice, α is set around 0.4, so that the strike is aligned with a gridpoint.

In Tables 5 and 6, we report, for several discretization sizes, the discrete L^∞-norm error over all gridpoints at time $\tau = T$ ($t = 0$), the associated convergence orders, and the total and average per timestep number of iterations, if uniform discretization is used. Compared to the results with nonuniform grid as shown in Tables 2 and 3, it is obvious that, if the same number of grid points and time steps are used (same computation cost), finer gridpoints around the strike result in smaller maximum error. It is worth noting that all these methods have second order convergence, as expected.

N	N_t	error in U	order	iter tot 2	iter avg 2
50	100	2.64e-03	–	135	1.35
100	200	6.33e-04	2.06	250	1.25
200	400	1.57e-04	2.01	427	1.08
400	800	3.95e-06	1.99	822	1.03
800	1600	9.87e-06	2.00	1614	1.01

Table 5: Results from solving (6) for a European Put using Algorithm 2 with the parameters in Table 1. Uniform discretization is used.

N	N_t	error in U	order	iter tot 2	iter avg 2
50	100	2.64e-03	–	104	1.04
100	200	6.33e-04	2.06	208	1.04
200	400	1.57e-04	2.01	420	1.05
400	800	3.95e-05	1.99	846	1.06
800	1600	9.88e-06	2.00	1665	1.04

Table 6: Results from solving (6) for a European Call using Algorithm 2 with the parameters in Table 1. Uniform discretization is used.

5.1.2 Forward contract

In this example, we present results from pricing the XVA of a 5-year long forward contract with the parameters in Table 1. For this problem, we don't have an exact solution to calculate exact errors and corresponding convergence orders. Thus the errors at one resolution are estimated from the difference to the previous (coarser) resolution.

In Table 7, we report, for several discretization sizes, the differences on the coarser gridpoints at time $\tau = T$ ($t = 0$), the associated convergence orders, and the total and average per timestep number of iterations. We observe similar performance in convergence and number of iterations as in the call and put

option examples. The average number of iterations with Algorithm 2 is still very close to 1, independently of the grid size. The average number of iterations with Algorithm 1 is about 2 to 3, independently of the grid size as well. The order of convergence is stably 2.

N	N_t	diff in U	order	iter tot 1	iter avg 1	iter tot 2	iter avg 2
50	100	–	–	400	4.00	102	1.02
100	200	7.58e-04	–	600	3.00	206	1.03
200	400	1.90e-04	2.00	1200	3.00	412	1.03
400	800	4.76e-05	2.00	2400	3.00	821	1.03
800	1600	1.19e-05	2.00	4800	3.00	1644	1.03

Table 7: Results from solving (6) for a long forward contract using Algorithms 1 and 2 with the parameters in Table 1. Nonuniform grids are used.

The right part of Figure 1 shows the long forward contract value with and without XVA. The difference between the two curves is XVA. In this figure, the two curves cross each other. To the left of the crossing point, the value with XVA is higher than the value without XVA, which means that, when the contract is made, party B needs to pay the XVA to party C. On the contrary, to the right of the crossing point, where the value with XVA is less than the value without XVA, the party B needs to charge the XVA to party C.

5.2 Examples of XVA in American derivatives

We consider the pricing of American Put, Call and Long Forward including XVA, with the parameter settings in Table 8, and the Mark-to-Market value $M = \hat{V}$. The parameter settings are the same as in [3] (except that the approach in [3] is a Monte Carlo one and there is no preset S_{max}). Tables 9 and 10 show the results from solving (28), using centered differences in space, Crank-Nicolson with Rannacher smoothing in time, and Algorithm 3 for the nonlinearity. The tolerance for Algorithm 3 was set to 10^{-7}, while the penalty parameter p was set to 10^7. We experimented with the far-side condition (30), discretized by backward differences, and with its alternative implementation which gives rise to Dirichlet conditions (31) or (32), and the numerical results obtained were identical (for the points shown and within the digits of precision displayed).

Parameter	Value
Domain of S	$[0, 10K]$
Strike Price, K	15
Time to maturity, T	0.5
Volatility, σ	0.25
Repo rate minus dividend, $q - \gamma$	0.06
Interest rate, r	0.04
Default intensity of party B, λ_B	0.04
Default intensity of party C, λ_C	0.04
Recovery rate of party B, R_B	0.3
Recovery rate of party C, R_C	0.3
Funding spread, s_F	$(1 - R_B)\lambda_B$

Table 8: Model parameters for American derivatives' pricing including bilateral XVA.

For the American derivatives valuation including XVA, the exact solution is not available. To calculate approximate errors, for the American Put case, we pick three points around or on the strike $S = K = 15$,

one at the money, one in the money, and one out of the money. For the American Call and Long Forward cases, we only show results at the money. At each point, the error at one resolution is estimated from the difference to the previous (coarser) resolution. The numerically observed orders of convergences are around two.

Looking at the number of iterations, we easily see that the average number of penalty iterations in each timestep is around 1.2, which is slightly larger than in the European case. We expect an increase in the number of iterations, since (28) has more nonlinear terms than (6). It is also important to mention that the convergence of penalty iteration for American derivatives including XVA is still independent of the discretization size.

N	N_t	iter tot	iter avg	$\hat{V}$ value for $S = 15$		
				value	diff in $\hat{V}$	order
50	42	50	1.19	0.86422480	–	–
100	82	98	1.20	0.86679812	2.57e-03	–
200	162	196	1.21	0.86751795	7.20e-04	1.84
400	322	396	1.23	0.86771336	1.95e-04	1.88
800	642	801	1.25	0.86776884	5.55e-05	1.82

N	N_t	$\hat{V}$ value for $S = 14$			$\hat{V}$ value for $S = 16$		
		value	diff in $\hat{V}$	order	value	diff in $\hat{V}$	order
50	42	1.37720716	–	–	0.51568505	–	–
100	82	1.37913317	1.93e-03	–	0.51844504	2.76e-03	–
200	162	1.37960142	4.68e-04	2.04	0.51909720	6.52e-04	2.08
400	322	1.37973315	1.32e-04	1.83	0.51928378	1.87e-04	1.81
800	642	1.37976510	3.19e-05	2.04	0.51933352	4.97e-05	1.91

Table 9: Results on three points from solving (28) for American Put option valuation including bilateral XVA using Algorithm 3 with the parameters in Table 8. Nonuniform grids are used.

N	N_t	iter tot	iter avg	$\hat{V}$ value for Call Option		
				value	diff in $\hat{V}$	order
50	42	43	1.02	1.25145331	–	–
100	82	84	1.02	1.25384939	2.40e-03	–
200	162	165	1.02	1.25445008	6.01e-04	2.00
400	322	330	1.02	1.25460036	1.50e-04	2.00
800	642	654	1.02	1.25463794	3.76e-05	2.00
N	N_t	iter tot	iter avg	$\hat{V}$ value for Long Forward		
				value	diff in $\hat{V}$	order
50	42	47	1.12	0.42849973	–	–
100	82	92	1.12	0.42848609	1.36e-05	–
200	162	183	1.13	0.42848264	3.46e-06	1.98
400	322	366	1.14	0.42848177	8.64e-07	2.00
800	642	726	1.13	0.42848156	2.16e-07	2.00

Table 10: Results from solving (28) for other American derivatives including bilateral XVA using Algorithm 3 with the parameters in Table 8 when S is at the money ($S = K = 15$). Nonuniform grids are used.

In order to compare the American and European option prices with and without XVA, we consider again the parameter settings in Table 8, and solve the respective problems with a Put initial condition. Figure 2 plots the prices. The centre plot is a zooming of the large box in left plot, while the right plot is a zooming of the small box in left plot. We observe that both V and $\hat{V}$ of American type options are above the pay-off and above V and $\hat{V}$ of European type. European V and $\hat{V}$ cross the pay-off. In both American and European type options, the values of $\hat{V}$ are lower than the values of V, as expected.

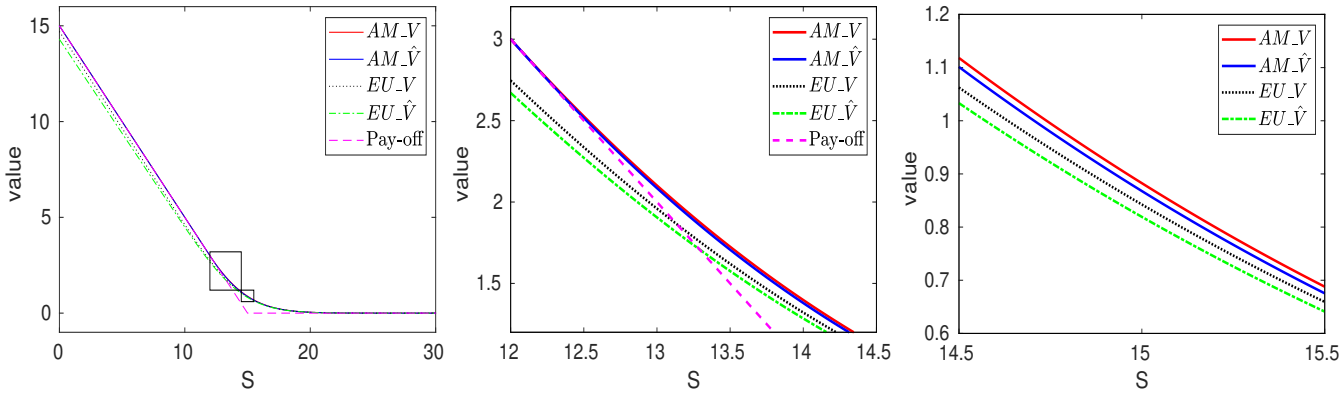

Figure 2: A visualization of European and American Put V and $\hat{V}$. The center plot is a zooming of the large box in the left plot, while the right plot is a zooming of the small box in left plot. AM_V and $AM\_\hat{V}$ (EU_V and $EU\_\hat{V}$) are abbreviations for American (European) derivative prices not including and including XVA, respectively.

6 Conclusions

We formulated and studied iterative methods for the nonlinearity of the PDE problem arising when taking into account the (possibly bilateral) credit risk in valuing financial derivatives. It is shown that simple fixed-point iteration methods for the European XVA PDE problem require 3-4 iterations per timestep, while the proposed discrete penalty iteration converges in less than 1.1 (average) iterations per timestep. For the American XVA PDE problem the proposed double-penalty iteration converges in less than 1.15 (average) iterations per timestep. The main conclusion of this work is that the proposed penalty iteration methods are powerful (nearly optimal) techniques for handling nonlinearities in the context of financial derivative XVA pricing by a PDE model. The penalty methods proposed can be extended to multi-asset financial derivatives pricing, and to stochastic default intensities XVA pricing, once the associated PDE problem is derived.

Regarding the boundary conditions, a secondary conclusion is that the linear far-side condition and its proposed discretization is appropriate for all European and American contingent claims considered.

Appendix A. Proof of Theorem 1

Recall that, during timestep j, the system to be solved at iteration k, for $k \geq 1$, is

$$(\mathbb{I} - \theta\Delta\tau^j(A + P^{k-1}))u^{j,k} = (\mathbb{I} + (1-\theta)\Delta\tau^j A)u^{j-1} + \theta\Delta\tau^j P^{k-1}v^j + b^j. \tag{43}$$

where $b^j = (1-\theta)\Delta\tau^j f(u^{j-1}, v^{j-1})$, for simplicity. The proof follows the lines of [15], except that we need to distinguish cases for the values of $-\lambda_B(1-R_B)$ and $-\lambda_C(1-R_C) - s_F$.

First, let us prove the monotone property of the iteration.
Equation (43) can be rewritten as

$$(\mathbb{I} - \theta\Delta\tau^j(A + P^k))u^{j,k} + \theta\Delta\tau^j(P^k - P^{k-1})u^{j,k} = (\mathbb{I} + (1-\theta)\Delta\tau^j A)u^{j-1} + \theta\Delta\tau^j P^{k-1}v^j + b^j. \quad (44)$$

Furthermore, equation (43) can also be written for the version of iteration $k+1$ as

$$(\mathbb{I} - \theta\Delta\tau^j(A + P^k))u^{j,k+1} = (\mathbb{I} + (1-\theta)\Delta\tau^j A)u^{j-1} + \theta\Delta\tau^j P^k v^j + b^j. \quad (45)$$

Subtracting equation (44) from equation (45) gives

$$(\mathbb{I} - \theta\Delta\tau^j(A + P^k))(u^{j,k+1} - u^{j,k}) = \theta\Delta\tau^j(P^k - P^{k-1})(u^{j,k} + v^j) \quad (46)$$

We, firstly, work on the case that

$$-\lambda_C(1 - R_C) - s_F \leq -\lambda_B(1 - R_B) \leq 0.$$

Examining each component of the right hand side of equation (46), there are two possible subcases:

- If $(u^{j,k} + v^j)_i \geq 0$, then $P^k_{ii} = -\lambda_C(1 - R_C) - s_F$.
  If $P^{k-1}_{ii} = -\lambda_C(1 - R_C) - s_F$, then $(P^k_{ii} - P^{k-1}_{ii}) = 0$.
  If $P^{k-1}_{ii} = -\lambda_B(1 - R_B)$, then $(P^k_{ii} - P^{k-1}_{ii}) \leq 0$.
  Hence $\theta\Delta\tau^j[(P^k - P^{k-1})(u^{j,k} + v^j)]_i \leq 0$.

- If $(u^{j,k} + v^j)_i < 0$, then $P^k_{ii} = -\lambda_B(1 - R_B)$.
  If $P^{k-1}_{ii} = -\lambda_C(1 - R_C) - s_F$, then $(P^k_{ii} - P^{k-1}_{ii}) \geq 0$.
  If $P^{k-1}_{ii} = -\lambda_B(1 - R_B)$, then $(P^k_{ii} - P^{k-1}_{ii}) = 0$.
  Hence $\theta\Delta\tau^j[(P^k - P^{k-1})(u^{j,k} + v^j)]_i \leq 0$.

Therefore, for $k \geq 1$, the vector $\theta\Delta\tau^j(P^k - P^{k-1})(u^{j,k} + v^j)$ is always non-positive. In equation (46), since $(\mathbb{I} - \theta\Delta\tau^j(A + P^k))$ is monotone, we have $u^{j,k+1} \leq u^{j,k}$ for $k \geq 1$. Thus, the iterates decrease monotonically.

The other case, i.e. $-\lambda_B(1 - R_B) < -\lambda_C(1 - R_C) - s_F \leq 0$, can be analyzed in a similar way. In this case, the iterates increase monotonically.

Second, we prove the iteration has finite termination to a solution of (19).
We consider first the case $-\lambda_C(1 - R_C) - s_F < -\lambda_B(1 - R_B) \leq 0$. Let us define

$$S^k_1 = \{i | P^k_{ii} = -\lambda_B(1 - R_B)\} \quad (47)$$
$$S^k_2 = \{i | P^k_{ii} = -\lambda_C(1 - R_C) - s_F\}. \quad (48)$$

Therefore, $S^k_1 \dot{\cup} S^k_2$ is the set of all the nodes in the discretization. Since in the case of $-\lambda_C(1 - R_C) - s_F < -\lambda_B(1 - R_B) \leq 0$, $u^{j,k}$ decreases monotonically, any node in S^k_1 remains in S^l_1 if $l \geq k \geq 1$. At the same time, if $S^k_1 = S^{k-1}_1, S^k_2 = S^{k-1}_2$, then $P^k = P^{k-1} = 0$, and the iteration terminates. Therefore, during each iteration, at least one node will move into the set S^k_1 and remain in S^{k+1}_1 before the termination. Hence the iteration always terminates within at most $N + 1$ iterations, where N is the size of the linear system, although, numerical experiments indicate that the iteration needs only one or two steps. Moreover, at termination, we have $(\mathbb{I} - \theta\Delta\tau^j(A + P^{k-1}))u^{j,k} = (\mathbb{I} + (1-\theta)\Delta\tau^j A)u^{j-1} + \theta\Delta\tau^j P^{k-1}v^j + b^j$, thus,

$(\mathbb{I}-\theta\Delta\tau^j(A+P^k))u^{j,k} = (\mathbb{I}+(1-\theta)\Delta\tau^j A)u^{j-1}+\theta\Delta\tau^j P^k v^j + b^j$. Therefore, $u^{j,k}$ is a solution to (19). Additionally, the iterates are clearly bounded.

We now consider the case $-\lambda_C(1-R_C)-s_F = -\lambda_B(1-R_B) \le 0$. In this case, it is easy to see that $P^k = P^{k-1}$. Thus, the iteration terminates in one step.

Finally, the case $-\lambda_B(1-R_B) < -\lambda_C(1-R_C)-s_F \le 0$ can be handled similarly as the case $-\lambda_C(1-R_C)-s_F < -\lambda_B(1-R_B) \le 0$.

Last, we demonstrate the uniqueness of the solution to the nonlinear problem (19).

Suppose there are two solutions u_1 and u_2 and $P_1 = P(u_1, v)$ and $P_2 = P(u_2, v)$ for any arbitrary v. Then, u_1 and u_2 must satisfy

$$(\mathbb{I}-\theta\Delta\tau^j(A+P_1))u_1 = (\mathbb{I}+(1-\theta)\Delta\tau^j A)u_{j-1}+\theta\Delta\tau^j P_1 v + b^j, \tag{49}$$
$$(\mathbb{I}-\theta\Delta\tau^j(A+P_2))u_2 = (\mathbb{I}+(1-\theta)\Delta\tau^j A)u_{j-1}+\theta\Delta\tau^j P_2 v + b^j. \tag{50}$$

Let us rewrite equation (49) as

$$(\mathbb{I}-\theta\Delta\tau^j(A+P_2))u_1 + \theta\Delta\tau^j[P_2-P_1]u_1 = (\mathbb{I}+(1-\theta)\Delta\tau^j A)u^{j-1}+\theta\Delta\tau^j P_1 v + b^j. \tag{51}$$

Subtracting equation (50) from equation (51) gives

$$(\mathbb{I}-\theta\Delta\tau^j(A+P_2))(u_1-u_2) = \theta\Delta\tau^j(P_1-P_2)(u_1+v). \tag{52}$$

By a similar argument we used in proving the monotone property of the iteration, we have that

$$\theta\Delta\tau^j(P_1-P_2)(u_1+v) \le 0, \text{ if } -\lambda_B(1-R_B) \ge -\lambda_C(1-R_C)-s_F, \tag{53}$$
$$\theta\Delta\tau^j(P_1-P_2)(u_1+v) \ge 0, \text{ if } -\lambda_B(1-R_B) < -\lambda_C(1-R_C)-s_F. \tag{54}$$

Since the left hand side matrix of (52) $(\mathbb{I}-\theta\Delta\tau^j(A+P_2))$ is monotone, we have that

$$u_1-u_2 \le 0, \text{ if } -\lambda_B(1-R_B) \ge -\lambda_C(1-R_C)-s_F, \tag{55}$$
$$u_1-u_2 \ge 0, \text{ if } -\lambda_B(1-R_B) < -\lambda_C(1-R_C)-s_F. \tag{56}$$

We can interchange the subscripts of u_1 and u_2, and obtain

$$u_1-u_2 \ge 0, \text{ if } -\lambda_B(1-R_B) \ge -\lambda_C(1-R_C)-s_F, \tag{57}$$
$$u_1-u_2 \le 0, \text{ if } -\lambda_B(1-R_B) < -\lambda_C(1-R_C)-s_F. \tag{58}$$

Hence, in any case, $u_1 = u_2$, indicating that the iterations converge to a unique solution.

7 Acknowledgements

This work is supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada, and the Ontario Graduate Scholarship (OGS).

8 Declaration of Interest

The authors report no conflicts of interest. The authors alone are responsible for the content and writing of the paper.

References

[1] I. ARREGUI, B. SALVADOR, AND C. VÁZQUEZ, *CVA computing by PDE models*, in NAA16: Numerical Analysis and its Applications, I. Dimov et al, ed., LNCS 10187, Springer, 2017, pp. 15–24.

[2] I. ARREGUI, B. SALVADOR, AND C. VÁZQUEZ, *PDE models and numerical methods for total value adjustment in European and American options with counterparty risk*, Applied Mathematics and Computation, 308 (2017), pp. 31–53.

[3] ——, *A Monte Carlo approach to American options pricing including counterparty risk*, International Journal of Computer Mathematics, 96 (2019), pp. 2157–2176.

[4] M. BICHUCH, A. CAPPONI, AND S. STURM, *Arbitrage-free XVA*, Mathematical Finance, 28 (2018), pp. 582–620.

[5] D. BRIGO AND A. CAPPONI, *Bilateral counterparty risk valuation with stochastic dynamical models and application to credit default swaps*, https://arxiv.org/pdf/0812.3705, (2008).

[6] ——, *Bilateral counterparty risk with application to CDSs*, Risk, 23 (2010), pp. 85–90.

[7] D. BRIGO, A. CAPPONI, AND A. PALLAVICINI, *Arbitrage-free bilateral counterparty risk valuation under collateralization and application to credit default swaps*, Mathematical Finance: An International Journal of Mathematics, Statistics and Financial Economics, 24 (2014), pp. 125–146.

[8] C. BURGARD AND M. KJAER, *In the balance*, Risk, 24 (2011), pp. 72–75.

[9] C. BURGARD AND M. KJAER, *Partial differential equation representations of derivatives with bilateral counterparty risk and funding costs*, The Journal of Credit Risk, 7 (2011), pp. 75–93.

[10] C. BURGARD AND M. KJAER, *The FVA debate: In theory and practice*, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2157634, (2012).

[11] ——, *Funding strategies, funding costs*, Risk, 26 (2013), pp. 82–87.

[12] C. CHRISTARA AND D. M. DANG, *Adaptive and high-order methods for valuing American options*, J. Comput. Finance, 14 (2011), pp. 73–113.

[13] S. CRÉPEY, *Bilateral counterparty risk under funding constraints, part II: CVA*, Mathematical Finance, 25 (2015), pp. 23–50.

[14] P. A. FORSYTH AND G. LABAHN, *Numerical methods for controlled Hamilton-Jacobi-Bellman PDEs in finance*, J. Comput. Finance, 11 (2007), pp. 1–43.

[15] P. A. FORSYTH AND K. R. VETZAL, *Quadratic convergence for valuing American options using a penalty method*, SIAM J. Sci. Comput., 23 (2002), pp. 2095–2122.

[16] A. GREEN, *XVA: Credit, Funding and Capital Valuation Adjustments*, Wiley, 2015.

[17] J. GREGORY AND I. GERMAN, *Closing out DVA?*, Risk, 26 (2013), pp. 96–100.

[18] J. HULL AND A. WHITE, *The FVA debate*, Risk, 25 (2012), pp. 83–85.

[19] J. C. HULL, *Options, Futures, and Other Derivatives*, Pearson, 10 ed., 2017.

[20] K. ITO, C. REISINGER, AND Y. ZHANG, *A neural network based policy iteration algorithm with global H^2-superlinear convergence for stochastic games on domains*, 2019. arXiv:1906.02304v2.

[21] R. KANGRO AND R. NICOLAIDES, *Far field boundary conditions for Black-Scholes equations*, SIAM J. Numer. Anal., 38 (2000), pp. 1357–1368.

[22] V. PITERBARG, *Funding beyond discounting: collateral agreements and derivatives pricing*, Risk, 23 (2010), pp. 42–47.

[23] P. WILMOTT, S. HOWISON, AND J. DEWYNNE, *The Mathematics of Financial Derivatives*, Cambridge University Press, 1995.

[24] H. WINDCLIFF, P. A. FORSYTH, AND K. R. VETZAL, *Analysis of the stability of the linear boundary condition for the Black-Scholes equation*, J. Comput. Finance, 8 (2004), pp. 65–92.