Numerical Solution of Two Asset Jump Diffusion Models for Option Valuation

Simon S. Clift and Peter A. Forsyth

Original: December 5, 2005
Revised: January 31, 2007

Abstract

Under the assumption that two financial assets evolve by correlated finite activity jumps superimposed on correlated Brownian motion, the value of a contingent claim written on these two assets is given by a two dimensional parabolic partial integro-differential equation (PIDE). An implicit, finite difference method is derived in this paper. This approach avoids a dense linear system solution by combining a fixed point iteration scheme with an FFT. The method prices both American and European style contracts independent (under some simple restrictions) of the option payoff and distribution of jumps. Convergence under the localization from the infinite to a finite domain is investigated, as are the theoretical conditions for the stability of the discrete approximation under maximum and von Neumann analysis. The analysis shows that the fixed point iteration is rapidly convergent under typical market parameters. The rapid convergence of the fixed point iteration is verified in some numerical tests. These tests also indicate that the method used to localize the PIDE is inexpensive and easily implemented.

Keywords: Two-asset, option pricing, partial integro-differential equation, finite difference, American option, jump diffusion.

Acknowledgment: This work was supported by the Natural Sciences and Engineering Research Council of Canada, and ITO33, Paris.

1 Introduction

To price financial option contracts under actual market conditions, models for the evolution of the underlying asset prices are required which are more complex than a simple Black-Scholes diffusion process. This paper presents a finite difference pricing method for options on two assets where the asset prices evolve by jumps superimposed on Brownian motion. In this case, the pricing equation is a two dimensional, parabolic, partial integro-differential equation (PIDE). The method derived here can price both European options and those having an American early exercise feature. In the American case, no assumptions are made concerning the location of the exercise boundary.

The jumps in the logarithm of the prices may be distributed by any finite activity process; we demonstrate two-dimensional correlated Normal [39] and exponential [33, 27] distributions. The technique can be implemented as a direct extension to existing two-asset finite difference codes for American options. An implicit time stepping technique is used, which eliminates time step restrictions due to stability considerations. A fixed point iteration is used to avoid a dense linear system solution that would otherwise arise from the integral term. The average additional work for computing jump diffusion prices, compared to plain Brownian motion, consists of two to four FFT operations per time step.

Intuitively, the jump diffusion model is attractive because it adds the idea of asset price jumps induced by discrete economic events (for example, earnings surprises) to the idea of an otherwise efficient market moving by Brownian motion. It is well known that implied volatilities, computed from market prices for options, vary over different strikes and maturities. This results in effects such as a volatility smile or skew which, in many cases, jump diffusions can explain. For an overview of the evidence that suggests that jump processes are an important factor in observed market prices, we refer the reader to [11].

Option pricing with these market models in the one-asset case has been explored for both European and American options by several researchers. Andersen and Andreasen [4] developed an operator splitting approach for European options which was unconditionally stable and second order in time. A general semi-analytic solution was described by Lewis [31, 32] for European options on one asset and by Fonseca et. al. for multiple assets [18].

Pham [42] discussed theoretical properties of the solutions to American options under jump diffusion processes. An approximation method is discussed by Mulinacci [40]. Binomial lattice methods are the equivalent of explicit finite difference methods [55], and an approach to the one-dimensional problem using this technique is discussed by Amin [3]. Zhang [52] developed a semi-implicit approach for American options using a traditional linear complementarity solver for jump diffusion processes with Normally distributed jumps. Wavelet methods for implicit solution of single factor infinite activity jump diffusion problems were developed in [37, 35, 36].

Recent work in finite difference approaches by Briani et. al. [9, 8] and Cont et. al. [12] use explicit time stepping for the integral operator introduced by the jump process. An implicit, finite difference approach for single asset American and European options was explored by d'Halluin, Forsyth, et. al. [15, 14]. This method employed a penalty method and was demonstrated to be quadratically convergent versus grid spacing and time step size for both American and European options. A similar approach, which uses an iterative method to solve the implicit discretized PIDE and which also uses an FFT to carry out the dense matrix-vector multiply, was developed by Almendral and Oosterlee [2].

Two asset American claims under jump diffusion were priced using a Markov chain approach in [34]. A Markov chain can be viewed as essentially an explicit finite difference method. The jump terms were handled using an extension of the method in [3].

The two-asset, correlated Brownian motion model [49] is a simple extension of the one-asset Black-Scholes model [6, 38]. The work in this paper adapts the finite difference jump diffusion work of [15, 14] to the work on two-factor option pricing of Zvan, Forsyth et. al. [53, 54, 19] to produce a similarly quadratically convergent method. This new two-asset technique retains the advantages of being able to price options with general types of payoffs and barriers for American as well as European options.

The following are the main results of this paper.

- A fixed point iteration method is developed which allows implicit time stepping for the PIDE. It avoids a dense matrix multiply by utilizing an FFT. A convergence analysis shows that we can expect this iteration to converge rapidly for normal market parameters i.e. a reduction of the l_∞ or l_2 norm by 10^{-6} in $2-3$ iterations. This is verified in some numerical experiments. A major advantage of this approach is that it is straightforward to add a jump model to existing software which prices two asset claims under Brownian motion. The fixed point iteration effectively decouples the jump process terms from the Brownian motion terms.

- The fixed point iteration can be easily extended to handle American options (as in [14]) using a penalty method.

- In the case of constant coefficients, we can rotate the grid so that the fully implicit discretization is monotone. Consistency, monotonicity and stability imply convergence to the viscosity solution for American options with non-smooth payoffs [8, 12]. However, numerical experiments reveal that the error for a given mesh size on the rotated grid is in fact larger compared to the error on a conventional grid. This suggests that attempting to force a monotone discretization scheme may not be necessary in practice.

1.1 Overview

Section 2 of this work reviews the equations governing option valuation over two assets with jump diffusion. The localization of the equations from an infinite to a finite domain, and the control of the resulting error, is discussed in Section 3. The discretization method discussed in Section 4 is studied to determine the theoretical conditions for stability in Section 5. In Section 6, the fixed point iteration used to advance the solution by one time step is demonstrated to be rapidly convergent under normal parameter ranges. The application of the fixed point iteration to American option pricing is also discussed in Section 6. Section 7 gives a number of numerical examples to demonstrate the techniques presented in the paper. Experimentally, we see that this method is quadratically convergent. Certain details of the problem specification and the longer proofs are in the appendices.

2 Governing Equations

Our approach to valuing option contracts uses a time to expiry and the prices of the underlying assets as independent variables. The value of the option will be determined after applying a logarithmic transform to the two asset prices. This "log-price scaling" is convenient for analysis, although numerically our approach works equally well in the original price scaling.

2.1 The Finite Activity Jump Diffusion Model

To compute the value of an option we use, as independent variables, the time to expiry $\tau = T - t$ and two asset prices S_1 and S_2 where

$$\begin{aligned} t &\in [t_0, t_0 + T] \\ \tau &\in [0, T] \\ S = (S_1, S_2) &\in \Omega_{\widetilde{\infty}} \\ \Omega_{\widetilde{\infty}} &= [0, \infty] \times [0, \infty] \quad . \end{aligned} \tag{2.1}$$

In Appendix A, we describe the risk neutral price processes assumed for (S_1, S_2). We apply a logarithmic transform to the asset prices

$$\begin{aligned} y = (y_1, y_2) = (\log(S_1), \log(S_2)) &\in \Omega_\infty \\ \Omega_\infty &= [-\infty, \infty] \times [-\infty, \infty] \end{aligned} \tag{2.2}$$

and wish to determine the value of a European option $U(y, \tau)$. By taking expectations under the risk neutral price process described in Appendix A [8, 11], we obtain the following parabolic partial integro-differential equation (PIDE) for the value $U(y, \tau)$

$$\begin{aligned} U_\tau &= \mathcal{L}U + \lambda \mathcal{H}U \\ U(y, 0) &= \mathcal{I}(y) \end{aligned} \tag{2.3}$$

which is defined over $\Omega_\infty \times [0, T]$. The linear differential operator $\mathcal{L}$ is defined as

$$\mathcal{L}U = (\mathbf{D}\nabla) \cdot \nabla U + \mathbf{V} \cdot \nabla U - rU \tag{2.4}$$

$$\mathbf{D} \in \mathbb{R}^2 \times \mathbb{R}^2 \; ; \quad \mathbf{V} \in \mathbb{R}^2 \; ; \quad \nabla = \left(\frac{\partial}{\partial y_1}, \frac{\partial}{\partial y_2} \right)^T$$

$$\mathbf{V} = \begin{pmatrix} r - {\sigma_1}^2/2 \\ r - {\sigma_2}^2/2 \end{pmatrix} \tag{2.5}$$

$$\mathbf{D} = \frac{1}{2} \begin{pmatrix} {\sigma_1}^2 & \rho_v \sigma_1 \sigma_2 \\ \rho_v \sigma_1 \sigma_2 & {\sigma_2}^2 \end{pmatrix} \quad . \tag{2.6}$$

The coefficients $\sigma_1 > 0$, $\sigma_2 > 0$ and $-1 \leq \rho_v \leq 1$ are the volatility magnitudes and correlation, respectively, of the Brownian motion processes on the two assets, and $r \geq 0$ represents a risk-free rate of return. For simplicity, we do not include dividends in Equation 2.4. The operator $\lambda\mathcal{H}$ represents the effects of finite activity asset price jumps generated by a Poisson process with mean arrival rate $\lambda > 0$. For brevity, we write

$$\begin{aligned} J &= (J_1, J_2) \\ y + J &= (y_1 + J_1, y_2 + J_2) \\ (e^J - 1) &= \left(e^{J_1} - 1, e^{J_2} - 1\right)^T \end{aligned} \tag{2.7}$$

and then we can write the integral term as

$$\lambda\mathcal{H}U = \lambda \int_{-\infty}^{\infty} g(J) \left[U\left(y + J, \tau\right) - U(y, \tau) - \left(e^J - 1\right) \cdot \nabla U(y, \tau)\right] \, dJ \tag{2.8}$$

where jump magnitudes J are distributed according to a probability density function $g(J)$. In this study we make the standard assumption that $g(J)$ is independent of y, and assume that $g(J)$ satisfies the technical conditions of [23] §II.1.2 Definition 1.6 (see also [11] Proposition 3.18) in particular, that we may write separate integrals for the second term of $\mathcal{H}U$

$$\int_{-\infty}^{\infty} g(J)U(y, \tau) \, dJ = U(y, \tau)$$

and the third term

$$\int_{-\infty}^{\infty} g(J) \left[\left(e^J - 1\right) \cdot \nabla U(y, \tau)\right] \, dJ = \left[\int_{-\infty}^{\infty} g(J) \left(e^J - 1\right) \, dJ\right] \cdot \nabla U(y, \tau) = \begin{pmatrix} \kappa_1 \\ \kappa_2 \end{pmatrix} \cdot \nabla U(y, \tau) \quad .$$

The values $\kappa_1, \kappa_2 < \infty$ correct for the mean drift due to the first term of operator $\mathcal{H}U$. This first term is written separately as

$$\mathcal{J}U = \int_{-\infty}^{\infty} g(J)U\left(y + J, \tau\right) \, dJ = \int_{-\infty}^{\infty} g(J - y)U\left(J, \tau\right) \, dJ \tag{2.9}$$

which are equivalent forms of a correlation product. Specific formulations of $g(J)$ with Normal and exponentially distributed jumps, which are analogous to the one-dimensional jump models of Merton [39] and Kou [27] respectively, are given in Appendix B.

An American option may be exercised for its terminal payoff at any time. We may write the American option price as the solution to a linear complementarity problem [42, 52]

$$\begin{aligned} U_\tau &\geq \mathcal{L}U + \lambda\mathcal{H}U & (2.10) \\ U &\geq \mathcal{I}(y) & (2.11) \\ U(y, 0) &= \mathcal{I}(y) \end{aligned}$$

where at least one of Equations 2.10 or 2.11 must hold with equality.

2.2 Price Scaling Notes

In "price scaling", the value of the option $\tilde{U}(S, t)$ in time t and two space dimensions $S \in \Omega_{\tilde{\infty}}$ is determined by solving a PIDE analogous to Equation 2.3. It is defined on $\Omega_{\tilde{\infty}} \times [0, T]$. Of particular note are the analogues of the advection tensor $\mathbf{V}$ and diffusion tensor $\mathbf{D}$ of Equations 2.5 and 2.6. In price scaling they become

$$\begin{aligned} \tilde{\mathbf{V}} &= \begin{pmatrix} r\,S_1 \\ r\,S_2 \end{pmatrix} \\ \tilde{\mathbf{D}} &= \frac{1}{2} \begin{pmatrix} {\sigma_1}^2 {S_1}^2 & \rho_v \sigma_1 \sigma_2 S_1 S_2 \\ \rho_v \sigma_1 \sigma_2 S_1 S_2 & {\sigma_2}^2 {S_2}^2 \end{pmatrix} \end{aligned} \tag{2.12}$$

which we observe have zeros at the natural lower boundary of the problem at $S_1 = 0$ and $S_2 = 0$. These lines correspond to $y_1 = -\infty$ and $y_2 = -\infty$ in log-price scaling, and we shall exploit this feature of the problem when we localize it to a finite domain.

2.3 Contract Types and Initial Conditions

The examples in this paper value four types of option contracts which differ in their payoffs. These are specified in price scaling. Initial conditions $\tilde{\mathcal{I}}(S)$ are defined by function $\tilde{\mathcal{P}}(S)$ and an option exercise price K to define

$$\tilde{\mathcal{I}}(S) = \begin{cases} \max\left(\tilde{\mathcal{P}}(S) - K, 0\right) & \text{call, and} \\ \max\left(K - \tilde{\mathcal{P}}(S), 0\right) & \text{put} \end{cases} \tag{2.13}$$

options where the underlying is either

$$\tilde{\mathcal{P}}(S) = \begin{cases} \max\left(S_1, S_2\right) & \text{the maximum, or} \\ \min\left(S_1, S_2\right) & \text{the minimum} \end{cases} \tag{2.14}$$

of the two underlying assets. If we refer to the payoff in log-price scaling we write $\mathcal{P}(y)$, which is used to form our initial condition $\mathcal{I}(y)$. These four examples are not specific to our numerical solution technique, which can be used for any payoff that can be bounded linearly (see Assumption 3.1 below).

3 Localization

Our solution technique requires truncating the infinite domains $\Omega_{\widetilde{\infty}}$ and Ω_∞ at finite boundaries. In this section we discuss this localization and the associated convergence issues in log-price scaling. We shall discuss these issues in the context of pricing European options (Equation 2.3). We will use the same approach for localizing American options, which can justified in this case by numerical experiment.

The localization method which follows is easy to visualize and implement. Essentially, we divide the computational domain into an inner or core region, and an outer region. In the inner region, the full PIDE is solved. In the outer region, under log-price scaling, we set all terms involving the integral term to zero and simply solve a parabolic PDE. This can be justified on the basis of the properties of the Green's function of the PIDE [23]. As well, the integral term $\mathcal{H}$ is asymptotically zero in regions where the solution is asymptotically linear; linearity is a common assumption for far-field boundary conditions in finance [51]. The outer region then acts as a buffer zone, so that the integral terms in the inner region have enough information for a sufficiently accurate evaluation.

3.1 Localization in Log-Price Scaling

In log-price scaling (Equation 2.2) we define an interior domain nested in a finite domain $\Omega_C \subset \Omega_D \subset \Omega_\infty$ (see Figure 1) and apply the bounds $0 < Y_C < Y_D$ to the upper and lower limits

$$\begin{aligned} \Omega_C &= [-Y_C, Y_C] \times [-Y_C, Y_C] \\ \Omega_D &= [-Y_D, Y_D] \times [-Y_D, Y_D] \\ \partial\Omega_D &= \{y_1 = [-Y_D, Y_D], y_2 = \{-Y_D, Y_D\}\} \cup \{y_2 = [-Y_D, Y_D], y_1 = \{-Y_D, Y_D\}\} \quad . \end{aligned} \tag{3.1}$$

In general, the upper and lower limits need not be equal nor the domains square. We determine the approximate option value $V(y, \tau)$, $y \in \Omega_D$ with boundary conditions $B(y, \tau)$ by solving

$$\begin{aligned} V_\tau &= \mathcal{L}V + \lambda_C \mathcal{H}_D V \\ V(y, 0) &= \mathcal{I}(y) \\ V(y, \tau) &= B(y, \tau) \; ; \quad y \in \partial\Omega_D \end{aligned} \tag{3.2}$$

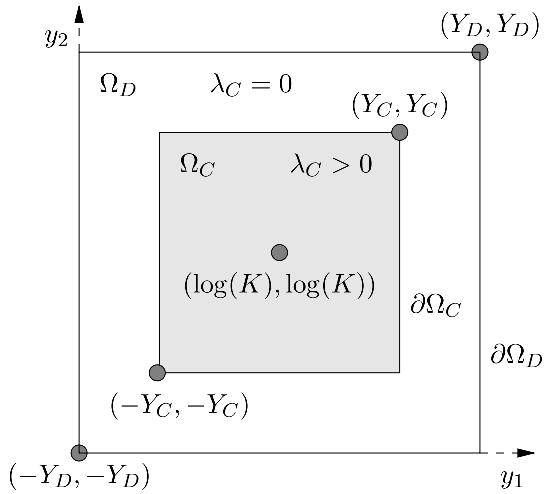

FIGURE 1: *Domains $\Omega_C \subset \Omega_D$ in log-price scaling truncate the infinite domain Ω_∞. In Ω_C we apply $\lambda_C > 0$ so that the PIDE 3.2 is computed with the jump component in Ω_C only. Ω_D has a lower bound at a finite point hence, in price scaling, above the zero axis of the asset price.*

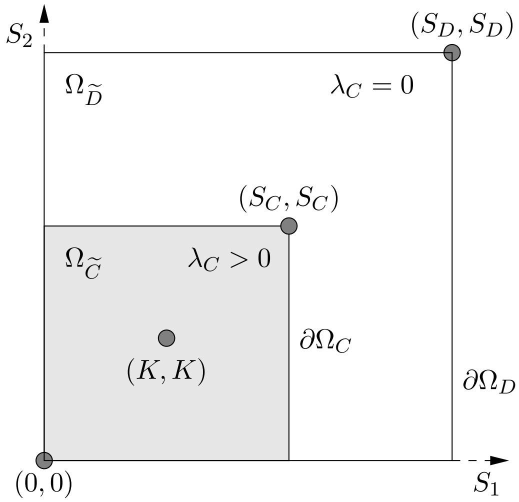

FIGURE 2: *Domains $\Omega_{\widetilde{C}} \subset \Omega_{\widetilde{D}}$ in price scaling truncate the infinite domain $\Omega_{\widetilde{\infty}}$. In $\Omega_{\widetilde{C}}$ we apply $\lambda_C > 0$ so that the PIDE is computed with the jump component in that domain. Outside of $\Omega_{\widetilde{C}}$ we have only diffusion.*

which is defined over $\Omega_D \times [0, T]$, where the integral operator is altered from Equation 2.8 to become

$$\begin{aligned}
\lambda_C &= \begin{cases} \lambda & y \in \Omega_C \\ 0 & y \in \Omega_D \setminus \Omega_C \end{cases} \\
\mathcal{H}_D V &= \int_{(y+J)\in\Omega_D} g(J) V\left(y + J, \tau\right)\, dJ - \int_{-\infty}^{\infty} g(J) \left[V(y,\tau) + \left(e^J - 1\right) \cdot \nabla V(y,\tau)\right]\, dJ \qquad (3.3) \\
&= \left[\int_{J\in\Omega_D} g(J - y) V\left(J, \tau\right)\, dJ\right] - V(y,\tau) - \begin{pmatrix} \kappa_1 \\ \kappa_2 \end{pmatrix} \cdot \nabla V(y,\tau) \quad .
\end{aligned}$$

We have written the first term of $\mathcal{H}_D$ in two ways, corresponding to the two forms of $\mathcal{J}$ in Equation 2.9. In the first form, the integration for a point $y \in \Omega_C$ must be performed over $(y + J) \in \Omega_D$; the value of V is not defined outside of Ω_D so the integration limit for J depends on y. The second form of $\mathcal{H}_D$ is equivalent, but since $g(J - y)$ is defined over $(J - y) \in \mathbb{R}^2$, the integration limit is simpler and independent of y. The second form also illustrates one reason why λ_C is set so that $\lambda_C \mathcal{H}_D V(y) = 0\ \forall\ y \notin \Omega_C$. For $y \in \Omega_C$ the range of evaluation of $g(J - y), J \in \Omega_D$ is not severely truncated in any given direction compared to the infinite integral $J \in \Omega_\infty$ used for the second and third terms. We shall set the size of Ω_C and Ω_D so that this truncation occurs when $g(J - y)$ is small, so that the finite evaluation range has a small, controlled impact on the solution over Ω_C.

For this study we consider, as payoffs, the put on the minimum of two assets and the call on the maximum (Equation 2.13). Thus the upper boundary may be approximated by a constant value

$$B(y,\tau) = \mathcal{I}(y), \quad y \in \partial\Omega_D,\ y_1 = Y_D \text{ or } y_2 = Y_D \qquad (3.4)$$

and enforced using a Dirichlet boundary condition. For a complete review of the possible boundary condition assumptions and their implications see [51]. Along the lower boundary we set B by approximating $\mathbf{V}$ and

D by

$$\mathbf{V}_l = \begin{cases} \begin{pmatrix} r - \sigma_1^2/2 \\ 0 \end{pmatrix} & y \in \partial\Omega_D,\ y_1 \in (-Y_D, Y_D),\ y_2 = -Y_D \\ \\ \begin{pmatrix} 0 \\ r - \sigma_2^2/2 \end{pmatrix} & y \in \partial\Omega_D,\ y_1 = -Y_D,\ y_2 \in (-Y_D, Y_D) \end{cases} \tag{3.5}$$

$$\mathbf{D}_l = \begin{cases} \frac{1}{2}\begin{pmatrix} {\sigma_1}^2 & 0 \\ 0 & 0 \end{pmatrix} & y \in \partial\Omega_D,\ y_1 \in (-Y_D, Y_D),\ y_2 = -Y_D \\ \\ \frac{1}{2}\begin{pmatrix} 0 & 0 \\ 0 & {\sigma_2}^2 \end{pmatrix} & y \in \partial\Omega_D,\ y_1 = -Y_D,\ y_2 \in (-Y_D, Y_D)\ . \end{cases} \tag{3.6}$$

At the corner point $B(y = (-Y_D, -Y_D), \tau)$ we set the boundary condition to be $V_\tau = -rV$, which is the solution to the governing PDE as $y_1, y_2 \to -\infty$. The modified differential terms at the lower boundaries also match those on the lower boundaries $S_1 = 0$ or $S_2 = 0$ in price-scaling (Equation 2.12). In log-price scaling, the lower boundary is in extended domain Ω_D thus $\lambda_C = 0$, which eliminates the integral terms. In the typical case $\partial\Omega_D$ is spaced sufficiently far from Ω_C that the error from boundary approximations are well controlled (see Sections 6.4 and 7.2).

3.2 Localization in Price Scaling

Our initial condition $\tilde{\mathcal{I}}(S)$ is defined in price scaling, hence we also note the localization for price scaling. We use a finite domain $\Omega_{\widetilde{D}} \subset \Omega_{\widetilde{\infty}}$ with an interior domain $\Omega_{\widetilde{C}} \subset \Omega_{\widetilde{D}}$ as shown in Figure 2. In general, again, the domains need not be square, but for ease of exposition we define the finite domains to be square with upper bounds $0 < S_C < S_D$ where $S_C = \exp(Y_C)$ and $S_D = \exp(Y_D)$

$$\begin{aligned} \Omega_{\widetilde{D}} &= [0, S_D] \times [0, S_D] \\ \Omega_{\widetilde{C}} &= [0, S_C] \times [0, S_C] \\ \partial\Omega_{\widetilde{D}} &= \{S_1 = [0, S_D], S_2 = S_D\} \cup \{S_2 = [0, S_D], S_1 = S_D\} \quad . \end{aligned} \tag{3.7}$$

In price scaling the lower boundary is a 1D PIDE, where the jumps have the marginal distributions of $g(J)$ along S_1 and S_2. The numerical solution approach in this scaling would require the addition of a 1D solution along this boundary, a minor extension of the approach which is not required when working in log-price scaling.

3.3 Localization Convergence Estimates for European Options

This choice of localization to these finite domains and their accompanying approximations is convenient for two reasons. First, if $Y_D > Y_C$ $(S_D > S_C)$ are sufficiently large then we expect that the error due to approximating $\mathcal{H}$ by $\mathcal{H}_D$ and λ by λ_C will be negligible in Ω_C, particularly near the strike. Secondly, we shall see that this localization allows us to apply an efficient, FFT-based computation for the integral term $\mathcal{H}_D V$. We make the following assumptions.

Assumption 3.1 *The initial condition (the option payoff) $\mathcal{I}(y)$ can be bounded by*

$$\mathcal{I}(y) \leq c_1 + c_2\left(e^{y_1} + e^{y_2}\right)$$

for some constants c_1 and c_2 and the jump distribution must be such that $|\mathcal{H}\mathcal{I}| < \infty$ for $|y| < \infty$.

A payoff which is linearly valued in price scaling, such as those listed in Section 2.3, satisfies Assumption 3.1 for the Normal and exponentially distributed jumps which we use in our numerical examples.

Assumption 3.2 *The solution $U(y,\tau)$ to Equation 2.3 satisfies the condition*

$$|\mathcal{H}U| \leq c_3 + c_4\left(e^{y_1} + e^{y_2}\right) \quad \textit{for} \quad y \in \Omega_\infty \setminus \Omega_C \tag{3.8}$$

for constants c_3 and c_4.

We rely on the assumption that, besides being finite, the action of the jump operator can be bounded by a plane in price scaling. Again, this will be satisfied by our examples, where the jump density functions are independent of y. This limitation could apply, for example, to jump density functions which increase jump magnitude with log-price.

Assumption 3.3 *The artificial boundary condition $B(y,\tau)$ of Equation 3.2 is bounded by the growth in the exact solution $U \in \Omega_\infty$, i.e.*

$$|B(y,\tau) - U(y,\tau)| \leq c_5 + c_6\, U(y,\tau) \tag{3.9}$$

for some constants c_5 and c_6.

Note that Equation 3.9 is satisfied if $B(S,\tau) = 0$.

Assumption 3.4 *The PIDE's 2.3 and 3.2 must satisfy the conditions in Garroni and Menaldi [23] §I, II. In particular, the diffusion coefficients must be bounded on Ω_∞ and the operator $\mathcal{L}$ must be uniformly elliptic, so that a smooth, classical, bounded solution exists. With these conditions, the solution can be represented by convolutions of Green's functions and Poisson functions, as in [23] §IV.*

Note that the option pricing PIDE formulated in price scaling over $\Omega_{\widetilde{\infty}}$ does not satisfy Assumption 3.4: the differential operator does not have bounded coefficients on $\Omega_{\widetilde{\infty}}$ and is not uniformly elliptic [23] because of the zero diffusion tensor coefficients on $S_1 = 0$ and $S_2 = 0$. Hence we proceed in log-price scaling: the domain is bounded away from the $S_1 = 0$ and $S_2 = 0$ axes and the diffusion tensor coefficients are constant.

Assumption 3.5 *The initial and boundary conditions are smooth and have finite first and second derivatives with respect to y_1 and y_2 (see [23] §II.1.1).*

The initial conditions in Equation 2.13 do not meet Assumption 3.5, however, we may make an arbitrarily close, but smooth approximation to $\mathcal{I}(y)$ to satisfy the theory (independent of the numerical solution approach). Typically, this regularization is done using a mollification of the initial condition, with which the resulting error in the final solution may also be bounded to an arbitrarily small value. See [21, 20] for the classical mollification method for PDE's, and a survey by Lamm [29] for its application to integral equations. Recently this approach has been applied in practice to financial problems by Friz and Gatheral [22] and in theory by Jakobsen et. al. [24] (particularly Lemma 3.1).

Remark 3.1 *Assumptions 3.1 to 3.5 are taken as fulfilled for the following theorems and the remaining discussion of the localization of the continuous operators.*

We must now show that the error due to the solution V of Equation 3.2 over $y \in \Omega_D$ satisfies $|V(y,\tau) - U(y,\tau)| \to 0$ as $\Omega_D, \Omega_C \to \Omega_\infty$ where U is the solution to Equation 2.3 over Ω_∞. We do this in two parts.

Theorem 3.1 *Let U be the solution to Equation 2.3. Let V be the solution to the localized PIDE 3.2 embedded in Ω_∞ to form the initial value problem*

$$V_\tau = \mathcal{L}V + \lambda_C \mathcal{H}_D V \ , \quad V(y,0) = \mathcal{I}(y) \ , \quad y \in \Omega_\infty \ . \tag{3.10}$$

Define the ***cutoff error*** *$E_c = U - V$. The value of $E_c(y,\tau)$ over $y \in \Omega_\infty$ due to the approximation of λ by λ_C and $\mathcal{H}$ by $\mathcal{H}_D$ obeys*

$$\lim_{\Omega_C, \Omega_D \to \Omega_\infty} |E_c(y,\tau)| = 0 \quad . \tag{3.11}$$

Proof. See Appendix C.1. □

Theorem 3.2 *Let Y be the solution to Equation 3.2 with the approximate boundary condition $Y(y,\tau) = B(y,\tau), y \in \partial\Omega_D$. Let W be the solution to Equation 3.2 when we set the boundary $W(y,\tau) = V(y,\tau), y \in \partial\Omega_D$, where V is the exact value from the solution of Equation 3.10. Define the error due to approximating the exact boundary condition $V(y,\tau)$ with the approximate boundary condition $B(y,\tau)$ on $\partial\Omega_D$ as $E_b = W - Y$. The error $E_b(y,\tau)$ is bounded as*

$$\lim_{\Omega_D \to \Omega_\infty} |E_b(y,\tau)| = 0 \quad . \tag{3.12}$$

Proof. See Appendix C.2. □

Approximating Equation 2.3 by Equation 3.2 causes an error which tends to zero as $\Omega_C, \Omega_D \to \Omega_\infty$. These bounds appear at first to be disappointingly weak, however, more precise bounds would depend on the exact form of the jump size distribution $g(J)$. Using a different localization technique, the bounds for this error in one dimension were estimated in [12] using a probabilistic approach. Defining the computational domain in price scaling by $[0, S^*]$, the localization error was estimated to be [12]

$$\text{LocalizationError} \leq \frac{1}{(S^*)^\alpha}, \qquad \alpha > 0 \tag{3.13}$$

which is a similar bound as in Equations 3.11 and 3.12.

The above estimates of the localization error are overly pessimistic. To see this, we note that in many cases large regions of the payoff are asymptotically linear in price scaling as $S_1, S_2 \to \infty$. We also expect this to hold for the solution. Consider such a region $\Omega_P \subset \Omega_D$ where, in log-price scaling,

$$\begin{aligned} V(y+q,\tau) = V(y,\tau) + (Ce^y)\cdot(e^q - 1) \quad &\forall\, y, (y+q) \in \Omega_P \\ C, q \in \mathbb{R}^2 \quad \text{with} \quad C = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \quad &\text{constant,} \\ Ce^y = \begin{pmatrix} c_1 e^{y_1} \\ c_2 e^{y_2} \end{pmatrix} \ , \text{ and}& \\ (e^q - 1) = \begin{pmatrix} e^{q_1} - 1 \\ e^{q_2} - 1 \end{pmatrix} \quad .& \end{aligned}$$

Note that for $y \in \Omega_P$, we have

$$\nabla V\big|_y = \begin{pmatrix} c_1 e^{y_1} \\ c_2 e^{y_2} \end{pmatrix} \quad .$$

Examine the integral term $\mathcal{H}V$ of Equation 2.8, defining $(e^J - 1)$ as in Equation 2.7. If we limit the integral so that it is taken only over $y, (y+J) \in \Omega_P$ then

$$\begin{aligned} \mathcal{H}V(y,\tau) &\approx \mathcal{H}_P V(y,\tau) \\ &= \int_{y,y+J\in\Omega_P} g(J)\left[V(y+J,\tau) - V(y,\tau) - \left(e^J - 1\right)\cdot\nabla V\right] dJ \\ &= \int_{y,y+J\in\Omega_P} g(J)\left[V(y,\tau) + (Ce^y)\cdot\left(e^J - 1\right) - V(y,\tau) - (Ce^y)\cdot\left(e^J - 1\right)\right] dJ = 0 \ . \end{aligned}$$

In such regions we expect that the error due to dropping the integral term by setting $\lambda_C = 0$, or due to limiting the region of integration of $\mathcal{H}_D$, will be small.

4 Discretization

Recall from Equation 3.2 that our localized problem is posed on the finite domain $y \in \Omega_D$. To simplify the discretization we write a linear differential operator $\mathcal{G}$ which contains only partial differential terms. Separating out the three terms of $\mathcal{H}$, we rewrite Equation 3.2 over $\Omega_D \times [0, T]$ as

$$\begin{aligned} V_\tau &= \mathcal{G}V - (r + \lambda_C)V + \lambda_C \mathcal{J}_D V \\ V(y, 0) &= \mathcal{I}(y) \\ V(y, \tau) &= B(y, \tau) \quad ; \quad y \in \partial\Omega_D \\ \mathcal{G}V &= (\mathbf{D}\nabla) \cdot \nabla V + \begin{pmatrix} r - {\sigma_1}^2/2 - \lambda_C\,\kappa_1 \\ r - {\sigma_2}^2/2 - \lambda_C\,\kappa_2 \end{pmatrix} \cdot \nabla V \\ \mathcal{J}_D V &= \int_{J \in \Omega_D} g(J - y) V(J, \tau)\, dJ \end{aligned} \tag{4.1}$$

where $\mathbf{D}$ is as in Equation 2.6, and boundary conditions $B(y, \tau)$ are unchanged. Note that we have chosen $\mathcal{J}_D$ to match the second form of $\mathcal{H}_D$ in Equation 3.3.

We semi-discretize Equation 4.1 in time by the Crank-Nicolson method with constant time step weight $0 \le \theta \le 1$

$$\begin{aligned} \frac{W^{n+1} - W^n}{\Delta\tau} &= (1 - \theta)\left[\mathcal{G} - (r + \lambda_C) + \lambda_C \mathcal{J}_D\right] W^{n+1} \\ &\quad + \theta\left[\mathcal{G} - (r + \lambda_C) + \lambda_C \mathcal{J}_D\right] W^n \end{aligned} \tag{4.2}$$

where $W^n = W(y, \tau^n)$ is the solution to the semi-discretized problem. We define this form for use later in Section 6.1. We shall only consider the cases $\theta = 1/2$ and $\theta = 0$, which are the second order Crank-Nicolson time step and the first order fully implicit time step respectively.

4.1 General Discrete Form

The discrete equations are first written in a general form using matrices and vectors. This abstraction guides our final, specific form and permits the application of some useful general stability results.

We discretize Equation 4.2 over a grid of points

$$p_i \in \mathbb{R}^2 \ , \quad p_i \in \Omega_D \setminus \partial\Omega_D \ , \quad i = 1 \ldots P$$

so that we can form a vector of solution values at these points

$$\mathsf{w} \in \mathbb{R}^P \ , \quad \text{with elements } \mathsf{w_i} \approx W(p_i) \quad .$$

We also require a boundary condition enforcement vector

$$\mathsf{b} \in \mathbb{R}^P \ , \quad \text{with elements } \mathsf{b_i} \approx b(p_i) \quad .$$

Vector b can be seen as encoding the boundary condition nodes on $\partial\Omega_D$, where the option value is known at time steps n and $n + 1$, after these nodes are eliminated from the solution vector (see Appendix E) and hence from the discrete equations. We note that vector b is not a representation of the values of $B(y, \tau)$. The linear differential operator is discretized to form a matrix G such that

$$\mathcal{G}W \approx \mathsf{Gw} \ ; \quad \mathsf{G} \in \mathbb{R}^P \times \mathbb{R}^P, \tag{4.3}$$

and the integral operator discretized to form a matrix J such that

$$\begin{aligned} \lambda_C \mathcal{J}_D W \approx \lambda \mathsf{I_c} \mathsf{J} \mathsf{w} \ ; \quad \mathsf{I_c}, \mathsf{J} \in \mathbb{R}^P \times \mathbb{R}^P \\ [\mathsf{I_c}]_{\mathsf{ij}} = \begin{cases} 1 & \text{if } \mathsf{i} = \mathsf{j} \text{ and } p_i \in \Omega_C \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{4.4}$$

Using $\lambda \mathsf{I_c}$ to replace λ_C, the discrete form of the operator terms of Equation 4.2 can be written using a matrix

$$\mathsf{M} = - \left[\mathsf{G} + \lambda \mathsf{I_c} (\mathsf{J} - \mathsf{I}) - r\, \mathsf{I} \right] \quad . \tag{4.5}$$

We now write the discrete version of the time step Equation 4.2 so that it matches the formulation of [28, 30, 48]. The Crank-Nicolson time discretization is written using a rational polynomial $\varphi(\Delta\tau\ \mathsf{M})$ defined similarly to [28][1] with $\varphi(z) = [1 + (1-\theta)z]^{-1}\ [1 - \theta z]$. The full, general, discrete system is thus

$$[\mathsf{I} + (1-\theta)\Delta\tau\ \mathsf{M}]\, \mathsf{w}^{n+1} = [\mathsf{I} - \theta\Delta\tau\ \mathsf{M}]\, \mathsf{w}^{n} + \mathsf{b} \tag{4.6}$$

which is the form we require to apply some of the stability analyses of [28, 30] in Section 5 below.

In Section 5 we shall also see that we require two further conditions,

$$\mathsf{J_{ij}} \geq 0 \ , \quad \text{and} \quad \sum_j \mathsf{J_{ij}} \leq 1 \tag{4.7}$$

to ensure that M represents a stable discretization. Since the entries of J are defined by the values of the PDF g, we shall see that these are reasonable restrictions.

4.2 Finite Difference Form of G

The **FD grid** is a rectangular, finite difference grid defined on the finite domain $\Omega_{\widetilde{D}}$ in price scaling. At grid points near the location $(S_1 = K, S_2 = K)$, the grid is fine and has a constant spacing between nodes of (h_1, h_2) in the (S_1, S_2) directions. To save computational effort, we increase grid spacing in regions away from the strike where the high resolution is not required. Previous work [54] has shown that the original grid should be specified in price scaling in order to accurately capture the details of the option contract, in particular, the payoff and barriers need to be accurately represented. For the actual computation we transform this grid into log-price co-ordinates and, where it is required, replace the lines $S_1 = 0$ and $S_2 = 0$ with lines at $y_1 = -Y_D$ and $y_2 = -Y_D$ respectively. The grid line intersections define our P solution points $p_i \in \Omega_D \setminus \partial\Omega_D$ in log-price scaling.

After the FD grid is transformed to log-price co-ordinates (rendering it a grid with non-constant spacing everywhere) we use it to create the sparse matrix G. In Appendix E, we give the details of the discretization assuming constant grid spacing. In the interests of brevity, we omit the details for non-constant spacing, since this is completely standard. The cross-partial derivatives are discretized with a seven-point formula, using the non-constant spacing versions of Equation E.5 or E.6 when the diffusion correlation $\rho_v \geq 0$ or $\rho_v < 0$ respectively. Under some conditions (see Remark 5.3 below) we use the first-order approximation to the first partial derivatives. Again, we omit the details here, since this is completely analogous to the one dimensional case, described in [15].

The order of both approaches is $\mathrm{O}(h^2)$ for constant grid spacing h (assuming central differencing for the first order terms) to match the $\mathrm{O}((\Delta\tau)^2)$ obtained when $\theta = 1/2$ in Equation 4.6.

4.3 Discrete Integral Operator J

In our context, it is not necessary to achieve a high accuracy evaluation of the jump integral term; second order accuracy at each time step is good enough. This is in contrast to the application in [10]. We shall first motivate our method for the evaluation of the jump integral term.

Consider a simple form of J created when a second order, trapezoidal rule is used for the approximation in Equation 4.4 of $\mathscr{J}_D$. We may write it in the following form over the FD grid

$$[\mathsf{I_c J w}]_\mathsf{i} = \begin{cases} [\mathsf{J w}]_\mathsf{i} = \sum_{j=1}^{P} \gamma_{ij}\, g(p_j - p_i)\, \mathsf{w_j} & \text{for } p_i \in \Omega_C \text{ and} \\ \qquad\qquad\qquad\qquad\quad 0 & \text{for } p_i \notin \Omega_C \quad . \end{cases} \tag{4.8}$$

[1] In [28] the values of θ and $(1-\theta)$ are reversed to the sense in which we use them here.

Matrix $\gamma \in \mathbb{R}^p \times \mathbb{R}^p$ are weights set so that the result is second order accurate over the grid of P points and so that J satisfies Equation 4.7. Note that option value W is only used at points $\mathsf{w_j}$, located at nodes p_j on the grid, and in general the evaluations of $g(p_j - p_i)$ are not on grid nodes. This results in a dense matrix J.

Using Equation 4.8 would mean that Equation 4.6, although useful for theoretical analysis, could not be used for a practical algorithm. Matrix J, and hence M, is dense. Solving Equation 4.6 would require the solution to a dense linear system and a dense matrix multiply at each time step. We can avoid the dense linear system solution by using the iterative method described in Section 6, however, this still leaves us with a dense matrix-vector multiply.

There are fast methods which, under certain conditions, can be used to carry out Equation 4.8 (see [15]). If the sum is performed over a rectangular grid with constant spacing, then Equation 4.8 can be performed by exploiting the algebraic identity which uses the discrete Fourier transform (DFT), and in turn, the fast Fourier transform (FFT) ([7] §13). To exploit this approach, we create a two dimensional version of the method used in [15]. This method requires an interpolation of the original FD grid of values onto a DFT grid, a summation corresponding to Equation 4.8 by FFT, then an interpolation of the result back to the FD grid. Since all stages are second order accurate the approach satisfies our accuracy requirement.

There are several algorithms which can be used to determine the FFT of input data on unequally spaced grids [50, 17, 45]. These eliminate the need for interpolation between grids. However, some previous tests [13] indicate that these approaches were no more efficient than the simple interpolation strategy used here. We have previously experimented with a Fast Gauss Transforms [10] for g distributed as Normal, which also does not require a regular grid. However, this method did not appear to be any more efficient than FFT–based methods, at least for the order of accuracy required here.

4.3.1 DFT Domain and Grid

To apply the discrete Fourier transform to Equation 4.8, and hence allow the use of an FFT, we must compute the sum over a rectangular grid with constant grid line spacing. A rectangular domain Ω_D^* is defined with dimensions such that $\Omega_C, \Omega_D \subset \Omega_D^*$. The **DFT grid** over Ω_D^* is defined with $Q = Q_x \times Q_y$ nodes at the grid line intersections, and it tiles Ω_D^* with identically sized cells centered on those nodes. Nodes are denoted

$$q_k \in \mathbb{R}^2 \ , \quad k = 1 \ldots Q \ , \quad q_k \in \Omega_D^* \quad . \tag{4.9}$$

The integers Q_x and Q_y are chosen so that the FFT grid spacing h_1, h_2 is at least as fine as the log-transformed finite difference grid at the option strike node. In general $Q_x \neq Q_y$ and $h_1 \neq h_2$.

We define a vector of solution values

$$\mathsf{x} \in \mathbb{R}^Q \ , \quad \text{with elements } \mathsf{x_k} \approx W(q_k) \ , \quad q_k \in \Omega_D^* \quad \forall\, k = 1 \ldots Q$$

at nodes of the DFT grid. The matrix form of the integral operator of Equation 4.4 over the DFT grid is

$$\mathcal{J}_D W \approx \mathsf{J_r x} \ ; \quad \mathsf{J_r} \in \mathbb{R}^Q \times \mathbb{R}^Q \quad . \tag{4.10}$$

As in Equation 4.8, the correlation is integrated over the DFT grid with a cell-centered trapezoidal rule. If we set $\gamma_{ij} = 1$ then for point $q_k \in \Omega_C$ on the DFT grid

$$[\mathsf{J_r x}]_\mathsf{k} = \sum_{l=1}^{Q} f_g(q_l - q_k)\mathsf{x_k} \tag{4.11}$$

where f_g is given by

$$f_g(q_l - q_k) = \int_{-h_1/2}^{+h_1/2} \int_{-h_2/2}^{+h_2/2} g(q_l - q_k + (z_1, z_2))\, dz_1\, dz_2 \quad . \tag{4.12}$$

This ensures that the conditions given in Equation 4.7 are satisfied[2]. We note that the coefficients given by $f_g(q_l - q_k)$ are also on a regular grid and are identical surrounding each node q_k on the DFT grid.

4.3.2 Interpolating the FD and DFT Grids

A mapping is formed between a price vector $\mathsf{w} \in \mathbb{R}^P$ over the nodes of the FD grid in log-price scaling and the price vector $\mathsf{x} \in \mathbb{R}^Q$ on the nodes of the DFT grid, also in log-price scaling. The mapping can be written as a $Q \times P$ sparse matrix L so that $\mathsf{x} = \mathsf{L}\mathsf{w}$. The entries of L interpolate using a local, bi-linear Lagrange interpolation over the FD grid. We choose the interpolation stencil at each node so that $0 \le \mathsf{L_{ij}} \le 1$ and $\sum_\mathsf{j} \mathsf{L_{ij}} \le 1$. Where DFT grid point $q_i \in \Omega_D^*$ but $q_i \notin \Omega_D$ then $\mathsf{L_{ij}} = 0 \; \forall j$ to set $\mathsf{x_i} = 0$ (rather than extrapolate). We apply a bi-linear interpolation in the other direction using $P \times Q$ matrix K so that we can compute $\mathsf{w} = \mathsf{K}\mathsf{x}$.

To apply the discrete integral term over the FD grid, we approximate Equation 4.8 by

$$\mathsf{I_c J w} \simeq \mathsf{I_c} \left[\mathsf{K} \cdot \mathsf{J_r} \cdot \mathsf{L}\right] \mathsf{w} \tag{4.13}$$

where $\mathsf{J_r}$ of Equation 4.10 computes over the DFT grid. If h is the grid spacing on the DFT grid, then Equation 4.13 is an $O(h^2)$ approximation to Equation 4.8, which is the same order of error as the finite difference operators. Note that if $\mathsf{J_r}$ satisfies the conditions of Equation 4.7, then the construction of L and K preserves this result for $\mathsf{K} \cdot \mathsf{J_r} \cdot \mathsf{L}$.

4.3.3 Fast Fourier Computation of the Integral Term

The details of the reduction of Equation 4.11 to an operation involving the DFT is described in detail in standard texts (e.g. [7] §13). However, to exploit this method we require a new approximation $\mathsf{J_f}$ which is a Toeplitz matrix. $\mathsf{J_f} \neq \mathsf{J_r}$ because, in effect, we must replace $f_g(q_k)$ in Equation 4.11 with a periodic function

$$\begin{aligned} f_g^\circ(q_k) &= f_g(q_k) \quad \forall\, q_k \in \left(-\frac{Q_x}{2}h_1, -\frac{Q_y}{2}h_2\right) \times \left(\frac{Q_y - 1}{2}h_1, \frac{Q_y - 1}{2}h_2\right) \\ f_g^\circ(q_k) &= f_g^\circ(q_k + (a\, Q_x h_1, b\, Q_y h_2)), \quad \forall\, a, b \text{ integers.} \end{aligned}$$

We write the DFT as $\mathcal{D}$ and its inverse $\mathcal{D}^{-1}$ (see Appendix F). For grid points q_k, $k = 1 \ldots Q$ the Fourier transform form of Equation 4.11 is given by the identity

$$\left[\mathsf{J_f}\ \mathsf{x}\right]_\mathsf{k} = \frac{1}{Q}\, \mathcal{D}^{-1} \left[\hat{X} \cdot \overline{\hat{f_g^\circ}}\right](q_k) \tag{4.14}$$

where $\hat{X} = \mathcal{D}(X)$, $\hat{f_g^\circ} = \mathcal{D}(f_g^\circ)$ are multiplied at each node on the Fourier-space grid and $\overline{\hat{f_g^\circ}}$ is the complex conjugate of $\hat{f_g^\circ}$. By using an FFT to compute the DFT on the Q nodes, an $\mathrm{O}(Q^2)$ dense matrix multiplication is reduced to an $\mathrm{O}(Q \log(Q))$ operation. The scaling factor $1/Q$ is a side effect of the form of our DFT (Equation F.2).

Any solution using $\mathsf{J_f}$ will (typically near $\partial\Omega_D^*$) have been contaminated by values where $f_g^\circ(q_k - q_l) \neq f_g(q_k - q_l)$ because of the periodicity. Fortunately, our grid nesting strategy already dictates that we retain the more accurate values in the core domain $\Omega_C \subset \Omega_D^*$, and discard the rest of the computation. The final form of our approximation to Equation 4.8 is given by

$$\mathsf{I_c J w} \approx \mathsf{I_c} \left(\mathsf{K} \cdot \mathsf{J_f} \cdot \mathsf{L}\right) \mathsf{w} \tag{4.15}$$

where $\mathsf{J_f}$ is computed by applying Equation 4.14.

We shall use the approximation in Equation 4.15 in the iterative method described in Section 6.1 to solve time step Equation 4.6. We further control the wrap-around error by domain sizing methods discussed in Section 6.4 below.

[2] Where no CDF is available and the PDF g is sufficiently smooth, Equation 4.12 can be computed using a standard, high-accuracy numerical technique. We need evaluate $f_g(q_k)$ only once during the option pricing process for each grid node, so this does not incur an undue computational cost. Where the PDF is non-smooth, as with the Marshall-Olkin Bi-variate Exponential Distribution (see Section B.2) the integral must be done directly by evaluating the cumulative distribution.

4.3.4 Grid Alignment of the PDF

We note that

$$\int U(y+J)g(J-z_0)\,dJ = \int U(y+z_0+J)g(J)\,dJ$$

where $z_0 \in \mathbb{R}^2$ is an arbitrary shift of the jump PDF function g. This may be used to align discontinuities in a jump PDF to fall exactly between DFT grid nodes. A translation of the PDF can be corrected when the result for point y is interpolated back from the correlation by simply interpolating at $y+z_0$. Our DFT-based procedure is equivalent to the cell-centered integration rule. If PDF discontinuities can be aligned to fall on cell edges then the integration captures the discontinuity location exactly and the quadratic convergence of the integral can be preserved. This is particularly convenient for jumps of exponential types, where the continuous marginal probability distribution is defined with a peak point, and the two-dimensional probability distribution is the linear combination of a PDF in each of the four quadrants around the peak.

5 Stability

Definition 1 *Modern stability analysis (for example [28, 30, 48]) defines general categories of stability under an arbitrary norm $\|\cdot\|$ using a rational polynomial $\varphi(z)$ (e.g. as in Equation 4.6). Nomenclature varies, so we settle on the following names for three cases of interest. We have* ***algebraic stability*** *if $\|\varphi(\Delta\tau T)^n\| \le C\,p^\alpha\,n^\beta$ where the linear system has order $p \ge 1$, for time step $n \ge 1$, with $C, \alpha, \beta > 0$ independent of n and p. We have* ***strong stability*** *if $\|\varphi(\Delta\tau T)^n\| \le C$ for $C > 0$, and we have* ***strict stability*** *if $0 < C \le 1$.*

Definition 2 *If a matrix A has elements $a_{ii} > 0$ and $a_{ij} \le 0$ for $i \ne j$ and every row sum is non-negative with at least one row sum positive in each connected part of A, then A is an* ***M-matrix*** *(see [43, 47]).*

Remark 5.1 *If matrix A is an M-matrix then A^{-1} exists and $\mathsf{A}^{-1} \ge 0$ [43, 47].*

Definition 3 *If a matrix A has elements $a_{ii} \ge 0$ and $a_{ij} \le 0$ for $i \ne j$ and each row sum is non-negative then we say that A is* ***M-compatible****.*

Thus the sum of an M-compatible matrix and an M-matrix is an M-matrix. In Section 4.3.3 we saw that discrete integral operator matrix $\mathsf{J} \ge 0$ has maximum row sum $(\max_\mathsf{i} \sum_\mathsf{j} \mathsf{J}_\mathsf{ij}) \le 1$ and thus $\|\mathsf{J}\mathsf{x}\|_\infty \le \|\mathsf{x}\|_\infty$, $\|\mathsf{J}\underline{1}\|_\infty \le 1$ and $-(\mathsf{J}-\mathsf{I})$ is M-compatible.

Remark 5.2 *Let the conditions*

- *$r > 0$,*
- *$(-\mathsf{G})$ is M-compatible,*
- *$\mathsf{J} \ge 0$ and*
- *$(\max_\mathsf{i} \sum_\mathsf{j} \mathsf{J}_\mathsf{ij}) \le 1$ hold.*

Then, from Definition 2, M of Equation 4.5 is an M-matrix.

We shall show that if M is an M-matrix, then Crank-Nicolson time stepping is unconditionally algebraically stable in the l_∞ norm and, under a time step restriction, strictly stable. In this approach stability implies that at time step n the error $E_n = \bar{W}_n - W_n$ due to a perturbed solution $\bar{W}_n$ can be bounded in terms of the initial error $E_0 = \bar{W}_0 - W_0$ where $\bar{W}_0$ is a perturbed initial solution. We then show that Crank-Nicolson time stepping is unconditionally stable in the l_2 norm in the sense of von Neumann analysis. Under von Neumann analysis we determine the conditions under which the finite difference and integral operator reduce, rather than amplify, the l_2 norm of a perturbation error $E_n = \bar{W}_n - W_n$ as it propagates to time step $n+1$.

5.1 Stability in the l_∞ Norm, European Options

Theorem 5.1 *If M is an M-matrix then the time step method of Equation 4.6 is unconditionally algebraically stable in the l_∞ norm for $\theta = 1/2$ and unconditionally strictly stable in the l_∞ norm for $\theta = 0$.*

Equation 4.6 is strictly stable in the l_∞ norm for $\theta = 1/2$ if the time step is bounded using the maximum diagonal of M such that $[(\Delta\tau)/2]\max_{\mathsf{i}}(\mathsf{M}_{\mathsf{ii}}) < 1$.

Proof. For algebraic stability see Kraaijevanger et. al. [28], and for strict stability this result is proved by simple maximum analysis (e.g. as an extension of the result of [14]). □

Thus we can expect the finest grid spacing in the problem to determine the maximum Crank-Nicolson time step for which the solution is strictly stable in the l_∞ norm.

5.2 Stability in the l_∞ Norm, American Options

The above analysis refers only to European options. For American options, if M in equation (4.5) is an M-matrix, then it is straightforward to extend the analysis in [14] to show that fully implicit time stepping coupled with a penalty method [19] is unconditionally stable and monotone.

5.3 M-Compatibility of Finite Differences

Remark 5.3 *Negative coefficients arising from the use of a central difference scheme (e.g. Equation E.1) for the drift operator $\mathbf{V}$ can be made positive by replacing central differencing with either forward (Equation E.2) or backward (Equation E.3) differencing, as appropriate. Further details can be found in [55]. We take it as given, in the development that follows, that this change to the discrete drift term has been applied for proofs which require $(-\mathsf{G})$ to be M-compatible.*

For theoretical purposes, we assume in the following that the discretization is performed on a rectangular grid with constant grid line spacing. We denote the elements of the diffusion tensor $\mathbf{D}$ of Equation 2.6 as d_{ij} for $i, j = 1 \ldots 2$. We denote the grid spacing as (h_1, h_2) in the (y_1, y_2) direction.

Theorem 5.2 *Consider the seven-point finite difference approximation G_s to $\mathcal{G}$. The discretization is M-compatible if the following constraints hold.*

1. *We must select Equation E.5 if $\rho_v > 0$, and Equation E.6 if $\rho_v < 0$, for cross-partial derivatives.*
2. *For each point on the grid*

$$\frac{d_{ii}(y)}{h_i} > \left|\frac{d_{ij}(y)}{h_j}\right| \quad , \quad i \neq j \quad . \tag{5.1}$$

With these conditions and where Remark 5.3 holds, $-\mathsf{G} = -\mathsf{G}_s$ is M-compatible and satisfies Remark 5.2 so that the stability statements of Theorem 5.1 hold for $r > 0$.

Proof. See [5] and [41] §9.4. □

Remark 5.4 *Consider the finite difference approximation G_n to $\mathcal{G}$ on a nine-point stencil with the four-point second order cross-partial derivative given by Equation E.7. This formulation results in negative off-diagonal coefficients in G_n for $\rho_v \neq 0$. Thus this discretization does not result in $(-\mathsf{G})$ M-compatible for correlation $\rho_v \neq 0$ and Theorem 5.1 cannot be shown to hold by the M-matrix approach.*

5.4 Von Neumann Stability Analysis for European Options

The von Neumann stability analysis examines a problem over a periodic domain to determine conditions for the stability of its discrete operators under the l_2 norm ([16] §8.3 and [46] §6.8).

Theorem 5.3 *Consider a periodic formulation of Equation 3.2, discretized with a finite difference approximation on a grid with constant spacing. The problem is formed with constant coefficients* $\mathbf{D}$ *and* $\mathbf{V}$ *(Equations 2.6 and 2.5),* $-1 \leq \rho_v \leq 1$ *and* $\lambda_C = \lambda \geq 0$ *constant. We may use the cross-partial finite difference of Equations E.5 or E.6 to form the approximation* G_s *to* $\mathcal{G}$ *on a seven-point stencil, or use Equation E.7 to form* G_n *over a nine-point stencil.*

The time step Equation 4.6 is unconditionally von Neumann stable in the l_2 *norm for* $\theta = 0$ *and for* $\theta = 1/2$.

Proof. See Appendix F, in particular Appendix F.4 Remark F.1. □

5.5 Stability Summary

Our numerical approach can deviate from the theoretical conditions for stability. The FD method we employ for grids with non-constant spacing has the same conditions for M-compatibility as Theorem 5.2. However, the structure of a computationally efficient FD grid is not always such that these grid spacing conditions are met. Thus we may not be able to guarantee the conditions for M-compatibility at every point in the solution domain, although the conditions will often be met locally in the region of interest near the strike price of the option. In turn, this has implications for both European and American options: we cannot globally guarantee the conditions for l_∞ norm stability. This issue was studied in [55] for the pure diffusion case, where it was shown that if the option value is Lipschitz continuous then coefficients in the discretization which are not M-compatible cause at worst an $\mathrm{O}(h)$ error. The convergence of the method was demonstrated numerically. We note, also, that the contribution to the linear system from the integral term works in favour of stability: it tends to correct, rather than worsen, the problem of the differential term not being M-compatible.

The von Neumann stability analysis fails to apply to our method for European options where the grid spacing is not constant in each direction. We also use a non-constant λ_C, although we note that the analysis demonstrates stability where either $\lambda = 0$ or $\lambda > 0$ over the entire domain. In the region of interest for the problem, the grid spacing we use for numerical demonstrations will be constant only in price scaling, and not over the entire problem domain or in log-price scaling. Nonetheless, the von Neumann analysis provides an even less restrictive result than the l_∞ norm analysis, indicating strict l_2 norm stability regardless of the ratio of grid spacing between the axes, the time step and the choice of discretization for the cross-partial derivatives.

The numerical demonstrations in Section 7 show our method is quadratically convergent for European and American options despite the violations of the theoretical stability conditions that we identify. Stability can be guaranteed by employing a rotated co-ordinate system which eliminates the correlation in the Brownian motion. However, we shall see in Section 7.4 that this approach results in higher errors in the solution.

6 Solution of the Discrete Equations

The previous sections show how PIDE 2.3 has been localized to Equation 3.2 and discretized. In this section we focus on how the time step Equation 4.6 is solved using a fixed point iteration, and how American option values are computed using a penalty method.

6.1 Fixed Point Iterative Solution for One Time Step

As a motivation for our fixed point iteration scheme we consider the semi-discretized Equation 4.2. We analyze a fixed point iteration scheme whereby the integral terms are handled in an iterative manner which avoids having to use the Green's and Poisson functions of the PIDE. Rather, only the Green's and Poisson functions of the differential terms are required.

6.1.1 The Semi-discretized Equations

Let Z^k be the k-th iterate towards a solution W^{n+1} of Equation 4.2. One step of the iteration we shall use is given by

$$\begin{aligned}\left[1-(1-\theta)\Delta\tau\,(\mathcal{G}-r-\lambda_C)\right]Z^{k+1} &= (1-\theta)\Delta\tau\,\lambda_C\mathcal{J}_D Z^k \\ &\quad + \left[1+\theta\Delta\tau\,(\mathcal{G}+\lambda_C\mathcal{J}_D-r-\lambda_C)\right]W^n\end{aligned} \tag{6.1}$$

which is repeated until convergence.

Theorem 6.1 *Let $E^k = W^{n+1} - Z^k$ be the error in the solution to the semi-discretized Equation 4.2 at iteration k of the functional iteration given in Equation 6.1. The iteration is convergent to zero as*

$$\|E^{k+1}\|_\infty \le \frac{(1-\theta)\Delta\tau\,\lambda}{1+(1-\theta)\Delta\tau\,(r+\lambda)}\|E^k\|_\infty \quad .$$

Proof. See Appendix D.1. □

6.1.2 The Fully Discrete Equations

We now consider the solution of the fully discrete problem in Equation 4.5. We wish to avoid having to invert any matrix such as M of Equation 4.5 formed by a sum containing the dense matrix J. To do so we use the discrete version of the iteration of Equation 6.1.

Let z^k be the k-th iterate towards a solution w^{n+1} of Equation 4.6. We specify the fixed point iteration

$$\begin{aligned}\left\{\mathsf{I}-(1-\theta)\Delta\tau\left[\mathsf{G}-r\,\mathsf{I}-\lambda\mathsf{I_c}\right]\right\}\mathsf{z}^{k+1} &= (1-\theta)\Delta\tau\,\lambda\mathsf{I_c}\mathsf{J}\mathsf{z}^k \\ &\quad + \left\{\mathsf{I}+\theta\Delta\tau\left[\mathsf{G}+\lambda\mathsf{I_c}(\mathsf{J}-\mathsf{I})-r\,\mathsf{I}\right]\right\}\mathsf{w}^n+\mathsf{b} \quad .\end{aligned} \tag{6.2}$$

We compute $\mathsf{I_c}\mathsf{J}\mathsf{z}^k$ using the method described in Equation 4.15.

Theorem 6.2 *Let $\mathsf{e}^k = \mathsf{w}^{n+1} - \mathsf{z}^k$ be the error in the solution to Equation 4.6 at iteration k of the fixed-point iteration given in Equation 6.2. If $\mathsf{J} \ge 0$ has maximum row sum $(\max_\mathsf{i}\sum_\mathsf{j}\mathsf{J}_\mathsf{ij}) \le 1$ and $(-\mathsf{G})$ is M-compatible with $\|\mathsf{G}\cdot\underline{1}\|_\infty = 0$, then the error in the iterative solution z^{k+1} in Equation 6.2 is convergent to zero as*

$$\|\mathsf{e}^{k+1}\|_\infty \le \frac{(1-\theta)\Delta\tau\,\lambda}{1+(1-\theta)\Delta\tau\,(r+\lambda)}\|\mathsf{e}^k\|_\infty \quad .$$

Proof. $\|\mathsf{G}\cdot\underline{1}\|_\infty = 0$ should hold for any consistent finite difference approximation. See Appendix D.2. □

Theorem 6.3 *Consider a periodic formulation of Equation 3.2, discretized with a finite difference approximation. Let $(-\mathsf{G})$ be formed either by the 7-point or 9-point finite difference stencil on a grid with constant spacing, as in Theorem 5.3.*

Then in the sense of von Neumann analysis the iterative solution to Equation 4.6 by Equation 6.2 is unconditionally convergent in the l_2 norm (i.e. regardless of whether $(-\mathsf{G})$ is M-compatible) at a rate which is rapid if $\lambda\Delta\tau \ll 1$.

Proof. See Appendix F, in particular Appendix F.4 Remark F.2. □

For most practical situations we have $\lambda\Delta\tau \ll 1$ and we can expect rapid convergence. To summarize, we have the following results.

- The functional fixed-point iteration Equation 6.1 for the semi-discrete Equation 4.2 is convergent in the l_∞ norm. This suggests that, for a sufficiently fine grid, the discrete iteration should also be convergent.
- If $(-\mathsf{G})$ is M-compatible then by maximum analysis the discrete fixed point iteration is convergent in the l_∞ norm.
- By von Neumann analysis, the iteration is convergent in the l_2 norm for a periodic problem with constant grid spacing, with no restrictions on grid spacing ratio or time step, using any standard second-order finite difference approximation of the differential operators.

6.2 American Options by Penalty Iteration

To solve the discrete, localized version of the linear complementarity problem of Equations 2.10 and 2.11 we use the penalty iteration of [53, 19]. In [14] this method was shown to be l_∞ stable for jump diffusion processes provided that the discrete diffusion portion of the process was M-compatible, and that iterations of the form of Equation 6.2 should be rapidly convergent. If the discretized diffusion operator is an M-matrix then, by maximum analysis, for fully implicit time stepping the method is l_∞ stable and monotone. Since a consistent scheme is used for the differential and integral terms, we can expect convergence to the viscosity solution of the localized problem [42, 8, 9, 14]. We note that the concept of a viscosity solution permits non-smooth solutions.

We define a penalty vector $\mathsf{c_p}$ with elements

$$(\mathsf{c_p})_\mathsf{i} = \begin{cases} \mathit{Large} & \text{if } \mathsf{w}^*_\mathsf{i} > \mathsf{w_i} \\ 0 & \text{if } \mathsf{w}^*_\mathsf{i} \leq \mathsf{w_i} \end{cases} \tag{6.3}$$

where $\mathsf{w}^*_\mathsf{i} = \mathcal{I}(p_i)$ is the vector of option payoff values, the minimum value of an American option at any time. The value *Large* is chosen sufficiently large to impose the condition without causing numerical inaccuracy; a value of $\mathit{Large} \approx 10^5$ is usually appropriate.

To impose the American constraint, we solve a modified version of Equation 4.6 by iterating from $\mathsf{z}^0 = \mathsf{w}^n$ for solutions z^{k+1} for $k = 0$ to convergence. The penalty iteration is incorporated into our fixed point iteration Equation 6.2 without adding another level of iteration. The resulting non-linear system solution approach is given in Algorithm 6.1 at Steps 3 through 5.

Intuitively, we may think of the penalty iteration as the adaptive imposition of a Dirichlet free boundary condition. The same approach can be used to impose a maximum value on an option.

6.3 Linear System Solution

Each iteration towards the solution of a time step requires solving the linear system given in Equation 6.2. In the case of American options, the linear system also contains a penalty constraint as in Step 5 of Algorithm 6.1. For one-factor options, a direct solution method based on Gaussian elimination is suitable. For two-factor options a direct method would be unacceptably expensive, thus we use a preconditioned, Krylov-subspace, iterative method. Bi-CGStab was selected combined with an ILU(1) preconditioner and RCM re-ordering [14, 47]. We consider a linear system solution to be converged when the Bi-CGStab update to the solution value is, pointwise, less in magnitude than the relative tolerance ϵ_l.

The entire non-linear system is solved to an update tolerance of ϵ_u, as given in Step 6 of Algorithm 6.1. The constant one in the denominator of the convergence test is set assuming that options are priced in dollars. This ensures that the convergence requirement does not become extreme for grid points with small option values. Section 7.5 discusses the actual amount of computation required for each time step.

6.4 Sizing Ω_D and Ω_C for Error Control

The error generated when Equation 2.3 is truncated on Ω_D to Equation 3.2 is difficult to characterize in general. It depends on the jump distribution g and the option payoff. We can, however, estimate the

Algorithm 6.1 Solve one time step using a simultaneous fixed point, penalty iteration.

```
FixedPointIteration( $w^n$, $w^*$, $\theta$, $\Delta\tau$, M, G, J, $\lambda I_c$, b, $\epsilon_u$ )
where
        $w^n$                     the price at time step $n$
        $w^*$                     the minimum option value (usually the payoff)
        $\theta$                  time step weight
        $\Delta\tau$              time step size
        $M, G, J, \lambda I_c$    the discrete PIDE to be solved, Equation 4.5
        $b$                       boundary condition imposition vector
        $\epsilon_u$              required solution update tolerance

1.  Set $z^{k=0} = w^n$.

2.  For $k = 0, 1, 2, 3, \ldots$ until convergence (tested in Step 6)

3.      For American options: set ${c_p}^k$ using Equation 6.3 where $w = z^k$.
        For European options: set ${c_p}^k = 0$.

4.      Set      $N^k = [I - (1-\theta)\Delta\tau\ (G - r\,I - \lambda I_c)] + ({c_p}^k)' I$
                 $y^k = [(1-\theta)\Delta\tau\ \lambda\, I_c J]\, z^k + [I - \theta\Delta\tau\ M]\, w^n + b + ({c_p}^k)' I\, w^*$
        where    $I_c J z^k \approx I_c\, (K \cdot J_f \cdot L)\, z^k$.                (see Equation 4.15)

5.      Solve    $N^k\, z^{k+1} = y^k$   using ILU(1) preconditioned, Bi-CGStab.   (see Section 6.3)

6.      If $\max_i \frac{\|z^{k+1} - z^k\|_i}{\max(1, \|z^{k+1}\|_i)} < \epsilon_u$ then the iteration is finished.

    End For

Return the solution vector $w^{n+1} = z^{k+1}$.
```

maximum error incurred for a single computation of the integral term $\mathcal{J}_D V$ of Equation 4.1. We keep in mind that our computation of the integral by the DFT approximation of Equation 4.14 causes a "wrap-around" that may cause the maximum of the option value V to factor into the error. We formulate the following rule of thumb by which we check that the distance between $\partial\Omega_C$ and $\partial\Omega_D$ is wide enough to control this error. We use the initial value $\mathcal{I}(y)$, $y \in \Omega_\infty$ as an approximation of V when we estimate a maximum option value.

Let u be the distance from $\partial\Omega_C$ to $\partial\Omega_D$ along axis y_i, $i = 1, 2$ at point $y \in \partial\Omega_C$ with outward facing unit vector e_i. We choose that distance u based on the marginal jump distribution g_i in the e_i direction and the initial value $\mathcal{I}$ such that a selected integral evaluation error tolerance ϵ_i satisfies

$$\max_{y \in \partial\Omega_C} \left[\int_u^\infty g_i(v\, e_i)\, \mathcal{I}(y + v\, e_i)\; dv \right] \leq \max_{y \in \partial\Omega_D} [\mathcal{I}(y)] \int_0^\infty g_i((u+v)\, e_i)\, e^{\alpha v}\; dv < \max_{y \in \partial\Omega_D} [\mathcal{I}(y)]\ \epsilon_i \tag{6.4}$$

where $\alpha = 0$ in the case of puts or the lower boundary of a call, and $\alpha = 1$ for the upper boundary of a call.

Using the marginal jump distribution and the diffusion parameters, a European solution in 1D can be computed [31] as well as a cumulative distribution function for the price process at the expiry time. As a rule of thumb to ensure that Ω_C is adequately large, we ensure that Ω_C extends by widths w_1 and w_2 in the y_1 and y_2 directions around the option strike K so that $(\log(K), \log(K)) \pm (w_1, w_2) \in \Omega_C$. Widths w_1 and

	Jump Distribution	
Diffusion Parameter	**Normal**	**MOBED**
σ_1	0.12	0.12
σ_2	0.15	0.15
ρ	0.30	0.30
Jump Parameter		
λ	0.60	0.50
$\breve{\mu}_1$	-0.10	0.00
$\breve{\mu}_2$	0.10	0.00
$\breve{\rho}$	-0.20	
$\breve{\sigma}_1$	0.17	
$\breve{\sigma}_2$	0.13	
$\breve{p}_1$		0.40
$\breve{p}_2$		0.60
$\breve{\eta}_{p,1}$		1/0.20
$\breve{\eta}_{p,2}$		1/0.18
$\breve{\eta}_{q,1}$		1/0.15
$\breve{\eta}_{q,2}$		1/0.14
$\breve{\eta}_{pp}$		1/0.15
$\breve{\eta}_{qq}$		1/0.16
$\breve{\eta}_{pq}$		1/0.12
$\breve{\eta}_{qp}$		1/0.15

TABLE 1: *The two price processes used for the numerical examples in this paper. These are not taken from actual market values, but represent parameter values in a plausible range for a market process. See Section B for the definitions of the model coefficients. Solutions were computed with* $r = 0.05$*, strike* $K = 100$ *and expiry* $T = 1.0$*.*

w_2 must be large enough that

$$\int_{-w_1}^{w_1} f_1(v_1;T)\,d\,v_1 > (1-\epsilon_w) \quad \text{and} \quad \int_{-w_2}^{w_2} f_2(v_2;T)\,d\,v_2 > (1-\epsilon_w)$$

where f_1 and f_2 are the marginal distributions of the entire jump diffusion process at T and ϵ_w is a bound on the cumulative distribution. These distributions can be computed numerically where the characteristic function of the jump PDF is available, or simply estimated using a Normal distribution of the same variance as the total asset price process. The reasoning behind this rule is to set Ω_C sufficiently large that the influence of solution details outside Ω_C have approximately ϵ_w proportional influence at the strike, at expiry time.

The FD grid defines the upper bound of $\Omega_{\widetilde{D}}$ and hence of Ω_D. Given this, a choice for the upper boundary of Ω_C may be selected by these rules of thumb, and given a choice for the lower boundary of $\Omega_{\widetilde{C}}$ the lower boundary of Ω_D may be set similarly. A good automatic choice of the lower bound of $\Omega_{\widetilde{C}}$ is the grid line closest to, but greater than, either of the zero axes in price scaling. For our demonstrations we require more control, so we shall specify the bounds of Ω_C and lower bound of Ω_D without using these grid lines or the rules of thumb.

7 Numerical Demonstrations

For our numerical demonstrations we use three option contracts with two different types of jump diffusion. We test the convergence as $\Omega_C, \Omega_D \to \pm\infty$ and demonstrate the quadratic convergence of both European and American options as the number of discrete solution nodes is increased in time and log-price scaling.

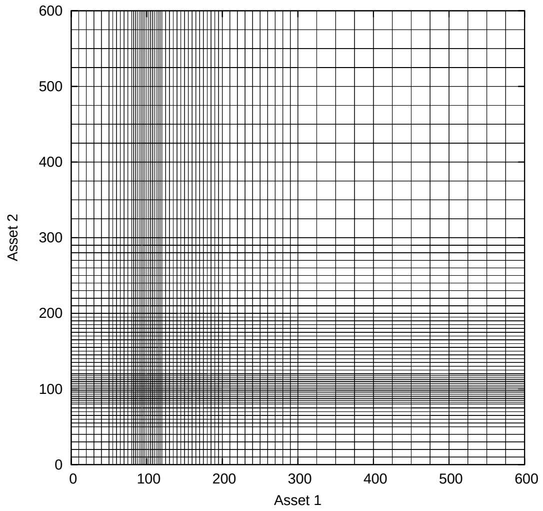

Figure 3: *This coarse grid in price scaling over $\Omega_{\widetilde{D}}$ shows grid line concentration in the $\Omega_{\widetilde{C}}$ region around the strike of 100.0. The actual grids were extended very coarsely to 600.0 to capture enough of the solution to control the error from the jump diffusion computation.*

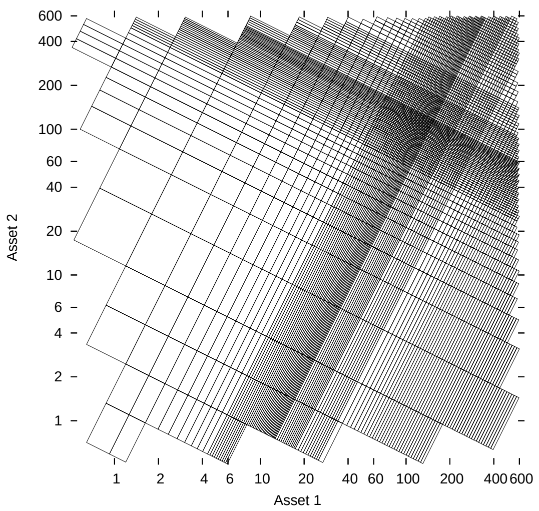

Figure 4: *This coarse grid in log-price scaling over Ω_D, with node concentration in the Ω_C region around the strike of 100.0, is rotated by -26.57 degrees around the strike node. This ensures that the fully implicit finite difference approximation to the problem specified in Table 1 is unconditionally stable.*

7.1 Sample Problem

We choose two jump PDF functions: the bi-variate Normal and the Marshall-Olkin Bi-variate Exponential Distribution (MOBED). Both are described in Appendix B. As noted in the introduction, these correspond in form to the well-known, one-asset models of Merton and of Kou [39, 27]. Table 1 lists our model coefficients, which are of a magnitude that would be plausible in a real market.

As a demonstration we solve for a European call on the maximum of two assets, and a European and American put on the minimum of two assets. We select a strike $K = 100$, expiry of $T = 1.0$ and a risk free rate $r = 0.05$. For the numerical solution of the European options we compare the fully numerical solution to a high-accuracy, semi-analytic solution computed using the method described in [18].

The region near (K, K) was discretized with a constant grid spacing in price scaling. Crank-Nicolson time stepping was used with a constant time step $\Delta\tau$, so that convergence could be demonstrated with respect to a controlled amount of computational effort. Variable grid spacing [44] or time stepping [54] can provide computational savings, but this study does not investigate these issues. Figure 3 shows the coarsest grid used for the demonstrations of Section 7.3 below. Over $\Omega_{\widetilde{D}} \setminus \Omega_{\widetilde{C}}$ the solution is expected to be mostly piecewise linear, hence it remains only coarsely resolved. Although regions of the problem generated discrete equations which did not result in an M-matrix, the overall problem solution proceeded with no actual numerical instability detected in the region of interest around the strike. This is consistent with previous efforts [55] for pure diffusion models.

The constant grid spacing $(\Delta S_1, \Delta S_2)$ near the strike node at (K, K) is used to define our FFT grid. We select the smallest integers Q_x and $Q_y = 2^a\, 3^b\, 5^c\, 7^d$, $a, b, c, d \in \mathbb{Z}$ where $a \geq 1$, $b, c, d \geq 0$ (as dictated by our choice of FFT solution package[3]) such that the DFT grid of $Q = Q_x \times Q_y$ nodes matches the log-scaled FD grid spacing near the strike. Recall that $Q_x \neq Q_y$ in general, although in the following experiments the two values will be equal.

[3] The FFTW library, available at `http://fftw.org`, implements an efficient Winograd transform algorithm.

Grid Range		**Normal**			**Difference vs. Largest Ω_D**		
(Price Scale)	S_2	$S_1 = 90$	100	110	90	100	110
$\Omega_C : 20 \to 160$	90	15.6811	13.3970	12.1132	-0.0031	-0.0064	-0.0154
$\Omega_D : 4 \to 300$	100	12.1763	9.1178	7.4938	-0.0084	-0.0132	-0.0214
DFT 384×384	110	10.3573	6.6910	4.7925	-0.0218	-0.0301	-0.0379
$\Omega_C : 10 \to 220$	90	15.6841	13.4032	12.1279	-0.0001	-0.0002	-0.0006
$\Omega_D : 2 \to 400$	100	12.1846	9.1308	7.5145	-0.0000	-0.0001	-0.0006
DFT 480×480	110	10.3793	6.7215	4.8302	0.0003	0.0003	-0.0001
$\Omega_C : 5 \to 300$	90	15.6842	13.4034	12.1282	0.0000	-0.0000	-0.0004
$\Omega_D : 1 \to 600$	100	12.1849	9.1312	7.5150	0.0002	0.0002	-0.0002
DFT 576×576	110	10.3800	6.7224	4.8314	0.0009	0.0013	0.0010
$\Omega_C : 2.5 \to 400$	90	15.6842	13.4034	12.1286			
$\Omega_D : 0.5 \to 900$	100	12.1847	9.1309	7.5152			
DFT 672×672	110	10.3791	6.7211	4.8303			

Table 2: *To show the effects of extending the domain, we use the European put on the minimum of two assets using the Normal jumps model. The Ω_C and Ω_D ranges are given in the left column in price scaling, along with the DFT grid size which most closely matched the $\Delta S_1 = \Delta S_2 = 1.25$ at the strike. The time step was fixed at $\Delta\tau = 0.02$. Crank-Nicolson time stepping was used. Points at $S_1, S_2 = \{90, 100, 110\}$ are given for each grid. The three larger grids were formed by extending the next smallest grid with additional lines. The error is measured against the solution over the largest domain $\Omega_D : 0.5 \to 900$. As the upper and lower limits of Ω_C and Ω_D are extended, the difference tends to diminish, most noticeably between the first two tests.*

The iteration of Algorithm 6.1 was solved until the maximum relative update to any solution node in Ω_C was $\epsilon_u = 10^{-6}$. The convergence tolerance for the linear system solution was $\epsilon_l = 10^{-8}$ (Section 6.3). For the grid refinement tests below, the "rule of thumb" tolerances (Section 6.4) were employed to check that the size of the grid and domains gave a Ω_D sizing tolerance of at least $\epsilon_i = 10^{-3}$ and the Ω_C sizing tolerance was at least $\epsilon_w = 10^{-2}$.

7.2 Convergence with $\Omega_C, \Omega_D \to \Omega_\infty$

As a partial demonstration of Theorem 3.1, we computed four solutions to the European put on the minimum of two assets using grids of increasing size and Normally distributed jumps. The four ranges for Ω_C and Ω_D are given in price scaling in the left column of Table 2.

In order to focus on the effect of the localization error, all grids had a spacing at the strike of $\Delta S_1 = \Delta S_2 = 1.25$, and each larger grid was formed as a simple extension of the previous one. In other words, we are not attempting to converge to the exact solution, but we are examining the effect of the localization error for a fixed grid spacing. The time step was $\Delta\tau = 0.02$. The DFT grids were set to match the FD grid spacing near the strike.

The option values given in Table 2 also show the difference measured against the solution on the largest domain. We note that the first two grids are, using the ϵ_i specified in our rule of thumb above, somewhat too small. Since our convergence theorem does not address specific cases we can only note that the error tends to diminish as the domains are extended. Very little difference is to be noted between the computations over the three largest domains. Note that since we are keeping the grid size constant near the strike, there will be some error introduced due to interpolation on different sized DFT grids.

Theorem 3.2 notes that, as the domain size increases, a perturbed boundary condition should generate a smaller error in the solution. The computations of Table 2 were repeated with a +50% lower Dirichlet boundary condition error. No significant difference to the values reported in Table 2 was noted. Recall that in the $\Omega_D \setminus \Omega_C$ region only the diffusion equation is solved. The rapid decay of error from the boundary condition is consistent with the results of [25, 54]. From these results we conclude that the propagation of that error into the interior domain by the jump process is too small to note.

7.3 Quadratic Convergence with Grid Refinement

We examined convergence with respect to grid and time step refinement using $\Omega_D = (1, 1) \times (600, 600)$, $\Omega_C = (5, 5) \times (300, 300)$ (in price scaling). We started with a coarse grid with a spacing of $\Delta S_1 = \Delta S_2 = 2.5$ in a region around the strike as shown in Figure 3. For each of the next two grids we refined the grid spacing by two, doubling the number of finite difference grid lines in each direction. At each grid refinement, we also halved the time step from the coarse grid value of $\Delta\tau = 0.04$. Again, the DFT grid refinement was set so that at the strike the two grids had, as near as possible, the same spacing in each direction.

The grid sizes and the computed results for the European call on the maximum are given in Table 3. We report the results for the MOBED jumps model. In this case, we can obtain a semi-analytic solution using Fourier methods [18], which allows us to determine the error in our numerical scheme. If ϵ is the error in the numerical solution, we assume that $\epsilon = ch^\alpha$ where h is the grid spacing with $\Delta\tau = \mathrm{O}(h)$, and report the convergence exponent α. The asymptotic convergence of the price is roughly quadratic, which we compute from the error values ϵ_2 and ϵ_3 on the two finest grids. The result from the coarse grid solution at the strike was unusually accurate, which we do not expect to be typical of our approach.

In this experiment, and those that follow, the integral and differential computations are converging on different scales. We refine the differential computation in price-scaling on the FD grid, and match the DFT grid spacing to it at the strike. However, the DFT grid has constant spacing in log-price scaling and has restrictions on the number of nodes we can use along each axis. We may therefore expect that the order of the convergence, which we calculate with respect to the price-scale refinement of the grid spacing, may fluctuate around the ideal of quadratic. When we compute a problem without jumps we obtain a convergence exponent $1.97 \leq \alpha \leq 2.05$. This indicates that much of the deviation from quadratic convergence is due to the jump calculation.

7.4 Convergence with Rotated Coordinates

As noted in [55], rotation of the coordinate system and finite difference grid in log-price scaling by

$$\theta_r = \frac{1}{2} \tan^{-1} \left(\frac{2\rho_v \sigma_1 \sigma_2}{{\sigma_1}^2 - {\sigma_2}^2} \right)$$

will result in a correlation $\rho_{vr} = 0$ in the diffusion tensor of the rotated system. The cross-partial derivative is thus eliminated and the FD approximation is then M-compatible. Note that in this case, fully implicit time stepping results in a monotone, consistent and stable method. Consequently, convergence to the viscosity solution is guaranteed [8]. As well, both the fixed point and penalty iteration are also guaranteed to be globally convergent. From a theoretical point of view, this is highly beneficial.

We performed a set of computations with a rotated grid, shown in Figure 4. The lower boundary condition specified in Equation 3.6 was not applied on the rotated grid since this would have brought back potential violations of the theoretical stability conditions. Instead, a Dirichlet boundary condition was imposed on all of $\partial\Omega_D$ using the initial conditions. To estimate the effect this had on the solution, the unrotated grid tests were repeated with this new lower boundary condition. The absolute difference in the solution at the strike was less than 10^{-6}. Thus we concluded that, for this problem, this boundary condition approximation is acceptable for the rotated grid case.

Previous research [55] has shown that, for the pure diffusion case, rotating the co-ordinate systems and grids produces a solution which is less accurate than the unrotated computation for the same grid spacing. In the rotated co-ordinates, initial conditions and barriers cannot be represented exactly and points at asset values of interest usually require interpolation. The grid rotation ensured that there was a node at the

Grid		**MOBED**			**Absolute Error**		
	S_2	$S_1 = 90$	100	110	90	100	110
$\Delta S_1 = \Delta S_2 = 2.5$	90	5.9683	10.1902	17.1677	-1.23e-3	-4.82e-3	4.36e-3
4356 Nodes, $\Delta\tau = 0.04$	100	10.5155	13.6119	19.3080	4.24e-3	2.52e-4	6.98e-3
DFT 288×288	110	17.4414	19.3702	23.4029	1.29e-2	1.02e-2	1.30e-2
$\Delta S_1 = \Delta S_2 = 1.25$	90	5.9693	10.1943	17.1650	-2.59e-4	-7.53e-4	1.73e-3
17161 Nodes, $\Delta\tau = 0.02$	100	10.5126	13.6124	19.3038	1.32e-3	8.28e-4	2.82e-3
DFT 576×576	110	17.4321	19.3635	23.3945	3.66e-3	3.48e-3	4.64e-3
$\Delta S_1 = \Delta S_2 = 0.625$	90	5.9695	10.1949	17.1638	-4.60e-5	-1.66e-4	4.49e-4
68121 Nodes, $\Delta\tau = 0.01$	100	10.5116	13.6118	19.3017	3.54e-4	2.37e-4	7.33e-4
DFT 1134×1134	110	17.4294	19.3609	23.3911	9.30e-4	8.95e-4	1.20e-3
Convergence	90				2.49	2.18	1.94
Exponent α	100				1.90	1.80	1.94
Grid 2 to 3	110				1.98	1.96	1.96

Table 3: *Numerical solution of the European call on a maximum of two assets with the parameters given in Table 1. We report the MOBED model with its absolute error. Note that the asymptotic convergence, reported here for the two finest grids, is approximately the ideal quadratic* $\mathrm{O}((\Delta\tau)^2 + (\Delta S)^2)$ *as the grid and time step is refined. Crank-Nicolson time stepping was used. The error was computed by comparison with the semi-analytic solution [18].*

strike $(S_1, S_2) = (100, 100)$ in common with the original grid, but other nodes did not line up with points of non-smoothness of the payoff. The consequences of this are clearly seen in Table 4. Note that the region where the grid was most highly refined was slightly enlarged to ensure that the region around the strike remained well resolved.

If we compare Table 4 with the results for the conventional grid given in Table 3, we can see that the error is larger for the rotated grid at the same mesh size, significantly so for the coarse and medium grids. The solutions generated by the conventional grids generated no actual instabilities. Using grid rotation to to ensure that the discretization of the diffusion terms yields an M-matrix seems unjustified in a practical setting. In any case, if the coefficients of the PIDE are not constant, grid rotation may not guarantee that the discrete diffusion operator is M compatible.

7.5 Quadratic Convergence of American Options

We repeated our demonstrations with an American put on the minimum of two assets for both the Normal and MOBED jump distributions. We used the same domain, spacing, set of grids and time steps as in Section 7.3. Crank-Nicolson time stepping was used. The results are given in Table 5 for nine points which are outside the region where the American minimum value constraint is imposed. In this case the exact solution is not available. Consequently, we assume that the error $\epsilon = ch^\alpha$ with grid spacing h and $\Delta\tau = \mathrm{O}(h)$, and compute the convergence exponent by examining the ratio of the difference in the computed option values for three mesh sizes. The α exponent is reported at the same point on the three grids and is approximately 2 in all cases.

Figure 5 shows the results over the core of $\Omega_{\widetilde{C}}$ in price scaling for the option using MOBED jumps. The two disconnected, dark regions on the surface represent areas where the penalty method has imposed the American minimum constraint on the solution, usually called the "early exercise" region. In this region the numerical error is controlled by *Large* of Equation 6.3 and is not significant at the grid nodes. The spatial location of the free boundary between the constrained and unconstrained region is resolved to within the grid spacing. Along the $S_1 = 100$ line the boundary of the American payoff is, for the MOBED problem, at

Rotated	**MOBED**	**Absolute**
Grid	$(100, 100)$	**Error**
$\Delta S_1 = \Delta S_2 = 2.5$, $\Delta\tau = 0.04$ 6847 Nodes, DFT 336×336	13.6221	1.05e-2
$\Delta S_1 = \Delta S_2 = 1.25$, $\Delta\tau = 0.02$ 26891 Nodes, DFT 672×672	13.6142	2.64e-3
$\Delta S_1 = \Delta S_2 = 0.625$, $\Delta\tau = 0.01$ 106861 Nodes, DFT 1344×1344	13.6123	6.76e-4
Convergence Exponent α, Grid 2 to 3		1.97

Table 4: *Numeric solution to the European call on a maximum of two assets with the parameters given in Table 1. We report the MOBED model with its absolute error using a grid rotated -26.57 degrees to guarantee that the finite difference approximation is M-compatible. Crank-Nicolson time stepping was used. Convergence is asymptotically quadratic, but the absolute error indicates that the rotation of the grid incurs an increase in absolute error, compared with Table 3.*

$S_2 = 82.5$ for all three grid resolutions. Along the $S_2 = 100$ line, the boundary is at $S_1 = 85$ for the coarsest grid ($\Delta S_1 = 2.5$), then at 83.75 and 84.375 for the two finer grids. In both the Normal and MOBED test cases, each refinement of the grid placed the boundary within $(\Delta S_1, \Delta S_2)$ of its location on the coarser grids.

Table 6 shows the total number of fixed point and linear solver iterations required for the entire solution of the put on the minimum of two assets in both the European and American cases. As predicted for the European case in Section 6.1, the number of fixed-point iterations required to advance a single time step diminishes with $\Delta\tau$: an average of 3 iterations were required with $\Delta\tau = 0.04$ but only 2 when $\Delta\tau = 0.01$. For an American option, on average, between 3.40 and 4.00 iterations of the fixed point algorithm were required. The addition of the penalty method to the fixed point iteration caused the iterations required to be roughly the same at each grid resolution. For both options, an average of between 2.24 and 4.00 linear solver iterations were required to converge to the solution of the equation in Step 5 of Algorithm 6.1. Thus we see that the rapid convergence indicated by our analyses in Section 6.1 is achieved in the actual tests, that the penalty iteration incurs only modest additional work, and that each linear system is fairly easy to solve.

8 Conclusions

We have developed an implicit method for computing the solution of the PIDE which gives the solution of a two asset option pricing problem under jump diffusion. A naive implicit computation of the integral term in the PIDE would involve the solution of a dense linear system. However, the use of a fixed point iteration reduces this problem to carrying out a dense matrix-vector multiply. The integration has the form of a discrete correlation, so by choosing a cell-centered discrete integration rule we can use an FFT to compute the matrix multiply. Thus the method is straightforward to implement, and jump diffusions can be added to an existing two asset Brownian motion pricing model at the expense of a few FFTs per time step.

The fixed point iteration is easily extended to handle American options through use of a penalty method. As our numerical tests show, there is no difficulty applying this technique to cases where the early exercise regions are multiply connected.

From a theoretical point of view, if the equation coefficients are constant (in log-price scaling), a grid rotation can be carried out with the result that the discrete equations are monotone for fully implicit time stepping. This property can be used to guarantee convergence of the fixed point iteration, as well as convergence to the viscosity solution. However, our numerical tests indicate that the solutions on the rotated

Grid		**Normal**			**MOBED**		
	S_2	$S_1 = 90$	100	110	90	100	110
$\Delta S_1 = \Delta S_2 = 2.5$	90	16.3559	13.9815	12.7464	13.7592	11.3827	10.7360
4356 Nodes, $\Delta\tau = 0.04$	100	12.9827	9.5970	7.8662	11.0477	6.8295	5.3499
DFT 288×288	110	11.4065	7.2016	5.1222	10.2586	4.8606	2.8249
$\Delta S_1 = \Delta S_2 = 1.25$	90	16.3818	13.9944	12.7544	13.7866	11.4012	10.7475
17161 Nodes, $\Delta\tau = 0.02$	100	13.0112	9.6138	7.8736	11.0676	6.8511	5.3655
DFT 576×576	110	11.4331	7.2200	5.1291	10.2718	4.8794	2.8377
$\Delta S_1 = \Delta S_2 = 0.625$	90	16.3885	13.9978	12.7566	13.7936	11.4058	10.7508
68121 Nodes, $\Delta\tau = 0.01$	100	13.0186	9.6183	7.8758	11.0727	6.8565	5.3693
DFT 1134×1134	110	11.4401	7.2251	5.1312	10.2756	4.8840	2.8406
Convergence	90	1.93	1.90	1.89	1.98	1.99	1.82
Exponent α	100	1.93	1.88	1.75	1.97	2.01	2.05
	110	1.93	1.87	1.68	1.78	2.06	2.11

TABLE 5: *Numeric solution to the American put on a minimum of two assets with the parameters given in Table 1. Crank-Nicolson time stepping was used. We report the Normal and MOBED jumps models. Note that convergence is approximately quadratic as the grid and time step is refined.*

European			**Normal**				**MOBED**			
		TS	FP	per	Linear	per	FP	per	Linear	per
ΔS	$\Delta\tau$	Count	Iters	TS	Iters	FP	Iters	TS	Iters	FP
2.500	0.04	25	75	3.00	175	2.33	75	3.00	175	2.33
1.250	0.02	50	106	2.12	263	2.48	112	2.24	275	2.46
0.625	0.01	100	200	2.00	800	4.00	200	2.00	800	4.00
American										
2.500	0.04	25	100	4.00	224	2.24	98	3.92	221	2.26
1.250	0.02	50	170	3.40	462	2.71	177	3.54	479	2.71
0.625	0.01	100	397	3.97	1315	3.31	396	3.96	1312	3.31

TABLE 6: *Shown are the total number of fixed point (FP) and linear solver iterations required to complete the solution of the European and American put options on the minimum of two assets. Crank-Nicolson time stepping was used. Data is given for each of the two jump models and three grids of Tables 3 and 5. Also shown are the average number of fixed point iterations required to advance each time step and the average number of linear solver iterations required to solve each fixed point iteration.*

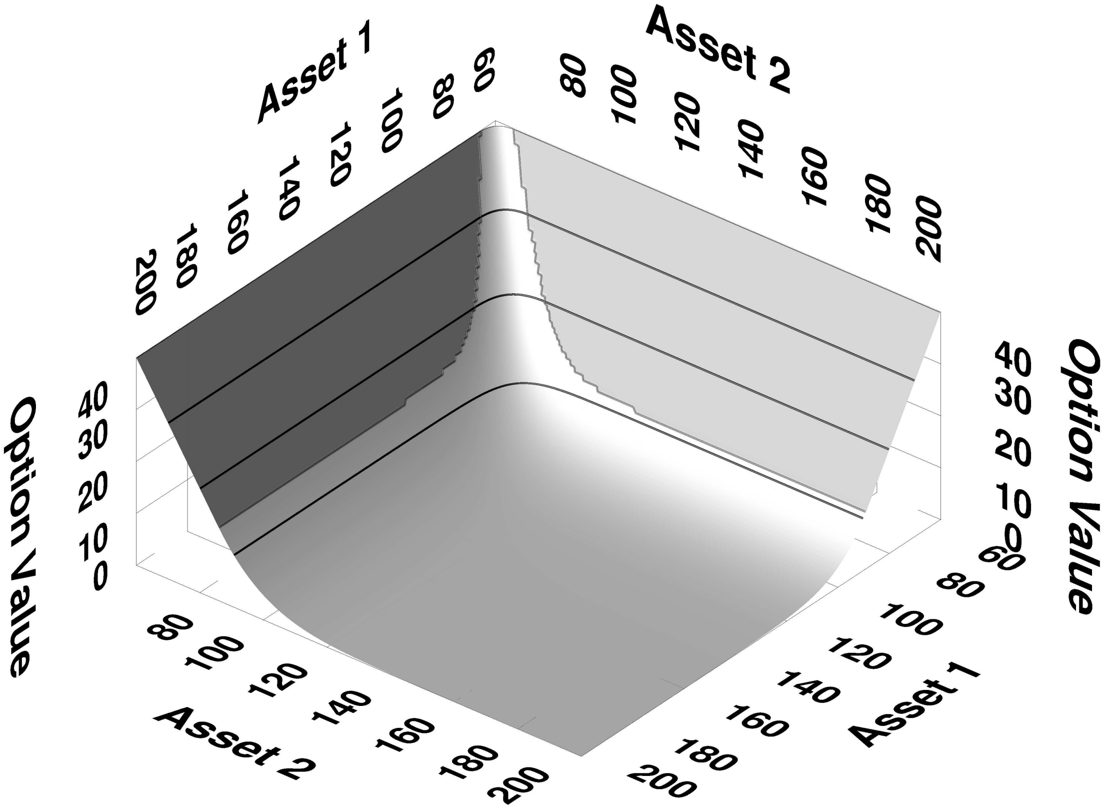

FIGURE 5: *The solution surface for the American put over the minimum of Assets 1 and 2 with MOBED jumps and parameters from Table 1. The two, disconnected, darker regions denote areas where the American minimum constraint is imposed.*

grid, for a given mesh size, have significantly more error than a comparable conventional grid. This result is consistent with the observations in [55]. This increased error is likely due to the poor resolution of the payoff on a rotated grid.

In the case where the discrete equations are non-monotone, a von Neumann analysis indicates that the fixed point iteration is still globally convergent. The analysis indicates that for typical market parameters, the fixed point iteration will reduce the initial residual by six orders of magnitude in $2-3$ iterations for European options. Numerical experiments confirm this. In the case of American options, the number of iterations required to solve the penalized equations (including the lagged integral terms) is about $3-4$ per time step. Numerical experiments also show quadratic convergence as the grid and time step are refined.

This technique can be adapted to other two-factor problems, such as option valuation under stochastic volatility with jumps, which will be the subject of a subsequent report. Clearly, the M-matrix condition for stability and convergence under the l_∞ norm is a very desirable property, although the traditional approach of grid rotation has been shown to degrade the accuracy of the solution, with no practical, evident improvement in solution quality. Future work will investigate other methods for obtaining an M-matrix representation of the discrete diffusion operator.

A Pricing PIDE

We assume that the underlying assets follow the risk neutral processes

$$\begin{aligned} dS_1 &= (r - \lambda\kappa_1)S_1\ dt + \sigma_1 S_1\ dZ_1 + (e^{J_1}-1)S_1\ dq\ , \\ dS_2 &= (r - \lambda\kappa_2)S_2\ dt + \sigma_2 S_2\ dZ_2 + (e^{J_2}-1)S_2\ dq\ , \end{aligned} \tag{A.1}$$

$$\begin{aligned} \sigma_1, \sigma_2 &= \text{asset volatilities}, \\ r &= \text{risk free rate}, \\ dZ_1, dZ_2 &= \text{increments of Wiener processes}, \\ dZ_1\ dZ_2 &= \rho_v\ dt\ , \\ dq &= \begin{cases} 0 & \text{with probability } 1-\lambda dt \\ 1 & \text{with probability } \lambda dt, \end{cases} \\ \lambda &= \text{mean arrival rate of Poisson jumps } (J_1, J_2)\ , \\ & (S_1, S_2) \rightarrow (e^{J_1}S_1, e^{J_2}S_2)\ , \\ \kappa_1 &= E[e^{J_1}-1]; \kappa_2 = E[e^{J_2}-1] \\ E[f(J_1,J_2)] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(J_1,J_2)g(J_1,J_2)\ dJ_1\ dJ_2\ , \\ g(J_1,J_2) &= \text{density function of the jump magnitudes in log-price scaling.} \end{aligned}$$

Then, using Ito's formula for finite activity jump processes, we can easily derive the pricing PIDE 2.3, by taking expectations under the risk neutral process ([11], [23] §1.4.5). Note that we assume that there is a single Poisson process which drives correlated jumps in both assets. This corresponds to a single market shock process which affects both prices [11].

B Jump Formulations

The two jump distributions used for the numerical examples of this paper are described in this appendix.

B.1 Bi-Variate Normal in Log-Price

The bi-variate Normal distribution is a straightforward and well defined extension of the univariate case. The probability density function for this distribution is, with parameters for the mean $\check{\mu}_1$, $\check{\mu}_2$, standard deviations $\check{\sigma}_1$, $\check{\sigma}_2$ and correlation $\check{\rho}$:

$$g_n(x_1, x_2; \check{\mu}_1, \check{\mu}_2, \check{\sigma}_1, \check{\sigma}_2, \check{\rho}) = \frac{1}{2\pi\check{\sigma}_1\check{\sigma}_2\sqrt{1-\check{\rho}^2}} \exp\left\{-\frac{z}{2(1-\check{\rho}^2)}\right\} \tag{B.1}$$

with

$$z = \left(\frac{x_1 - \check{\mu}_1}{\check{\sigma}_1}\right)^2 - \frac{2\check{\rho}\,(x_1 - \check{\mu}_1)\,(x_2 - \check{\mu}_2)}{\check{\sigma}_1\check{\sigma}_2} + \left(\frac{x_2 - \check{\mu}_2}{\check{\sigma}_2}\right)^2 \quad .$$

B.2 Marshall-Olkin Bi-Variate Exponential

The two-asset extension of the one-dimensional, double-sided, exponential distribution of Kou [27] is not as straightforward as with the Normal case. Numerous bi-variate exponential distributions have been studied (see [26], Chapter 47). Of these, the Marshall-Olkin bi-variate exponential distribution (MOBED) [33] retains the lack-of-memory property of the one-dimensional distribution. The PDF of this distribution is discontinuous

$$g_x(x_1, x_2; \check{\eta}_1, \check{\eta}_2, \check{\eta}_{12}) = \exp\left\{-\check{\eta}_1 x_1 - \check{\eta}_2 x_2 - \check{\eta}_{12}\max(x_1, x_2)\right\} \times \begin{cases} \check{\eta}_1(\check{\eta}_2 + \check{\eta}_{12}) & 0 < x_1 < x_2 \\ \check{\eta}_{12} & 0 < x_1 = x_2 \\ \check{\eta}_2(\check{\eta}_1 + \check{\eta}_{12}) & 0 < x_2 < x_1 \end{cases} \tag{B.2}$$

where $1/\check{\eta}_i$, $i = 1, 2$ is the mean jump in direction of x_i by itself, and $1/\check{\eta}_{12}$ is the mean jump of both assets together. Although the PDF is discontinuous and the line $x_1 = x_2$ has a two-dimensional Lebesgue measure of zero, there is a positive probability associated with this line. For our computations (Section 4.3.1) we must express the integrated probability in terms of the cumulative distribution function given in [33] §3.1.

To define the PDF in the entire real plane we write it in four quadrants and, following the approach of [27], assign a probability to each quadrant. We take $\check{p}_1, \check{p}_2 \in [0, 1]$ as the probability of a positive jump in x_1 and x_2 respectively with $\check{q}_1 = (1 - \check{p}_1)$ and $\check{q}_2 = (1 - \check{p}_2)$ the probability of a negative jump. The PDF is then

$$\begin{aligned} g_m(x_1, x_2; &\check{\eta}_{p,1}, \check{\eta}_{q,1}, \check{\eta}_{p,2}, \check{\eta}_{q,2}, \\ &\check{\eta}_{pp}, \check{\eta}_{qp}, \check{\eta}_{pq}, \check{\eta}_{qq}, \check{p}_1, \check{p}_2) = \quad \check{p}_1\check{p}_2 \cdot g_x(+x_1, +x_2; \check{\eta}_{p,1}, \check{\eta}_{p,2}, \check{\eta}_{pp}) \cdot 1_{x_1, x_2 \geq 0} \\ &+\check{q}_1\check{p}_2 \cdot g_x(-x_1, +x_2; \check{\eta}_{q,1}, \check{\eta}_{p,2}, \check{\eta}_{qp}) \cdot 1_{x_1 < 0, x_2 \geq 0} \\ &+\check{p}_1\check{q}_2 \cdot g_x(+x_1, -x_2; \check{\eta}_{p,1}, \check{\eta}_{q,2}, \check{\eta}_{pq}) \cdot 1_{x_1 \geq 0, x_2 < 0} \\ &+\check{q}_1\check{q}_2 \cdot g_x(-x_1, -x_2; \check{\eta}_{q,1}, \check{\eta}_{q,2}, \check{\eta}_{qq}) \cdot 1_{x_1, x_2 < 0} \end{aligned} \tag{B.3}$$

where the $\check{\eta}$ parameters are defined for each of the four quadrants. To avoid further complication we have not encoded the case where the peak of the distribution is offset by $(\check{\mu}_1, \check{\mu}_2)$. We note that to satisfy Assumptions 3.1 and 3.2 the positive jumps must have $\check{\eta} > 1$, as in [27].

C Localization Error Proofs

In Section 3.3 we leave the proofs of Theorems 3.1 and 3.2 to the following two sections.

C.1 Proof of Theorem 3.1 in Log-Price Scaling: Cutoff Error

In this section, we determine the effect of the approximations to the operators of Equation 2.3 in Equation 3.2. These approximations are

$$\begin{aligned} \lambda &\simeq \lambda_C \\ \mathcal{H} &\simeq \mathcal{H}_D \end{aligned} \tag{C.1}$$

We note that the difference between $\mathcal{H}_D$ in Equation 3.3 and $\mathcal{H}$ given in Equation 2.8 is in the range of the correlation integral, which is performed over Ω_D instead of Ω_∞ respectively. In the following we treat the problem as embedded in the infinite domain Ω_∞ and hence the solution is defined outside of Ω_D and we can extend the domain of integration of the first term to $(y+J) \in \Omega_\infty$.

First consider the solution of the following PIDE on Ω_∞. Let U be the solution to Equation 2.3. Let V be the solution to Equation 3.2 embedded in Ω_∞ to form the initial value problem

$$\begin{aligned} V_\tau =& \mathcal{L}V + \lambda_C \mathcal{H}_D V \\ V(y,0) =& \mathcal{I}(y) \quad . \end{aligned} \tag{C.2}$$

Define the cutoff error $E = U - V$. After manipulating Equations 2.3 and C.2 we have

$$\begin{aligned} &E_\tau = \mathcal{L}E + \lambda_C \mathcal{H}_D E + (\lambda - \lambda_C)\mathcal{H}U + \lambda(\mathcal{H} - \mathcal{H}_D)U \\ &E(y,0) = 0 \quad . \end{aligned} \tag{C.3}$$

PIDE C.3 satisfies the conditions of Assumption 3.4 so that a classical solution can be expressed as a convolution of a Green's function and a source function. With the conditions thus satisfied, the solution to $E(y,\tau)$ can be written as [23]

$$\begin{aligned} &E(y,\tau) = \int_0^\tau \int_{\Omega_\infty} G(y,\tau,y',\tau') \left[(\lambda - \lambda_C)\mathcal{H}U(y',\tau') + \lambda(\mathcal{H} - \mathcal{H}_D)U(y',\tau')\right] \, dy' \, d\tau' \\ &E(y,0) = 0 \end{aligned} \tag{C.4}$$

where $G(y,\tau,y',\tau') \geq 0$ is the Green's function of Equation C.2, which is the formal solution to

$$G_\tau = \mathcal{L}G + \lambda_C \mathcal{H}_D G + \delta(y - y', \tau - \tau') \quad . \tag{C.5}$$

We can rewrite Equation C.4 in two components: the first of which expresses error due to a finite $\Omega_C \subset \Omega_\infty$ and the second expresses error due to a finite $\Omega_D \subset \Omega_\infty$. Thus we have two errors to bound separately

$$\begin{aligned} E(y,\tau) &= E_1(y,\tau) + E_2(y,\tau) \\ E_1 &= \int_0^\tau \int_{\Omega_\infty} G(y,\tau,y',\tau') \left[(\lambda - \lambda_C)\mathcal{H}U(y',\tau')\right] \, dy' \, d\tau' \\ E_2 &= \int_0^\tau \int_{\Omega_\infty} G(y,\tau,y',\tau') \left[\lambda(\mathcal{H} - \mathcal{H}_D)U(y',\tau')\right] \, dy' \, d\tau' \quad . \end{aligned} \tag{C.6}$$

First, we estimate E_1 as $\Omega_C \to \Omega_\infty$. We can bound the integral of the Green's function by noting that the solution to

$$\begin{aligned} W_\tau &= \mathcal{L}W + \lambda_C \mathcal{H}_D W \\ W(y,0) &= \mathcal{I}(y) \end{aligned} \tag{C.7}$$

on Ω_∞ can be written as

$$W(y,\tau) = \int_{\Omega_\infty} G(y,\tau,y',0)\mathcal{I}(y') \, dy' \quad . \tag{C.8}$$

Assuming that a bounded solution (for fixed y) exists, then we must have

$$\lim_{\Omega_C \to \Omega_\infty} \int_{\Omega_\infty \backslash \Omega_C} G(y,\tau,y',0)\mathcal{I}(y') \, dy' = 0 \tag{C.9}$$

and hence

$$\lim_{\Omega_C \to \Omega_\infty} \int_0^\tau \int_{\Omega_\infty \backslash \Omega_C} G(y,\tau,y',\tau')\mathcal{I}(y') \, dy' \, d\tau' = 0 \quad . \tag{C.10}$$

In particular, we take Assumption 3.1 to hold so that a solution to Equation C.7 exists for any

$$\mathcal{I}(y) \leq c_1 + c_2 \left(e^{y_1} + e^{y_2}\right) \tag{C.11}$$

so that using Equations C.10 and C.11 gives

$$\lim_{\Omega_C \to \Omega_\infty} \int_0^\tau \int_{\Omega_\infty \setminus \Omega_C} G(y,\tau,y',\tau') \left[c_1 + c_2 \left(e^{y_1'} + e^{y_2'}\right)\right] \, dy' \, d\tau' = 0 \quad . \tag{C.12}$$

If Assumption 3.2 holds for the solution U to Equation 2.3 over $\Omega_\infty \setminus \Omega_C$

$$|\mathcal{H}U(y,\tau)| \le c_3 + c_4 \left(e^{y_1} + e^{y_2}\right) \tag{C.13}$$

then, because $\lambda - \lambda_C = 0$ on Ω_C, we have that

$$\begin{aligned}
\lim_{\Omega_C \to \Omega_\infty} |E_1| &\le \lim_{\Omega_C \to \Omega_\infty} \int_0^\tau \int_{\Omega_\infty} G(y,\tau,y',\tau') \, |(\lambda - \lambda_C)\mathcal{H}U(y',\tau')| \, dy' \, d\tau' \\
&= \lim_{\Omega_C \to \Omega_\infty} \int_0^\tau \int_{\Omega_\infty \setminus \Omega_C} G(y,\tau,y',\tau') \, |\lambda \mathcal{H}U(y',\tau')| \, dy' \, d\tau' \\
&\le \lim_{\Omega_C \to \Omega_\infty} \int_0^\tau \int_{\Omega_\infty \setminus \Omega_C} G(y,\tau,y',\tau') \left[c_5 + c_6 \left(e^{y_1'} + e^{y_2'}\right)\right] \, dy' \, d\tau' = 0
\end{aligned} \tag{C.14}$$

which follows from Equation C.12.

Next, we bound E_2 as $\Omega_D \to \Omega_\infty$. Note that from Equations 2.8 and 3.3

$$(\mathcal{H} - \mathcal{H}_D)U(y,\tau) = \int_{(y+J) \in \Omega_\infty \setminus \Omega_D} g(J) U(y+J,\tau) \, dJ \quad .$$

The preconditions for the existence of

$$\int_{-\infty}^{\infty} g(J) U(y+J,\tau) \, dJ$$

imply that

$$\lim_{\Omega_D \to \Omega_\infty} \int_{(y+J) \in \Omega_\infty \setminus \Omega_D} g(J) U(y+J,\tau) \, dJ = 0$$

hence

$$\begin{aligned}
\lim_{\Omega_D \to \Omega_\infty} E_2 &= \lim_{\Omega_D \to \Omega_\infty} \lambda \int_0^\tau \int_{\Omega_\infty} G(y,\tau,y',\tau') (\mathcal{H} - \mathcal{H}_D) U(y',\tau') \, dy' \, d\tau' \\
&= \lim_{\Omega_D \to \Omega_\infty} \lambda \int_0^\tau \int_{\Omega_\infty} G(y,\tau,y',\tau') \int_{(y'+J) \in \Omega_\infty \setminus \Omega_D} g(J) U(y'+J,\tau') \, dJ \, dy' \, d\tau' = 0
\end{aligned} \tag{C.15}$$

and thus

$$\lim_{\Omega_C, \Omega_D \to \Omega_\infty} |E(y,\tau)| \le |E_1(y,\tau)| + |E_2(y,\tau)| = 0 \quad .$$

C.2 Proof of Theorem 3.2 in Log-Price Scaling: Error Due to Artificial Dirichlet Condition

In this section, we determine the error due to approximating the exact boundary condition for Equation 3.2. Note that in the previous section we have shown that Equation C.2 converges, in theory, to the solution of Equation 2.3 as $\Omega_C, \Omega_D \to \Omega_\infty$.

Now suppose we solve for W on the finite domain Ω_D

$$\begin{aligned}
W_\tau &= \mathcal{L}W + \lambda_C \mathcal{H}_D W \\
W(y,0) &= \mathcal{I}(y) \\
W(y,\tau) &= V(y,\tau) \quad ; \quad y \in \partial\Omega_D
\end{aligned} \tag{C.16}$$

where $V(y,\tau)$ is the exact Dirichlet boundary condition, given from the solution to Equation C.2 embedded in Ω_∞. Therefore, noting that the correlation integral of $(y+J) \in \mathcal{H}_D$ of Equation C.16 is truncated to operate on Ω_D only, $W = V$ on Ω_D. The solution to Equation C.16 is [23] §IV

$$W(y,\tau) = \int_{\Omega_D} G^W(y,\tau,y',0)\mathcal{I}(y')\;dy' + \int_0^\tau \int_{\partial\Omega_D} P(y,\tau,y',\tau')V(y',\tau')\;dy'\;d\tau' \tag{C.17}$$

where the Green's function $G^W(y,\tau,y',\tau') \geq 0$ is the formal solution to

$$\begin{aligned} G^W_\tau &= \mathcal{L}G^W + \lambda_C \mathcal{H}_D G^W + \delta(y-y',\tau-\tau') \\ G^W(y,\tau,y',\tau') &= 0\;; \quad y \in \partial\Omega_D \end{aligned}$$

and $P(y,\tau,y',\tau') \geq 0$ is the Poisson function ([23] §IV[4]) of Equation C.16. The solution for V depends only on initial conditions hence can be written

$$V(y,\tau) = \int_{\Omega_\infty} G(y,\tau,y',0)\mathcal{I}(y')\;dy' \tag{C.18}$$

where the Green's function G is the solution to Equation C.5. Since $V = W$ on Ω_D, and as $\Omega_D \to \Omega_\infty$, $G^W \to G$, then, for any fixed point (y,τ),

$$\lim_{\Omega_D \to \Omega_\infty} \int_0^\tau \int_{\partial\Omega_D} P(y,\tau,y',\tau')V(y',\tau')\;dy'\;d\tau' = 0 \tag{C.19}$$

In particular, this holds for a value V constant in y. If $\mathcal{I}(y) = c_0 > 0$, then $V = c_0 e^{-r\tau}$, and then $c_1 = c_0 e^{-rT} \leq c_0 e^{-r\tau}$ (for $\tau \leq T$) and

$$\begin{aligned} &\lim_{\Omega_D \to \Omega_\infty} \int_0^\tau \int_{\partial\Omega_D} P(y,\tau,y',\tau')\;c_1\;dy'\;d\tau' \\ &\qquad\qquad \leq \lim_{\Omega_D \to \Omega_\infty} \int_0^\tau \int_{\partial\Omega_D} P(y,\tau,y',\tau')c_0 e^{-r\tau}\;dy'\;d\tau' = 0 \end{aligned} \tag{C.20}$$

Now, suppose we approximate W on Ω_D by Y where

$$\begin{aligned} Y_\tau &= \mathcal{L}Y + \lambda_C \mathcal{H}_D Y \\ Y(y,0) &= \mathcal{I}(y) \\ Y(y,\tau) &= B(y,\tau) \quad ; \quad y \in \partial\Omega_D \end{aligned}$$

so that the error $E = W - Y$ satisfies

$$\begin{aligned} E_\tau &= \mathcal{L}E + \lambda_C \mathcal{H}_D E \\ E(y,0) &= 0 \\ E(y,\tau) &= V(y,\tau) - B(y,\tau) \quad ; \quad y \in \partial\Omega_D \end{aligned} \tag{C.21}$$

with solution [23]

$$E(y,\tau) = \int_0^\tau \int_{\partial\Omega_D} P(y,\tau,y',\tau')\left[V(y',\tau') - B(y',\tau')\right]\;dy'\;d\tau' \quad . \tag{C.22}$$

Without loss of generality, we can assume that $\mathcal{I}(y) \geq 0$, so that $V(y,\tau) \geq 0$. Suppose that, by Assumption 3.3

$$|V(y,\tau) - B(y,\tau)| \leq c_1 + c_2 V(y,\tau)$$

[4] Intuitively, the Poisson function serves to encode the boundary conditions of the problem and Green's functions solve for the interior when the boundary condition is zero.

(which is trivially satisfied if $B(y,\tau)=0$), then Equation C.22 becomes

$$|E(y,\tau)| \leq \int_0^{\tau} \int_{\partial\Omega_D} P(y,\tau,y',\tau')\,[c_1 + c_2 V(y',\tau')]\ dy'\ d\tau' \quad . \tag{C.23}$$

From Equations C.19 and C.20 we then have

$$\lim_{\Omega_D \to \Omega_\infty} |E(y,\tau)| = 0 \quad .$$

D Convergence of the Fixed Point Iteration

In this appendix we demonstrate the convergence rate for the fixed point iteration of Section 6.1 both as a functional iteration using the continuous operators and as an iteration using the general discrete linear system operators as outlined in Section 4.1.

D.1 Proof of Theorem 6.1: Convergence of the Functional Fixed Point Iteration

Starting from the iteration defined in Equation 6.1 we demonstrate the convergence rate noted by Theorem 6.1. Note that $\mathcal{G}$ contains only differential terms so if c is a constant then $\mathcal{G}c = 0$. We impose a Dirichlet boundary condition over all of $\partial\Omega_D$.

Let $E^k = W^{n+1} - Z^k$ to obtain the error propagation equation

$$\begin{aligned} \left[1 + (1-\theta)\Delta\tau\,(r + \lambda_C - \mathcal{G})\right] E^{k+1} &= (1-\theta)\Delta\tau\,\lambda_C \mathcal{J}_D E^k \\ E^{k+1}(y,\tau) &= 0\ ; \quad y \in \partial\Omega_D \end{aligned}$$

with solution [23] §IV

$$E^{k+1}(y) = \int_{\Omega_D} G(y,y')\,(1-\theta)\Delta\tau\,\lambda_C \mathcal{J}_D E^k(y')\ dy' \tag{D.1}$$

where $G(y,y') > 0$ is the formal solution of

$$\begin{aligned} \left[1 + (1-\theta)\Delta\tau\,(r + \lambda_C - \mathcal{G})\right] G &= \delta(y - y') \\ G(y,y') &= 0\ ; \quad y \in \Omega_D \quad . \end{aligned}$$

Note that $E^0 = W^{n+1} - W^n$ is the difference between two smooth functions, hence is also smooth, and that $\mathcal{G}$ is a uniformly elliptic operator with bounded coefficients on Ω_D (see [23] §IV.2).

Now, consider the equation

$$\begin{aligned} \left[1 + (1-\theta)\Delta\tau\,(r + \lambda_C - \mathcal{G})\right] A &= f(y)\ ; \quad y \in \Omega_D \\ A(y) &= B(y)\ ; \quad y \in \partial\Omega_D \end{aligned} \tag{D.2}$$

which has the formal solution

$$A(y) = \int_{\Omega_D} G(y,y') f(y')\ dy' + \int_{\partial\Omega_D} P(y,y') B(y')\ dy' \tag{D.3}$$

where $P(y,y') \geq 0$ is a Poisson function induced by the non-zero boundary conditions ([23] §IV.3). Let $A(y) = 1$. We can then compute the RHS of Equation D.2 directly

$$\left[1 + (1-\theta)\Delta\tau\,(r + \lambda_C - \mathcal{G})\right] A = \left[1 + (1-\theta)\Delta\tau(r + \lambda_C)\right]$$

so that, if

$$\begin{aligned} f(y) &= 1 + (1-\theta)\Delta\tau\,(r + \lambda_C) \\ B(y) &= 1\ ; \quad A(y) = 1 \end{aligned} \tag{D.4}$$

then combining equations (D.3) and (D.4) and noting that $P(y,y') \geq 0$

$$\int_{\Omega_D} G(y,y')\,[1+(1-\theta)\Delta\tau\,(r+\lambda_C)]\ dy' + \int_{\partial\Omega_D} P(y,y')\ dy' = 1$$

$$[1+(1-\theta)\Delta\tau\,(r+\lambda)] \int_{\Omega_C} G(y,y')\ dy' \leq 1$$

or

$$\int_{\Omega_C} G(y,y')\ dy' \leq \frac{1}{1+(1-\theta)\Delta\tau\,(r+\lambda)} \quad . \tag{D.5}$$

Noting that

$$\mathcal{J}_D \|E^k\|_\infty \leq \|E^k\|_\infty$$

and $G(y,y') \geq 0$ then Equation D.1 gives

$$E^{k+1} \leq (1-\theta)\Delta\tau\,\lambda \|E^k\|_\infty \int_{\Omega_C} G(y,y')\ dy' \tag{D.6}$$

which becomes, by Equation D.5,

$$\|E^{k+1}\|_\infty \leq \frac{(1-\theta)\Delta\tau\,\lambda}{1+(1-\theta)\Delta\tau\,(r+\lambda)} \|E^k\|_\infty \quad .$$

Hence the functional iteration (6.1) is unconditionally convergent, with rapid convergence in the usual case where $\lambda\Delta\tau \ll 1$.

D.2 Proof of Theorem 6.2: Convergence of the Discrete Fixed Point Iteration

For the following we denote as $\underline{1}$ the vector $\underline{1}_{\mathsf{i}} = 1\ \forall \mathsf{i}$, and we must have three conditions:

1. $\|\mathsf{G}\cdot\underline{1}\|_\infty = 0$, i.e. the differential approximation is exact for a constant vector,
2. $\mathsf{J} \geq 0$ and $\max_{\mathsf{i}} \sum_{\mathsf{j}} \mathsf{J}_{\mathsf{ij}} \leq 1$ thus $\|\mathsf{I_c}\mathsf{J}\cdot\underline{1}\|_\infty \leq 1$, and
3. $(-\mathsf{G})$ is M-compatible so that $\mathsf{N}^{-1} \geq 0$ exists ([43] Theorem F15) where $\mathsf{N} = \{\mathsf{I} - (1-\theta)\Delta\tau\,[\mathsf{G} - r\,\mathsf{I} - \lambda\mathsf{I_c}]\}$ is the matrix of the LHS of the iteration.

Let $\mathsf{e}^k = \mathsf{w}^{n+1} - \mathsf{z}^k$. Thus the error in the solution iteration Equation 6.2 propagates as

$$\mathsf{e}^{k+1} = \left[(1-\theta)\Delta\tau\,\lambda\right]\ \left[\mathsf{N}^{-1}\mathsf{I_c}\mathsf{J}\right]\ \mathsf{e}^k \quad . \tag{D.7}$$

Taking the norm of Equation D.7

$$\left\|\mathsf{e}^{k+1}\right\|_\infty = \left\|\left[(1-\theta)\Delta\tau\,\lambda\right]\ \left[\mathsf{N}^{-1}\mathsf{I_c}\mathsf{J}\right]\ \mathsf{e}^k\right\|_\infty \leq [(1-\theta)\Delta\tau\,\lambda]\ \left\|\mathsf{N}^{-1}\mathsf{I_c}\right\|_\infty\ \|\mathsf{J}\|_\infty\ \|\mathsf{e}^k\|_\infty \tag{D.8}$$

by the compatibility of the $\|\cdot\|_\infty$ norm and triangle inequality. Condition 1 holds, therefore

$$\mathsf{N}\cdot\underline{1} \geq [1+(1-\theta)\Delta\tau\,(r+\lambda)]\,\mathsf{I_c}\,\underline{1} \quad \text{and} \quad \mathsf{N}^{-1}\mathsf{I_c}\,\underline{1} \leq \frac{1}{1+(1-\theta)\Delta\tau\,(r+\lambda)}\underline{1}$$

and because Condition 3 holds $\mathsf{N}^{-1} \geq 0$ exists and we may write

$$\left\|\mathsf{N}^{-1}\underline{1}\right\|_\infty = \left\|\mathsf{N}^{-1}\right\|_\infty$$

by the definition of the row maximum norm of a matrix. This leads in turn to $\|\mathsf{I_c}\mathsf{J}\|_\infty \leq 1$ by Condition 2, and we may write Equation D.8 as

$$\|\mathsf{e}^{k+1}\|_\infty \leq \frac{(1-\theta)\Delta\tau\,\lambda}{1+(1-\theta)\Delta\tau\,(r+\lambda)} \|\mathsf{e}^k\|_\infty \quad . \tag{D.9}$$

We conclude that the fixed point iteration Equation 6.2 is convergent with the rate stated in Theorem 6.2.

E Finite Difference Stencils

For reference, we note the following finite difference stencils over a function $U(y)$. For further information the reader may wish to consult [1] §25.3, [41] §9.4, or [16] §8. For each finite difference we write a compact linear operator form, for example $\partial_i^{h_i}$. The extension of these operators to a non-constant grid spacing is omitted for brevity, but is straightforward.

The grid, on which this set of finite difference stencils is defined, has constant spacing h_1 and h_2 in y_1 and y_2 with $h_1 \neq h_2$ in general. We use e_1 and e_2 to denote unit vectors in the co-ordinate directions of y_1 and y_2 respectively. We use the notation y_i, e_i and h_i where $i = 1, 2$ where an equation applies identically in each direction.

When Dirichlet boundary conditions are applied we do not include the boundary nodes in the solution vector. Where entries in a finite difference stencil centered on an interior node refer to adjacent boundary nodes, they do not appear in the matrix G of the discrete differential operator and hence are, in effect, zero. The off-diagonal entries which cannot be represented in G are multiplied by the known boundary condition values then summed into the boundary enforcement vector b of Equation 4.6. Other methods of imposing Dirichlet boundary conditions are equally effective; this approach permits the ready application of the l_∞ norm stability analyses. We note that this approach does not change the M-compatibility of G.

We approximate first derivatives with a second order central difference

$$\frac{\partial U(y)}{\partial y_i} \approx \frac{1}{2\,h_i}\left[U(y + e_i\,h_i) - U(y - e_i\,h_i)\right] = \partial_i^{h_i} U(y) \quad . \tag{E.1}$$

Where the second order approximation results in negative values for the off-diagonal coefficients of matrix G of Equation 4.5 we may resort to a first order approximation. Either the forward

$$\frac{\partial U(y)}{\partial y_i} \approx \frac{1}{h_i}\left[U(y + e_i\,h_i) - U(y)\right] = \partial_i^{h_i +} U(y) \tag{E.2}$$

or backward

$$\frac{\partial U(y)}{\partial y_i} \approx \frac{1}{h_i}\left[U(y) - U(y - e_i\,h_i)\right] = \partial_i^{h_i -} U(y) \tag{E.3}$$

difference may be applied, depending on the leading coefficients. The second order partial difference is taken with the second order equation ([1] §25.3.23)

$$\frac{\partial^2 U(y)}{\partial y_i^2} \approx \frac{1}{h_i^2}\left[U(y + e_i\,h_i) + U(y - e_i\,h_i) - 2U(y)\right] = \partial_{ii}^{h_i} U(y) \quad . \tag{E.4}$$

We may take a cross-partial derivative on a seven-point stencil by using ([1] §25.3.27) and one of two complementary choices. The first is preferred for our problem where $\rho_v > 0$

$$\begin{aligned}\frac{\partial^2 U(y)}{\partial y_1\,\partial y_2} \approx \frac{1}{2\,h_1\,h_2}\Big[&U(y + e_1\,h_1 + e_2\,h_2) + U(y - e_1\,h_1 - e_2\,h_2) + 2\,U(y)\\ &- U(y + e_1\,h_1) - U(y - e_1\,h_1) - U(y + e_2\,h_2) - U(y - e_2\,h_2)\Big] = \partial_{1,2}^{h_1 h_2 +} U(y)\end{aligned} \tag{E.5}$$

and the second is appropriate for $\rho_v < 0$

$$\begin{aligned}\frac{\partial^2 U(y)}{\partial y_1\,\partial y_2} \approx \frac{-1}{2\,h_1\,h_2}\Big[&U(y - e_1\,h_1 + e_2\,h_2) + U(y + e_1\,h_1 - e_2\,h_2) + 2\,U(y)\\ &- U(y + e_1\,h_1) - U(y - e_1\,h_1) - U(y + e_2\,h_2) - U(y - e_2\,h_2)\Big] = \partial_{1,2}^{h_1 h_2 -} U(y) \quad .\end{aligned} \tag{E.6}$$

We may also use a nine-point stencil to obtain a four point cross-partial difference ([1] §25.3.26)

$$\begin{aligned}\frac{\partial^2 U(y)}{\partial y_1\,\partial y_2} \approx \frac{1}{2\,h_2}\Bigg[&\frac{U(y + e_1\,h_1 + e_2\,h_2) - U(y - e_1\,h_1 + e_2\,h_2)}{2\,h_1}\\ &- \frac{U(y + e_1\,h_1 - e_2\,h_2) - U(y - e_1\,h_1 - e_2\,h_2)}{2\,h_1}\Bigg] = \partial_{1,2}^{h_1 h_2 \circ} U(y) \quad .\end{aligned} \tag{E.7}$$

F Proof of Theorems 5.3 and 6.3: Stability and Convergence by von Neumann Analysis

The von Neumann analysis in this appendix applies to a periodic, initial value problem with the same operators as the PIDE 2.3. The following sections employ a number of elements in common, hence we proceed with both the time step and fixed point iteration analysis only after some preliminary discussions. Sections F.1 and F.2 describe some basic mathematics we shall use for the analysis. In Section F.3 we state the problem and error propagation equations to be analyzed, then in Section F.4 discuss the approach to the final analysis and the relations that must hold to prove Theorems 5.3 and 6.3. In Sections F.5 and F.6 we demonstrate those relations.

We write the periodic, discrete form of the PIDE 2.3 localized over a periodic domain Ω°_D. The problem is phrased as a pure initial value problem with nodes of $\partial\Omega_D$ included in Ω°_D. We use a regular rectangular grid of $Q = Q_x \times Q_y$ equally spaced nodes where, without loss of generality, we take Q_x, Q_y to be powers of 2. The finite difference discretization on this grid is defined in Section E above using node spacing (h_1, h_2). As in Section 4.3.1 we define the location of the points on the grid as

$$q_i \in \mathbb{R}^2 \ , \quad i = 1 \dots Q \ , \quad q_i \in \Omega^\circ_D \quad . \tag{F.1}$$

The DFT operation makes it convenient to introduce the double-subscript notation $U_{kl} = U(q_i)$ to denote a point on the grid where (k, l) are the grid line coordinates. We relate the grid point q_i, at position i in the solution vector, to grid line coordinates (k, l) by a mapping such as

$$i = k + l\, Q_x + 1 \ , \quad k = 0 \dots (Q_x - 1) \ , \quad l = 0 \dots (Q_y - 1) \quad .$$

Where a vector u of a value over the grid is required, we denote its components $\mathsf{u}_\mathsf{i} = U_{kl} = U(q_i)$ and assume this ordinate mapping holds (although we may use different letters in the subscripts). The same double-subscript notation is used to locate points on the grid in the Fourier-transformed space.

The DFT $\mathcal{D}(U)$ over periodic domain Ω°_D results in a $Q_x \times Q_y$ grid of coefficients denoted as $\hat{U}_{mn}$ where we use the following, compact notation

$$\begin{aligned} \hat{U}_{mn} &= \sum_{i=0}^{Q_x-1} \sum_{j=0}^{Q_y-1} U_{ij}\ \xi(-im - jn) = \sum_{ij} U_{ij}\ \xi(-im - jn) \\ U_{ij} &= \frac{1}{Q} \sum_{m=-Q_x/2+1}^{Q_x/2} \sum_{n=-Q_x/2+1}^{Q_y/2} \hat{U}_{mn}\ \xi(im + jn) = \frac{1}{Q} \sum_{mn} \hat{U}_{mn}\ \xi(im + jn) \end{aligned} \tag{F.2}$$

where

$$\xi(k) = \exp\left\{\sqrt{-1}\,\zeta\, k\right\} \ , \text{ and } \ \zeta = \frac{2\pi}{Q} \quad .$$

Note that the correction for the grid node count is done during the inverse transform. We shall usually drop the ranges on the summations as we have in Equation F.2, assuming that the log-price space indices (i, j) and the Fourier-space indices (m, n) refer to their periodic image ("wrap around") under an addition which crosses the grid boundary. We have the useful identities

$$\xi(+m) + \xi(-m) = 2\cos(\zeta m) \ , \quad \xi(+m) - \xi(-m) = 2\sqrt{-1}\sin(\zeta m) \ , \text{ and } \ \frac{\xi(m+i)}{\xi(m)} = \xi(i) \tag{F.3}$$

and note that, for example,

$$\begin{aligned} U_{i+1,j+1} - U_{i-1,j-1} &= \sum_{mn} \Big[\hat{U}_{mn}\ \xi((i+1)m + (j+1)n) - \hat{U}_{mn}\ \xi((i-1)m + (j-1)n)\Big] \\ &= \sum_{mn} \Big[\xi(+m+n) - \xi(-m-n)\Big] \hat{U}_{mn}\ \xi(m+n) \end{aligned} \tag{F.4}$$

which will be used to reduce finite difference expressions.

F.1 Discrete Fourier Transform of Finite Difference Stencils

The coefficients of $\mathbf{V}$ and $\mathbf{D}$ are constant in our initial value problem. This means we may apply the DFT to each of the finite difference stencils in Section E ([7] §13). For clarity, we write the value for a single coefficient $\hat{U}_{mn}$ of the Fourier transform $\hat{\partial}$ of difference operator ∂ at a single node U_{ij}, leaving off the leading sums and coefficients. Our objective is to write the difference equations in terms of trigonometric functions, thus we apply the identities of Equations F.3 through F.4. For brevity we write $\xi_p = \xi(im + jn)$ in the final form, and omit discretizations that require only a change of axes.

$$\begin{aligned}
(h_1)\ \hat{\partial}_1^{h_1} \hat{U}_{mn}\Big|_{ij} &= \big\{\xi((i+1)m+jn) - \xi((i-1)m+jn)\big\}\, \hat{U}_{mn} \\
&= 2\sqrt{-1}\sin(\zeta m)\ \xi_p\, \hat{U}_{mn} \quad , \qquad \text{(F.5)} \\
\left(h_1{}^2\right)\ \hat{\partial}_{1,1}^{h_1} \hat{U}_{mn}\Big|_{ij} &= \big\{\xi((i+1)m+jn) - 2\xi(im+jn) + \xi((i-1)m+jn)\big\}\, \hat{U}_{mn} \\
&= 2\big\{\cos(\zeta m) - 1\big\}\ \xi_p\, \hat{U}_{mn} \quad . \qquad \text{(F.6)}
\end{aligned}$$

The following two cross-partial derivatives complete the seven point discretization:

$$\begin{aligned}
(2\,h_1\,h_2)\ \hat{\partial}_{1,2}^{h_1,h_2+} \hat{U}_{mn}\Big|_{ij} &= \big\{\xi((i+1)m+(j+1)n) + \xi((i-1)m+(j-1)n) + 2\xi(im+jn) \\
&\qquad - \xi((i+1)m+jn) - \xi((i-1)m+jn) \\
&\qquad - \xi(im+(j+1)n) - \xi(im+(j-1)n)\big\}\, \hat{U}_{mn} \\
&= 2\big\{1 + \cos(\zeta m + \zeta n) - \cos(\zeta m) - \cos(\zeta n)\big\}\ \xi_p\, \hat{U}_{mn} \quad , \qquad \text{(F.7)} \\
(2\,h_1\,h_2)\ \hat{\partial}_{1,2}^{h_1,h_2-} \hat{U}_{mn}\Big|_{ij} &= -\big\{\xi((i+1)m+(j-1)n) + \xi((i-1)m+(j+1)n) + 2\xi(im+jn) \\
&\qquad - \xi((i+1)m+jn) - \xi((i-1)m+jn) \\
&\qquad - \xi(im+(j+1)n) - \xi(im+(j-1)n)\big\}\, \hat{U}_{mn} \\
&= -2\big\{1 + \cos(\zeta m - \zeta n) - \cos(\zeta m) - \cos(\zeta n)\big\}\ \xi_p\, \hat{U}_{mn} \quad . \qquad \text{(F.8)}
\end{aligned}$$

The following cross-partial derivative is used in the nine-point discretization:

$$\begin{aligned}
(4\,h_1\,h_2)\ \hat{\partial}_{1,2}^{h_1,h_2\circ} \hat{U}_{mn}\Big|_{ij} &= \big\{\xi((i+1)m+(j+1)n) - \xi((i-1)m+(j-1)n) \\
&\qquad -\xi((i-1)m+(j+1)n) + \xi((i+1)m+(j-1)n)\big\}\, \hat{U}_{mn} \\
&= \big\{-4\sin(\zeta m)\sin(\zeta n)\big\}\ \xi_p\, \hat{U}_{mn} \quad . \qquad \text{(F.9)}
\end{aligned}$$

F.2 Discrete Fourier Transform of the Correlation Term

We recall from Section 4.3.3 that we may write the discrete version of the integral correlation term between values U and g as a dense matrix-vector product Ju. In Equation 4.14 we write this product as an operation on the DFT of the jump distribution and the option value vector on the periodic grid.

We recall that $g(y)$ is a PDF, and that the the points on the grid defined by Equation 4.12 for the correlation Equation 4.14 are defined by integrating $g(y)$ over the DFT cell. Let

$$f_{ij} = f(q_k) = \int_{-h_1/2}^{+h_1/2} \int_{-h_2/2}^{+h_2/2} g(q_k + (z_1, z_2))\ dz_1\ dz_2$$

and note that that

$$f_{ij} \in \mathbb{R}, \quad f_{ij} \geq 0 \quad \text{and} \quad \sum_{ij} f_{ij} = 1 \quad .$$

Thus, taking the Fourier transform

$$\hat{f}_{mn} = \sum_{ij} f_{ij}\ \xi(-im - jn) \quad \text{implies that} \quad \left|\hat{f}_{0,0}\right| \leq 1 \quad \text{and} \quad \left|\hat{f}_{mn}\right| \leq 1$$

We shall require the magnitude of the DFT of the jump distribution and its real and imaginary components

$$\overline{(\hat{f}_{mn})} = \hat{f}_{-m,-n} = \left(\hat{f}^R_{-m,-n} + \hat{f}^I_{-m,-n}\sqrt{-1}\right) \tag{F.10}$$

in the final proof. Using this notation we note that

$$0 \leq \left(1 - \hat{f}^R_{mn}\right) \leq 2 \quad . \tag{F.11}$$

F.3 Discrete Option Value and Error Propagation PIDE

In this section it is useful to relate this analysis to the original form of the problem by using a matrix, vector notation similar to that of Section 4.1. We write the approximation to PIDE 2.3 as a periodic initial value problem on a grid with discrete solution points written in a vector u. It is discretized using a Crank-Nicolson method from step t to $t+1$ with time step weight θ as in Equation 4.6 and a finite difference method selected from Section E. The time step is taken by solving an equation in the form of Equation 4.6

$$\left[\mathsf{I} + (1-\theta)\Delta\tau\ \mathsf{M}\right]\mathsf{u}^{t+1} = \left[\mathsf{I} - \theta\Delta\tau\ \mathsf{M}\right]\mathsf{u}^{t} \tag{F.12}$$

where

$$\mathsf{M} = -\left[\mathsf{G} + \lambda(\mathsf{J} - \mathsf{I}) - r\,\mathsf{I}\right] = -\mathsf{G} - \lambda\,\mathsf{J} + \lambda\,\mathsf{I} + r\,\mathsf{I} \quad . \tag{F.13}$$

Matrix G is defined by coefficients $\mathbf{D}$ and $\mathbf{V}$ (Equations 2.6 and 2.5) and difference Equations E.1 and E.4 with either Equation E.5, E.6 or E.7 for the cross-partial derivative. J is as defined in Equation 4.14. Note that we have not written λ_C or I_c since, for this analysis, the matrix coefficients must be the same for each point in the system.

Let e^t be an arbitrary perturbation error to the solution u. The error propagates by

$$\left[\mathsf{I} + (1-\theta)\Delta\tau\ \mathsf{M}\right]\mathsf{e}^{t+1} = \left[\mathsf{I} - \theta\Delta\tau\ \mathsf{M}\right]\mathsf{e}^{t} \quad . \tag{F.14}$$

The fixed point iteration method solves Equation F.12 with approximate solution z^k at the k-th fixed point iteration solving from time step t to $t+1$

$$\left\{\mathsf{I} + (1-\theta)\Delta\tau\left[-\mathsf{G} + \lambda\mathsf{I} + r\,\mathsf{I}\right]\right\}\mathsf{z}^{k+1} = (1-\theta)\Delta\tau\lambda\mathsf{J}\mathsf{z}^k + \left\{\mathsf{I} - \theta\Delta\tau\left[-\mathsf{G} - \lambda\mathsf{J} + \lambda\mathsf{I} + r\,\mathsf{I}\right]\right\}\mathsf{u}^t \quad . \tag{F.15}$$

For the fixed point iteration we write the solution error $\mathsf{e}^k = \mathsf{u}^{t+1} - \mathsf{z}^k$. The error e^k of the intermediate solution vector z^k propagates by

$$\left\{\mathsf{I} + (1-\theta)\Delta\tau\left[-\mathsf{G} + r\,\mathsf{I} + \lambda\mathsf{I}\right]\right\}\mathsf{e}^{k+1} = (1-\theta)\Delta\tau\lambda\mathsf{J}\mathsf{e}^k \quad . \tag{F.16}$$

F.4 General Approach to the Proof

We shall arrange the Fourier transform of the time step Equation F.14 into a complex valued form for a single coefficient $\hat{E}_{mn}$ of the transform. This must not increase during the time step. The ratio of this coefficient between time step t and $t+1$ is given by

$$\frac{\hat{E}^{t+1}_{mn}}{\hat{E}^{t}_{mn}} = \frac{1 - \theta\Delta\tau\left(-a - b\sqrt{-1} + r + \lambda\right) + \theta\Delta\tau\lambda\hat{f}_{-m,-n}}{1 + (1-\theta)\,\Delta\tau\left(-a - b\sqrt{-1} + r + \lambda\right) - (1-\theta)\,\Delta\tau\lambda\hat{f}_{-m,-n}} \tag{F.17}$$

where a and b represent the real and imaginary contributions of the Fourier transform of the finite difference approximation G. We take the magnitude which must satisfy

$$\begin{aligned}\left|\frac{\hat{E}^{t+1}_{mn}}{\hat{E}^{t}_{mn}}\right|^2 &= \frac{\left|\hat{E}^{t+1}_{mn}\right|^2}{\left|\hat{E}^{t}_{mn}\right|^2} \\ &= \frac{\left\{1-\theta\Delta\tau\left[-a+r+\lambda\left(1-\hat{f}^R_{-m,-n}\right)\right]\right\}^2+\left\{\theta\Delta\tau\left(b+\lambda\hat{f}^I_{-m,-n}\right)\right\}^2}{\left\{1+(1-\theta)\,\Delta\tau\left[-a+r+\lambda\left(1-\hat{f}^R_{-m,-n}\right)\right]\right\}^2+\left\{(1-\theta)\,\Delta\tau\left(b+\lambda\hat{f}^I_{-m,-n}\right)\right\}^2} \leq 1\end{aligned} \tag{F.18}$$

with $\hat{f}_{-m,-n} = \left(\hat{f}^R_{-m,-n} + \hat{f}^I_{-m,-n}\sqrt{-1}\right)$ as in Equation F.10 and by Equation F.11 the term $\lambda\left(1-\hat{f}^R_{-m,-n}\right) > 0$ when $\lambda > 0$

Remark F.1 *The term $(-a+r)$ has a different sign in the numerator and denominator of Equation F.18. We have $r, \lambda \geq 0$. If $a \leq 0$ then Equation F.18 is satisfied for $\theta = 0$ and $\theta = 1/2$ and Theorem 5.3 is proved: the time step is unconditionally strictly stable in the l_2 norm by von Neumann analysis.*

We shall arrange the Fourier transform of fixed point iteration Equation F.16 into a complex valued form for a single coefficient E^k_{mn}. This coefficient must decrease in the iteration. Thus we have the condition

$$\begin{aligned}\frac{\left|\hat{E}^{k+1}_{mn}\right|^2}{\left|\hat{E}^{k}_{mn}\right|^2} &= \left|\frac{(1-\theta)\Delta\tau\,\lambda\hat{f}}{1+(1-\theta)\,\Delta\tau\,(-a+r+\lambda)-(1-\theta)\,\Delta\tau b\sqrt{-1}}\right|^2 \\ &\leq \frac{\left[(1-\theta)\Delta\tau\,\lambda\right]^2}{\left[1+(1-\theta)\,\Delta\tau\,(-a+r+\lambda)\right]^2+\left[\sqrt{-1}\,(1-\theta)\,b\right]^2} \\ &\leq \frac{\left[(1-\theta)\Delta\tau\,\lambda\right]^2}{\left[1+(1-\theta)\,\Delta\tau\,(-a+r+\lambda)\right]^2} < 1 \quad .\end{aligned} \tag{F.19}$$

Note that we have used Equation F.11 again, and that the a of Equation F.19 is identical to that of Equation F.18.

Remark F.2 *For $\theta = 0$ and $\theta = 1/2$ it is sufficient, using Equation F.19, to demonstrate that $a \leq 0$ for the error magnitude to be reduced at each iteration. Thus if $a \leq 0$ then Theorem 6.3 is proved: the fixed point iteration is unconditionally convergent in the l_2 norm by von Neumann analysis.*

In both Remark F.1 and F.2, the stability or, respectively, convergence, is dependent only on the real component of the Fourier transform of the finite difference approximation used to generate G. We shall see that this depends, in turn, only on the discrete form of the diffusion term.

F.5 Analysis of the 7-pt Stencil

To prove the seven-point stencil variant of Remarks F.1 and F.2 we expand the partial derivative term of Equation F.12 in finite differences in Fourier space. We use the cross-partial derivative Equation E.5, which normally would apply for $0 < \rho_v \leq 1$,

$$\begin{aligned}
\Big[v_1\ \hat{\partial}_1^{h_1} + v_2\ \hat{\partial}_2^{h_2} + d_{11}\ \hat{\partial}_{11}^{h_1}{}^2 + d_{22}\ \hat{\partial}_{22}^{h_2}{}^2 + 2\, d_{12}\ \hat{\partial}_{12}^{h_1 h_2 +}\Big] \hat{U}_{mn} \\
= 2 \bigg\{ \left[\frac{v_1}{h_1} \sin(\zeta m) + \frac{v_2}{h_2} \sin(\zeta n) \right] \sqrt{-1} \\
+ \frac{d_{11}}{{h_1}^2} \left[\cos(\zeta m) - 1 \right] + \frac{d_{22}}{{h_2}^2} \left[\cos(\zeta n) - 1 \right] \\
+ \frac{d_{12}}{h_1 h_2} \left[1 + \cos(\zeta m + \zeta n) - \cos(\zeta m) - \cos(\zeta n) \right] \bigg\} \xi_p\ \hat{U}_{mn} \\
= 2 \left\{ a + b\sqrt{-1}\ \right\} \xi_p\ \hat{U}_{mn}
\end{aligned} \tag{F.20}$$

where v_1, v_2 are the elements of vector $\mathbf{V}$ and d_{11}, d_{12}, d_{22} are the elements of tensor $\mathbf{D}$. From this expression

$$a = \left(\frac{\sigma_1}{h_1} \right)^2 \left[\cos(\zeta m) - 1 \right] + \left(\frac{\sigma_2}{h_2} \right)^2 \left[\cos(\zeta n) - 1 \right] + \rho_v \frac{\sigma_1}{h_1} \frac{\sigma_2}{h_2} \left[1 + \cos(\zeta m + \zeta n) - \cos(\zeta m) - \cos(\zeta n) \right] \quad . \tag{F.21}$$

We differentiate a to find its extrema by solving for the simultaneous zeros of

$$\begin{aligned}
\frac{\partial a}{\partial(\zeta m)} &= - \left(\frac{\sigma_1}{h_1} \right)^2 \sin(\zeta m) + \rho_v \frac{\sigma_1}{h_1} \frac{\sigma_2}{h_2} \left(\sin(\zeta m) - \sin(\zeta m + \zeta n) \right) = 0 \\
\frac{\partial a}{\partial(\zeta n)} &= - \left(\frac{\sigma_2}{h_2} \right)^2 \sin(\zeta n) + \rho_v \frac{\sigma_1}{h_1} \frac{\sigma_2}{h_2} \left(\sin(\zeta n) - \sin(\zeta m + \zeta n) \right) = 0 \quad .
\end{aligned}$$

We have four solutions at $(\zeta m, \zeta n) = (\{0, \pi\}, \{0, \pi\})$. Of these, the maximum is $a = 0$ at $(0, 0)$ where all $\cos(\cdot)$ terms of a equal 1. Thus $a \leq 0$ and Remarks F.1 and F.2 hold.

The case using the finite difference $\hat{\partial}_{12}^{h_1 h_2 -}$ for the cross-partial term (Equation E.6) differs only in the leading sign of the d_{12} term and the sign of ζn in the first cosine term. The final result is the same: $a \leq 0$ unconditionally. In neither case is there a time step or correlation sign restriction for strict stability, and the coefficients of the advection term do not appear in the final condition.

F.6 Analysis of the 9-pt Stencil

To prove the nine-point stencil variant of Remarks F.1 and F.2 we expand the partial derivative term of Equation F.12 in finite differences in Fourier space. We use the cross-partial derivative Equation E.7:

$$\begin{aligned}
\Big[v_1\ \hat{\partial}_1^{h_1} + v_2\ \hat{\partial}_2^{h_2} + d_{11}\ \hat{\partial}_{11}^{h_1}{}^2 + d_{22}\ \hat{\partial}_{22}^{h_2}{}^2 + 2\, d_{12}\ \hat{\partial}_{12}^{h_1 h_2 \circ}\Big] \hat{U}_{mn} \\
= 2 \bigg\{ \left[\frac{v_1}{h_1} \sin(\zeta m) + \frac{v_2}{h_2} \sin(\zeta n) \right] \sqrt{-1} \\
+ \frac{d_{11}}{{h_1}^2} \left[\cos(\zeta m) - 1 \right] + \frac{d_{22}}{{h_2}^2} \left[\cos(\zeta n) - 1 \right] - \frac{d_{12}}{h_1 h_2} \sin(\zeta m) \sin(\zeta n) \bigg\} \xi_p\ \hat{U}_{mn} \\
= 2 \left\{ a + b\sqrt{-1}\ \right\} \xi_p\ \hat{U}_{mn}
\end{aligned} \tag{F.22}$$

where v_1, v_2 are the elements of vector $\mathbf{V}$ and d_{11}, d_{12}, d_{22} are the elements of tensor $\mathbf{D}$. From this expression we require

$$a = \left(\frac{\sigma_1}{h_1} \right)^2 \left[\cos(\zeta m) - 1 \right] + \left(\frac{\sigma_2}{h_2} \right)^2 \left[\cos(\zeta n) - 1 \right] + \rho_v \frac{\sigma_1}{h_1} \frac{\sigma_2}{h_2} \left[\sin(\zeta m) \sin(\zeta n) \right] \leq 0 \quad . \tag{F.23}$$

Again, we proceed as in the previous section, differentiating a and finding the zeros which solve

$$\begin{aligned}\frac{\partial a}{\partial(\zeta m)} &= -\left(\frac{\sigma_1}{h_1}\right)^2 \sin(\zeta m) + \rho_v \frac{\sigma_1}{h_1}\frac{\sigma_2}{h_2}\left(\cos(\zeta m)\sin(\zeta n)\right) = 0 \\ \frac{\partial a}{\partial(\zeta n)} &= -\left(\frac{\sigma_2}{h_2}\right)^2 \sin(\zeta n) + \rho_v \frac{\sigma_1}{h_1}\frac{\sigma_2}{h_2}\left(\sin(\zeta m)\cos(\zeta n)\right) = 0 \quad .\end{aligned}$$

We have four solutions at $(\zeta m, \zeta n) = (\{0, \pi\}, \{0, \pi\})$. Of these, $(0, 0)$ is the maximum with $a = 0$. Thus $a \leq 0$ and Remarks F.1 and F.2 hold.

References

[1] M. Abramowitz, I. A. Stegun, Handbook of Mathematical Functions, No. 55 in National Bureau of Standards, Applied Mathematics Series, United States Department of Commerce, 1972.

[2] A. Almendral, C. Oosterlee, Numerical valuation of options with jumps in the underlying, Applied Numerical Mathematics 53 (2005) 1–18.

[3] K. I. Amin, Jump diffusion option valuation in discrete time, Journal of Finance 48 (5) (1993) 1833–1863.

[4] L. Andersen, J. Andreasen, Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing, Review of Derivatives Research 4 (3) (2000) 231–262.

[5] G. Barles, P. E. Souganidis, Convergence of approximation schemes for fully nonlinear second order equations, Asymptotic Analysis 4 (2) (1991) 271–283.

[6] F. Black, M. Scholes, The pricing of options and corporate liabilities, The Journal of Political Economy 81 (3) (1973) 637–654.

[7] R. N. Bracewell, The Fourier transform and its applications, 3rd ed., McGraw-Hill, 2000.

[8] M. Briani, C. L. Chioma, R. Natalini, Convergence of numerical schemes for viscosity solutions to integro-differential degenerate parabolic problems arising in financial theory, Numerische Mathematik 98 (4) (2004) 607–646.

[9] M. Briani, R. Natalini, G. Russo, Implicit-explicit numerical schemes for jump–diffusion processes, iAC Report 38 (4/2004) (2004).

[10] M. Broadie, Y. Yamamoto, Application of the fast Gauss transform to option pricing, Management Science 49 (8) (2003) 1071–1088.

[11] R. Cont, P. Tankov, Financial Modelling with Jump Processes, CRC Financial Mathematics Series, Chapman and Hall, 2004.

[12] R. Cont, E. Voltchkova, A finite difference scheme for option pricing in jump diffusion and exponential Lévy models, SIAM Journal on Numerical Analysis 43 (4) (2005) 1596–1626.

[13] Y. d'Halluin, Numerical methods for real options in telecommunications, Ph.D. thesis, School of Computer Science, Faculty of Mathematics, University of Waterloo, Waterloo, Ontario, Canada (2004).

[14] Y. d'Halluin, P. A. Forsyth, G. Labahn, A penalty method for American options with jump diffusion processes, Numerische Mathematik 97 (2) (2004) 321–352.

[15] Y. d'Halluin, P. A. Forsyth, K. R. Vetzal, Robust numerical methods for contingent claims under jump diffusion processes, IMA Journal of Numerical Analysis 25 (1) (2005) 87–112.

[16] P. DuChateau, D. Zachmann, Applied partial differential equations, Dover Publications, Mineola, New York, U.S.A., 2002, (1989 edition, reprint).

[17] A. J. W. Duijndam, M. A. Schonewille, Nonuniform fast Fourier transform, Geophysics 64 (2) (1999) 539–551.

[18] J. D. Fonseca, M. Grasselli, C. Tebaldi, Option pricing when correlations are stochastic: an analytical framework, Working Paper ESILV, RR-33, Ecole Supérieure d'Ingénieurs Léonard de Vinci, Département Mathématiques et Ingénierie Financié're, Paris (July 2006).

[19] P. A. Forsyth, K. R. Vetzal, Quadratic convergence for valuing American options using a penalty method, SIAM Journal on Scientific Computing 23 (6) (2002) 2095–2122.

[20] A. Friedman, Partial Differential Equations of Parabolic Type, Prentice-Hall, Englewood Cliffs, NJ, U.S.A., 1964.

[21] K. O. Friedrichs, The identity of weak and strong extensions of differential operators, Transactions of the American Mathematical Society 55 (1) (1944) 132–151.

[22] P. Friz, J. Gatheral, Valuing volatility derivatives as an inverse problem, Quantitative Finance 5 (6) (2005) 531–542.

[23] M. G. Garroni, J. L. Menaldi, Green functions for second order parabolic integro-differential problems, No. 275 in Pitman Research Notes in Mathematics, Longman Scientific and Technical, Harlow, Essex, UK, 1992.

[24] E. R. Jakobsen, K. H. Karlsen, C. L. Chioma, Error estimates for approximate solutions to Bellman equations associated with controlled jump-diffusions, department of Mathematical Sciences, Norwegian University of Science and Technology (October 2006).

[25] R. Kangro, R. Nicolaides, Far field boundary conditions for Black–Scholes equations, SIAM Journal on Numerical Analysis 38 (4) (2000) 1357–1368.

[26] S. Kotz, Continuous multivariate distributions, 2 ed., vol. 1, Wiley, New York, 2000.

[27] S. G. Kou, A jump-diffusion model for option pricing, Management Science 48 (8) (2002) 1086–1101.

[28] J. F. B. M. Kraaijevanger, H. W. J. Lenferink, M. N. Spijker, Stepsize restrictions for stability in the numerical solution of ordinary and partial differential equations, Journal of Computational and Applied Mathematics (1987) 67–81.

[29] P. K. Lamm, A survey of regularization methods for first-kind Volterra equations, in: D. Colton, H. W. Engl, A. Louis, J. R. McLaughlin, W. Rundell (eds.), Surveys on Solution Methods for Inverse Problems, Springer, Vienna, New York, 2000, pp. 53–82.

[30] H. W. J. Lenferink, M. N. Spijker, On the use of stability regions in the numerical analysis of initial value problems, Mathematics of Computation 57 (195) (1991) 221–237.

[31] A. L. Lewis, A simple option formula for general jump-diffusion and other exponential Lévy processes, Tech. rep., Envision Financial Systems and OptionCity.net (August 2001).

[32] A. L. Lewis, Fear of jumps, Wilmott Magazine (2002) 60–67.

[33] A. W. Marshall, I. Olkin, A multivariate exponential distribution, Journal of the American Statistical Association 62 (317) (1967) 30–44.

[34] S. H. Martzoukos, Contingent claims on foreign assets following jump-diffusion processes, Review of Derivatives Research 6 (1) (2003) 27–45.

[35] A.-M. Matache, P.-A. Nitsche, C. Schwab, Wavelet Galerkin pricing of American options on Lévy driven assets, Tech. Rep. 2003-06, Seminar for Applied Mathematics, Swiss Federal Institute of Technology, Zurich (2003).

[36] A.-M. Matache, C. Schwab, T. P. Wihler, Fast numerical solution of parabolic integrodifferential equations with applications in finance, SIAM Journal on Scientific Computing 27 (2) (2005) 369–393.

[37] A.-M. Matache, T. von Petersdorff, C. Schwab, Fast deterministic pricing of options on Lévy driven assets, Mathematical Modelling and Numerical Analysis 38 (1) (2004) 37–72.

[38] R. C. Merton, Theory of rational option pricing, The Bell Journal of Economics and Management Science 4 (1) (1973) 141–183.

[39] R. C. Merton, Option pricing when underlying stock returns are discontinuous, Journal of Financial Economics 3 (1-2) (1976) 125–144.

[40] S. Mulinacci, An approximation of American option prices in a jump-diffusion model, Stochastic Processes and their Applications 62 (1) (1996) 1–17.

[41] B. Oksendal, A. Sulem, Applied Stochastic Control of Jump Diffusions, Springer Verlag, Berlin, 2000.

[42] H. Pham, Optimal stopping of controlled jump-diffusion processes: a viscosity solutions approach, Journal of Mathematical Systems, Estimation and Control 8 (1) (1998) 1–27.

[43] R. J. Plemmons, M-matrix characterizations. I–nonsingular M-matrices, Linear Algebra and its Applications 18 (2) (1977) 175–188.

[44] D. M. Pooley, P. A. Forsyth, K. R. Vetzal, Numerical convergence properties of option pricing PDEs with uncertain volatility, IMA Journal of Numerical Analysis 23 (2) (2003) 241–267.

[45] D. Potts, G. Steidl, M. Tasche, Fast Fourier Transforms for Nonequispaced Data: A Tutorial, chap. 12, Applied and Numerical Harmonic Analysis Series, Birkhauser, Boston, 2001, pp. 249–274.

[46] R. D. Richtmyer, K. W. Morton, Difference methods for initial-value problems, 2nd edition, Interscience Publishers, New York, 1967.

[47] Y. Saad, Iterative Methods for Sparse Linear Systems (2nd ed.), Society for Industrial and Applied Mathematics, Philadelphia, PA, U.S.A., 2003.

[48] F. Straetemans, Resolvent conditions for discretizations of diffusion-convection-reaction equations in several space dimensions, Applied Numerical Mathematics 28 (1) (1998) 45–67.

[49] R. M. Stulz, Options on the minimum or the maximum of two risky assets, Journal of Financial Economics 10 (2) (1982) 161–185.

[50] A. F. Ware, Fast approximate Fourier transforms for irregularly spaced data, SIAM Review 40 (4) (1998) 838–856.

[51] H. A. Windcliff, P. A. Forsyth, K. R. Vetzal, Analysis of the stability of the linear boundary condition for the Black–Scholes equation, Journal of Computational Finance 8 (1) (2004) 65–92.

[52] X. L. Zhang, Numerical analysis of American option pricing in a jump-diffusion model, Mathematics of Operations Research 22 (3) (1997) 668–690.

[53] R. Zvan, P. A. Forsyth, K. R. Vetzal, Penalty methods for American options with stochastic volatility, Journal of Computational and Applied Mathematics 91 (2) (1998) 199–218.

[54] R. Zvan, P. A. Forsyth, K. R. Vetzal, A finite volume approach for contingent claims valuation, IMA Journal of Numerical Analysis 21 (3) (2001) 703–731.

[55] R. Zvan, P. A. Forsyth, K. R. Vetzal, Negative coefficients in two-factor option pricing models, Journal of Computational Finance 7 (1) (2003) 37–73.