UCD GEARY INSTITUTE DISCUSSION PAPER SERIES

Pricing Options under Heston's Stochastic Volatility Model via Accelerated Explicit Finite Differencing Methods

Conall O'Sullivan
University College Dublin*

and

Stephen O'Sullivan
Dublin Institute of Technology

Current Draft: June 2010

*Address for correspondence: Conall O'Sullivan, Banking and Finance, The Michael Smurfit Graduate School of Business, University College Dublin, Blackrock, Co. Dublin, Ireland. Email: conall.osullivan@ucd.ie. The authors would like to thank the participants of the Bachelier World Congress in Finance, London, July 2008 and the participants of the Quantitative Methods in Finance Conference, Sydney, December 2009.

Pricing Options under Heston's Stochastic Volatility Model via Accelerated Explicit Finite Differencing Methods

Conall O'Sullivan
University College Dublin*
and
Stephen O'Sullivan
Dublin Institute of Technology

Current draft: June 2010.

Abstract

We present an acceleration technique, effective for explicit finite difference schemes describing diffusive processes with nearly symmetric operators, called Super-Time-Stepping (STS). The technique is applied to the two-factor problem of option pricing under stochastic volatility. It is shown to significantly reduce the severity of the stability constraint known as the Courant-Friedrichs-Lewy condition whilst retaining the simplicity of the chosen underlying explicit method.

For European and American put options under Heston's stochastic volatility model we demonstrate degrees of acceleration over standard explicit methods sufficient to achieve comparable, or superior, efficiencies to a benchmark implicit scheme. We conclude that STS is a powerful tool for the numerical pricing of options and propose them as the method-of-choice for exotic financial instruments in two and multi-factor models.

*Address for correspondence: Conall O'Sullivan, Banking and Finance, The Michael Smurfit Graduate School of Business, University College Dublin, Blackrock, Co. Dublin, Ireland. Email: conall.osullivan@ucd.ie. The authors would like to thank the participants of the Bachelier World Congress in Finance, London, July 2008 and the participants of the Quantitative Methods in Finance Conference, Sydney, December 2009.

1 Introduction

The Black-Scholes partial differential equation (PDE), Black & Scholes (1973) and Merton (1973), laid the foundations for modern derivatives pricing. However the assumptions made in the Black-Scholes model are known to be overly restrictive. In particular, the Black-Scholes model assumes that the underlying asset price follows a geometric Brownian motion with a fixed volatility.

Many derivative pricing models have been developed subsequently which use more sophisticated stochastic processes for the underlying asset which result in a better match to empirically observed details.

Using such stochastic processes is often more straightforward than relaxing the Black-Scholes assumptions to allow for discrete time trading, transactions costs and other market imperfections. Examples of more realistic stochastic processes include: jump-diffusion (Merton, 1976); Lévy (Cont & Tankov, 2004); stochastic volatility (SV) (Heston, 1993); stochastic volatility jump-diffusion (Bates, 1996); and also combinations of these that exhibit SV as well as jumps in both the asset price and volatility (Duffie et al., 2000).

In many of these cases there may be no analytical solution to the PDE describing the corresponding vanilla European option price. The PDE describing the American analog does not have a closed form solution in any of these models.

When a closed form solution does not exist one popular way to proceed is to solve the PDE numerically using finite difference (FD) methods, see Wilmott et al. (1993) and Tavella & Randall (2000).

Despite their increased complexity numerical valuations of options using FD methods based on implicit discretizations are usually superior in terms of efficiency to approaches based on conventional explicit discretizations. The principal reason for this is the Courant-Friedrichs-Lewy (CFL) stability constraint on explicit schemes which limits the size of the time step relative to the *square* of the spatial step size. In this paper, the restriction that the CFL constraint imposes is reduced significantly using an acceleration technique for explicit algorithms known as Super-Time-Stepping (STS). STS was recently introduced to the finance literature for the first time in the one-factor Black-Scholes setting by O'Sullivan & O'Sullivan (2009) for vanilla European and American put options.

While the behaviour of STS has only been analytically established for symmetric operators Alexiades et al. (1996), O'Sullivan & O'Sullivan (2009) described the implementation

of a novel splitting approach to treating non-symmetric operators. This splitting method is based on the unique decomposition of the operator into its symmetric and skew-symmetric parts. The former may then be treated via STS while the latter is efficiently integrated via a suitable scheme in a procedure introduced by O'Sullivan & Downes (2006, 2007). In this way, O'Sullivan & O'Sullivan (2009) demonstrated formal stability for the accelerated scheme and showed that the splitting procedure is unnecessary in the case of a *weakly* non-symmetric operator.

In this work, we follow that precedent and do not split the operator. As will be shown, for the problems under consideration here, the performance of STS does not appear to suffer any discernable adverse implications to its stability properties as a result.

The contribution of this paper to computational finance is to extend the application of the STS acceleration technology to the two-factor problem of pricing European and American put options under Heston's SV model. STS is compared to a number of standard FD methods used frequently in the literature for SV options pricing. We demonstrate that the efficiencies attained using the STS algorithm are comparable, and often superior, to those of common implicit differencing techniques. Crucially, this acceleration is achieved without any significant increase in implementation complexity relative to the underlying standard explicit scheme.

There exist many methods to numerically solve European and American style options using the PDE approach. Successive overrelaxation (SOR) is the most popular iterative method for obtaining the solution in the European option case (see Crank (1984) for details) while projected SOR (PSOR) (PSOR) (Cryer, 1971) is one of the most popular methods used to solve higher dimensional linear complementarity problems (LCPs) that result from pricing American options. With (P)SOR, as we refine the computational grid to obtain more accurate option prices the number of iterations required to converge grows, however this effect can be reduced by appropriate (problem dependent) tuning of the overrelaxation parameter.

Other approaches used to numerically solve the option pricing PDE under the SV model of Heston include: an ILU pre-conditioned conjugate gradient method with a penalty term devised by Zvan et al. (1998) to handle the early exercise feature of American options; a multigrid method used by Clarke & Parrot (1999) and Oosterlee (2003) to price American options; a Hopscotch scheme applied by Kurpiel & Roncalli (2000) to price American options; a finite element approach constructed by Winkler et al. (2001) to price European and

barrier options.

A number of schemes used to solve LCPs were also compared by Ikonen & Toivanen (2007a,b) including a PSOR method, a projected multigrid method, an operator splitting method and a component wise splitting method. All of the schemes examined by these authors displayed varying levels of superiority over the PSOR method in efficiently pricing American options under the SV model.

All of the methods mentioned above, with the exception of the Hopscotch method which is a mixed implicit-explicit method, are implicit algorithms which are faster but significantly more complex in their implementation than their explicit differencing cousins. In particular, these methods are global in the sense that the entire solution must be available in order to advance any point and are therefore inherently difficult to parallelize.

The crucial difference in the STS algorithm allows explicit FD schemes to become viable alternatives to their implicit counterparts. The relative simplicity of explicitly differenced PDEs is inherited by the scheme including the local nature of the FD stencil. Parallelization is therefore a trivial endeavor in this context.

This paper is arranged as follows.

Section II reviews Heston's SV model, the corresponding PDEs describing vanilla European and American option prices, and the associated boundary conditions. Section III describes the implementation details required to construct the spatial (asset price and variance) grid and reviews the standard FD schemes used to benchmark the STS implementation employed. Section IV introduces the STS technique. The application of STS to option pricing under SV is the main contribution of the paper. Section V contains comparative timings and discussion of results. Finally, in Section VI, we offer concluding remarks.

2 Review

This section provides a review of the option pricing PDE and associated boundary conditions for both European and American options under Heston's SV model.

2.1 Heston's Stochastic Volatility Model

Heston's risk neutral SV model can be written as follows:

$$\begin{aligned} \mathrm{d}x_t &= rx_t\mathrm{d}t + \sqrt{y_t}x_t\mathrm{d}z_t, \\ \mathrm{d}y_t &= \left\{\alpha\left(\beta - y_t\right) - \lambda\gamma\sqrt{y_t}\right\}\mathrm{d}t + \gamma\sqrt{y_t}\mathrm{d}w_t, \\ \rho\mathrm{d}t &= \mathrm{d}z_t\mathrm{d}w_t, \end{aligned} \tag{1}$$

where x_t and y_t are the asset price and variance at time t respectively, r is the risk-free rate, α is the mean reversion of the variance, β is the long run mean of the variance, γ is the volatility of the variance, ρ is the correlation of the asset price and the variance and λ is the market price of risk. We denote by $u(x, y, \tau)$ an option price with a time-to-maturity of $\tau = T - t$ where t is the observation time and T is the expiry.

2.2 European Put Options

European options satisfy the following PDE under these asset price dynamics:

$$\mathcal{L}u = \frac{\partial u}{\partial \tau} + \mathcal{A}u = 0, \tag{2}$$

where $\mathcal{L}$ is a generalised Black-Scholes type operator and $\mathcal{A}$ is its spatial component defined by

$$\mathcal{A}u = -\frac{1}{2}yx^2u_{xx} - \rho\gamma yxu_{xy} - \frac{1}{2}\gamma^2 yxu_{yy} - rxu_x - \left\{\alpha\left(\beta - y\right) - \lambda\gamma\sqrt{y}\right\}u_y + ru. \tag{3}$$

The payoff, or initial condition, on a Eurpoean put option is

$$u(x, y, 0) = g(x, y) = \max\left[E - x, 0\right]. \tag{4}$$

European put options satisfy

$$u(0, y, \tau) = e^{-r\tau}g(0, y) \tag{5}$$

on the boundary of $x = 0$.

The PDE on the boundary $y = 0$ reduces to

$$\mathcal{L}u = \frac{\partial u}{\partial \tau} + \mathcal{A}u = \frac{\partial u}{\partial \tau} - rxu_x - \alpha\beta u_y + ru = 0. \tag{6}$$

We use the following asymptotic boundary conditions at $x = \infty$ and $y = \infty$:

$$\lim_{x\to\infty} \frac{\partial u(x,y,\tau)}{\partial x} = 0; \tag{7}$$

$$\lim_{y\to\infty} \frac{\partial u(x,y,\tau)}{\partial y} = 0. \tag{8}$$

This is the equivalent system to that described by Clarke & Parrot (1999), Ikonen & Toivanen (2007b), Oosterlee (2003) and Zvan et al. (1998).

2.3 American Put Options

American options may be excercised early thereby demanding the price u must at least be as large as the early exercise value g. This leads to the early exercise constraint $u(x, y, \tau) \geq g(x, y)$.

In the continuation region where the contraint is inactive u satisfies the same PDE as the European option i.e. $\mathcal{L}u = 0$ as given by Eq. 2.

Combining these relations we can write the American option pricing problem as a time dependent LCP (for example see Ikonen & Toivanen (2007b))

$$\begin{cases} \mathcal{L}u \geq g, \quad u \geq g, \\ (\mathcal{L}u)\,(u - g) = 0, \end{cases} \tag{9}$$

in a domain $\{(x, y, \tau)\,|x \geq 0, y \geq 0, \tau \in [0, T]\,\}$.

Note that the payoff or initial condition of the American put option is the same as the European option.

At $x = 0$, the pertinent boundary condition is

$$u(0, y, \tau) = g(0, y). \tag{10}$$

The linear complementarity conditions on the boundary $y = 0$ are

$$\begin{cases} \mathcal{L}u = \frac{\partial u}{\partial \tau} + \mathcal{A}u = \frac{\partial u}{\partial \tau} - rxu_x - \alpha\beta u_y + ru \geq g, \quad u \geq g, \\ (\mathcal{L}u)\,(u - g) = 0. \end{cases} \tag{11}$$

The boundary conditions at $x = \infty$ and $y = \infty$ are the same as those for the European put option as given by Eq. 7 and Eq. 8.

3 Implementation Details

In this section we provide the implementation details of the FD methods considered in this work.

Following Ikonen & Toivanen (2007b), we use grid generating functions to increase computational efficiency by decreasing the density of mesh points away from regions of interest. We also employ upwinding when the PDE becomes convection dominated to avoid spurious oscillations in these regions as described by Ikonen & Toivanen (2007b). Unlike these authors, however, we use a conventional nine point stencil as opposed to their seven point prescription.

3.1 Spatial Discretization on Non-Uniform Grids

The FD discretizations are constructed on a non-uniform grid

$$(x_i, y_j, \tau_k) \in \{0 = x_0, \ldots, x_m = X\} \times \{0 = y_0, \ldots, y_n = Y\} \times \{0 = \tau_0, \ldots, \tau_l = T\}. \quad (12)$$

Standard FD methods are used to discretise the spatial operator $\mathcal{A}$ in Eq. 3. Centred FD is used for the diffusion terms and, in most cases, for the convection terms. However, at certain boundaries, and in areas where convection dominates over diffusion, upwinding is beneficial and one-sided differences are used. More details on the discretization scheme used at the boundaries and on the implementation of upwinding techniques are given in A and B respectively.

We now present a nine point stencil with respect to a reference point (x_i, y_j).

The spatial intervals used in the discretization are then given as

$$h_l = x_i - x_{i-1}, \quad h_r = x_{i+1} - x_i, \quad h_d = y_j - y_{j-1}, \quad \text{and} \quad h_u = y_{j+1} - y_j. \quad (13)$$

For clarity of notation from this point on we denote $u(x_i, y_j)$ as $u_{i,j}$.

The central FD schemes used for the diffusion components at the inner points of the

computional mesh (away from the boundaries) are

$$\begin{aligned}
\frac{\partial^2 u}{\partial x^2} &\approx a_l^D u_{i-1,j} + a^D u_{i,j} + a_r^D u_{i+1,j},\\
\frac{\partial^2 u}{\partial y^2} &\approx b_d^D u_{i,j-1} + b^D u_{i,j} + b_u^D u_{i,j+1},\\
\frac{\partial^2 u}{\partial x \partial y} &\approx c_{ld}^D u_{i-1,j-1} + c_l^D u_{i-1,j} + c_{lu}^D u_{i-1,j+1}\\
&\quad + c_d^D u_{i,j-1} + c^D u_{i,j} + c_u^D u_{i,j+1}\\
&\quad + c_{rd}^D u_{i+1,j-1} + c_r^D u_{i+1,j} + c_{ru}^D u_{i+1,j+1},
\end{aligned} \tag{14}$$

where

$$\begin{aligned}
a_l^D &= \frac{2}{h_l\left(h_l+h_r\right)}, \quad a^D = \frac{-2}{h_l h_r}, \quad a_r^D = \frac{2}{h_r\left(h_l+h_r\right)},\\
b_d^D &= \frac{2}{h_d\left(h_d+h_u\right)}, \quad b^D = \frac{-2}{h_d h_u}, \quad b_u^D = \frac{2}{h_u\left(h_d+h_u\right)},\\
c_{ld}^D &= a_l^C b_d^C, \quad c_l^D = a_l^C b^C, \quad c_{lu}^D = a_l^C b_u^C,\\
c_d^D &= a^C b_d^C, \quad c^D = a^C b^C, \quad c_u^D = a^C b_u^C,\\
c_{rd}^D &= a_r^C b_d^C, \quad c_r^D = a_r^C b^C, \quad c_{ru}^D = a_r^C b_u^C.
\end{aligned} \tag{15}$$

The FD schemes used for the convection components at any inner points of the computional mesh where upwinding is not necessary are

$$\begin{aligned}
\frac{\partial u}{\partial x} &\approx a_l^C u_{i-1,j} + a^C u_{i,j} + a_r^C u_{i+1,j},\\
\frac{\partial u}{\partial y} &\approx b_d^C u_{i,j-1} + b^C u_{i,j} + b_u^C u_{i,j+1},
\end{aligned} \tag{16}$$

where

$$\begin{aligned}
a_l^C &= \frac{-h_r}{h_l\left(h_l+h_r\right)}, \quad a^C = \frac{h_r-h_l}{h_l h_r}, \quad a_r^C = \frac{h_l}{h_r\left(h_l+h_r\right)},\\
b_d^C &= \frac{-h_u}{h_d\left(h_d+h_u\right)}, \quad b^C = \frac{h_u-h_d}{h_d h_u}, \quad b_u^C = \frac{h_d}{h_u\left(h_d+h_u\right)}.
\end{aligned} \tag{17}$$

Substituting these approximations into Heston's PDE for European options, Eq. 2, we

have

$$\begin{aligned}
&\frac{\partial u}{\partial \tau} - \left(\frac{1}{2}yx^2a^D + \rho\gamma yxc^D + \frac{1}{2}\gamma^2 b^D + rxa^C + \alpha\left(\beta - y\right)b^C\right)u_{i,j} \\
&\quad - \left(\frac{1}{2}yx^2a_l^D + \rho\gamma yxc_l^D + rxa_l^C\right)u_{i-1,j} \\
&\quad - \left(\frac{1}{2}yx^2a_r^D + \rho\gamma yxc_r^D + rxa_r^C\right)u_{i+1,j} \\
&\quad - \left(\rho\gamma yxc_d^D + \frac{1}{2}\gamma^2 b_d^D + +\alpha\left(\beta - y\right)b_d^C\right)u_{i,j-1} \\
&\quad - \left(\rho\gamma yxc_u^D + \frac{1}{2}\gamma^2 b_u^D + +\alpha\left(\beta - y\right)b_u^C\right)u_{i,j+1} \\
&\quad - \left(\rho\gamma yxc_{ld}^D\right)u_{i-1,j-1} - \left(\rho\gamma yxc_{lu}^D\right)u_{i-1,j+1} \\
&\quad - \left(\rho\gamma yxc_{rd}^D\right)u_{i+1,j-1} - \left(\rho\gamma yxc_{ru}^D\right)u_{i+1,j+1} = 0.
\end{aligned} \tag{18}$$

The governing PDE may then written compactly as

$$\mathcal{L}u = \frac{\partial u}{\partial \tau} + \mathcal{A}u = 0, \tag{19}$$

in which $\mathcal{A}$ is a nine component operator matrix given by

$$\mathcal{A} = \begin{bmatrix} A_{lu} & A_u & A_{ru} \\ A_l & A_c & A_r \\ A_{ld} & A_d & A_{rd} \end{bmatrix}, \tag{20}$$

with

$$\begin{aligned}
A_c &= -\left(\frac{1}{2}yx^2a^D + \rho\gamma xyc^D + \frac{1}{2}\gamma^2 yb^D + rxa^C + \alpha\left(\beta - y\right)b^C - r\right), \\
A_l &= -\left(\frac{1}{2}yx^2a_l^D + \rho\gamma xyc_l^D + rxa_l^C\right), \\
A_r &= -\left(\frac{1}{2}yx^2a_r^D + \rho\gamma xyc_r^D + rxa_r^C\right), \\
A_d &= -\left(\frac{1}{2}\gamma^2 yb_d^D + \rho\gamma xyc_d^D + \alpha\left(\beta - y\right)b_d^C\right), \\
A_u &= -\left(\frac{1}{2}\gamma^2 yb_u^D + \rho\gamma xyc_u^D + \alpha\left(\beta - y\right)b_u^C\right), \\
A_{ld} &= -\rho\gamma xyc_{ld}^D, \quad A_{lu} = -\rho\gamma xyc_{lu}^D \\
A_{rd} &= -\rho\gamma xyc_{rd}^D, \quad A_{ru} = -\rho\gamma xyc_{ru}^D.
\end{aligned} \tag{21}$$

We have relegated discussion of the discretization techniques employed at the boundaries to A, and on the interior in zones where upwinding is necessitated by the dominance of convection effects in the solution to B.

The approach described above is easily applied to the case of American options as given by the time dependent LCP Eq. 9.

3.2 Grid Generating Functions

Grid generating functions are used to increase the density of the computational nodes around regions of importance in order to increase the efficiency of the method i.e. to reach a desired accuracy level with less grid points. The grid in the $x-$direction is constructed according to

$$x_{i+1} = x_i + h_r(x_i) \text{ for } i = 0, \ldots, m, \tag{22}$$

with the functional form used for the grid generating function in the $x-$direction, $h_r(x_i)$, given by:

$$h_r(x_i) = E + \frac{c}{p}\text{sinh}\left(px_i + \text{arcsinh}\left(-\frac{p}{c}E\right)\right) \text{ for } i = 0, \ldots, m. \tag{23}$$

as described by Kluge (2002).

This grid generating function increases the number of nodes around the exercise price E. The parameter c controls the density of the grid points close to E relative to the density of the grid points at $x_{\max}$. The parameter p is obtained by requiring $x_0 + \sum_{i=1}^{m} h_r(x_i) = x_{\max}$ and solving numerically for p.

By way of illustration, Figure 1(a) depicts the non-uniform grid step size versus the stock price for $x_0 = 0, x_{\max} = 20, m = 128, p = 0.2177$ and $c = 0.5$. These parameters result in a ratio of step sizes in the x-direction at $x_{\max}$ and E of 4.3939.

In the $y-$direction a linear grid generating function, $h_u(y_j)$, is used where

$$y_{j+1} = y_j + h_u(y_j) \text{ for } j = 0, \ldots, n. \tag{24}$$

The grid generating function $h_u(y_j)$ increases the density of grid points at low variance values relative to high variance values and is specified by

$$h_u(y_j) = \frac{q^{j/n} - 1}{q - 1} y_j \text{ for } j = 0, \ldots, n. \tag{25}$$

The parameter q is chosen to ensure that a given reference value in variance occurs on some point on the grid, y_k. For the tests under consideration in this work we choose $y_k = 0.0625$ or $y_k = 0.25$.

Figure 1(b) plots the non-uniform grid step size against the variance y for $y_k = 0.0625, n = 64$ and $q = 4.0734$. In this illustrative case, the ratio of step sizes in the y-direction at $y_{\max}$ and y_k is 2.1566.

3.3 Time Discretization

In this section the explict, implicit and Crank-Nicolson (CN) time discretization methods are briefly described.

The Heston PDE for European options over the entire solution space can be written as

$$\frac{\partial \mathbf{u}}{\partial \tau} + \mathbf{A}\mathbf{u} = 0, \tag{26}$$

where $\mathbf{A}$ is a block tridiagonal $(m+1)(n+1) \times (m+1)(n+1)$ matrix, $\mathbf{u}$ is a vector of length $(m+1)(n+1)$, m is the number of steps in the $x-$direction and n is the number of steps in the $y-$direction. The vector $\mathbf{u}$ and tridiagonal matrix $\mathbf{A}$ are constructed by stacking each solution u_{ij} and corresponding nine component operator matrix $\mathcal{A}$ into a column vector and block tridiagonal matrix respectively, for $i = 0, \ldots, m$ and $j = 0, \ldots, n$.

We begin by introducing the θ-method of discretization. Applied to Eq. 26 we obtain

$$\begin{aligned} &\frac{\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}}{\Delta\tau} + \theta \mathbf{A}\mathbf{u}^{(k+1)} + (1-\theta)\,\mathbf{A}\mathbf{u}^{(k)} = 0 \\ \Rightarrow &\left(\mathbf{I} + \theta\Delta\tau\mathbf{A}\right)\mathbf{u}^{(k+1)} - \left(\mathbf{I} - (1-\theta)\,\Delta\tau\mathbf{A}\right)\mathbf{u}^{(k)} = 0, \text{ for } k = 0, 1, \ldots, l-1. \end{aligned}$$

This can be written more compactly as

$$\mathbf{B}\mathbf{u}^{(k+1)} - \mathbf{C}\mathbf{u}^{(k)} = 0, \tag{27}$$

where $\mathbf{B} = \mathbf{I} + \theta\Delta\tau\mathbf{A}$ and $\mathbf{C} = \mathbf{I} - (1-\theta)\,\Delta\tau\mathbf{A}$.

For American options the scheme in Eq. 27 becomes a time dependent LCP as described in Eq. 9

$$\begin{cases} \mathbf{B}\mathbf{u}^{(k+1)} \geq \mathbf{C}\mathbf{u}^{(k)}, \ \mathbf{u}^{(k+1)} \geq \mathbf{g} \\ \left(\mathbf{B}\mathbf{u}^{(k+1)} - \mathbf{C}\mathbf{u}^{(k)}\right)^T \left(\mathbf{u}^{(k+1)} - \mathbf{g}\right) = 0, \end{cases} \tag{28}$$

where $\mathbf{g}$ is the early exercise value of the option.

When $\theta = 0$ the θ-method corresponds to the explicit Euler scheme. This scheme is first order accurate in time and second order accurate in space. The Euler scheme is very simple to implement, however, stability depends on the size of the time step which is in turn dictated by the spatial step size and the coefficients of the governing PDE (see Wilmott et al. (1993); Tavella & Randall (2000) for further details). In particular when correlation $\rho = 0$, the theoretical upper limit on the time step is given by

$$\Delta\tau_{\text{expl}} \leq \min_{\substack{0\leq i\leq m \\ 0\leq j\leq n}} \left\{ \left[\frac{x_i^2 y_j}{h\left(x_i\right)^2} + \frac{\gamma^2 y_j}{h\left(y_j\right)^2} \right]^{-1} \right\}. \tag{29}$$

In the more general case of non-zero correlation Eq. 29 is found to be an effective estimate of the critical upper limit on the time step in the test cases considered in this paper. This constraint on the time step is known as the Courant-Friedrichs-Lewy (CFL) stability constraint and can be severely restrictive.

When $\theta = 1$ the θ-method corresponds to the fully implicit scheme. The fully implicit scheme is first order accurate in time, second order accurate in space and has no limitations on the size of the time-step for stability, however, the desired accuracy of the solution still imposes a constraint on the minimum number of time steps that may be used.

An alternative approach frequently used in the literature is the CN scheme which corresponds to $\theta = 1/2$. The CN scheme is second order accurate in time and space. Similarly to the fully implicit method, CN has no limitations on the size of the time-step for stability. However the CN method can lead to solutions with spurious oscillations if the initial value is not smooth (which is the case for option payoffs). To alleviate this problem Rannacher (1984) time-stepping is typically used in the CN algorthim[1].

In the fully implicit and CN schemes, LCPs can be solved iteratively at each time step by employing any of a number of methods including PSOR, projected multigrid, and the penalty method (see Clarke & Parrot (1999); Oosterlee (2003); Ikonen & Toivanen (2007b)). Direct methods also exist such as the formulation presented by Ikonen & Toivanen (2007b) of the Brennan & Schwartz (1977) UL decomposition algorithm.

[1] Rannacher time-stepping is where the first few time steps are performed with the fully implicit scheme with a time step size of $\Delta\tau/2$ and thereafter the time steps are performed with the CN scheme with a time step size of $\Delta\tau$. We use the first four steps with the implicit scheme and the remaining $l-4$ steps with the CN scheme.

We elect CN with PSOR, denoted CN-PSOR, as the standard benchmark scheme for comparison with the proposed STS approach in tests solving Eq. 28. CN-PSOR is a suitable choice due to its competitive performance with respect to other methods and its widespread usage (e.g. Tavella & Randall (2000); Wilmott et al. (1993)). A short description of the PSOR algorithm is included in C for the reader's reference.

The equation for the European option price, Eq. 2, may be solved analytically (Heston, 1993). However, as analytical solutions do not exist in general for exotic options with European payoff functions, we test the efficacy of STS with respect to CN-SOR, a CN scheme iteratively solved via simple successive overrelaxation (SOR).

Finally, the standard explicit scheme to first order accurate in time, denoted EXPL-1, is included in the tests to assess the relative acceleration achieved by STS over its base scheme in both the American and European cases.

4 Super-Time-Stepping

In the previous section we reviewed a number of well known time discretization approaches frequently used in FD methods for the integration of time-dependent PDEs. This section introduces an alternative time discretization method known as super-time-stepping (STS) which can be used to accelerate conventional explicit schemes for parabolic problems with *nearly* symmetric positive definite evolution operators.

In the following, we shall use the description of Alexiades et al. (1996), itself a variant of a method presented by Gentzsch (1979), which is in turn essentially a pared-down Runge-Kutta-Chebyshev (RKC) method (van der Houwen (1977); van der Houwen et al. (1980); Verwer et al. (1990); Verwer (1996); Sommeijer et al. (1997)).

Despite the fact that the STS method is approximately 30 years old, it has been reported in use by relatively few researchers. The instances in the literature of STS being used which we are aware of are in engineering and physical disciplines. These include: nonlinear degenerate convection-diffusion Evje et al. (2001); electromagnetic wave scattering Shi et al. (2006); isotropic and anisotropic diffusion on biological membranes Sbalzarini et al. (2006); and magnetic field diffusion in astrophysics O'Sullivan & Downes (2007); Mignone et al. (2007). Recently, STS has been applied in finance and rigorously tested for the Black-Scholes pricing model by O'Sullivan & O'Sullivan (2009) where it was shown to be comparable, and in many cases superior, to the CN-SOR/CN-PSOR scheme in terms of accuracy

and computional speed for European/American put options.

To proceed we consider the PDE

$$\frac{\partial \mathbf{u}}{\partial \tau} + \mathbf{A}\mathbf{u} = 0; \quad \mathbf{u}(\mathbf{0}) = \mathbf{u}^{\mathbf{0}}, \tag{30}$$

and we assume a linear explicit scheme on $\mathbf{u} \in \mathbb{R}^{(m+1)(n+1)}$ of the form

$$\mathbf{u}^{k+1} = (\mathbf{I} - \Delta\tau_{\text{sts}}\mathbf{A})\mathbf{u}^k \tag{31}$$

where the solutions at time levels k and $k+1$ are known and unknown respectively, $\mathbf{I}$ is the identity matrix and $\mathbf{A} \in \mathbb{R}^{(m+1)(n+1)} \times \mathbb{R}^{(m+1)(n+1)}$ is a symmetric positive definite matrix.

Eq. 31 corresponds to the $\theta = 0$ instance of the θ-method corresponding to the explicit Euler scheme as described in Section 3.3. For stability the CFL contraint requires that

$$|1 - \Delta\tau\lambda| < 1 \tag{32}$$

for all eigenvectors λ of $\mathbf{A}$. The maximal value of $\Delta\tau$ may therefore be defined by

$$\Delta\tau_{\text{expl}} \equiv \frac{2}{\lambda_{\max}} \tag{33}$$

where $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{A}$.

The essence of STS is that rather than requiring stability at each step of the time integration, N_{sts} sub-steps of varying size $\Delta\tau_j$ ($j = 1$ to N_{sts}) are rolled together into a single super-step $\Delta\tau_{\text{sts}}$ according to

$$\Delta\tau_{\text{sts}} = \sum_{j=1}^{N_{\text{sts}}} \Delta\tau_j. \tag{34}$$

and stability is only demanded at the end of the super-step [2]. We write the compound scheme as

$$\mathbf{u}^{k+1} = \left[\prod_{j=1}^{N_{\text{sts}}} (\mathbf{I} - \Delta\tau_j \mathbf{A})\right] \mathbf{u}^k \tag{35}$$

[2] Verwer (1996) has claimed that factorized RKC methods are impractical as they suffer from internal instability. O'Sullivan & O'Sullivan (2009) find no evidence of any such instability influencing their solutions for cases with $N_{\text{sts}} \lesssim 30$.

and for stability we require

$$\left| \prod_{j=1}^{N_{\rm sts}} (1 - \Delta\tau_j \lambda) \right| < 1 \tag{36}$$

for all eigenvectors λ of $\mathbf{A}$.

The properties of Chebyshev polynomials of degree $N_{\rm sts}$ (Markoff, 1916) provide the means to explicitly enforce stability while maximizing $\Delta\tau_{\rm sts}$. The optimal values for the sub-steps obtained in this way are given by

$$\Delta\tau_j = \Delta\tau_{\rm expl} \left[(-1+\nu) \cos\left(\frac{2j-1}{N_{\rm sts}} \frac{\pi}{2} \right) + 1 + \nu \right]^{-1} \tag{37}$$

where ν is a user defined damping factor. The scheme is stable for $\nu > 0$ and unstable in the limit $\nu \to 0$ with the property

$$\Delta\tau_{\rm sts} \to N_{\rm sts}^2 \Delta\tau_{\rm expl} \qquad \text{as } \nu \to 0. \tag{38}$$

In practice the scheme may be marginally stable for low enough values of ν for a given choice of $N_{\rm sts}$. A balance between robustness and acceleration should therefore be struck by the user with appropriate choices of the two free parameters.

We illustrate the efficacy of the acceleration process for $N_{\rm sts} = 30$ in Figure 2 for various choices of ν. It can be seen that the first substep may be up to 25 times the stable limit for a standard explicit integration as $\nu \to 0$ but subsequent substeps become small. The effect of this is a cumulative error cancellation which recovers stability over the composite superstep. Crucially, there is a net payoff in terms of the size of the superstep with respect to $N_{\rm sts}$ steps of size $\Delta t_{\rm expl}$ according to equation 38.

STS applied directly as described above results in a scheme which is first order in time (Alexiades et al., 1996). From this point we shall refer to this scheme as the STS-1 scheme. It is not possible to introduce additional temporal structure to an STS step since intermediate values obtained during an STS cycle are physically meaningless and may not be used as approximations to the solution in any sense. Therefore, predictor-corrector style methods are not applicable should higher order convergence be required. On the other hand, we have found that Richardson extrapolation (RE) works well. By this method

all the advantages of the STS-1 method are easily transferred to second (or higher) order schemes.

The principal advantage of the STS method is not efficiency however, but simplicity. Explicit discretizations of even the most complex systems of parabolic equations are very straightforward within this discretization paradigm. In particular, implementation of adaptive mesh refinement (AMR) technologies and/or parallelization via domain decomposition techniques present no great challenges from within an explicit framework. On the contrary, when implicit methods are involved, tackling problems of even a moderate level of complexity can be an exceedingly intricate task, especially in parallel applications.

Note that although formal results only exist for linear schemes, there is ample evidence, as described above, that non-linear target systems may be equally amenable to the STS method. Formally, stability of the STS scheme is assured if $\mathbf{A}$ in equation (35) is symmetric positive definite, Alexiades et al. (1996). However, in the Black Scholes PDE and the more general Heston PDE the spatial operator $\mathbf{A}$ is weakly non-symmetric.

In the Black Scholes case a formal stability analysis for an alternative discretization of the non-symmetric Black Scholes spatial operator $\mathbf{A}$ is given in O'Sullivan & O'Sullivan (2009). The scheme presented therein is formally stable under application of STS to the symmetric component of the multiplicitavely split operator. The skew symmetric part of the operator is then integrated via a novel scheme developed by O'Sullivan & Downes (2006, 2007). While the split scheme presented by O'Sullivan & O'Sullivan (2009) is formally stable and may be of particular interest for systems with moderate to strong skew symmetric evolution operators, it was found by these authors that this alternative scheme was not strictly necessary for weakly non-symmetric operators. In agreement with this result, we find that splitting is unnecessary and that direct application of the STS scheme to the Heston PDE is appropriate even though the Heston spatial operator $\mathbf{A}$ is not fully symmetric.

Before comparing the performance of the STS method applied via equation (35) to the FD schemes described in Section 3.3 we include a brief section on the use of RE in the STS scheme.

4.1 Richardson Extrapolation

In this paper we employ two RE methods to render STS schemes second order accurate in time.

The first approach is to use RE in a step-wise fashion. We assume a smoothly convergent first order accurate method for the temporal integration of the semi-discrete equation (30) with exact solution $u_{\Delta x,\Delta y}(x, y, \tau)$ on a grid with spatial intervals Δx, Δy. Given a second order accurate solution at time level k such that $u^k_{i,j} = u_{\Delta x,\Delta y}(i\Delta x, j\Delta y, k\Delta\tau) + (L - k)\mathcal{O}(\Delta\tau^3)$ we may take a single step of size $\Delta\tau$ to approximate the solution at time level $k+1$ using $u^{k+1}_{ij}(\Delta\tau) = u_{\Delta x,\Delta y}(i\Delta x, j\Delta y, (k+1)\Delta\tau)+C\Delta\tau^2+\mathcal{O}(\Delta\tau^3)$ for some constant C determined by the leading truncation error term of the scheme. Similarly, taking two steps of size $\Delta\tau/2$, we have $u^{k+1}_{ij}(\Delta\tau/2) = u_{\Delta x,\Delta y}(i\Delta x, j\Delta y, (k+1)\Delta\tau) + (C/2)\Delta\tau^2 + \mathcal{O}(\Delta\tau^3)$. Subtracting the expression for $u^{k+1}_{ij}(\Delta\tau)$ from twice the expression for $u^{k+1}_{ij}(\Delta\tau/2)$ yields a second order advancement in the solution from time level k to level $k+1$ according to

$$u^{k+1}_{ij} = 2u^{k+1}_{ij}(\Delta\tau/2) - u^{k+1}_{ij}(\Delta\tau), \text{ for } k = 0, 1, \ldots, l-1. \tag{39}$$

We also consider the more usual post-processed form which requires two independently derived solutions for use in the extrapolation of the final solution $u^l_{ij} = 2u^l_{ij}(\Delta\tau/2) - u^l_{ij}(\Delta\tau)$, see for example Geske & Johnson (1984).

We refer to the former approach as local RE and the latter as global RE. The significant difference is that, for local RE, a second order solution is immediately available at every temporal mesh point. For global RE, this would require significant additional data storage and handling. However, we find that global RE is less prone to numerical oscillations for very small values of the damping parameter ν and is therefore preferable in general for use with STS.

We note that RE is computationally more expensive than some other higher order reconstructions. It requires 50% more computational effort than the predictor-corrector approach on a per-step basis. However, it is simple to implement as it merely requires a reapplication of the first order scheme. When used with STS it is of greater applicability than CN-PSOR as shall be demonstrated.

5 Numerical Experiments

In this section we analyse the efficiency of the STS accelerated explicit scheme relative to the unaccelerated explicit scheme and a CN scheme applied to the pricing of European and American options under Heston's SV model.

The default parameters of the problem are chosen to be

$$E = 10, T = 0.25, r = 0.1, \alpha = 5, \beta = 0.16, \gamma = 0.9,\ \lambda = 0,\ \text{and}\ \rho = 0.1,$$

in order to permit direct comparison with the results of Zvan et al. (1998); Clarke & Parrot (1999); Oosterlee (2003); Ikonen & Toivanen (2007b)[3]. The computational domain's extent is defined by setting $x_{\min} = 0, x_{\max} = 20$ and $y_{\min} = 0, y_{\max} = 1$. These values are again chosen for consistency with the referenced works. We note, however, that $x_{\max} = 20$ is close enough to the exercise price $E = 10$ for the influence of the boundary to be apparent on the solution. This point is discussed further in the subsequent error analysis of this section. Grid generating functions, as described in section 3.2, are used to prescribe the interior grid points. The convention we use for a given test case for denoting the number of stock price steps, variance steps and time steps respectively is $\{m, n, l\}$.

Firstly, by way of an illustration the behaviour of STS driven acceleration as $\nu \to 0$ over the time-to-maturity of the option T, we fix $N_{sts} = 30$ and consider the performance of STS-1 for various small values of ν. Panel (a) of Figure 3 shows the number of supersteps, l, required to complete the integration over time T versus the damping parameter ν on a grid where $(m, n) = (512, 256)$, the finest grid test case considered in the experiments below. The convergence of l to 77 as $\nu \to 0$ is clearly evident in panel (b) (where the abcissa are logarithmic rather than linear in ν). From Eq. 38, and using the minimum number of explicit steps required for stability on this grid (estimated to be $l_{expl} = 68,878$ from Eq. 29), we expect a limiting value of $l \approx 77$. This is clearly in good agreement with our experimentally derived value and repesents a speed-up by a factor of approximately 30 relative to the explicit method.

We note that accelerated schemes are not stable for vanishing ν and some small but finite value of ν is required. By way of illustration of this detail, Figure 3 includes a reference line at $l = 81$ corresponding to the acceleration obtained for $\nu = 4 \times 10^{-5}$. This is the

[3]Other parameter settings were chosen including cases in which the Feller condition was not satisfied. However results were broadly similar across all parameter settings examined.

experimentally derived stability limit for the STS-1 scheme below which prices are found to exhibit spurious oscillations. At this value of ν a speed-up factor of approximately 28 is achieved.

Stable solutions are guaranteed on all the grid resolutions in the experiments considered below for $\nu = 0.0006$. Moreover, as the grid resolution is increased the damping parameter can be pushed closer to zero whilst still maintaining stable solutions.

In the following sections we shall examine the behaviour of the schemes as both the spatial and temporal resolutions are increased simulataneously. For EXPL-1, in order to achieve a stable solution the number of timesteps is increased by 4 for every doubling of the spatial resolution. In all other cases the resolution in time is scaled linearly with the spatial resolution.

5.1 European Options

We now present a series of test results for the pricing of European put options under Heston's SV model.

Table 1 displays European put option prices at five stock prices $x = 8, \ldots, 12$ for an initial variance $y = 0.0625$ at a number of different resolutions. Five numerical schemes are tabulated: the STS-1 method; the STS method with local RE (STS-RE-L); the STS method with global RE (STS-RE-G); the standard explicit approach to first order accuracy in time (EXPL-1); and the CN-SOR scheme. We will refer to the STS-RE-L and STS-RE-G schemes collectively as the STS-2 schemes. Two reference methods derived via the semi-analytical results obtained using the FFT approach of Carr & Madan (1999)[4] and a high resolution CN-SOR method are included for reference.

The CN-SOR European put prices are computed with a convergence tolerence measure $tol = 1 \times 10^{-4}$ defined similarly to tol in C ($tol = 1 \times 10^{-5}$ on the high resolution grid). These values are approximately optimal in the sense that we find decreasing the tolerance increases the run time for the scheme without improving the accuracy. The overrelaxation parameter is varied with the grid resolution to ensure optimal convergence rates for the scheme. All European STS prices are computed with a fixed number of substeps $N_{sts} = 30$ and a fixed damping parameter of $\nu = 0.0006$. The number of supersteps, l, is scaled in

[4]The accuracy of the FFT prices were ensured by varying the FFT inputs until the relative price changes were of the order of 1×10^{-10}.

proportion to the number of points on the spatial grid in each dimension.

It is clear from Table 1 that all five FD methods result in European put prices that converge as expected to the reference prices as the grid and temporal resolutions increase. To evaluate the rate of this convergence we examine the errors in the solutions obtained from the FD schemes relative to reference prices.

Figure 4 illustrates the errors in the solutions obtained from the FD schemes as a function of the stock price at an initial variance $y = 0.0625$ on a range of resolutions. As stated previously, the boundary at $x_{\max} = 20$ is close enough to the exercise price $E = 10$, that the influence of the boundary conditions are evident in the solution. While we have carried out tests with $x_{\max}$ moved to larger values and confirmed that boundary induced errors may be reduced, as previously remarked, we use $x_{\max} = 20$ for consistency with the referenced tests from the literature. For the error analysis we use a high resolution reference solution obtained via CN-SOR with $(m, n, l) = (2048, 1024, 2050)$. Note that the semi-analytical FFT results are inappropriate as the reference solution for this analysis since they are not subject to the boundary conditions prescribed for the FD numerical schemes. We note however that we have confirmed the high resolution CN-SOR solution converges to the semi-analytical FFT solution at a rate of second order accuracy at ten sample interior points $(x,\, y) = ([8, 9, \ldots, 12] \times [0.0625, 0.25])$.

The error in the prices is in good agreement in all cases. STS-1 demonstrates larger errors than the others due to its lower (first) order of accuracy in time. Despite also being first order in time, the error in the EXPL-1 solution is close to those of the second order integrations. This is due to the substantially larger number of timesteps required to maintain a stable solution which results in a negligible first order temporal error with respect to the second order spatial error. The total error therefore takes on the second order charasteristics of the spatial error. We also note that the plots are qualitively similar for different variance values y, however, the errors do increase in magnitude across all schemes when y is near the boundary values of $y_0 = 0$ and $y_n = 1$.

Table 2 displays the l_2 norm errors calculated using ten European put prices at five stock prices about the exercise price E, $x = 8, 9, \ldots, 12$, and two different initial variance values $y = 0.0625$ and $y = 0.25$. As before, since it is a first order accurate method in time, the STS-1 method displays the largest errors in the l_2 norm. We also found that the explicit method displays the smallest l_2 norm error values since, as previously remarked, very high temporal resolution is necessary to maintain stability. We may also observe that

the STS-2 schemes provide l_2 norm errors that are similar to standard explicit case. We may deduce therefore that the temporal error in the STS-2 schemes is also small despite the more efficient time integration. Finally, we note that, while not dramatically different, the CN-SOR scheme has the highest l_2 norm error of the second order schemes.

Table 2 also provides error ratios between test cases with successively increased resolution. The error ratios for STS-1 are approximately 2 which is a consequence of scheme with dominant first order temporal error. In the case of EXPL-1 the error ratios scale as roughly 4 on doubling the spatial resolution. As remarked earlier, the temporal resolution is necessarily very high for reasons of stability resulting in a negligible temporal error and dominant second order spatial error. This is compatible with our assumption of a scheme which is second order in space and first order in time. The remaining schemes yield ratios of approximately 4 suggesting schemes which are second order both in space and time. We remark that the convergence rates of the STS-2 methods are almost identical to the CN-SOR and the standard explicit method.

Lastly, we note that table 2 gives CPU times for each test. These indicate that on the finest grid the STS-1 method is least computationally expensive, followed by STS-RE-L, STS-RE-G, CN-SOR and finally EXPL-1 in that order. Notably, the STS-2 methods are approximately twice as efficient as the CN-SOR scheme. This is despite the advantage of the overrelaxation parameter in CN-SOR being chosen experimentally to ensure the fastest possible convergence of the algorithm in each case.

5.2 American Options

When pricing American options with the STS schemes, the early exercise constraint is applied at the end of each superstep since the solution conditions inside the superstep are not meaningful as option prices. Consequently, an increase (decrease) in the number of supersteps results in an increase (decrease) in the frequency at which the early exercise condition is enforced. This generates a trade-off in the speed of the algorithm versus the resolution of the early exercise boundary. A similar trade-off in speed versus accuracy typically arises for implicit methods. For PSOR, for example, projection (replacement of the continuation value of the American option with its early exercise value when optimal) must be carried out within each iteration (see C). A direct consequence is that achieving convergence becomes more laborious with increasing timestep size.

Table 3 displays reference American put option prices at five stock prices $x = 8, \ldots, 12$ for an initial variance $y = 0.0625$. High accuracy reference prices were calculated in two independent ways; firstly using a FD scheme from the QUANTLIB library with a grid of size $(m, n, l) = (4096, 2048, 4098)$; and secondly using a CN-PSOR scheme on a grid of size $(m, n, l) = (2048, 1024, 2050)$. We note that little difference between the prices obtained via QUANTLIB and CN-PSOR is observed.

Table 3 also displays the FD American put prices derived using the STS-1, STS-RE-L, STS-RE-G, EXPL-1, and CN-PSOR schemes using various grid resolutions. All STS American put prices are computed with a fixed number of substeps $N_{sts} = 15$ and a fixed damping parameter of $\nu = 0.002$. The number of supersteps, l, scales linearly with the spatial resolution in all cases except EXPL-1 where l is scaled quadratically in order to maintain stability. We choose a smaller value of N_{sts} relative to the European case in order to reduce errors associated with numerically capturing the early exercise free boundary. The CN-PSOR prices are computed with $tol = 1 \times 10^{-4}$ ($tol = 1 \times 10^{-5}$ is used for the reference CN-PSOR prices) whilst the overrelaxation parameter is varied with the grid step size to ensure optimal convergence.

Figure 5 illustrates the errors in the American solutions obtained from the FD schemes as a function of the stock price at an initial variance $y = 0.0625$ on a range of grid resolutions. For error analysis we used a high resolution solution obtained using CN-PSOR with $(m, n, l) = (2048, 1024, 2050)$. We note that the error observed in the prices is small in all tests. As in the European case, STS-1 demonstrates errors of a greater magnitude due to its lower (first) order of accuracy in time. Again, similarly to the European case, the error in the EXPL-1 solution is close to those of the second order integrations because of the large number of timesteps required for stability resulting in a negligible first order temporal error. The American option error plots of Figure 5 also illustrate the influence of the free boundary which can be most clearly seen as a downward spike in the error for EXPL-1 for a stock price of about 8.

Table 4 displays the l_2 norm errors calculated using ten American put prices at five stock prices about the exercise price E, $x = 8, 9, \ldots, 12$, and two different initial variance values $y = 0.0625$ and $y = 0.25$, where the CN-PSOR prices computed with 2050 time steps on a grid of size $(m, n) = (2048, 1024)$ are the high accuracy reference American put prices.

We remark that the error ratios in Table 4 are approximately 2 for STS-1 indicative of a scheme with first order accuracy in time. The remaining schemes show ratios of roughly 4

on successive doubling of spatial resolution suggesting second order accuracy in the leading error terms. In all cases except EXPL-1 this is the expected behaviour for schemes of second order accuracy in both space and time when the spatial and temporal errors are comparable in magnitude. EXPL-1 has negligible temporal error and second order accuracy may only be inferred in the spatial terms.

In terms of relative error magnitudes, the STS-1 method may be seen to have the largest error followed by CN-PSOR. Furthermore, it is clear that the STS-2 methods have only slightly larger errors relative to EXPL-1 despite the latter's far greater, albeit first order accurate, temporal resolution.

With regard to the CPU times also provided by table 4, we find that, similarly to the European case, STS-2 schemes are faster than the CN-PSOR scheme and display lower errors. Additionally, the STS-1 scheme is observed to be approximately three to four times faster than the CN-PSOR scheme.

As a final exercise in validation, table 5 displays American put option prices at the five stock prices $x = 8, 9, \ldots, 12$ calculated with variances of $y = 0.0625$ and 0.25. Along with the two reference prices from table 4, three sets of STS-RE-G prices are displayed for a single spatial grid of size $(m, n) = (2048, 1024)$ but different STS parameter settings. We first set $N_{sts} = 30, \nu = 0.0006$ and $l = 2050$, so that results are comparable to the CN-PSOR results. We also examine two additional choices of parameter sets with $N_{sts} = 25, \nu = 0.0006, l = 2750$ and $N_{sts} = 35, \nu = 0.0006, l = 1800$. These results are compared to the American put option prices published in the literature using the same sets of SV parameters. We find that the American option prices from all three STS parameter settings agree very closely with the reference prices and those prices in the literature. In particular, agreement with prices quoted by Ikonen & Toivanen (2007b) is strong.

6 Conclusion

An acceleration technique, known as Super-Time-Stepping (STS), for explicit finite difference (FD) algorithms is introduced for the first time in the two-factor setting of stochastic volatility. We demonstrate the efficacy of the method by pricing European and American put options in a series of bench-tests with well-known FD techniques.

We demonstrate degrees of acceleration provided by the STS method which yield comparable, and even superior, efficiencies to implicit differencing methods. Of central impor-

tance, this is achieved with no significant increase in implementation complexity over and above that of the underlying standard explicit algorithm.

Given that STS accelerated methods inherit the simplicity of explicit methods whilst achieving high accuracy at low computational cost, we conclude that when faced with complex numerical pricing problems, the approach presented here offers a compelling alternative to conventional implicit techniques. Models involving multi-dimensional parameter spaces, non-uniform meshes, moving boundaries, or meshes which are distributed in parallel over several processors will be particularly amenable to STS accelerated explicit methods.

A Boundary Discretization

The following discretizations are used at the boundaries. At x_0 we use a Dirichlet boundary condition with $u(x_0, y_j, \tau_k) = \exp(-rk\Delta t)\max[E - x_0, 0]$ for European options and $u(x_0, y_j, \tau_k) = \max[E - x_0, 0]$ for American options. At $x_{\max}$ we use a Neumann condition given by

$$\begin{aligned}\frac{\partial u_{m,j}}{\partial x} &\approx a_{ll}^C u_{m-2,j} + a_l^C u_{m-1,j} + a^C u_{m,j} = 0,\\ \Rightarrow u_{m,j} &= -\frac{1}{a^C}\left(a_{ll}^C u_{m-2,j} + a_l^C u_{m-1,j}\right),\end{aligned}$$

where

$$\begin{aligned}a_{ll}^C &= \frac{h_r(x_{m-1})}{h_l(x_{m-1})\left(h_l(x_{m-1}) + h_r(x_{m-1})\right)},\\ a_l^C &= -\frac{h_l(x_{m-1}) + h_r(x_{m-1})}{h_l(x_{m-1})\, h_r(x_{m-1})},\\ a^C &= \frac{h_l(x_{m-1}) + 2h_r(x_{m-1})}{h_r(x_{m-1})\left(h_l(x_{m-1}) + h_r(x_{m-1})\right)}.\end{aligned}$$

At $y_{\max}$ a Neumann condition is also used

$$\begin{aligned}\frac{\partial u_{i,n}}{\partial y} &\approx b_{dd}^C u_{i,n-2} + b_d^C u_{i,n-1} + b^C u_{i,n} = 0,\\ \Rightarrow u_{i,n} &= -\frac{1}{b^C}\left(b_{dd}^C u_{i,n-2} + b_d^C u_{i,n-1}\right),\end{aligned}$$

where

$$\begin{aligned}b_{dd}^C &= \frac{h_u(y_{n-1})}{h_d(y_{n-1})\left(h_d(y_{n-1}) + h_u(y_{n-1})\right)},\\ b_d^C &= -\frac{h_d(y_{n-1}) + h_u(y_{n-1})}{h_d(y_{n-1})\, h_u(y_{n-1})},\\ b^C &= \frac{h_d(y_{n-1}) + 2h_u(y_{n-1})}{h_u(y_{n-1})\left(h_d(y_{n-1}) + h_u(y_{n-1})\right)}.\end{aligned}$$

At y_0 the value for $u_{i,0}$ satisfies a reduced PDE and is solved explicitly or implicitly in the same way as the solution to $u_{i,j}$ at the inner points on the grid. The PDE at y_0 is given by

$$\begin{aligned}&\frac{\partial u_{i,0}}{\partial \tau} - rx_i\frac{\partial u_{i,0}}{\partial x} - \alpha\beta\frac{\partial u_{i,0}}{\partial y} + ru_{i,0} = 0,\\ \Rightarrow &\frac{\partial u_{i,0}}{\partial \tau} - rx_i\left(\frac{u_{i+1,0} - u_{i,0}}{h_r(x_i)}\right) - \alpha\beta\left(\frac{u_{i,1} - u_{i,0}}{h_u(y_0)}\right) + ru_{i,0} = 0.\end{aligned}$$

B Upwinded Differencing

Recall from section 3.1 that the Heston PDE at a reference point $u = u_{ij}$ can be written as follows:

$$\mathcal{L}u = \frac{\partial u}{\partial \tau} + \mathcal{A}u = 0,$$

in which $\mathcal{A}$ is a nine component operator matrix given by

$$\mathcal{A} = \begin{bmatrix} A_{lu} & A_u & A_{ru} \\ A_l & A_c & A_r \\ A_{ld} & A_d & A_{rd} \end{bmatrix},$$

with

$$\begin{aligned}
A_c &= -\left(\frac{1}{2}yx^2a^D + \rho\gamma xyc^D + \frac{1}{2}\gamma^2 yb^D + rxa^C + \alpha\left(\beta - y\right)b^C - r\right), \\
A_l &= -\left(\frac{1}{2}yx^2a_l^D + \rho\gamma xyc_l^D + rxa_l^C\right), \\
A_r &= -\left(\frac{1}{2}yx^2a_r^D + \rho\gamma xyc_r^D + rxa_r^C\right), \\
A_d &= -\left(frac12\gamma^2 yb_d^D + \rho\gamma xyc_d^D + \alpha\left(\beta - y\right)b_d^C\right), \\
A_u &= -\left(\frac{1}{2}\gamma^2 yb_u^D + \rho\gamma xyc_u^D + \alpha\left(\beta - y\right)b_u^C\right), \\
A_{ld} &= -\rho\gamma xyc_{ld}^D, \quad A_{lu} = -\rho\gamma xyc_{lu}^D \\
A_{rd} &= -\rho\gamma xyc_{rd}^D, \quad A_{ru} = -\rho\gamma xyc_{ru}^D,
\end{aligned}$$

where the superscript D denotes diffusion terms and the superscript C denotes convection terms defined in section 3.1. To adjust for upwinding we first define forward and backward convection terms as follows:

$$\begin{aligned}
a_l^{CF} = 0, a_r^{CF} = \frac{1}{h_r}, a_l^{CB} = -\frac{1}{h_l}, a_r^{CB} = 0, \\
b_d^{CF} = 0, b_u^{CF} = \frac{1}{h_u}, b_d^{CB} = -\frac{1}{h_d}, a_r^{CB} = 0.
\end{aligned}$$

At each point on the grid we adjust the nine component operator matrix $\mathcal{A}$ to implement upwinding. For exampe A_c is adjusted as follows:

$$\begin{aligned} A_c^{uw} = -\Big(&\frac{1}{2}yx^2a^D + \rho\gamma xyc^D + \frac{1}{2}\gamma^2 yb^D \\ &+ rx\left[a^C\mathbf{1}_{\{A_l<0,A_r<0\}} + a_r^{CF}\mathbf{1}_{\{A_l>0\}} + a_l^{CB}\mathbf{1}_{\{A_r>0\}}\right] \\ &+ \alpha\left(\beta - y\right)\left[b^C\mathbf{1}_{\{A_u<0,A_d<0\}} + b_u^{CF}\mathbf{1}_{\{A_d>0\}} + b_d^{CB}\mathbf{1}_{\{A_u>0\}}\right] - r\Big). \end{aligned}$$

Similarly we adjust the other convection components of the operator matrix $\mathcal{A}$ as follows:

$$\begin{aligned} a_l^C &\to a_l^C\mathbf{1}_{\{A_l<0,A_r<0\}} + a_l^{CF}\mathbf{1}_{\{A_l>0\}} + a_l^{CB}\mathbf{1}_{\{A_r>0\}}, \\ a_r^C &\to a_r^C\mathbf{1}_{\{A_l<0,A_r<0\}} + a_r^{CF}\mathbf{1}_{\{A_l>0\}} + a_r^{CB}\mathbf{1}_{\{A_r>0\}}, \\ b_d^C &\to b_l^C\mathbf{1}_{\{A_d<0,A_u<0\}} + b_d^{CF}\mathbf{1}_{\{A_d>0\}} + b_d^{CB}\mathbf{1}_{\{A_u>0\}}, \\ b_u^C &\to b_u^C\mathbf{1}_{\{A_d<0,A_u<0\}} + b_u^{CF}\mathbf{1}_{\{A_d>0\}} + b_u^{CB}\mathbf{1}_{\{A_u>0\}}. \end{aligned}$$

C Projected Successive OverRelaxation

PSOR is an important method for solving LCPs and is widely used to price American options, see for example Tavella & Randall (2000); Wilmott et al. (1993). For completeness we include a short description of the PSOR algorithm in this section of the appendix.

To solve the LCP in Eq 28 assume the solution $u_{ij}^{(k)}$ is known at time k and we need to determine the solution $u_{ij}^{(k+1)}$. For notional clarity denote $u_{ij} = u_{ij}^{(k)}$ and $v_{ij} = u_{ij}^{(k+1)}$. PSOR approximates the value v_{ij} to within a specified tolerance with the iterated solution $\tilde{v}_{ij}^{s+1}$. The algorithm proceeds as follows:

1. Initialize unknown value with solution at previous time step i. e. $\tilde{v}_{ij}^{(0)} = u_{ij}$.

2. Perform the following sequence of iteration steps over s:

$$\begin{aligned} cv_{i,j}^{(s+1)} &= (1-\omega)\,u_{ij} + \frac{\omega}{B_{i,i}}\left(C_{i,j}u_{ij} - \sum_{p=0}^{i-1}\sum_{q=0}^{j-1}B_{p,q}\tilde{v}_{p,q}^{(s+1)} - \sum_{p=i+1}^{M}\sum_{q=j+1}^{N}B_{p,q}\tilde{v}_{p,q}^{(s)}\right), \\ \tilde{v}_{i,j}^{(s+1)} &= \max\left[cv_{i,j}^{(s+1)}, g_{i,j}\right], \text{ for } i = 0,\ldots,m,\ j = 0,\ldots,n, \end{aligned}$$

   where $cv_{i,j}^{(s)}$ denotes the approximated continuation value of the option at iteration s and g_{ij} denoted the immediate exercise value of the option.

3. Do until $\left\| \tilde{v}_{i,j}^{(s+1)} - \tilde{v}_{i,j}^{(s)} \right\| \leq tol$ or until $s = I_{\max}$, where tol is the tolerance level and $I_{\max}$ is the maximum number of iterations.

4. Set $u_{ij}^{(k+1)} = \tilde{v}_{i,j}^{(s+1)}$.

References

Alexiades, V., Amiez, G., and P. Gremaud, 1996, "Super-Time-Stepping acceleration of explicit schemes for parabolic problems", *Com. Num. Meth. Eng.*, 12, 31-42.

Bates, D., 1973. "Jumps and Stochastic Volatility: Exchange Rate Processes Implicit in Deutshe Mark Options", *Review of Financial Studies*, 9, 69-107.

Black, F. and M.S. Scholes, 1973. "The pricing of options and corporate liabilities", *Journal of Political Economy*, 81, 637-654.

Brennan, M., and E. Schwartz, 1977. "The valuation of American put options", *Journal of Finance*, 32, 449-462.

Carr, P. and D. Madan, 1999. "Option valuation using the fast Fourier transform", *Journal of Computational Finance* 2, 61-73.

Clarke, N., and K. Parrot, 1999. "Multigrid for American option pricing with stochastic volatility", *Applied Mathematical Finance*, 177-195.

Cont, R. and P. Tankov, 2004. "Financial Modelling With Jump Processes", *Chapman & Hall/CRC Financial Mathematics Series*.

Crank, J., 1984. "Free and moving boundary problems", *Oxford: Clarendon Press*.

C.W. Cryer, 1971. "The solution of a quadratic programming problem using systematic overrelaxation", *SIAM Journal on Control*, 9, 385-392.

Duffie, D., J. Pan, and K. Singleton (2000). Transform analysis and asset pricing for affine jump-diffusions. Econometrica 68, 1343-76.

Evje, S., Karlsen, K. H., Risebro, N. H., 2001. "A continuous dependence result for nonlinear degenerate parabolic equations with spatially dependent flux function", in *Int. Series of Numerical Mathematics*, Birkhauser-Verlag, 140, 337-346.

Gentzsch, W., 1979. "Numerical solution of linear and non-linear parabolic differential equations by a time-discretisation of third order accuracy", in Hirschel, E. H. (ed.), *Proceeding of the Third GAMM-Conference on Numerical Methods in Fluid Mechanics*, Friedr. Vieweg & Sohn, pp. 109-117.

Geske, R., and H.E. Johnson, 1984. "The American Put Option Valued Analytically." *The Journal of Finance*, 39, 1511-1524.

Heston, S.L., 1993. "A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options", *The Review of Financial Studies*, Vol. 6, No. 2, pp. 327-343.

Kluge, T., 2002. "Pricing derivatives in stochastic volatility models using the finite difference method", *Dimploma Thesis*, Chemnitz University of Technology.

Kurpiel, A. and T. Roncalli, 2000. "Hopscotch Methods for Two State Financial Models", *Journal of Computational Finance*, Vol. 2/3, 2000.

Ikonen, S. and J. Toivanen, 2007a. "Componentwise splitting methods for pricing American options under stochastic volatility", *International Journal of Theoretical and Applied Finance*, Vol. 10, No. 2, pp. 331-361.

Ikonen, S. and J. Toivanen, 2007b. "Efficient numerical methods for pricing American options under stochastic volatility", *Numerical Methods for Partial Differential Equations*, Vol. 24, Issue 1, 104-126.

Markoff, W., 1916. "Über Polynome, die in einem gegebenen Intervalle möglichst wenig von null abweichen", *Mathematische Annalen*, 77, 213-258, translated by J. Grossman from original Russian article published in 1892.

Merton, R.C., 1973. "The theory of rational option pricing", *Bell Journal of Economics and Management Science*, 4, 141-183.

Merton, R.C., 1976."Option Pricing when the Underlying Returns are Discontinuous". *Journal of Financial Economics* 3, 125-144.

Mignone, A., Bodo, G., Massaglia, S., Matsakos, T., Tesileanu, O., Zanni, C., Ferrari, A., 2007. "PLUTO: A Numerical Code for Computational Astrophysics", *ApJS*, 170, 228

Nielsen, B.,Skavhaug,O. and A.Tveito, 2002. "Penalty and front-fixing methods for the numerical solution of American option problems", *Journal of Computational Finance*, 5.

Oosterlee, C.W. 2003. "On multigrid for linear complementarity problems with applications to American-style options", *Electronic Transactions Numerical Analysis*, 15, 165-185.

O'Sullivan, S., Downes, T. P., 2006. "An explicit scheme for multifluid magnetohydrodynamics", *MNRAS*, 366, 1329

O'Sullivan, S., Downes, T. P., 2007. "A three-dimensional numerical method for modelling weakly ionized plasmas", *MNRAS*, 376, 1648

O'Sullivan, S., O'Sullivan, C., 2010. "On the acceleration of explicit finite difference methods for option pricing", *forthcoming Quantitative Finance*.

R. Rannacher, 1984. "Finite element solution of diffusion problems with irregular data", *Numerical Mathematics*, 43, 309-327.

Sbalzarini, I. F., Hayer, Helenius, A., Koumoutsakos, P., 2006. "Simulations of (an) isotropic diffusion on curved biological surfaces", *Biophys. J.*, 90, 84788485.

Shi, Y., Li, L., Liang, C. H., 2006. "Multidomain pseudospectral time-domain algorithm based on super-time-stepping method", *IEE Proceedings Microwaves, Antennas and Propagation*, 153, 55-60.

Sommeijer, B. P., Shampine, L. F., Verwer, J. G., 1997. "RKC: An explicit solver for parabolic PDEs", *Technical Report MAS-R9715, CWI Amsterdam*.

Tavella, D. and C. Randall, 2000. "Pricing financial instruments: the finite difference method" *John Wiley and Sons, New York*.

van der Houwen, P. J., 1977. "Construction of integration formulas for initial value problems", *North-Holland, Amsterdam-New York*.

van der Houwen, P. J., Sommeijer, B. P., 1980. "On the internal stability of explicit m-stage Runge–Kutta methods for large values of m", *Z. Angew. Math. Mech.*, 60, 479-485.

Verwer, J. G., Hundsdorfer, W. H., Sommeijer, B. P., 1990. "Convergence Properties of the Runge-Kutta-Chebyshev Method", *Numer. Math.*, 57, 157-178.

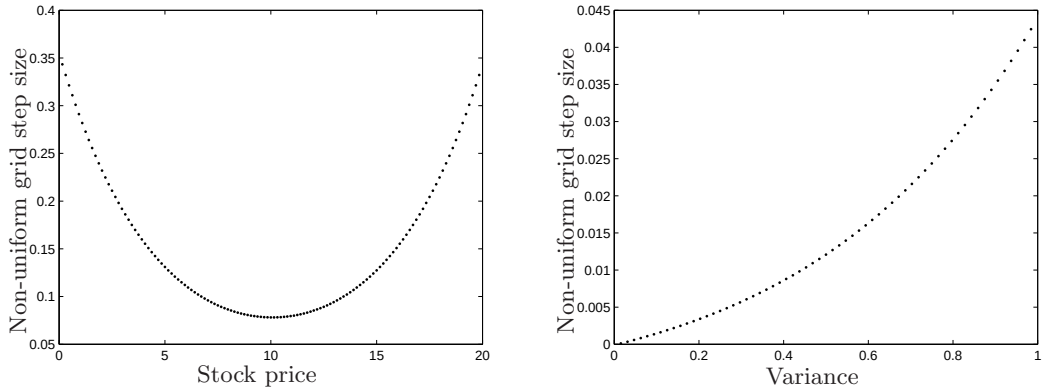
Verwer, J. G., 1996. "Explicit Runge-Kutta methods for parabolic partial differential equations", *Appl. Num. Math*, 22, 359-379.

Wilmott, P., Howison, S. and Dewynne, J., 1995, "Option pricing: mathematical models and computation", *Oxford Financial Press*.

Winkler, G., Apel, T. and U. Wystup, 2001. "Valuation of options in Heston's stochastic volatility model using finite element methods", *Working Paper, Chemnitz University of Technology*.

Zvan, R., Forsyth, P.A. and Vetzal K.R., 1998. "Penalty methods for pricing American options with stochastic volatility", *Journal of Computational and Applied Mathematics*, 91, 199-218.

Zvan, R., Forsyth, P.A. and Vetzal K.R., 2003. "Negative coefficients in two-factor option pricing models", *Journal of Computational Finance*, 7, 37-73.



(a) Non-uniform grid step size in the x-direction (b) Non-uniform grid step size in the y-direction

Figure 1: **Grid step size as a function the stock price and variance.**
These figures plot the stock price grid step size against the stock price and the variance grid step size against the variance. The plots emphasise how the grid step size reduces around areas of interest such as the exercise price or low variance values.

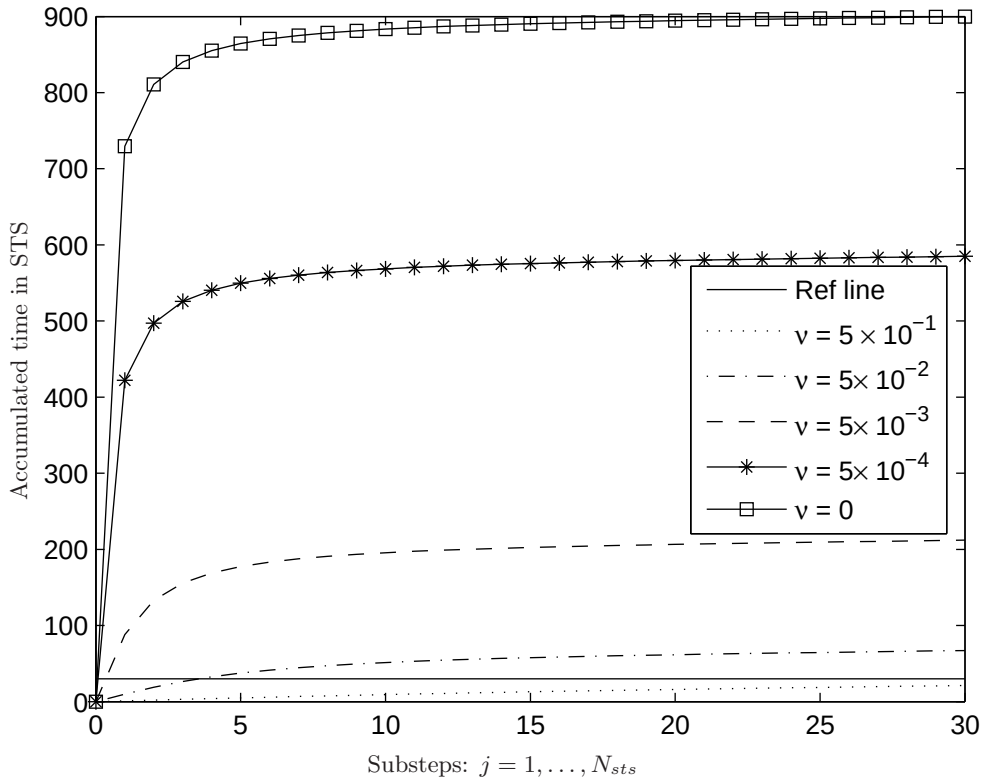
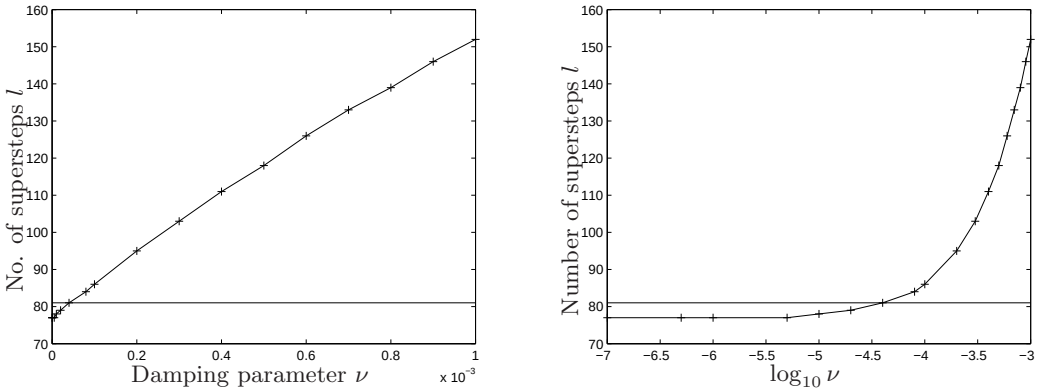


Figure 2: **Illustration of acceleration via STS.**

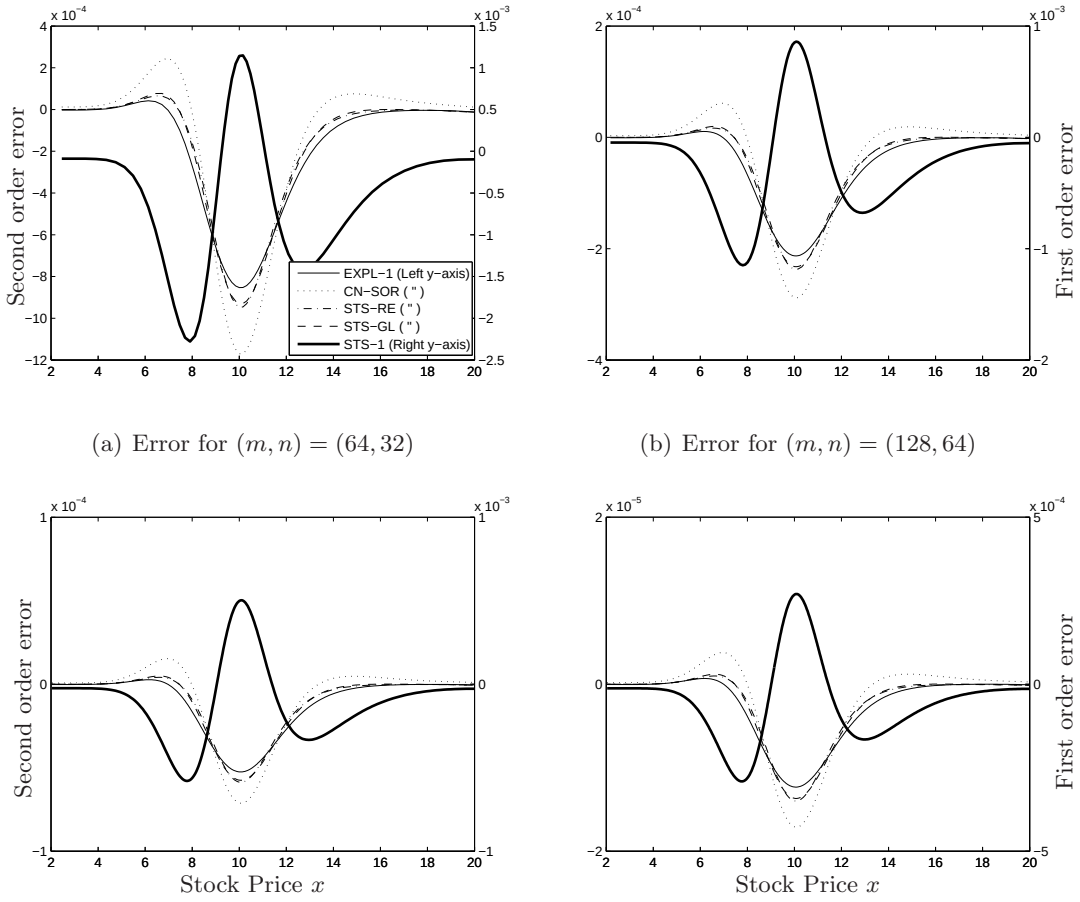
Figure 2 plots accumulated time, $\sum_{k=1}^{j} \Delta\tau_k$, in STS versus the substep number j over a single superstep $\Delta\tau_{\rm sts}$ with $N_{\rm sts} = 30$ for a range of damping factors ν. The accumulated time is plotted in units of the standard explicit timestep $\Delta\tau_{\rm expl}$. A reference line at $N_{\rm sts}\Delta\tau_{\rm expl}$ ($= 30$ in units of $\Delta\tau_{\rm expl}$) indicates the time attained over $N_{\rm sts}$ unaccelerated standard explicit steps. Note that acceleration approaches $N_{\rm sts}$ times this value as $\nu \to 0$, in agreement with equation (38). Note also that deceleration occurs for the highest considered damping factor of $\nu = 0.5$.



(a) Number of supersteps l versus ν

(b) Number of supersteps l versus $\log_{10}\nu$

Figure 3: **Number of supersteps as a function of ν and $\log_{10}\nu$.**
This figure depicts the number of supersteps as a function of ν and as a function of the logarithm of ν (log base 10) with N_{sts} fixed at 30 on a grid of size $(m, n) = (512, 256)$. In this example the minimum number of standard explicit steps required for stability is 68,878. The minimum number of supersteps possible is 77. The total number of steps taken in the STS scheme using minimal l is equal to $l \times N_{sts} = 77 \times 30 = 2310$. This is approximately a factor of 30 times less steps than that required for the standard explicit scheme. However, we do not expect the solutions to be stable for vanishing ν. In our numerical experiments on European options we choose $\nu = 0.0006$ which results in 126 supersteps. This repesents a speed-up by a factor of approximately 18 relative to the explicit scheme. The plot also includes a reference line at $l = 81, \nu = 4 \times 10^{-5}$ which is an experimentally approximated stability limit for the STS-1 scheme below which prices exhibit spurious oscillations. This stability limit repesents a speed-up by a factor of approximately 28 thereby achieving a very high proportion of the total acceleration available whilst maintaining stable prices and illustrating that the numerical experiments conducted in this paper were well within the limits of stability.



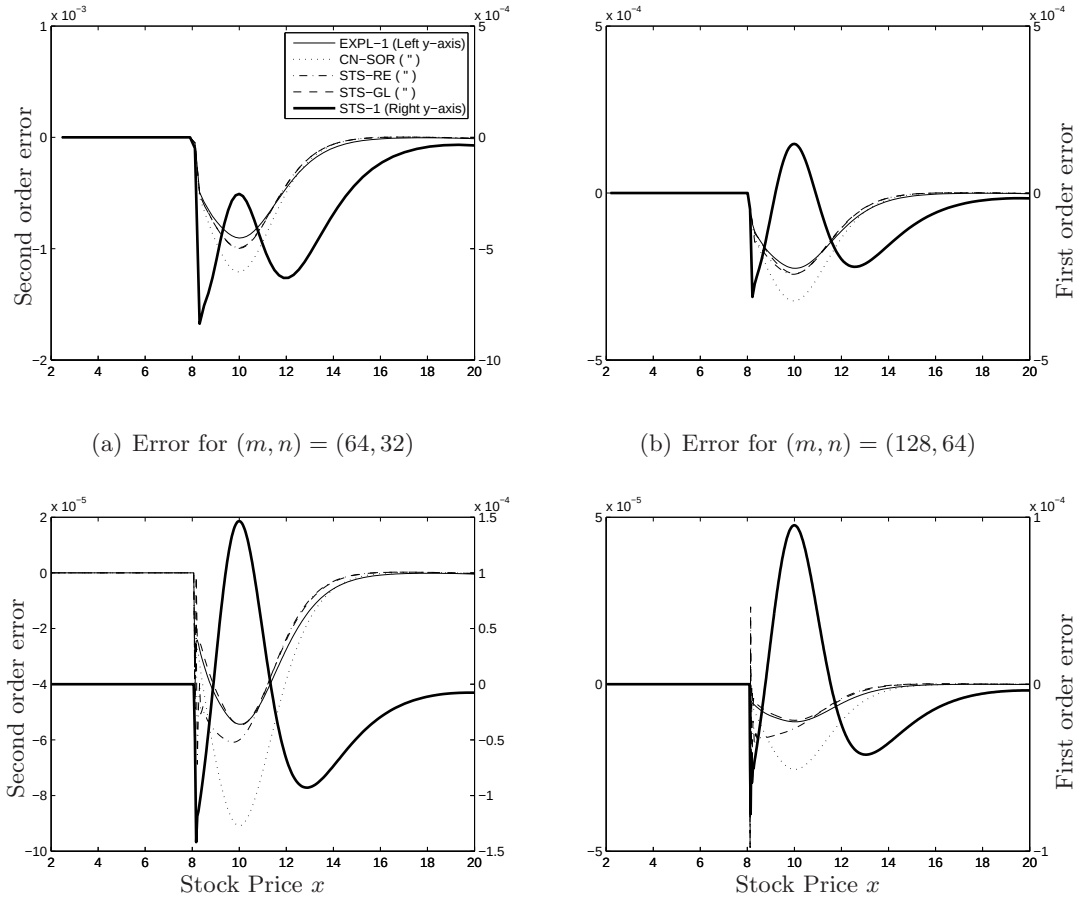
(a) Error for $(m, n) = (64, 32)$

(b) Error for $(m, n) = (128, 64)$

(c) Error for $(m, n) = (256, 128)$

(d) Error for $(m, n) = (512, 256)$

Figure 4: **Error in European put prices versus the stock price for $y = 0.0625$.** This figure depicts the error in European put option prices as a function of the stock price using various FD schemes. The error is given by $u_{fd} - u_{ref}$ where u_{fd} and u_{ref} denote the FD prices and the reference CN-SOR prices respectively. EXPL-1 is the explicit scheme, CN-SOR is the CN successive overrelaxation scheme, STS-RE-L is the STS scheme with local RE, STS-RE-G is the STS scheme with global RE and STS is the STS-1 scheme without any extrapolation. The left hand y-axis is to be used for following schemes: EXPL-1, CN-SOR, STS-RE-L and STS-RE-G, and the right hand y-axis is to be used for the STS-1 scheme.



(a) Error for $(m, n) = (64, 32)$

(b) Error for $(m, n) = (128, 64)$

(c) Error for $(m, n) = (256, 128)$

(d) Error for $(m, n) = (512, 256)$

Figure 5: **Error in American put prices versus the stock price for $y = 0.0625$.** This figure depicts the error in American put option prices as a function of the stock price using various FD schemes. The error is given by $u_{fd} - u_{ref}$ where u_{fd} and u_{ref} denote the FD prices and the reference CN-SOR prices respectively. EXPL-1 is the explicit scheme, CN-SOR is the CN successive overrelaxation scheme, STS-RE-L is the STS scheme with local RE, STS-RE-G is the STS scheme with global RE and STS is the STS-1 scheme without any extrapolation. The left hand y-axis is to be used for following schemes: EXPL-1, CN-SOR, STS-RE-L and STS-RE-G, and the right hand y-axis is to be used for the STS-1 scheme.

Table 1: European put prices evaluated using stock prices $x = 8, 9, \ldots, 12$ and an initial variance value of $y = 0.0625$ with STS parameters of $\nu = 0.0006$ and $N_{sts} = 30$.

		x				
Method	Grid (m, n, l)	8	9	10	11	12
Reference (FFT)		1.838868	1.048347	0.501465	0.208187	0.080428
Reference (CN-SOR)	$(2048, 1024, 2050)$	1.838868	1.048347	0.501465	0.208186	0.080428
STS-1	$(64, 32, 18)$	1.837355	1.048321	0.502606	0.208727	0.079657
	$(128, 64, 34)$	1.837830	1.048418	0.502317	0.208617	0.079983
	$(256, 128, 66)$	1.838343	1.048341	0.501963	0.208402	0.080225
	$(512, 256, 130)$	1.838598	1.048352	0.501733	0.208306	0.080328
STS-RE-L	$(64, 32, 18)$	1.839445	1.048331	0.500517	0.207842	0.080441
	$(128, 64, 34)$	1.838916	1.048414	0.501228	0.208148	0.080381
	$(256, 128, 66)$	1.838895	1.048337	0.501406	0.208159	0.080425
	$(512, 256, 130)$	1.838877	1.048349	0.501451	0.208183	0.080429
STS-RE-G	$(64, 32, 18)$	1.839424	1.048301	0.500535	0.207825	0.080415
	$(128, 64, 34)$	1.838910	1.048406	0.501232	0.208143	0.080374
	$(256, 128, 66)$	1.838894	1.048335	0.501407	0.208158	0.080423
	$(512, 256, 130)$	1.838877	1.048349	0.501451	0.208183	0.080428
EXPL-1	$(64, 32, 1064)$	1.839334	1.048302	0.500611	0.207860	0.080379
	$(128, 64, 4284)$	1.838887	1.048407	0.501252	0.208152	0.080364
	$(256, 128, 17192)$	1.838888	1.048335	0.501412	0.208161	0.080421
	$(512, 256, 68878)$	1.838875	1.048349	0.501452	0.208183	0.080428
	Grid $(m, n, l, iter_{av}, w)$					
CN-SOR	$(64, 32, 18, 22.3, 1.59)$	1.839586	1.048240	0.500295	0.207683	0.080432
	$(128, 64, 34, 28.6, 1.75)$	1.838951	1.048393	0.501177	0.208110	0.080378
	$(256, 128, 66, 35.5, 1.84)$	1.838904	1.048331	0.501393	0.208150	0.080424
	$(512, 256, 130, 50.4, 1.87)$	1.838879	1.048348	0.501448	0.208181	0.080429

Table 2: The (l^2-norm) errors calculated using ten European put prices at stock prices of $x = 8, 9, \ldots, 12$, and initial variance values of $y = 0.0625$ and 0.25. Also reported are the the ratio of consective errors, and the CPU times in seconds.

Method	Grid (m, n, l)	Error	Ratio	CPU
STS-1	$(64, 32, 18)$	0.005124		0.07
	$(128, 64, 34)$	0.003068	1.67	0.30
	$(256, 128, 66)$	0.001620	1.89	2.48
	$(512, 256, 130)$	0.000845	1.92	24.54
STS-RE-L	$(64, 32, 18)$	0.001657		0.22
	$(128, 64, 34)$	0.000365	4.54	0.90
	$(256, 128, 66)$	0.000101	3.60	7.45
	$(512, 256, 130)$	0.000023	4.37	69.80
STS-RE-G	$(64, 32, 18)$	0.001543		0.24
	$(128, 64, 34)$	0.000348	4.43	0.99
	$(256, 128, 66)$	0.000097	3.59	8.23
	$(512, 256, 130)$	0.000022	4.42	89.87
EXPL-1	$(64, 32, 1064)$	0.001341		0.14
	$(128, 64, 4284)$	0.000298	4.50	1.25
	$(256, 128, 17192)$	0.000078	3.79	22.16
	$(512, 256, 68878)$	0.000021	3.65	452.76
	Grid $(m, n, l, iter_{av}, w)$			
CN-SOR	$(64, 32, 18, 22.3, 1.59)$	0.002085		0.39
	$(128, 64, 34, 28.6, 1.75)$	0.000473	4.41	2.26
	$(256, 128, 66, 35.5, 1.84)$	0.000131	3.61	15.24
	$(512, 256, 130, 50.4, 1.87)$	0.000030	4.36	129.61

Table 3: American put prices evaluated using stock prices $x = 8, 9, \ldots, 12$ and an initial variance value of $y = 0.0625$ with STS parameters of $\nu = 0.002$ and $N_{sts} = 15$.

		x				
Method	Grid (m, n, l)	8	9	10	11	12
Reference (Quantlib)	$(4096, 2048, 4098)$	2.000000	1.107611	0.520024	0.213675	0.082043
Reference (CN-PSOR)	$(2048, 1024, 2050)$	2.000000	1.107620	0.520030	0.213676	0.082043
STS-1	$(64, 32, 66)$	2.000000	1.107834	0.519776	0.213609	0.081817
	$(128, 64, 130)$	2.000000	1.107836	0.520177	0.213805	0.081892
	$(256, 128, 258)$	2.000000	1.107677	0.520176	0.213742	0.081996
	$(512, 256, 514)$	2.000000	1.107662	0.520125	0.213725	0.082025
STS-RE-L	$(64, 32, 66)$	2.000000	1.107601	0.519038	0.213308	0.082022
	$(128, 64, 130)$	2.000000	1.107700	0.519787	0.213639	0.081988
	$(256, 128, 260)$	2.000000	1.107599	0.519970	0.213651	0.082040
	$(512, 256, 514)$	2.000000	1.107619	0.520016	0.213675	0.082045
STS-RE-G	$(64, 32, 66)$	2.000000	1.107600	0.519033	0.213300	0.082016
	$(128, 64, 130)$	2.000000	1.107712	0.519788	0.213636	0.081985
	$(256, 128, 258)$	2.000000	1.107614	0.519975	0.213651	0.082039
	$(512, 256, 514)$	2.000000	1.107626	0.520019	0.213675	0.082045
EXPL-1	$(64, 32, 1064)$	2.000000	1.107652	0.519126	0.213336	0.081992
	$(128, 64, 4284)$	2.000000	1.107719	0.519804	0.213640	0.081977
	$(256, 128, 17192)$	2.000000	1.107612	0.519975	0.213650	0.082036
	$(512, 256, 68878)$	2.000000	1.107625	0.520018	0.213674	0.082044
	Grid $(m, n, l, \text{iter-avg}, w)$					
CN-PSOR	$(64, 32, 66, 9.8, 1.60)$	2.000000	1.107465	0.518820	0.213154	0.081951
	$(128, 64, 130, 13.5, 1.75)$	2.000000	1.107656	0.519707	0.213577	0.081956
	$(256, 128, 258, 18.8, 1.84)$	2.000000	1.107589	0.519939	0.213624	0.082025
	$(512, 256, 514, 23.4, 1.87)$	2.000000	1.107616	0.520004	0.213664	0.082038

Table 4: The (l^2-norm) errors calculated using ten American put prices at stock prices of $x = 8, 9, \ldots, 12$, and initial variance values of $y = 0.0625$ and 0.25. Also reported are the the ratio of consective errors, and the CPU times in seconds.

Method	Grid (m, n, l)	Error	Ratio	CPU
STS-1	$(64, 32, 66)$	0.001091		0.18
	$(128, 64, 130)$	0.000784	1.39	0.58
	$(256, 128, 260)$	0.000445	1.76	4.90
	$(512, 256, 514)$	0.000262	1.70	38.53
STS-RE-L	$(64, 32, 66)$	0.001521		0.49
	$(128, 64, 130)$	0.000360	4.22	1.76
	$(256, 128, 258)$	0.000098	3.69	14.84
	$(512, 256, 514)$	0.000019	5.22	119.74
STS-RE-G	$(64, 32, 66)$	0.001531		0.44
	$(128, 64, 130)$	0.000358	4.27	1.74
	$(256, 128, 258)$	0.000086	4.17	14.62
	$(512, 256, 514)$	0.000018	4.86	118.02
EXPL-1	$(64, 32, 1064)$	0.001351		0.46
	$(128, 64, 4284)$	0.000329	4.10	1.70
	$(256, 128, 17192)$	0.000083	3.95	27.04
	$(512, 256, 68878)$	0.000017	4.81	452.48
	Grid $(m, n, l, \text{iter-avg}, w)$			
CN-PSOR	$(64, 32, 66, 9.8, 1.60)$	0.001885		0.96
	$(128, 64, 130, 13.5, 1.75)$	0.000501	3.76	3.44
	$(256, 128, 258, 18.8, 1.84)$	0.000160	3.14	28.00
	$(512, 256, 514, 23.4, 1.87)$	0.000042	3.81	226.95

Table 5: American put reference prices calculated using the STS-RE-G and CN-PSOR schemes at various stock prices, x, and for two initial variances: $y = 0.0625, 0.25$ on a grid of size (m, n, l). These price are compared to benchmark prices computed using code from quantlib, a freeware financial software resource. Other reference prices published in the literature are also included.

			x				
Reference		y	8	9	10	11	12
STS-RE-G	$m = 2048, n = 1024, l = 2050$	0.0625	2.000000	1.107622	0.520032	0.213678	0.082044
	$N_{sts} = 30, \nu = 0.0006$	0.25	2.078364	1.333633	0.795977	0.448273	0.242811
STS-RE-G	$m = 2048, n = 1024, l = 2750$	0.0625	2.000000	1.107622	0.520032	0.213678	0.082044
	$N_{sts} = 25, \nu = 0.0006$	0.25	2.078364	1.333633	0.795977	0.448273	0.242810
STS-RE-G	$m = 2048, n = 1024, l = 1800$	0.0625	2.000000	1.107622	0.520032	0.213678	0.082044
	$N_{sts} = 35, \nu = 0.0006$	0.25	2.078357	1.333623	0.795967	0.448264	0.242803
STS-RE-G	$m = 2048, n = 1024, l = 1600$	0.0625	2.000000	1.107622	0.520032	0.213678	0.082045
	$N_{sts} = 30, \nu = 0.0002$	0.25	2.078364	1.333633	0.795977	0.448274	0.242811
STS-RE-G	$m = 2048, n = 1024, l = 1400$	0.0625	2.000000	1.107622	0.520033	0.213678	0.082045
	$N_{sts} = 30, \nu = 0.00005$	0.25	2.078364	1.333633	0.795977	0.448274	0.242811
CN-PSOR	$m = 2048, n = 1024, l = 2050$	0.0625	2.000000	1.107620	0.520030	0.213676	0.082043
	$tol = 1 \times 10^{-5}, w = 1.85$	0.25	2.078363	1.333631	0.795974	0.448271	0.242809
Quantlib	$m = 4096, n = 2048, l = 4098$	0.0625	2.000000	1.107611	0.520024	0.213675	0.082043
		0.25	2.078355	1.333626	0.795972	0.448271	0.242803
Ikonen & Toivanen (2007b)		0.0625	2.000000	1.107629	0.520038	0.213681	0.082046
		0.25	2.078372	1.333640	0.795983	0.448277	0.242813
Clarke & Parrot (1999)		0.0625	2.0000	1.1080	0.5316	0.2261	0.0907
		0.25	2.0733	1.3290	0.7992	0.4536	0.2502
Zvan et al. (1998)		0.0625	2.0000	1.1076	0.5202	0.2138	0.0821
		0.25	2.0784	1.3337	0.7961	0.4483	0.2428
Oosterlee (2003)		0.0625	2.000	1.107	0.517	0.212	0.815
		0.25	2.079	1.334	0.796	0.449	0.243