

2013-5

Pricing European And American Options In The Heston Model With Accelerated Explicit Finite Differencing Methods

Conall O'Sullivan
University College Dublin

Stephen O'Sullivan
Technological University Dublin, stephen.osullivan@tudublin.ie

Follow this and additional works at: <https://arrow.tudublin.ie/scschmatart>



Part of the [Finance and Financial Management Commons](#), and the [Mathematics Commons](#)

Recommended Citation

O'Suillivan, C., O'Suillivan, S. : (2013) Pricing European and American Options under Heston's Stochastic Volatility Model with Accelerated Explicit Finite Differencing Methods, *International Journal of Theoretical and Applied Finance* 16 (03) doi/abs/10.1142/S0219024913500155

This Article is brought to you for free and open access by the School of Mathematics and Statistics at ARROW@TU Dublin. It has been accepted for inclusion in Articles by an authorized administrator of ARROW@TU Dublin. For more information, please contact arrow.admin@tudublin.ie, aisling.coyne@tudublin.ie, vera.kilshaw@tudublin.ie.

Pricing European and American Options in the Heston Model with Accelerated Explicit Finite Differencing Methods

CONALL O'SULLIVAN *

*Michael Smurfit Graduate School of Business, University College Dublin,
Blackrock, Co. Dublin, Ireland.
Conall.OSullivan@ucd.ie*

STEPHEN O'SULLIVAN

*School of Mathematical Sciences, Dublin Institute of Technology,
Kevin Street, Dublin 8, Ireland.
Stephen.OSullivan@dit.ie*

Electronic version of an article published as
International Journal of Theoretical and Applied Finance
Vol. 16, No. 3 (2013) 1350015 (35 pages)
©World Scientific Publishing Company
DOI: 10.1142/S0219024913500155

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 benchmark implicit schemes. We conclude that STS accelerated methods are a powerful numerical tool for the pricing of options that inherit the simplicity of explicit methods whilst achieving high accuracy at low computational cost and offer a compelling alternative to conventional implicit techniques.

Keywords American option pricing; stochastic volatility; finite difference methods; super-time-stepping.

*Corresponding author. Both 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

“Surprisingly, there seem to be no recognized rules for the comparison of alternative difference schemes. Clearly there are three fundamental criteria-accuracy, simplicity, and stability”

Gilbert Strang, 1968 [39].

The Black-Scholes partial differential equation [5,27] 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 that 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 [28], Lévy [9], stochastic volatility (SV) [17], SV jump-diffusion [4], and also combinations of these that exhibit SV as well as jumps in both the asset price and volatility [12]. 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 [46] and [40].

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 [33] for vanilla European and American put options.

While the behavior of STS has only been analytically established for symmetric operators [2], a novel splitting approach was used in [33] to deal with non-symmetric operators under STS. 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 [31,32]. In this way, reference [33] 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 discernible adverse implications to its stability properties as a result.

The contribution of this paper to mathematical and 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 the Heston 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.

Monte Carlo (MC) methods can be used to price American style options under the Heston model, such as the least squares MC algorithm of [25]. There are also a number of tree based (explicit) methods, see for example [1,24,14,45,3]. Both MC and tree-based methods discretize the stochastic differential equation describing the joint dynamics of the stock price and its variance and price options by backward recursion. MC methods are very general in terms of their applicability but are computationally expensive. The tree based methods mentioned above are promising alternatives to implicit FD methods however the number of nodes in the grid for the stock price and variance must grow at least quadratically with the number of time steps. Furthermore traditionally two-dimensional tree based methods are complex to implement, although [3] and [1] have made significant progress in reducing implementation complexity.

The focus of this paper is on numerically solving American style options under the Heston model by discretizing and then solving the option pricing PDE. Successive overrelaxation (SOR) is the most popular iterative method for obtaining the solution in the European option case (see [10] for details) while projected SOR (PSOR) [11] 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 Heston model include: an incomplete LU (ILU) pre-conditioned conjugate gradient method with a penalty term devised by [49] to handle the early exercise feature of American options; multigrid methods used by [8] and [30] to price American options; a hopscotch scheme applied by [19] to price American options; a finite element approach constructed by [47] to price European and barrier options; an exponential fitting in combination with splitting and Yanenko method by [36]; various ADI schemes augmented to include the cross derivative correlation term by [22] and an ADI predictor-corrector

scheme by [48]. A number of schemes used to solve LCPs were also compared by [20,21] 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 Heston model.

All of the PDE solver 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 principal selling point of the STS algorithm is that it preserves the simplicity and flexibility of explicit FD schemes while providing efficiency to compete with their implicit counterparts. The local nature of the FD stencil for explicitly differenced PDEs is inherited by the STS accelerated scheme. Parallelization is therefore a trivial endeavor in this context.

The paper is arranged as follows. Sec. 2 reviews the Heston model, the corresponding PDEs describing vanilla European and American option prices, and the associated boundary conditions. Sec. 3 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. Sec. 4 introduces the STS technique. The application of STS to option pricing under SV is the main contribution of the paper. Sec. 5 contains comparative timings and discussion of results. Finally, in Sec. 6, 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 the Heston model. Furthermore this section contains a brief discussion of the implementation issues concerned when the Heston option pricing PDE is solved using FD methods.

2.1. Heston's Stochastic Volatility Model

The Heston model, in the risk neutral measure, will be taken to be of the form:

$$\begin{aligned} dx_t &= rx_t dt + \sqrt{y_t} x_t dz_t, \\ dy_t &= \{\alpha(\beta - y_t) - \lambda\gamma\sqrt{y_t}\} dt + \gamma\sqrt{y_t} dw_t, \\ \rho dt &= dz_t dw_t, \end{aligned} \tag{2.1}$$

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:

$$\mathcal{L}u = \frac{\partial u}{\partial \tau} + \mathcal{A}u = 0, \quad (2.2)$$

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

$$\mathcal{A}u = -\frac{1}{2}yx^2u_{xx} - \rho\gamma yxu_{xy} - \frac{1}{2}\gamma^2yu_{yy} - rxu_x - \{\alpha(\beta - y) - \lambda\gamma\sqrt{y}\}u_y + ru. \quad (2.3)$$

The payoff, or initial condition, on a European put option is $u(x, y, 0) = g(x, y) = \max[E - x, 0]$. European put options satisfy $u(0, y, \tau) = e^{-r\tau}g(0, y)$ on the boundary of $x = 0$. The PDE on the boundary $y = 0$ reduces to

$$\mathcal{L}u = \frac{\partial u}{\partial \tau} + \mathcal{A}u = \frac{\partial u}{\partial \tau} - rxu_x - \alpha\beta u_y + ru = 0. \quad (2.4)$$

We note that through consideration of the Fichera function, the necessity for a boundary condition at $y = 0$ is only present when the Feller condition, $\alpha\beta \geq \gamma^2/2$, is violated. It has been argued in [8] and [48] that, when required, the appropriate boundary condition to use at this boundary is the payoff function. Furthermore, it is claimed in [48] that even when the Feller condition is not violated, the solution should converge to the payoff function at this boundary. We note here that this behavior is not observed in our solutions. As the present work is concerned with the efficiency of numerical methods applied to the Heston model and our results are in good agreement with the literature, we defer further discussion on boundary conditions for the model.

We use the following asymptotic boundary conditions for $x = x_{\max}$ and $y = y_{\max}$ (where x_{\max} is the maximum value of x considered in the approximation of the solution and y_{\max} is defined similarly):

$$\lim_{x \rightarrow \infty} \frac{\partial^2 u(x, y, \tau)}{\partial x^2} = 0, \quad \lim_{y \rightarrow \infty} \frac{\partial^2 u(x, y, \tau)}{\partial y^2} = 0, \quad (2.5)$$

except for the componentwise split scheme (introduced in Sec. 3.2) and the C code implementations used for speed tests in Secs. 5.1 and 5.2 where the boundary condition

$$\lim_{x, y \rightarrow \infty} \frac{\partial^2 u(x, y, \tau)}{\partial x \partial y} = 0 \quad (2.6)$$

is deployed.

2.3. American Put Options

American options may be exercised early leading to the early exercise constraint $u(x, y, \tau) \geq g(x, y)$. In the continuation region where the constraint is inactive u satisfies the same PDE as the European option, $\mathcal{L}u = 0$, as given by Eq. 2.2. Combining these relations we can write the American option pricing problem as a time dependent LCP (for example see [21])

$$\begin{cases} \mathcal{L}u \geq g, & u \geq g, \\ (\mathcal{L}u)(u - g) = 0, \end{cases} \quad (2.7)$$

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 $u(0, y, \tau) = g(0, y)$. The linear complementarity conditions on the boundary $y = 0$ are

$$\begin{cases} \mathcal{L}u = \frac{\partial u}{\partial \tau} + \mathcal{A}u = \frac{\partial u}{\partial \tau} - rxu_x - \alpha\beta u_y + ru \geq g, & u \geq g, \\ (\mathcal{L}u)(u - g) = 0. \end{cases} \quad (2.8)$$

The boundary conditions adopted at $x = \infty$ and $y = \infty$ are the same as those for the European put option given by Eqs. 2.5 or 2.6. This is the nearly equivalent system to that described by [8,21,30,49], the only difference being we use second derivative asymptotic boundary conditions at $x = \infty$ and $y = \infty$ as opposed to first derivative boundary conditions.

3. Implementation Details

Although implementation details are crucial in any FD scheme the focus of this paper is on the application of super-time-stepping in solving the option pricing PDE from the Heston model. Hence implementation details are briefly covered in this section with additional detail available from quoted references.

Our implementation details follow closely the details described by [21]. We use grid generating functions (GGFs) to increase computational efficiency by decreasing the density of mesh points away from regions of interest. Furthermore we restrict the GGFs order to ensure the spatial operator is an \mathbf{M} -matrix (this is done to improve the stability properties of a numerical scheme called componentwise splitting that is used as a benchmark in this paper, for more details see [21]). Like [21] we choose the stock price GGF to be a quadratic function of the stock price subject to the \mathbf{M} -matrix constraints. The stock price grid spacing is set to be four times as dense at the exercise price compared to the grid spacing at the maximum stock price. Preserving the \mathbf{M} -matrix condition constricts the shape of the stock price GGF at lower stock prices to be a straight line from the origin superimposed onto the quadratic GGF as illustrated in figure 3(a). Furthermore we use a linear

GGF for the variance and set the variance grid spacing to be twice as dense at low variance values compared to the grid spacing at high variance values, see figure 3(b).

We also employ upwinding when the PDE becomes convection dominated to avoid spurious oscillations in these regions as described by [21]. The main difference in our grid generation procedure is that we iterate to fit the parameters dictating the point density over the domain exactly. We use a conventional nine point stencil as opposed to the seven point prescription of [21], except for in the implementation of the componentwise splitting benchmark method, and in all cases for the C code timings quoted in Tabs. 2 and 4, where we use a seven point stencil.

3.1. Spatial Discretization

The FD discretizations are constructed on a non-uniform grid

$$(x_i, y_j, \tau_k) \in \{0 = x_0, \dots, x_m = X\} \times \{0 = y_0, \dots, y_n = Y\} \times \{0 = \tau_0, \dots, \tau_l = T\}. \quad (3.1)$$

Standard FD methods are used to discretize the spatial operator \mathcal{A} in Eq. 2.3. Centered 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. Details on the treatment of one particular choice of boundary conditions and the implementation of upwinding techniques are given in Appendix A and Appendix B respectively.

The Heston PDE at a reference point $u = u(x_i, y_j)$ can be written compactly as

$$\mathcal{L}u = \frac{\partial u}{\partial \tau} + \mathcal{A}u = 0, \quad (3.2)$$

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

$$\mathcal{A} = \begin{bmatrix} A_{lu} & A_u & A_{ru} \\ A_l & A_c & A_r \\ A_{ld} & A_d & A_{rd} \end{bmatrix}, \quad (3.3)$$

with

$$\begin{aligned} A_c &= - \left(\frac{1}{2} y x^2 a^D + \rho \gamma x y c^D + \frac{1}{2} \gamma^2 y b^D + r x a^C + \alpha (\beta - y) b^C - r \right), \\ A_p &= - \left(\frac{1}{2} y x^2 a_p^D + \rho \gamma x y c_p^D + r x a_p^C \right), \text{ for } p \in \{r, l\} \\ A_p &= - \left(\frac{1}{2} \gamma^2 y b_p^D + \rho \gamma x y c_p^D + \alpha (\beta - y) b_p^C \right), \text{ for } p \in \{u, d\} \\ A_p &= - \rho \gamma x y c_p^D, \text{ for } p \in \{ru, rd, lu, ld\}. \end{aligned}$$

where a_p^q, b_p^q, c_p^q for $p = \{r, l, u, d, c, ru, rd, lu, ld\}$ and $q = \{C, D\}$ are the coefficients applied to the central FD schemes used for the convection, C , and diffusion, D , components at the inner points

of the computational mesh (away from the boundaries) when using grid generating functions. The approach described above is easily applied to the case of American options as given by the time dependent LCP Eq. 2.7.

3.2. Time Discretization

In this section the explicit, 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

$$\frac{\partial \mathbf{u}}{\partial \tau} + \mathbf{A}\mathbf{u} = 0, \quad (3.4)$$

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 and n are the number of steps in the x -direction and y -direction respectively. 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 $i = 0, \dots, m$ and $j = 0, \dots, n$. The θ -method of discretization applied to Eq. 3.4 results in the following system:

$$\mathbf{B}\mathbf{u}^{(k+1)} - \mathbf{C}\mathbf{u}^{(k)} = 0, \text{ for } k = 0, 1, \dots, l-1, \quad (3.5)$$

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. 3.5 becomes a time dependent LCP as described in Eq. 2.7

$$\begin{cases} \mathbf{B}\mathbf{u}^{(k+1)} \geq \mathbf{C}\mathbf{u}^{(k)}, \mathbf{u}^{(k+1)} \geq \mathbf{g} \\ (\mathbf{B}\mathbf{u}^{(k+1)} - \mathbf{C}\mathbf{u}^{(k)})^T (\mathbf{u}^{(k+1)} - \mathbf{g}) = 0, \end{cases} \quad (3.6)$$

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 explicit 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 [46,40] for further details). In particular when correlation $\rho = 0$, the theoretical upper limit on the time step is given by

$$\Delta\tau_{\text{expl}} \leq \min_{\substack{0 \leq i \leq m \\ 0 \leq j \leq n}} \left\{ \left[\frac{x_i^2 y_j}{h(x_i)^2} + \frac{\gamma^2 y_j}{h(y_j)^2} \right]^{-1} \right\} \quad (3.7)$$

where $h(x_i) = x_{i+1} - x_i$ for $i = 0, \dots, m$ and $h(y_j) = y_{j+1} - y_j$ for $j = 0, \dots, n$ are the non-uniform grid spacings in the x and y directions. In the more general case of non-zero correlation Eq. 3.7 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 is the CFL constraint on the time step which may be severely restrictive.

When $\theta = 1$ the θ -method corresponds to the fully implicit Euler 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 time-stepping [34] is typically used in the CN algorithm. For completeness, we note that 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.

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 penalty (see [8,30,21]). Direct methods also exist such as the componentwise splitting formulation presented by [21] of the LU decomposition algorithm in [6].

We elect CN with PSOR, denoted CN-PSOR, as the standard benchmark scheme for comparison with the proposed STS approach in tests solving Eq. 3.6. CN-PSOR is a suitable choice due to its competitive performance with respect to other methods and its widespread usage (e.g. [40,46]). The second benchmark scheme we chose is the CN componentwise splitting method of [21], denoted by CN-PLU. This method splits the two dimensional spatial operator into three spatial operators along the x , y and xy directions. Then a series of one-dimensional LCPs are directly solved along each direction using LU decomposition with projection of the early exercise value in the backward substitution step. This method is more efficient than CN-PSOR.

As noted by [21], CN-(P)LU is sensitive to the \mathbf{M} -matrix property (realized by a diagonally dominant matrix with positive diagonal elements and non-positive off-diagonal elements). To maintain the \mathbf{M} -matrix property requires non-trivial tuning of the grid-generation procedure. In particular, for given grid characterization parameter values (such as the ratios of grid spacing at different positions), the problem can become intractable. We note that in this work, we employ an exact (iterative) approach to the fitting of these parameters in contrast to [21] approximate method. Requiring the \mathbf{M} -matrix property also imposes the condition that the lower boundary in x may not be set identically to zero. The projection employed by the Brennan Schwartz algorithm [6] for American style payoff functions is only effective when the location of the optimal exercise boundary is single valued for all sweeps in all directions. As noted by [21], it may be possible to relax this restriction at the

expense of efficiency.

We also observe that the implementation of self-consistent boundary conditions becomes difficult under componentwise splitting. This is due to the fact that at each of the far-field boundaries the one-dimensional sweeps in the directions both normal and oblique to the boundary influence the solution. The farfield Neumann boundary conditions $\lim_{x \rightarrow \infty} \frac{\partial u(x,y,\tau)}{\partial x} = 0$ and $\lim_{y \rightarrow \infty} \frac{\partial u(x,y,\tau)}{\partial y} = 0$, as used by [21], provide one possible consistent treatment. In this work, we prefer to apply the second order condition given by Eq. 2.6 through an *a priori* adjustment of the terms in the coefficient matrix.

On a note of more debatable significance, since directionally split schemes necessarily order the directional sweeps, one must be conscious of the possibility of the influence of the inherent directional bias in any solution obtained. In this case it would appear the influence is not significant (unless subtle). However, in general, it is not a desirable property that the concern exists in the first place. Lastly, it is worth stating that the LU scheme is significantly more complex in its implementation than any of the other schemes considered here.

The equation for the European option price, Eq. 2.2, may be solved analytically [17]. However, as a proof of concept we test the efficacy of STS in solving Eq. 2.2 with respect to CN-SOR, a CN scheme iteratively solved via simple successive overrelaxation (SOR) and with respect to CN-LU, the CN componentwise splitting scheme described above without the projection of the early exercise value. 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 [2], itself a variant of a method presented by [15], which is in turn essentially a pared-down Runge-Kutta-Chebyshev (RKC) method [41,42,43,44,38].

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 [13]; electromagnetic wave scattering [37]; isotropic and anisotropic diffusion on biological membranes [35]; and magnetic field diffusion in astrophysics [32,29]. Recently, STS has been applied in finance and rigorously tested for the Black-Scholes pricing model by [33] where it was shown to

be comparable, and in many cases superior, to the CN-(P)SOR scheme in terms of accuracy and computational speed for European (American) put options.

To proceed we consider the PDE

$$\frac{\partial \mathbf{u}}{\partial \tau} + \mathbf{A}\mathbf{u} = 0, \quad \mathbf{u}(\mathbf{0}) = \mathbf{u}^0, \quad (4.1)$$

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

$$\mathbf{u}^{k+1} = (\mathbf{I} - \Delta\tau_{\text{sts}}\mathbf{A})\mathbf{u}^k \quad (4.2)$$

where the solution at time level $k+1$ is solved recursively using the solution at time level k , \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. 4.2 corresponds to the $\theta = 0$ instance of the θ -method corresponding to the explicit Euler scheme as described in Sec. 3.2. For stability the CFL constraint requires that

$$|1 - \Delta\tau\lambda| < 1 \quad (4.3)$$

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

$$\Delta\tau_{\text{expl}} \equiv \frac{2}{\lambda_{\text{max}}} \quad (4.4)$$

where λ_{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

$$\Delta\tau_{\text{sts}} = \sum_{j=1}^{N_{\text{sts}}} \Delta\tau_j \quad (4.5)$$

and stability is only demanded at the end of the super-step ^a. We write the compound scheme as

$$\mathbf{u}^{k+1} = \left[\prod_{j=1}^{N_{\text{sts}}} (\mathbf{I} - \Delta\tau_j \mathbf{A}) \right] \mathbf{u}^k. \quad (4.6)$$

^aIn [44] it is claimed that factorized RKC methods are impractical as they suffer from internal instability. Reference [33] find no evidence of any such instability influencing their solutions for cases with $N_{\text{sts}} \lesssim 30$.

[2] prove that the error at time level k between the exact solution to Eq. 4.1 and the approximate solution to scheme in Eq. 4.6 is given by

$$\|\mathbf{e}^k\| \leq k \frac{\lambda_{\max}}{2} \left(\sum_{j=1}^{N_{\text{sts}}} \Delta\tau_j^2 \right) \|\mathbf{u}^0\|. \quad (4.7)$$

Hence the method is essentially of order one with respect to the super-time-step $\Delta\tau_{\text{sts}}$.

For stability we require

$$\left| \prod_{j=1}^{N_{\text{sts}}} (1 - \Delta\tau_j \lambda) \right| < 1 \quad (4.8)$$

for all eigenvectors λ of \mathbf{A} . The properties of Chebyshev polynomials of degree N_{sts} [26] provide the means to explicitly enforce stability while maximizing $\Delta\tau_{\text{sts}}$. The optimal values for the sub-steps obtained in this way are given by

$$\Delta\tau_j = \Delta\tau_{\text{expl}} \left[(-1 + \nu) \cos \left(\frac{2j-1}{N_{\text{sts}}} \frac{\pi}{2} \right) + 1 + \nu \right]^{-1} \quad (4.9)$$

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

$$\Delta\tau_{\text{sts}} \rightarrow N_{\text{sts}}^2 \Delta\tau_{\text{expl}} \quad \text{as } \nu \rightarrow 0. \quad (4.10)$$

In practice the scheme may be marginally stable for low enough values of ν for a given choice of N_{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 $N_{\text{sts}} = 30$ in Fig. 4 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 $\nu \rightarrow 0$ 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_{sts} steps of size Δt_{expl} according to Eq. 4.10.

STS applied directly as described above results in a scheme which is first order in time [2]. 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 Eq. (4.6) is symmetric positive definite [2]. 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 [33]. The scheme presented therein is formally stable under application of STS to the symmetric component of the multiplicatively split operator. The skew symmetric part of the operator is then integrated via a novel scheme developed by [31,32]. While the split scheme presented by [33] 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 Eq. (4.6) to the FD schemes described in Sec. 3.2 we include a brief section on the use of RE in the STS scheme to achieve a second order (in time) 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 for STS as applied by [23] to first order fully implicit schemes. We assume a smoothly convergent first order accurate method for the temporal integration of the semi-discrete Eq. (4.1) 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 $u_{i,j}^k = u_{\Delta x, \Delta y}(i\Delta x, j\Delta y, k\Delta\tau) + (L-k)\mathcal{O}(\Delta\tau^3)$ we may take a single step of size $\Delta\tau$ to approximate the solution at time level $k+1$

using $u_{ij}^{k+1}(\Delta\tau) = u_{\Delta x, \Delta y}(i\Delta x, j\Delta y, (k+1)\Delta\tau) + C\Delta\tau^2 + \mathcal{O}(\Delta\tau^3)$ 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 $u_{ij}^{k+1}(\Delta\tau/2) = u_{\Delta x, \Delta y}(i\Delta x, j\Delta y, (k+1)\Delta\tau) + (C/2)\Delta\tau^2 + \mathcal{O}(\Delta\tau^3)$. Subtracting the expression for $u_{ij}^{k+1}(\Delta\tau)$ from twice the expression for $u_{ij}^{k+1}(\Delta\tau/2)$ yields a second order advancement in the solution from time level k to level $k+1$ according to

$$u_{ij}^{k+1} = 2u_{ij}^{k+1}(\Delta\tau/2) - u_{ij}^{k+1}(\Delta\tau), \text{ for } k = 0, 1, \dots, l-1. \quad (4.11)$$

We also consider the more usual post-processed form which requires two independently derived solutions for use in the extrapolation of the final solution $u_{ij}^l = 2u_{ij}^l(\Delta\tau/2) - u_{ij}^l(\Delta\tau)$, see for example [16].

We refer to the scheme presented in [23] as local RE and the latter scheme 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 analyze the efficiency of the STS accelerated explicit scheme relative to the unaccelerated explicit scheme, the Crank Nicolson scheme and the componentwise splitting scheme applied to the pricing of European and American options under the Heston 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 [49,8,30,21]. (Other parameter settings are also tested at the end of this section.) 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. The same grid generating functions, as described in Sec. 3, are used to prescribe the interior grid points for all the FD schemes considered in these experiments. 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 behavior of STS driven acceleration as $\nu \rightarrow 0$ 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 Fig. 5 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 50 as $\nu \rightarrow 0$ is clearly evident in panel (b) (where the abscissa are logarithmic rather than linear in ν). From Eq. 4.10, and using the minimum number of explicit steps required for stability on this grid (estimated to be approximately $l_{expl} = 31,453$ from Eq. 3.7), we expect a limiting value of $l = \frac{l_{expl}}{N_{sts}^2} \approx 51$. This is clearly in good agreement with our experimentally derived value and represents 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. In these numerical experiments the STS-1 scheme remains stable for all values of l greater than or equal to 51. However to ensure stability we step back from the stability limit and set $\nu = 0.001$ for the European option experiments and $\nu = 0.002$ for the American option experiments where more timesteps are needed to capture the early exercise boundary. 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 behavior of the schemes as both the spatial and temporal resolutions are increased simultaneously. 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 the Heston model. Table 1 displays European put option prices at five stock prices $x = 8, \dots, 12$ for an initial variance $y = 0.0625$ at a number of different resolutions. Six 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); the componentwise splitting scheme with LU decomposition (CN-LU) and the the Crank-Nicolson scheme solved by means of successive overrelaxation (CN-SOR). 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 [7] and a high resolution CN-SOR method are included for reference. 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} .

The CN-SOR European put prices are computed with a convergence tolerance measure $tol =$

1×10^{-4} ($tol = 1 \times 10^{-5}$ 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 $N_{sts} = 25$ and a fixed damping parameter of $\nu = 0.001$. The number of supersteps, l , is scaled in proportion to the number of points on the spatial grid in each dimension although we purposely use a low number of timesteps to emphasize the acceleration achieved by the STS scheme.

It is clear from Table 1 that all six 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.

Fig. 6 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 $(x, y) = ([8, 9, \dots, 12] \times [0.0625, 0.25])$.

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 characteristics of the spatial error. We also note that the plots are qualitatively 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 , $x = 8, 9, \dots, 12$, 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 (STS-RE-L and STS-RE-G) 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 and CN-LU schemes have the highest l_2 norm errors 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, CN-LU and the standard explicit methods.

Table 2 gives the times for each scheme using MATLAB implementations. These indicate that on the finest grid the STS-1 method is least computationally expensive, followed by STS-RE-L, STS-RE-G, CN-LU, 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.

The timings for the tests listed in Tab. 2 are plotted in Fig. 1. The left panel presents the timings for all of the MATLAB codes as a function of the error as calculated in Tab. 2. The right panel provides the C code timings. In each case reference lines are shown for inverse quadratic and inverse quartic dependencies. It is evident that, for the schemes which are nominally first order accurate in time (STS-1 and EXPL-1), to reduce the error in the solution by a factor of 2 requires an increase in the time expended in calculation by a factor of 16. For the remaining schemes, to reduce the error by the same factor requires an increase in the computational timeload by a factor of 4. At the lowest, resolution for which the spatial and temporal intervals are extremely coarse, this factor is closer lower. This is possibly related to the influence of the computational overheads for each of the runs.

Lastly, when scheme timings are evaluated in C implementations the CN-LU scheme is the least computationally expensive scheme, followed by STS-1, STS-RE-L, STS-RE-G, CN-SOR and finally

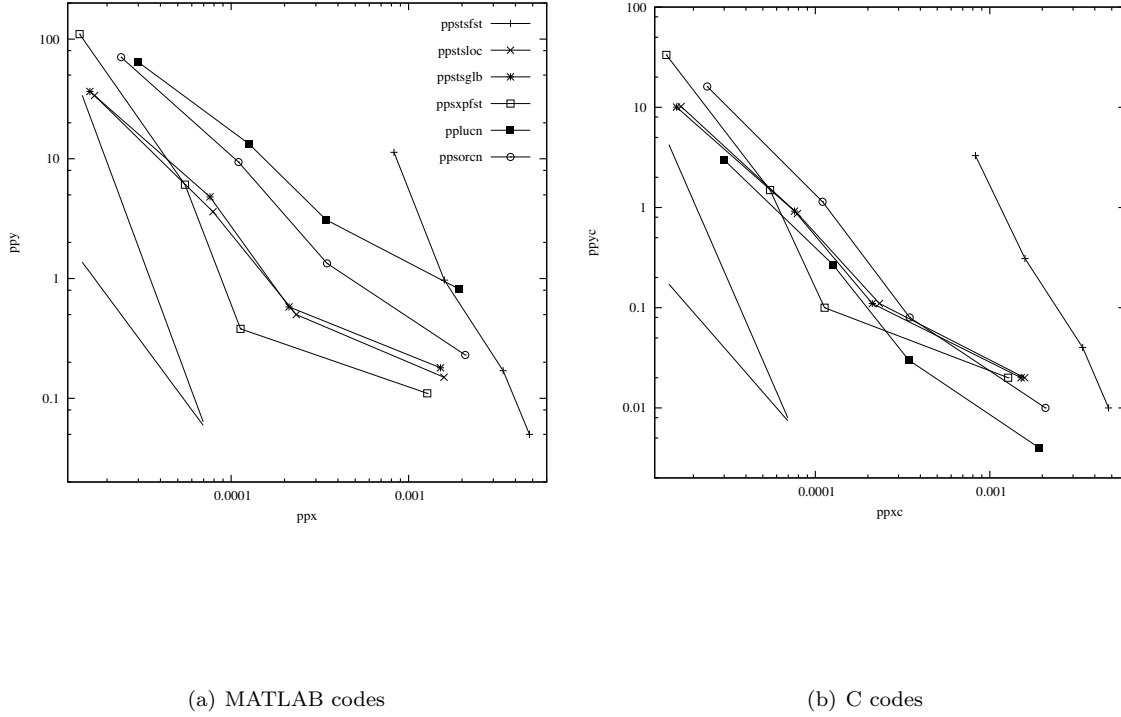


Fig. 1. **Benchmark timings on European option pricing against error.**

The time in seconds for various schemes to achieve the test solution to the given error as presented in Tab. 2. The error is measured as the l^2 -norm error of ten European put prices at stock prices of $x = 8, 9, \dots, 12$, and initial variance values of $y = 0.0625$ and 0.25 . The left panel presents the timings for the MATLAB codes and the right panel presents the timings for the C codes. In each panel reference lines proportional to $(\text{time})^{-2}$ and $(\text{time})^{-4}$ are shown.

EXPL-1. However the CN-LU is a highly complex scheme to implement relative to all other schemes tested in this paper including the STS-2 schemes. Furthermore the CN-LU scheme requires that the spatial operator of the discretized PDE must be an \mathbf{M} -matrix. This condition is satisfied for the European options tested at the default parameters in Tabs. 1 and 2. However the \mathbf{M} -matrix condition cannot be satisfied using a different set of parameters relevant to equity market options, see Tabs. 6 and 7. The CN-LU scheme does not converge to a suitable solution under these more typical Heston parameter values. In this case STS-2 schemes are the most efficient schemes amongst those considered.

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. A direct consequence is that achieving convergence becomes more laborious with increasing timestep size. The CN-PLU scheme also projects the early exercise condition onto the solution in a backward substitution step of the LU decomposition.

Table 3 displays reference American put option prices at five stock prices $x = 8, \dots, 12$ 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, CN-PLU and CN-PSOR schemes using various grid resolutions. All STS American put prices are computed with a reduced 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. The number of timesteps used in the EXPL-1, l , doubles from the coarsest to next coarsest grid but thereafter scales quadratically due to the non-uniform grid generating functions. 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.

Fig. 7 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 Fig. 7 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 , $x = 8, 9, \dots, 12$, 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 behavior 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 scheme timings also provided by table 4, we find that, similarly to the European case, STS-2 schemes are faster than the CN-PLU and CN-PSOR schemes and display lower errors when using MATLAB implementations. Additionally, the STS-1 scheme is observed to be approximately six times faster than the CN-PSOR scheme. When timings are compared in C implementations, CN-PLU is the most efficient scheme, followed by the STS-2 schemes. However as previously mentioned the CN-PLU scheme only converges if the spatial operator of the discretized PDE is an M-matrix and the location of the optimal exercise boundary is single valued for all sweeps in all directions. The second condition is satisfied by the American options considered in this paper but the first condition is only satisfied at the default parameters. When a different set of Heston parameters are used in Tabs. 6 and 7 the CN-PLU scheme does not converge hence the STS-2 schemes become the most efficient schemes amongst those tested.

The timings for the tests listed in Tab. 4 are plotted in Fig. 2. As for Fig. 1, the left panel presents the timings for all of the MATLAB codes as a function of the error as calculated in Tab. 4. The right panel provides the C code timings. In each case reference lines are shown for inverse quadratic and inverse quartic dependencies. Similarly to Fig. 1, it is clear that, for the schemes which are nominally first order accurate in time (STS-1 and EXPL-1), to reduce the error in the solution by a factor of 2 requires an increase in the time expended in calculation by a factor of 16. For the remaining schemes, to reduce the error by the same factor requires an increase in the computational timeload by a factor of 4. Again, at the lowest resolution, there is a deviation from this behavior, perhaps related to initialisation and assignment overheads in carrying out timing tests over very short times.

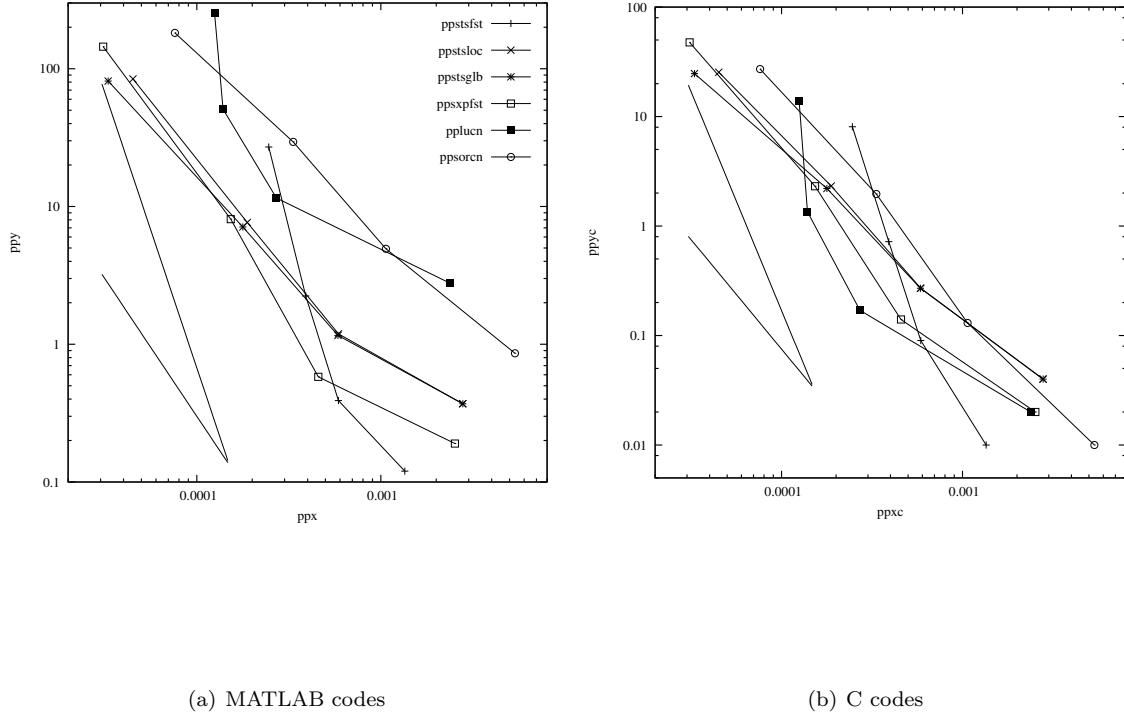


Fig. 2. **Benchtest timings on Amerpean option pricing against error.**

The time in seconds for various schemes to achieve the test solution to the given error as presented in Tab. 4. The error is measured as the l^2 -norm error of ten Amerpean put prices at stock prices of $x = 8, 9, \dots, 12$, and initial variance values of $y = 0.0625$ and 0.25 . The left panel presents the timings for the MATLAB codes and the right panel presents the timings for the C codes. In each panel reference lines proportional to $(\text{time})^{-2}$ and $(\text{time})^{-4}$ are shown.

Notably, CN-LU appears to hit a wall in its accuracy at the highest resolution presented. As this does not occur in testing for the European analog, it is suggested that this is a consequence of the influence of the free boundary on the componentwise split scheme.

Table 5 displays American put option prices at the five stock prices $x = 8, 9, \dots, 12$ 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 [21] is strong.

As a final exercise in validation, Tabs. 6 and 7 display the convergence of STS-RE-G European and American put prices to benchmark prices for a different set of Heston parameters that would be typical of a calibration of the model to equity market options. We also test different time-to-maturity values and initial variance values. Both tables indicate the second order convergence of STS-RE-G European and American put prices to benchmark prices and illustrate the high accuracy achieved by the scheme with percentage differences between STS-RE-G prices and reference prices being less than 1% on coarse resolutions and quickly converging to even higher accuracy on finer resolution grids. Although not reported similar tables were produced for the CN-(P)SOR scheme where the European and American put prices displayed a lower accuracy and required longer run times. Finally the CN-(P)LU scheme could not produce prices at these parameters values.

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 importance, 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.

Appendix 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

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

where

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

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

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

where

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

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

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

Appendix B. Upwinded Differencing

Recall from Sec. 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

$$\begin{aligned} A_c &= - \left(\frac{1}{2} y x^2 a^D + \rho \gamma x y c^D + \frac{1}{2} \gamma^2 y b^D + r x a^C + \alpha (\beta - y) b^C - r \right), \\ A_l &= - \left(\frac{1}{2} y x^2 a_l^D + \rho \gamma x y c_l^D + r x a_l^C \right), \\ A_r &= - \left(\frac{1}{2} y x^2 a_r^D + \rho \gamma x y c_r^D + r x a_r^C \right), \\ A_d &= - \left(\frac{1}{2} \gamma^2 y b_d^D + \rho \gamma x y c_d^D + \alpha (\beta - y) b_d^C \right), \\ A_u &= - \left(\frac{1}{2} \gamma^2 y b_u^D + \rho \gamma x y c_u^D + \alpha (\beta - y) b_u^C \right), \\ A_{ld} &= -\rho \gamma x y c_{ld}^D, \quad A_{lu} = -\rho \gamma x y c_{lu}^D \\ A_{rd} &= -\rho \gamma x y c_{rd}^D, \quad A_{ru} = -\rho \gamma x y c_{ru}^D, \end{aligned}$$

where the superscript D denotes diffusion terms and the superscript C denotes convection terms defined in Sec. 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 example A_c is adjusted as follows:

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

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

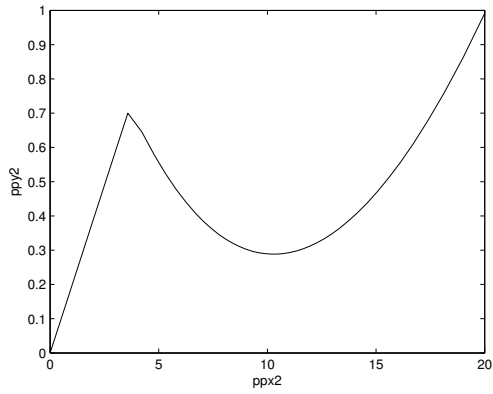
$$\begin{aligned} a_l^C &\rightarrow a_l^C \mathbf{1}_{\{A_l < 0, A_r < 0\}} + a_l^{CF} \mathbf{1}_{\{A_l > 0\}} + a_l^{CB} \mathbf{1}_{\{A_r > 0\}}, \\ a_r^C &\rightarrow a_r^C \mathbf{1}_{\{A_l < 0, A_r < 0\}} + a_r^{CF} \mathbf{1}_{\{A_l > 0\}} + a_r^{CB} \mathbf{1}_{\{A_r > 0\}}, \\ b_d^C &\rightarrow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

References

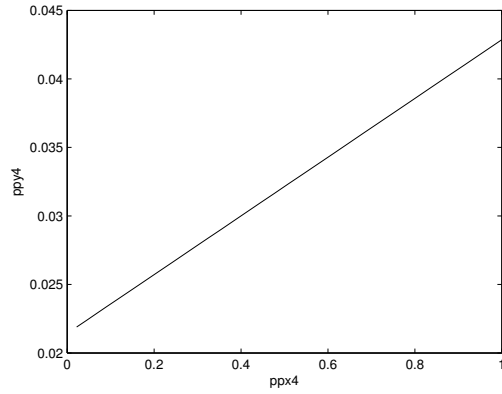
- [1] E. Akyildirim, Y. Dolinsky and H. Mete Soner, *Approximating Stochastic Volatility by Recombinant Trees* (arXiv:1205.3555v1, 2012).
- [2] V. Alexiades, G. Amiez and P. Gremaud, Super-Time-Stepping acceleration of explicit schemes for parabolic problems, *Com. Num. Meth. Eng.* **12** (1996) 31–42.
- [3] N. A. Beliaeva and S. K. Nawalkha, A Simple Approach to Pricing American Options Under the Heston Stochastic Volatility Model, *Journal of Derivatives* **17**(4) (2010) 25–43.
- [4] D. Bates, Jumps and Stochastic Volatility: Exchange Rate Processes Implicit in Deutsche Mark Options, *Review of Financial Studies* **9** (1973) 69–107.
- [5] F. Black and M. S. Scholes, The pricing of options and corporate liabilities, *Journal of Political Economy* **81** (1973) 637–654.
- [6] M. Brennan and E. Schwartz, The valuation of American put options, *Journal of Finance* **32** (1977) 449–462.
- [7] P. Carr and D. Madan, Option valuation using the fast Fourier transform, *Journal of Computational Finance* **2** (1999) 61–73.
- [8] N. Clarke and K. Parrot, Multigrid for American option pricing with stochastic volatility, *Applied Mathematical Finance* **6** (1999) 177–195.
- [9] R. Cont and P. Tankov, *Financial Modelling With Jump Processes* (Chapman & Hall/CRC Financial Mathematics Series, 2004).
- [10] J. Crank, *Free and moving boundary problems* (Oxford: Clarendon Press, 1984).
- [11] C. W. Cryer, The solution of a quadratic programming problem using systematic overrelaxation, *SIAM Journal on Control* **9** (1971) 385–392.
- [12] D. Duffie, J. Pan and K. Singleton, Transform analysis and asset pricing for affine jump-diffusions, *Econometrica* **68** (2000) 1343–76.
- [13] S. Evje, K. H. Karlsen and N. H. Risebro, A continuous dependence result for nonlinear degenerate parabolic equations with spatially dependent flux function, in *Int. Series of Numerical Mathematics* **140** (Birkhauser-Verlag, 2001) 337–346.
- [14] I. Florescu, and F. G. Viens, Stochastic Volatility: Option Pricing using a Multinomial Recombining Tree, *Applied Mathematical Finance* **15** (2008) 151–181.
- [15] W. Gentsch, Numerical solution of linear and non-linear parabolic differential equations by a time-discretisation of third order accuracy, in *Proceeding of the Third GAMM-Conference on Numerical Methods in Fluid Mechanics in Hirschel, E. H. (Eds.)* (Friedr. Vieweg & Sohn, 1979) 109–117.
- [16] R. Geske and H. E. Johnson, The American Put Option Valued Analytically, *The Journal of Finance* **39** (1984) 1511–1524.
- [17] S. L. Heston, A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options, *The Review of Financial Studies* **6**(2) (1993) 327–343.
- [18] T. Kluge, *Pricing derivatives in stochastic volatility models using the finite difference method* (Diploma Thesis, Chemnitz University of Technology, 2002).
- [19] A. Kurpiel and T. Roncalli, Hopscotch Methods for Two State Financial Models, *Journal of Computational Finance* **3**(2) (2000) 53–89.
- [20] S. Ikonen and J. Toivanen, Componentwise splitting methods for pricing American options under stochastic volatility, *International Journal of Theoretical and Applied Finance*, **10**(2) (2007) 331–361.

- [21] S. Ikonen and J. Toivanen, Efficient numerical methods for pricing American options under stochastic volatility, *Numerical Methods for Partial Differential Equations*, **24**(1) (2007) 104–126.
- [22] K. J. In't Hout and S. Foulon, ADI Finite Difference Schemes for Option Pricing in the Heston Model with Correlation, *International Journal of Numerical Analysis and Modeling* **7**(2) (2010) 303–320.
- [23] J. D. Lawson and J. Ll. Morris, The Extrapolation of First Order Methods for Parabolic Partial Differential Equations, *SIAM Journal on Numerical Analysis* **15**(6) (1978) 1212–1224.
- [24] D. Leisen, Stock Evolution under Stochastic Volatility: A Discrete Approach, *Journal of Derivatives* **8**(2) (2000) 9–27.
- [25] F. A. Longstaff and E. S. Schwartz, Pricing American Options by Simulation: A Simple Least-Squares Approach, *Review of Financial Studies*, **14**(1) (2001) 113–147.
- [26] W. Markoff, Über Polynome, die in einem gegebenen Intervalle möglichst wenig von null abweichen, *Mathematische Annalen*, **77** (1916) 213–258, translated by J. Grossman from original Russian article published in 1892.
- [27] R. C. Merton, The theory of rational option pricing, *Bell Journal of Economics and Management Science* **4** (1973) 141–183.
- [28] R. C. Merton, Option Pricing when the Underlying Returns are Discontinuous, *Journal of Financial Economics* **3** (1976) 125–144.
- [29] A. Mignone, G. Bodo, S. Massaglia, T. Matsakos, O. Tesileanu, C. Zanni and A. Ferrari, PLUTO: A Numerical Code for Computational Astrophysics, *The Astrophysical Journal Supplement Series* **170**(1) (2007) 228–242.
- [30] C. W. Oosterlee, On multigrid for linear complementarity problems with applications to American-style options, *Electronic Transactions Numerical Analysis* **15** (2003) 165–185.
- [31] S. O'Sullivan and T. P. Downes, An explicit scheme for multifluid magnetohydrodynamics, *Monthly Notices of the Royal Astronomical Society* **366** (2006) 1329–1336.
- [32] S. O'Sullivan and T. P. Downes, A three-dimensional numerical method for modelling weakly ionized plasmas, *Monthly Notices of the Royal Astronomical Society* **376** (2007) 1648–1658.
- [33] S. O'Sullivan and C. O'Sullivan, On the acceleration of explicit finite difference methods for option pricing, *Quantitative Finance* **11**(8) (2011) 1177–1191.
- [34] R. Rannacher, Finite element solution of diffusion problems with irregular data, *Numerical Mathematics* **43** (1984) 309–327.
- [35] I. F. Sbalzarini, A. Hayer, A. Helenius and P. Koumoutsakos, Simulations of (an)isotropic diffusion on curved biological surfaces, *Biophysical Journal* **90**(3) (2006) 878–885.
- [36] R. Sheppard, *Pricing Equity Derivatives under Stochastic Volatility: A Partial Differential Equation Approach* (MSc Thesis, University of the Witwatersrand, 2007).
- [37] Y. Shi, L. Li and C. H. Liang, Multidomain pseudospectral time-domain algorithm based on super-time-stepping method, *IEEE Proceedings Microwaves, Antennas and Propagation* **153** (2006) 55–60.
- [38] B. P. Sommeijer, L. F. Shampine and J. G. Verwer, *RKC: An explicit solver for parabolic PDEs* (Technical Report MAS-R9715, CWI Amsterdam, 1997).
- [39] G. Strang, On the construction and comparison of difference schemes, *SIAM Journal of Numerical Analysis*, **5** (1968) 506–517.
- [40] D. Tavella and C. Randall, *Pricing financial instruments: the finite difference method* (John Wiley and Sons, New York, 2000).
- [41] P. J. van der Houwen, *Construction of integration formulas for initial value problems* (North-Holland,

- Amsterdam-New York, 1977).
- [42] P. J. van der Houwen and B. P. Sommeijer, On the internal stability of explicit m-stage Runge–Kutta methods for large values of m, *Z. Angew. Math. Mech.* **60** (1980) 479–485.
 - [43] J. G. Verwer, W. H. Hundsdorfer and B. P. Sommeijer, Convergence Properties of the Runge-Kutta-Chebyshev Method, *Numer. Math.* **57** (1990) 157–178.
 - [44] J. G. Verwer, Explicit Runge-Kutta methods for parabolic partial differential equations, *Appl. Num. Math* **22** (1996) 359–379.
 - [45] M. Vellekoop and H. Nieuwenhuis, A tree-based method to price American options in the Heston model, *Journal of Computational Finance*, **13**(1) (2009) 1–21.
 - [46] P. Wilmott, S. Howison and J. Dewynne, *Option pricing: mathematical models and computation* (Oxford Financial Press, 1995).
 - [47] G. Winkler, T. Apel and U. Wystup, *Valuation of options in Heston's stochastic volatility model using finite element methods* (Working Paper, Chemnitz University of Technology, 2001). .
 - [48] S.-P. Zhu and W.-T. Chen, A predictorcorrector scheme based on the ADI method for pricing American puts with stochastic volatility, *Computers and Mathematics with Applications* **62** (2011) 1–26.
 - [49] R. Zvan, P. A. Forsyth, and K. R. Vetzal, Penalty methods for pricing American options with stochastic volatility, *Journal of Computational and Applied Mathematics* **91** (1998) 199–218.



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



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

Fig. 3. 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 emphasize how the grid step size reduces around areas of interest such as the exercise price or low variance values.

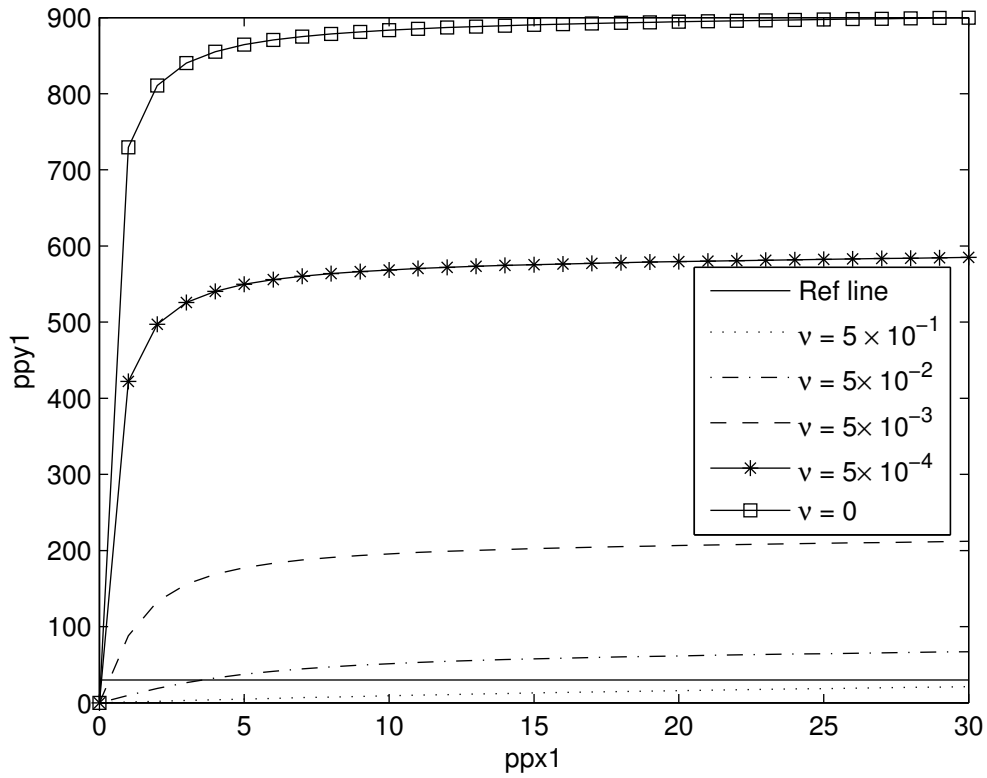


Fig. 4. Illustration of acceleration via STS.

Fig. 4 plots accumulated time, $\sum_{k=1}^j \Delta\tau_k$, in STS versus the substep number j over a single superstep $\Delta\tau_{sts}$ with $N_{sts} = 30$ for a range of damping factors ν . The accumulated time is plotted in units of the standard explicit timestep $\Delta\tau_{expl}$. A reference line at $N_{sts}\Delta\tau_{expl}$ ($= 30$ in units of $\Delta\tau_{expl}$) indicates the time attained over N_{sts} unaccelerated standard explicit steps. Note that acceleration approaches N_{sts} times this value as $\nu \rightarrow 0$, in agreement with equation (4.10). Note also that deceleration occurs for the highest considered damping factor of $\nu = 0.5$.

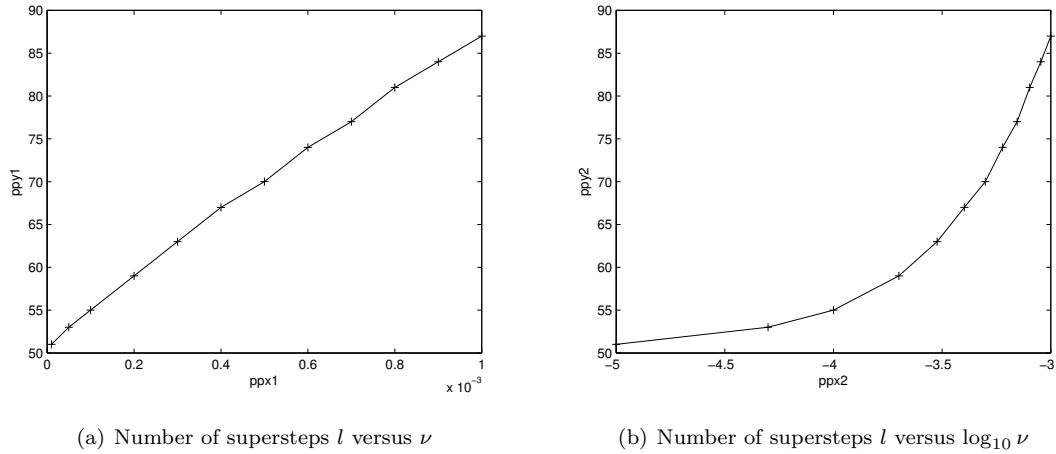
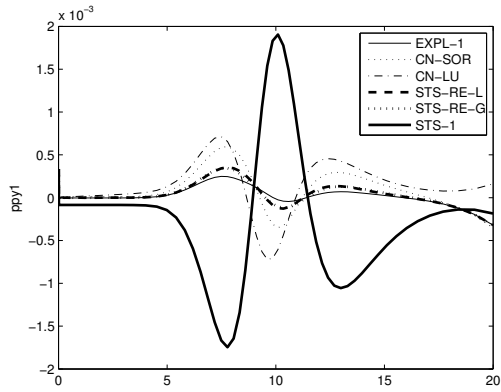
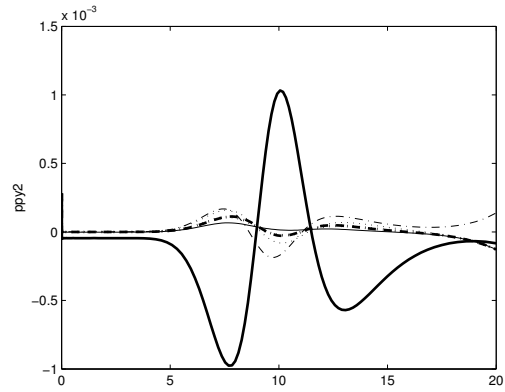


Fig. 5. Number of supersteps as a function of ν and $\log_{10} \nu$.

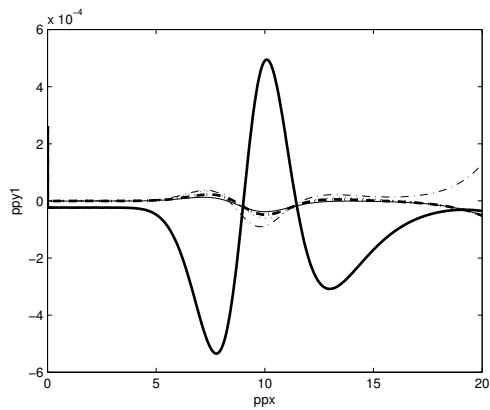
This figure depicts the number of supersteps as a function of ν and as a function of the logarithm of ν (log base 10) with N_{sts} fixed at 25 on a grid of size $(m, n) = (512, 256)$. In this example the minimum number of standard explicit steps required for stability is 31,453. The minimum number of supersteps possible is equal to $\frac{l_{expl}}{N_{sts}^2} \approx 51$. The total number of steps taken in the STS scheme using minimal l is equal to $l \times N_{sts} = 51 \times 25 = 1,275$. This is approximately a factor of 24.7 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 $\nu = 0.0001$ which results in 130 supersteps for the grid of size $(m, n) = (512, 256)$. This represents a speed-up by a factor of approximately 9.7 relative to the explicit scheme. This illustrates that the numerical experiments conducted in this paper were well within the limits of stability.



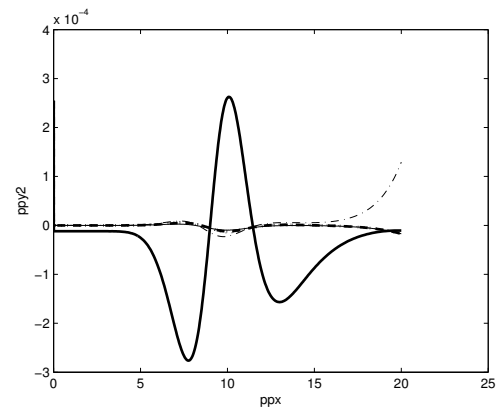
(a) Error for $(m, n) = (64, 32)$



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



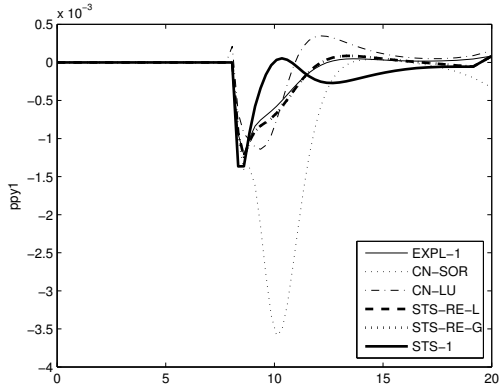
(c) Error for $(m, n) = (256, 128)$



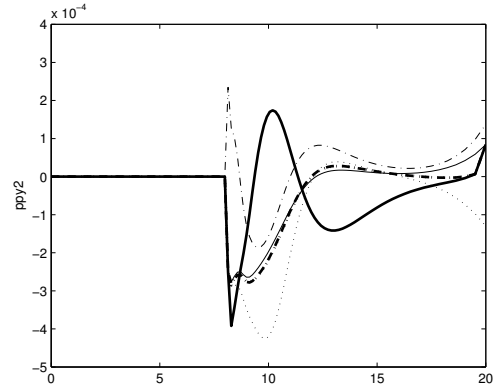
(d) Error for $(m, n) = (512, 256)$

Fig. 6. Error in European put prices versus the stock price for $y = 0.0625$.

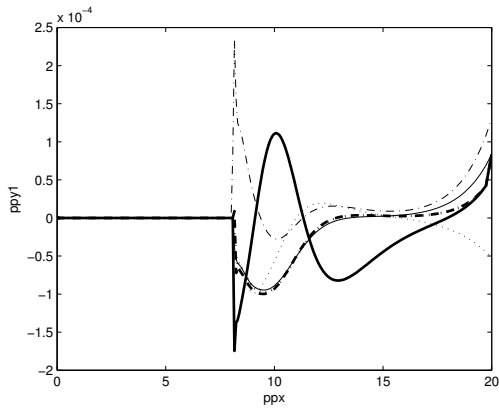
This figure depicts the error in European put option prices as a function of the stock price using various FD schemes. The error is given by $u_{fd} - u_{ref}$ where u_{fd} and u_{ref} denote the FD prices and the reference CN-SOR prices respectively. EXPL-1 is the explicit scheme, CN-SOR is the CN successive overrelaxation scheme, CN-LU is the CN LU decomposition 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.



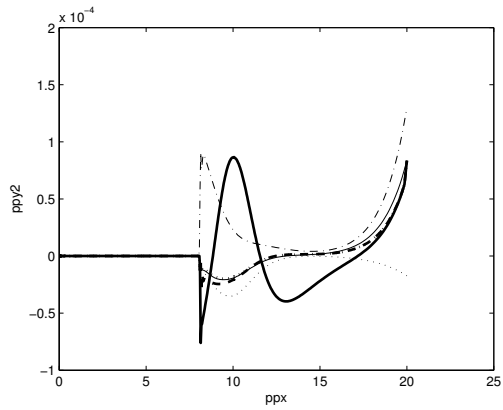
(a) Error for $(m, n) = (64, 32)$



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



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



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

Fig. 7. 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-PSOR prices respectively. EXPL-1 is the explicit scheme, CN-PSOR is the CN projected successive overrelaxation scheme, CN-PLU is the CN projected LU decomposition 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.

Table 1. European put prices evaluated using stock prices $x = 8, 9, \dots, 12$ and an initial variance value of $y = 0.0625$ with STS parameters of $\nu = 0.001$ and $N_{sts} = 25$.

Method	Grid (m, n, l)	x				
		8	9	10	11	12
Reference (FFT)		1.838868	1.048347	0.501466	0.208187	0.080429
Reference (CN-SOR)	(2048, 1024, 2050)	1.838868	1.048347	0.501465	0.208187	0.080428
STS-1	(64, 32, 18)	1.837184	1.048494	0.503367	0.208989	0.079770
	(128, 64, 34)	1.837928	1.048394	0.502491	0.208642	0.080088
	(256, 128, 66)	1.838351	1.048332	0.501956	0.208392	0.080231
	(512, 256, 130)	1.838602	1.048344	0.501726	0.208298	0.080330
STS-RE-L	(64, 32, 18)	1.839215	1.048509	0.501356	0.208138	0.080530
	(128, 64, 34)	1.838979	1.048391	0.501439	0.208188	0.080472
	(256, 128, 66)	1.838886	1.048328	0.501417	0.208158	0.080425
	(512, 256, 130)	1.838872	1.048342	0.501453	0.208179	0.080427
STS-RE-G	(64, 32, 18)	1.839197	1.048481	0.501372	0.208121	0.080507
	(128, 64, 34)	1.838974	1.048383	0.501443	0.208184	0.080466
	(256, 128, 66)	1.838885	1.048326	0.501418	0.208157	0.080423
	(512, 256, 130)	1.838872	1.048341	0.501453	0.208179	0.080427
EXPL-1	(64, 32, 1×10^3)	1.839102	1.048481	0.501452	0.208157	0.080469
	(128, 64, 2×10^3)	1.838932	1.048384	0.501480	0.208201	0.080450
	(256, 128, 8×10^3)	1.838874	1.048326	0.501428	0.208161	0.080419
	(512, 256, 32×10^3)	1.838869	1.048341	0.501456	0.208180	0.080426
CN-LU	(64, 32, 18)	1.839425	1.048000	0.500811	0.208246	0.080858
	(128, 64, 34)	1.839000	1.048254	0.501291	0.208175	0.080528
	(256, 128, 66)	1.838890	1.048292	0.501377	0.208150	0.080437
	(512, 256, 130)	1.838873	1.048333	0.501443	0.208178	0.080430
CN-SOR	Grid ($m, n, l, iter_{av}, w$)					
	(64, 32, 18, 17.17, 1.59)	1.839444	1.048513	0.501125	0.208092	0.080657
	(128, 64, 34, 20.29, 1.75)	1.839017	1.048374	0.501386	0.208155	0.080475
	(256, 128, 66, 24.61, 1.84)	1.838895	1.048323	0.501404	0.208149	0.080425
	(512, 256, 130, 31.59, 1.87)	1.838874	1.048341	0.501450	0.208177	0.080427

Table 2. The (l^2 -norm) errors calculated using ten European put prices at stock prices of $x = 8, 9, \dots, 12$, and initial variance values of $y = 0.0625$ and 0.25 . Also reported are the the ratio of consecutive errors, and the run times in seconds for codes implemented in MATLAB. The run times for C implementations of the schemes are shown in parentheses (with $iter_{av}$, w in the case of the SOR scheme).

Method	Grid (m, n, l)	Error	Ratio	Run Times
STS-1	(64, 32, 18)	0.004797		0.05 (0.01)
	(128, 64, 34)	0.003041	1.58	0.17 (0.04)
	(256, 128, 66)	0.001591	1.91	0.97 (0.31)
	(512, 256, 130)	0.000827	1.92	11.33 (3.30)
STS-RE-L	(64, 32, 18)	0.001584		0.15 (0.02)
	(128, 64, 34)	0.000233	6.80	0.50 (0.11)
	(256, 128, 66)	0.000079	2.94	3.62 (0.87)
	(512, 256, 130)	0.000017	4.73	33.77 (10.17)
STS-RE-G	(64, 32, 18)	0.001512		0.18 (0.02)
	(128, 64, 34)	0.000212	7.12	0.58 (0.11)
	(256, 128, 66)	0.000076	2.78	4.81 (0.92)
	(512, 256, 130)	0.000016	4.66	36.48 (10.04)
EXPL-1	(64, 32, 1×10^3)	0.001274		0.11 (0.02)
	(128, 64, 2×10^3)	0.000113	11.23	0.38 (0.10)
	(256, 128, 8×10^3)	0.000055	2.04	6.08 (1.49)
	(512, 256, 32×10^3)	0.000014	4.06	109.94 (33.30)
CN-LU	(64, 32, 18)	0.001915		0.82 (0.004)
	(128, 64, 34)	0.000343	5.58	3.09 (0.03)
	(256, 128, 66)	0.000126	2.73	13.33 (0.27)
	(512, 256, 130)	0.000030	4.15	64.19 (2.98)
CN-SOR	(64, 32, 18)	0.002086		0.23 (0.01 7.05 1.44)
	(128, 64, 34)	0.000347	6.01	1.34 (0.08 12.89 1.58)
	(256, 128, 66)	0.000110	3.16	9.40 (1.14 23.54 1.68)
	(512, 256, 130)	0.000024	4.64	70.51 (16.15 41.45 1.77)

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

Method	Grid (m, n, l)	x				
		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)	1.999921	1.106901	0.520042	0.213628	0.081801
	(128, 64, 130)	1.999984	1.107469	0.520194	0.213745	0.081947
	(256, 128, 258)	2.000000	1.107598	0.520140	0.213714	0.081985
	(512, 256, 514)	2.000000	1.107636	0.520115	0.213714	0.082021
STS-RE-L	(64, 32, 66)	1.999872	1.106672	0.519336	0.213349	0.082015
	(128, 64, 130)	1.999984	1.107342	0.519812	0.213584	0.082044
	(256, 128, 260)	2.000002	1.107525	0.519936	0.213625	0.082030
	(512, 256, 514)	1.999999	1.107595	0.520008	0.213665	0.082041
STS-RE-G	(64, 32, 66)	1.999874	1.106666	0.519331	0.213343	0.082011
	(128, 64, 130)	1.999984	1.107343	0.519810	0.213582	0.082043
	(256, 128, 258)	2.000001	1.107529	0.519937	0.213624	0.082029
	(512, 256, 514)	1.999999	1.107600	0.520010	0.213666	0.082041
EXPL-1	(64, 32, 1×10^3)	1.999875	1.106723	0.519428	0.213380	0.081985
	(128, 64, 2×10^3)	1.999984	1.107374	0.519863	0.213604	0.082032
	(256, 128, 8×10^3)	2.000001	1.107535	0.519948	0.213628	0.082025
	(512, 256, 32×10^3)	1.999999	1.107601	0.520012	0.213666	0.082039
CN-PLU	(64, 32, 66)	2.000044	1.106537	0.519185	0.213711	0.082387
	(128, 64, 130)	1.999997	1.107487	0.519872	0.213675	0.082121
	(256, 128, 258)	2.000000	1.107654	0.520002	0.213667	0.082055
	(512, 256, 514)	2.000000	1.107675	0.520052	0.213688	0.082052
CN-PSOR	Grid $(m, n, l, \text{iter-avg}, w)$					
	(64, 32, 66, 10.26, 1.60)	1.999880	1.106140	0.516512	0.211312	0.081334
	(128, 64, 130, 14.16, 1.75)	1.999984	1.107239	0.519397	0.213218	0.081876
	(256, 128, 258, 17.84, 1.84)	2.000001	1.107499	0.519826	0.213520	0.081974
	(512, 256, 514, 20.16, 1.87)	1.999999	1.107589	0.519984	0.213644	0.082030

Table 4. The (l^2 -norm) errors calculated using ten American put prices at stock prices of $x = 8, 9, \dots, 12$, and initial variance values of $y = 0.0625$ and 0.25 . Also reported are the the ratio of consecutive errors, and the run times in seconds for codes implemented in MATLAB. The run times for C implementations of the schemes are shown in parentheses (with $iter_{av}$, w in the case of the SOR scheme).

Method	Grid (m, n, l)	Error	Ratio	Run Times
STS-1	(64, 32, 66)	0.001350		0.12 (0.01)
	(128, 64, 130)	0.000588	2.30	0.39 (0.09)
	(256, 128, 260)	0.000391	1.50	2.24 (0.72)
	(512, 256, 514)	0.000246	1.59	27.02 (8.08)
STS-RE-L	(64, 32, 66)	0.002780		0.37 (0.04)
	(128, 64, 130)	0.000588	4.73	1.19 (0.27)
	(256, 128, 258)	0.000188	3.12	7.66 (2.32)
	(512, 256, 514)	0.000045	4.22	84.30 (25.38)
STS-RE-G	(64, 32, 66)	0.002788		0.37 (0.04)
	(128, 64, 130)	0.000585	4.76	1.16 (0.27)
	(256, 128, 258)	0.000178	3.29	7.10 (2.20)
	(512, 256, 514)	0.000033	5.42	81.34 (24.68)
EXPL-1	(64, 32, 1×10^3)	0.002525		0.19 (0.02)
	(128, 64, 2×10^3)	0.000457	5.53	0.58 (0.14)
	(256, 128, 8×10^3)	0.000153	2.98	8.10 (2.31)
	(512, 256, 32×10^3)	0.000031	4.99	144.60 (47.65)
CN-PLU	(64, 32, 66)	0.002381		2.77 (0.02)
	(128, 64, 130)	0.000271	8.79	11.53 (0.17)
	(256, 128, 258)	0.000139	1.95	50.85 (1.35)
	(512, 256, 514)	0.000125	1.11	251.88 (14.00)
CN-PSOR	(64, 32, 66)	0.005354		0.86 (0.01 3.15 1.22)
	(128, 64, 130)	0.001066	5.02	4.93 (0.13 5.61 1.35)
	(256, 128, 258)	0.000334	3.19	29.42 (1.96 11.03 1.47)
	(512, 256, 514)	0.000076	4.37	182.13 (27.15 18.32 1.59)

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.

Reference		x					
		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
	[21]	0.0625	2.000000	1.107629	0.520038	0.213681	0.082046
		0.25	2.078372	1.333640	0.795983	0.448277	0.242813
	[8]	0.0625	2.0000	1.1080	0.5316	0.2261	0.0907
		0.25	2.0733	1.3290	0.7992	0.4536	0.2502
	[49]	0.0625	2.0000	1.1076	0.5202	0.2138	0.0821
		0.25	2.0784	1.3337	0.7961	0.4483	0.2428
	[30]	0.0625	2.000	1.107	0.517	0.212	0.0815
		0.25	2.079	1.334	0.796	0.449	0.243

Table 6. Convergence of European put prices versus analytical solution of Heston. Parameters: $E = 100$, $r = 0.05$, $\gamma = 0.01$, $\alpha = 3$, $\beta = 0.04$, and $\rho = -0.7$. For a given number of spatial steps m , $n = \frac{m}{2}$ and $l = \frac{m}{4} + 2$. The STS parameters are $N_{sts} = 25$ and $\nu = 0.001$ when $T = 0.0833$, $N_{sts} = 30$ and $\nu = 0.0006$ when $T = 0.25$ and $N_{sts} = 35$ and $\nu = 0.0003$ when $T = 0.5$.

x	y	T	STS-RE-G			FFT solution	Error %		
			$m = 128$	$m = 254$	$m = 512$		$m = 128$	$m = 254$	$m = 512$
90	0.04	0.0833	9.655248	9.653773	9.653443	9.653325	0.019921	0.004639	0.001215
95	0.04	0.0833	5.206435	5.207229	5.207338	5.207351	-0.017595	-0.002347	-0.000241
100	0.04	0.0833	2.094308	2.096436	2.096933	2.097064	-0.131443	-0.029954	-0.006259
105	0.04	0.0833	0.604844	0.605187	0.605291	0.605300	-0.075244	-0.018580	-0.001502
110	0.04	0.0833	0.127056	0.126702	0.126586	0.126538	0.409478	0.129959	0.038288
90	0.09	0.0833	9.991716	9.990898	9.990553	9.990508	0.012095	0.003901	0.000453
95	0.09	0.0833	6.012879	6.014981	6.015478	6.015535	-0.044151	-0.009208	-0.000950
100	0.09	0.0833	3.125651	3.129033	3.130110	3.130211	-0.145683	-0.037643	-0.003231
105	0.09	0.0833	1.394354	1.395989	1.396666	1.396709	-0.168641	-0.051533	-0.003105
110	0.09	0.0833	0.537258	0.536843	0.536736	0.536712	0.101657	0.024401	0.004379
90	0.16	0.0833	10.566300	10.566655	10.566812	10.566820	-0.004916	-0.001562	-0.000073
95	0.16	0.0833	6.930261	6.932232	6.933392	6.933456	-0.046085	-0.017659	-0.000921
100	0.16	0.0833	4.180959	4.183520	4.185075	4.185155	-0.100260	-0.039081	-0.001914
105	0.16	0.0833	2.319261	2.321053	2.322144	2.322195	-0.126319	-0.049148	-0.002185
110	0.16	0.0833	1.187492	1.187961	1.188209	1.188216	-0.060924	-0.021414	-0.000529
90	0.04	0.25	9.568362	9.569655	9.569749	9.569752	-0.014526	-0.001012	-0.000025
95	0.04	0.25	5.965771	5.968806	5.969099	5.969191	-0.057279	-0.006440	-0.001536
100	0.04	0.25	3.375074	3.376676	3.376922	3.377001	-0.057059	-0.009629	-0.002320
105	0.04	0.25	1.740750	1.740859	1.740991	1.741018	-0.015381	-0.009117	-0.001538
110	0.04	0.25	0.826157	0.825904	0.825947	0.825944	0.025842	-0.004860	0.000375
90	0.09	0.25	10.585299	10.588851	10.589306	10.589348	-0.038243	-0.004700	-0.000397
95	0.09	0.25	7.322196	7.330208	7.331539	7.331628	-0.128641	-0.019365	-0.001215
100	0.09	0.25	4.820075	4.829095	4.830881	4.830971	-0.225545	-0.038832	-0.001855
105	0.09	0.25	3.030404	3.037092	3.038697	3.038754	-0.274793	-0.054706	-0.001875
110	0.09	0.25	1.828656	1.831585	1.832510	1.832528	-0.211292	-0.051507	-0.000983
90	0.16	0.25	11.821484	11.825471	11.828632	11.828688	-0.060906	-0.027202	-0.000478
95	0.16	0.25	8.792611	8.798588	8.803449	8.803526	-0.123978	-0.056086	-0.000871
100	0.16	0.25	6.361643	6.368078	6.373400	6.373476	-0.185666	-0.084688	-0.001197
105	0.16	0.25	4.487568	4.493018	4.497581	4.497641	-0.223954	-0.102789	-0.001341
110	0.16	0.25	3.094478	3.098059	3.101081	3.101118	-0.214138	-0.098645	-0.001207
90	0.04	0.5	9.752382	9.756872	9.757127	9.757185	-0.049225	-0.003205	-0.000587
95	0.04	0.5	6.715471	6.719646	6.719801	6.719880	-0.065620	-0.003483	-0.001178
100	0.04	0.5	4.428801	4.431038	4.431119	4.431181	-0.053725	-0.003220	-0.001407
105	0.04	0.5	2.810043	2.810586	2.810656	2.810692	-0.023107	-0.003792	-0.001273
110	0.04	0.5	1.724230	1.723919	1.723997	1.724013	0.012551	-0.005497	-0.000972
90	0.09	0.5	11.070181	11.079542	11.080573	11.080652	-0.094497	-0.010016	-0.000706
95	0.09	0.5	8.220840	8.234499	8.236168	8.236268	-0.187323	-0.021484	-0.001217
100	0.09	0.5	5.959693	5.974240	5.976189	5.976281	-0.277568	-0.034152	-0.001550
105	0.09	0.5	4.229980	4.242325	4.244187	4.244254	-0.336312	-0.045439	-0.001574
110	0.09	0.5	2.948419	2.956688	2.958192	2.958229	-0.331618	-0.052075	-0.001234
90	0.16	0.5	12.600626	12.610309	12.617017	12.617096	-0.130535	-0.053796	-0.000628
95	0.16	0.5	9.901845	9.913698	9.922244	9.922334	-0.206491	-0.087035	-0.000906
100	0.16	0.5	7.675352	7.687440	7.696449	7.696535	-0.275224	-0.118158	-0.001117
105	0.16	0.5	5.878813	5.889482	5.897678	5.897750	-0.321095	-0.140183	-0.001218
110	0.16	0.5	4.456918	4.465086	4.471552	4.471605	-0.328447	-0.145773	-0.001182

Table 7. Convergence of American put prices versus benchmark PSOR American put prices. Parameters: $E = 100$, $r = 0.05$, $\gamma = 0.01$, $\alpha = 3$, $\beta = 0.04$, and $\rho = -0.7$. For a given number of spatial steps m , $n = \frac{m}{2}$ and $l = m + 2$. The STS parameters are $N_{sts} = 15$ and $\nu = 0.002$ when $T = 0.0833$ and $T = 0.25$, and $N_{sts} = 25$ and $\nu = 0.001$ when $T = 0.5$.

x	y	T	STS-RE-G			PSOR benchmark	Error %		
			$m = 128$	$m = 254$	$m = 512$		$m = 128$	$m = 254$	$m = 512$
90	0.04	0.0833	10.000159	9.999994	10.000000	10.000000	0.001586	-0.000057	0.000002
95	0.04	0.0833	5.322143	5.324939	5.325618	5.325781	-0.068307	-0.015809	-0.003050
100	0.04	0.0833	2.122503	2.125806	2.126623	2.126811	-0.202537	-0.047228	-0.008819
105	0.04	0.0833	0.610137	0.610812	0.611020	0.611051	-0.149621	-0.039097	-0.005170
110	0.04	0.0833	0.127813	0.127533	0.127442	0.127402	0.322032	0.101892	0.030569
90	0.09	0.0833	10.164952	10.165177	10.165012	10.165053	-0.000997	0.001218	-0.000401
95	0.09	0.0833	6.088480	6.091362	6.092038	6.092144	-0.060138	-0.012832	-0.001741
100	0.09	0.0833	3.154605	3.158548	3.159786	3.159902	-0.167642	-0.042864	-0.003669
105	0.09	0.0833	1.404123	1.406051	1.406825	1.406877	-0.195731	-0.058740	-0.003710
110	0.09	0.0833	0.540182	0.539886	0.539823	0.539814	0.068141	0.013424	0.001737
90	0.16	0.0833	10.680890	10.681852	10.682066	10.682113	-0.011448	-0.002446	-0.000442
95	0.16	0.0833	6.989753	6.992251	6.993482	6.993559	-0.054429	-0.018703	-0.001108
100	0.16	0.0833	4.209737	4.212704	4.214339	4.214416	-0.111007	-0.040610	-0.001829
105	0.16	0.0833	2.332273	2.334329	2.335482	2.335527	-0.139313	-0.051286	-0.001945
110	0.16	0.0833	1.193015	1.193633	1.193919	1.193926	-0.076342	-0.024529	-0.000577
90	0.04	0.25	10.121287	10.122365	10.122803	10.122932	-0.016247	-0.005598	-0.001273
95	0.04	0.25	6.210118	6.215412	6.216273	6.216466	-0.102110	-0.016954	-0.003098
100	0.04	0.25	3.477684	3.480630	3.481205	3.481323	-0.104547	-0.019922	-0.003391
105	0.04	0.25	1.781571	1.782295	1.782596	1.782637	-0.059830	-0.019177	-0.002335
110	0.04	0.25	0.841591	0.841603	0.841728	0.841731	-0.016658	-0.015173	-0.000410
90	0.09	0.25	10.954471	10.957124	10.957261	10.957328	-0.026071	-0.001858	-0.000615
95	0.09	0.25	7.532389	7.540086	7.541313	7.541403	-0.119533	-0.017462	-0.001200
100	0.09	0.25	4.935069	4.944235	4.946053	4.946130	-0.223635	-0.038307	-0.001551
105	0.09	0.25	3.091239	3.098209	3.099886	3.099928	-0.280288	-0.055432	-0.001332
110	0.09	0.25	1.859958	1.863124	1.864113	1.864119	-0.223207	-0.053410	-0.000342
90	0.16	0.25	12.113858	12.117522	12.119984	12.120029	-0.050920	-0.020681	-0.000373
95	0.16	0.25	8.981922	8.987820	8.992250	8.992302	-0.115432	-0.049839	-0.000580
100	0.16	0.25	6.481603	6.488135	6.493254	6.493299	-0.180122	-0.079522	-0.000695
105	0.16	0.25	4.562212	4.567829	4.572326	4.572355	-0.221830	-0.098990	-0.000631
110	0.16	0.25	3.140229	3.143974	3.146992	3.147002	-0.215226	-0.096230	-0.000316
90	0.04	0.5	10.559084	10.565530	10.566484	10.566684	-0.071922	-0.010918	-0.001892
95	0.04	0.5	7.150543	7.157001	7.157677	7.157826	-0.101743	-0.011516	-0.002074
100	0.04	0.5	4.660408	4.664079	4.664455	4.664531	-0.088382	-0.009683	-0.001622
105	0.04	0.5	2.931602	2.932948	2.933187	2.933212	-0.054881	-0.009002	-0.000862
110	0.04	0.5	1.787218	1.787347	1.787522	1.787521	-0.016997	-0.009765	0.000030
90	0.09	0.5	11.758555	11.765023	11.765729	11.765798	-0.061559	-0.006588	-0.000588
95	0.09	0.5	8.664867	8.677192	8.678778	8.678854	-0.161166	-0.019158	-0.000876
100	0.09	0.5	6.240929	6.255178	6.257198	6.257257	-0.260949	-0.033229	-0.000956
105	0.09	0.5	4.405683	4.418195	4.420181	4.420213	-0.328714	-0.045665	-0.000729
110	0.09	0.5	3.057096	3.065630	3.067257	3.067262	-0.331450	-0.053219	-0.000167
90	0.16	0.5	13.219887	13.227923	13.232897	13.232940	-0.098639	-0.037911	-0.000325
95	0.16	0.5	10.343800	10.354643	10.362094	10.362140	-0.176995	-0.072353	-0.000443
100	0.16	0.5	7.987239	7.998831	8.007258	8.007296	-0.250477	-0.105709	-0.000478
105	0.16	0.5	6.097026	6.107525	6.115470	6.115495	-0.301997	-0.130326	-0.000400
110	0.16	0.5	4.608622	4.616785	4.623179	4.623187	-0.315041	-0.138483	-0.000188