To appear in the *International Journal of Computer Mathematics*
Vol. 00, No. 00, Month 20XX, 1–16

Adaptive finite differences and IMEX time-stepping to price options under Bates model

L. von Sydow[a]*, J. Toivanen[b,c], C. Zhang[a]

[a]Department of Information Technology, Uppsala University, Sweden; [b]Institute for Computational and Mathematical Engineering (ICME), Stanford University, USA; [c]Department of Mathematical Information Technology, University of Jyväskylä, Finland

(*Received 00 Month 20XX; accepted 00 Month 20XX*)

In this paper we consider numerical pricing of European and American options under the Bates model, a model which gives rise to a partial-integro differential equation. This equation is discretized in space using adaptive finite differences while an IMEX scheme is employed in time. The sparse linear systems of equations in each time-step are solved using an LU-decomposition and an operator splitting technique is employed for the linear complementarity problems arising for American options. The integral part of the equation is treated explicitly in time which means that we have to perform matrix-vector multiplications each time-step with a matrix with dense blocks. These multiplications are accomplished through fast Fourier transforms. The great performance of the method is demonstrated through numerical experiments.

Keywords: Option pricing; numerical methods; Bates model, adaptive finite differences, IMEX time-stepping

2010 AMS Subject Classification: 65M06, 91G60

1. Introduction

The model for the underlying asset in the seminal paper [5] by Black and Scholes in 1973 on option pricing is a geometrical Brownian motion. A few years later in [20] Merton adds log-normally distributed jumps to this model as empirical studies on stock price series suggest existence of jumps. In the end of 1980s it was widely recognized that the constant volatility assumption of the Black–Scholes model is unrealistic for many underlying assets and several models with stochastic volatility were proposed. Probably the most popular among these models is the one proposed by Heston in [11]. Bates combined the Merton jump-diffusion model and the Heston stochastic volatility model in his 1996 paper [4]. As this model by Bates is reasonably realistic for many underlying assets we consider option pricing based on it in this paper.

Under the Bates model a parabolic partial-integro differential equation (PIDE) can be derived for the prices of European options. The variables of this PIDE are the asset value and its variance. The underlying partial differential operator is of convection-diffusion type while the integral operator integrates over all values in the asset value direction. For the price of American options, a linear complementarity problem (LCP) can be formulated with the same underlying partial-integro differential operator. We consider

*Corresponding author. Email: lina@it.uu.se pricing European and American options by employing finite difference discretizations for differential operators and a suitable quadrature for the integral operator. The most commonly used finite differences lead to a sparse block tridiagonal matrices while the quadrature leads to block diagonal matrices with full diagonal blocks.

The efficient discretization and solution techniques derived for these option pricing problems is discussed in the following. Implicit time discretizations require the solution of a system with a coefficient matrix having full diagonal blocks at each time step. The iterative methods considered in [33], [27] can solve these problems reasonably efficiently. Alternatively the ADI-type operator splitting method in [3] gives an efficent approximate solution procedure. Still avoiding an implicit treatment of the integral operator can lead to a simpler and more efficient solution procedure. Cont and Voltchkova consider in [8] the IMEX-Euler method which treats the integral operator explicitly and the differential operator implicitly. This method is only first-order accurate in time. The IMEX-midpoint [18], [17] and IMEX-CNAB [28], [30] methods are similar, but second-order accurate methods in time. In this paper, we employ the IMEX-CNAB method. The explicit treatment of the integral operator leads to multiplications by a matrix with full blocks. These can be performed efficiently using fast Fourier transforms (FFT); see [2], [10], for example.

The IMEX time discretizations require the solution of a two-dimensional convection-diffusion-reaction type problem at each time step. Several different efficient solution and approximation methods have been proposed for these problems. These include multigrid methods [7], [22], preconditioned iterations [40], [25], and directional operator splitting methods [13], [16]. Here we employ a direct solver based on LU decomposition. Recently this approach was shown to be efficient and convenient for these problems in [30]. For American options an LCP with the same operator needs to be solved at each time step. The LCP can be approximated and solved using various methods including an operator splitting method [12], [15], [30], penalty methods [39], [9], and multigrid methods [7], [22], [26], [14], [35]. Here we use the operator splitting method to approximate the solutions of LCPs as it is accurate and easy to implement. Futhermore, it leads to systems of linear equations which can be solved using standard efficient solution methods.

Finite difference discretizations have been used dominantly for the Bates model; see [6], [34], [29], [30]. In [21], a finite element discretization was used. The main topic of this paper is the choice of a good nonuniform finite difference grid. The grid should be as coarse as possible to save computational time, but fine enough to reach the desired accuracy. Here such a grid is constructed adaptively. We employ the approach described in [23], [24] for a multi-dimensional Black–Scholes model and the Heston model. The basic idea is to estimate the spatial discretization error on a coarse grid based on the order of convergence of the discretization and then by employing this error estimate construct a fine grid. As the grid used for error estimation can be fairly coarse the computional cost of the error estimation is small while the obtained fine grid is nearly optimal. An alternative, more elaborate approach would be to use a goal oriented adjoint equation based method described in [19]. For more discussion on adaptive discretizations see the book [1].

The outline of the paper is the following. We begin by describing the Bates model and the resulting PIDE and LCP for European and American options in Section 2. For a given grid the spatial discretization is constructed in Section 3. The temporal IMEX discretization and the use of FFT to evaluate the integrals in each time-step is described in Section 4. The adaptive construction of spatial grids is introduced in Section 5. The operator splitting method for American options is given in Section 6. Numerical results and conclusions are presented in Sections 7 and 8, respectively. Acknowledgments end the paper.

2. Mathematical model

Under the Bates model [4] the asset value S_t and the instantaneous variance V_t satisfy the stochastic differential equations

$$\begin{aligned} dS_t &= (r - q - \lambda\xi)S_t dt + \sqrt{V_t} S_t dW_t^1 + (\mathcal{J} - 1)S_t dN, \\ dV_t &= \kappa(\theta - V_t)dt + \sigma\sqrt{V_t} dW_t^2, \end{aligned} \tag{1}$$

where r is the risk free interest rate, q is the dividend yield, λ is the intensity of the Poisson process N, the jump $\mathcal{J}$ has a log-normal distribution

$$f(\mathcal{J}) = \frac{1}{\sqrt{2\pi}\delta\mathcal{J}} \exp\left(-\frac{\left(\ln\mathcal{J} - (\gamma - \delta^2/2)\right)^2}{2\delta^2}\right),$$

where γ and δ define the mean and variance of the jump and ξ is given by $\xi = \exp(\gamma) - 1$. The mean-reversion level of the variance is θ and κ is the rate of reversion to this mean level and finally W_1 and W_2 are Wiener processes with the correlation ρ.

The price u of a European option issued on an asset following the process defined in (1) can be computed by solving the following PIDE

$$\begin{aligned} \frac{\partial u(s,v,\tau)}{\partial\tau} &= \frac{1}{2}vs^2\frac{\partial^2 u(s,v,\tau)}{\partial s^2} + \frac{1}{2}v\sigma^2\frac{\partial^2 u(s,v,\tau)}{\partial v^2} + \rho\sigma vs\frac{\partial^2 u(s,v,\tau)}{\partial s\partial v} \\ &+ (r - q - \lambda\xi)\frac{\partial u(s,v,\tau)}{\partial s} + \kappa(\theta - v)\frac{\partial u(s,v,\tau)}{\partial v} \\ &- (r + \lambda)u(s,v,\tau) + \lambda\int_0^\infty u(\mathcal{J}s,v,\tau)f(\mathcal{J})d\mathcal{J} = \mathcal{L}u, \end{aligned} \tag{2}$$

where $\tau = T - t$ is the time to expiry. The initial condition for (2) is defined by the payoff function Φ for the option, which for a European call option is

$$u(s,v,0) = \Phi(s) = \max(s - K, 0). \tag{3}$$

For the American options we must take into account the possibility for early exercise which leads to the following LCP:

$$\begin{aligned} &\tfrac{\partial u}{\partial\tau} + \mathcal{L}u \geq 0, \\ &u \geq \Phi, \\ &(u - \Phi)\left(\tfrac{\partial u}{\partial\tau} + \mathcal{L}u\right) = 0, \end{aligned} \tag{4}$$

with the initial condition (3) for an American call option. We will denote the free boundary by s_f that separates the stopping region where $u = \Phi$ and the continuation region defined by $\frac{\partial u}{\partial\tau} + \mathcal{L}u = 0$.

3. Spatial discretization

We will discretize the spatial operator $\mathcal{L}u$ on a structured but nonequidistant grid (s_i, v_j), $0 \le s_i \le s_{\max}$, $0 \le v_j \le v_{\max}$ with m_s grid-points in s and m_v grid-points in v. The boundary conditions used for the European call option are

$$\begin{cases} u(0, v, \tau) = 0, \\ u(s_{\max}, v, \tau) = s_{\max} e^{-q\tau} - K e^{-r\tau}, \\ \dfrac{\partial u(s, v_{\max}, \tau)}{\partial v} = 0, \end{cases}$$

and for the American call option

$$\begin{cases} u(0, v, \tau) = 0, \\ u(s_{\max}, v, \tau) = \max(s_{\max} e^{-q\tau} - K e^{-r\tau}, s_{\max} - K), \\ \dfrac{\partial u(s, v_{\max}, \tau)}{\partial v} = 0. \end{cases}$$

For the parameter settings we consider we have $u(s_{\max}, v, \tau) = s_{\max} - K$ for the American option.

We discretize the derivatives in (2) using centered, second-order finite differences on a nonequidistant grid, see [23], [24]. For the derivatives in the v-directions that do not vanish at $v = 0$, we use one-sided first-order finite differences there. The discrete approximation of the integral term in (2) is evaluated by first making a transformation from the original computational grid s_i to an equidistant grid x_k. Let

$$I = \int_0^\infty u(Js, v, \tau) f(J) dJ = \int_{-\infty}^\infty \bar{u}(x + z, v, \tau) \bar{f}(z) dz,$$

where $x = \log s$, $z = \log \mathcal{J}$, $\bar{u}(z, v, \tau) = u(e^z, v, \tau)$ and $\bar{f}(z) = e^z f(e^z)$. Then we define a new variable $\zeta = z + x$ and obtain at $x = x_k$ the integral

$$\begin{aligned} I_k = \int_{-\infty}^{\infty} \bar{u}(\zeta, v, \tau) \bar{f}(\zeta - x_k) d\zeta &= \int_{x_{\min}}^{x_{\max}} \bar{u}(\zeta, v, \tau) \bar{f}(\zeta - x_k) d\zeta \\ + \int_{-\infty}^{x_{\min}} \bar{u}(\zeta, v, \tau) \bar{f}(\zeta - x_k) d\zeta + \int_{x_{\max}}^{\infty} \bar{u}(\zeta, v, \tau) \bar{f}(\zeta - x_k) d\zeta &= I_k^{(1)} + I_k^{(2)} + I_k^{(3)}. \end{aligned} \tag{5}$$

We compute the first part of (5) using the trapezoidal quadrature rule on an equidistant grid in x with spacing Δx and m_x grid-points in $[x_{\min}, x_{\max}]$ giving

$$\begin{aligned} I_k^{(1)} &= \textstyle\int_{x_{\min}}^{x_{\max}} \bar{u}(\zeta, v, \tau) \bar{f}(\zeta - x_k) d\zeta \approx \Delta x \sum_{j=1}^{m_x} \bar{u}(\zeta_j, v, \tau) \bar{f}(\zeta_j - x_k) \\ &- \tfrac{\Delta x}{2} \left(\bar{u}(x_{\min}, v, \tau) \bar{f}(x_{\min} - x_k) + \bar{u}(x_{\max}, v, \tau) \bar{f}(x_{\max} - x_k) \right). \end{aligned}$$

The second part $I_k^{(2)}$ can be approximated by

$$I_k^{(2)} = \int_{-\infty}^{x_{\min}} \bar{u}(\zeta, v, \tau)\bar{f}(\zeta - x_k)d\zeta \approx \bar{u}(x_{\min})\frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{x_{\min} - x_k - (\gamma - \delta^2/2)}{\sqrt{2}\delta}\right)\right)$$

Finally we compute $I_k^{(3)}$ where we use $u(s, v, \tau) \approx se^{-q\tau} - Ke^{-r\tau}$, $s \geq s_{\max}$ for European options and $u(s, v, \tau) \approx s - K$, $s \geq s_{\max}$ for American options. For European options we obtain

$$\begin{aligned} I_k^{(3)} = \int_{x_{\max}}^{\infty} \bar{u}(\zeta, v, \tau)\bar{f}(\zeta - x_k)d\zeta &\approx \frac{s_k}{2}e^{\gamma - q\tau}\left(1 - \operatorname{erf}\left(\frac{x_{\max} - x_k - \gamma - \delta^2/2}{\sqrt{2}\delta}\right)\right) \\ &- \frac{K}{2}e^{-r\tau}\left(1 - \operatorname{erf}\left(\frac{x_{\max} - x_k - (\gamma - \delta^2/2)}{\sqrt{2}\delta}\right)\right). \end{aligned}$$

and similarly for American options we obtain

$$\begin{aligned} I_k^{(3)} = \int_{x_{\max}}^{\infty} \bar{u}(\zeta, v, \tau)\bar{f}(\zeta - x_k)d\zeta &\approx \frac{s_k}{2}e^{\gamma}\left(1 - \operatorname{erf}\left(\frac{x_{\max} - x_k - \gamma - \delta^2/2}{\sqrt{2}\delta}\right)\right) \\ &- \frac{K}{2}\left(1 - \operatorname{erf}\left(\frac{x_{\max} - x_k - (\gamma - \delta^2/2)}{\sqrt{2}\delta}\right)\right). \end{aligned}$$

4. Temporal discretization and Fast Fourier Transforms (FFTs)

The discretization of the differential operator in (2) leads to a sparse matrix A with 9 non-zero elements per row while the numerical quadrature rule for the integral part leads to a block-diagonal matrix J with full diagonal blocks. Due to the different structures of the matrices we employ an implicit/explicit (IMEX) scheme in time which treats the discrete differential operator A implicitly and J explicitly. This way we avoid having to solve a dense linear system of equations each time-step.

First we employ the IMEX-Euler scheme [8], four time-steps using the time-step $\frac{\Delta t}{2}$

$$u^{n+1/2} = \frac{\Delta t}{2}Au^{n+1/2} + \frac{\Delta t}{2}Ju^n + u^n \ , \ n = 0, \tfrac{1}{2}, 1, \tfrac{3}{2}, \tag{6}$$

which can be written as

$$\left(I - \frac{\Delta t}{2}A\right)u^{n+1/2} = \frac{\Delta t}{2}Ju^n + u^n \ , \ n = 0, \tfrac{1}{2}, 1, \tfrac{3}{2}. \tag{7}$$

For the remaining time-steps we use IMEX Crank-Nicolson/Adams-Bashforth (IMEX-CNAB) scheme [28], with the time-step Δt

$$u^{n+1} = \frac{\Delta t}{2}A\left(u^{n+1} + u^n\right) + \frac{\Delta t}{2}J\left(3u^n - u^{n-1}\right) + u^n \ , n = 2, \ldots, N-1. \tag{8}$$

Reordering of (8) gives

$$\left(I-\frac{\Delta t}{2}A\right)u^{n+1}=\frac{\Delta t}{2}Au^{n}+\frac{\Delta t}{2}J\left(3u^{n}-u^{n-1}\right)+u^{n}\ ,\ n=2,\ldots,N-1. \tag{9}$$

From (7) and (9) we deduce that at each time-step we have to solve a sparse linear system of linear equations and multiply a vector with the block-diagonal matrix J with full diagonal blocks and in (9) also with the sparse matrix A.

We will solve the system of linear equations in (7) and (9) using an LU-decomposition. Note that both (7) and (9) has the same coefficient-matrix which means that we can form the LU-decomposition once prior to the time-stepping solving these systems.

For the matrix-vector multiplications with J we will use FFTs to increase the efficiency of the computations. We follow the discussion in [30] and define

$$\bar{I}_k=\Delta x\textstyle\sum_{j=1}^{m_x}\bar{u}(\zeta_j,v,\tau)\bar{f}(\zeta_j-x_k)\ ,\ k=1,\ldots,m_x.$$

After defining the Toeplitz matrix

$$T_{m_x}=\begin{pmatrix}\bar{f}(0) & \bar{f}(\Delta x) & \cdots & \bar{f}((m_x-1)\Delta x)\\ \bar{f}(-\Delta x) & \bar{f}(0) & \cdots & \bar{f}((m_x-2)\Delta x)\\ \vdots & \vdots & \ddots & \vdots\\ \bar{f}(-(m_x-1)\Delta x) & \bar{f}(-(m_x-2)\Delta x) & \cdots & \bar{f}(0)\end{pmatrix} \tag{10}$$

we can compute $\bar{I}=T_{m_x}\bar{u}$, where $\bar{I}=(\,\bar{I}_1\ \bar{I}_2\cdots\ \bar{I}_{m_x}\,)$ and $\bar{u}=(\,\bar{u}_1\ \bar{u}_2\cdots\ \bar{u}_{m_x}\,)$. Noticing that T_{m_x} in (10) can be embedded in a circulant matrix C_{2m_x-1} of size $2m_x-1$ we can first compute $\tilde{I}=C_{2m_x-1}\tilde{u}$ where $\tilde{u}=(\,\bar{u}_1\ \bar{u}_2\cdots\ \bar{u}_{m_x}\ 0\cdots 0\,)$ and then obtain $\bar{I}$ as the first m_x elements in $\tilde{I}$. The circulant matrix C_{2m_x-1} can be decomposed as $C_{2m_x-1}=F^{-1}_{2m_x-1}\Lambda F_{2m_x-1}$, where F_{2m_x-1} is a Fourier-matrix of order $2m_x-1$ and Λ is a diagonal matrix with the eigenvalues of C_{2m_x-1} on the diagonal. Hence $\tilde{I}$ can be computed as $\tilde{I}=F^{-1}_{2m_x-1}\Lambda F_{2m_x-1}\tilde{u}$ which can be accomplished by 1 FFTs and 1 inverse FFT (IFFT).

We choose $m_x=$ the number of grid-points in x to be twice the number $m_s=$ the number of grid-points in s in order to make the error caused by the approximation on the equidistant grid x_k a small fraction of the total error. In order to make efficient use of the FFTs described above, we will embed T_{m_x} in a matrix C_{M_x} where M_x is the smallest power of 2 such that $M_x\geq 2m_x-1$. Note that the eigenvalues of C_{M_x} can be computed once prior to the time-stepping. Summing this up we conclude that we can compute the matrix-vector multiplications by J as

- Interpolate $\bar{u}$ from the computational grid s_i to the equidistant grid x_k.
- Compute the Toeplitz matrix T_{m_x} in (10) and embed it into a circulant matrix C_{M_x}.
- Compute $\tilde{I}=F^{-1}_{2m_x-1}\Lambda F_{2m_x-1}\tilde{u}$ using FFT and IFFT.
- Obtain $\bar{I}$ by interpolating the first m_x elements of $\tilde{I}$ back to s_i.

5. Adaptivity

To enhance the performance of the method we will use adaptivity in space with the aim to place the grid-points where they increase accuracy the most, [23], [24].

We start by considering a PIDE $\frac{\partial u}{\partial \tau} + \mathcal{L}u = 0$ in one spatial dimension x that we discretize with a second-order method such that for a computed solution $u_h \in C^2$ it holds

$$u_h = u + h^2 c(x) \tag{11}$$

after neglecting high-order terms and hence $u_{2h} = u + (2h)^2 c(x)$. Using the second-order accuracy also in the local discretization error in space φ_h we get

$$\varphi_h = h^2 \eta(x). \tag{12}$$

From the definition of the local truncation error $\varphi_h = \mathcal{L}_h u - \mathcal{L}u$ and (11) we get

$$\varphi_h = \mathcal{L}_h u_h - \mathcal{L}u - h^2 \mathcal{L}_h c(x) \tag{13}$$

and

$$\varphi_{2h} = \mathcal{L}_{2h} u_h - \mathcal{L}u - h^2 \mathcal{L}_{2h} c(x), \tag{14}$$

where the term $\mathcal{L}_{2h} u_h$ is defined as the operator $\mathcal{L}_{2h}$ acting on every second element in u_h. Subtracting (13) from (14) and defining $\delta_h = \mathcal{L}_h u_h$ and $\delta_{2h} = \mathcal{L}_{2h} u_h$ gives

$$\varphi_{2h} - \varphi_h = \delta_{2h} - \delta_h - h^2(\mathcal{L}_{2h} - \mathcal{L}_h)c(x) = \delta_{2h} - \delta_h + \mathcal{O}(h^4).$$

Now using (12) and omitting high-order terms we get

$$\eta(x) \approx \frac{\delta_{2h} - \delta_h}{3h^2}\ , \varphi(x) = \frac{\delta_{2h} - \delta_h}{3}, \tag{15}$$

i.e. we can estimate $\eta(x)$ by computing a solution $u_{\bar{h}}$ using the spatial discretization $\bar{h}$ and employ (15). If we require $|\varphi_h| = |h^2 \eta(x)| < \epsilon$ for some tolerance ϵ we can obtain this by computing a solution using the new spatial discretization $h(x)$ defined by

$$h(x) = \bar{h}\sqrt{\frac{\epsilon}{|\varphi_{\bar{h}}(x)|}}.$$

To prevent us from using too large spatial steps, we introduce a small parameter d and define

$$h(x) = \bar{h}\sqrt{\frac{\epsilon}{|\varphi_{\bar{h}}(x)| + \epsilon \cdot d}}. \tag{16}$$

We will use extrapolation of $\varphi_{\bar{h}}$ in two grid-points at the boundaries $s = s_{\min}$, $s = s_{\max}$ and $v = v_{\max}$ to remove the effects caused by the boundary conditions used. To ensure a smooth $\varphi_{\bar{h}}$ we perform some smoothing iterations according to

$$\varphi_{\bar{h}}(x_k) = \left(\varphi_{\bar{h}}(x_{k-1}) + 2\varphi_{\bar{h}}(x_k) + \varphi_{\bar{h}}(x_{k+1})\right)/4.$$

For the PIDE in (2) we get that the local discretization error in space φ_{h_s,h_v} can be approximated by $\varphi_{h_s,h_v} = h_s^2 \eta_s(s,v) + h_v^2 \eta_v(s,v) + O(h_s^3) + O(h_v^3)$ if we neglect the influence from the mixed derivatives. Now, proceeding as in the one-dimensional case we can estimate $\varphi_{h_s}(s,v)$ by computing with $\bar{h}_s$ and $\bar{h}_{2s}$ and similarly for $\varphi_{h_v}(s,v)$. Then we compute

$$\hat{\varphi}_{h_s}(s) = \max_v |\varphi_{\bar{h}_s}(s,v)|,$$
$$\hat{\varphi}_{h_v}(v) = \max_s |\varphi_{\bar{h}_v}(s,v)|.$$

From $|\hat{\varphi}_{h_s}(s)| < \epsilon_s$ and $|\hat{\varphi}_{h_v}(v)| < \epsilon_v$ we can compute new one-dimensional grids $h_s(s)$ and $h_v(v)$ and form a tensor-product grid from these.

Since (2) is time-dependent the local discretization error φ_h will vary in time. We will use the solution u_h at three different time-levels $T/3$, $2T/3$ and T and use $\max|\varphi_h|$ over these time-steps when we compute the new computational grids.

We end up this section by summarizing the algorithm for adaptivity as follows:

(1) Compute a solution using a coarse spatial grid (m_s^c, m_v^c) and a coarse temporal discretization with N^c time-steps.
(2) Estimate the local truncation error on this grid and compute a new spatial grid (m_s^f, m_v^f) using (16) for some given ϵ.
(3) Compute a new solution using the new spatial grid (m_s^f, m_v^f) and N^f time-steps.

The values of m_s^c, m_v^c, N^c, ϵ, and N^f that we have used can be found in Table 2.

For American options the second derivative of the solution over the free boundary s_f is discontinuous. Hence $u_h \in C^2$ does not hold locally there and we will remove points in this region in our estimates of η and φ.

Note that the FFTs described in Section 4 are always performed in an equidistant grid in x no matter what the grid looks like in s. This means that for the integral part we will not really make use of the adaptive grid. However, this will not degrade the performance of our method as long as the grid x_k is fine enough which will be demonstrated in Section 7.

6. Operator splitting method

In this section we will describe how we solve the LCP (4). We will use the operator splitting method described in [12], [15], [30], for example. For the IMEX-Euler method (6) the operator splitting method is defined by

$$\left(I - \frac{\Delta t}{2}A\right)\tilde{u}^{n+1/2} = \frac{\Delta t}{2}Ju^n + \frac{\Delta t}{2}\lambda^n \text{ , } n = 0, \tfrac{1}{2}, 1, \tfrac{3}{2}. \tag{17}$$

$$\begin{aligned} &\frac{1}{\Delta t}\left(u^{n+1} - \tilde{u}^{n+1}\right) - \frac{1}{2}\left(\lambda^{n+1} - \lambda^n\right) = 0, \\ &\left(\lambda^{n+1}\right)^T\left(u^{n+1} - \Phi\right) = 0, \;\; u^{n+1} \geq \Phi \text{ and } \lambda^{n+1} \geq 0. \end{aligned} \tag{18}$$

Similarly for the IMEX-CNAB method (8) we have

$$\left(I - \frac{\Delta t}{2}A\right)\tilde{u}^{n+1} = \left(I + \frac{\Delta t}{2}A\right)u^n + \frac{\Delta t}{2}J\left(3u^n - u^{n-1}\right) + \Delta t\lambda^n \ , n = 2, \ldots, M-1, \tag{19}$$

$$\begin{aligned} &\frac{1}{\Delta t}\left(u^{n+1} - \tilde{u}^{n+1}\right) - \left(\lambda^{n+1} - \lambda^n\right) = 0, \\ &\left(\lambda^{n+1}\right)^T\left(u^{n+1} - \Phi\right) = 0, \ \ u^{n+1} \geq \Phi \ \text{ and } \ \lambda^{n+1} \geq 0. \end{aligned} \tag{20}$$

Hence, we first solve for $\tilde{u}^{n+1}$ using (17) and (19) and then update for u^{n+1} using (18) and (20), respectively.

7. Numerical results

We consider the numerical pricing of European and American call options under the model parameters given in Table 1. The parameters used for the discretization are listed in Table 2.

Table 1. Model parameters

Parameter	Value
Strike price K	100
Time of maturity T	0.5
Risk free interest rate r	0.02
Dividend yield q	0.06
Correlation between the Wiener processes ρ	-0.5
Mean level of variance θ	0.04
Rate of reversion κ	2.0
Volatility of the variance σ	0.25
Intensity rate of the jumps λ	0.2
Mean of the jumps γ	-0.5
Variance of the jumps δ	0.4

Table 2. Discretization parameters

Parameter	Value
$s_{\max}$	$4K$
$v_{\max}$	1
$x_{\min}$	log(100/1024)
$x_{\max}$	log($s_{\max}$)
Adaptivity parameter d	0.01
Coarse grid European m_s^c	41
Coarse grid European m_v^c	41
Coarse grid European N^c	40
Coarse grid American m_s^c	61
Coarse grid American m_v^c	61
Coarse grid American N^c	60

Our developed method has been implemented in MATLAB and the experiments have been performed on an AMD Opteron (Bulldozer) CPU in the Tintin cluster at Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX), Uppsala University.

To validate our method we compare the prices with reference prices compute using finite difference and Monte Carlo methods. The finite difference method is similar to the one described in this paper. It employs 8193 × 4097 refined grid and 2048 time steps. The Monte Carlo prices are computed based on 268 million paths with 1000 time steps. The reference Monte Carlo (MC) and finite difference (FD) prices are given in Table 3 in the points $p_n = (s_n, v_n)$ defined by $p_1 = (80, 0.04)$, $p_2 = (90, 0.04)$, $p_3 = (100, 0.04)$, $p_4 = (110, 0.04)$ and $p_5 = (120, 0.04)$.

Table 3. Reference prices

Option type	Price at p_1	Price at p_2	Price at p_3	Price at p_4	Price at p_5
European Call, MC	0.276117	1.852700	6.156860	12.955925	21.188638
European Call, FD	0.275908	1.852625	6.157288	12.956590	21.189415
American Call, FD	0.276239	1.853514	6.161108	12.980262	21.298121

For the equidistant method we will use the grids defined in Table 4.

Table 4. Computational grids for equidistant method

Grid	# grid-points in s, m_s	# grid-points in v, m_v	# time-steps, N
1	34	16	16
2	64	32	32
3	130	64	64
4	258	128	128
5	514	256	256

For the adaptive method we will use two different strategies. With *Parameter setting I* we will adjust ϵ_s and ϵ_v such that the resulting number of grid-points in the adaptive grid is the same as in Table 4. The required ϵ_s and ϵ_v can be found in Table 5. With *Parameter setting II* we will use $\epsilon_s = \epsilon_v$ defined in Table 6.

Table 5. Tolerances and grid sizes for Parameter setting I

European call option					
Grid	ϵ_s	ϵ_v	m_s^f	m_v^f	N^f
1^I	3.60e-1	1.30e-1	34	16	16
2^I	9.00e-2	2.50e-2	66	32	32
3^I	2.25e-2	5.50e-3	130	64	64
4^I	5.65e-3	1.34e-3	258	128	128
5^I	1.42e-4	3.27e-4	514	256	256
American call option					
Grid	ϵ_s	ϵ_v	m_s^f	m_v^f	N^f
1^I	4.50e-1	2.50e-1	34	16	16
2^I	1.13e-1	4.30e-2	66	32	32
3^I	2.05e-2	9.50e-3	130	64	64
4^I	7.15e-3	2.30e-3	258	128	128
5^I	1.79e-3	5.70e-4	514	256	256

Table 6. Tolerances and grid sizes for Parameter setting II

European call option				
Grid	$\epsilon_s = \epsilon_v$	m_s^f	m_v^f	N^f
1^{II}	3.60e-1	34	11	16
2^{II}	9.00e-2	66	18	32
3^{II}	2.25e-2	130	33	64
4^{II}	5.65e-3	258	64	128
5^{II}	1.42e-4	514	124	256
American call option				
Grid	$\epsilon_s = \epsilon_v$	m_s^f	m_v^f	N^f
1^{II}	4.50e-1	34	13	16
2^{II}	1.13e-1	66	21	32
3^{II}	2.05e-2	130	38	64
4^{II}	7.15e-3	258	74	128
5^{II}	1.79e-3	514	145	256

In Figure 1 we display the computational grids 1^I and 1^{II} for the European call option respectively.

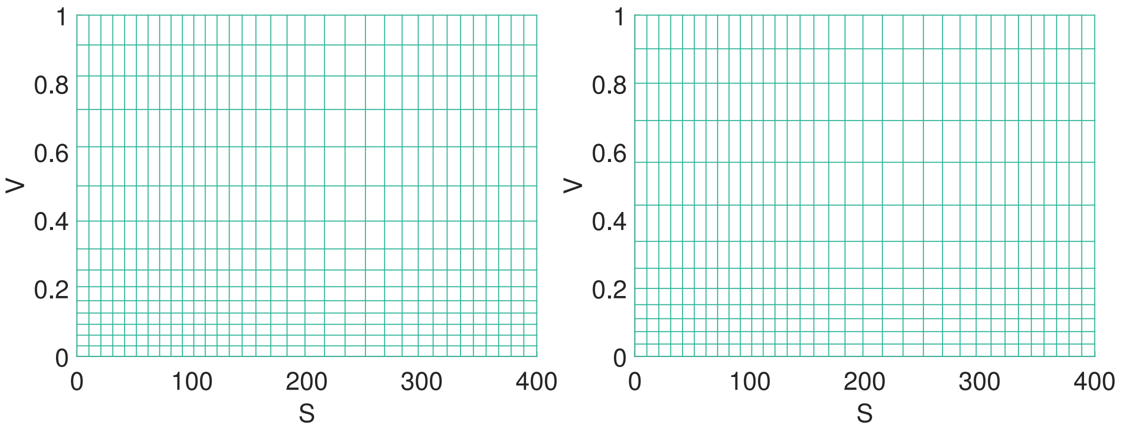

Figure 1. Computational grids 1^I (left) and 1^{II} (right).

Table 7. European call option

Equidistant grid

Grid	Value at p_1	Value at p_2	Value at p_3	Value at p_4	Value at p_5
1	1.009591	2.256089	6.133288	13.147451	21.418795
2	0.321351	1.832901	6.110899	12.964181	21.199079
3	0.288398	1.848922	6.146920	12.954746	21.191282
4	0.279098	1.851941	6.154781	12.955890	21.189602
5	0.276716	1.852481	6.156602	12.956333	21.189251

Adaptive grid, Parameter setting I

Grid	Value at p_1	Value at p_2	Value at p_3	Value at p_4	Value at p_5
1^{I}	0.372584	1.928847	6.126786	12.962764	21.217882
2^{I}	0.287465	1.820170	6.096471	12.923203	21.178242
3^{I}	0.282472	1.854420	6.153026	12.955278	21.190261
4^{I}	0.277213	1.852106	6.155264	12.955607	21.189121
5^{I}	0.276306	1.852689	6.156970	12.956404	21.189210

Adaptive grid, Parameter setting II

Grid	Value at p_1	Value at p_2	Value at p_3	Value at p_4	Value at p_5
1^{II}	0.386846	1.922644	6.080406	12.977948	21.239566
2^{II}	0.292318	1.806235	6.105908	12.931003	21.176725
3^{II}	0.283437	1.851852	6.154529	12.955998	21.190187
4^{II}	0.277414	1.851720	6.155369	12.955788	21.189136
5^{II}	0.276360	1.852589	6.156992	12.956455	21.189214

Table 8. American call option

Equidistant grid

Grid	Value at p_1	Value at p_2	Value at p_3	Value at p_4	Value at p_5
1	1.079182	2.294828	6.136774	13.181413	21.528324
2	0.324001	1.832550	6.108420	12.984336	21.303739
3	0.289376	1.849453	6.149199	12.977637	21.299406
4	0.279598	1.852736	6.158231	12.979404	21.298314
5	0.277090	1.853348	6.160339	12.980003	21.298094

Adaptive grid, Parameter setting I

Grid	Value at p_1	Value at p_2	Value at p_3	Value at p_4	Value at p_5
1^{I}	1.998162	2.915485	6.621065	13.311913	21.521047
2^{I}	0.303324	1.854678	6.128726	12.966343	21.298226
3^{I}	0.284322	1.856819	6.157372	12.979790	21.299793
4^{I}	0.277328	1.851382	6.156941	12.977977	21.297360
5^{I}	0.276565	1.853128	6.160219	12.979758	21.297917

Adaptive grid, Parameter setting II

Grid	Value at p_1	Value at p_2	Value at p_3	Value at p_4	Value at p_5
1^{II}	2.114465	2.989597	6.631520	13.356493	21.540491
2^{II}	0.305768	1.848060	6.130321	12.970583	21.297324
3^{II}	0.284911	1.855239	6.158241	12.980255	21.299679
4^{II}	0.277467	1.851111	6.157018	12.978109	21.297352
5^{II}	0.276601	1.853060	6.160236	12.979793	21.297915

In Tables 7 and 8 we present the computed solutions using the grids defined in Table 4–6. From the tables we see that the computed option prices converge towards the reference prices in Table 3 as the spatial grid is refined.

Next we will perform numerical experiments to verify the efficiency of the adaptive method. In Table 9–12 we display the error for both equidistant grids and the adaptive grids defined in Section 5. A comparison with precomputed non-uniform grids would of course also be of interest. However, it is not clear how such a grid should be constructed in the general case and we see the adaptive method presented in this paper as a simple and efficient way to create such grids. In [36] a comparison of e.g. finite difference methods on equidistant grids, precomputed non-uniform grids and adaptive grids is presented for a set of benchmarking problems.

We compare the computed solutions with a computed reference solution on a fine adaptive grid with 1026 grid-points in s and 512 points in v. The error is measured in the domain

$$\Omega_K = \{(s, v) | \tfrac{K}{2} \leq s \leq \tfrac{3K}{2}, 0 \leq v \leq 0.16\},$$

which we consider to be the domain where we are most interested in having an accurate solution. Both a numerical approximation of the L_2-norm of the error and the max-norm of the error is displayed. The quotients Q_i presented are defined as

$$Q_i = \frac{\text{Error in grid } (i-1)}{\text{Error in grid } i}.$$

We also show the average quotient $\bar{Q} = \left(\sum_{i=2}^{5} Q_i\right)/4$.

Table 9. Convergence in L_2-norm for European call option in Ω_K.

Grid	# time-steps	Equidistant	Q	Adaptive I	Q	Adaptive II	Q
$1 / 1^I / 1^{II}$	16	2.19e-0	-	1.87e-1	-	2.57e-1	-
$2 / 2^I / 2^{II}$	32	1.77e-1	12.4	1.04e-1	1.80	1.55e-1	1.70
$3 / 3^I / 3^{II}$	64	4.78e-2	3.71	1.05e-2	9.90	2.54e-2	6.10
$4 / 4^I / 4^{II}$	128	9.41e-3	5.08	2.97e-3	3.54	5.00e-3	5.04
$5 / 5^I / 5^{II}$	256	1.61e-3	5.84	4.46e-4	6.65	7.76e-4	6.55
$\bar{Q}$			6.76		5.47		4.85

Table 10. Convergence in max-norm for European call option in Ω_K.

Grid	# time-steps	Equidistant	Q	Adaptive I	Q	Adaptive II	Q
$1 / 1^I / 1^{II}$	16	2.01e-0	-	2.41e-1	-	3.52e-1	-
$2 / 2^I / 2^{II}$	32	4.32e-1	4.65	2.36e-1	1.02	4.61e-1	0.76
$3 / 3^I / 3^{II}$	64	1.94e-1	2.27	3.04e-2	7.76	1.13e-1	4.08
$4 / 4^I / 4^{II}$	128	4.60e-2	4.21	7.04e-3	4.31	2.43e-2	4.65
$5 / 5^I / 5^{II}$	256	7.84e-3	5.86	1.08e-3	6.52	4.04e-3	6.01
$\bar{Q}$			4.25		4.92		3.88

Table 11. Convergence in L_2-norm for American call option in Ω_K.

Grid	# time-steps	Equidistant	Q	Adaptive I	Q	Adaptive II	Q
$1 / 1^I / 1^{II}$	16	2.41e-0	-	5.95e-0	-	6.46e-0	-
$2 / 2^I / 2^{II}$	32	1.82e-1	13.2	6.99e-2	85.1	1.01e-1	64.0
$3 / 3^I / 3^{II}$	64	4.92e-2	3.70	1.24e-2	5.64	2.07e-2	4.88
$4 / 4^I / 4^{II}$	128	9.74e-3	5.05	6.62e-3	1.87	7.34e-3	2.82
$5 / 5^I / 5^{II}$	256	1.65e-3	5.90	1.18e-3	5.61	1.26e-3	5.83
$\bar{Q}$			6.96		24.6		19.4

From Tables 9–12 we see that we obtain in almost all cases at least the expected second-order convergence for both the equidistant and adaptive methods. We also see

Table 12. Convergence in max-norm for American call option in Ω_K.

Grid	# time-steps	Equidistant	Q	Adaptive I	Q	Adaptive II	Q
$1 / 1^I / 1^{II}$	16	2.07e-0	-	4.92e-0	-	5.35e-0	-
$2 / 2^I / 2^{II}$	32	4.34e-1	4.77	1.65e-1	29.8	2.82e-1	19.0
$3 / 3^I / 3^{II}$	64	1.95e-1	2.23	2.97e-2	5.55	8.43e-2	3.35
$4 / 4^I / 4^{II}$	128	4.62e-2	4.22	1.05e-2	2.82	2.18e-2	3.87
$5 / 5^I / 5^{II}$	256	7.79e-3	5.93	1.90e-3	5.53	3.80e-3	5.74
$\bar{Q}$			4.29		10.9		7.99

that for a given number of grid-points, the error for the adaptive method is most often smaller than the error for the equidistant one. Note however, that for the finest adaptive grids, the spatial discretization parameter is quite close to the one from the reference grid. Hence, this error estimate is not as accurate as for the coarser grids.

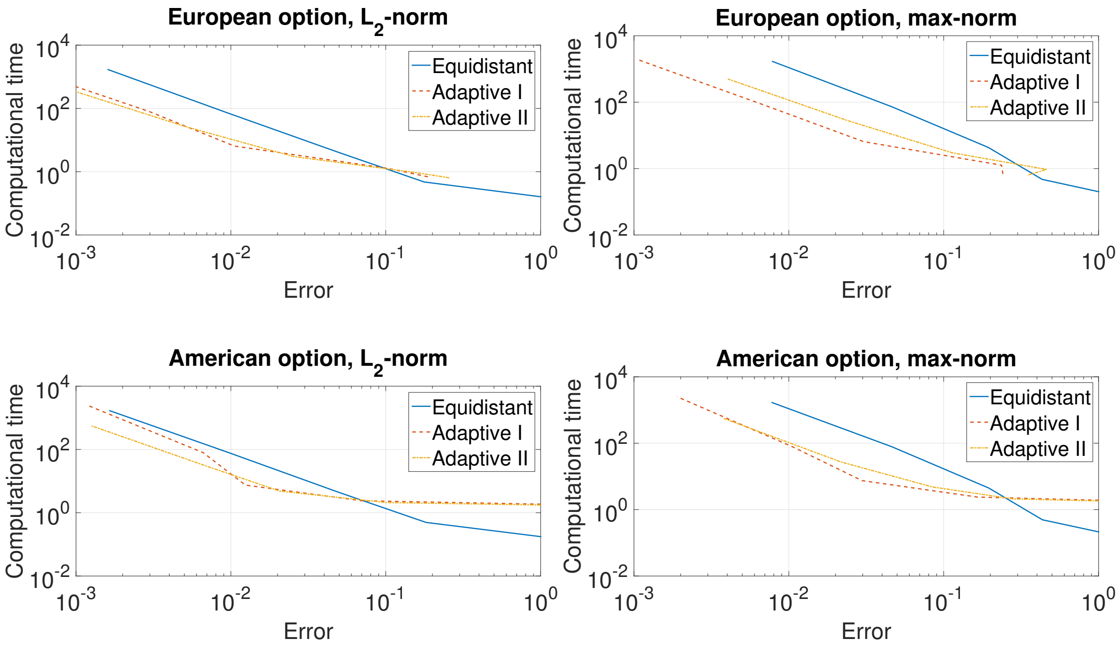

Figure 2. Computational time as a function of error for European option, L_2-norm (top left), European option, max-norm (top right), American option, L_2-norm (bottom left) and American option, max-norm (bottom right).

In Figure 2 we display the computational time as a function of the error in Tables 7 and 8. From this figure it is clear that for errors less than approximately 10^{-1} in the L_2-norm and less than approximately $3 \cdot 10^{-1}$ in the max-norm, it is beneficial to use the adaptive methods. For errors less than approximately $5 \cdot 10^{-2}$–$10 \cdot 10^{-2}$, the gain in computational time by using the adaptive methods is up to 20 times, depending on which option, parameter-setting and norm we are considering.

For the larger errors ($> 10^{-1}$), the computation of the solution on the coarse grid to estimate the local truncation error (step (1) in the adaptive method) takes relatively too much time of the whole adaptive algorithm. Thus, the adaptive methods are not competitive when we are satisfied with relatively large errors in the final solution.

We see that the gain by using the adaptive technique is larger in the max-norm which makes sense since the refinement of the grid is localized in the most difficult areas where it is likely the maximal error occurs.

8. Conclusions

In this paper we have developed an adaptive finite difference method to price options under the Bates model. This model gives rise to a parabolic PIDE that we discretize using an IMEX-scheme in time. The integrals that occur on the explicit side are computed using FFTs, while the spatial derivatives are discretized using second-order finite differences. For the LCPs occuring in the pricing of American options we employ an operator splitting method.

By estimating the local truncation error on a coarse equidistant grid, a new adaptive grid is computed such that an estimate of the final local truncation error is below a prescribed tolerance level. We have tried two different strategies in the computation of the adaptive grids. For both strategies it holds that if we want reasonably sized errors in the final solution, it is always beneficial to use the adaptive method compared to equidistant grids. To reach a given fairly high accuracy level, the computational time can be reduced up to 20 times.

Acknowledgments

The work in this paper builds on the sequence of Master theses/project reports [32], [31], [37] and [38]. The authors are thankful for the early contributions of these students.

The computations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) through Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX) under Projects snic2014-3-24 and snic2015-6-77.

The first author was financed by the Swedish Research Council under contract number 621-2007-6388.

We thank the referees for their suggestions and comments.

References

[1] Y. Achdou and O. Pironneau, *Computational methods for option pricing*, Frontiers in Applied Mathematics, Vol. 30, SIAM, Philadelphia, PA, 2005.

[2] A. Almendral and C.W. Oosterlee, *Numerical valuation of options with jumps in the underlying*, Appl. Numer. Math. 53 (2005), pp. 1–18.

[3] L. Andersen and J. Andreasen, *Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing*, Rev. Deriv. Res. 4 (2000), pp. 231–262.

[4] D.S. Bates, *Jumps and stochastic volatility: Exchange rate processes implicit Deutsche mark options*, Review Financial Stud. 9 (1996), pp. 69–107.

[5] F. Black and M. Scholes, *The pricing of options and corporate liabilities*, J. Polit. Econ. 81 (1973), pp. 637–654.

[6] C. Chiarella, B. Kang, G.H. Meyer, and A. Ziogas, *The evaluation of American option prices under stochastic volatility and jump-diffusion dynamics using the method of lines*, Int. J. Theor. Appl. Finance 12 (2009), pp. 393–425.

[7] N. Clarke and K. Parrott, *Multigrid for American option pricing with stochastic volatility*, Appl. Math. Finance 6 (1999), pp. 177–195.

[8] R. Cont and E. Voltchkova, *A finite difference scheme for option pricing in jump diffusion and exponential Lévy models*, SIAM Numer. Anal. 43 (2005), pp. 1596–1626.

[9] Y. d'Halluin, P.A. Forsyth, and G. Labahn, *A penalty method for American options with jump diffusion processes*, Numer. Math. 97 (2004), pp. 321–352.

[10] Y. d'Halluin, P.A. Forsyth, and K.R. Vetzal, *Robust numerical methods for contingent claims under jump diffusion processes*, IMA J. Numer. Anal. 25 (2005), pp. 87–112.

[11] S. Heston, *A closed-form solution for options with stochastic volatility with applications to bond and currency options*, Rev. Financial Stud. 6 (1993), pp. 327–343.

[12] S. Ikonen and J. Toivanen, *Operator splitting methods for American option pricing*, Appl. Math. Lett. 17 (2004), pp. 809–814.

[13] S. Ikonen and J. Toivanen, *Componentwise splitting methods for pricing American options under stochastic volatility*, Int. J. Theor. Appl. Finance 10 (2007), pp. 331–361.

[14] S. Ikonen and J. Toivanen, *Efficient numerical methods for pricing American options under stochastic volatility*, Numer. Methods Partial Differential Equations 24 (2008), pp. 104–126.

[15] S. Ikonen and J. Toivanen, *Operator splitting methods for pricing American options under stochastic volatility*, Numer. Math. 113 (2009), pp. 299–324.

[16] K.J. In 't Hout and S. Foulon, *ADI finite difference schemes for option pricing in the Heston model with correlation*, Int. J. Numer. Anal. Model. 7 (2010), pp. 303–320.

[17] Y. Kwon and Y. Lee, *A second-order finite difference method for option pricing under jump-diffusion models*, SIAM J. Numer. Anal. 49 (2011), pp. 2598–2617.

[18] Y. Kwon and Y. Lee, *A second-order tridiagonal method for American options under jump-diffusion models*, SIAM J. Sci. Comput. 33 (2011), pp. 1860–1872.

[19] P. Lötstedt, J. Persson, L. von Sydow, and J. Tysk, *Space-time adaptive finite difference method for European multi-asset options*, Comput. Math. Appl. 53 (2007), pp. 1159–1180.

[20] R.C. Merton, *Theory of rational option pricing*, Bell J. Econom. Man. Sci. 4 (1973), pp. 141–183.

[21] E. Miglio and C. Sgarra, *A finite element discretization method for option pricing with the Bates model*, S$\vec{e}$MA J. (2011), pp. 23–40.

[22] C.W. Oosterlee, *On multigrid for linear complementarity problems with application to American-style options*, Electron. Trans. Numer. Anal. 15 (2003), pp. 165–185.

[23] J. Persson and L. von Sydow, *Pricing European multi-asset options using a space-time adaptive FD-method*, Comput. Vis. Sci. 10 (2007), pp. 173–183.

[24] J. Persson and L. von Sydow, *Pricing American options using a space-time adaptive finite difference method*, Math. Comput. Simulation 80 (2010), pp. 1922–1935.

[25] A. Ramage and L. von Sydow, *A multigrid preconditioner for an adaptive Black-Scholes solver*, BIT 51 (2011), pp. 217–233.

[26] C. Reisinger and G. Wittum, *On multigrid for anisotropic equations and variational inequalities: Pricing multi-dimensional European and American options*, Comput. Vis. Sci. 7 (2004), pp. 189–197.

[27] S. Salmi and J. Toivanen, *An iterative method for pricing American options under jump-diffusion models*, Appl. Numer. Math. 61 (2011), pp. 821–831.

[28] S. Salmi and J. Toivanen, *IMEX schemes for pricing options under jump-diffusion models*, Appl. Numer. Math. 84 (2014), pp. 33–45.

[29] S. Salmi, J. Toivanen, and L. von Sydow, *Iterative methods for pricing American options under the Bates model*, in *Proceedings of 2013 International Conference on Computational Science*, Procedia Computer Science Series, Vol. 18, Elsevier, 2013, pp. 1136–1144.

[30] S. Salmi, J. Toivanen, and L. von Sydow, *An IMEX-scheme for pricing options under stochastic volatility models with jumps*, SIAM J. Sci. Comput. 36 (2014), pp. B817–B834.

[31] A. Sjöberg, *Adaptive finite differences to price European options under the Bates model*, Master's thesis, Department of Information Technology, Uppsala University, 2013, iT Report 13 063.

[32] R. Skogberg and J. Markhed, *Pricing European options using stochastic volatility and integral jump term* (2009).

[33] D. Tavella and C. Randall, *Pricing financial instruments: The finite difference method*, John Wiley & Sons, Chichester, 2000.

[34] J. Toivanen, *A componentwise splitting method for pricing American options under the Bates model*, in *Applied and numerical partial differential equations*, Comput. Methods Appl. Sci., Vol. 15, Springer, New York, 2010, pp. 213–227.

[35] J. Toivanen and C.W. Oosterlee, *A projected algebraic multigrid method for linear complementarity problems*, Numer. Math. Theory Methods Appl. 5 (2012), pp. 85–98.

[36] von Sydow L., H. L.J., L. E., L. E., M. S., P. J., S. V., S. Y., S. S., T. J., W. J., W. M., L. J., L. J., O. C.W., R. M.J., T. A., and Z. Y., *BENCHOP–the BENCHmarking project in Option Pricing* Accepted for publication.

[37] A. Westlund, *An IMEX-method for pricing options under Bates model using adaptive finite differences* (2014).

[38] C. Zhang, *Fast Fourier transforms in IMEX-schemes to price options under Bates model*, Master's thesis, Uppsala University, 2014.

[39] R. Zvan, P.A. Forsyth, and K.R. Vetzal, *Penalty methods for American options with stochastic*

volatility, J. Comput. Appl. Math. 91 (1998), pp. 199–218.

[40] R. Zvan, P.A. Forsyth, and K.R. Vetzal, *Robust numerical methods for PDE models of Asian options*, J. Comput. Finance 1 (1998), pp. 39–78.