UPPSALA
UNIVERSITET

PROJEKTRAPPORT

An IMEX-method for pricing options under Bates model using adaptive finite differences

Arvid Westlund

Rapport i Teknisk-vetenskapliga datorberäkningar

Jan 2014

INSTITUTIONEN FÖR INFORMATIONSTEKNOLOGI

Abstract

Trading with options in the financial market has increased dramatically during the last decades. It is therefore of great importance to have fast and accurate methods for pricing them, which is the purpose of this report. The problem that has been examined is Bates model, which is a two-dimensional jump-diffusion model. Solving the problem with an adaptive method has proven to be faster than using an equidistant method, for a predetermined error. Using an IMEX-scheme (implicit-explicit) is shown to be faster than a fully implicit scheme, such as BDF2, without any loss of accuracy.

Introduction

In the financial market, an option is a contract issued between two parties where the buyer is given the right to buy (call option) or sell (put option) an asset, X, at the strike price, K, at a maturity date, T. However, the buyer has no obligation to exercise the option. The most common options are American and European options. An American option gives the buyer the right to exercise the option at any time, from the day the option is purchased, to maturity date. A European option can only be exercised at time of maturity. Since options are widely and heavily traded throughout the world, it is of great importance to have fast and accurate methods for pricing them.

The value of a European call option at time of maturity is given by the following equation,

$$\phi(X) = V(X_T, T) = max(X_T - K, 0), \tag{1}$$

where X is the value of the underlying asset. It is obvious that the option will only be exercised if the value of the asset is larger than the strike price, K. Otherwise the value of the option is 0.

One of the most common models for option pricing is the Black-Scholes model. It describes the value of an asset as a stochastic differential equation (SDE),

$$dX_t = \mu X_t dt + \sigma X_t dW_t. \tag{2}$$

Here, W is a Wiener process, μ is the drift term and σ is the volatility which describes the stochastic behavior of the asset. Values of σ indicates how influential the Wiener process in equation (2) will be. However, for pricing options that have particularly short maturities it is necessary to modify the Black-Scholes equation. The Bates model, see [6], uses the Black-Scholes equation but also introduces a jump-diffusion term,

$$dX_t = (r - q - \lambda\xi)X_t dt + \sqrt{Y_t} X_t dW_t^1 + (B-1)X_t dn\,, \tag{3}$$

$$dY_t = \kappa(\theta - Y_t)dt + \sigma\sqrt{Y_t} dW_t^2.$$

Here r is the risk free rate, q the dividend yield, ξ the mean jump, B the jump size, λ the intensity of the Poisson process, N, κ the rate of reversion of the variance Y, W^1 and W^2 are Wiener processes with correlation ρ. θ and σ are the mean level and volatility of Y repectively.

The distribution of the jump is a Poisson process

$$f(B) = \frac{1}{\sqrt{2\pi}\delta B} exp\left(-\frac{[lnB - (\gamma - \delta^2/2)]^2}{2\delta^2}\right) \tag{4}$$

where γ and δ describe the mean of the jump and the variance of the jump respectively. This model takes into account that a sudden large change in the asset might happen. In the case where we have long maturities, the jump term will not be as influential since it is likely that the effect of a sudden jump will even out over time. Let $\tau = T - t$ and $\xi = e^{\gamma - 1}$ then the price of the option under Bates model can be described by the following partial integro-differential equation (PIDE):

$$\frac{\partial u(x,y,\tau)}{\partial t} = \frac{1}{2}yx^2\frac{\partial^2 u(x,y,\tau)}{\partial x^2} + \frac{1}{2}\sigma^2 y\frac{\partial^2 u(x,y,\tau)}{\partial y^2}$$

$$+\rho\sigma yx\frac{\partial^2 u(x,y,\tau)}{\partial x\partial y} + (r-q-\lambda\xi)x\frac{\partial u(x,y,\tau)}{\partial x} + \kappa(\theta - Y_t)\frac{\partial u(x,y,\tau)}{\partial y} \quad (5)$$

$$-(r+\lambda)\partial u(x,y,\tau) + \lambda\int_0^\infty u(Bx,y,\tau)f(B)dJ.$$

The integral term comes from the jump-diffusion term in equation (3). The boundary and initial conditions that are imposed are the following:

$$\begin{cases} u(0,y,\tau) = \phi(0,y) * e^{-r\tau}, \\ u(X,y,\tau) = \phi(X,y) * e^{-r\tau}, \\ \frac{\partial u(x,Y,\tau)}{\partial y} = 0, \\ x \in (0,X), y \in [0,Y], \\ u(x,y,0) = \phi(x,y). \end{cases} \quad (6)$$

In [1] the same problem was solved with an adaptive finite difference method. However, the adaptive grid in [1] only had adaptivity in x. In time the problem was solved with a two step implicit scheme. The purpose of this project is as follows. (i), complete the adaptive grid by adding adaptivity in y. (ii), implement an IMEX-scheme when solving the problem. This is an implicit explicit scheme that is expected to save CPU compared to an implicit scheme.

Discrete approximation in space

The space discretization is done as in [1] and [3] and we refer for more details to these articles. In brief, the problem is solved on a two dimensional grid with x and y as underlying variables. The discrete solution is sorted lexicographically in a column vector, ψ.

$$\psi \equiv \left(\psi_{1,1}\ \psi_{1,2} \dots\ \psi_{1,m} \dots \psi_{n,1}\psi_{n,2}\ \dots\ \psi_{n,m}\right)^T$$

Second order finite differences are used to approximate the derivatives in equation (5)

$$u_{xx} = \frac{\psi_{i+1,j}+2\psi_{i,j}-\psi_{i-1,j}}{h_x^2},$$

$$u_{yy} = \frac{\psi_{i,j+1}+2\psi_{i,j}-\psi_{i,j-1}}{h_y^2},$$

$$u_{xy} = \frac{\psi_{i+1,j+1}-\psi_{i+1,j-1}-\psi_{i-1,j+1}+\psi_{i-1,j-1}}{4h_xh_y}, \quad (7)$$

$$u_x = \frac{\psi_{i+1,j}-\psi_{i-1,j}}{2h_x},$$

$$u_y = \frac{\psi_{i,j+1}-\psi_{i,j-1}}{2h_y}.$$

There are similar approximations for non-equidistant grids that we will use for the adaptive grid, see [1] The discrete approximation of the integral for each x_i is defined by

$$J_i = \int_0^{\infty} u(Bx_i, y, \tau)\ f(B)dB. \tag{8}$$

Making the change of variables $B = e^s$ leads to

$$J_i = \int_{-\infty}^{\infty} e^s u(e^s x_i, y, \tau)\ f(e^s)ds. \tag{9}$$

Equation (9) can then be written as

$$J_i = \sum_{k=0}^{n} J_{i,k} + \int_{ln\frac{x_{max}}{x_i}}^{\infty} \phi(e^s x_i, y)\ p(s)ds \tag{10}$$

where $p(s) = e^s f(e^s)$ and

$$J_{i,k} = \int_{ln\frac{x_k}{x_i}}^{ln\frac{x_{k+1}}{x_i}} u(e^s x_i, y)\ p(s)ds. \tag{11}$$

.

The trapezoidal rule is applied to (11)

$$\begin{aligned} J_{i,k} &\approx \frac{1}{2} ln\frac{x_{k+1}}{x_k}\left[u\left(e^{ln\frac{x_k}{x_i}} x_i, y, \tau\right) p\left(ln\frac{x_k}{x_i}\right) + u\left(e^{ln\frac{x_{k+1}}{x_i}} x_i, y, \tau\right) p\left(ln\frac{x_{k+1}}{x_i}\right)\right] \\ &= \frac{1}{2} ln\frac{x_{k+1}}{x_k}\left[u(x_k, y, \tau) p\left(ln\frac{x_k}{x_i}\right) + u(x_k, y, \tau) p\left(ln\frac{x_{k+1}}{x_i}\right)\right]. \end{aligned} \tag{12}$$

When $x_k = x_0 = 0$ in Equation (12) it can be approximated as

$$\begin{aligned} J_{i,0} &= \int_{-\infty}^{ln\frac{x_1}{x_i}} u(e^s x_i, y, \tau)\ p(s)ds \approx u(x_i, y, \tau) \int_{-\infty}^{ln\frac{x_1}{x_i}} p(s)\ ds \\ &= \frac{1}{\sqrt{2\pi}\delta} u(x_i, y, \tau) \int_{-\infty}^{ln\frac{x_1}{x_i}} e^{\frac{-[s-(\gamma-\delta^2/2)]^2}{2\delta^2}} ds \\ &= \frac{1}{\sqrt{\pi}} u(x_i, y, \tau) \int_{-\hat{z}}^{ln\frac{x_1}{x_i}} e^{z^2}\, dz = \frac{1}{2} u(x_i, y, \tau) erfc(-\hat{z}) \end{aligned} \tag{13}$$

where $\hat{z} = \frac{-\left[ln\frac{x_1}{x_i} - (\gamma - \delta^2/2)\right]^2}{2\delta^2}$.

Finally, the second part of Equation (10) needs to be written in discrete form. In order to do this one uses the approximations $erfc\left(\infty - (\gamma - \delta^2/2)\right) \approx -1$ and $erfc\left(\gamma + \frac{\delta^2}{2} - \infty\right) \approx 1$. Hence

$$\begin{aligned} &\int_{ln\frac{x_{max}}{x_i}}^{\infty} \phi(e^s x_i, y)\ p(s)ds = \frac{1}{\sqrt{2\pi}\delta} \int_{ln\frac{x_{max}}{x_i}}^{\infty} max(e^s x_i - K, 0)\ e^{\frac{-[s-(\gamma-\delta^2/2)]^2}{2\delta^2}} ds \\ &\approx \frac{1}{2} x_i e^{\gamma}\left[1 + erf\left(\frac{\gamma + \delta^2/2 - ln\frac{x_{max}}{x_i}}{\sqrt{2}\delta}\right)\right] + \frac{1}{2} K e^{-rt}\left[-1 + erf\left(\frac{ln\frac{x_{max}}{x_i} - (\gamma - \delta^2/2)}{\sqrt{2}\delta}\right)\right] \end{aligned} \tag{14}$$

Adaptive method in space

To improve accuracy and efficiency the problem is solved using an adaptive method. This is done by first solving the problem on a sparse equidistant grid. Given a tolerance for the error, a new adaptive grid is constructed where points are placed where they are needed for accuracy. The problem is then solved once more on the new adaptive grid. The method presented is the same that are presented in [6].

The adaptive grid in x is derived in the following manner (y-dimension is analogously). Let u be the exact solution and M the "exact" discrete representation for the right hand side of equation (5). Then

$$M_{h_x h_y}\psi_{h_x h_y} = Mu + \tau_{h_x h_y} \tag{15}$$

where $M_{h_x h_y}$ is the discrete approximation in space for equation (5),$\psi_{h_x h_y}$,the numerical solution and $\tau_{h_x h_y}$,the local truncation error. The local truncation error is approximated by

$$\tau_{h_x h_y} \approx h_x^2\eta_x(x,y) + h_y^2\eta_y(x,y) = \tau_{h_x} + \tau_{h_y}. \tag{16}$$

Define

$$\begin{aligned}\delta_{h_x} &= Mu + \tau_{h_x} = M_{h_x h_y}\psi_{h_x h_y}\\ \delta_{2h_x} &= Mu + \tau_{2h_x} = M_{2h_x h_y}\psi_{2h_x h_y}\end{aligned} \tag{17}$$

and we use Richardson interpolation to get an expression for τ_{h_x}

$$\tau_{h_x} = \frac{1}{2^2-1}\left(\delta_{2h_x} - \delta_{h_x}\right). \tag{18}$$

In order to get an estimate for $n_x(x,y)$ we compute a solution with step-length $\hat{h}_x$. Then we compute a new matrix $M_{2\hat{h}_x}$ with step-length $2\hat{h}_x$. Then from equation (17) we get

$$\eta_x(x,y) = \frac{\tau_{\hat{h}_x}}{\hat{h}_x^2}. \tag{19}$$

To control the truncation error we combine equations (16) and (19) to get

$$\left|\tau_{h_x}(x)\right| = \left|h_x^2(x)\eta_x(x,y)\right| \approx \left|h_x^2(x)\frac{\tau_{\hat{h}_x}(x,y)}{\hat{h}_x^2(x)}\right| \le \epsilon_x. \tag{20}$$

Using $\hat{\tau}_{\hat{h}_x}(x) = max_y\left(\tau_{\hat{h}_x}(x,y)\right)$ we get,

$$h_x(x) = \hat{h}_x^2(x)\left(\frac{\epsilon_x}{\hat{\tau}_{\hat{h}_x}(x)}\right)^{1/2}. \tag{21}$$

Finally an additional parameter, γ_x , is added to the expression for $h_x(x)$ in order to prevent too big space-steps when the truncation error is small. This leads to the final expression for the step length

$$h_x(x) = \hat{h}_x^2(x)\left(\frac{\epsilon_x}{\epsilon_x\gamma_x + \hat{\tau}_{\hat{h}_x}(x)}\right)^{1/2}. \tag{22}$$

The adaptivity in x was implemented and described in [1]. In this project we have also included adaptivity in the $\boldsymbol{y}$-direction making the adaptive grid complete. However, some difficulties were encountered regarding this. Close to the boundary where $y = Y_{max}$ the step lengths diverged. This was fixed by extrapolating δ_{h_y} and δ_{2h_y}in the last two grid points at the boundary.

An IMEX method

The discrete approximation of Equation (5) gives two matrices, $\mathbf{A}$ and $\mathbf{J}$.

$$\frac{\partial u(x,y,\tau)}{\partial t} \approx \mathbf{A}\psi + \mathbf{J}\psi + \mathbf{a} + \mathbf{b}. \tag{23}$$

$\mathbf{A}$ comes from the terms that contains derivatives and $\mathbf{J}$ from the integral in (5). In order to maintain symmetry in matrices $\mathbf{A}$ and $\mathbf{J}$, the terms from the boundaries are excluded in $\mathbf{A}$ and $\mathbf{J}$. Instead, these terms are gathered in vectors $\mathbf{a}$ and $\mathbf{b}$ which are added afterwards. This is done for computational reasons. Due to the structure of matrices the $\mathbf{A}$ and $\mathbf{J}$ different discretization schemes in time may be less or more suitable.

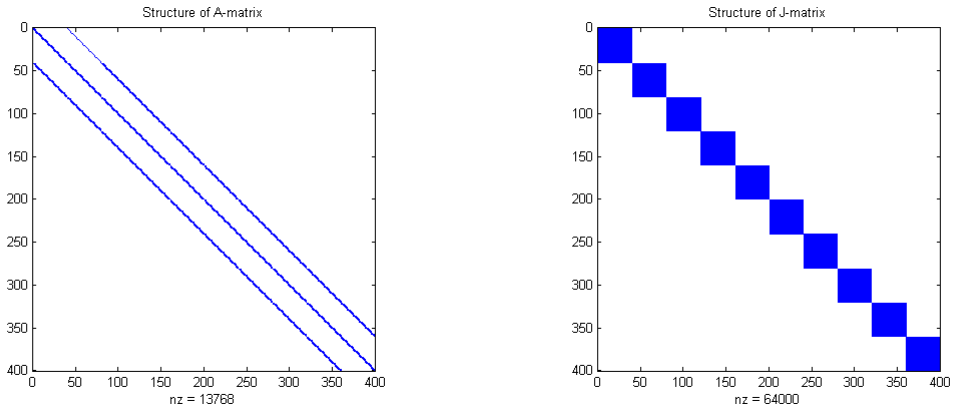

*Figure1. The structure of matrix **A** and **J***

In [1] the Bates model is solved with a BDF2-scheme. The BDF2-scheme is written as

$$\frac{3}{2}\psi^{n+1} \approx \Delta t(\mathbf{A}+\mathbf{J})\psi^{n+1} + \Delta t(\mathbf{a}^{n+1}+\mathbf{b}^{n+1}) + 2\psi^{n} - \frac{1}{2}\psi^{n-1}. \tag{24}$$

This is an unconditionally stable scheme with the drawback that it is time consuming when dealing with full matrices. One way to circumvent this problem is to treat the problem both implicitly and explicitly. This is called an IMEX-scheme. The sparse matrices are solved implicitly whereas the full matrices are treated explicitly. In this thesis the Crank/Nicholson-Adams/Bashforth scheme is chosen

$$\psi^{n+1} \approx \Delta t\mathbf{A}\frac{\psi^{n+1}+\psi^{n}}{2} + \Delta t\mathbf{J}\frac{3\psi^{n}-\psi^{n-1}}{2} + \Delta t\frac{\boldsymbol{a}^{n+1}+\boldsymbol{a}^{n}}{2} + \Delta t\frac{3\boldsymbol{b}^{n}-\boldsymbol{b}^{n-1}}{2} + \psi^{n}. \tag{25}$$

This gives us a conditionally stable scheme. It will be faster than a fully implicit scheme and more stable than a fully explicit scheme. Looking at Equation (25) one can see that the first part of the right hand side is treated implicitly whereas the second part is treated explicitly. A more detailed description of the IMEX method can be found in [2].

Since equation (25) is a two step scheme we first need two solutions before we can start. We already have ψ^0 but ψ^1 still needs to be evaluated. Backward Euler is used to compute ψ^1

$$\psi^{1} \approx \Delta t\mathbf{A}\psi^{1} + \Delta t\mathbf{J}\psi^{1} + \Delta t(\boldsymbol{a}^{1}+\boldsymbol{b}^{1}) + \psi^{0}. \tag{26}$$

Numerical results

All computations have been carried out in Matlab and performed on a Dell Lattitude E6400I, Intel(R)Core(TM)2 Duo,2.53GHz,1.89 GB RAM. When testing the program, values from a previous report, written on the same subject [4], has been used as a reference.

In the numerical experiments the risk free rate is $r = 0.03$, the dividend yield $q = 0.05$, the intensity of the Poisson process $\lambda = 0.2$, the mean of the jump $\gamma = -0.5$, the variance of the jump $\delta = 0.4$, the rate of reversion $\kappa = 2.0$, the correlation between $W^1$and W^2 are $\rho = -0.5$, the mean level of the variance $\theta = 0.04$ and volatility of the variance $\sigma = 0.25$. The strike price is $K = 100$, $X = 4 * K$ and $Y = 1$.

The solution of (5) looks like a smooth slope rising from $x = 100$. We can expect the price of the option to be slightly larger where the volatility is greater. This is clear from Figure 2 where the transition is smoother and not as abrupt where the volatility is 1 compared to where it is 0. The reason for this is that when the volatility increases so does the variance and also the chance that the asset might reach higher values.

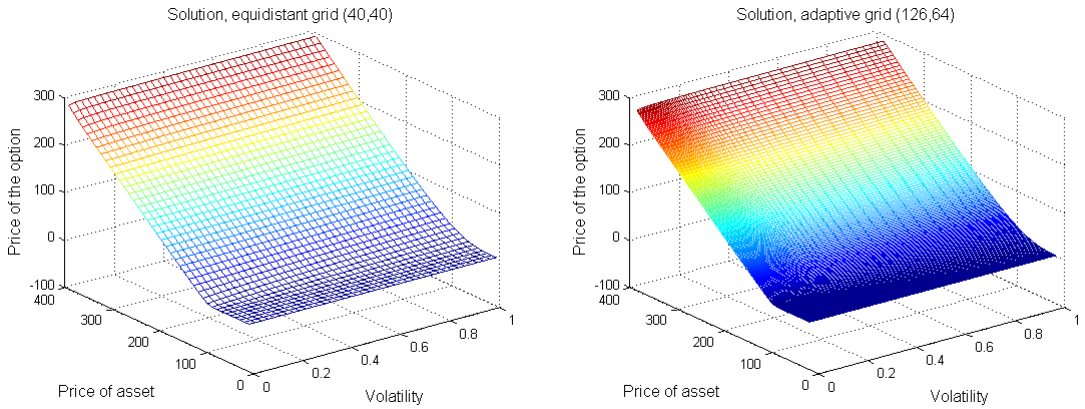

Figure 2. The solution using an equidistant and- adaptive grid.

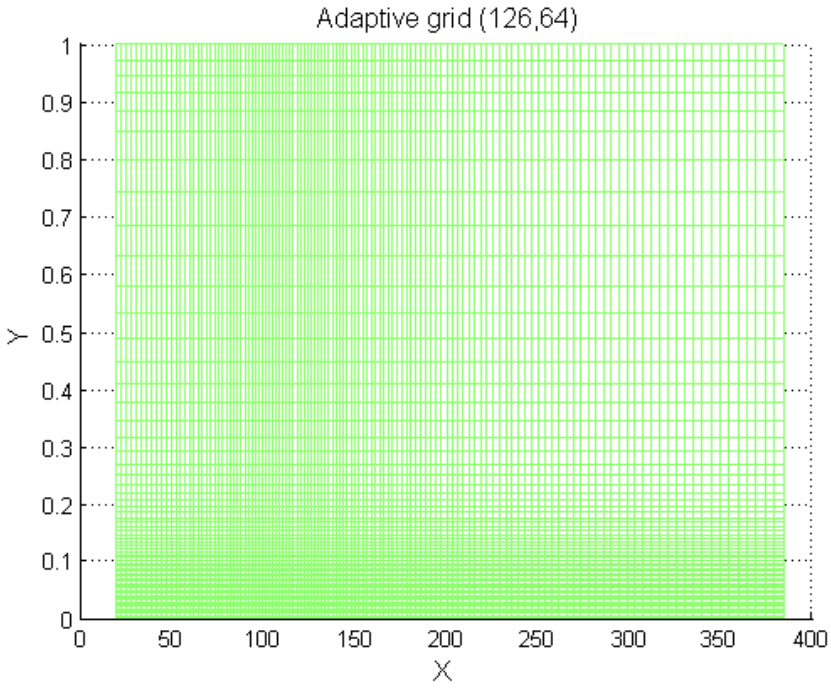

Figure 3. The adaptive grid seen from above. The x-axis is the price of the asset and the y-axis is the volatility.

Looking at the adaptive solution in Figure 2 and at the grid in Figure 3 we see that the number of grid points are denser in the regions $x = 100$ and $y = 0$. This makes sense when looking at Figure 2 where the solution suddenly increases. This is to be expected since the strike price, K, is 100. At the boundary of $y = 0$ the derivatives in the solution are big which results in a denser distribution of grid points.

In order to compare the accuracy and efficiency between the two methods a reference solution is needed. The solution from an equidistant grid with 1022 points in x and 512 points in y is considered a good estimate. Let the reference solution be u, then the absolute error can be written

$$E = \max_{(x,y)\in\Omega} |u(x,y,T) - \psi(x,y,T)| \tag{27}$$

$$\Omega = \left\{x \middle| {}^{K}/_{3} \le x \le {}^{5K}/_{3}, y \middle| 0.1 \le y \le 0.9\right\}.$$

When using the adaptive method the problem is solved on the same coarse grid, $(n, m, k) = (40,40,256)$, where n is the number of points in x, m is the number of points in y and k is the number of steps in time. We want to compare the error in the equidistant and adaptive method for the same amount of grid points.

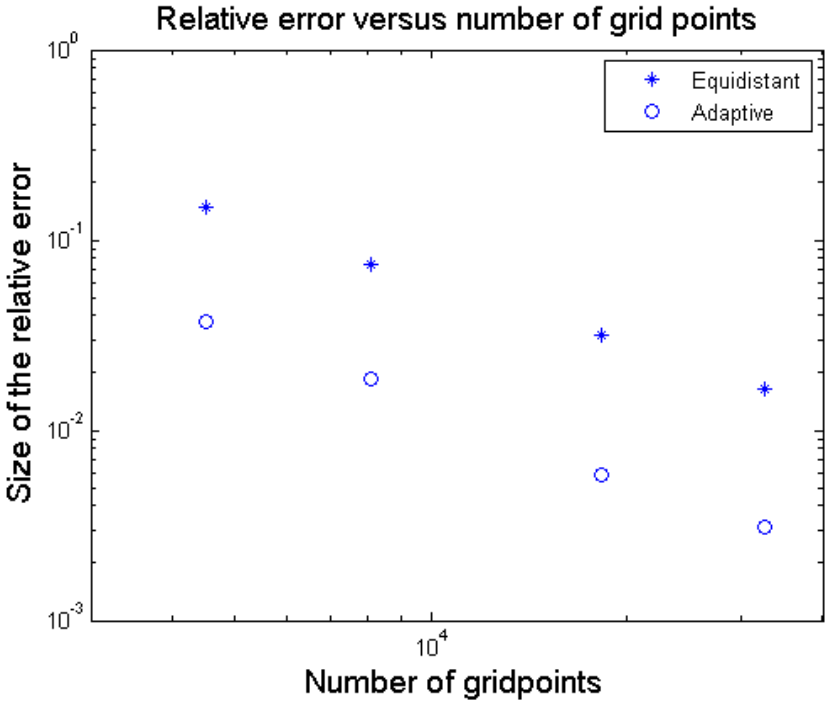

Figure 4. The relative error versus number of grid points for the equidistant and adaptive method

Figure 4 shows that the adaptive method is more accurate than the equidistant method, for the same amount of grid points. We also want to examine how the consumption of time is affected for a specific error for both methods.

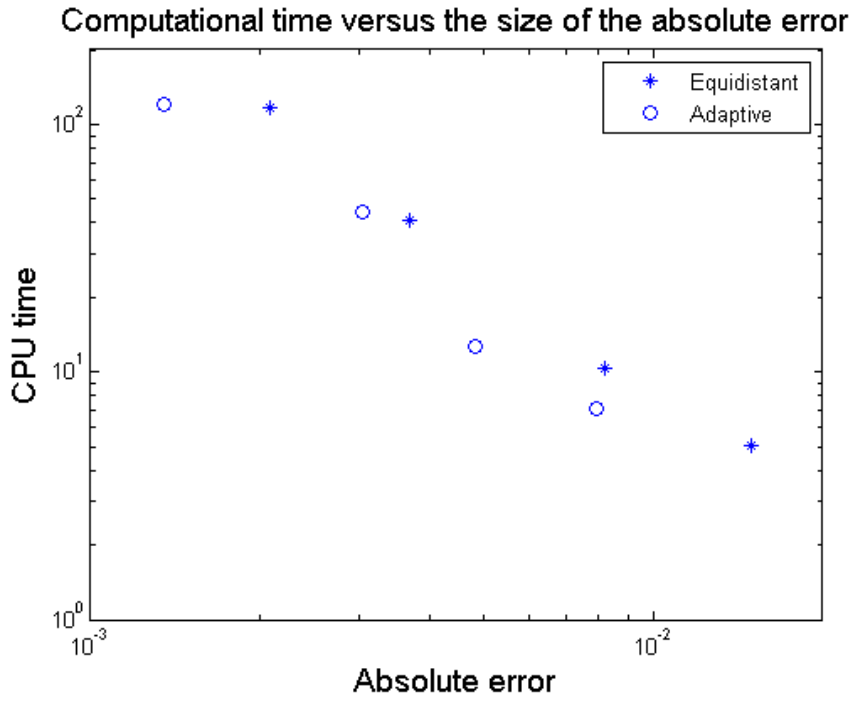

Figure 5. The computational time vs the size of the absolute error for the equdistant and adaptive method.

In Figure 5 we see that the adaptive method is faster than the equidistant method when we demand a small error. For a larger error the equidistant eventually out performs the adaptive method.

To be sure that the IMEX method is accurate through the whole time interval a plot where one studies the error through time might be insightful.

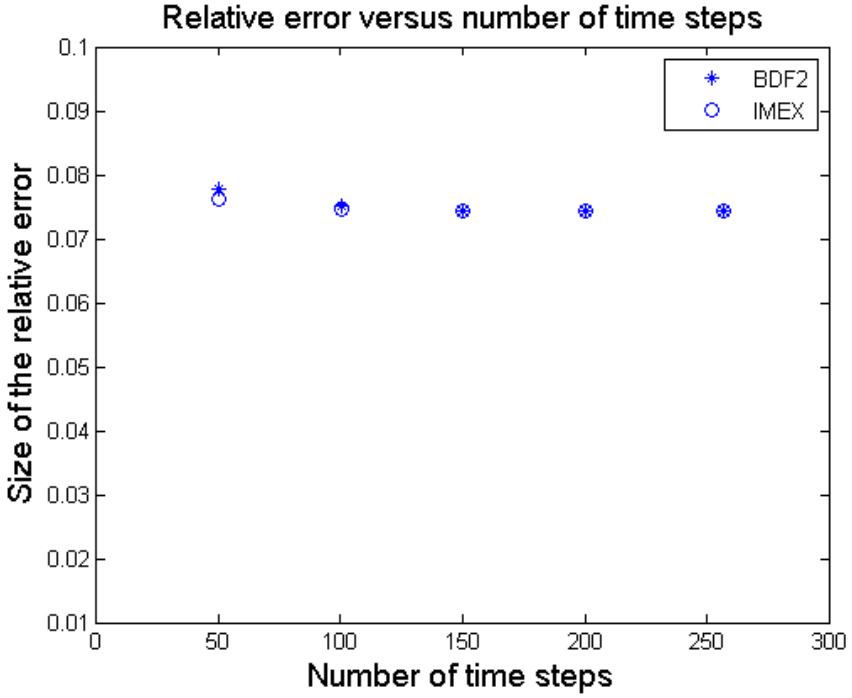

Figure 6. The relative error versus number of time steps for the IMEX- and BDF2 scheme

Figure 6 shows the relative error through time for both methods. The figure shows that both schemes produce approximately the same error for the same number of time steps.

When comparing the IMEX method to the BDF2 method we also want to know whether time has been saved without any loss of accuracy. The main plot of interest is therefore CPU time vs absolute error.

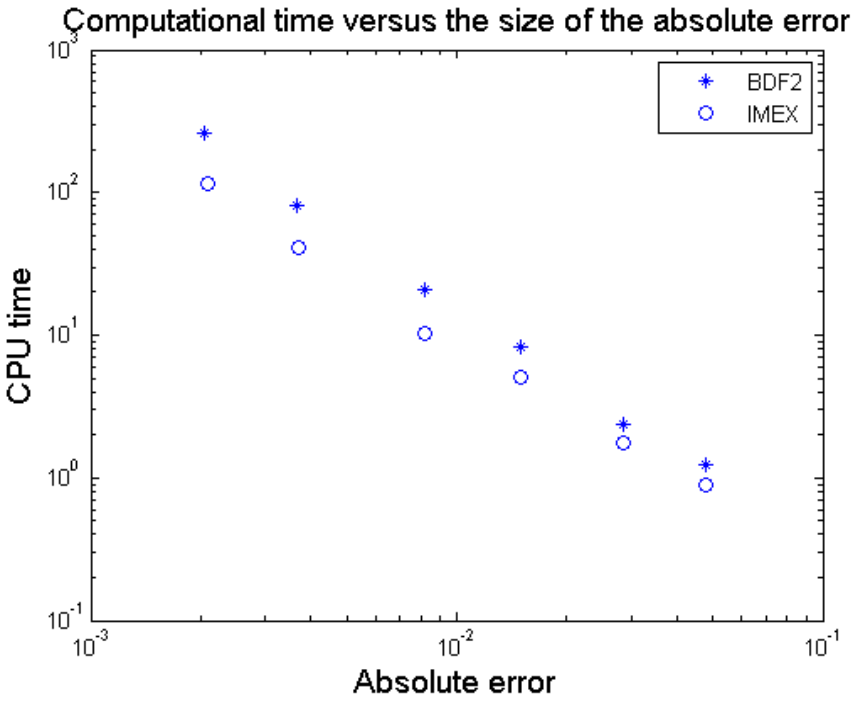

Figure 7 The CPU time versus the absolute error for the IMEX – and BDF2 Scheme.

In Figure 7, we compare the accuracy in the computed solution using the IMEX-scheme and BDF-2. It is clear from the figure that the IMEX scheme is faster than the BDF2 for the same error.

Conclusion

The results show that the adaptive method is faster than the equidistant method when we require a small error. However, if a larger error is accepted, the equidistant method eventually outperforms the adaptive method. This can be seen in Figure 5 going to the right in the figure. It can be explained from the fact that the adaptive method solves the problem twice. If the error is allowed to be big it might be enough to solve the problem on the sparse equidistant grid. Then solving it again on an adaptive grid will be excessive.

Choosing an IMEX- scheme instead of an implicit scheme turns out to save time without any loss of accuracy. This might raise the question: why bother using an implicit scheme in the first place? The answer is that the IMEX-scheme is not unconditionally stable. However, in this case it works fine to have a middle path between fully implicit and fully explicit which is what an IMEX scheme is.

References

[1] A.Sjöberg, Adaptive finite differences to price European options under the Bates model, Department of Scientific Computing September 3 2013

[2] S. Salmi and J. Toivanen, Robust and efficient IMEX schemes for option pricing under jump-diffusion models, Submitted to journal, (2013). Available at SSRN: http://dx.doi.org/10.2139/ssrn.2247916.

[3] R. Cont, E. Voltchkova, Finite difference methods for option pricing in jump-diffusion and exponential Lévy models, SIAM J. Numer. Anal. 43, (2005) pp. 1596–1626.

[4] Jari Toivanen. A componentwise splitting method for pricing american options under the bates model. In Applied and Numerical Partial Differential Equations, pages 213-227. Springer, 2010.

[5] Jonas Persson and Lina von Sydow. Pricing european multi-asset options using a space-time adaptive fd-method. Computing and Visualization in Science, 10(4):173-183, 2007.

[6] David S Bates. Jumps and stochastic volatility: Exchange rate processes implicit in deutsche mark options. Review of _nancial studies, 9(1):69-107, 1996.