UPPSALA
UNIVERSITET

TVE 16029 maj

Examensarbete 15 hp
Juni 2016

Smoothing of initial conditions for high order approximations in option pricing

Andreas Abrahamsson
Rasmus Pettersson

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Smoothing of initial conditions for high order approximations in option pricing

Andreas Abrahamsson and Rasmus Pettersson

In this article the Finite Difference method is used to solve the Black Scholes equation. A second order and fourth order accurate scheme is implemented in space and evaluated.
The scheme is then tried for different initial conditions. First the discontinuous pay off function of a European Call option is used. Due to the nonsmooth characteristics of the chosen initial conditions both schemes show an order of two. Next, the analytical solution to the Black Scholes is used when t=T/2. In this case, with a smooth initial condition, the fourth order scheme shows an order of four. Finally, the initial nonsmooth pay off function is modified by smoothing. Also in this case, the fourth order method shows an order of convergence of four.

Handledare: Lina von Sydow
Ämnesgranskare: Irina Dolguntseva
Examinator: Martin Sjödin
ISSN: 1401-5757, UPTEC F16 029

Contents

2 Populärvetenskaplig sammanfattning

Att prissätta optioner korrekt är naturligtvis av stor vikt för många aktörer på de finansiella marknaderna. I en tid där allt mer avancerade tillgångar handlas i ett allt snabbare tempo ställs det allt större krav på de värderingsmetoder som används.

I den här rapporten används finita differensmetoden för att prissätta en europeisk köpoption. Det arbete som krävs för att en finit differensmetod ska lösa en ekvation är beroende av storleken på det diskretiseringsnät som används. I och med den höga tidspress som ställs på dessa beräkningar, priserna på marknaden uppdateras kontinuerligt, används med fördel ett glesare nät för snabbare uträkningar. Nackdelen med ett glesare nät är att det ger en mindre korrekt lösning, det uträknade priset blir helt enkelt mer fel.

För att få en högre noggrannhet utan att öka beräkningsbehovet krävs det att den metod som används har en högre konvergensordning. Problemet med finita differenser och det "kantiga"initialvillkor som används för europeiska köpoptioner är att nogrannhetsordningen inte kan bli högre än två, oavsett vilken finit differensstencil som används. Lösningen på det är att runda till det kantiga initialvillkoret så att högre konvergensordningar kan uppnås.

3 Acknowledgments

We would like to thank our supervisor Lina von Sydow at the Division of Scientific Computing at Uppsala University for all help from everything between objective to project planning to practical implementations.

We would also like to thank Slobodan Milovanovic at the same department at Uppsala University for all help with supervision, theory and implementations.

4 Background

4.1 European Call options

The option market has become more and more important during the last decades. Options can both be used to hedge portfolios, moving or reducing risk, and leverage positions and modify risk. As a consequence financial derivatives are widely used and effective pricing methods are necessary.

The derivatives market contains a very wide spectrum of different products, with different underlying assets, different risks and different market projections. These products have different pay off functions and different structures. One of the most common, and basic, assets is a European Call option.

A European Call option gives the owner the right, but not the obligation, to buy an underlying asset S, such as a stock, for the strike price K at the expiration date T. Thus, at T the call option will have the value $S - K$. Since the owner has the right but not the obligation to exercise the call option, the option will only be used if $S - K$ is greater than 0, otherwise the owner would loose money on purpose. As a consequence the pay-off function, the value at T of a European call can be written $\Phi(S,T) = max\{S - K, 0\}$. In Figure 1, an example of a pay off function at final expiration date is shown as a function of S with a strike price $K = 100$ (Seydel 2012, p.1-3).

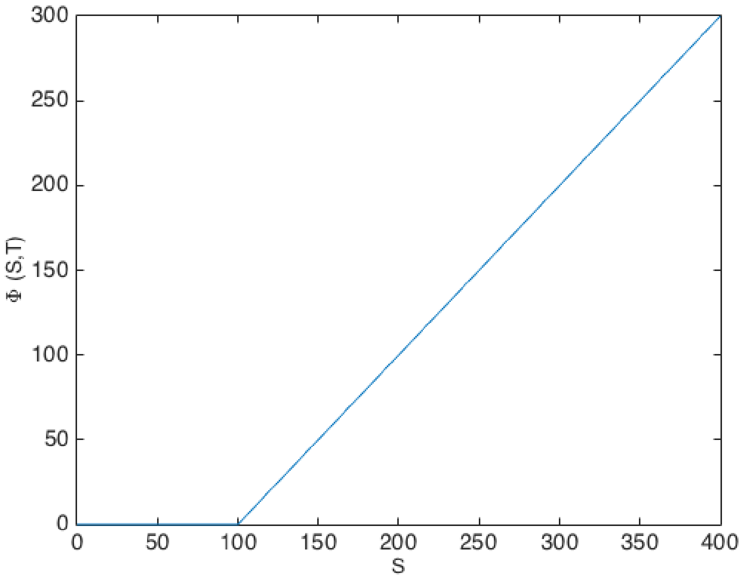


Figure 1: European call option with strike price 100

4.2 Black Scholes equation

In order to price a European Call option the Black Scholes equation, (1), is used. It is a differential equation with second and first order derivatives in space and first order derivative in time. It gives the value of an option V based on volatility σ, price of the underlying asset S, riskfree interest rate r and time until date of expiration T (Björk 2009, p.105),

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0. \tag{1}$$

The Black Scholes equation can be solved analytically and the solution for an European call option denoted C is provided in (2) (Björk 2009, p.105),

$$C(S,t) = N(d_1)S - N(d_2)Ke^{-r(T-t)}, \tag{2}$$

$$d_1 = \frac{1}{\sigma\sqrt{T-t}}[ln(\frac{S}{K}) + (r + \frac{\sigma^2}{2})(T-t)],$$

$$d_2 = d_1 - \sigma\sqrt{T-t}.$$

4.3 Central Finite Differences in space

The Finite Difference method is a numerical method for solving differential equations by approximating the derivatives. The method uses a grid in discretized space and time and the grid points are used to approximate the derivatives.

The first order Finite Difference can be straight forward translated from the definition of the derivative provided in (3). If one wants to take into account values on both sides of the current grid point Central Finite Differences can be used. The second order Central Finite Difference can be written as in (4),

$$f'(x_j) = \lim_{h\to 0}\frac{f(x_j+h) - f(x_j)}{h} \approx \frac{f(x_j+h) - f(x_j)}{h}, \tag{3}$$

$$f'(x_j) = \frac{\frac{1}{2}f(x_{j-1}) + \frac{1}{2}f(x_{j+1})}{h_x} + \mathcal{O}(h_x^2), \tag{4}$$

where f is the function to approximate, h_x is the grid size and $\mathcal{O}$ is a measurement of the error. The second order derivative with second order accuracy can be written as in (5),

$$f''(x_j) = \frac{f(x_{j-1}) - 2(x_j) + f(x_{j+1})}{h_x^2} + \mathcal{O}(h_x^2). \tag{5}$$

By Taylor expansions the first order derivative with fourth order accuracy can be written as in (6),

$$f'(x_j) = \frac{\frac{1}{12}f(x_{j-2}) - \frac{2}{3}(x_{j-1}) + \frac{2}{3}f(x_{j+1}) - \frac{1}{12}f(x_{j+2})}{h_x} + \mathcal{O}(h_x^4), \tag{6}$$

and finally the second order derivative with fourth order accuracy can be written as in (7),

$$f''(x_0) = \frac{-\frac{1}{12}f(x_{j-2}) - \frac{4}{3}(x_{j-1}) - \frac{5}{2}f(x_0) + \frac{4}{3}f(x_{j+1}) - \frac{1}{12}f(x_{j+2})}{h_x^2} + \mathcal{O}(h_x^4). \tag{7}$$

4.4 Backward Differentiation Formula in time

The Backward Differentiation Formulas are linear multisteps method that approximate the derivative of a specific function using information from already computed times. If the first time derivative is denoted as in (8) a general formula of the Backward Differentiation Formulas can be expressed as in (9) where h is the step size in time and a_k and β are order dependent coefficients,

$$y' = f(t, y), \ y(t^0) = y^0, \tag{8}$$

$$\sum_{k=0}^{s} a_k y^{n+k} = h\beta f(t^{n+s}, y^{n+s}). \tag{9}$$

The first order Backward Differentiation Formula, denoted BDF1 or Euler Backward method, is given by (10),

$$y^{n+1} - y^n = hf(t^{n+1}, y^{n+1}). \tag{10}$$

The second order Backward Differentiation Formula, BDF2 is given by (11),

$$y^{n+2} - \frac{4}{3}y^{n+1} + \frac{1}{3}y^n = \frac{2}{3}hf(t^{n+2}, y^{n+2}). \tag{11}$$

A scheme which uses BDF1 to approximate the first time step and BDF2 to approximate the following is still second order accurate in time.

4.5 The necessity of smoothing of initial conditions

Due to the kink at the strike price in the initial condition the convergence rate is lower, even for higher orders of Finite Differences (Tangman et al 2008), (Heston and Zhou 2000). The low order of convergence can be improved by either modifying the Finite Difference stencil or smoothing the initial condition (During et al 2015). In this report we consider the later approach.

Smoothing nonsmooth initial conditions based on averaging was introduced by Kreiss et al. in 1970. The smoothing is constructed by calculating the average value on the intervals over the function. The first order initial condition smoothing is straight forward and given in (12),

$$M_h^{(1)}v(x) = \frac{1}{dx}\int_{-h/2}^{h/2} v(x-y)dy, \tag{12}$$

where M_h is the smoothing operator working on the initial condition v on the interval $\{-h/2, h/2\}$ (Kreiss et al 1970).

However, in order to get higher orders of accuracy the implementation is not as straight forward. In general the smoothing operator is given by (13),

$$M_h^{(\mu)}v(x) = h^{-1}\int \chi_\mu(h^{-1}y)v(x-y)dy, \tag{13}$$

where μ is the order of accuracy, $v = \mu - 1$ when μ is odd and $v = \mu - 2$ when μ is even. χ is a piecewise polynomial of degree $\mu - 1$ which vanishes outside $\{-\mu + \frac{1}{2}, \mu - \frac{1}{2}\}$ for μ odd and $\{\mu + 1, \mu - 1\}$ for μ even (Kreiss et al 1970).

For the fourth order case in two dimensions the implementation of (13) is given by (14) where χ_4 is the inverse Fourier transform to (15), (Düring et al 2015),

$$M_h^{(4)}v(x_1, x_2) = h^{-2}\int_{-3h}^{3h}\int_{-3h}^{3h} \chi_4(\frac{x}{h})\chi_4(\frac{y}{h})v(x_1 - x, x_2 - y)dxdy. \tag{14}$$

$$\hat{\chi}_4(\omega) = (\frac{sin(\frac{\omega}{2})}{\frac{\omega}{2}})^4[1 + \frac{2}{3}sin^2(\frac{\omega}{2})]. \tag{15}$$

5 Objective

The objective of this report was to investigate how the order of convergence is dependent on smoothness of the initial conditions when solving the Black Scholes equation with higher order of Finite Differences. A spatial Finite Difference scheme was implemented and used together with a smoothing algorithm and a time discretization scheme, both implemented by Slobodan Milovanovic, Division of Scientific Computing, Uppsala University.

6 Method

In order to pursue the objective of this report a fourth order Finite Difference scheme was implemented. First the Finite Difference scheme was tested on an already smooth initial condition to make sure that it really is of fourth order. Then smoothing of the actual initial condition was implemented and the order of convergence was examined.

6.1 Finite Difference scheme

The equation to solve in order to price the European Call option is the Black Scholes equation, which can be seen in (1).

The algorithm for second order Central Finite Difference discretization is based on (Hirsa 2013, p.117) although the time discretization has been removed.

The Backward Differentiation Formulas are used to discretize the equation in time. The Central Finite Difference scheme is used to discretize the equation in space. By putting the time derivative alone on one side, the following equation is provided,

$$\frac{\partial V}{\partial t} = rV - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rS\frac{\partial V}{\partial S}. \tag{16}$$

Now the second and fourth order Finite Differences can be used in order to create the scheme.

The coefficients in front of the space derivatives are further denoted by α_j and β_j as in (17) and (18)

$$\alpha_j = \frac{\sigma^2 S_j^2}{2\Delta S^2}, \tag{17}$$

$$\beta_j = \frac{rS_j}{\Delta S}. \tag{18}$$

6.2 Second order Finite Difference scheme

The second order Central Finite Difference scheme is provided in (19),

$$\frac{\partial V_j}{\partial t} = rV_j - \alpha_j(V_{j-1} - 2V_j + V_{j+1}) - \beta_j(\frac{1}{2}V_{j-1} + \frac{1}{2}V_{j+1}), \tag{19}$$

which can be rewritten as (20), in order to create diagonal Finite Difference scheme,

$$\frac{\partial V_j}{\partial t} = (-\alpha_j + \frac{1}{2}\beta_j)V_{j-1} + (r + 2\alpha_j)V_j + (-\alpha_j - \frac{1}{2}\beta_j)V_{j+1}. \tag{20}$$

6.3 Fourth order Finite Difference scheme

By using the Finite Difference method the first and second derivative for the fourth order can be implemented. The scheme is shown in (21),

$$\begin{aligned}\frac{\partial V_j}{\partial t} &= rV_j - \alpha_j(-\frac{1}{12}V_{j-2} + \frac{4}{3}V_{j-1} - \frac{5}{2}V_j + \frac{4}{3}V_{j+1} - \frac{1}{12}V_{j+2}) \\ &- \beta_j(\frac{1}{12}V_{j-2} - \frac{2}{3}V_{j-1} + \frac{2}{3}V_{j+1} - \frac{1}{12}V_{j+2}),\end{aligned} \tag{21}$$

which we rewrite as in (22),

$$\begin{aligned}\frac{\partial V_j}{\partial t} &= (\frac{1}{12}\alpha_j - \frac{1}{12}\beta_j)V_{j-2} + (-\frac{4}{3}\alpha_j + \frac{2}{3}\beta_j)V_{j-1} - (-r + \frac{5}{2})V_j \\ &+ (-\frac{4}{3}\alpha_j - \frac{2}{3}\beta_j)V_{j+1} + (\frac{1}{12}\alpha_j + \frac{1}{12}\beta_j)V_{j+2}.\end{aligned} \tag{22}$$

6.4 Matrices

When implementing the second order central Finite Difference scheme in Matlab the following matrix is used

$$cd\_scheme_{2nd} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ l_2 & d_2 & u_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & l_3 & d_3 & u_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & l_4 & d_4 & u_4 & 0 & 0 & 0 \\ . & & & . & . & . & & \\ . & & & & . & . & . & \\ . & & & & & l_{n-1} & d_{n-1} & u_{n-1} \\ 0 & . & . & . & . & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{aligned} l_j &= -\alpha_j + \frac{1}{2}\beta_j, \\ d_j &= r + 2\alpha_j, \\ u_j &= -\alpha_j - \frac{1}{2}\beta_j. \end{aligned}$$

We approximate Dirichlet boundary conditions. The left hand side can be approximated with $V_0 = 0$ and the right hand side can be approximated with the discounted pay off function for S_{max} (Hirsa 2013, p.122). The boundary terms are shown in (23),

$$V(0, t) = 0, \qquad V(S_{max}, t) = S_{max} - Ke^{-r(T-t)}. \tag{23}$$

For the fourth order Finite Difference scheme the matrix looks like

$$cd\_scheme_{4th} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ l_2 & l_1 & d & u_1 & u_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & l_2 & l_1 & d & u_1 & u_2 & 0 & 0 & 0 & 0 \\ . & & . & . & . & . & & & & \\ . & & & . & . & . & . & & & \\ . & & & & . & . & . & . & & \\ . & & & & & l_2 & l_1 & d & u_1 & u_2 \\ 0 & . & . & . & . & . & . & . & 0 & 0 \\ 0 & . & . & . & . & . & . & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{aligned} l_2 &= \frac{1}{12}\alpha_j - \frac{1}{12}\beta_j, \\ l_1 &= -\frac{4}{3}\alpha_j + \frac{2}{3}\beta_j, \\ d &= -r + \frac{5}{2}, \\ u_1 &= -\frac{4}{3}\alpha_j - \frac{2}{3}\beta_j, \\ u_2 &= \frac{1}{12}\alpha_j + \frac{1}{12}\beta_j. \end{aligned}$$

For the fourth order Finite Difference scheme we approximate the boundary conditions in the same way as in the second order case although we now, due to the larger Finite Difference stencil, also have to approximate the first inner points. These are shown in (24) and (25),

$$V(0,t) = 0, \quad V(\Delta S, t) = 0, \tag{24}$$

$$V(S_{max}, t) = S_{max} - Ke^{-r(T-t)}, \quad V(S_{max} - \Delta S, t) = (S_{max} - \Delta S) - Ke^{-r(T-t)}. \tag{25}$$

6.5 Time discretization

The following implementation was done by Slobodan Milovanovic at Division of Scientific Computing, Uppsala University. In the first time step BDF1 is used. Let the developed *cd_scheme* = W and thus the first time step is discretized as in (26) - (30),

$$\frac{\partial V^{n-1}}{\partial t} = WV^{n-1}, \tag{26}$$

$$\frac{V^n - V^{n-1}}{\Delta t} = WV^{n-1}, \tag{27}$$

$$V^n - V^{n-1} = WV^{n-1}\Delta t, \tag{28}$$

$$V^n = WV^{n-1}\Delta t + V^{n-1} = (I + W\Delta t)V^{n-1} = AV^{n-1}. \tag{29}$$

This gives (24) which can be solved for V^{n-1},

$$AV^{n-1} = V^n. \tag{30}$$

The second time step uses BDF2 and is discretized as in (31) - (35),

$$\frac{\partial V^{n-2}}{\partial t} = WV^{n-2}, \tag{31}$$

$$V^{n-2} + \frac{2}{3}\Delta t WV^{n-2} = \frac{4}{3}V^{n-1} - \frac{1}{3}V^n, \tag{32}$$

$$V^{n-2} + \frac{2}{3}\Delta t WV^{n-2} = (I + \frac{2}{3}\Delta t W)V^{n-2} = AV^{n-2}, \tag{33}$$

$$\frac{4}{3}V^{n-1} - \frac{1}{3}V^n = b, \tag{34}$$

(29) can be solved which gives the solution for V^{n-2},

$$AV^{n-2} = b. \tag{35}$$

6.6 Smoothing of initial conditions

In this article the smoothing algorithm described by (Kreiss et al. 1970) is used, the derivation in the fourth order case is built on (Düring et al 2015) as described in Section 4.5 and the implementation was done by Slobodan Milovanovic, Division of Scientific Computing, Uppsala University.

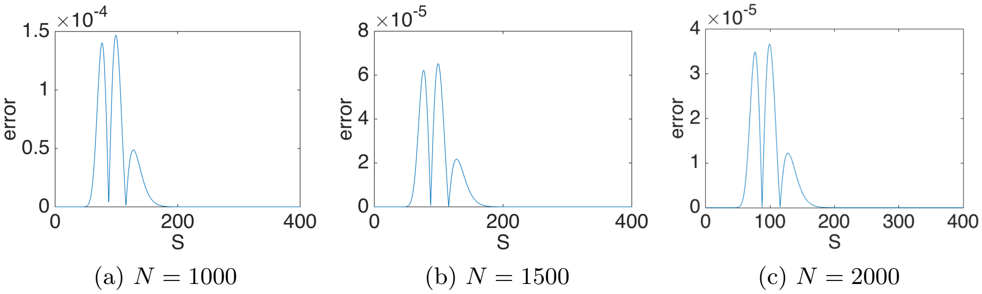
Since the problem is formulated in one dimension (14) is modified to (36)

$$M_h^{(4)}v(x) = h^{-1}\int_{-3h}^{3h} \chi_4(\frac{x}{h})v(x-y)dy. \tag{36}$$

7 Results

7.1 Second order Finite Difference scheme

In Figure 2 we show the results from the second order Finite Difference scheme, described in Section 5.2, provided. The initial condition is the original nonsmooth pay off function. The number of time steps $M = 20000$ is large to make the error dependent on the space discretization. The number of spatial grid points is denoted by N.



(a) $N = 1000$ (b) $N = 1500$ (c) $N = 2000$

Figure 2: Error as a function of S

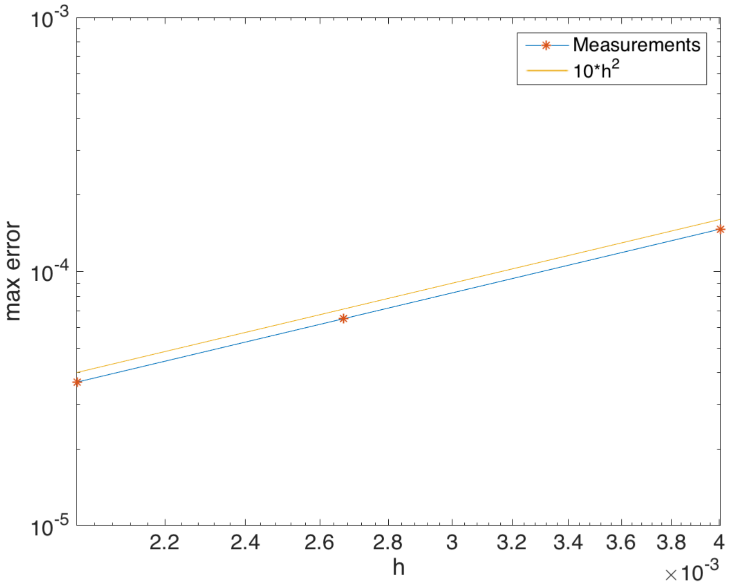


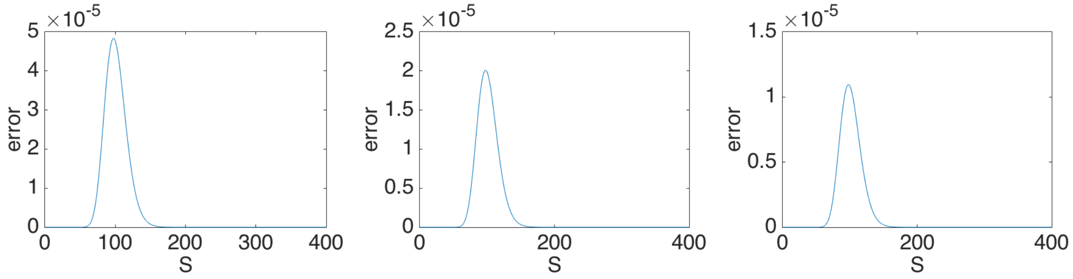
Figure 3: Logarithmic plot of maximum error as a function spatial grid size

As it can be seen in Figure 2 the error is dominant around the strike price and decreasing for finer grids.

The maximum error in space is found by measuring the maximum difference between the calculated result and the analytical solution to the Black Scholes equation and the results are shown in Figure 3. As it can be seen, the estimated order of convergence is 2.

7.2 Fourth order Finite Difference scheme on nonsmooth initial condition

The fourth order Finite Difference scheme is run with the original nonsmooth pay off. As previously, the number of time steps is set to $M = 20000$.



(a) $N = 1000$ (b) $N = 1500$ (c) $N = 2000$

Figure 4: Error as a function of S

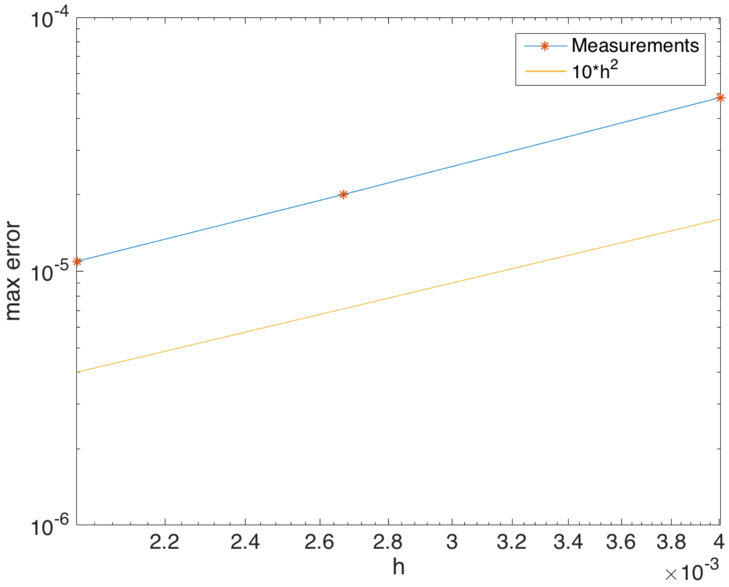
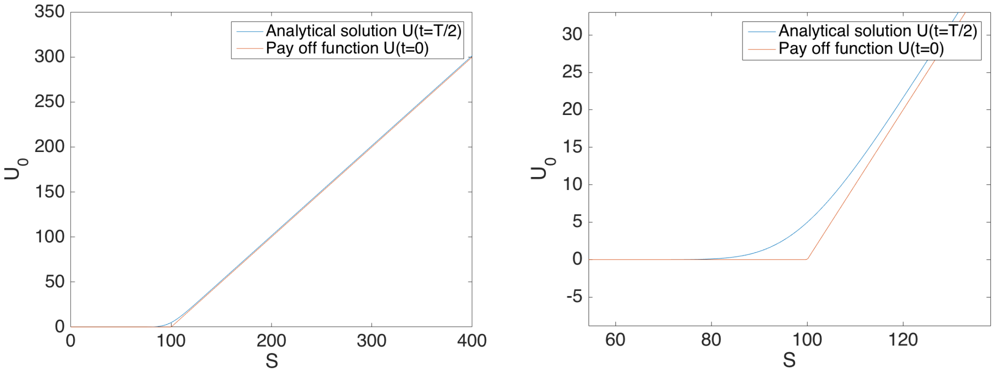


Figure 5: Logarithmic plot of maximum error as a function spatial grid size

In Figure 4 the error over the space domain is shown for different grid sizes, $N = 1000, 1500, 2000$. In Figure 5 a convergence plot is shown for the measured maximum error. As can be seen in Figure 4 the error is dominant around the strike price and decreasing for finer grids. In Figure 5 the convergence is estimated to 2.

7.3 Fourth order Finite Difference scheme on already smooth initial condition

To measure the order of convergence on an already smooth initial conditions we set the initial condition to the analytical solution of the Black Scholes equation for $t = T/2$. The initial condition, $u(T/2)$, is provided together with the actual pay off in Figure 6.

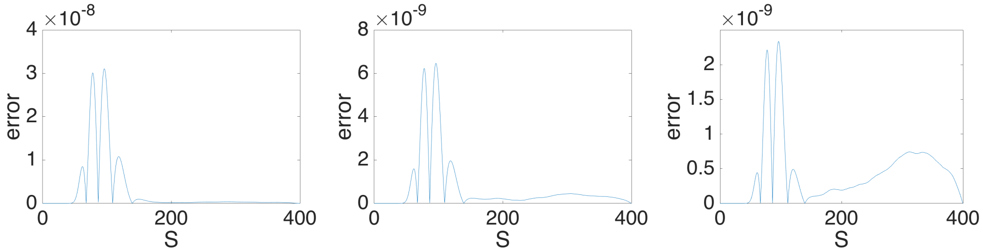


(a) The smooth initial condition ($U^0 = U_a(T/2)$) and the pay off function

(b) Initial condition zoomed in around the strike price

Figure 6: The smooth initial condition

The simulation is then run until $t = T$ for different space grid sizes.



(a) $N = 1000$

(b) $N = 1500$

(c) $N = 2000$

Figure 7: Error as a function of S

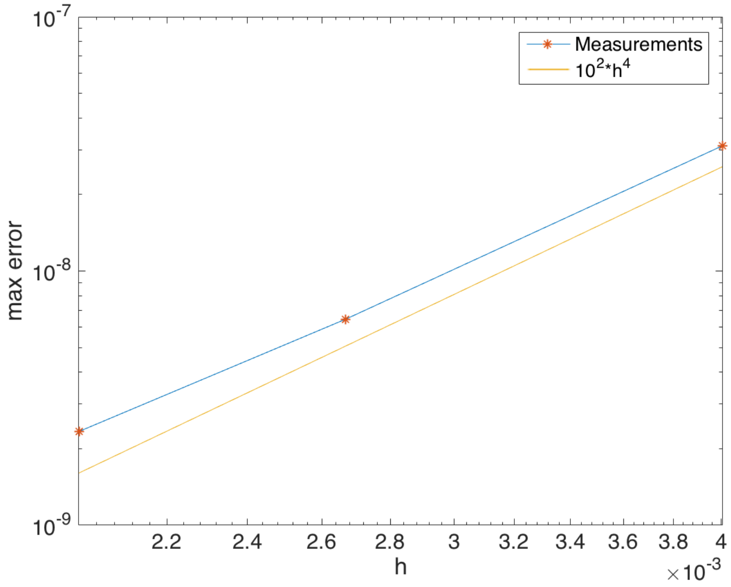
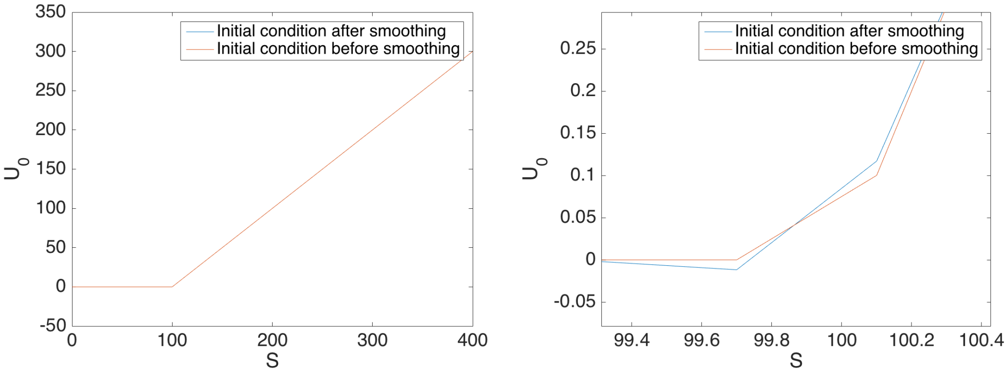


Figure 8: Logarithmic plot of maximum error as a function spatial grid size

In Figure 7 the error is shown over the space domain for different grid sizes, analogical to section 6.2. In the same way a convergence plot is shown in Figure 8 with the measured values. The convergence for the already smooth initial condition is measured to 3.7. However, a error emerges near the right boundary, this error is constant regardless of the number of space steps.

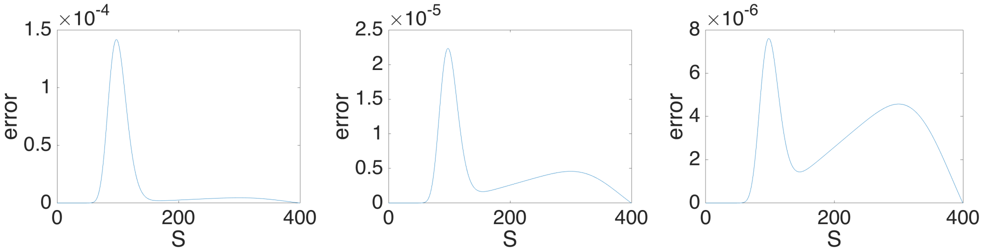
7.4 Fourth order Finite Difference scheme on smoothed initial condition

Now we let the initial condition be the smoothed pay off function. In Figure 9 the smoothed initial condition is shown together with the pay off function.



(a) Smoothing of the pay off function (b) Smoothing zoomed in around strike price

Figure 9: The payoff function before and after the smoothing of the initial condition, to the right is a close up on the strike price



(a) $N = 1000$ (b) $N = 1500$ (c) $N = 2000$

Figure 10: Error as a function of S

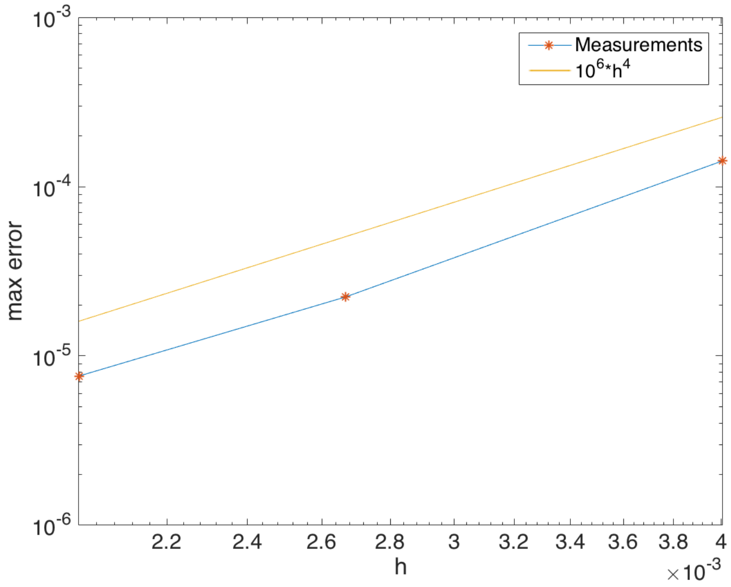


Figure 11: Logarithmic plot of maximum error as a function spatial grid size

In Figure 10, the error in space domain is shown for different number of space grid steps. As in Section 6.3, the error around the strike price decreases with a convergence rate of order 4.2. However the same error as previously seen, close to the right boundary, is constant independently of the number of space steps.

8 Discussion

As can be seen in the result section we have second order of convergence for the second order Finite Difference scheme.

It can also be seen that the order of convergence for the nonsmooth initial condition is 2 even though the scheme is of fourth order. This is according to theory.

The fourth order scheme shows a fourth order of convergence between a very specific amount of grid sizes. The error around pay off continues to converge with order 4 but a error from the boundary gets dominant.

This error is probably due to our Finite Difference scheme, since it occurs for both smooth and smoothed initial conditions. It is probably caused by our strong implementation of the first inner boundary points and in further studies, a skew matrix with backward and forward Finite Differences for the first inner points should be used instead.

9 Conclusion

In order to investigate how the order of convergence is dependent on smoothness of the initial conditions when solving the Black Scholes equation with higher order of Finite Differences two Finite Difference schemes of second and fourth order were implemented. The second order scheme shows second order convergence regardless of the smoothness of the initial condition. The fourth order scheme does not show a fourth order convergence without smoothing of the initial condition. However, when the smoothing algorithm is implemented the results show that the fourth order scheme gives a convergence order of four and thus the smoothing algorithm works. It can be concluded that when fourth order Finite Difference schemes are used, a smooth initial conditions is necessary to show correct convergence.

10 References

Heston and Zhou, *On the rate of convergence of discrete-time contingent claims*, Mathematical Finance, Vol. 10, No. 1, (January 2000)

Kreiss, Thomée and Widlund, *Smoothing of Initial Data and Rates of Convergence for Parabolic Difference Equations*, Communications on pure and applied mathematics, Vol. XXIII, 241-259, (1970)

Tangman and Buruth, *Numerical pricing of options using high-order compact finite difference schemes*, Journal of Computational and Applied Mathematics, 218, (2008)

Düring and Heuer, *High-order compact schemes for parabolic problems with mixed derivatives in multiple space dimensions*, SIAM Journal on Numerical Analysis 53:5, 2113-2134, (2015)

Hirsa, *Computational Methods in Finance*, London: Chapman & Hall/CRC Financial Mathematics Series, (2013)

Björk, *Arbitrage Theory in Continuous Time, Third Edition*, Oxford University Press, (2009)

Seydel, *Tolls for Computational Finance, Fifth Edition*, London: Springer, (2012)

11 Appendix

Matlab 2015

11.1 Second order cd_scheme

```
s=x;
ds=dx;
sigma=sig;
gamma=1;

alpha=0.5*sigma^2*s.^2/(ds^2);
beta=0.5*r*s/ds;
l=beta-alpha;
d=r+2*alpha;
upper=-(alpha+beta);

cd_scheme=diag(d)+diag(l(2:end),-1)+diag(upper(1:end-1),1);
cd_scheme(1,:)=zeros(1,N);
cd_scheme(end,:)=zeros(1,N);
W=cd_scheme;
```

11.2 Fourth order cd_scheme

```
s=x;
ds=dx;
sigma=sig;
gamma=1;

alpha=0.5*sigma^2*s.^2/(ds^2);
beta=r*s/ds;
l2=1/12*alpha-1/12*beta;
l1=-4/3*alpha+2/3*beta;
d=r+5/2*alpha;
upper1=-4/3*alpha-2/3*beta;
upper2=1/12*alpha+1/12*beta;

cd_scheme=diag(l2(3:end),-2)+diag(l1(2:end),-1)+diag(d)+...
diag(upper1(1:end-1),1)+diag(upper2(1:end-2),2);
cd_scheme(1,:)=zeros(1,N);
cd_scheme(2,:)=zeros(1,N);
cd_scheme(end-1,:)=zeros(1,N);
cd_scheme(end,:)=zeros(1,N);
W=cd_scheme;
```