University of Amsterdam

Assignment 3: The finite difference Methode

Computational Finance

Group 3:
Chaitanya Kumar - 13821369 -
MSc. Computational Science
Nadav Levi - 11806990
MSc. Computational Science
Laurens Prast - 11297018
MSc. Computational Science

March 28, 2022

1 Introduction

In our previous report we studied how we could price vanilla options analytically with the Black-Scholes equation and numerically with the Binomial tree method and the Monte Carlo method. In this report we will discuss another method to price options called the Finite Difference (FD) method to solve the Black-Scholes PDE. We will use two different schemes called the FTCS (Forward in Time Central in Space) scheme and the CN (Crank-Nicolson) scheme. The advantage of FD method is that for small dimensions problems, the FD method may be faster than Monte Carlo. It can also handle more exotic options with early exercise options and complex boundaries and barriers. For some of the Greeks the FD method is very handy. The most important disadvantage is that the method is not feasible for high dimensional problem (≥ 4). We will also discuss another method to calculate the option price called the COS method, which is a relativalet new method. This method uses the Fourier cosine series and transform. The main advantage of this method is that is fast and efficient, but the characteristic function must be known.

The structure of this report will be as following: we start with the theoratical backgound in the methods section. Afterwards will show the results of the FD and the COS method in the results section. In this section we will also compare our results with the analytical Black-Scholes and determine the hedge parameter Δ with the FD method. Lastly, we will discuss our results that we obtained in the the discussion section.

2 Methods

2.1 The Black-Scholes PDE

The BS-PDE for a plain vanilla option, assuming a geometric brownian motion process for the underlying asset price, is given by

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

where $V(S,t)$ is the option value at time t and asset/stock price $S = S(t)$, r is the risk-free interest rate and σ is the volatility in the stock price. When pricing e.g. a European call option using (1), we have the following boundary conditions:

1. At time $t = T$, i.e. at maturity, the value of the option is given by the payoff i.e.

$$V(S,T) = (S-K)^+, \tag{2}$$

   where K is the strike price.

2. $\lim_{S\to\infty} V(t) = \lim_{S\to\infty} S - K\exp(-rT)$. If we assume $S \leq S_{\max}$ for some very large $S_{\max}$, this boundary condition can be approximated as

$$V(S = S_{\max}, t) \approx S_{\max} \tag{3}$$

3. The option is worthless for $S = 0$. So, we have

$$V(S = 0, t) = 0 \tag{4}$$

The BS-PDE (1) along with the boundary conditions (2), (3) and (4) can be used to solve for the option value $V(t)$ numerically. In its current form, this is a boundary-value problem (BVP). In practice, the equation is usually transformed into a linear PDE with constant coefficients before numerical being solved numerically. The time coordinate is also transformed so as to transform the BVP to an initial value problem (IVP).

2.2 Transformed BS-PDE

We start with the transformation $\tau = T - t$, so that (1) becomes

$$\frac{\partial V}{\partial \tau} = rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV \tag{5}$$

We can eliminate the S-dependent coefficients by using the coordinate transformation $X = \ln S$, so that

$$\frac{\mathrm{d}X}{\mathrm{d}S} = \frac{1}{S}$$
$$\implies \frac{\partial V}{\partial S} = \frac{\partial V}{\partial X}\frac{\mathrm{d}X}{\mathrm{d}S} = \frac{1}{S}\frac{\partial V}{\partial X},$$

and

$$\begin{aligned}
\frac{\partial^2 V}{\partial S^2} &= \frac{\partial}{\partial S}\left(\frac{1}{\exp(X)}\frac{\partial V}{\partial X}\right) \\
\implies &= \frac{\partial}{\partial X}\left(\frac{1}{\exp(X)}\frac{\partial V}{\partial X}\right)\frac{\mathrm{d}X}{\mathrm{d}S} \\
\implies &= \frac{1}{S}\left(-\frac{1}{S}\frac{\partial V}{\partial X} + \frac{1}{S}\frac{\partial^2 V}{\partial S^2}\right) \\
\implies &= \frac{1}{S^2}\left(\frac{\partial^2 V}{\partial X^2} - \frac{\partial V}{\partial X}\right).
\end{aligned}$$

Thus, we get

$$S\frac{\partial V}{\partial S} = \frac{\partial V}{\partial X}, \tag{6}$$

$$S^2\frac{\partial^2 V}{\partial S^2} = \frac{\partial^2 V}{\partial X^2} - \frac{\partial V}{\partial X}. \tag{7}$$

Substitute (6) and (7) in (5) to get

$$\begin{aligned}
\frac{\partial V}{\partial \tau} &= r\frac{\partial V}{\partial X} + \frac{1}{2}\sigma^2\left(\frac{\partial^2 V}{\partial X^2} - \frac{\partial V}{\partial X}\right) - rV \\
\implies \frac{\partial V}{\partial \tau} &= \left(r - \frac{1}{2}\sigma^2\right)\frac{\partial V}{\partial X} + \frac{1}{2}\sigma^2\frac{\partial^2 V}{\partial X^2} - rV.
\end{aligned} \tag{8}$$

The boundary conditions (2), (3), and (4) then transform to the following in the new coordinate system.

1. $t = T \implies \tau = T - t = 0$, so (2) becomes

$$V(X, 0) = (\exp(X) - K)^+ \tag{9}$$

2. Let $X_{\max} = \ln S_{\max}$, so that (3) becomes

$$V(X_{\max}, \tau) \approx \exp(X_{\max}) \tag{10}$$

3. (4) cannot be transformed as-is because 0 is not in the domain of ln. So, we make another approximation similar to the one made in (10); let $S_{\min} > 0$ be some very low asset price value that is practically impossible to reach. The resulting premium is negligible, i.e.

$$V(X_{\min}, \tau) \approx 0, \tag{11}$$

where $X_{\min} = \ln S_{\min}$.

For convenience we can allow the transformation to be $X = \frac{S}{S_{\min}}$ so that (11) would return to its original form as $V(0, \tau) \approx 0$. The transformed equation remains unchanged as $\frac{\mathrm{d}\ln(\alpha S)}{\mathrm{d}S} = \frac{\mathrm{d}\ln S}{\mathrm{d}S} = 1/S$ for any $\alpha > 0$.

2.3 Basic FDMs

The constant coefficient linear PDE (8) is amenable finite difference schemes that work with uniform discretisation of the X and τ axes. We explore two of them: the forward in time and central in space (FTCS), and the Crank-Nicolson (CN) scheme.

The first step in the process is discretisation of the X and τ coordinates. More sophisticated techniques would use adaptive discretisation, but for the scope of this assignment confine ourselves to uniform discretisation. We take M points on the X axis a uniform distance apart, i.e. we construct this set of points as

$$X_i = X_{\min} + (i-1)\Delta x \quad \forall i \in \{1, \dots M\}$$

where $\Delta x = \frac{X_{\max} - X_{\min}}{M-1}$. Similarly the τ coordinate is discretised into N uniformly distributed points

$$\tau_n = n\Delta\tau \quad \forall n \in \{0 \dots N-1\},$$

where $\Delta\tau = T/(N-1)$.

2.3.1 FTCS Scheme

The forward difference approximation of the derivative of a differentiable function is

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}, \quad h > 0, \tag{12}$$

while the central difference approximation is given by

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}, \quad h > 0. \tag{13}$$

As the name of the scheme suggests, the partial derivatives with respect to time are approximated using (12), while the partial derivatives with respect to the "spatial coordinate" X are approximated using (13).

These approximations are direct results of Taylor's theorem. Let $v_j^n = V(X_j, \tau_n)$, $\Delta x = X_{j+1} - X_j$, and $\Delta\tau = \tau_{n+1} - \tau_n$. Also let $(v_j^n)' = \frac{\partial V}{\partial X}\big|_{X=X_j, \tau=\tau_n}$. We have the following equations from Taylor's theorem.

$$v_{j+1}^n = v_j^n + \Delta x (v_j^n)' + \frac{\Delta x^2}{2}(v_j^n)'' + \mathcal{O}\left(\Delta x^3\right) \tag{14}$$

$$v_{j-1}^n = v_j^n - \Delta x (v_j^n)' + \frac{\Delta x^2}{2}(v_j^n)'' + \mathcal{O}\left(\Delta x^3\right) \tag{15}$$

Subtract (15) from (14) to get

$$\begin{aligned} v_{j+1}^n - v_{j-1}^n &= 2\Delta x (v_j^n)' + \mathcal{O}\left(\Delta x^3\right). \\ \implies (v_j^n)' &= \frac{v_{j+1}^n - v_{j-1}^n}{2\Delta x} + \mathcal{O}\left(\Delta x^2\right), \end{aligned} \tag{16}$$

which is the central difference approximation shown in (13). We can see from (16) that the convergence of the central difference is second order. However in practice Δx cannot be arbitrarily small - the optimum value of Δx is usually of the order of $\sqrt{\varepsilon_{\text{mach}}}$, where $\varepsilon_{\text{mach}}$ is the machine epsilon $\approx 10^{-8}$ for IEEE 754 double precision numbers.

We then add (14) and (15) to get the central difference approximation for the second partial derivative with respect to X.

$$\begin{aligned} v_{j+1}^n + v_{j-1}^n &= 2v_j^n + \Delta x^2 \left(\frac{\partial^2 2}{\partial V \partial X}\right)_j^n + \mathcal{O}(\Delta x^3) \\ \implies \left(\frac{\partial^2 V}{\partial X^2}\right)_j^n &= \frac{v_{j+1}^n - 2v_j^n + v_{j-1}^n}{\Delta x^2} + \mathcal{O}\left(\Delta x^2\right) \end{aligned} \tag{17}$$

Since this PDE is linear and the truncation errors in both derivatives (17) and (18) is $O(\Delta x^2)$, the convergence of the overall scheme is second order in space.

Similarly, we can use Taylor's theorem for approximating the time partial derivative $\frac{\partial V}{\partial \tau}$ using forward difference.

$$\begin{aligned} v_j^{n+1} &= v_j^n + \Delta\tau \left(\frac{\partial V}{\partial \tau}\right)\Big|_{X_j, \tau_n} + \mathcal{O}\left(\Delta\tau^2\right) \\ \implies \left(\frac{\partial V}{\partial \tau}\right)_j^n &= \frac{v_j^{n+1} - v_j^n}{\Delta\tau} + \mathcal{O}(\Delta\tau) \end{aligned} \tag{18}$$

(18) shows that the forward difference approximation of the first derivative has an order of convergence of $\mathcal{O}(\Delta\tau)$. Again, this too cannot be arbitrarily small and is limited by the machine epsilon of the floating point numbers in use.

Substituting (18), (17), and (16) into the transformed BS-PDE (8), we get

$$\frac{v_j^{n+1} - v_j^n}{\Delta\tau} = \left(r - \frac{1}{2}\sigma^2\right)\frac{v_{j+1}^n - v_{j-1}^n}{2\Delta x} + \frac{1}{2}\sigma^2 \frac{v_{j+1}^n - 2v_j^n + v_{j-1}^n}{\Delta x^2} - r v_j^n \tag{19}$$

$$\implies v_j^{n+1} = \left(r - \frac{1}{2}\sigma^2\right)\frac{\Delta\tau}{2\Delta x}\left(v_{j+1}^n - v_{j-1}^n\right) + \frac{\Delta\tau}{2\Delta x^2}\sigma^2\left(v_{j+1}^n - 2v_j^n + v_{j-1}^n\right) - r\Delta\tau v_j^n + v_j^n \tag{20}$$

2.3.2 Crank-Nicolson scheme

The CN scheme is an implicit method method that is constructed as a linear combination of a forward-Euler method (explicit) and its corresponding backward Euler method (implicit). (19) is a forward-Euler method. Rearranging the terms on the RHS to get

$$\frac{v_j^{n+1} - v_j^n}{\Delta\tau} = \frac{1}{2\Delta x}\left(r - \frac{\sigma^2}{2} + \frac{\sigma^2}{\Delta x}\right)v_{j+1}^n - \left(\frac{\sigma^2}{\Delta x^2} + r\right)v_j^n + \frac{1}{2\Delta x}\left(\frac{\sigma^2}{\Delta x} - r + \frac{\sigma^2}{2}\right)v_{j-1}^n. \tag{21}$$

Denote the RHS by F_j^n. The corresponding backward-Euler scheme can be constructed by letting the RHS in (21) be F_j^{n+1} i.e.

$$F_j^{n+1} = \frac{1}{2\Delta x}\left(r - \frac{\sigma^2}{2} + \frac{\sigma^2}{\Delta x}\right)v_{j+1}^{n+1} - \left(\frac{\sigma^2}{\Delta x^2} + r\right)v_j^{n+1} + \frac{1}{2\Delta x}\left(\frac{\sigma^2}{\Delta x} - r + \frac{\sigma^2}{2}\right)v_{j-1}^{n+1}. \tag{22}$$

In the CN scheme, we take the average of the explicity and implicity methods.

$$\begin{aligned}\frac{v_j^{n+1} - v_j^n}{\Delta\tau} &= \frac{1}{2}\left(F_j^n + F_j^{n+1}\right)\\ \implies v_j^{n+1} &= v_j^n + \frac{\Delta\tau}{2}\left(F_j^n + F_j^{n+1}\right)\end{aligned} \tag{23}$$

Consider F_j^n in (21). We can rewrite it as

$$F_j^n = \frac{1}{2\Delta x}\left(r - \frac{\sigma^2}{2}\right)\left(v_{j+1}^n - v_{j-1}^n\right) + \frac{\sigma^2}{2\Delta x^2}\left(v_{j+1}^n + v_{j-1}^n - 2v_j^n\right) - rv_j^n. \tag{24}$$

Similarly, F_j^{n+1} can also be rewritten as

$$F_j^{n+1} = \frac{1}{2\Delta x}\left(r - \frac{\sigma^2}{2}\right)\left(v_{j+1}^{n+1} - v_{j-1}^{n+1}\right) + \frac{\sigma^2}{2\Delta x^2}\left(v_{j+1}^{n+1} + v_{j-1}^{n+1} - 2v_j^{n+1}\right) - rv_j^{n+1}. \tag{25}$$

Substitute (24) and (25) in (23) to get

$$\begin{aligned}v_j^{n+1} = v_j^n &+ \left(r - \frac{\sigma^2}{2}\right)\frac{\Delta\tau}{4\Delta x}\left(v_{j+1}^n - v_{j-1}^n + v_{j+1}^{n+1} - v_{j-1}^{n+1}\right)\\ &+ \frac{1}{4}\sigma^2\frac{\Delta\tau}{\Delta x^2}\left(v_{j+1}^n - 2v_j^n + v_{j-1}^n + v_{j+1}^{n+1} - 2v_j^{n+1} + v_{j-1}^{n+1}\right) - \frac{r\Delta\tau}{2}\left(v_j^n + v_j^{n+1}\right)\end{aligned} \tag{26}$$

Again, since the PDE is linear and we showed using Taylor's theorem that the finite difference approximations of the derivatives have a truncation error $\in O(\Delta x^2)$, the convergence of the Crank-Nicolson scheme is also $O(\Delta x^2)$.

2.4 Numerical Considerations

(26) and (20) are linear relations; numerical linear algebra can be used for solving them.

2.4.1 Explicit FTCS Scheme

Consider (20). We can rewrite it like so:

$$v_j^{n+1} = \frac{\Delta\tau}{2\Delta x}\left(\frac{\sigma^2}{\Delta x} - \left(r - \frac{\sigma^2}{2}\right)\right)v_{j-1}^n + \left(1 - r\Delta\tau - \frac{\sigma^2\Delta\tau}{\Delta x^2}\right)v_j^n + \frac{\Delta\tau}{2\Delta x}\left(\left(r - \frac{\sigma^2}{2} + \frac{\sigma^2}{\Delta x}\right)\right)v_{j+1}^n \tag{27}$$

$$\implies v_j^{n+1} = k_l v_{j-1}^n + k_m v_j^n + k_u v_{j+1}^n, \tag{28}$$

where

$$\begin{aligned}k_l &= \frac{\Delta\tau}{2\Delta x}\left(\frac{\sigma^2}{\Delta x} - \left(r - \frac{\sigma^2}{2}\right)\right),\\ k_m &= 1 - r\Delta\tau - \frac{\sigma^2\Delta\tau}{\Delta x^2}, \quad \text{and}\\ k_u &= \frac{\Delta\tau}{2\Delta x}\left(\left(r - \frac{\sigma^2}{2}\right) + \frac{\sigma^2}{\Delta x}\right).\end{aligned}$$

k_l, k_m, and K_u are the lower subdiagonal, main diagonal, and upper subdiagonal entries of a tridiagonal matrix. Along with (28) we also have the boundary conditions

$$\begin{aligned} v_1^n &= 0 \\ v_M^n &= S_{\max} - K\exp\{(-r\cdot n\Delta\tau)\} \\ v_j^0 &= (S_0 - K)^+ \end{aligned} \tag{29}$$

Therefore, using (29) and (28), we can express the FTCS scheme like so. Therefore we can write the explicit FTCS scheme in from (29), the following formand (28)

$$\begin{bmatrix} v_1^{n+1} \\ v_2^{n+1} \\ v_3^{n+1} \\ \vdots \\ v_{M-1}^{n+1} \\ v_M^{n+1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ k_l & k_m & k_u & 0 & \dots & 0 \\ 0 & k_l & k_m & k_u & \dots & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & & \dots & k_l & k_m & k_u \\ 0 & & \dots & & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1^n \\ v_2^n \\ v_3^n \\ \vdots \\ v_{M-1}^n \\ v_M^n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ v_M^0 \end{bmatrix} \tag{30}$$

2.4.2 Implicit FTCS

Explicit schemes are generally only conditionally stable, and stability imposes some constraints on the values of $\Delta\tau$ and Δx relative to each other and to $T, S_{\min}, S_{\max}$.

Implicit methods on the other hand are generally unconditionally stable, but result in a greater FLOPs count. The choice between implicit and explicit schemes is a trade-off between speed and accuracy.

The implicit FTCS scheme can be constructed in two ways.

1. Do not do the coordinate transformation in the $t \to \tau = T - t$, or
2. Use the implicit part of the CN scheme (as in (22))

We found item 1 attractive because by not doing the time coordinate transformation we directly get the implicit scheme with no extra work. As such both schemes will be numerically equivalent, i.e. identical with respect to FLOPs count, convergence and stability.

(20) without the backward time transformation becomes

$$v_j^{n+1} = v_j^n + r\Delta t v_j^n - \frac{\Delta t}{2\Delta x}\left(r - \frac{\sigma^2}{2}\right)\left(v_{j+1}^n - v_{j-1}^n\right) - \frac{\Delta t}{2\Delta x^2}\sigma^2\left(v_{j+1}^n - 2v_{j-1}^n + v_{j-1}^n\right), \tag{31}$$

$$\implies -\frac{\Delta t}{2\Delta x}\left(\frac{\sigma^2}{\Delta x} - \left(r - \frac{\sigma^2}{2}\right)\right)v_{j-1}^n - \left(1 + r\Delta t + \frac{\Delta t\sigma^2}{\Delta x}\right)v_j^n - \frac{\Delta t}{2\Delta x}\left(\frac{\sigma^2}{\Delta x} + \left(r - \frac{\sigma^2}{2}\right)\right)v_{j+1}^n = v_j^{n+1} \tag{32}$$

and the boundary conditions become

1. $V(X, t = T) = (\exp(S) - K)^+$,
2. $V(X = X_{\max}, t) = \exp(X_{\max}) - K\exp(-r(T - t))$, and
3. $V(X_{\min}, t) \approx 0$.

Keep in mind that the LHS of (32) is unknown while the RHS is known as we start stepping from $t = T$. The fact that this is technically backward in time is merely pedantic, as mentioned earlier that it is equivalent to the "truly" forward in time (in the τ coordinate) and implicit.

Where numerically solving the explicit scheme requires a matrix-vector multiplication, numerically solving the implicit equations (32) requires solving a tridiagonal linear system. Again, we can represent (32) more succintely by defining using a shorthand for the tridiagonal matrix entries.

$$k_l v_{j-1}^n + k_m v_j^n + k_u v_{j+1}^n = v_j^{n+1}. \tag{33}$$

Here ,

$$\begin{aligned} k_l &= -\frac{\Delta t}{2\Delta x}\left(\frac{\sigma^2}{\Delta x} - \left(r - \frac{\sigma^2}{2}\right)\right) \\ k_m &= -\left(1 + r\Delta t + \frac{\Delta t\sigma^2}{\Delta x}\right) \\ k_u &= -\frac{\Delta t}{2\Delta x}\left(\frac{\sigma^2}{\Delta x} + \left(r - \frac{\sigma^2}{2}\right)\right) \end{aligned} \tag{34}$$

We now have everything to write the implicit scheme in the matrix form. There is one important difference from the explicit scheme though; to incorporate the Dirichlet boundary conditions, we we zeroed the first and last rows of the matrix in (30). That cannot be done here as doing so would result in a singular matrix, which cannot be inverted/factorised. So, for M points in the discretised X-space, the first and last are fixed by the Dirichlet boundary conditions, and we only need to solve the tridiagonal linear system for the points in between.

$$\begin{bmatrix} k_l & k_m & k_u & 0 & \dots & 0 \\ 0 & k_l & k_m & k_u & \dots & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & & \dots & k_l & k_m & k_u \end{bmatrix} \begin{bmatrix} v_2^{n+1} \\ v_3^{n+1} \\ \vdots \\ v_{M-1}^{n+1} \end{bmatrix} = \begin{bmatrix} v_2^n \\ v_3^n \\ \vdots \\ v_{M-1}^n \end{bmatrix} \tag{35}$$
$$v_1^{n+1} = 0$$
$$v_M^{n+1} = \exp(X_{\max}) - K\exp\left(-r(T-(n-1)\Delta t)\right)$$

2.4.3 Crank-Nicolson Scheme

Rearranging the terms in (26), bringing $n+1$ terms to the LHS and the remaining terms to the RHS, we get

$$v_j^{n+1} - \left(r - \frac{\sigma^2}{2}\right)\frac{\Delta\tau}{4\Delta x}\left(v_{j+1}^{n+1} - v_{j-1}^{n+1}\right) + \frac{r\Delta\tau}{2}v_j^{n+1} - \frac{1}{4}\sigma^2\frac{\Delta\tau}{\Delta x^2}\left(v_{j+1}^{n+1} - 2v_j^{n+1} + v_{j-1}^{n+1}\right) =$$
$$v_j^n + \left(r - \frac{\sigma^2}{2}\right)\frac{\Delta\tau}{4\Delta x}\left(v_{+1}^n - v_{j-1}^n\right) + \frac{1}{4}\sigma^2\frac{\Delta\tau}{\Delta x^2}\left(v_{j+1}^n - 2v_j^n + v_{j-1}^n\right) - \frac{r\Delta\tau}{2}v_j^n$$

We introduce the following for brevity.

$$\begin{aligned} \alpha &= \left(r - \frac{\sigma^2}{2}\right)\frac{\Delta\tau}{4\Delta x} \\ \beta &= \frac{1}{4}\sigma^2\frac{\Delta\tau}{\Delta x^2} \\ \gamma &= \frac{r\Delta\tau}{2} \end{aligned} \tag{36}$$

$$\begin{aligned} \implies & v_j^{n+1} - \alpha\left(v_{j+1}^{n+1} - v_{j-1}^{n+1}\right) + \gamma v_j^{n+1} - \beta\left(v_{j+1}^{n+1} - 2v_j^{n+1} + v_{j-1}^{n+1}\right) \\ & \qquad = v_j^n + \alpha\left(v_{j+1}^n - v_{j-1}^n\right) + \beta\left(v_{j+1}^n - 2v_j^n + v_{j-1}^n\right) - \gamma v_j^n \\ \implies & (\alpha-\beta)\,v_{j-1}^{n+1} + (1+2\beta+\gamma)\,v_j^{n+1} - (\alpha+\beta)v_{j+1}^{n+1} \\ & \qquad = (\beta-\alpha)v_{j-1}^n + (1-2\beta-\gamma)v_j^n + (\alpha+\beta)v_{j+1}^n \end{aligned} \tag{37}$$

The boundary conditions are the same as before. This time we have both tridiagonal matrix factorisation and a matrix-vector multiplication. So, we can write (37) in the following matrix form.

$$\begin{bmatrix} \alpha-\beta & 1+2\beta+\gamma & -(\alpha+\beta) & 0 & \dots & 0 \\ 0 & \alpha-\beta & 1+2\beta+\gamma & -(\alpha+\beta) & \dots & 0 \\ \vdots & & \ddots & & & \\ 0 & \dots & & \alpha-\beta & 1+2\beta+\gamma & -(\alpha+\beta) \end{bmatrix} \begin{bmatrix} v_2^{n+1} \\ v_3^{n+1} \\ \vdots \\ v_{M-1}^{n+1} \end{bmatrix}$$
$$= \begin{bmatrix} \beta-\alpha & 1-2\beta-\gamma & \alpha+\beta & 0 & \dots & 0 \\ 0 & \beta-\alpha & 1-2\beta-\gamma & \alpha+\beta & \dots & 0 \\ \vdots & & \ddots & & & \\ 0 & \dots & & \beta-\alpha & 1-2\beta-\gamma & \alpha+\beta \end{bmatrix} \begin{bmatrix} v_2^n \\ v_3^n \\ \vdots \\ v_{M-1}^n \end{bmatrix} \tag{38}$$

$$v_1^n = 0$$
$$v_M^n = S_{\max} - K\exp(-r\tau)$$

More succinctly,

$$\begin{bmatrix} b_l & b_m & b_u & 0 & \dots & 0 \\ 0 & b_l & b_m & b_u & \dots & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & & \dots & b_l & b_m & b_u \end{bmatrix} \begin{bmatrix} v_2^{n+1} \\ v_3^{n+1} \\ \vdots \\ v_{M-1}^{n+1} \end{bmatrix} = \begin{bmatrix} a_l & a_m & a_u & 0 & \dots & 0 \\ 0 & a_l & a_m & a_u & \dots & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & & \dots & a_l & a_m & a_u \end{bmatrix} \begin{bmatrix} v_2^n \\ v_3^n \\ \vdots \\ v_{M-1}^n \end{bmatrix}, \tag{39}$$

where

$$b_l = \alpha - \beta = \frac{\Delta\tau}{4\Delta x}\left(\left(r - \frac{\sigma^2}{2}\right) - \frac{\sigma^2}{\Delta x}\right)$$
$$b_m = 1 + 2\beta + \gamma = 1 + \frac{1}{2}\sigma^2\frac{\Delta\tau}{\Delta x^2} + r\Delta\frac{\tau}{2}$$
$$b_u = -\alpha - \beta = \frac{\Delta\tau}{4\Delta x}\left(-\left(r - \frac{\sigma^2}{2}\right) - \frac{\sigma^2}{\Delta x}\right)$$
$$a_l = \beta - \alpha = \frac{\Delta\tau}{4\Delta x}\left(-\left(r - \frac{\sigma^2}{2}\right) + \frac{\sigma^2}{\Delta x}\right)$$
$$a_m = 1 - 2\beta - \gamma = 1 - \frac{1}{2}\sigma^2\frac{\Delta\tau}{\Delta x^2} - r\Delta\frac{\tau}{2}$$
$$a_u = \alpha + \beta = \frac{\Delta\tau}{4\Delta x}\left(\left(r - \frac{\sigma^2}{2}\right) + \frac{\sigma^2}{\Delta x}\right)$$

2.4.4 Calculating if scheme follows a sceond order convergence in number of grid points in space

We calculated the convergence with eq. (40) and presumed if that was constant then the FD method would follow a second order convergence.

$$\frac{V_{analytical} - V_{FD}}{dx^2} \tag{40}$$

2.4.5 Computing Δ for the FD method

To compute Δ we will use a forward finite difference method. So eq. (41) becomes eq. (42).

$$\frac{\partial V}{\partial S} = \Delta \tag{41}$$

$$\frac{V_{j+1}^n - V_j^n}{S_{j+1}^n - S_j^n} = \Delta \tag{42}$$

2.5 The COS Method

The Cos method is a method of approximating option prices which relies on Fourier-cosine expansions in the context of numerical integration, and is an alternative for the methods based on the FFT. In the following section, we will introduce the method and show how applying it on european-type options can result in a good approximation of the option value while improving speed.

Let the option value V be given by the risk-neutral valuation formula:

$$V(x, t_0) = e^{-r\Delta t}\int_{\mathbb{R}} g(y)f(y \mid x)dy \tag{43}$$

where $f(y \mid x)$ is the conditional density of y given x, and $g(y)$ denotes the payoff function.

Furthermore, x and y are simply transformations given by:

$$x = \log\left(\frac{S_0}{K}\right) \text{ and } y = \log\left(\frac{S_t}{K}\right) \tag{44}$$

The core idea here is to reconstruct the integral from its Fourier-cosine series expansion, and with that extracting the series coefficients directly from the integrand.

In order to extract the cosine coefficients we introduce the following:

Proposition 2.1. *For a function supported on $[0, \pi]$, the cosine expansion reads*

$$f(\theta) = \sum_{k=0}^{1\infty} A_k \cdot \cos(k\theta) \quad \textit{with} \quad A_k = \frac{2}{\pi}\int_0^\pi f(\theta)\cos(k\theta)d\theta \tag{45}$$

In our case, for pricing european options, we introduce following:

Let the density and its characteristic function denoted by $f(\omega)$ and $\phi(\omega)$, which form an example of a Fourier pair and are given by the following:

$$\phi(u) = \int_{\mathbb{R}} e^{i\omega u}f(u)du, \quad f(u) = \frac{1}{2\pi}\int_{\mathbb{R}} e^{-iu\omega}\phi(\omega)d\omega \tag{46}$$

From there, we know that the characteristic equation of x is given by:

$$\phi_{GBM}(u) = e^{iu(r-\frac{1}{2}\sigma^2)t-(\frac{1}{2}\sigma^2tu^2)}e^{iux} \tag{47}$$

and define a, b as

$$\begin{aligned} a &= \log\tfrac{S_0}{K} + rT - 12\sqrt{\sigma^2 T} \\ b &= \log\tfrac{S_0}{K} + rT + 12\sqrt{\sigma^2 T} \end{aligned} \tag{48}$$

Then, we can transform the Fourier equation so that it applies on any interval $[a, b]$ by a simple change of variable:

$$\theta := \frac{x-a}{b-a}\pi; \quad x = \frac{b-a}{\pi}\theta + a \tag{49}$$

such that $f(x)$ becomes:

$$f(x) = \sum_{k=0}^{\infty} A_k \cdot \cos\left(k\pi\frac{x-a}{b-a}\right) \tag{50}$$

and

$$A_k = \frac{2}{b-a}\int_a^b f(x)\cos\left(k\pi\frac{x-a}{b-a}\right)dx \tag{51}$$

$$F_k \equiv \frac{2}{b-a}\,\mathrm{Re}\left\{\phi\left(\frac{k\pi}{b-a}\right)\cdot\exp\left(-i\frac{ka\pi}{b-a}\right)\right\} \tag{52}$$

Furthermore, from [1] we know that the cosine coefficients of the payoff function are given by:

$$G_k = \frac{2}{b-a}\int_0^b K\left(e^y-1\right)\cos\left(k\pi\frac{y-a}{b-a}\right)dy = \frac{2}{b-a}K\left(\chi_k(0,b)-\psi_k(0,b)\right)$$

where $\chi_k(a,b)$ and $\psi_k(a,b)$ are given as:

$$\begin{aligned} \chi_k(c,d) :=& \frac{1}{1+\left(\frac{k\pi}{b-a}\right)^2}\left[\cos\left(k\pi\frac{d-a}{b-a}\right)e^d - \cos\left(k\pi\frac{c-a}{b-a}\right)e^c\right. \\ &\left. + \frac{k\pi}{b-a}\sin\left(k\pi\frac{d-a}{b-a}\right)e^d - \frac{k\pi}{b-a}\sin\left(k\pi\frac{c-a}{b-a}\right)e^c\right], \\ \psi_k(c,d) :=& \begin{cases} \left[\sin\left(k\pi\frac{d-a}{b-a}\right) - \sin\left(k\pi\frac{c-a}{b-a}\right)\right]\frac{b-a}{k\pi} & \text{if } k \neq 0; \\ d-c & \text{if } k = 0. \end{cases} \end{aligned}$$

Combining the two results, this shows that we can approximate the initial integral into

$$V(x,t_0) = e^{-r\Delta t}\int_{\mathbb{R}} g(y)f(y \mid x)dy \approx e^{-rt}\frac{b-a}{2}\sum_{k=0}^{N} F_k(x)G_k \tag{53}$$

where the finite sums of the Fourier cosine coefficients of the conditional density function and the payoff function are denoted by F_n and G_n respectively

3 Results

3.1 Finite Difference schemes for calculating the option price of an European call

In this section we want to determine the price for an European call option with the FD method. We will use two different schemes: FTCS implicit scheme and the CN-scheme. We used the implicit method instead of the explicit method for the FTCS method since the explicit method a has severe stability condition on $\Delta\tau$. The vectors a_{-1}, a_0, a_1 were computed with k_l of k_m and k_u of eq. (34) respectively for the implicit FTCS scheme. For the implicit scheme B_{-1} is 0, B_0 is 1, B_1 and 0, since matrix B is an identity matrix. The coefficients for the CN-scheme can be seen in eq. (39).

We will calculate the option prices for the following parameters:

- $r = 4\%$

- $\sigma = 30\%$
- $S0 = 100$, $S0 = 110$ and $S0 = 120$
- $K = 110$
- $T = 1$ *year*

First we will determine the ideal FD-mesh for every $S0$ and the two different schemes. In fig. 1 the absolute error can be seen between the analytical solution for the European call option with the parameter above ($S0 = 100$) and the solution obtained with the finite difference method with the FTCS-scheme for different FD-mesh sizes. The mesh size that got the smallest absolute difference was: 250 stock steps and 900 time steps with an absolute error of $3.04925 * 10^{-5}$. This resulted in an European option price of 9.62533 with the FD method with the FTCS scheme compared to the analytical solution of 9.62536 with the Black-scholes equation.

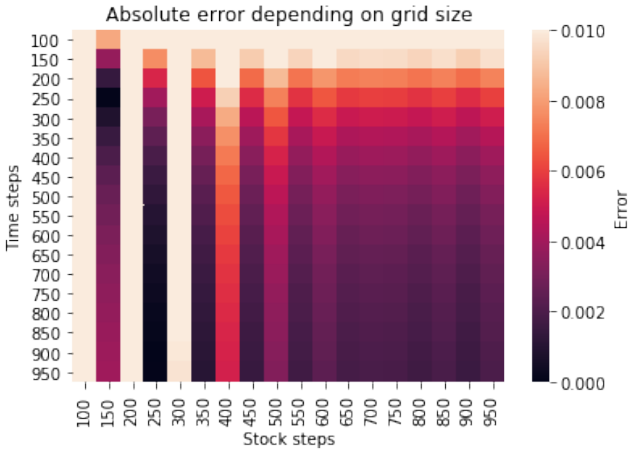

Figure 1: The absolute error between the analytical solution for the European call option with the parameter above ($S0 = 100$) and the solution obtained with the finite difference method with the FTCS-scheme for different FD-mesh sizes.

A 3d graph of the option can be seen in fig. 2. This graph looks the same for all the options of $S0$, since this parameter does not influence how the final graph looks like. Also the changing of the scheme did not give notable differences in the final FD-grid graph. It did give different results, but not notable in the scale of such a graph. Finally the different mesh sizes also did not give notably different graphs, so we will show the FD-grid once with a X and with the actual stock price S. The right graph is actually a zoomed in version of the left graph.

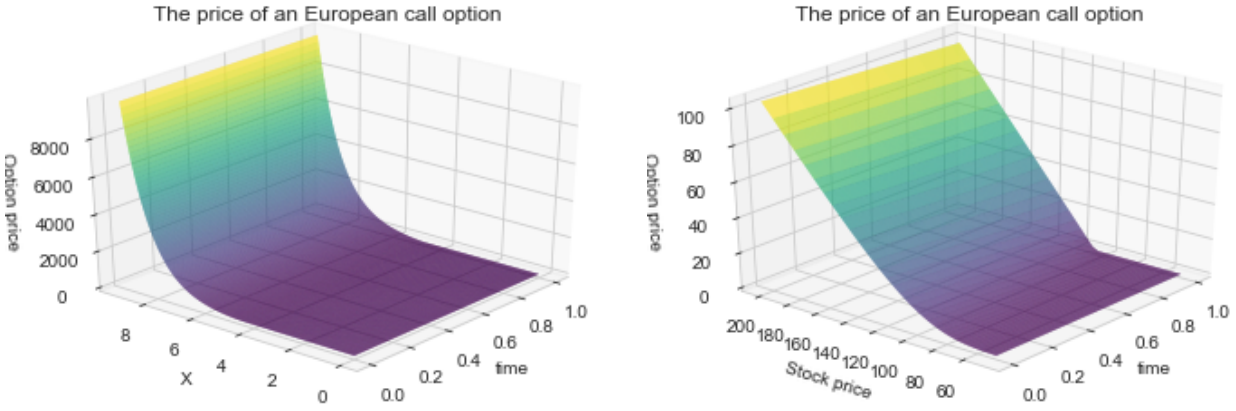

Figure 2: A FD grid calculated with the parameters named above. In this graph the option price can be seen on different times and for different stock prices. The stock prices are noted as X in the left graph. The actual stock price is $S = e^X$, which can be seen in the right graph. In the right graph we also zoomed in on the option prices around strike price K

In this report we also studied if the implicit FTCS scheme had a second order convergence in number of grid points, which can be seen in fig. 3.

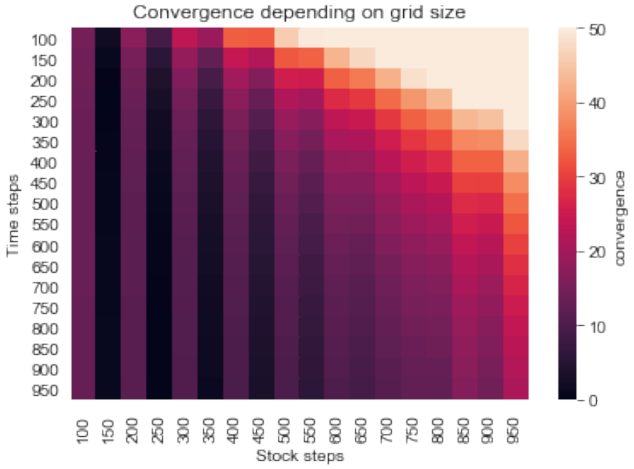

Figure 3: Convergence of the implicit FTCS scheme in number of grid points

In fig. 4 the absolute error can be seen with the same parameters and the FTCS-scheme, but $S0$ is changed to $S0 = 110$. The mesh size that got the smallest absolute difference was: 650 stock steps and 800 time steps with an absolute error of $1.39560 * 10^{-5}$. This resulted in an European option price of 15.128605 with the FD method with the FTCS scheme compared to the analytical solution of 15.12859 with the Black-scholes equation.

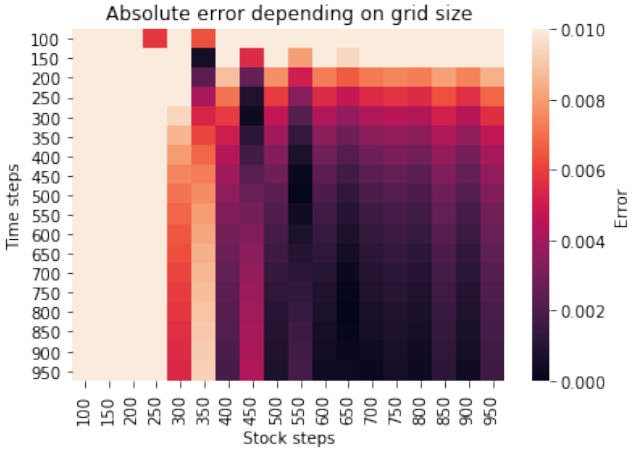

Figure 4: The absolute error between the analytical solution for the European call option with the parameter above ($S0 = 110$) and the solution obtained with the finite difference method with the FTCS-scheme for different FD-mesh sizes.

In fig. 5 the absolute error can be seen with the same parameters and the FTCS-scheme, but $S0$ is changed to $S0 = 120$. The mesh size that got the smallest absolute difference was: 450 stock steps and 850 time steps with an absolute error of $1.99444 * 10^{-5}$. This resulted in an European option price of 21.78879 with the FD method with the FTCS scheme compared to the analytical solution of 21.788808 with the Black-scholes equation.

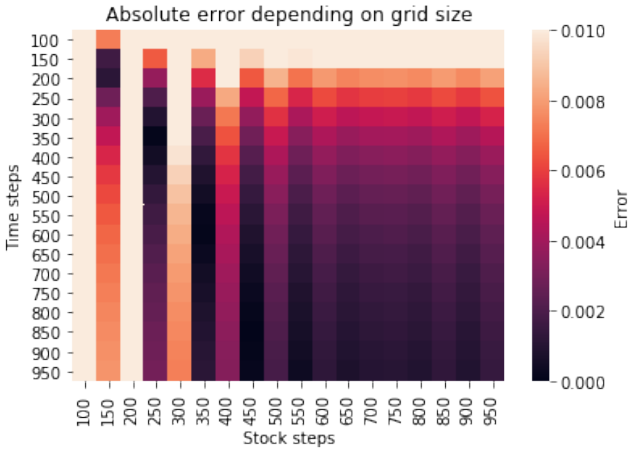

Figure 5: The absolute error between the analytical solution for the European call option with the parameter above ($S0 = 120$) and the solution obtained with the finite difference method with the FTCS-scheme for different FD-mesh sizes.

In fig. 6 the absolute error can be seen with the same parameters and the CN-scheme, but $S0$ is changed to $S0 = 100$. The mesh size that got the smallest absolute difference was: 450 stock steps and 150 time steps with an absolute error of $1.89483 * 10^{-7}$. This resulted in an European option price of 9.62536 with the FD method with the FTCS scheme compared to the analytical solution of 9.62536 with the Black-scholes equation.

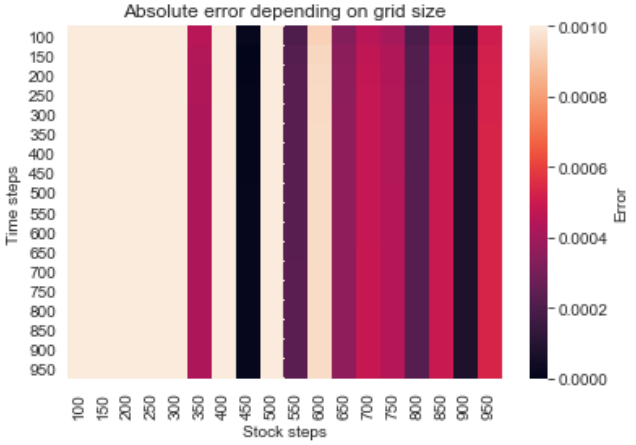

Figure 6: The absolute error between the analytical solution for the European call option with the parameter above ($S0 = 100$) and the solution obtained with the finite difference method with the CN-scheme for different FD-mesh sizes.

In this report we also studied if the CN-scheme had a second order convergence in number of grid points, which can be seen in fig. 7.

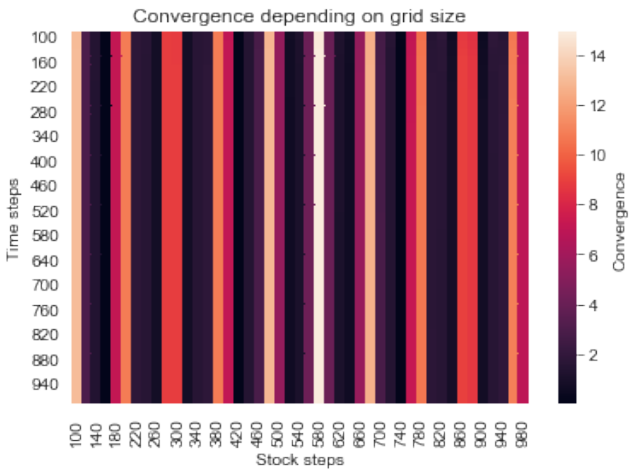

Figure 7: Convergence of the CN-scheme in number of grid points

In fig. 8 the absolute error can be seen with the same parameters and the CN-scheme, but $S0$ is changed to $S0 = 110$. The mesh size that got the smallest absolute difference was: 400 stock steps and 100 time steps with an absolute error of $9.59493 * 10^{-6}$. This resulted in an European option price of 15.12858 with the FD method with the FTCS scheme compared to the analytical solution of 15.12859 with the Black-scholes equation.

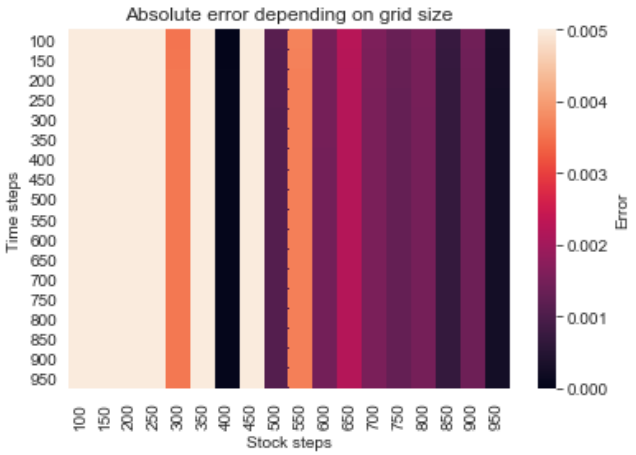

Figure 8: The absolute error between the analytical solution for the European call option with the parameter above ($S0 = 110$) and the solution obtained with the finite difference method with the CN-scheme for different FD-mesh sizes.

In fig. 9 the absolute error can be seen with the same parameters and the CN-scheme, but $S0$ is changed to $S0 = 120$. The mesh size that got the smallest absolute difference was: 500 stock steps and 100 time steps with an absolute error of $2.06269 * 10^{-5}$. This resulted in an European option price of 21.78879 with the FD method with the FTCS scheme compared to the analytical solution of 21.78880 with the Black-scholes equation.

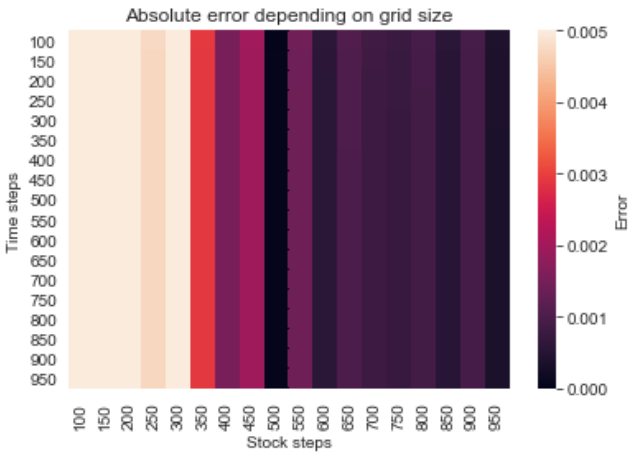

Figure 9: The absolute error between the analytical solution for the European call option with the parameter above ($S0 = 120$) and the solution obtained with the finite difference method with the CN-scheme for different FD-mesh sizes.

3.2 Finite difference schems for the delta of an European call

When we sell one European call option the Δ parameter can be used to hedge our risk by buying Δ stocks. In fig. 10 Δ can be seen in the left figure for stock prices from X_{min} to X_{max} at $t = 0$ and in the right figure the Δ can be seen where it is the most variable for each S. For the calculation of the Δ in the section the parameters and scheme were used for the FD method that had the smallest absolution error in section 3.1 compared to the analytical Black-Scholes method. This was the CN-scheme for the calculation of $S0 = 100$ with 450 stock steps and 150 time steps.

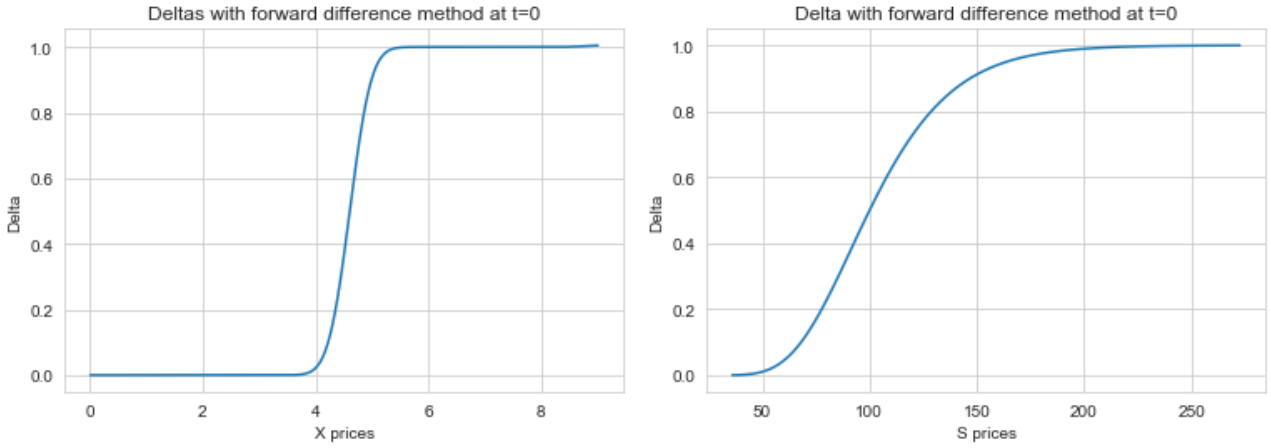

Figure 10: In the left figure Δ can be seen for stock prices from X_{min} to X_{max} at $t = 0$, where $S = e^X$. In the right figure the Δ can be seen where it is the most variable for each S.

In fig. 11 Δ can be seen in the left figure for stock prices from X_{min} to X_{max} at every t and in the right figure the Δ can be seen where it is the most variable for each X.

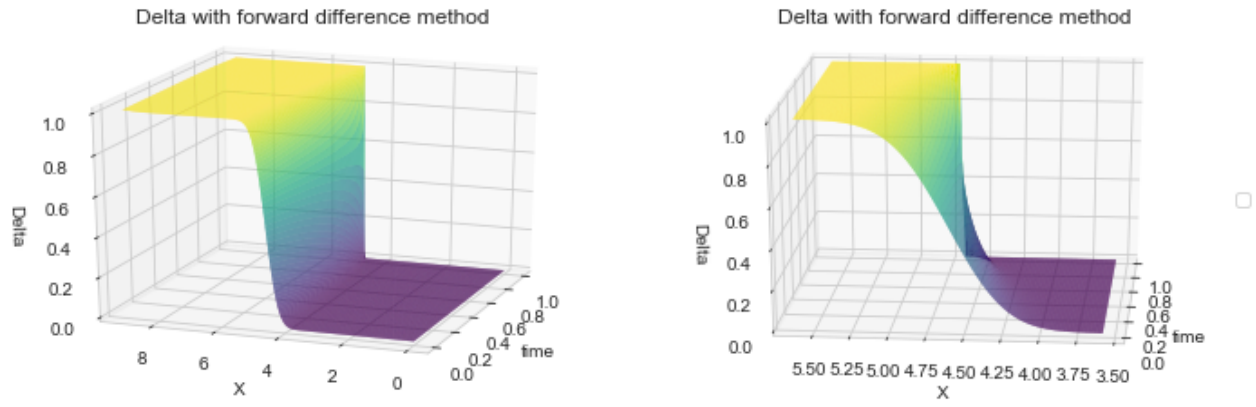

Figure 11: In the left figure Δ can be seen for stock prices from X_{min} to X_{max} for every t from $t = 0$ to $t = T$, where $S = e^X$. In the right figure the Δ can be seen where it is the most variable for each X.

In fig. 12 Δ can be seen in the left figure at the part where the option price is most variable for the stock prices S and in the right figure the same graph can be seen from the other side.

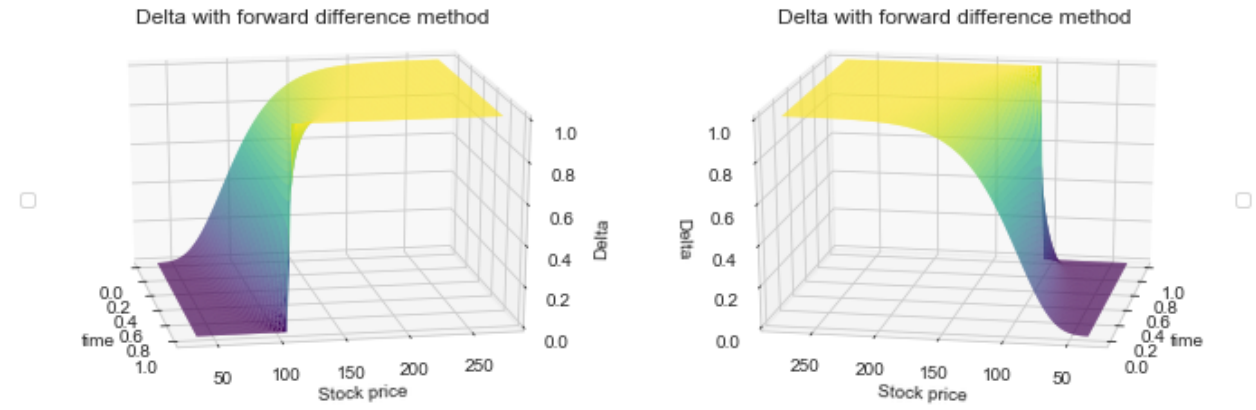

Figure 12: The same figures as the right figure of fig. 11, but the axis with X is changed to the stock price S to show more clearly what happens around the strike price K of 110.

3.3 COS method

In this section we will show the result obtained by using the COS method as outlined in section 2.5. We will be using the same parameters as before, namely:

- $r = 4\%$
- $\sigma = 30\%$
- $S0 = 100$
- $K = 110$
- $T = 1\ year$

In our simulation, we will be approximating the value of the call option V using k Fourier cosine coefficients, where $k \in [1, 196]$. Throughout this comparison, the analytical solution given by the Black-Scholes equation will be used as a benchmark.

Coefficients	16	32	64	96	128	160	196
Time	0.002699	0.004250	0.009164	0.026433	0.017502	0.02281300	0.040599
Error (%)	10.648960	0.003539	7.2498e-12	7.2498e-12	7.2498e-12	7.2498e-12	7.2498e-12

Table 1: Computational time(in seconds) for the COS method alongside relative errors as % of the Black-Scholes solution found for $16, 32, 64, 128, 160$ and 196 coefficients.

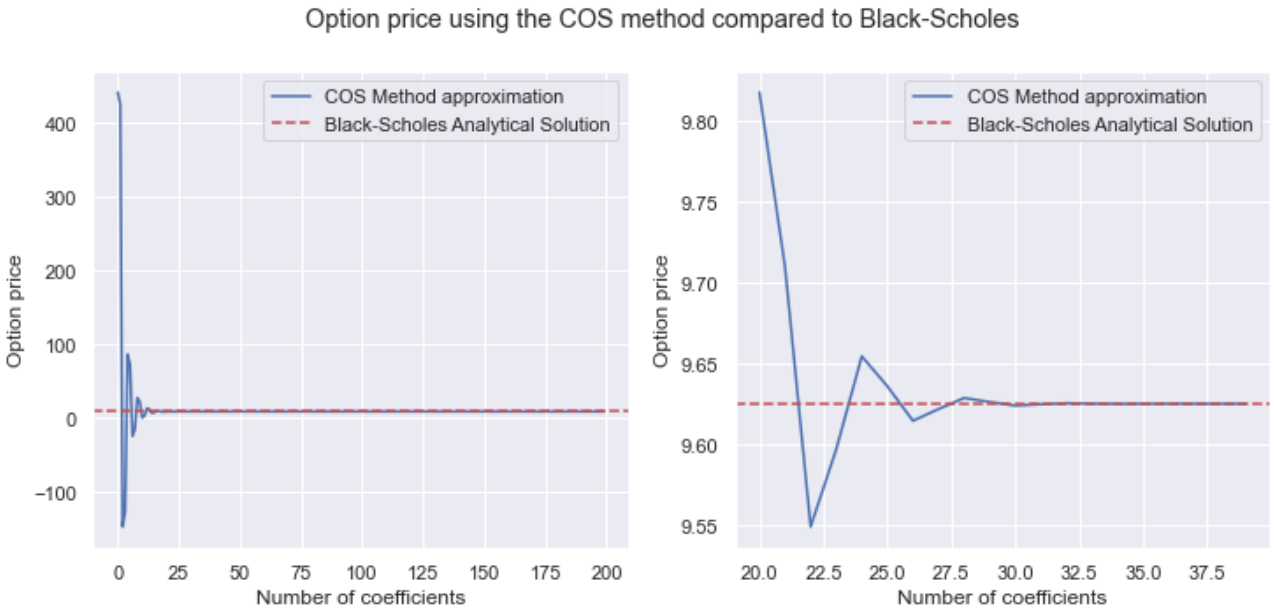

Figure 13: COS Method values

As can be seen from figure 13, the method begins oscillating around the analytical solution, and begins to converges towards values of k higher than 25.

Furthermore, we calculate the relative and absolute errors against the Black-Scholes analytical solution to obtain:

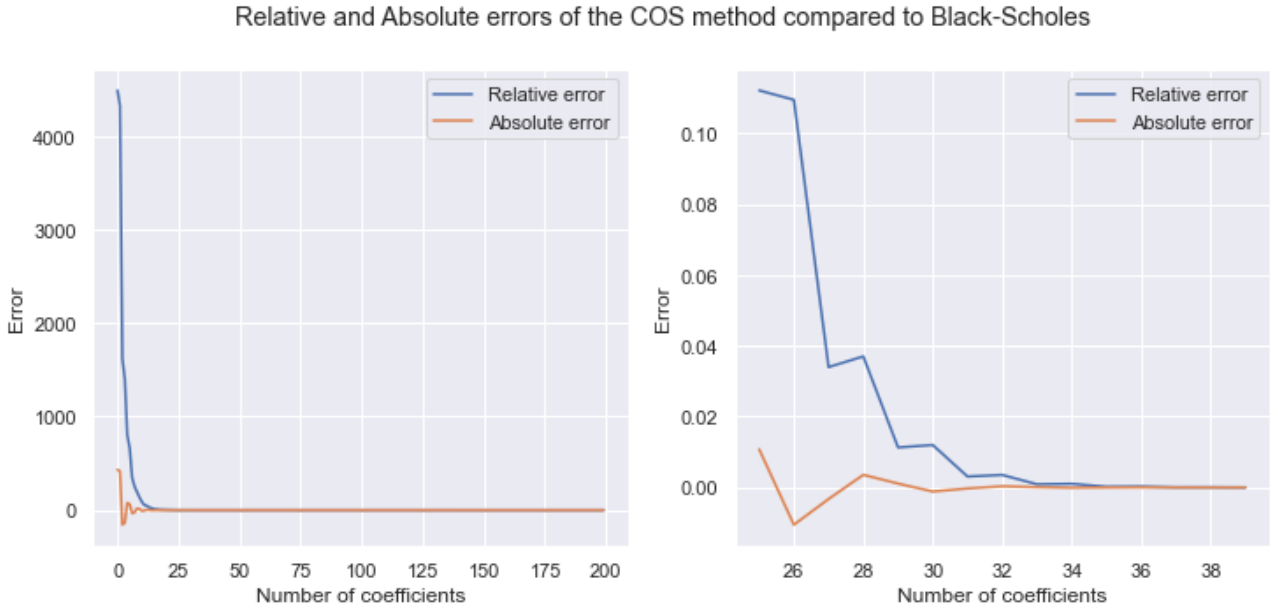

Figure 14: COS Method errros

As can be seen from the plot in 14 and subsequently in table 1, the errors start high for a low number of coefficients, while the solution starts to get more and more accurate as we approach higher values of k. In fact,

when $k = 64$ we already achieve strong convergence, as the error is minuscule. Furthermore, the method was able to accurately compute the option value in a significantly fast way.

4 Discussion

In this report we first derived the transformed Black-scholes PDE. Then we showed the derivation of the FD approach with which you can solve the transformed Black-scholes PDE numerically with the FTCS scheme in an explicit way and an implicit way and with the CN-scheme. Lastly we showed the derivation of the COS method in the methods. In the results section we studied both the FD method and the COS method and compared it to the analytical Black-scholes method.

In fig. 2 the option price of the European call option can be seen for the FTCS-scheme with the parameters that were named in section 3.1. This graph was the same for every $S0$ and every scheme. The only difference was the number of time steps and stock steps that were used for each specific combination of $S0$ and scheme. The combination of scheme and the number of stock and time steps did give different absolute errors for $S0$, which can be seen in fig. 1, fig. 4, fig. 5, fig. 6, fig. 8 and fig. 9. In all the graphs of the absolute errors between the option price calculated with the FD method and the analytical Black-Scholes method it can be seen that generally increasing the stock steps means that the absolute error will decrease. For the FTCS-scheme we can also state that generally increasing the time steps means that we decrease the absolute error. For the CN-scheme this is result is not seen. The CN-scheme gave a better result for every $S0$.

For both the implicit FTCS and CN-scheme we studied if they had a second order convergence in number of grid point for $S0 = 100$. For the FTCS scheme this can be seen fig. 3 and for the CN scheme this can be seen fig. 7. We calculated the convergence parameter for every point with the eq. (40) and presumed if the convergence parameter would not increase over the number of stock steps than it would follow a second order convergence. For the FTCS scheme we could say that this is very roughly true when we use more than 600 time steps, but still doubtful. For the CN scheme this is not constant, so we do not see a clue of second order convergence in our programm.

In section 3.2 we already told that the Δ can be used to hedge your risk when you sell an European call option. This can be done by buying Δ amount of stock of the underlying assets. We studied Δ at $t = 0$ from X_{min} to X_{max} as can be seen in the left figure of fig. 10 and in the right figure we zoomed in on the part where the Δ is the most variable. In the right figure we also changed the axis from X to S, where $S = e^X$. In the right figure of fig. 11 the same can be seen but then for all t and in the left figure of fig. 11 we again zoom in on the part where Δ is the most variable. In fig. 12 the same figure can be seen as in fig. 11 but X was changed to S to show more clearly that at the maturity time $t = T$ the Δ is either 1 or 0. The seperation between if the Δ is 1 or 0 is exactly at the strike price K, which is logical since at the maturity time the European call option is either worth something if the stock price S is larger than K or it is worth nothing if the the stock price S is lower than K. The graph of fig. 10 can get used to hedge your option at $t = 0$ by buying Δ number of stock of the underlying asset of the sold European call option and the graph of fig. 11 can get used to hedge your option dynamically over time.

And finally, as for the COS method, the results shown in figure 13, figure 14 and subsequently in table 1, we can see that the errors start high for a low number of coefficients, while the solution starts to get more and more accurate as we approach higher values of k. In fact, when $k = 64$ we already achieve strong convergence, as the error is miniscule. Furthermore, the time complexity of the algorithm is given by $\mathcal{O}(n)$, which can be seen in the time row in the table, as the time increases linearly for higher values of k. For every value of k, the method was able to accurately compute the option value in a significantly fast way. In fact, there was no need to compute the option value with more than 64 coefficients, as that resulted already in a fast, accurate approximation based on the error.

References

[1] F. Fang and C. W. Oosterlee, "A novel pricing method for european options based on fourier-cosine series expansions," SIAM Journal on Scientific Computing, vol. 31, no. 2, pp. 826–848, 2009.