Finite Difference Methods For Financial Partial Differential Equations
Københavns Universitet
Assignment

Patrik H. Rauff KNJ284

Contents

1 Convergence in time for different θ-schemes:

Given the parameters for the assignment, which is:

$$T = 5,\ K = 1.025,\ S(0) = 1,\ r = 0.04,\ \mu = -0.03,\ \sigma = 0.2,\ n = 100,\ \theta = 0.5 \tag{1}$$

We use our implemented `kBlack::fdRunner()` but were we use a log-transformation, $x(t_0) = \log S(t_0)$ of the underlying and hence our Black-Scholes PDE to solve for the Finite-difference scheme becomes:

$$\begin{aligned} & rV(t,x) = \partial_t V(t,x) + \mu \partial_x V(t,x) + \frac{1}{2}\sigma^2 \left\{\partial_{xx} V(t,x) - \partial_x V(t,x)\right\}, \quad v(T,x) = IV_T \\ \Rightarrow \quad & 0 = \partial_t V(t,x) + Av(t,x), \quad A = r - \left(\mu - \frac{1}{2}\sigma^2\right)\partial_x + \frac{1}{2}\sigma^2 \partial_{xx} \end{aligned} \tag{2}$$

where A is the differential operator and IV_T is the instrinsic value at T ie. for a call option we have:

$$IV_T = \max\left\{S(T) - K\right\}^+$$

We define our grid of time and space, which is the bedrock of our finite difference algorithm:

$$\begin{aligned} & t_h = h \cdot \Delta t,\ \Delta t = T/m \\ & x_0 = x(t_0) - 5\sigma\sqrt{T},\ x_n = x(t_0) + 5\sigma\sqrt{T},\ x_i = x_0 + i \cdot \Delta x, \quad \Delta x = \frac{x_n - x_0}{n+1} \end{aligned}$$

Furthermore we will use a discretised version of the differential operator A by:

$$\bar{A} = -r + \left(\mu - \frac{1}{2}\sigma^2\right)\delta_x + \frac{1}{2}\sigma\delta_{xx}$$

where δ_x and δ_{xx} are the first and second order discretised derivatives.
Since the PDE above is a backward looking PDE we solve it in our theta scheme starting at the terminal boundary condition and then steps back in time:

$$t_{h+1} \rightarrow t_h \rightarrow t_{h-1}$$

Implementing `kBlack:fdRunner()` as the theta solver presented in [1], we solve the Backward scheme:

$$\begin{aligned} v(t_{h+1/2}) &= \{I + (1-\theta)A\} v(t_{h+1}) \\ \left[I - \theta\delta t\bar{A}\right] v(t_h) &= v(t_{t+1/2}) \end{aligned} \tag{3}$$

We test the convergence in time for three different versions of the theta scheme, namely $\theta = (0.0, 1.0, 0.5)$. These are respectively known as the explicit, implicit and Crank-Nicolson method. The convergence in time is tested on the different time steps:

$$m = (10, 25, 50, 75, 100, 150, 200, 400, 800, 1600, 3200, 6400) \tag{4}$$

Our results are shown and depicted in Table 1 and Figure 1 respectively. in Figure 1, and the rest of the error plots in this assignment, we use the $\log - \log$ plots where the error measure in the plots are computed as follows

$$\log\{|v(m) - v(100.000)|\}$$

Just as in [1].

We see that both the explicit and implicit schemes have the same order on convergence in time with both schemes being convergent of rate 1, ie $a = 1$ when $\mathcal{O}(\Delta t^a)$. This is however not the case for the the Crank-Nicolson Scheme. As we see in Table 1 and Figure 1 the Crank-Nicolson scheme converges much faster with $a \approx 2$. Furthermore we see that instability of the explicit method for $\theta = 0$ since it is not Von neumann, when $\Delta t \leq \mathcal{O}(\Delta x^2)$, which is the case for $m =$ 10, 25 , 50 and. Our results coincides quite well with the cheat sheet from [1]

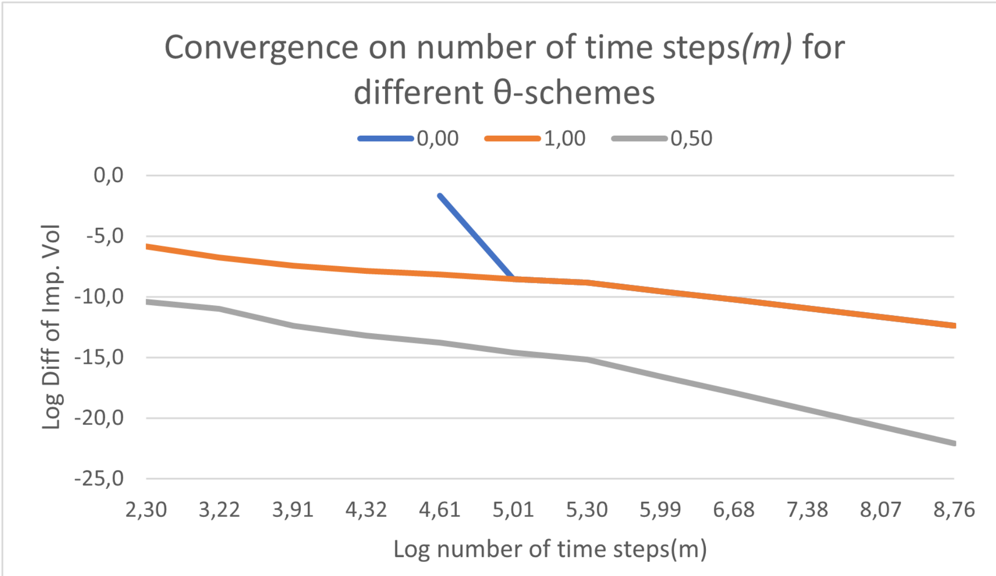


Figure 1: Convergence in time of implied volatility for the 3 different theta schemes.

m	θ =0.00	$\theta = 1.00$	$\theta = 0.50$
10	N/A	0.196978689	0.199872192
25	N/A	0.198691986	0.199859594
50	N/A	0.199266647	0.199847129
75	N/A	0.199458574	0.199844819
100	0.388011164	0.199554606	0.199844011
150	0.200035437	0.199650683	0.199843434
200	0.199987304	0.199698738	0.199843231
400	0.199915126	0.199770842	0.199843037
800	0.199879046	0.199806904	0.199842988
1.600	0.199861008	0.199824937	0.199842976
3.200	0.19985199	0.199833954	0.199842973
6.400	0.199847481	0.199838463	0.199842972

Table 1: Convergence table

2 Convergence of Early Exercise boundary

We want to find the american premium, which is:

$$\pi^{prem} = C^{\text{EU}} - C^{\text{AM}} \geq 0 \tag{5}$$

we need compute the price of an american option in our `fdRunner()`.
Formally what changes is that our PDE from (2) gets the additional boundary condition:

$$v(t,x) \geq \{e^x - K\}^+ \,,\ t \leq T$$

Implementing this into the finite difference solcer can be done in more than one way. However the easiest is to set the value at time t_h to the maximum of the value computed by the backward roll, $v(t_h, x_i)$ for $t_{h+1} \to t_h$ and the intrinsic value, $IV(S_i = e^{x_i})$:

$$V(t_h, x_i) = \max\{e^{x_i} - K, V(t_h, x_i)\}\,, \quad \text{for } i = 0, \ldots, n$$

Implementing this into our theta scheme and activating this, whenever our contract is american is somewhat misleading. In fact we compute the price of a Bermuda option with exercise points on each time point, t_h on our grid. Hence we approximate the american option price with a bermudan option price, however this does not come without a sacrifice.

This is seen in Figure 2 and Table 2, Where the American premium is found by the difference between the American option price found by our finite difference implementation subtracted by the discounted option price found by using the `xBlack()` formula. Acutally we find the slope/order of convergence in Δt to be $a \approx 1.025$ hence we numerically observe that just by implementing a check for early exercise actually reduces the accuracy in time by halve.

Nevertheless we are left wondering how such a dramatic loss might occur. Has it something to do with the difference between the price of a bermudan and amrican option respectively? If we compare the two types of contracts we see that the american option will in theory be more expensive than the bermudan, since the buyer of an american option can exercise the option at any time, including the discrete exercise points of the bermudan option. The intuition is that optimal exercise strategy τ maximizes the instrinsic value of the option. e We let $\tau < T$ denote the optimal exercise strategy and the bermudan that has exercise points $t_h = h\Delta t$. the American-Bermudan limit will be smaller than the difference between the american and an option forced to exercise at $\tau + \Delta t$. Hence we can approximate the Am-Ber premium by:

$$\begin{aligned}
0 \leq C^{\text{Am}} - C^{\text{Ber}} &\leq \mathbb{E}_\tau\left\{IV_\tau - IV_{\tau+\Delta t}\right\} \\
&= (S(\tau) - K) - e^{-r\Delta t}(e^{\mu\Delta t}S(\tau) - K) = S(\tau) - e^{-q\Delta t}S(\tau) - K(1 - e^{-r\Delta t}) \\
&= S(\tau)\left\{1 - (1 - \mathcal{O}(\Delta t))\right\} - K\left\{1 - (1 - \mathcal{O}(\Delta t))\right\} + \mathcal{O}(\Delta t^2) \\
&= \mathcal{O}\left(\Delta t\right)\left[S(\tau) - K\right]
\end{aligned}$$

At last we use a Taylor expansion and we get that the time error for our American-Bermudan premium is at most $\mathcal{O}(\Delta t)$.
However this error might be quite small, since the european part of the american and bermudan option is still quite large in comparison to the premiums.

It is possible to find algorithms that makes the american option have $\mathcal{O}(\Delta t^2)$ for $\theta = 0.5$ (such as the PSOR method), however this may take more computational load than one benefits from the higher order in convergence.

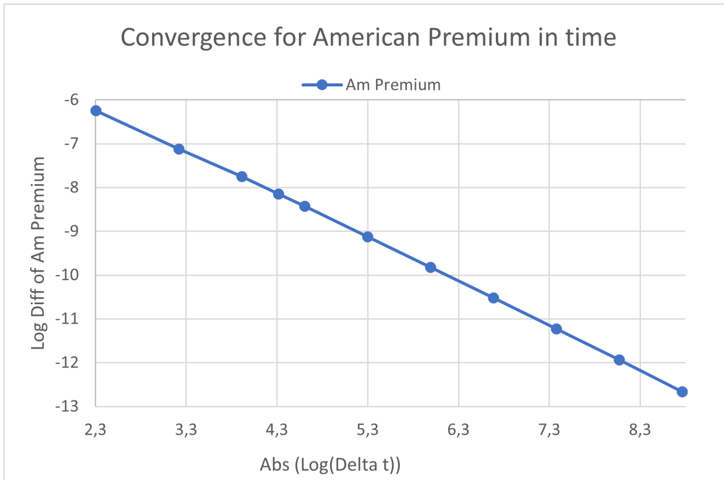


Figure 2: convergence for american premium in time

m	Am Premium
10	0.021500621277
25	0.022628931560
50	0.023006277905
75	0.023146374353
100	0.023216849067
200	0.023326847195
400	0.023381555930
800	0.023408745758
1600	0.023422334834
3200	0.023429102363
6400	0.023432483045

Table 2: Convergence table for american premiums.

Furthermore we are asked to make our finite difference solver print the early exercise boundary. For selected number of time steps, m from our array in (4) we see the optimal exercise boundary in Figure 3. Here we observe that it does not look that continous as one might think. Nevertheless this is due to the discritization on the spacial domain, which is limited to $n = 100$ and thus for relative high numbers of m it is hard to distinquish the exercise boundaries apart. It should be noted that we have forced the optimal exercise boundary at $B(\tau = 0) = K$, just indicating that if the option is ITM, then it shall be exercised.

Figure 3: Optimal Exercise boundary for $n = 100$ and various m.

3 Smoothing:

We fix the grid in time and space to $(n, m) = (100, 100)$ and now consider a european digital call option instead of just a standard call option, which then as we know changes the instrinsic value to:

$$IV_T = \mathbf{1}_{\{S(T) \geq K\}}$$

Nothing else changes, but as we observe from Figure 4 and Table 3 it has quite an effect.

What is really happening here is to showcase that one may meet trouble with non-linear or discontinued pay-off structures if the spacial grid is not too dense such that Δx is too large. Even though the strikes changes in a certain range, the price of the digital call, does not change along side it. It is constant until a certain limit is hit and then it takes a step down the price-ladder. On the other hand, when the payoff is 'smoothed', we get a more continous curve of prices wrt. strike, as one would expect.

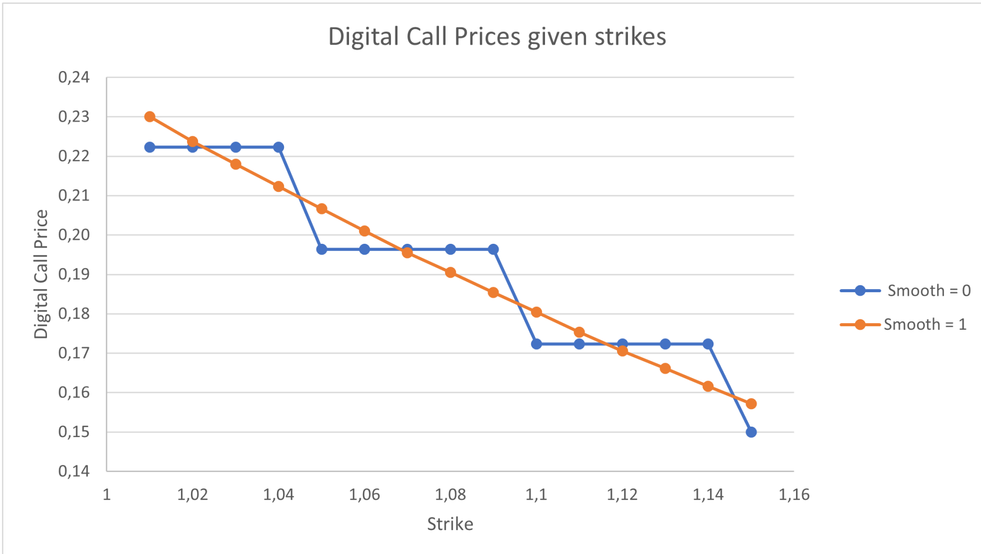


Figure 4: Digital call prices to strike.

K	Non-Smoothed	Smoothed
1.01	0.222264566	0.230028782
1.02	0.222264566	0.223772323
1.03	0.222264566	0.21798295
1.04	0.222264566	0.212341882
1.05	0.196414792	0.206700813
1.06	0.196414792	0.201059744
1.07	0.196414792	0.195525299
1.08	0.196414792	0.190488044
1.09	0.196414792	0.185450788
1.1	0.172297377	0.180413533
1.11	0.172297377	0.175376277
1.12	0.172297377	0.17056547
1.13	0.172297377	0.166110683
1.14	0.172297377	0.161655896
1.15	0.150012761	0.157201109

Table 3: some text on digital call option prices given smoothed and non-smoothed payoffs

The reason for this is showcased in Figure 5. The figure shows how 4-5 point of different strikes is placed in between the spacial points. since the pay-off is just a 1-0 payoff it does

not matter for the solver how large or small the strike is. It will be valued the same, if it is between to points on the grid.
Obviously this is counter intuitive, so we are interested in smoothing out the payoff for our digital option. As implemented in our solver we just make the payoff a linear interpolation of the two points next to a given point x_i:

$$PO^{smoothed}(x_i) = \frac{e^{x_{i+1}} - K}{e^{x_{i+1}} - e^{x_{i-1}}}$$

Using this this we are able to somewhat smooth out discontinuities of the payoff function within an interval.

In general one could suggest various methods for smoothing payoff functions, $g(x)$ for instance:

$$\frac{1}{\Delta x} \int_{x_i - \Delta x/2}^{x_i + \Delta x/2} g(x) dx.$$

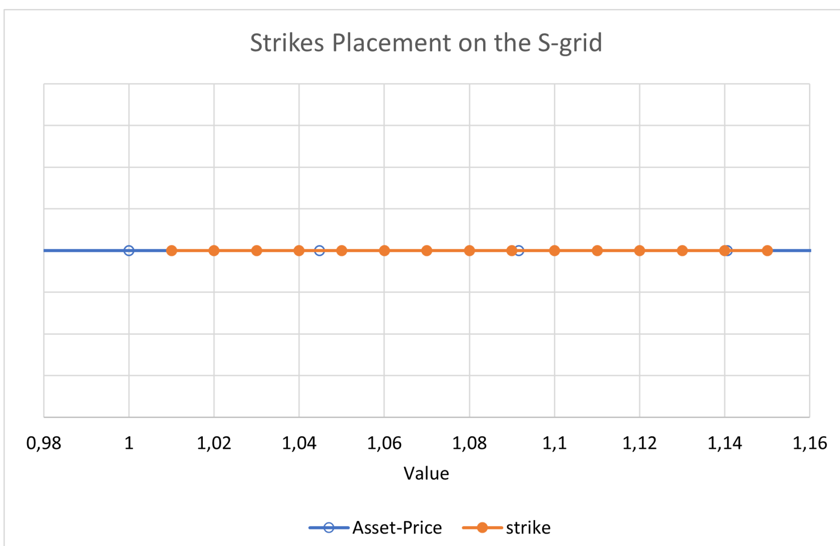


Figure 5: Explanation of Strikes on gridpoints

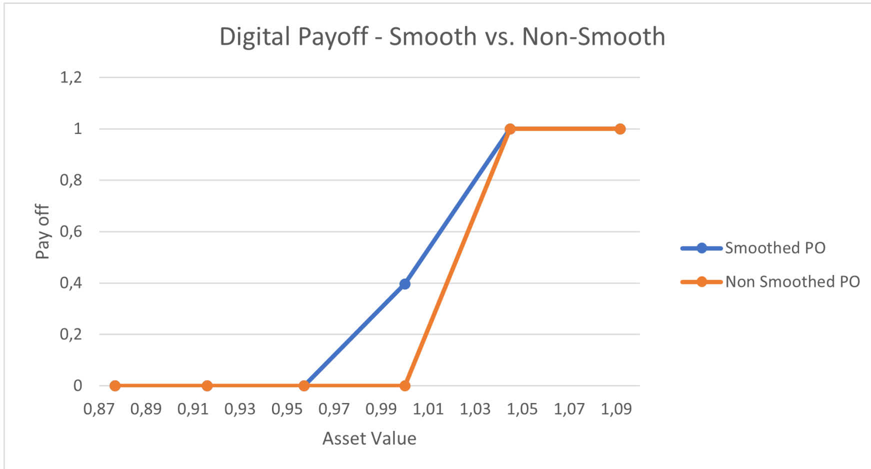


Figure 6: Example of smoothed payoff for Digital option, $K = 1.01$

4 Grid width and Sweet spot:

We take the our initial grid and considers the case for $m = 100$. This is kept fixed for the rest of the exercise. Likewise we fix our $\log \Delta S$ for this case, which is

$$\log \Delta S = \frac{\log S_{max} - \log S_{min}}{n} = \frac{5\sigma\sqrt{T} + \log S(0) - (\log S(0) - 5\sigma\sqrt{T}}{n} = 10\frac{\sigma\sqrt{T}}{n} \approx 0,04384447$$

Now we take and keep our $\Delta \log S$ fixed and change number of standard deviations. The results are shown in Figure 4. As we observe not much happens after 5 number of standard deviations not much change. This coincides quite well with the rule of thumb being 5 standard deviations for standard constant parameter models.
To answer why we consult Figure 5, which really shows how much probability area is covered, when taking 5 number of standard deviations on both sides. this is relevant since $\log S$ is normally distributed.

Num std	Imp Vol
1	0.1930169717034130
2	0.1998555659573630
3	0.1998395719913250
4	0.1998388875592440
5	0.1998388779433230
6	0.1998388778835730
7	0.1998388778834050
8	0.1998388778834040
9	0.1998388778834040
10	0.1998388778834040

Table 4: Imp. Vol to number of standard deviations - Fixed $\Delta \log S$.

std	right tail	left tail	Probability Covered
1	0.841344746068543	0.158655253931457	0.682689492137086
2	0.977249868051821	0.022750131948179	0.954499736103642
3	0.998650101968370	0.001349898031630	0.997300203936740
4	0.999968328758167	0.000031671241833	0.999936657516334
5	0.999999713348428	0.000000286651572	0.999999426696856
6	0.999999999013412	0.000000000986588	0.999999998026825
7	0.999999999998720	0.000000000001280	0.999999999997440
8	0.999999999999999	0.000000000000000	0.999999999999999
9	1.000000000000000	0.000000000000000	1.000000000000000
10	1.000000000000000	0.000000000000000	1.000000000000000

Table 5: The standard Gaussian CDF $\Phi(i \cdot \sigma)$ for different number of standard deviations and the respective tails and probability covered. Here we see that after 5 standard deviations our probability mass convered only changes on the 8 decimal and below.

5 Forward equation:

Up until now we have used the backward PDE, but we can modify our theta scheme to solve a forward scheme, but instead of solving for $v(t,x)$ we solve for the transition probabilities of the $p(t,x)$, where we go forward in time with our initial condition being

$$\{p(t_0, x_i)\}_{i=0,\dots,n} = \mathbf{1}_{x_i = x(0)}$$

Implementing the forward scheme we essentially solves the discrete version of Green's function for the backward linear system that is implemented to solve (2). The forward scheme in its discritized version is given as:

$$\begin{aligned} p(t_h) &= \left[I - \theta \Delta t \bar{A}\right]^\top p(t_{t+1/2}) \\ \left[I + (1-\theta)\Delta t \bar{A}\right]^\top p(t_{h+1/2}) &= p(t_{h+1}) \end{aligned} \tag{6}$$

The result is a full grid of probabilities $p(t_h, x_i)$.

To find a grid of strikes and expiry we firstly construct a grid of strikes. For convenience we choose our strike grid to be the same as our spacial grid ie $\{K_j\}_{j=0,\ldots,n} = \{S_i\}_{i=0,\ldots,n}$. Now we fix the grid (T, K) and for each point (T_h, K_j) we find the price by:

$$C(T_h, K_j) = \sum_{i=j}^{n} (e^{x_i} - K_j) \cdot p\left(T_h, x_i\right), \quad for\ j = 0, \ldots, n,\ h = 0, \ldots, m-1 \tag{7}$$

The results for a small grid for the prices can be found in Figure 7. Here we test this value with the price found for a similar grid for the `fdRunner()`-function, which is found by setting $K = 1$ and we get:

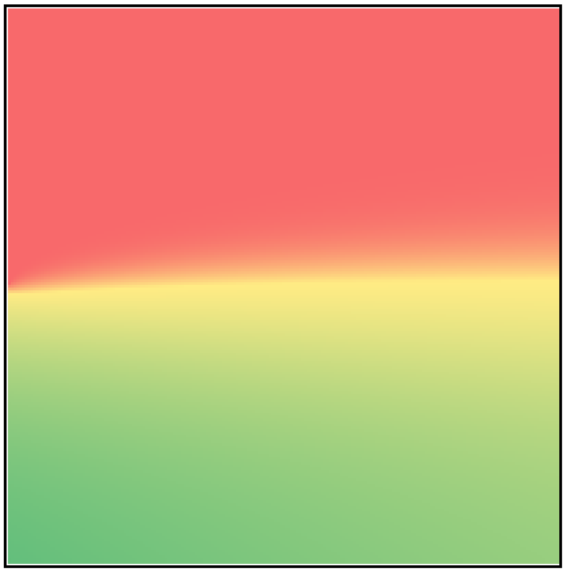
$$V(T = 5, S(0) = 0) = 0.07964998536319280$$

One can check for even more digits than can can be observed, bute two values are completely identical. This shows the duality of the forward and backward theta scheme.

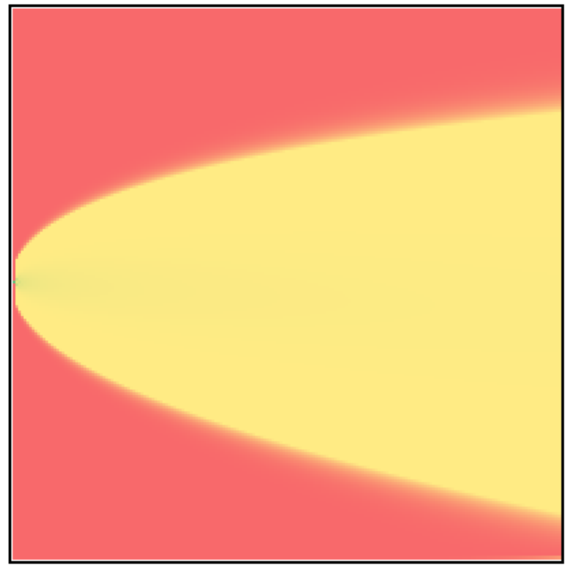
An even larger (T, K)-grid can be seen in Figure 8. Here the transistion probabilities are shown as well indicating how the asset is supposed two evolve over time given its log-normal distribution.

	Expiry																				
Strike	0,00	0,25	0,50	0,75	1,00	1,25	1,50	1,75	2,00	2,25	2,50	2,75	3,00	3,25	3,50	3,75	4,00	4,25	4,50	4,75	**5,00**
7,63535	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6,23083	-	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000
5,08468	-	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00001	0,00001	0,00001
4,14935	-	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00001	0,00001	0,00001	0,00002	0,00002	0,00003	0,00004	0,00005
3,38608	-	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00001	0,00001	0,00002	0,00002	0,00003	0,00005	0,00006	0,00008	0,00010	0,00013	0,00016	0,00020
2,76321	-	0,00000	0,00000	0,00000	0,00000	0,00000	0,00001	0,00001	0,00002	0,00004	0,00006	0,00009	0,00013	0,00017	0,00023	0,00029	0,00036	0,00044	0,00053	0,00062	0,00073
2,25492	-	0,00000	0,00000	0,00000	0,00001	0,00003	0,00006	0,00010	0,00017	0,00025	0,00036	0,00049	0,00063	0,00080	0,00099	0,00119	0,00141	0,00165	0,00189	0,00214	0,00240
1,84013	-	0,00000	0,00002	0,00005	0,00013	0,00027	0,00046	0,00071	0,00103	0,00139	0,00180	0,00225	0,00274	0,00325	0,00378	0,00432	0,00486	0,00541	0,00596	0,00650	0,00704
1,50164	-	0,00004	0,00023	0,00063	0,00124	0,00205	0,00300	0,00407	0,00522	0,00641	0,00763	0,00884	0,01004	0,01121	0,01234	0,01343	0,01447	0,01546	0,01641	0,01729	0,01813
1,22541	-	0,00087	0,00309	0,00598	0,00917	0,01242	0,01558	0,01858	0,02138	0,02398	0,02637	0,02854	0,03053	0,03232	0,03395	0,03542	0,03673	0,03792	0,03897	0,03992	0,04075
1,00000	-	0,01773	0,03100	0,04112	0,04898	0,05516	0,06010	0,06409	0,06733	0,06998	0,07215	0,07392	0,07538	0,07655	0,07749	0,07822	0,07878	0,07919	0,07946	0,07961	**0,07965**
0,81605	0,18395	0,17604	0,17058	0,16666	0,16369	0,16131	0,15930	0,15749	0,15579	0,15416	0,15255	0,15094	0,14933	0,14769	0,14604	0,14437	0,14269	0,14098	0,13927	0,13754	0,13580
0,66594	0,33406	0,32335	0,31313	0,30355	0,29462	0,28633	0,27861	0,27141	0,26466	0,25830	0,25230	0,24660	0,24117	0,23597	0,23099	0,22620	0,22157	0,21710	0,21278	0,20858	0,20450
0,54344	0,45656	0,44456	0,43282	0,42139	0,41028	0,39953	0,38914	0,37912	0,36947	0,36018	0,35123	0,34263	0,33433	0,32635	0,31864	0,31121	0,30403	0,29709	0,29038	0,28388	0,27758
0,44347	0,55653	0,54352	0,53078	0,51830	0,50608	0,49412	0,48244	0,47103	0,45990	0,44906	0,43849	0,42821	0,41820	0,40846	0,39900	0,38980	0,38085	0,37215	0,36370	0,35547	0,34748
0,36190	0,63810	0,62429	0,61074	0,59746	0,58443	0,57166	0,55914	0,54688	0,53487	0,52310	0,51158	0,50031	0,48929	0,47851	0,46797	0,45767	0,44761	0,43778	0,42818	0,41881	0,40965
0,29533	0,70467	0,69020	0,67599	0,66206	0,64839	0,63498	0,62182	0,60892	0,59626	0,58385	0,57168	0,55975	0,54805	0,53658	0,52534	0,51433	0,50355	0,49298	0,48263	0,47250	0,46258
0,24100	0,75900	0,74398	0,72924	0,71478	0,70058	0,68665	0,67298	0,65957	0,64640	0,63349	0,62081	0,60837	0,59617	0,58420	0,57246	0,56094	0,54964	0,53855	0,52768	0,51702	0,50657
0,19667	0,80333	0,78787	0,77270	0,75780	0,74318	0,72882	0,71473	0,70090	0,68733	0,67400	0,66092	0,64808	0,63548	0,62311	0,61097	0,59906	0,58737	0,57589	0,56463	0,55358	0,54273
0,16049	0,83951	0,82369	0,80816	0,79291	0,77794	0,76324	0,74880	0,73463	0,72072	0,70706	0,69365	0,68049	0,66757	0,65488	0,64242	0,63019	0,61818	0,60640	0,59483	0,58347	0,57232
0,13097	0,86903	0,85292	0,83709	0,82156	0,80630	0,79132	0,77661	0,76216	0,74797	0,73405	0,72037	0,70694	0,69375	0,68080	0,66808	0,65560	0,64334	0,63130	0,61948	0,60788	0,59648
0,10688	0,89312	0,87677	0,86071	0,84494	0,82945	0,81424	0,79930	0,78462	0,77021	0,75606	0,74217	0,72852	0,71512	0,70195	0,68903	0,67633	0,66387	0,65163	0,63960	0,62780	0,61621

Figure 7: Table for (T, K)-grid with $(m, n) = (20, 20)$ with related Call prices found by the `FdFwdRunner()` function.



(a) Heat map for Option Grid.



(b) Heat map for transition probabilities

Figure 8: Both subfigures are formatted ala traffic lighting, where green is the largest values of the grid, yellow represent the middle values and red being the lowest values. Subfigure (b) show the transition probabilities, $p(t_h, x_i)$for the underlying and (a) shows option prices for the (T, K) grid.

We now turn our attention to the what setting of the finite difference solver that guarantees $\delta_{xx}C(t_h) \geq 0$? At first we consider the discrete derivative being for (7) where we observe that:

$$
\begin{aligned}
0 \leq \delta_{SS}C(t_h) &= \delta_K \left(\delta_K \left\{ \sum_{i=j}^{n-1} (S_i - K_j) p(t_h, S_i) \right\} \right) \\
&= \delta_K \left[\frac{1}{\Delta K} \left\{ \sum_{i=j+1}^{n-1} (S_i - K_{j+1}) p(t_h, S_i) - \sum_{i=j}^{n-1} (S_i - K_j) p(t_h, S_i) \right\} \right] \\
&= \delta_K \left[\frac{-1}{\Delta K} \left\{ \sum_{i=j+1}^{n-1} \Delta K p(t_h, S_i) \right\} \right] = \delta_K \left\{ - \sum_{i=j+1}^{n-1} p(t_h, S_i) \right\} \\
&= - \sum_{j=0}^{n-1} \frac{\mathbf{1}_{S_i > K_{j+1}} - \mathbf{1}_{\{S_i > K_j\}}}{\Delta K} p(t_h, S_i) = \frac{1}{\Delta K} p(t_h, S_{j+1}) = \frac{1}{\Delta S} p(t_h, S_{j+1})
\end{aligned}
$$

Hence we can conclude that our transition probabilities $p(t_h, S_i)$ need to be positive in order for $\delta_{SS}C(t_h) \geq 0$. From the 'cheat sheet' from [1] we can induce that for $\theta = 1$ with winding always ensures positive transition probabilities.

At last we consider the forward Dupire scheme, where we set $\mu = r = 0$ which is defined as:

$$C(t_{h+1}) = \left[I + (1-\theta)\Delta t\bar{A}\right]\left[1 - \theta\Delta t\bar{A}\right]^{-1} C(t_h) \tag{8}$$

The only thing that has changed from our forward scheme in (6) is that the matrix operators are not transposed. We thus implement a version of the forward roll, where we do not transpose the matrices. We want to numerically show that (8) holds numerically. Hence we examine wether our forward roll ala Dupire yields the same results as our forward scheme for the $\mu = r = 0$ case. Size we want to show great the different tables and results we keep the grids fairly small hence $(n, m) = (10, 10)$. It is seen in Figure 9 that they numerically are the same.[1]

This is useful for fitting a local volatility surface of european option prices, since the Dupire forward scheme relates european option prices through the forward roll. The continous version of the Dupire PDE is as known from [2]:

$$\frac{\partial C}{\partial T} = \frac{1}{2}\sigma^2 K^2 \frac{\partial^2 C}{\partial^2 K}$$

We rearange to see that:

$$\sigma(K, T; S_0) = \sqrt{\frac{\frac{\partial C}{\partial T}}{K^2 \frac{\partial^2 C}{\partial^2 K}}}$$

Hence the local volatility function is uniquely determined by observed undiscounted option prices on a forward for an underlying asset.

[1] Even though some may think that it is exactly not equal to each other, Excel has some storage issues regarding doubles going past the 15th decimal: Link

Forward FD											
Strike	Expiry										
	0	0,5	1	1,5	2	2,5	3	3,5	4	4,5	5
6,4456	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
4,4403	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0001	0,0002	0,0002	0,0004
3,0588	0,0000	0,0000	0,0000	0,0000	0,0001	0,0002	0,0005	0,0008	0,0012	0,0017	0,0023
2,1072	0,0000	0,0000	0,0002	0,0006	0,0013	0,0024	0,0037	0,0054	0,0073	0,0096	0,0121
1,4516	0,0000	0,0010	0,0036	0,0075	0,0124	0,0179	0,0238	0,0301	0,0367	0,0433	0,0501
1,0000	0,0000	0,0245	0,0460	0,0650	0,0819	0,0971	0,1110	0,1236	0,1353	0,1461	0,1563
0,6889	0,3111	0,3118	0,3136	0,3163	0,3196	0,3234	0,3275	0,3319	0,3364	0,3410	0,3456
0,4746	0,5254	0,5255	0,5255	0,5257	0,5261	0,5266	0,5272	0,5280	0,5289	0,5300	0,5312
0,3269	0,6731	0,6731	0,6731	0,6731	0,6731	0,6732	0,6732	0,6733	0,6735	0,6736	0,6738
0,2252	0,7748	0,7748	0,7748	0,7748	0,7748	0,7748	0,7748	0,7748	0,7748	0,7748	0,7749
0,1551	0,8449	0,8449	0,8449	0,8449	0,8449	0,8449	0,8449	0,8449	0,8449	0,8449	0,8449
0,1069	0,8931	0,8931	0,8931	0,8931	0,8931	0,8931	0,8931	0,8931	0,8931	0,8931	0,8931

DUPIRE	Expiry										
Strike	0,0000	0,5000	1,0000	1,5000	2,0000	2,5000	3,0000	3,5000	4,0000	4,5000	5,0000
6,4456	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
4,4403	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0001	0,0002	0,0002	0,0004
3,0588	0,0000	0,0000	0,0000	0,0000	0,0001	0,0002	0,0005	0,0008	0,0012	0,0017	0,0023
2,1072	0,0000	0,0000	0,0002	0,0006	0,0013	0,0024	0,0037	0,0054	0,0073	0,0096	0,0121
1,4516	0,0000	0,0010	0,0036	0,0075	0,0124	0,0179	0,0238	0,0301	0,0367	0,0433	0,0501
1,0000	0,0000	0,0245	0,0460	0,0650	0,0819	0,0971	0,1110	0,1236	0,1353	0,1461	0,1563
0,6889	0,3111	0,3118	0,3136	0,3163	0,3196	0,3234	0,3275	0,3319	0,3364	0,3410	0,3456
0,4746	0,5254	0,5255	0,5255	0,5257	0,5261	0,5266	0,5272	0,5280	0,5289	0,5300	0,5312
0,3269	0,6731	0,6731	0,6731	0,6731	0,6731	0,6732	0,6732	0,6733	0,6735	0,6736	0,6738
0,2252	0,7748	0,7748	0,7748	0,7748	0,7748	0,7748	0,7748	0,7748	0,7748	0,7748	0,7749
0,1551	0,8449	0,8449	0,8449	0,8449	0,8449	0,8449	0,8449	0,8449	0,8449	0,8449	0,8449
0,1069	0,8931	0,8931	0,8931	0,8931	0,8931	0,8931	0,8931	0,8931	0,8931	0,8931	0,8931

ABS ERROR	Expiry										
Strike	0,0000	0,2500	0,5000	0,7500	1,0000	1,2500	1,5000	1,7500	2,0000	2,2500	2,5000
6,4456	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
4,4403	0,00E+00	1,32E-23	1,06E-22	4,24E-22	0,00E+00	3,39E-21	6,78E-21	2,71E-20	5,42E-20	1,63E-19	3,79E-19
3,0588	0,00E+00	6,35E-22	3,39E-21	0,00E+00	1,36E-20	5,42E-20	1,63E-19	2,17E-19	6,51E-19	1,30E-18	2,17E-18
2,1072	0,00E+00	6,78E-21	8,13E-20	0,00E+00	6,51E-19	1,30E-18	2,17E-18	3,47E-18	6,94E-18	1,04E-17	1,04E-17
1,4516	0,00E+00	2,17E-19	4,34E-19	0,00E+00	1,73E-18	1,04E-17	1,39E-17	2,08E-17	2,78E-17	2,78E-17	3,47E-17
1,0000	0,00E+00	3,47E-18	6,94E-18	1,39E-17	4,16E-17	2,78E-17	4,16E-17	4,16E-17	5,55E-17	8,33E-17	2,78E-17
0,6889	0,00E+00	5,55E-17	5,55E-17	5,55E-17	5,55E-17	5,55E-17	5,55E-17	5,55E-17	0,00E+00	5,55E-17	0,00E+00
0,4746	0,00E+00	0,00E+00	1,11E-16	0,00E+00	1,11E-16	1,11E-16	1,11E-16	0,00E+00	1,11E-16	1,11E-16	1,11E-16
0,3269	0,00E+00	1,11E-16	2,22E-16	1,11E-16	0,00E+00	1,11E-16	1,11E-16	0,00E+00	2,22E-16	1,11E-16	1,11E-16
0,2252	0,00E+00	1,11E-16	0,00E+00	0,00E+00	2,22E-16	0,00E+00	2,22E-16	3,33E-16	1,11E-16	1,11E-16	1,11E-16
0,1551	0,00E+00	1,11E-16	2,22E-16	2,22E-16	3,33E-16	3,33E-16	4,44E-16	4,44E-16	3,33E-16	4,44E-16	3,33E-16
0,1069	0,00E+00	1,11E-16	1,11E-16	1,11E-16	2,22E-16	0,00E+00	0,00E+00	3,33E-16	0,00E+00	3,33E-16	2,22E-16

Figure 9: Test of forward scheme with Dupires forward scheme.

6 Barrier:

At first we implement a Down-and-Out call, where the value of the option is set to 0 if the asset value hits or goes under a certain value called the barrier denoted B. This changes the payoff to:

$$IV_T = \begin{cases} (S_T - K)^+, & \text{if } \min_{0 \le t \le T} S_t > B \\ 0, & \text{if } \min_{0 \le t \le T} S_t \le B \end{cases} \tag{9}$$

We implement this by adding a boundary value condition to our PDE in (2):

$$V(t, x) = 0, \ if \ x \le \log(B), \quad for \ t \le T$$

So for every time step we check force $v(t, x_i) = 0$ if $x_i \le \log(B)$. This should after all reduce accuary in time, since we essentially price a discretely observed barrier option we expect a accuracy in time, Δt. We find for $\mathcal{O}(\Delta t^a)$ that $a \approx 1$, which is quite surprising, since it is expected that $a = 1/2$. However this may be explained by the small number of time steps analysed. It could take the discrete barrier a lot more to converge so even if we were to measure it up against a $m = 100.000$ it might still need to pump up m to get the expected order of convergence. One could have tested this more if time allowed it.

Nevertheless it is possible to modify the code to achieve better convergence. This is done by absorbtion($\mu = \sigma = 0$) at the barrier and placing the barrier on the grid. This achieves a convergence rate of $a \approx 2$.. The result can be found in Figure 10 and Table 6.

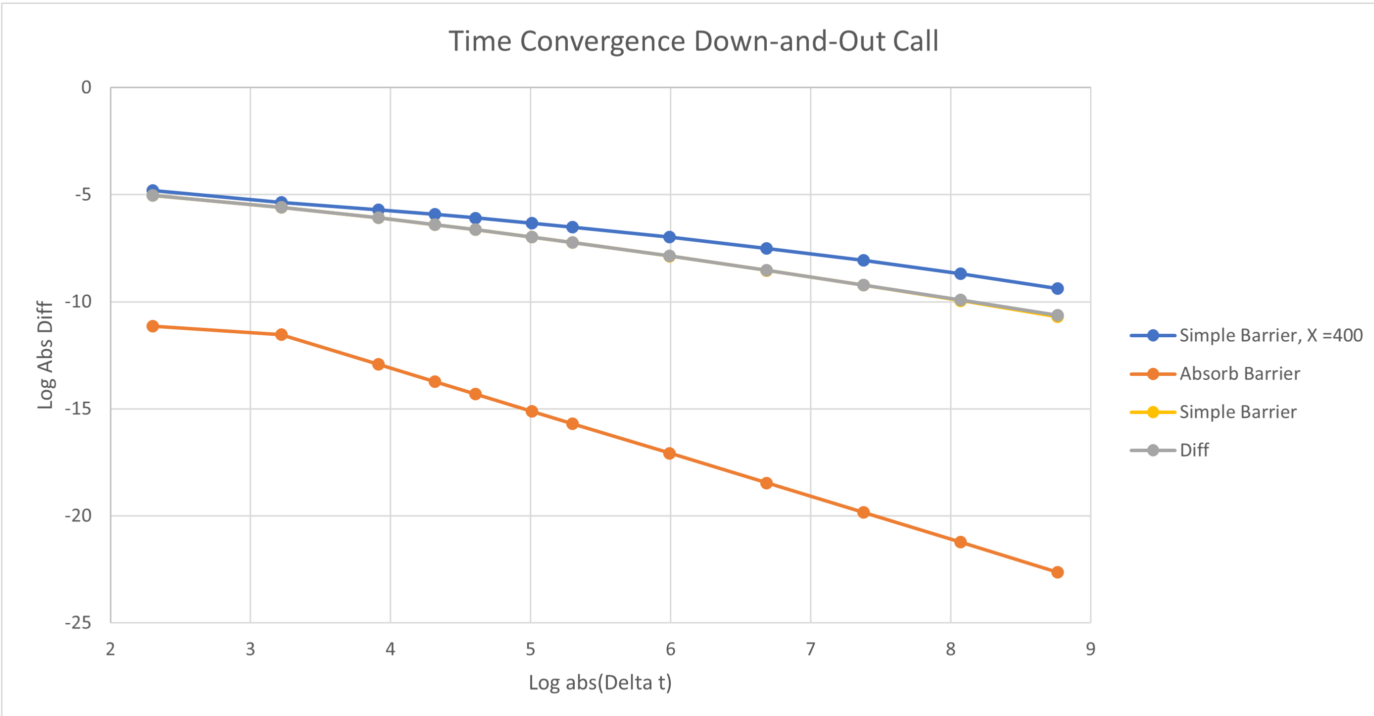


Figure 10: Down and out call time convergence

m	Simple Barrier	Absorb Barrier	Simple Barrier X=400
10	0.0681489498090903	0.0616078047383618	0.0712685498385125
25	0.0653908046062307	0.0616030851633704	0.0678480606230087
50	0.0639450811351924	0.0615957222610079	0.0664986175689473
75	0.0633294032420944	0.0615943574181758	0.0658374162680986
100	0.0629806434235437	0.0615938796320238	0.0654280842125784
150	0.0625950082479676	0.0615935383272682	0.0649359469316008
200	0.0623848689567275	0.0615934188649091	0.0646406247700183
400	0.0620422005316185	0.0615933036662624	0.0640855475956373
800	0.0618557984772617	0.0615932748661795	0.0637103155562729
1600	0.0617581032544487	0.0615932676661257	0.0634698419083173
3200	0.0617080166674308	0.0615932658660983	0.0633250019169038
6400	0.0616826471765202	0.0615932654160384	0.0632429106127364

Table 6: Values of barrier option for both the types of barrier, Discretely(simple) and continuously barriers with absorbtion.

Github Repository

References

[1] Jesper et. al Andreasen. "Fun with Finite Difference". In: Lecture notes (2022/2023).

[2] J. Gatheral. The Volatility Surface: A Practitioner's Guide. Wiley finance series. John Wiley & Sons, 2006. ISBN: 9781119202073. URL: https://books.google.dk/books?id=morbjwEACAAJ.