Numerical solution of differential equations
Project 1: Black-Scholes equations

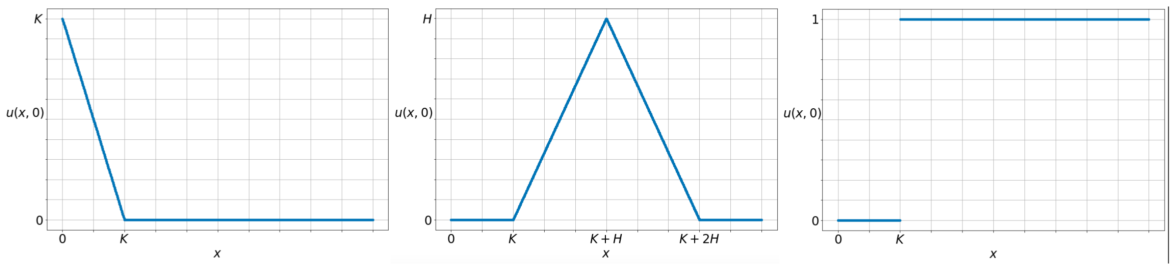
Lars August Melbye Olsen, Ida Sandum, Markus A. Stokkenes

Introduction

In this project, we analyse numerical methods for solving the Black-Scholes PDEs. This includes finding appropriate boundary conditions, deriving the numerical schemes, proving convergence and analysing and comparing the numerical error for the different methods.

We are given three different initial conditions.

- (a) European put: $u(x,0) = (K-x)^+$
- (b) Butterfly spread: $u(x,0) = (x-K)^+ - 2(x-(K+H))^+ + (x-(2H+K))^+$
- (c) Binary call: $u(x,0) = \text{sgn}^+(x-K)$



(a) (b) (c)

Part 1 – Linear case

The linear 1D Black-Scholes equation is given as

$$u_t - \frac{1}{2}\sigma^2 x^2 u_{xx} - rxu_x + cu = 0, \quad x \in \mathbb{R}^+ \quad t \in (0,T)$$
$$u(0,t) = g_0(t), \quad u(R,t) \overset{R\to\infty}{\to} g_1(t), \quad u(x,0) = f(x)$$

Boundary conditions

For the left boundary, we insert $x = 0$ and obtain the simple ODE

$$u_t(0,t) = -cu(0,t),$$

which, assuming c is positive, has the general solution $u(0,t) = Ce^{-ct}$ for some constant C. Inserting the initial condition yields $u(0,0) = C$, so our left boundary condition becomes

$$u(0,t) = f(0)e^{-ct}.$$

Inserting $x = 0$ into the three initial conditions yields $u(0,t) = Ke^{-ct}$ for European put and $u(0,t) = 0$ for the butterfly spread and binary call strategies. For the right boundary, we choose $u(R,t) \to 0$ for Eurpean put and butterfly spread, and $u(R,t) \to 1$ for binary call. This is sensible, looking at the plots of the three initial conditions as x gets large.

Forward Euler

For the forward Euler discretization, we use central differences in space and a forward difference in time. Replace the derivatives with differences:

$$\frac{U_m^{n+1} - U_m^n}{k} - \frac{1}{2}\sigma^2 x_m^2 \frac{U_{m-1}^n - 2U_m^n + U_{m+1}^n}{h^2} - rx_m \frac{U_{m+1}^n - U_{m-1}^n}{2h} + cU_m^n = 0.$$

Multiplying through by k, rearranging terms, factoring out the components of U and using the fact that $x_m = mh$, we obtain

$$U_m^{n+1} = (\frac{1}{2}km^2\sigma^2 - \frac{1}{2}kmr)U_{m-1}^n + (1 - km^2\sigma^2 - ck)U_m^n + (\frac{1}{2}km^2\sigma^2 + \frac{1}{2}kmr)U_{m+1}^n.$$

Define $\alpha_m = \frac{1}{2}km^2\sigma^2 - \frac{1}{2}kmr$, $\beta_m = km^2\sigma^2 + ck$, $\gamma = \frac{1}{2}km^2\sigma^2 + \frac{1}{2}kmr$. Then we can write the forward Euler scheme as follows:

$$\mathbf{U}^{n+1} = B\mathbf{U}^n + \mathbf{b}^n,$$

where

$$B = \text{tridiag}\{\alpha_m,\ 1 - \beta_m,\ \gamma_m\}, \quad \mathbf{b}^n = \begin{bmatrix}\alpha_1 g_0^n & 0 & \dots & 0 & \gamma_{M-1} g_1^n\end{bmatrix}^{\mathrm{T}}$$

Backward Euler

The backward Euler scheme is derived in a similar fashion, with backward time differences instead:

$$\frac{U_m^{n+1} - U_m^n}{k} - \frac{1}{2}\sigma^2 x_m^2 \frac{U_{m-1}^{n+1} - 2U_m^{n+1} + U_{m+1}^{n+1}}{h^2} - rx_m \frac{U_{m+1}^{n+1} - U_{m-1}^{n+1}}{2h} + cU_m^{n+1} = 0.$$

Again multiplying by k, rearranging terms, factoring and inserting $x_m = mh$ yields

$$(-\frac{1}{2}km^2\sigma^2 + \frac{1}{2}kmr)U_{m-1}^{n+1} + (1 + km^2\sigma^2 + ck)U_m^{n+1} + (-\frac{1}{2}km^2\sigma^2 - \frac{1}{2}kmr)U_{m+1}^{n+1}. = U_m^n$$

With α_m and β_m as before, the backward Euler scheme can be expressed as

$$A\mathbf{U}^{n+1} = \mathbf{U}^n + \mathbf{b}^{n+1},$$

with $A = \text{tridiag}\{-\alpha_m,\ 1 + \beta_m,\ -\gamma_m\}$ and $\mathbf{b}^{n+1}$ as before.

Crank-Nicolson

The Crank-Nicolson method combines forward Euler at n and backward Euler at $n+1$. With the usual notations for finite difference operators, the scheme becomes

$$\frac{1}{k}\Delta_t U_m^n - \frac{1}{2}\frac{\sigma^2 x_m^2}{h^2}(\delta_x^2 U_m^n + \delta_x^2 U_m^{n+1}) - \frac{1}{2}\frac{x_m}{2h}(\delta_x U_m^n + \delta_x U_m^{n+1}) + \frac{1}{2}c(U_m^n + U_m^{n+1}) = 0$$

After expanding the finite differences, rearranging such that $n+1$-terms are on the left and n-terms are on the right, the left hand side becomes

$$\frac{1}{2}(-\frac{1}{2}km^2\sigma^2 + \frac{1}{2}kmr)U_{m-1}^{n+1} + (1 + \frac{1}{2}km^2\sigma^2 + \frac{1}{2}ck)U_m^{n+1} + \frac{1}{2}(-\frac{1}{2}km^2\sigma^2 - \frac{1}{2}kmr)U_{m+1}^{n+1},$$

and similarly the right hand side becomes

$$\frac{1}{2}(\frac{1}{2}km^2\sigma^2 - \frac{1}{2}kmr)U_{m-1}^n + (1 - \frac{1}{2}km^2\sigma^2 - \frac{1}{2}ck)U_m^n + \frac{1}{2}(\frac{1}{2}km^2\sigma^2 + \frac{1}{2}kmr)U_{m+1}^n,$$

Setting the left hand side equal to the right hand side and writing the equation in terms of A, B and $\mathbf{b}$ yields

$$(I + A)\mathbf{U}^{n+1} = (I + B)\mathbf{U}^n + \mathbf{b}^{n+1} + \mathbf{b}^n$$

We have implemented these three schemes in our code.

Monotonicity

Forward Euler:
The scheme can be written as

$$U_m^{n+1} - \alpha_m U_{m-1}^n - (1-\beta_m)U_m^n - \gamma_m U_{m+1}^n = 0.$$

For monotonicity, we need α_m, $1-\beta_m$ and γ_m to be nonnegative, and for their sum to be less than 1. We have

- $\alpha_m \geq 0 \iff \frac{1}{2}km^2\sigma^2 - \frac{1}{2}kmr \geq 0 \iff \sigma^2 \geq \frac{r}{m}$
- $1-\beta_m \geq 0 \iff 1 - km^2\sigma^2 - ck \geq 0 \iff k \leq \frac{1}{\sigma^2 m^2 + c} < \frac{1}{\sigma^2 M^2 + c}$
- $\gamma_m \geq 0 \iff \frac{1}{2}km^2\sigma^2 + \frac{1}{2}kmr \geq 0$
- $1 \geq \alpha_m + \beta_m + \gamma_m \iff ck \geq 0$

Note that the first condition hold true because $\sigma^2 M^2 > r$, and the third and fourth conditions are always true, since all quantities are positive. The CFL condition is $k < \frac{1}{\sigma^2 M^2 + c}$, which needs to hold for Forward Euler to be monotone. We will use this to show stability later.

Backward Euler:
The scheme can be written as

$$(1+\beta_m)U_m^{n+1} - \alpha_m U_{m-1}^{n+1} - \gamma_m U_{m+1}^{n+1} - U_m^n = 0.$$

The monotonicity conditions are

- $\alpha_m \geq 0 \iff \frac{1}{2}km^2\sigma^2 - \frac{1}{2}kmr \geq 0 \iff \sigma^2 \geq \frac{r}{m}$
- $1+\beta_m \geq 0 \iff 1 + km^2\sigma^2 + ck \geq 0$
- $\gamma_m \geq 0 \iff \frac{1}{2}km^2\sigma^2 + \frac{1}{2}kmr \geq 0$
- $1+\beta_m \geq \alpha_m + \gamma_m + 1 \iff ck \geq 0$

In this case all the conditions are true, and the scheme is monotone unconditionally.

Crank-Nicolson:
The scheme can be written as

$$(1+\beta_m)U_m^{n+1} - \frac{1}{2}\alpha_m U_{m-1}^n - \frac{1}{2}\gamma_m U_{m+1}^{n+1} - \frac{1}{2}\alpha_m U_{m-1}^n - (1-\beta_m)U_m^n - \frac{1}{2}\gamma_m U_{m+1}^n = 0.$$

The only condition that is different from the other two schemes and does not always hold is the CFL condition

$$1 - \frac{1}{2}km^2\sigma^2 - \frac{1}{2}ck \geq 0 \iff k \leq \frac{2}{\sigma^2 m^2 + c} < \frac{2}{\sigma^2 M^2 + c}.$$

L^∞-stability

L^∞-stability for the forward Euler method:
If V solves the forward Euler scheme with respect to a right hand side f, such that $V_0 = 0 = V_M$ and $V^0 = \mathbf{0}$, we get the iteration

$$\begin{aligned}
\mathbf{V}^{n+1} &= B\mathbf{V}^n + k\mathbf{f}^n \\
&= B(B\mathbf{V}^{n-1} + k\mathbf{f}^{n-1}) + k\mathbf{f}^n \\
&= B^2(B\mathbf{V}^{n-2} + k\mathbf{f}^{n-2} + k(B\mathbf{f}^{n-1} + \mathbf{f}^n) \\
&\ldots \\
&= B^{n+1}\mathbf{V}^0 + k\Sigma_{l=0}^n A^l \mathbf{f}^{n-l},
\end{aligned}$$

where B is defined as before. Noticing that $\mathbf{V}^0 = \mathbf{0}$, and using the infinity norm on both sides, we get

$$||\mathbf{V}^{n+1}|| \leq k(n+1) \max_{l=0,..,N} ||\mathbf{f}^l|| \max_{l=0,...,N} ||B||^l = T \max_{l=0,..,N} ||\mathbf{f}^l|| \max_{l=0,...,N} ||B||^l.$$

Now we need to find a bound on $||B||$. We have that

$$||B||_\infty = \max_{1\leq i\leq n} \Sigma_{j=1}^n |b_{ij}| = |\alpha_m| + |1-\beta_m| + |\gamma_m| = |\frac{1}{2}km^2\sigma^2 - \frac{1}{2}kmr| + |1 - km^2\sigma^2 - ck| + |\frac{1}{2}km^2\sigma^2 + \frac{1}{2}kmr|.$$

From before we know that the first and the last term are always positive, so we can just remove the absolute values. Using the CFL condition for the scheme, we can also remove the absolute value of the middle term. Calculating yields

$$\max_{n=0,\dots,N} ||B||^n < (1+ck)^n \leq (e^{ck})^n \leq e^{cT} := L$$
$$\implies ||\mathbf{V}^{n+1}|| < C \max_{l=0,\dots,N} ||\mathbf{f}^l||,$$

with $LT = C$. This proves L^∞-stability.

Consistency

Consistency of the forward Euler method:
We have consistency if the local truncation error

$$\tau_m^n = (\frac{1}{k}\Delta_t - \frac{\sigma^2 x^2}{2h^2}\delta_x^2 - \frac{rx}{2h}\delta_x + c)u_m^n - (\partial_t - \frac{\sigma^2 x^2}{2}\partial_x^2 - rx\partial_x + c)u_m^n$$

goes to zero for all points (x_m, t_n) as $k, h \to 0$. To find an expression for τ_m^n we first Taylor expand $u(x_m + h, t)$, $u(x_m - h, t)$ and $u(x, t_m + k)$ in order to approximate the finite difference scheme on u.

- $u(x_m + h, t) = u(x_m, t) + h\partial_x u(x_m, t) + \frac{h^2}{2}\partial_x^2 u(x_m, t) + \frac{h^3}{6}\partial_x^3 u(x_m, t) + \frac{h^4}{24}\partial_x^4 u(\xi_1, t)$
- $u(x_m - h, t) = u(x_m, t) - h\partial_x u(x_m, t) + \frac{h^2}{2}\partial_x^2 u(x_m, t) - \frac{h^3}{6}\partial_x^3 u(x_m, t) + \frac{h^4}{24}\partial_x^4 u(\xi_1, t)$
- $u(x, t_n + k) = u(x, t_n) + k\partial_t u(x, t_n) + \frac{k^2}{2}\partial_t^2 u(x, t_n) + \frac{k^3}{6}\partial_t^3 u(x, \xi_3)$

Using the expansions instead of u_m^n in the first term and that $u(x_m, t_n) = u_m^n$ yields

$$\begin{aligned}
\frac{1}{k}\big(k\partial_t u(t_n) + \frac{k^2}{2}\partial_t^2 u(t_n) + \frac{k^3}{6}\partial_t^3 u(t_n)\big) - \frac{\sigma^2 x_m^2}{2}\big(\partial_x^2 u(x_m) + \frac{h^2}{12}\partial_x^4 u(x_m)\big) - \frac{rx_m}{2}\big(2\partial_x u(x_m) \\
+\frac{h^2}{3}\partial_x^3 u(x_m)\big) + cu_m^n - \partial_t u(t_n) + \frac{\sigma^2 x_m^2}{2}\partial_x^2 u(x_m) + rx_m\partial_x u(x_m) - cu_m^n + O(h^4) \\
= \frac{k}{2}\partial_t^2 u(t_n) - \frac{h^2 rx_m}{6}\partial_x^3 u(x_m) - \frac{h^2\sigma^2 x_m^2}{24}\partial_x^4 u(x_m) + O(h^4 + k^2)
\end{aligned}$$

$$\implies \tau_m^n = \frac{k}{2}\partial_t^2 u(t_n) - h^2\big(\frac{rx_m}{6}\partial_x^3 u(x_m) + \frac{\sigma^2 x_m^2}{24}\partial_x^4 u(x_m)\big) + O(h^4 + k^2).$$

The $O(h^4)$ part in the second line comes from the sixth order expansion term, the fifth order term is cancelled because of the second order central difference. We see that $\tau_m^n \to 0$ as $h, k \to 0$ for all m and n, as long as the derivatives are bounded, which shows consistency of the method of order 1 in time and order 2 in space.

Convergence

Convergence of the forward Euler method:
Let the error be $\mathbf{e}$. If $\mathbf{V} = \mathbf{e}$ solves our scheme with $\mathbf{f} = \tau$, we have $V_0 = 0 = V_M$ as there is no error at the boundary. Using stability and consistency, we thus get $|e_m^n| < C|\tau_m^n| \to 0$ as $h, k \to 0$. This shows convergence of order 1 in time and order 2 in space. Assuming $\left|\partial_t^2 u\right|$, $\left|\partial_x^3 u\right|$ and $\left|\partial_x^4 u\right|$ are less than infinity, we can bound this further. Letting $R = \max_m x_m$, we get

$$\begin{aligned}
\left|\tau_m^n\right| = \left|\frac{k}{2}\partial_t^2 u(t_n) - h^2(\frac{rx_m}{6}\partial_x^3 u(x_m) + \frac{\sigma^2 x_m^2}{24}\partial_x^4 u(x_m)) + O(h^4 + k^2)\right| \leq \\
\frac{k}{2}\max_{t\in(0,T),x\in\mathbb{R}^+}\left|\partial_t^2 u(x,t)\right| + h^2\big(\frac{rR}{6}\max_{t\in(0,T),x\in\mathbb{R}^+}\left|\partial_x^3 u(x,t)\right| + \frac{\sigma^2 R^2}{24}\max_{t\in(0,T),x\in\mathbb{R}^+}\left|\partial_x^4 u(x,t)\right|\big).
\end{aligned}$$

Now we have a bound for the error, using this bound on the local truncation error.

Numerical comparison, CFL and computational time

To get an exact solution we implement a right hand side $\varphi = e^{-t}sin(\pi x)$. With this comes a change to initial and boundary conditions. Our new I.C is $\varphi(x, 0) = sin(\pi x)$ while our B.C's are $\varphi(0, t) = 0$ and $\varphi(R, t) = e^{-t}sin(\pi R)$. To check that all three methods work with an exact solution we have plotted all three as well as the analytic solution in Figure 1. Now to check convergence for our three methods we add f to our methods:

- Forward Euler: we simply add f at the right side of the equation.

- Backward Euler: we add f to the right hand side before solving the linear system using numpy.linalg.solve.

- Crank-Nicolson: We add a right hand side f evaluated both at timestep n and n+1: $RHS = \frac{f^{n+1}+f^n}{2}$. We then add this right hand side, as in backward Euler, before solving the linear system.

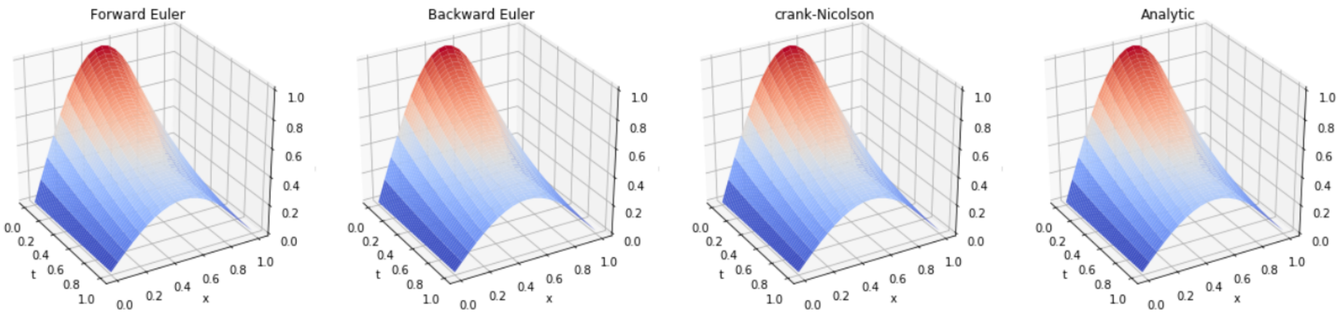


Figure 1: Comparison of the three methods against analytic solution

The CFL-condition is defined as in task b). To test it, we when running our forward Euler scheme, print our value for $1 - km^2\sigma^2 - ck \geq 0$, and seeing if and when it is less than zero. When we on purpose choose values so that the CFL-condition is not satisfied, we can see the solution either dropping to zero or oscillating.

From consistency above we know that Forward Euler has linear spatial convergence and quadratic temporal convergence. From theory we get that Backward Euler has linear spatial and quadratic temporal convergence and that Crank-Nicolson has quadratic both spatial and temporal convergence. To test our convergence rates we implement two functions convergence_h and convergence_k using the numpy.polyfit function. We can then see which degree of convergence we get in each the three methods both with regards to timesteps k and spatial steps h. The results shown in Figure 2 and Figure 3 are thus supported by theory.

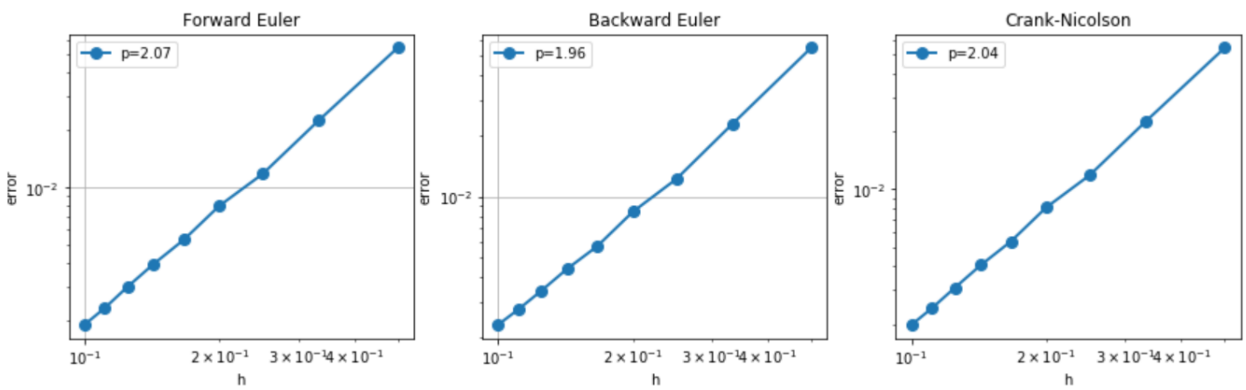


Figure 2: spatial convergence

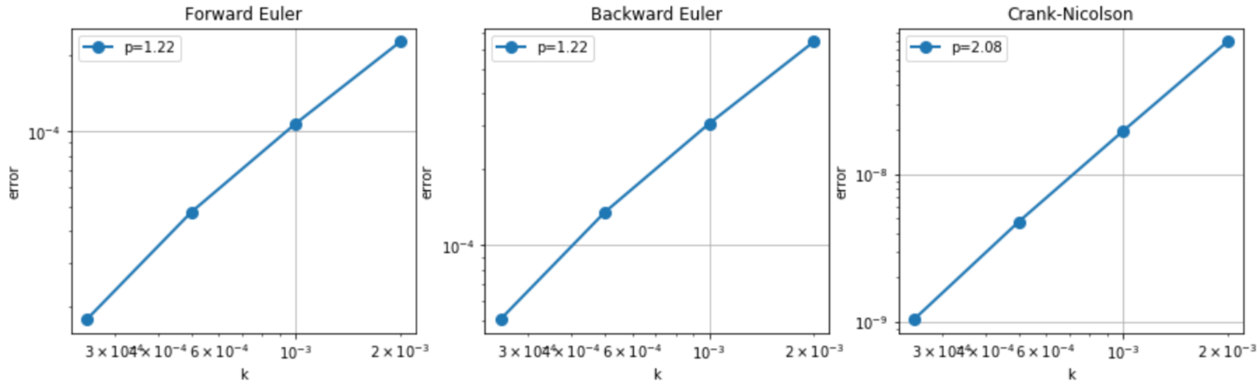


Figure 3: temporal convergence

Theoretically we should see that Forward Euler is quicker than Backward Euler and Crank-Nicolson is the slowest. This is due to the computational complexity in computing the next step. Both BE and CN require solving a linear system, and in CN's case this is more complicated.
To test the different methods runtime, we call each method 100 times with the same input variable. On our computer the result is that Forward Euler uses 3.321 seconds, Backward Euler uses 4.709 and Crank-Nicolson uses 4.963 seconds. This result is in accordance with the theory.

Variation in B.C's

Neumann boundary conditions:
Using a fictitious node x_{M+1}, we get Forward Euler schemes for the boundary conditions $u_x(R,t) = 0$ and $u_{xx}(R,t) = 0$. We will only show the derviation of the first scheme, but both schemes are included in the code.
We write the boundary condition as a central difference, where $x_M = R$:

$$U_x(R,t) = \frac{U^n_{M+1} - U^n_{M-1}}{2h} = 0 \implies U^n_{M+1} = U^n_{M-1} * .$$

Including the new internal point using the Forward Euler scheme yields

$$\frac{U^{n+1}_M - U^n_M}{k} - \frac{\sigma^2 x_M^2}{2h^2}(U^n_{M-1} - 2U^n_M + U^n_{M+1}) - \frac{rx_M}{2h}(U_{M+1} - U_{M-1}) = 0.$$

Now we can eliminate the fictitious node by using * in the equation above. We also multiply by k and rearrange:

$$U^{n+1}_M = (1-\beta_M)U^n_M + 2k\frac{\sigma^2 x_M^2}{2h^2}U^n_{M-1}.$$

All we do now is include this equation as a M-th row in the matrix B.
It seems like increasing R gives a bigger error. This could make sense as we have the same amount of spatial steps on a bigger space, which gives more inaccuracy.

Part 2 – Nonlinear case

We are given the nonlinear 1D Black-Scholes equation as

$$u_t - \frac{1}{2}x^2\varphi(u_{xx})u_{xx} = 0, \quad x \in \mathbb{R}^+ \quad t \in (0,T)$$
$$u(0,t) = g_0(t), \quad u(R,t) \overset{R\to\infty}{\to} g_1(t), \quad u(x,0) = f(x)$$

Boundary conditions

Setting $x = 0$ yields the ODE

$$u_t(0,t) = 0,$$

which has solution $u(0,t) = C$, where C is constant. Again inserting the initial condition gives

$$u(0,t) = f(0),$$

so for the left boundary we have $u(0,t) = K$ for European put and $u(0,t) = 0$ for the butterfly spread and binary call options. As before, we set $u(R,t) \to 0$ for European put and butterfly spread, and $u(R,t) \to 1$ for binary call.

IMEX

We discretize according to the IMEX method:

$$\frac{U_m^{n+1}-U_m^n}{k} = \frac{1}{2}x_m^2\varphi\Big(\frac{1}{h^2}\delta_x^2U_m^n\Big)\frac{U_{m+1}^{n+1}-2U_m^{n+1}+U_{m-1}^{n+1}}{h^2}.$$

With $\varphi_m^n = \varphi\Big(\frac{1}{h^2}\delta_x^2U_m^n\Big)$ and $x_m = mh$, the scheme becomes

$$-\frac{1}{2}km^2\varphi_m^nU_{m+1}^{n+1} + (1+km^2\varphi_m^n)U_m^{n+1} - \frac{1}{2}km^2\varphi_m^nU_{m-1}^{n+1} = U_m^n.$$

Define $\beta_m^n = \frac{1}{2}km^2\varphi_m^n$ and $\alpha_m^n = 1+km^2\varphi_m^n$ Then, similarly to before, we write the scheme as a linear system of equations:

$$A_n\mathbf{U}^{n+1} = \mathbf{U}^n + \mathbf{b}^{n+1},$$

where $A_n = \text{tridiag}\{-\beta_m^n,\ \alpha_m^n,\ -\beta_m^n\}$ and $\mathbf{b}^{n+1} = \begin{bmatrix}\beta_m^ng_0^{n+1} & 0 & \dots & 0 & \beta_m^ng_1^{n+1}\end{bmatrix}^{\mathrm{T}}$. The key difference this time around is that the matrix A_n is dependent on the current step, so it is updated for each iteration, as the notation indicates.

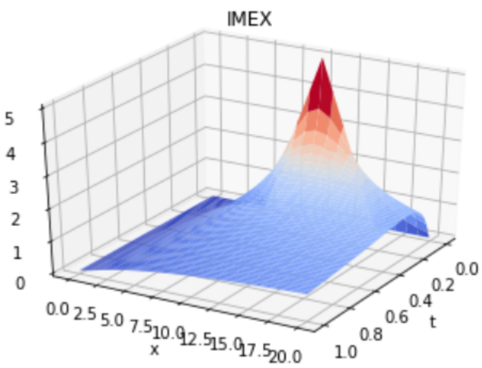


Figure 4: IMEX

Truncation Error

The truncation error of the IMEX method is given by

$$\frac{1}{k}\Delta_tU_m^{n+1} = \frac{1}{2}x_m^2\varphi(\frac{1}{h}\delta_x^2U_m^n)\frac{1}{h}\delta_x^2U_m^{n+1}$$

The truncation error of the BE method is given by

$$\frac{1}{k}\Delta_tU_m^{n+1} = \frac{1}{2}x_m^2\varphi(\frac{1}{h}\delta_x^2U_m^{n+1})\frac{1}{h}\delta_x^2U_m^{n+1}$$

As in task 1 d) we use a forward difference in time and a central difference in space. We also Taylor-expand the expression for φ as $\varphi(\partial_x^2u, \frac{h^2}{12}\partial_x^4u) = \varphi(\partial_x^2u) + \frac{h^2}{12}\partial_x^4u\cdot\partial_t\varphi(\partial_x^2u) + O(h^4)$. This solves for IMEX to

$$\tau_m^n = \frac{k}{2}\big(\partial_t^2u_m^n - x_m^2\partial_t\partial_x^2u_m^n\varphi(\partial_x^2u_m^n)\big) - \frac{1}{24}x_m^2h^2\big(\partial_x^4u_m^{n+1}\varphi(\partial_x^2u_m^n) + \partial_x^4u_m^n\cdot\partial_x^2u_m^{n+1}\cdot\partial_x\varphi(\partial_x^2u_m^n)\big) + O(k^2+h^3).$$

For BE we get

$$\tau_m^n = \frac{k}{2}\big(\partial_t^2u_m^n - x_m^2\partial_t\partial_x^2u_m^n\varphi(\partial_x^2u_m^n)\big) - \frac{1}{24}x_m^2h^2\partial_x^4u_m^{n+1}\big(\varphi(\partial_x^2u_m^{n+1} + \partial_x^4u_m^n\cdot\partial_x^2u_m^{n+1}\cdot\partial_x\varphi(\partial_x^2u_m^n)\big) + O(k^2+h^3).$$

Stability

Showing monotonicity for IMEX is straightforward. We need $\alpha_m^n > 0$ and $\beta_m^n \geq 0$, which is always true. Furthermore, we have $\beta_m^n + \beta_m^n = a_m^n \geq 0$, so the scheme has positive coefficients and is thus monotone, and the discrete maximum principle holds.

Next, consider the error equation $\vec{e}^n = \mathcal{L}_h\vec{\tau}^n$. We define the difference operator $\mathcal{L}_h := \frac{1}{k}\Delta_t - \frac{1}{2}x_m^2\varphi_m^n\frac{1}{h^2}\delta_x^2$, the auxiliary function $\psi(x,t) = t$ and the variable $W_m^n = e_m^n - \max_n \|\vec{\tau}^n\|_\infty \psi_m^n$.

Then we have

$$\begin{aligned}\mathcal{L}_h W_m^n &= \mathcal{L}_h(e_m^n - \psi_m^n \|\vec{\tau}^n\|_\infty) \\ &= \tau_m^n - t_n \|\vec{\tau}^n\|_\infty\end{aligned}$$

For the discrete maximum principle to hold for W_m^n, we need the above expression to be less than zero. We then have $\tau_m^n \leq t_n \|\vec{\tau}^n\|_\infty \leq T \|\vec{\tau}^n\|_\infty$

$$\max_m W_m^{n+1} \leq \max\{W_0^{n+1}, W_M^{n+1}, 0\} = 0$$

Thus,

$$\max \|\vec{e}^n\|_\infty \leq T \max \|\vec{\tau}^n\|_\infty$$

which proves stability in the max-infinity norm of the IMEX method.

Conclusion

We have implemented, tested and explored four different methods for solving PDEs, with different accuracy and computational time. Forward Euler is the worst, and Backward Euler and Crank-Nicolson are better.