MSc Course in Mathematics and Finance
Imperial College London, 2010-11

■

Finite Difference Methods

Mark Davis
Department of Mathematics
Imperial College London
www.ma.ic.ac.uk/∼mdavis

■

Part II
3. Numerical solution of PDEs

4. Numerical Solution of Partial Differential Equations

1. **Diffusion Equations of One State Variable.**

$$\frac{\partial u}{\partial t} = c^2 \frac{\partial^2 u}{\partial x^2}, \quad (x,t) \in D\,, \tag{1}$$

where t is a time variable, x is a state variable, and $u(x,t)$ is an unknown function satisfying the equation.

To find a well-defined solution, we need to impose the initial condition

$$u(x,0) = u_0(x) \tag{2}$$

and, if $D = [a,b] \times [0,\infty)$, the boundary conditions

$$u(a,t) = g_a(t) \quad \text{and} \quad u(b,t) = g_b(t)\,, \tag{3}$$

where u_0, g_a, g_b are continuous functions.

If $D = (-\infty, \infty) \times (0, \infty)$, we need to impose the boundary conditions

$$\lim_{|x| \to \infty} u(x,t)\, e^{-a\, x^2} = 0 \quad \text{for any } a > 0. \tag{4}$$

(4) implies $u(x,t)$ does not grow too fast as $|x| \to \infty$.

The diffusion equation (1) with the initial condition (2) and the boundary conditions (3) is well-posed, i.e. there exists a unique solution that depends continuously on u_0, g_a and g_b.

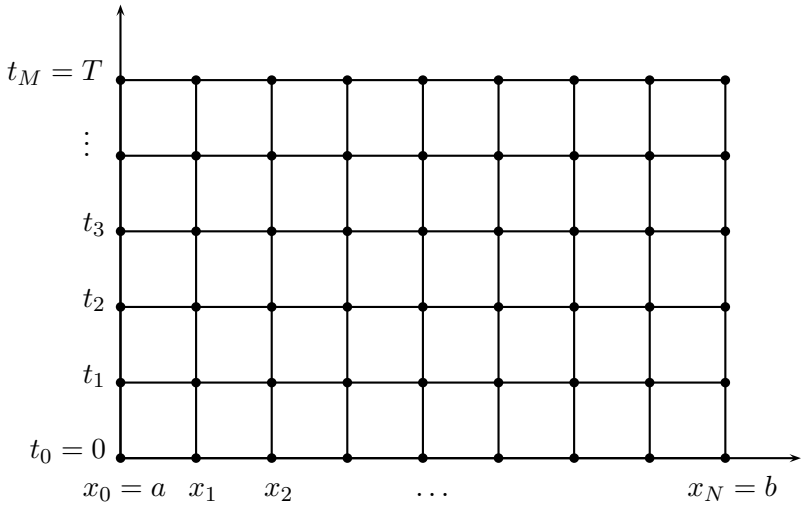
2. **Grid Points.**

To find a numerical solution to equation (1) with finite difference methods, we first need to define a set of grid points in the domain D as follows:

Choose a state step size $\Delta x = \frac{b-a}{N}$ (N is an integer) and a time step size Δt, draw a set of horizontal and vertical lines across D, and get all intersection points (x_j, t_n), or simply (j, n),

where $x_j = a + j\,\Delta x$, $j = 0, \ldots, N$, and $t_n = n\,\Delta t$, $n = 0, 1, \ldots$.

If $D = [a, b] \times [0, T]$ then choose $\Delta t = \frac{T}{M}$ (M is an integer) and $t_n = n\,\Delta t$, $n = 0, \ldots, M$.


$t_M = T$
$\vdots$
t_3
t_2
t_1
$t_0 = 0$
$x_0 = a$
x_1
x_2
$\ldots$
$x_N = b$

3. **Finite Differences.**

The partial derivatives

$$u_x := \frac{\partial u}{\partial x} \quad \text{and} \quad u_{xx} := \frac{\partial^2 u}{\partial x^2}$$

are always approximated by central difference quotients, i.e.

$$u_x \approx \frac{u^n_{j+1} - u^n_{j-1}}{2\,\Delta x} \;\text{ and }\; u_{xx} \approx \frac{u^n_{j+1} - 2\,u^n_j + u^n_{j-1}}{(\Delta x)^2} \tag{5}$$

at a grid point (j, n). Here $u^n_j = u(x_j, t_n)$.

Depending on how u_t is approximated, we have three basic schemes: explicit, implicit, and Crank–Nicolson schemes.

4. **Explicit Scheme.**

If u_t is approximated by a forward difference quotient

$$u_t \approx \frac{u_j^{n+1} - u_j^n}{\Delta t}$$

at (j, n),

then the corresponding difference equation to (1) at grid point (j, n) is

$$w_j^{n+1} = \lambda\, w_{j+1}^n + (1 - 2\,\lambda)\, w_j^n + \lambda\, w_{j-1}^n\,, \tag{6}$$

where

$$\lambda = c^2\, \frac{\Delta t}{(\Delta x)^2}\,.$$

The initial condition is $w_j^0 = u_0(x_j)$, $j = 0, \ldots, N$, and

the boundary conditions are $w_0^n = g_a(t_n)$ and $w_N^n = g_b(t_n)$, $n = 0, 1, \ldots$.

The difference equations (6), $j = 1, \ldots, N-1$, can be solved explicitly.

5. **Implicit Scheme.**

If u_t is approximated by a backward difference quotient

$$u_t \approx \frac{u_j^{n+1} - u_j^n}{\Delta t}$$

at $(j, n+1)$,

then the corresponding difference equation to (1) at grid point $(j, n+1)$ is

$$-\lambda\, w_{j+1}^{n+1} + (1 + 2\,\lambda)\, w_j^{n+1} - \lambda\, w_{j-1}^{n+1} = w_j^n\,. \tag{7}$$

The difference equations (7), $j = 1, \ldots, N-1$, together with the initial and boundary conditions as before, can be solved using the Crout algorithm or the SOR algorithm.

Explicit Method.

$$\frac{w_j^{n+1} - w_j^n}{\Delta t} = c^2 \, \frac{w_{j-1}^n - 2\, w_j^n + w_{j+1}^n}{(\Delta x)^2}$$

Letting $\lambda := c^2 \, \frac{\Delta t}{(\Delta x)^2}$ gives (6).

Implicit Method.

$$\frac{w_j^{n+1} - w_j^n}{\Delta t} = c^2 \, \frac{w_{j-1}^{n+1} - 2\, w_j^{n+1} + w_{j+1}^{n+1}}{(\Delta x)^2}$$

Letting $\lambda := c^2 \, \frac{\Delta t}{(\Delta x)^2}$ gives (7).
In matrix form

$$\begin{pmatrix} 1+2\,\lambda & -\lambda & & \\ -\lambda & 1+2\,\lambda & -\lambda & \\ & \ddots & \ddots & \\ & & -\lambda & 1+2\,\lambda \end{pmatrix} w = b\,.$$

The matrix is tridiagonal and diagonally dominant. $\Rightarrow$ Crout / SOR.

6. **Crank–Nicolson Scheme.**

The Crank–Nicolson scheme is the average of the explicit scheme at (j, n) and the implicit scheme at $(j, n+1)$.

The resulting difference equation is

$$-\frac{\lambda}{2} w_{j-1}^{n+1} + (1+\lambda) w_j^{n+1} - \frac{\lambda}{2} w_{j+1}^{n+1} = \frac{\lambda}{2} w_{j-1}^{n} + (1-\lambda) w_j^{n} + \frac{\lambda}{2} w_{j+1}^{n} . \quad (8)$$

The difference equations (8), $j = 1, \ldots, N-1$, together with the initial and boundary conditions as before, can be solved using Crout algorithm or SOR algorithm.

Crank–Nicolson.

$\frac{1}{2}\,[(\dagger) + (\ddagger)]$ gives

$$\frac{w_j^{n+1} - w_j^n}{\Delta t} = \frac{1}{2}\,c^2\,\frac{w_{j-1}^n - 2\,w_j^n + w_{j+1}^n}{(\Delta x)^2} + \frac{1}{2}\,c^2\,\frac{w_{j-1}^{n+1} - 2\,w_j^{n+1} + w_{j+1}^{n+1}}{(\Delta x)^2}$$

Letting $\mu := \frac{1}{2}\,c^2\,\frac{\Delta t}{(\Delta x)^2} = \frac{\lambda}{2}$ gives

$$\begin{aligned} -\,\mu\,w_{j+1}^{n+1} + (1 + 2\,\mu)\,w_j^{n+1} - \mu\,w_{j-1}^{n+1} &= \widehat{w}_j^{n+1} \\ \text{and} \quad \widehat{w}_j^{n+1} &= \mu\,w_{j+1}^n + (1 - 2\,\mu)\,w_j^n + \mu\,w_{j-1}^n\,. \end{aligned}$$

This can be interpreted as

$\widehat{w}_j^{n+1}$	—	predictor	(explicit method)
w_j^{n+1}	—	corrector	(implicit method)

7. **Local Truncation Errors.**

These are measures of the error by which the exact solution of a differential equation does not satisfy the difference equation at the grid points and are obtained by substituting the exact solution of the continuous problem into the numerical scheme.

A necessary condition for the convergence of the numerical solutions to the continuous solution is that the local truncation error tends to zero as the step size goes to zero. In this case the method is said to be *consistent*.

It can be shown that all three methods are consistent.

The explicit and implicit schemes have local truncation errors $O(\Delta t, (\Delta x)^2)$, while that of the Crank–Nicolson scheme is $O((\Delta t)^2, (\Delta x)^2)$.

Local Truncation Error.
For the explicit scheme we get for the LTE at (j, n)

$$\mathrm{E}_j^n = \frac{u(x_j, t_{n+1}) - u(x_j, t_n)}{\Delta t} - c^2 \, \frac{u(x_{j-1}, t_n) - 2\, u(x_j, t_n) + u(x_{j+1}, t_n)}{(\Delta x)^2} \, .$$

With the help of a Taylor expansion at (x_j, t_n) we find that

$$\begin{aligned} \frac{u(x_j, t_{n+1}) - u(x_j, t_n)}{\Delta t} &= u_t(x_j, t_n) + O(\Delta t) \, , \\ \frac{u(x_{j-1}, t_n) - 2\, u(x_j, t_n) + u(x_{j+1}, t_n)}{(\Delta x)^2} &= u_{xx}(x_j, t_n) + O((\Delta x)^2) \, . \end{aligned}$$

Hence

$$\mathrm{E}_j^n = \underbrace{u_t(x_j, t_n) - c^2 \, u_{xx}(x_j, t_n)}_{=\,0} + O(\Delta t) + O((\Delta x)^2) \, .$$

8. **Numerical Stability.**

Consistency is only a necessary but not a sufficient condition for convergence.

Roundoff errors incurred during calculations may lead to a blow up of the solution or erode the whole computation.

A scheme is *stable* if roundoff errors are not amplified in the calculations.

The *Fourier method* can be used to check if a scheme is stable.

Assume that a numerical scheme admits a solution of the form

$$v_j^n = a^{(n)}(\omega)\, e^{i\, j\, \omega\, \Delta x}\,, \tag{9}$$

where ω is the wave number and $i = \sqrt{-1}$.

Define

$$G(\omega) = \frac{a^{(n+1)}(\omega)}{a^{(n)}(\omega)},$$

where $G(\omega)$ is an amplification factor, which governs the growth of the Fourier component $a(\omega)$.

The *von Neumann stability condition* is given by

$$|G(\omega)| \leq 1$$

for $0 \leq \omega \Delta x \leq \pi$.

It can be shown that the explicit scheme is stable if and only if $\lambda \leq \frac{1}{2}$, called *conditionally stable*, and the implicit and Crank–Nicolson schemes are stable for any values of λ, called *unconditionally stable*.

Stability Analysis.

For the explicit scheme we get on substituting (9) into (6) that

$$a^{(n+1)}(\omega)\, e^{i\, j\, \omega\, \Delta x} = \lambda\, a^{(n)}(\omega)\, e^{i\,(j+1)\,\omega\,\Delta x} + (1-2\,\lambda)\, a^{(n)}(\omega)\, e^{i\, j\, \omega\, \Delta x} + \lambda\, a^{(n)}(\omega)\, e^{i\,(j-1)\,\omega\,\Delta x}$$

$$\Rightarrow \quad G(\omega) = \frac{a^{(n+1)}(\omega)}{a^{(n)}(\omega)} = \lambda\, e^{i\,\omega\,\Delta x} + (1-2\,\lambda) + \lambda\, e^{-i\,\omega\,\Delta x}\,.$$

The von Neumann stability condition then is

$$\begin{aligned}
|G(\omega)| \le 1 \qquad &\Longleftrightarrow \qquad |\lambda\, e^{i\,\omega\,\Delta x} + (1-2\,\lambda) + \lambda\, e^{-i\,\omega\,\Delta x}| \le 1 \\
&\Longleftrightarrow \qquad |(1-2\,\lambda) + 2\,\lambda\, \cos(\omega\,\Delta x)| \le 1 \\
&\Longleftrightarrow \qquad |1 - 4\,\lambda\, \sin^2\left(\frac{\omega\,\Delta x}{2}\right)| \le 1 \qquad [\cos 2\,\alpha = 1 - 2\,\sin^2\alpha] \\
&\Longleftrightarrow \qquad 0 \le 4\,\lambda\, \sin^2\left(\frac{\omega\,\Delta x}{2}\right) \le 2 \\
&\Longleftrightarrow \qquad 0 \le \lambda \le \frac{1}{2\,\sin^2\left(\frac{\omega\,\Delta x}{2}\right)}
\end{aligned}$$

for all $0 \le \omega\,\Delta x \le \pi$.

This is equivalent to $0 \le \lambda \le \frac{1}{2}$.

Remark.
The explicit method is stable, if and only if

$$\Delta t \leq \frac{(\Delta x)^2}{2\,c^2}. \qquad (\dagger)$$

($\dagger$) is a strong restriction on the time step size Δt. If Δx is reduced to $\frac{1}{2}\,\Delta x$, then Δt must be reduced to $\frac{1}{4}\,\Delta t$.
So the total computational work increases by a factor 8.
Example.

$$u_t = u_{xx} \qquad (x,t) \in [0,1] \times [0,1]$$

Take $\Delta x = 0.01$. Then

$$\lambda \leq \frac{1}{2} \quad \Rightarrow \qquad \Delta t \leq 0.00005$$

I.e. the number of grid points is equal to

$$\frac{1}{\Delta x}\,\frac{1}{\Delta t} = 100 \times 20,000 = 2 \times 10^6\,.$$

Remark.
In vector notation, the explicit scheme can be written as

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

where $w^n = (w_1^n, \ldots, w_{N-1}^n)^T \in \mathbb{R}^{N-1}$ and

$$A = \begin{pmatrix} 1-2\,\lambda & \lambda & & \\ \lambda & 1-2\,\lambda & \lambda & \\ & \ddots & \ddots & \\ & & \lambda & 1-2\,\lambda \end{pmatrix} \in \mathbb{R}^{(N-1)\times(N-1)}, \qquad b^n = \begin{pmatrix} \lambda\, w_0^n \\ 0 \\ \vdots \\ 0 \\ \lambda\, w_N^n \end{pmatrix} \in \mathbb{R}^{N-1}\,.$$

For the implicit method we get

$$B\, w^{n+1} = w^n + b^{n+1}\,, \qquad \text{where} \quad B = \begin{pmatrix} 1+2\,\lambda & -\lambda & & \\ -\lambda & 1+2\,\lambda & -\lambda & \\ & \ddots & \ddots & \\ & & -\lambda & 1+2\,\lambda \end{pmatrix}\,.$$

Remark.
Forward diffusion equation: $u_t - c^2\, u_{xx} = 0 \qquad t \geq 0.$
Backward diffusion equation:

$$\begin{aligned} u_t + c^2\, u_{xx} &= 0 && t \leq T \\ u(x,T) &= u_T(x) && \forall\ x \\ u(a,t) = g_a(t), \quad u(b,t) &= g_b(t) && \forall\ t \qquad \text{[as before]} \end{aligned}$$

[Note: We could use the transformation $v(x,t) := u(x, T-t)$ in order to transform this into a standard forward diffusion problem.]
We can solve the backward diffusion equation directly by starting at $t = T$ and solving "backwards", i.e. given w^{n+1}, find w^n.

$$\text{Implicit: } w^{n+1} = \widetilde{A}\, w^n + \tilde{b}^n \qquad \text{Explicit: } \widetilde{B}\, w^{n+1} = w^n + \tilde{b}^n$$

The von Neumann stability condition for the backward problem then becomes

$$|\widetilde{G}(\omega)| = \left| \frac{a^n(\omega)}{a^{n+1}(\omega)} \right| \leq 1\,.$$

Stability of the Binomial Model.

The binomial model is an explicit method for a backward equation.

$$V_j^n = \frac{1}{R}\left(p\,V_{j+1}^{n+1} + (1-p)\,V_{j-1}^{n+1}\right) = \frac{1}{R}\left(p\,V_{j+1}^{n+1} + 0\,V_j^{n+1} + (1-p)\,V_{j-1}^{n+1}\right)$$

for $j = -n, -n+2, \ldots, n-2, n$ and $n = N-1, \ldots, 1, 0$.

Here the initial values $V_{-N}^N, V_{-N+2}^N, \ldots, V_{N-2}^N, V_N^N$ are given.

Now let $V_j^n = a^{(n)}(\omega)\, e^{i\,j\,\omega\,\Delta x}$, then

$$\begin{aligned}
a^{(n)}(\omega)\, e^{i\,j\,\omega\,\Delta x} &= \frac{1}{R}\left(p\,a^{(n+1)}(\omega)\, e^{i\,(j+1)\,\omega\,\Delta x} + (1-p)\,a^{(n+1)}(\omega)\, e^{i\,(j-1)\,\omega\,\Delta x}\right) \\
\Rightarrow \quad \widetilde{G}(\omega) &= \left(p\,e^{i\,\omega\,\Delta x} + (1-p)\,e^{-i\,\omega\,\Delta x}\right) e^{-r\,\Delta t} \\
&= \left(\cos(\omega\,\Delta x) + \underbrace{(2\,p-1)}_{=\,q}\, i\,\sin(\omega\,\Delta x)\right) e^{-r\,\Delta t} \\
\Rightarrow \quad |\widetilde{G}(\omega)|^2 &= \left(\cos^2(\omega\,\Delta x) + q^2\,\sin^2(\omega\,\Delta x)\right) e^{-2\,r\,\Delta t} \\
&= \left(1 + (q^2-1)\,\sin^2(\omega\,\Delta x)\right) e^{-2\,r\,\Delta t} \\
&\le e^{-2\,r\,\Delta t} \le 1
\end{aligned}$$

if $q^2 \le 1 \iff -1 \le q \le 1 \iff p \in [0,1]$. Hence the binomial model is stable.

Stability of the CRR Model.

We know that the binomial model is stable if $p \in (0,1)$.

For the CRR model we have that

$$u = e^{\sigma\sqrt{\Delta t}}, \quad d = e^{-\sigma\sqrt{\Delta t}}, \quad p = \frac{R-d}{u-d},$$

so $p \in (0,1)$ is equivalent to $u > R > d$.

Clearly, for Δt small, we can ensure that

$$e^{\sigma\sqrt{\Delta t}} > e^{r\,\Delta t}.$$

Hence the CRR model is stable, if Δt is sufficiently small, i.e. if $\Delta t < \frac{\sigma^2}{r^2}$.

Alternatively, one can argue (less rigorously) as follows. Since $\Delta x = u\,S - S = S\,(e^{\sigma\sqrt{\Delta t}} - 1) \approx S\,\sigma\sqrt{\Delta t}$ and as the BSE can be written as

$$u_t + \underbrace{\frac{1}{2}\,\sigma^2\,S^2}_{c^2}\,u_{SS} + \ldots = 0\,,$$

it follows that

$$\lambda = c^2\,\frac{\Delta t}{(\Delta x)^2} = \frac{1}{2}\,\sigma^2\,S^2\,\frac{\Delta t}{S^2\,\sigma^2\,\Delta t} = \frac{1}{2} \qquad \Rightarrow \quad \text{CRR is stable.}$$

9. **Simplification of the BSE.**

Assume $V(S,t)$ is the price of a European option at time t.

Then V satisfies the Black–Scholes equation (**??**) with appropriate initial and boundary conditions.

Define

$$\tau = T - t, \quad x = \ln S, \quad w(\tau, x) = e^{\alpha\, x + \beta\, \tau}\, V(t, S)\,,$$

where α and β are parameters.

Then the Black–Scholes equation can be transformed into a basic diffusion equation:

$$\frac{\partial w}{\partial \tau} = \frac{1}{2}\,\sigma^2\,\frac{\partial^2 w}{\partial x^2}$$

with a new set of initial and boundary conditions.

Finite difference methods can be used to solve the corresponding difference equations and hence to derive option values at grid points.

Transformation of the BSE.

Consider a call option.

Let $\tau = T - t$ be the remaining time to maturity. Set $u(x, \tau) = V(x, t)$. Then $\frac{\partial u}{\partial \tau} = -\frac{\partial V}{\partial t}$ and the BSE (**??**) is equivalent to

$$u_\tau = \frac{1}{2}\,\sigma^2\,S^2\,u_{SS} + r\,S\,u_S - r\,u\,, \qquad (\dagger)$$

$$u(S,0) = V(S,T) = (S-X)^+\,, \qquad \text{(IC)}$$

$$u(0,\tau) = V(0,t) = 0\,, \quad u(S,\tau) = V(S,t) \approx S \quad \text{as } S \to \infty. \qquad \text{(BC)}$$

Let $x = \ln S$ ($\iff$ $S = e^x$). Set $\tilde{u}(x, \tau) = u(S, \tau)$. Then

$$\tilde{u}_x = u_S\,e^x = S\,u_S\,, \qquad \tilde{u}_{xx} = S\,u_S + S^2\,u_{SS}$$

and $(\dagger)$ becomes

$$\tilde{u}_\tau = \frac{1}{2}\,\sigma^2\,\tilde{u}_{xx} + (r - \frac{1}{2}\,\sigma^2)\,\tilde{u}_x - r\,\tilde{u}\,, \qquad (\ddagger)$$

$$\tilde{u}(x,0) = u(S,0) = (e^x - X)^+\,, \qquad \text{(IC)}$$

$$\tilde{u}(x,\tau) = u(0,\tau) = 0 \quad \text{as } x \to -\infty\,, \quad \tilde{u}(x,\tau) = u(e^x,\tau) \approx e^x \quad \text{as } x \to \infty. \qquad \text{(BC)}$$

Note that the growth condition (4), $\lim_{|x|\to\infty} \tilde{u}(x,\tau)\, e^{-a\,x^2} = 0$ for any $a > 0$, is satisfied. Hence (‡) is well defined.
Let $w(x,\tau) = e^{\alpha\,x+\beta\,\tau}\,\tilde{u}(x,\tau) \iff \tilde{u}(x,\tau) = e^{-\alpha\,x-\beta\,\tau}\,w(x,\tau) =: C\,w(x,\tau)$.
Then

$$\begin{aligned}
\tilde{u}_\tau &= C\,(-\beta\,w + w_\tau)\\
\tilde{u}_x &= C\,(-\alpha\,w + w_x)\\
\tilde{u}_{xx} &= C\,(-\alpha\,(-\alpha\,w + w_x) + (-\alpha\,w_x + w_{xx})) = C\,(\alpha^2\,w - 2\,\alpha\,w_x + w_{xx})\,.
\end{aligned}$$

So (‡) is equivalent to

$$C\,(-\beta\,w + w_\tau) = \frac{1}{2}\,\sigma^2\,C\,(\alpha^2\,w - 2\,\alpha\,w_x + w_{xx}) + (r - \frac{1}{2}\,\sigma^2)\,C\,(-\alpha\,w + w_x) - r\,C\,w\,.$$

In order to cancel the w and w_x terms we need to have

$$\begin{cases} -\beta = \frac{1}{2}\,\sigma^2\,\alpha^2 - (r - \frac{1}{2}\,\sigma^2)\,\alpha - r\,, \\ \quad 0 = \frac{1}{2}\,\sigma^2\,(-2\,\alpha) + r - \frac{1}{2}\,\sigma^2\,. \end{cases} \iff \begin{cases} \alpha = \frac{1}{\sigma^2}\,(r - \frac{1}{2}\,\sigma^2)\,, \\ \beta = \frac{1}{2\,\sigma^2}\,(r - \frac{1}{2}\,\sigma^2)^2 + r\,. \end{cases}$$

With this choice of α and β the equation ($\ddagger$) is equivalent to

$$w_\tau = \frac{1}{2}\,\sigma^2\, w_{xx}\,, \qquad (\sharp)$$

$$w(x,0) = e^{\alpha\, x}\,\tilde{u}(x,0) = e^{\alpha\, x}\,(e^x - X)^+\,, \qquad \text{(IC)}$$

$$w(x,\tau) = 0 \quad \text{as } x \to -\infty\,, \quad w(x,\tau) \approx e^{\alpha\, x + \beta\, \tau}\, e^x \quad \text{as } x \to \infty. \qquad \text{(BC)}$$

Note that the growth condition (4) is satisfied. Hence ($\sharp$) is well defined.

Implementation.

1. Choose a truncated interval $[a,b]$ to approximate $(-\infty,\infty)$.
   $e^{-8} = 0.0003$, $e^8 = 2981$ $\quad\Rightarrow\quad$ $[a,b] = [-8,8]$ serves all practical purposes.

2. Choose integers N, M to get the step sizes $\Delta x = \frac{b-a}{N}$ and $\Delta\tau = \frac{T-t}{M}$.
   Grid points (x_j, τ_n):
   $x_j = a + j\,\Delta x$, $j = 0, 1, \ldots, N$ and $\tau_n = n\,\Delta\tau$, $n = 0, 1, \ldots, M$.
   Note: x_0, x_N and τ_0 represent the boundary of the grid with known values.

3. Solve (♯) with

$$w(x,0) = e^{\alpha x}\,(e^x - X)^+\,, \tag{IC}$$

$$w(a,\tau) = 0\,, \quad w(b,\tau) = \begin{cases} e^{(\alpha+1)\,b+\beta\,\tau} & \text{or} \\ e^{\alpha\,b}\,(e^b - X)\,e^{\beta\,\tau} & \text{(a better choice)} \end{cases}. \tag{BC}$$

Note: If the explicit method is used, N and M need to be chosen such that

$$\frac{1}{2}\,\sigma^2\,\frac{\Delta\tau}{(\Delta x)^2} \le \frac{1}{2} \qquad \Longleftrightarrow \qquad M \ge \frac{\sigma^2\,(T-t)}{(b-a)^2}\,N^2\,.$$

If the implicit or Crank–Nicolson scheme is used, there are no restrictions on N, M. Use Crout or SOR to solve.

4. Assume $w(x_j,\tau_M)$, $j = 0,1,\ldots,N$ are the solutions from step 3, then the call option price at time t is

$$V(S_j,t) = e^{-\alpha\,x_j-\beta\,(T-t)}\,w(x_j,\underbrace{T-t}_{\tau_M}) \qquad j = 0,1,\ldots,N\,,$$

where $S_j = e^{x_j}$ and $T - t \equiv \tau_M$.

Note: The S_j are not equally spaced.

10. **Direct Discretization of the BSE.**

Exercise: Apply the Crank–Nicolson scheme directly to the BSE (**??**), i.e. there is no transformation of variables, and write out the resulting difference equations and do a stability analysis.

C++ Exercise: Write a program to solve the BSE (**??**) using the result of the previous exercise and the Crout algorithm. The inputs are the interest rate r, the volatility σ, the current time t, the expiry time T, the strike price X, the maximum price S_{max}, the number of intervals N in $[0, S_{max}]$, and the number of subintervals M in $[t, T]$. The output are the asset prices S_j, $j = 0, 1, \ldots, N$, at time t, and their corresponding European call and put prices (with the same strike price X).

11. **Greeks.**

Assume that the asset prices S_j and option values V_j, $j = 0, 1, \ldots, N$, are known at time t.

The sensitivities of V at S_j, $j = 1, \ldots, N-1$, are computed as follows:

$$\delta_j = \frac{\partial V}{\partial S}|_{S=S_j} \approx \frac{V_{j+1} - V_{j-1}}{S_{j+1} - S_{j-1}},$$

which is $\dfrac{V_{j+1} - V_{j-1}}{2\,\Delta S}$, if S is equally spaced.

$$\gamma_j = \frac{\partial^2 V}{\partial S^2}|_{S=S_j} \approx \frac{\frac{V_{j+1}-V_j}{S_{j+1}-S_j} - \frac{V_j - V_{j-1}}{S_j - S_{j-1}}}{S_j - S_{j-1}},$$

which is $\dfrac{V_{j+1} - 2\,V_j + V_{j-1}}{(\Delta S)^2}$, if S is equally spaced.

12. **Diffusion Equations of Two State Variables.**

$$\frac{\partial u}{\partial t} = \alpha^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \quad (x, y, t) \in [a, b] \times [c, d] \times [0, \infty). \tag{10}$$

The initial conditions are

$$u(x, y, 0) = u_0(x, y) \qquad \forall\ (x, y) \in [a, b] \times [c, d]\,,$$

and the boundary conditions are

$$u(a, y, t) = g_a(y, t), \quad u(b, y, t) = g_b(y, t) \qquad \forall\ y \in [c, d],\ t \geq 0\,,$$

and

$$u(x, c, t) = g_c(x, t), \quad u(x, d, t) = g_d(x, t) \qquad \forall\ x \in [a, b],\ t \geq 0\,.$$

Here we assume that all the functions involved are consistent, in the sense that they have the same value at common points, e.g. $g_a(c, t) = g_c(a, t)$ for all $t \geq 0$.

13. **Grid Points.**

(x_i, y_j, t_n), where

$$\begin{aligned} x_i &= a + i\,\Delta x, \quad \Delta x = \frac{b-a}{I}, \quad i = 0, \ldots, I\,, \\ y_j &= c + j\,\Delta y, \quad \Delta y = \frac{d-c}{J}, \quad j = 0, \ldots, J\,, \\ t_n &= n\,\Delta t, \quad \Delta t = \frac{T}{N}, \quad n = 0, \ldots, N \end{aligned}$$

and I, J, N are integers.

Recalling the finite differences (5), we have

$$u_{xx} \approx \frac{u^n_{i+1,j} - 2\,u^n_{i,j} + u^n_{i-1,j}}{(\Delta x)^2} \quad \text{and} \quad u_{yy} \approx \frac{u^n_{i,j+1} - 2\,u^n_{i,j} + u^n_{i,j-1}}{(\Delta y)^2}\,.$$

at a grid point (i, j, n).

Depending on how u_t is approximated, we have three basic schemes: explicit, implicit, and Crank–Nicolson schemes.

14. **Explicit Scheme.**

If u_t is approximated by a forward difference quotient

$$u_t \approx \frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t}$$

at (i, j, n),

then the corresponding difference equation at grid point (i, j, n) is

$$w_{i,j}^{n+1} = (1 - 2\,\lambda - 2\,\mu)\, w_{i,j}^{n} + \lambda\, w_{i+1,j}^{n} + \lambda\, w_{i-1,j}^{n} + \mu\, w_{i,j+1}^{n} + \mu\, w_{i,j-1}^{n} \quad (11)$$

for $i = 1, \ldots, I-1$ and $j = 1, \ldots, J-1$, where

$$\lambda = \alpha^2\, \frac{\Delta t}{(\Delta x)^2} \qquad \text{and} \qquad \mu = \alpha^2\, \frac{\Delta t}{(\Delta y)^2}\,.$$

(11) can be solved explicitly. It has local truncation error $O(\Delta t, (\Delta x)^2, (\Delta y)^2)$, but is only conditionally stable.

15. **Implicit Scheme.**

If u_t is approximated by a backward difference quotient $u_t \approx \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t}$ at $(i, j, n+1)$, then the difference equation at grid point $(i, j, n+1)$ is

$$(1 + 2\,\lambda + 2\,\mu)\, w_{i,j}^{n+1} - \lambda\, w_{i+1,j}^{n+1} - \lambda\, w_{i-1,j}^{n+1} - \mu\, w_{i,j+1}^{n+1} - \mu\, w_{i,j-1}^{n+1} = w_{i,j}^n \quad (12)$$

for $i = 1, \ldots, I-1$ and $j = 1, \ldots, J-1$.

For fixed n, there are $(I-1)\,(J-1)$ unknowns and equations. (12) can be solved by relabeling the grid points and using the SOR algorithm.

(12) is unconditionally stable with local truncation error $O(\Delta t, (\Delta x)^2, (\Delta y)^2)$, but is more difficult to solve, as it is no longer tridiagonal, so the Crout algorithm cannot be applied.

16. **Crank–Nicolson Scheme.**

It is the average of the explicit scheme at (i, j, n) and the implicit scheme at $(i, j, n+1)$. It is similar to the implicit scheme but with the improved local truncation error $O((\Delta t)^2, (\Delta x)^2, (\Delta y)^2)$.

Solving the Implicit Scheme.

$$(1 + 2\,\lambda + 2\,\mu)\,w_{i,j}^{n+1} - \lambda\,w_{i+1,j}^{n+1} - \lambda\,w_{i-1,j}^{n+1} - \mu\,w_{i,j+1}^{n+1} - \mu\,w_{i,j-1}^{n+1} = w_{i,j}^{n}$$

With SOR for $\omega \in (0, 2)$.

For each $n = 0, 1, \ldots, N$

1. Set $w^{n+1,0} := w^n$ and

   fill in the boundary conditions $w_{0,j}^{n+1,0}$, $w_{I,j}^{n+1,0}$, $w_{i,0}^{n+1,0}$, $w_{i,J}^{n+1,0}$ for all i, j.

2. For $k = 0, 1, \ldots$

   For $i = 1, \ldots I-1$, $j = 1, \ldots, J-1$

   $$\widehat{w}_{i,j}^{k+1} = \tfrac{1}{1+2\,\lambda+2\,\mu}\left(w_{i,j}^{n} + \lambda\,w_{i+1,j}^{n+1,k} + \lambda\,w_{i-1,j}^{n+1,k+1} + \mu\,w_{i,j+1}^{n+1,k} + \mu\,w_{i,j-1}^{n+1,k+1}\right)$$
   $$w_{i,j}^{n+1,k+1} = (1-\omega)\,w_{i,j}^{n+1,k} + \omega\,\widehat{w}_{i,j}^{k+1}$$

   until $\|w^{n+1,k+1} - w^{n+1,k}\| < \varepsilon$.

3. Set $w^{n+1} = w^{n+1,k+1}$.

With Block Jacobi/Gauss–Seidel.

Denote $\widetilde{w}_j = \begin{pmatrix} w_{1,j} \\ w_{2,j} \\ \vdots \\ w_{I-1,j} \end{pmatrix} \in \mathbb{R}^{I-1},\ j = 1, \ldots, J-1, \quad w = \begin{pmatrix} \widetilde{w}_1 \\ \widetilde{w}_2 \\ \vdots \\ \widetilde{w}_{J-1} \end{pmatrix} \in \mathbb{R}^{(I-1)\,(J-1)}.$

On setting $c = 1 + 2\,\lambda + 2\,\mu$, we have from (12) for j fixed

$$\begin{pmatrix} c & -\lambda & & \\ -\lambda & c & -\lambda & \\ & & \ddots & \\ & & -\lambda & c \end{pmatrix} \begin{pmatrix} w_{1,j} \\ w_{2,j} \\ \vdots \\ w_{I-1,j} \end{pmatrix} + \begin{pmatrix} -\mu & & \\ & \ddots & \\ & & -\mu \end{pmatrix} \begin{pmatrix} w_{1,j+1} \\ w_{2,j+1} \\ \vdots \\ w_{I-1,j+1} \end{pmatrix} + \begin{pmatrix} -\mu & & \\ & \ddots & \\ & & -\mu \end{pmatrix} \begin{pmatrix} w_{1,j-1} \\ w_{2,j-1} \\ \vdots \\ w_{I-1,j-1} \end{pmatrix}$$

$$= \begin{pmatrix} w_{1,j}^n + \lambda\, w_{0,j}^{n+1} \\ w_{2,j}^n \\ \vdots \\ w_{I-2,j}^n \\ w_{I-1,j}^n + \lambda\, w_{I,j}^{n+1} \end{pmatrix} \qquad \Longleftrightarrow \qquad A\,\widetilde{w}_j^{n+1} + B\,\widetilde{w}_{j+1}^{n+1} + B\,\widetilde{w}_{j-1}^{n+1} = d_j^n$$

Rewriting

$$A\,\widetilde{w}_j^{n+1} + B\,\widetilde{w}_{j+1}^{n+1} + B\,\widetilde{w}_{j-1}^{n+1} = d_j^n \qquad j = 1, \ldots, J-1$$

as

$$\begin{pmatrix} A & B & & & \\ B & A & B & & \\ & \ddots & \ddots & \ddots & \\ & & B & A & B \\ & & & B & A \end{pmatrix} \begin{pmatrix} \widetilde{w}_1^{n+1} \\ \widetilde{w}_2^{n+1} \\ \vdots \\ \vdots \\ \widetilde{w}_{J-1}^{n+1} \end{pmatrix} = \begin{pmatrix} d_1^n - B\,\widetilde{w}_0^{n+1} \\ d_2^n \\ \vdots \\ d_{J-2}^n \\ d_{J-1}^n - B\,\widetilde{w}_J^{n+1} \end{pmatrix} =: \begin{pmatrix} \widetilde{d}_1^n \\ d_2^n \\ \vdots \\ d_{J-2}^n \\ \widetilde{d}_{J-1}^n \end{pmatrix},$$

where $\widetilde{w}_0^{n+1}$ and $\widetilde{w}_J^{n+1}$ represent boundary points, leads to the following Block Jacobi iteration: For $k = 0, 1, \ldots$

$$\begin{aligned} A\,\widetilde{w}_1^{n+1,k+1} &= -B\,\widetilde{w}_2^{n+1,k} + \widetilde{d}_1^n \\ A\,\widetilde{w}_2^{n+1,k+1} &= -B\,\widetilde{w}_1^{n+1,k} - B\,\widetilde{w}_3^{n+1,k} + d_2^n \\ &\vdots \\ A\,\widetilde{w}_{J-2}^{n+1,k+1} &= -B\,\widetilde{w}_{J-3}^{n+1,k} - B\,\widetilde{w}_{J-1}^{n+1,k} + d_{J-2}^n \\ A\,\widetilde{w}_{J-1}^{n+1,k+1} &= -B\,\widetilde{w}_{J-2}^{n+1,k} + \widetilde{d}_{J-1}^n \end{aligned}$$

Similarly, the Block Gauss–Seidel iteration is given by:
For $k = 0, 1, \ldots$

$$\begin{aligned}
A\,\widetilde{w}_1^{n+1,k+1} &= -B\,\widetilde{w}_2^{n+1,k} + \widetilde{d}_1^n \\
A\,\widetilde{w}_2^{n+1,k+1} &= -B\,\widetilde{w}_1^{n+1,k+1} - B\,\widetilde{w}_3^{n+1,k} + d_2^n \\
&\vdots \\
A\,\widetilde{w}_{J-2}^{n+1,k+1} &= -B\,\widetilde{w}_{J-3}^{n+1,k+1} - B\,\widetilde{w}_{J-1}^{n+1,k} + d_{J-2}^n \\
A\,\widetilde{w}_{J-1}^{n+1,k+1} &= -B\,\widetilde{w}_{J-2}^{n+1,k} + \widetilde{d}_{J-1}^n
\end{aligned}$$

In each case, use the Crout algorithm to solve for $\widetilde{w}_j^{n+1,k+1}$, $j = 1, \ldots, J-1$.

Note on Stability.
Recall that in 1d a scheme was stable if $|G(\omega)| = \left|\frac{a^{(n+1)}(\omega)}{a^{(n)}(\omega)}\right| \leq 1$, where $v_j^n = a^{(n)}(\omega)\, e^{i\,j\,\omega\,\Delta x}$.
In 2d, this is adapted to

$$v_{i,j}^n = a^{(n)}(\omega)\, e^{\sqrt{-1}\,i\,\omega\,\Delta x + \sqrt{-1}\,j\,\omega\,\Delta y}\,.$$

17. **Alternating Direction Implicit (ADI) Method.**

An alternative finite difference method is the ADI scheme, which is unconditionally stable while the difference equations are still tridiagonal and diagonally dominant.

The ADI algorithm can be used to efficiently solve the Black–Scholes two asset pricing equation:

$$V_t + \tfrac{1}{2}\,\sigma_1^2\,S_1^2\,V_{S_1S_1} + \tfrac{1}{2}\,\sigma_2^2\,S_2^2\,V_{S_2S_2} + \rho\,\sigma_1\,\sigma_2\,S_1\,S_2 V_{S_1S_2} + r\,S_1 V_{S_1} + r\,S_2 V_{S_2} - r\,V = 0. \tag{13}$$

See Clewlow and Strickland (1998) for details on how to transform the Black–Scholes equation (13) into the basic diffusion equation (10) and then to solve it with the ADI scheme.

ADI scheme

Implicit method at $(i, j, n+1)$:

$$\frac{w_{i,j}^{n+1} - w_{i,j}^{n}}{\Delta t} = \underbrace{\alpha^2 \, \frac{w_{i+1,j}^{n} - 2\, w_{i,j}^{n} + w_{i-1,j}^{n}}{(\Delta x)^2}}_{\text{approx. } u_{xx} \text{ using } (i,j,n) \text{ data}} + \alpha^2 \, \frac{w_{i,j+1}^{n+1} - 2\, w_{i,j}^{n+1} + w_{i,j-1}^{n+1}}{(\Delta y)^2}$$

Implicit method at $(i, j, n+2)$:

$$\frac{w_{i,j}^{n+2} - w_{i,j}^{n+1}}{\Delta t} = \alpha^2 \, \frac{w_{i+1,j}^{n+2} - 2\, w_{i,j}^{n+2} + w_{i-1,j}^{n+2}}{(\Delta x)^2} + \alpha^2 \underbrace{\frac{w_{i,j+1}^{n+1} - 2\, w_{i,j}^{n+1} + w_{i,j-1}^{n+1}}{(\Delta y)^2}}_{\text{approx. } u_{yy} \text{ using } (i,j,n+1) \text{ data}}$$

We can write the two equations as follows:

$$-\mu \, w_{i,j+1}^{n+1} + (1 + 2\,\mu) \, w_{i,j}^{n+1} - \mu \, w_{i,j-1}^{n+1} = \lambda \, w_{i+1,j}^{n} + (1 - 2\,\lambda) \, w_{i,j}^{n} + \lambda \, w_{i-1,j}^{n} \qquad (\dagger)$$

$$-\lambda \, w_{i+1,j}^{n+2} + (1 + 2\,\lambda) \, w_{i,j}^{n+2} - \lambda \, w_{i-1,j}^{n+2} = \mu \, w_{i,j+1}^{n+1} + (1 - 2\,\mu) \, w_{i,j}^{n+1} + \mu \, w_{i,j-1}^{n+1} \qquad (\ddagger)$$

To solve (†), fix $i = 1, \ldots, I-1$ and solve a tridiagonal system to get $w_{i,j}^{n+1}$ for $j = 1, \ldots, J-1$.
This can be done with e.g. the Crout algorithm.
To solve (‡), fix $j = 1, \ldots, J-1$ and solve a tridiagonal system to get $w_{i,j}^{n+2}$ for $i = 1, \ldots, I-1$.

Currently the method works on the interval $[t_n, t_{n+2}]$ and has features of an explicit method. In order to obtain an (unconditionally stable) implicit method, we need to adapt the method so that it works on the interval $[t_n, t_{n+1}]$ and hence gives values $w_{i,j}^n$ for all $n = 1, \ldots, N$.
Introduce the intermediate time point $n + \frac{1}{2}$. Then (†) generates $w_{i,j}^{n+\frac{1}{2}}$ (not used) and (‡) generates $w_{i,j}^{n+1}$.

$$-\frac{\mu}{2}\, w_{i,j+1}^{n+\frac{1}{2}} + (1+\mu)\, w_{i,j}^{n+\frac{1}{2}} - \frac{\mu}{2}\, w_{i,j-1}^{n+\frac{1}{2}} = \frac{\lambda}{2}\, w_{i+1,j}^{n} + (1-\lambda)\, w_{i,j}^{n} + \frac{\lambda}{2}\, w_{i-1,j}^{n} \qquad (\dagger)$$

$$-\frac{\lambda}{2}\, w_{i+1,j}^{n+1} + (1+\lambda)\, w_{i,j}^{n+1} - \frac{\lambda}{2}\, w_{i-1,j}^{n+1} = \frac{\mu}{2}\, w_{i,j+1}^{n+\frac{1}{2}} + (1-\mu)\, w_{i,j}^{n+\frac{1}{2}} + \frac{\mu}{2}\, w_{i,j-1}^{n+\frac{1}{2}} \qquad (\ddagger)$$