Lecture 7

Numerical methods for PDEs

Lecture Notes
by Jan Palczewski

Numerical solution of heat equation

$$\frac{\partial y}{\partial \tau}(x,\tau) - \frac{\partial^2 y}{\partial x^2}(x,\tau) = 0$$

- $x \in (-\infty, \infty)$, $\quad \tau \in [0, \frac{1}{2}\sigma^2 T]$
- discretization of time τ: $\delta\tau = \frac{\frac{1}{2}\sigma^2 T}{N}$

$$\tau_\nu := \nu \cdot \delta\tau \quad \text{for} \quad \nu = 0, 1, \ldots, N.$$

- discretization of space x
  - x_{min}, x_{max}
  - $\delta x = \dfrac{x_{max} - x_{min}}{M}$

$$x_i := x_{min} + i \cdot \delta x \quad \text{for} \quad i = 0, 1, \ldots, M.$$

Derivatives – explicit method

We approximate partial derivatives of y by

$$\frac{\partial y}{\partial \tau}(x_i, \tau_\nu) \approx \frac{y(x_i, \tau_{\nu+1}) - y(x_i, \tau_\nu)}{\delta \tau}$$

$$\frac{\partial^2 y}{\partial x^2}(x_i, \tau_\nu) \approx \frac{y(x_{i+1}, \tau_\nu) - 2y(x_i, \tau_\nu) + y(x_{i-1}, \tau_\nu)}{(\delta x)^2}$$

Let $w_{i,\nu}$ be an approximation of $y(x_i, \tau_\nu)$. Then

$$\boxed{\frac{w_{i,\nu+1} - w_{i,\nu}}{\delta \tau} - \frac{w_{i+1,\nu} - 2w_{i,\nu} + w_{i-1,\nu}}{(\delta x)^2} = 0}$$

is an approximation to

$$\frac{\partial y}{\partial \tau}(x, \tau) - \frac{\partial^2 y}{\partial x^2}(x, \tau) = 0.$$

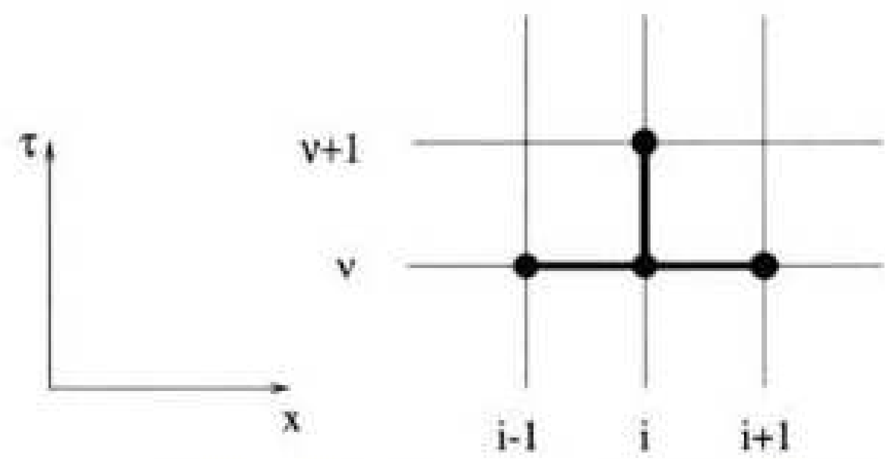
Explicit method

$$\boxed{w_{i,\nu+1} = \lambda w_{i+1,\nu} + (1-2\lambda) w_{i,\nu} + \lambda w_{i-1,\nu},}$$

where

$$\lambda = \frac{\delta\tau}{(\delta x)^2}.$$

Method's stencil



European put option

$$
\begin{cases}
\dfrac{\partial y}{\partial \tau}(x,\tau) - \dfrac{\partial^2 y}{\partial x^2}(x,\tau) = 0, \\[2ex]
y(x,0) = e^{-rT}(K - e^x)^+, \quad x \in \mathbb{R}, \\[2ex]
\lim\limits_{x \to -\infty} y(x,\tau) = Ke^{-rT}, \quad \tau \in \left[0, \dfrac{\sigma^2}{2}T\right], \\[2ex]
\lim\limits_{x \to \infty} y(x,\tau) = 0, \quad \tau \in \left[0, \dfrac{\sigma^2}{2}T\right].
\end{cases}
$$

Algorithm for a European put option

Input: x_{min}, x_{max}, M, N, K, T and the parameters of the model

---

$\delta\tau = \frac{\sigma^2 T}{2N}$, $\qquad \delta x = \frac{x_{max} - x_{min}}{M}$

Calculate τ_ν, $\nu = 0, 1, \ldots, N$, and x_i, $i = 0, 1, \ldots, M$

For $i = 0, 1, \ldots, M$

$\quad w_{i,0} = e^{-rT}(K - e^{x_i})^+$

For $\nu = 0, 1, \ldots, N-1$

$\quad w_{0,\nu+1} = Ke^{-rT}$

$\quad w_{M,\nu+1} = 0$

$\quad$ For $i = 1, 2, \ldots, M-1$

$\qquad w_{i,\nu+1} = \lambda w_{i+1,\nu} + (1 - 2\lambda) w_{i,\nu} + \lambda w_{i-1,\nu}$

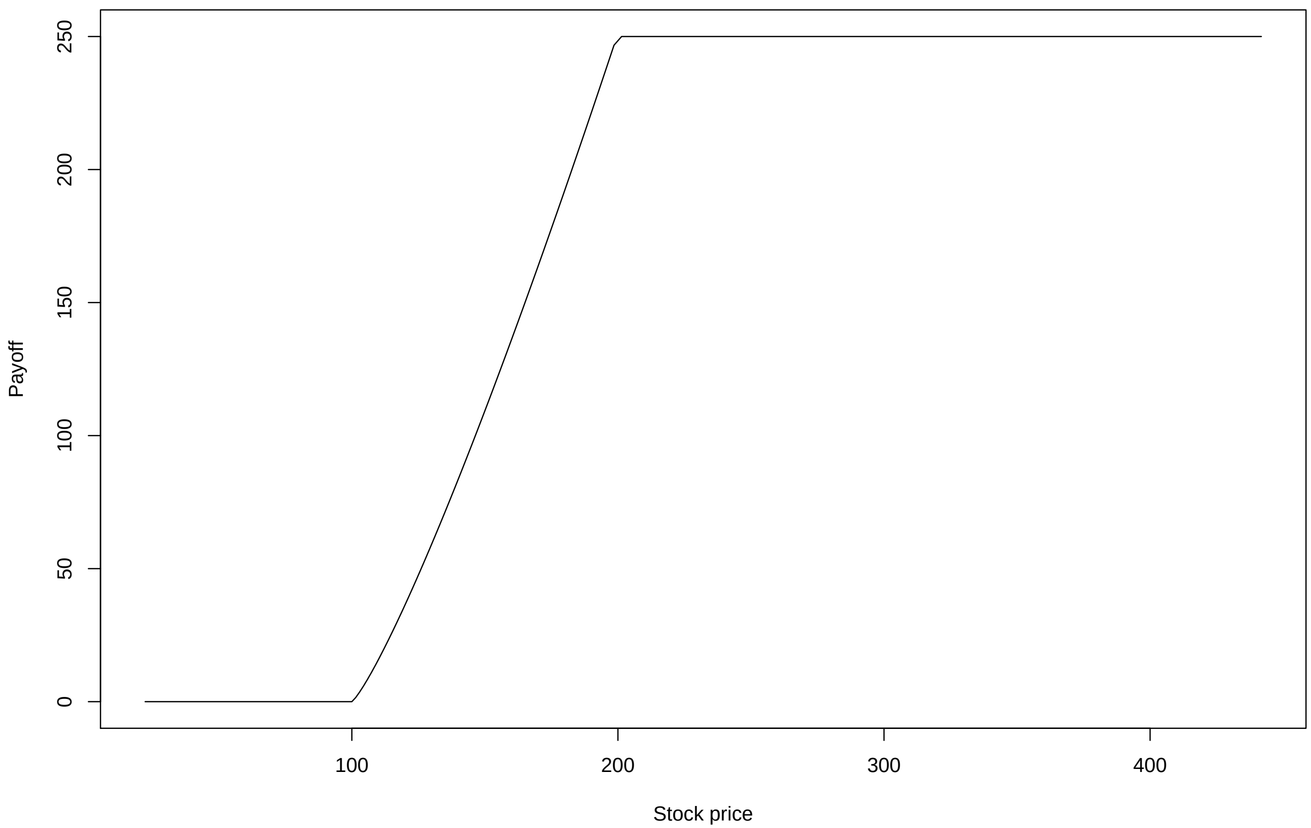
Output: $w_{i,\nu}$ for $i = 0, 1, \ldots, M$, $\quad \nu = 0, 1, \ldots, N$

Bounded power option

Consider an option with payoff $h(S_T)$, where

$$h(s) = \min(L, \left((s-K)^+\right)^p)$$

for some number L. The payoff graph below is for $p = 1.2$, $L = 250$, $K = 100$.



The corresponding Black-Scholes PDE has the form

$$\begin{cases} \dfrac{1}{2}\sigma^2 s^2 \dfrac{\partial^2 V(s,t)}{\partial s^2} + rs\dfrac{\partial V(s,t)}{\partial s} - rV(s,t) + \dfrac{\partial V(s,t)}{\partial t} = 0, \\ V(s,T) = \min(L, \big((s-K)^+\big)^p), \qquad s > 0, \\ \displaystyle\lim_{s\to 0} V(s,t) = 0, \qquad t \in [0,T], \\ \displaystyle\lim_{s\to\infty} V(s,t) = Le^{-r(T-t)}, \qquad t \in [0,T]. \end{cases}$$

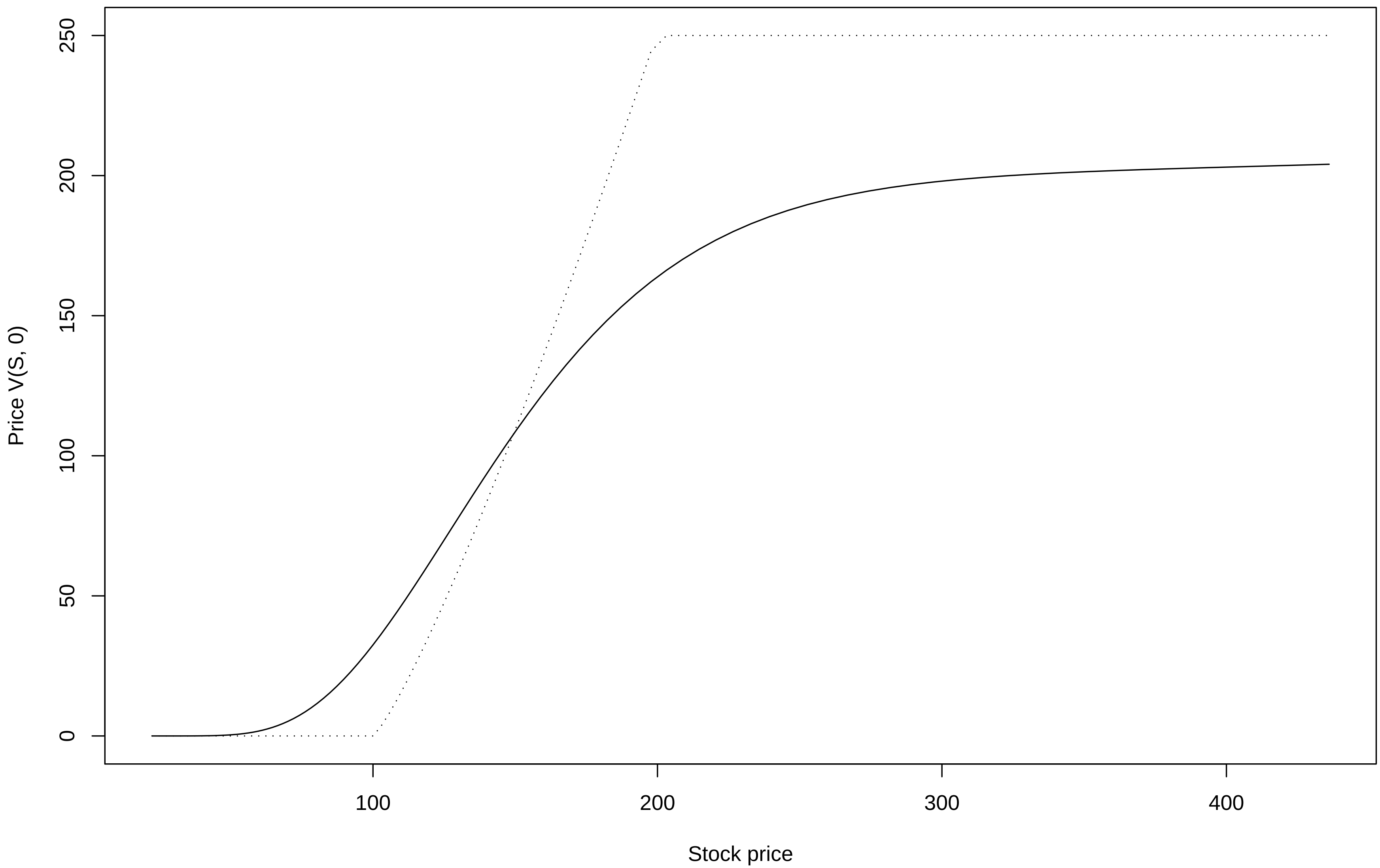
Transformed PDE

$$\begin{cases} \dfrac{\partial y}{\partial \tau}(x,\tau) - \dfrac{\partial^2 y}{\partial x^2}(x,\tau) = 0, \\ \\ y(x,0) = e^{-rT}\min\Big(L, \big((e^x - K)^+\big)^p\Big), \quad x \in \mathbb{R}, \\ \\ \displaystyle\lim_{x\to-\infty} y(x,\tau) = 0, \quad \tau \in \Big[0, \frac{\sigma^2}{2}T\Big], \\ \\ \displaystyle\lim_{x\to\infty} y(x,\tau) = Le^{-rT}, \quad \tau \in \Big[0, \frac{\sigma^2}{2}T\Big]. \end{cases}$$

For the following computations we take $p = 1.2$, $K = 100$, $L = 250$, $S_0 = 100$, $\sigma = 0.2$, $r = 0.05$, $T = 5$.

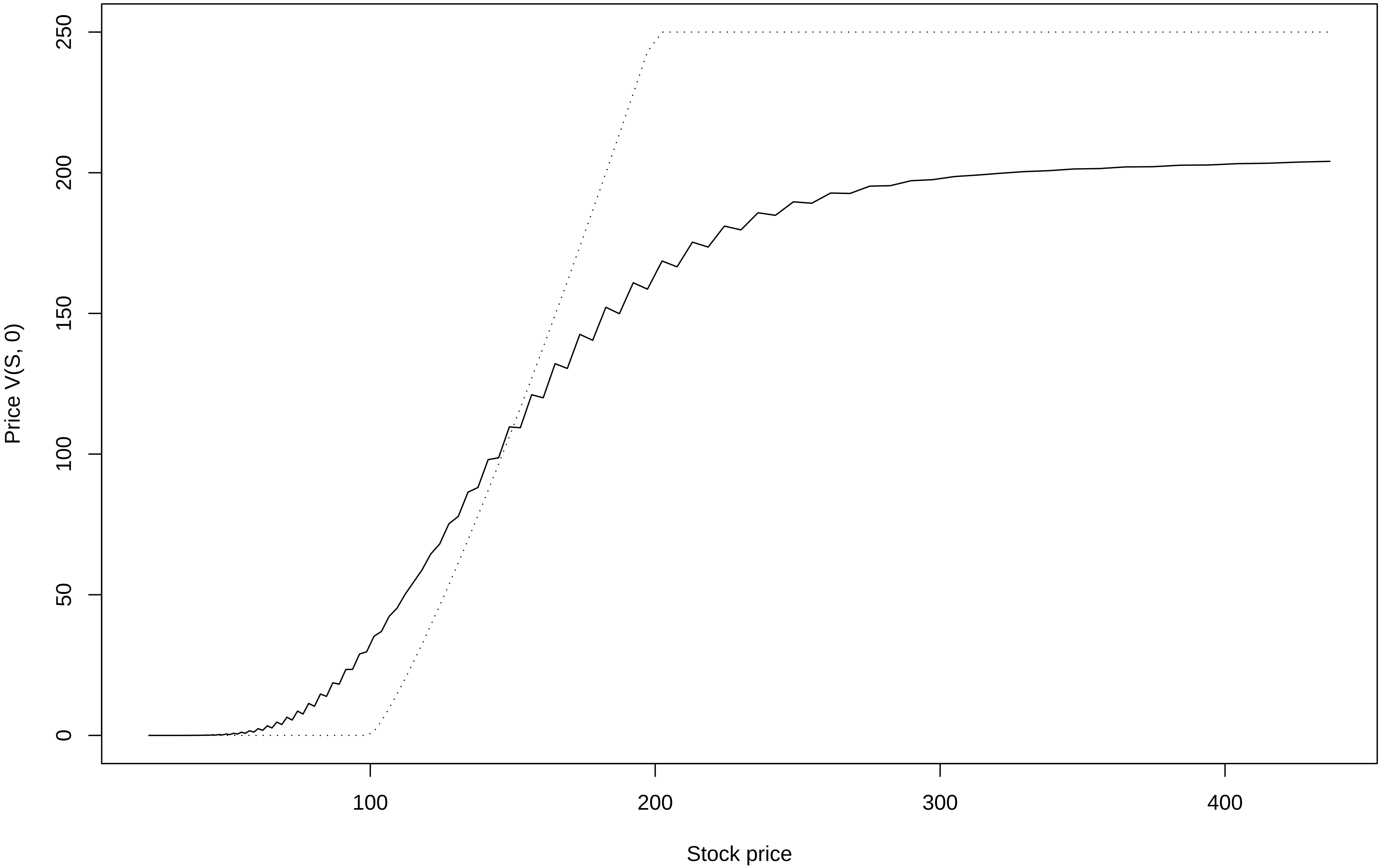
$M = 110,\qquad N = 150$

(dotted line – the payoff)



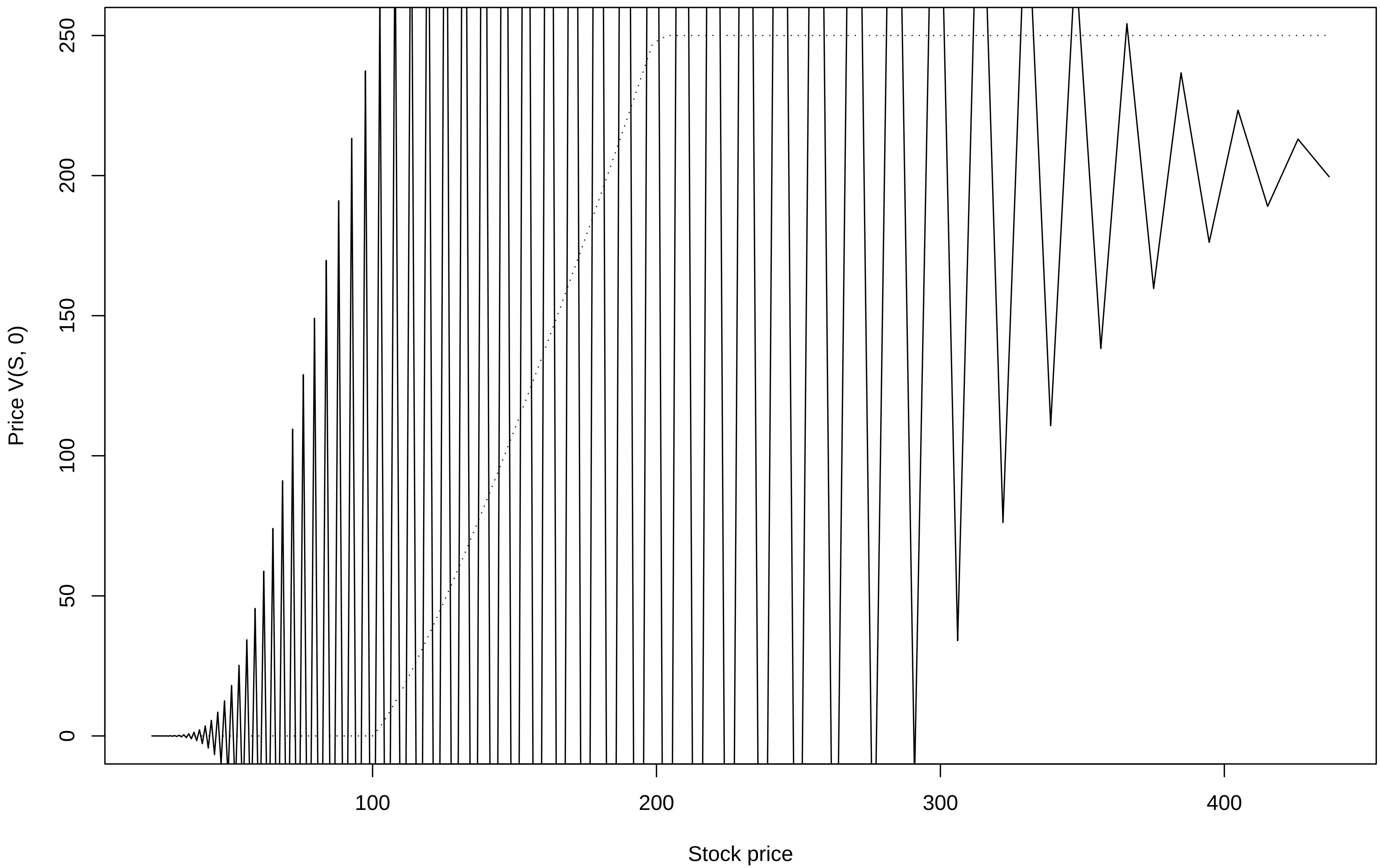
$M = 117, \qquad N = 150$

(dotted line – the payoff)



$M = 118,\qquad N = 150$

(dotted line – the payoff)



Stability

Strange results from the previous slides are caused by the **instability** of the algorithm, i.e. computational errors are amplified in each step. However, we can avoid this kind of problems.

For $\lambda = \dfrac{\delta\tau}{(\delta x)^2} \leq \dfrac{1}{2}$ the explicit method is stable.

Important remark

For the equation

$$\frac{\partial y}{\partial \tau}(x,\tau) - a^2 \frac{\partial^2 y}{\partial x^2}(x,\tau) = 0,$$

the stability condition reads

$$\lambda = \frac{a^2 \delta\tau}{(\delta x)^2} \leq \frac{1}{2}.$$

Are there methods that are unconditionally stable?

Implicit method

To derive the **explicit scheme** we approximated the derivative with respect to time with

$$\frac{\partial y}{\partial \tau}(x_i,\tau_\nu) \approx \frac{y(x_i,\tau_{\nu+1}) - y(x_i,\tau_\nu)}{\delta\tau}.$$

Now, we approximate partial derivatives of y by

$$\frac{\partial y}{\partial \tau}(x_i,\tau_\nu) \approx \frac{y(x_i,\tau_\nu) - y(x_i,\tau_{\nu-1})}{\delta\tau}$$

$$\frac{\partial^2 y}{\partial x^2}(x_i,\tau_\nu) \approx \frac{y(x_{i+1},\tau_\nu) - 2y(x_i,\tau_\nu) + y(x_{i-1},\tau_\nu)}{(\delta x)^2}$$

Let $w_{i,\nu}$ be an approximation of $y(x_i,\tau_\nu)$. Then the heat equation is approximated by

$$\boxed{\frac{w_{i,\nu} - w_{i,\nu-1}}{\delta\tau} - \frac{w_{i+1,\nu} - 2w_{i,\nu} + w_{i-1,\nu}}{(\delta x)^2} = 0.}$$

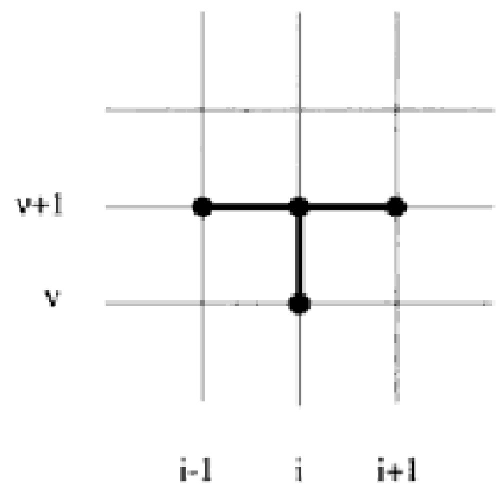
Implicit method

$$\boxed{-\lambda w_{i+1,\nu+1} + (1+2\lambda) w_{i,\nu+1} - \lambda w_{i-1,\nu+1} = w_{i,\nu},}$$

where

$$\lambda = \frac{\delta\tau}{(\delta x)^2}.$$

Method's stencil



This method is much more difficult to implement. We have to solve equations for all i together as a **system of linear equations**. Let

$$w^{(\nu)} = (w_{1,\nu}, w_{2,\nu}, \ldots, w_{M-1,\nu}).$$

We skip $w_{0,\nu}$ and $w_{M,\nu}$ since they are known from the boundary conditions.

Each step of the implicit scheme requires the solution of

$$Aw^{(\nu+1)} = w^{(\nu)} + d^{(\nu)},$$

where

$$A = \begin{pmatrix} 2\lambda+1 & -\lambda & & 0 \\ -\lambda & \ddots & -\lambda & \\ & \ddots & \ddots & \ddots \\ 0 & & \ddots & \ddots \end{pmatrix}, \qquad d^{(\nu)} = \begin{pmatrix} \lambda w_{0,\nu+1} \\ 0 \\ \vdots \\ 0 \\ \lambda w_{M,\nu+1} \end{pmatrix}.$$

$d^{(\nu)} \in \mathbb{R}^{M-1}$, so there are $M-3$ zeros in $d^{(\nu)}$. The matrix A is square with $M-1$ rows and $M-1$ columns.

Algorithm

Input: x_{min}, x_{max}, M, N, T, f and the parameters of the model

---

$\delta\tau = \frac{\sigma^2 T}{2N}$, $\qquad \delta x = \frac{x_{max} - x_{min}}{M}$

Calculate τ_ν, $\nu = 0, 1, \ldots, N$, and x_i, $i = 0, 1, \ldots, M$

For $i = 0, 1, \ldots, M$

  $w_{i,0} = e^{-rT} f(e^{x_i})$

For $\nu = 0, 1, \ldots, N-1$

  compute the boundary values $w_{0,\nu+1}$ and $w_{M,\nu+1}$

  compute the vector $d^{(\nu)}$

  solve the equation

$$Aw^{(\nu+1)} = w^{(\nu)} + d^{(\nu)}$$

Output: $w_{i,\nu}$ for $i = 0, 1, \ldots, M$, $\quad \nu = 0, 1, \ldots, N$

How to solve linear equations?

- Gaussian elimination
  - exact result,
  - time complexity n^3, where n is the size of a matrix (i.e. the number of rows or the number of columns),
- iterative methods (modifications of Newton's method)
  - approximate,
  - time complexity of one step n^2,
  - how many steps are needed?

Our matrix A is **very special**. We will use an algorithm that

- gives exact results,
- has time complexity n, where n is the size of a matrix.

$$
\begin{pmatrix}
\alpha_1 & \beta_1 & & & 0 \\
\gamma_2 & \alpha_2 & \beta_2 & & \\
 & \ddots & \ddots & \ddots & \\
 & & \gamma_{n-1} & \alpha_{n-1} & \beta_{n-1} \\
0 & & & \gamma_n & \alpha_n
\end{pmatrix}
\begin{pmatrix}
x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n
\end{pmatrix}
=
\begin{pmatrix}
d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n
\end{pmatrix}
$$

(Forward loop)

$\hat{\alpha}_1 = \alpha_1, \quad \hat{d}_1 = d_1.$

For $i = 2, 3, \ldots, n$

$$\hat{\alpha}_i = \alpha_i - \beta_{i-1}\frac{\gamma_i}{\hat{\alpha}_{i-1}}, \qquad \hat{d}_i = d_i - \hat{d}_{i-1}\frac{\gamma_i}{\hat{\alpha}_{i-1}}.$$

(Backward loop)

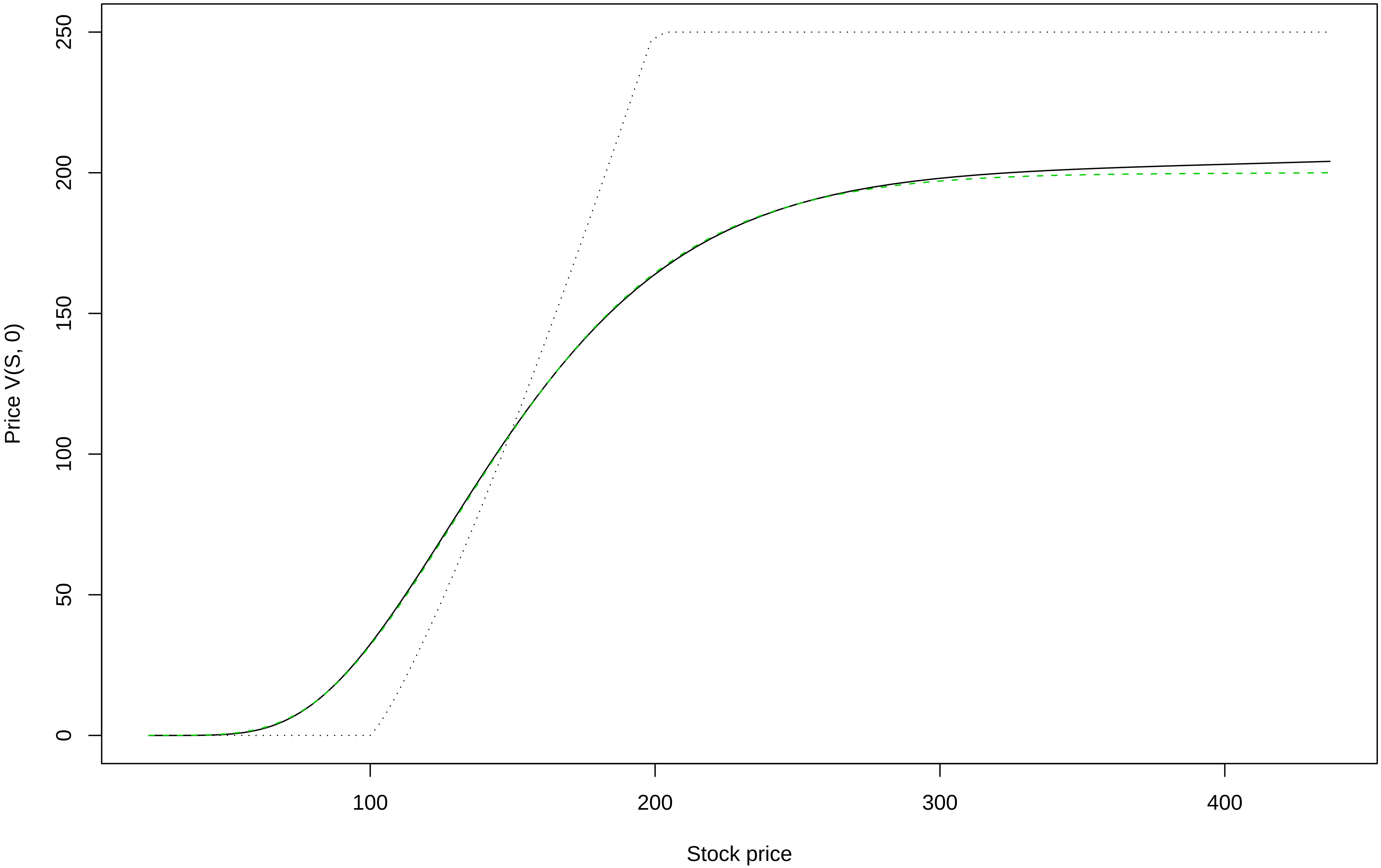
$x_n = \dfrac{\hat{d}_n}{\hat{\alpha}_n}.$

For $i = n-1, n-2, \ldots, 1$

$$x_i = \frac{1}{\hat{\alpha}_i}(\hat{d}_i - \beta_i x_{i+1}).$$

Power option

$M = 117$, $N = 150$ (solid line), $N = 10$ (dashed line)
(dotted line – the payoff)



What if we combine the explicit scheme with the implicit scheme?
Explicit method (for ν)

$$\frac{w_{i,\nu+1} - w_{i,\nu}}{\delta\tau} - \frac{w_{i+1,\nu} - 2w_{i,\nu} + w_{i-1,\nu}}{(\delta x)^2} = 0$$

Implicit method (for $\nu + 1$)

$$\frac{w_{i,\nu+1} - w_{i,\nu}}{\delta\tau} - \frac{w_{i+1,\nu+1} - 2w_{i,\nu+1} + w_{i-1,\nu+1}}{(\delta x)^2} = 0.$$

If we add them, we get the following approximation

$$2\frac{w_{i,\nu+1} - w_{i,\nu}}{\delta\tau} - \frac{w_{i+1,\nu+1} - 2w_{i,\nu+1} + w_{i-1,\nu+1} + w_{i+1,\nu} - 2w_{i,\nu} + w_{i-1,\nu}}{(\delta x)^2} = 0.$$

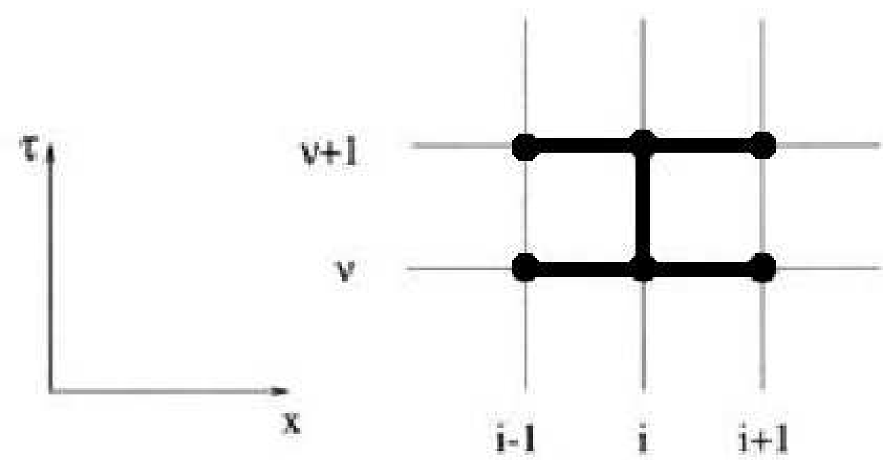
As always $w_{i,\nu}$ is an approximation of $y(x_i, \tau_\nu)$.

Crank-Nicolson method

$$-\frac{\lambda}{2}w_{i-1,\nu+1} + (1+\lambda)w_{i,\nu+1} - \frac{\lambda}{2}w_{i+1,\nu+1} = \frac{\lambda}{2}w_{i-1,\nu} + (1-\lambda)w_{i,\nu} + \frac{\lambda}{2}w_{i+1,\nu}$$

$$\lambda = \frac{\delta\tau}{(\delta x)^2}$$

Method's stencil



Crank-Nicolson method can be implemented in the same way as the implicit method. Namely, at each step we need to solve a system of linear equations

$$Aw^{(\nu+1)} = Bw^{(\nu)} + d^{(\nu)},$$

where

$$A = \begin{pmatrix} \lambda+1 & -\frac{\lambda}{2} & & 0 \\ -\frac{\lambda}{2} & \ddots & -\frac{\lambda}{2} & \\ & \ddots & \ddots & \ddots \\ 0 & & \ddots & \ddots \end{pmatrix}, \qquad d^{(\nu)} = \frac{\lambda}{2}\begin{pmatrix} w_{0,\nu} + w_{0,\nu+1} \\ 0 \\ \vdots \\ 0 \\ w_{M,\nu} + w_{M,\nu+1} \end{pmatrix},$$

$$B = \begin{pmatrix} 1-\lambda & \frac{\lambda}{2} & & 0 \\ \frac{\lambda}{2} & \ddots & \frac{\lambda}{2} & \\ & \ddots & \ddots & \ddots \\ 0 & & \ddots & \ddots \end{pmatrix}.$$

Comparison

- Explicit method
  - approximation order $O(\delta\tau) + O((\delta x)^2)$
  - easy to implement
  - stability problem
- Implicit method
  - approximation order $O(\delta\tau) + O((\delta x)^2)$
  - quite tricky to implement
  - unconditionally stable
- Crank-Nicolson method
  - approximation order $O((\delta\tau)^2) + O((\delta x)^2)$
  - quite tricky to implement
  - unconditionally stable

Straitforward solution

We can solve the Black-Scholes PDE without transformation to the heat equation. We shall present this approach on the example of the European call option.
The equation reads

$$\frac{\partial V(s,t)}{\partial t} + rs\frac{\partial V(s,t)}{\partial s} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 V(s,t)}{\partial s^2} - rV(s,t) = 0.$$

The terminal and boundary conditions are as follows:

$$V(s,T) = h(s),$$
$$V(0,t) = 0,$$
$$V(s,t) \approx s \quad \text{as} \quad s \to \infty.$$

Discretization

- $s \in (0, \infty)$, $\quad t \in [0, T]$
- discretization of time t: $\delta t = \frac{T}{N}$

$$t_\nu := \nu \cdot \delta t \quad \text{for} \quad \nu = 0, 1, \ldots, N.$$

- discretization of space s
  - $s_{min} = 0$, s_{max} – sufficiently large,
  - $\delta s = \dfrac{s_{max}}{M}$

$$s_i := i \cdot \delta s \quad \text{for} \quad i = 0, 1, \ldots, M.$$

Derivatives

We approximate partial derivatives of V by

backward differences in t

$$\frac{\partial V}{\partial t}(s_i, t_\nu) \approx \frac{V(s_i, t_\nu) - V(s_i, t_{\nu-1})}{\delta t},$$

central differences in s

$$\frac{\partial V}{\partial s}(s_i, t_\nu) \approx \frac{V(s_{i+1}, t_\nu) - V(s_{i-1}, t_\nu)}{2\delta s},$$

$$\frac{\partial^2 V}{\partial s^2}(s_i, t_\nu) \approx \frac{V(s_{i+1}, t_\nu) - 2V(s_i, t_\nu) + V(s_{i-1}, t_\nu)}{(\delta s)^2}.$$

Finite difference approximation

Let $v_{i,\nu}$ be an approximation of $V(s_i, t_\nu)$. Then

$$\boxed{v_{i,\nu-1} = a_i v_{i-1,\nu} + b_i v_{i,\nu} + c_i v_{i+1,\nu}}$$

is an explicit numerical algorithm for the Black-Scholes equation.
And we have the following expression for the coefficients:

$$a_i = \frac{1}{2}(\sigma^2 i^2 - ri)\delta t,$$
$$b_i = 1 - (\sigma^2 i^2 + r)\delta t,$$
$$c_i = \frac{1}{2}(\sigma^2 i^2 + ri)\delta t.$$

We can write one expression for all numerical schemes

$$A_i v_{i-1,\nu-1} + B_i v_{i,\nu-1} + C_i v_{i+1,\nu-1} = a_i v_{i-1,\nu} + b_i v_{i,\nu} + c_i v_{i+1,\nu},$$

where

$$\begin{aligned} A_i &= -\frac{1}{2}(\sigma^2 i^2 - ri)\theta\delta t, & a_i &= \frac{1}{2}(\sigma^2 i^2 - ri)(1-\theta)\delta t, \\ B_i &= 1 + (\sigma^2 i^2 + r)\theta\delta t, & b_i &= 1 - (\sigma^2 i^2 + r)(1-\theta)\delta t, \\ C_i &= -\frac{1}{2}(\sigma^2 i^2 + ri)\theta\delta t, & c_i &= \frac{1}{2}(\sigma^2 i^2 + ri)(1-\theta)\delta t. \end{aligned}$$

The choice of θ gives the explicit ($\theta = 0$), the implicit ($\theta = 1$) and the Crank-Nicolson ($\theta = \frac{1}{2}$) scheme.

Summary of the PDE method

Advantages

- one computation gives price for many stock quotations and for all times $[0, T]$,
- replicating strategy,
- usually fast and accurate.

Disadvantages

- limited number of instruments that can be priced,
- heavy mathematics involved, especially for more advanced models than Black-Scholes,
- more complicated models are not feasible for the PDE method.