Financial Systems Design Part II

Finite Difference Method (FDM)

July 2020

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

The Big Picture

- Develop a mathematical model based on tradable assets in the market to calculate price of derivatives
- Model tells you how to value a derivative and hedge/replicate the derivative
- Clients of the bank will come with requests for certain type of new options or the bank will develop new options as investment products for clients
- Option payoff + Model = Option price / risk management

Outline

- Black Scholes PDE
- Explicit Finite Difference Method
- Implicit Finite Difference Method
- Crank Nicholson
- Richardson Extrapolation

Black Scholes PDE

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Quick Recap

- Key step in the derivation by BSM is finding a self-financing portfolio growing at a deterministic rate
- This rate must equal the risk free rate in order not to allow arbitrage in the system
- Equating the portfolio growth rate to the risk free rate results in the famous PDE

$$f_t + rxf_x + \frac{1}{2}\sigma^2 x^2 f_{xx} - rf = 0$$

- This means the price function of the derivative must follow this PDE for the system to be arbitrage free

Black-Scholes Equation

- Black-Scholes PDE where f is price function of the option
- $f(x,t) =$ price at time t when underlying price is x

$$f_t + rxf_x + \frac{1}{2}\sigma^2 x^2 f_{xx} - rf = 0$$

- We know the value of an option at maturity when $t = T$ for all values of x
- What we want to know is $f(x, 0)$ i.e. the price at current time for all values of x
- In other words, we need to move backwards in time

Black-Scholes Equation

- Alternatively, we can perform a change of variables where τ = T – t
- Then we get $\frac{\partial f}{\partial t} = \frac{\partial f}{\partial \tau}\frac{\partial \tau}{\partial t} = -\frac{\partial f}{\partial \tau}$
- So we can rewrite the BS PDE as

$$f_{\tau} = rxf_x + \frac{1}{2}\sigma^2 x^2 f_{xx} - rf$$

- Transforms PDE to a forward equation which is how FDM is presented in most numerical analysis books

Black-Scholes Equation

- Taking a vanilla call option as an example

$$f(x,0) = \max(x - K, 0) \;\; \textit{where } K \textit{ is the strike}$$

- We already know $f(x,\tau)$ for $0 < \tau \leq \mathrm{T}$ from your previous course
- However, there are options where we do not have the exact form of $f(x,\tau)$, e.g. American options
- Ultimately we want to price American options
- We will first spend time approximating European options since we have the exact solution which allows us to check for the error of the approximation

Black-Scholes Equation

- We are looking to use similar approximation methods as was used for ODEs to find $f(x, \tau)$
- Since we are dealing with PDEs now, we need to extend the methodology to deal with partial derivatives

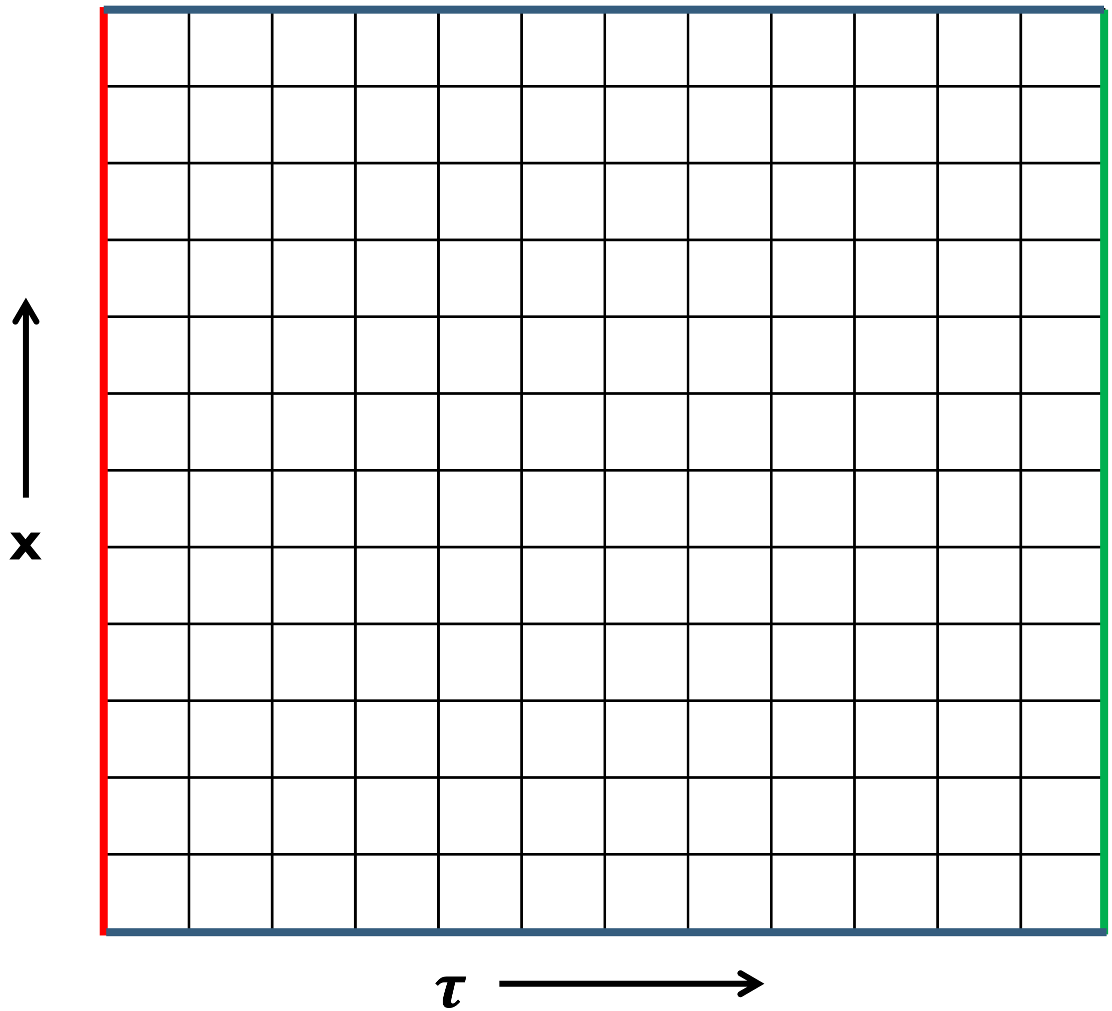
Explicit Finite Difference Method

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Explicit Finite Difference Method

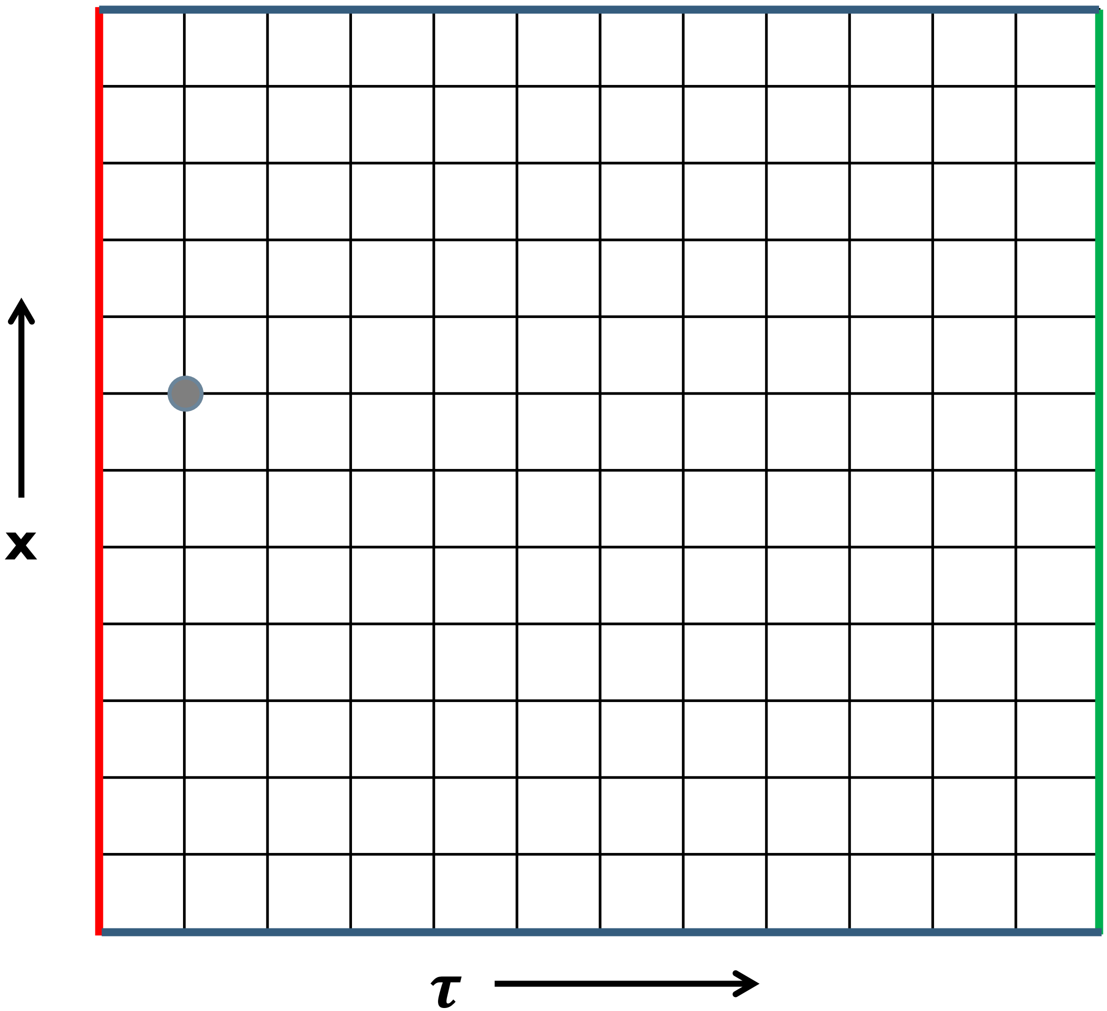
- Moving from a one dimensional problem with ODEs to 2 dimensions now
- Discretize the domain into equally spaced 2D mesh and use the PDE to evolve the solution
- An initial condition is needed to start similar to the Euler method for ODEs
- We also need to know what happens on the boundaries of the mesh i.e. boundary conditions
- This is thus called an Initial Boundary Problem

Explicit Finite Difference Method



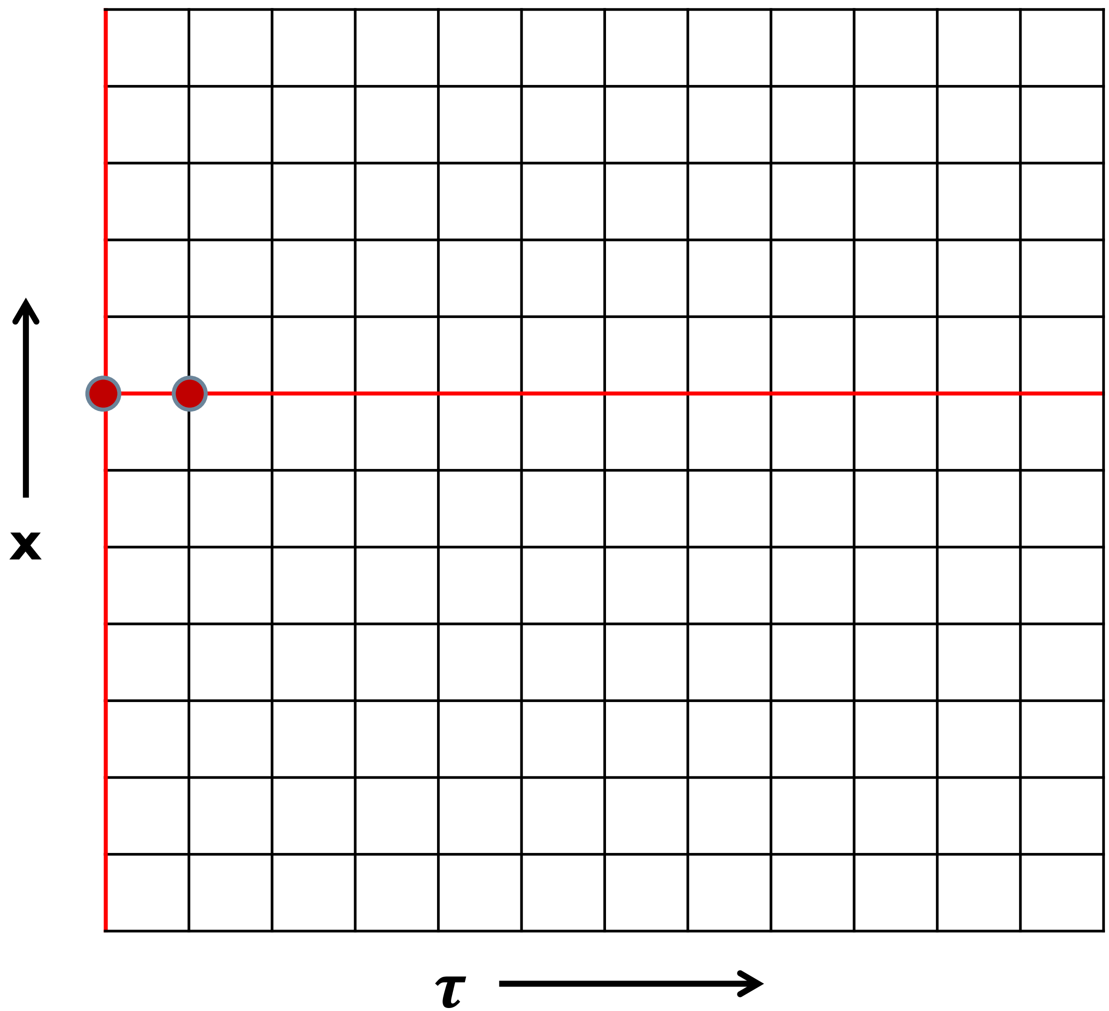
- Create equally spaced grid/mesh
- Initial boundary condition is the starting line
- We need the solution at the end
- To evolve forward with the PDE, we also need to know what happens at the boundaries

Explicit Finite Difference Method



- Start at $\tau = 0$ (red line)
- Value of $f(x, 0)$ which is the value of the option at maturity for all x is known
- Take a sample point where x = 100 and say this is a call option with strike at 90 so $f(100{,}0) = 10$
- We ultimately want to know $f(100, T)$ but we need to move in small steps to reduce the error so we start with $f(100, \Delta\tau)$ which is represented by the grey dot

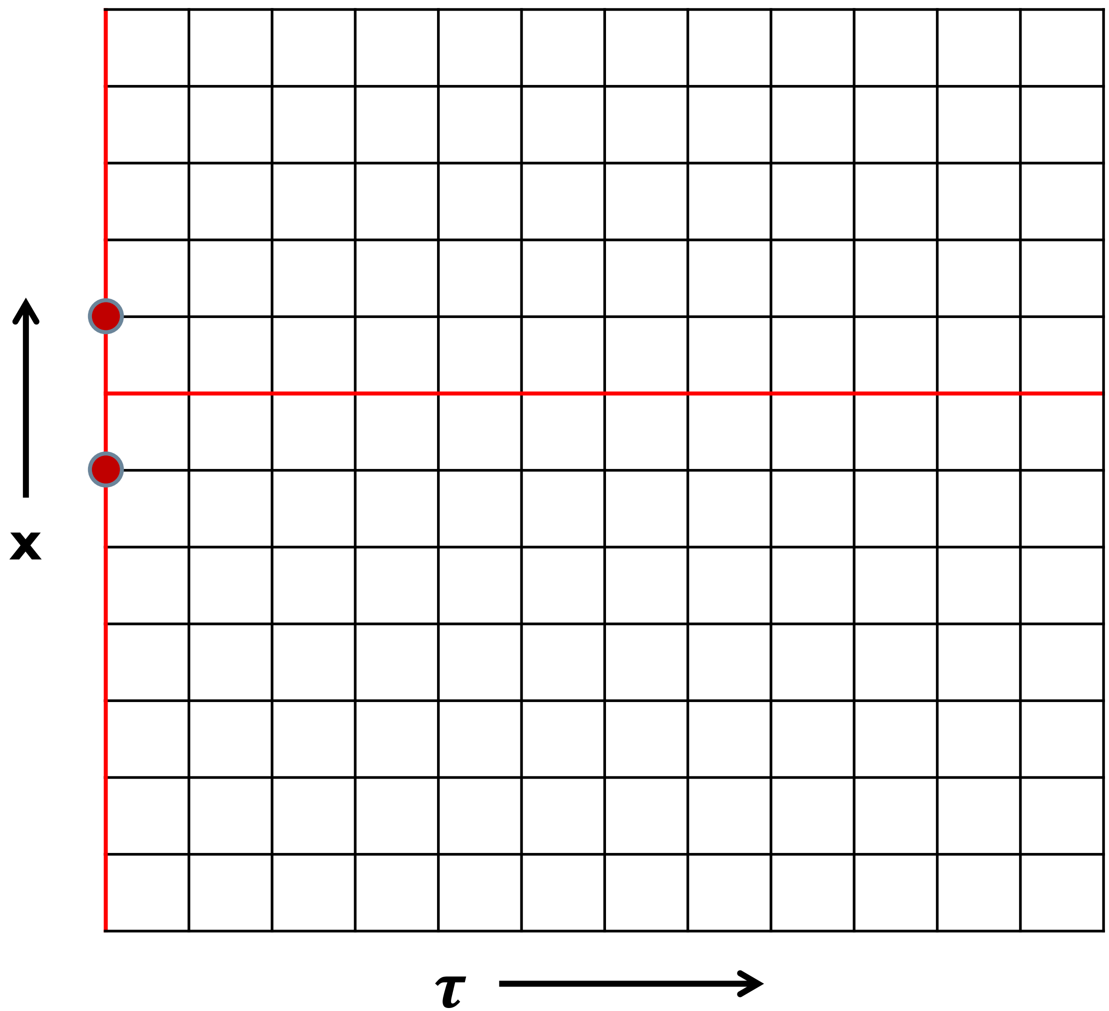
Explicit Finite Difference Method



Two points to compute Theta with forward differencing:

$$f_\tau = \frac{f(x,\tau+\Delta\tau)-f(x,\tau)}{\Delta\tau}$$

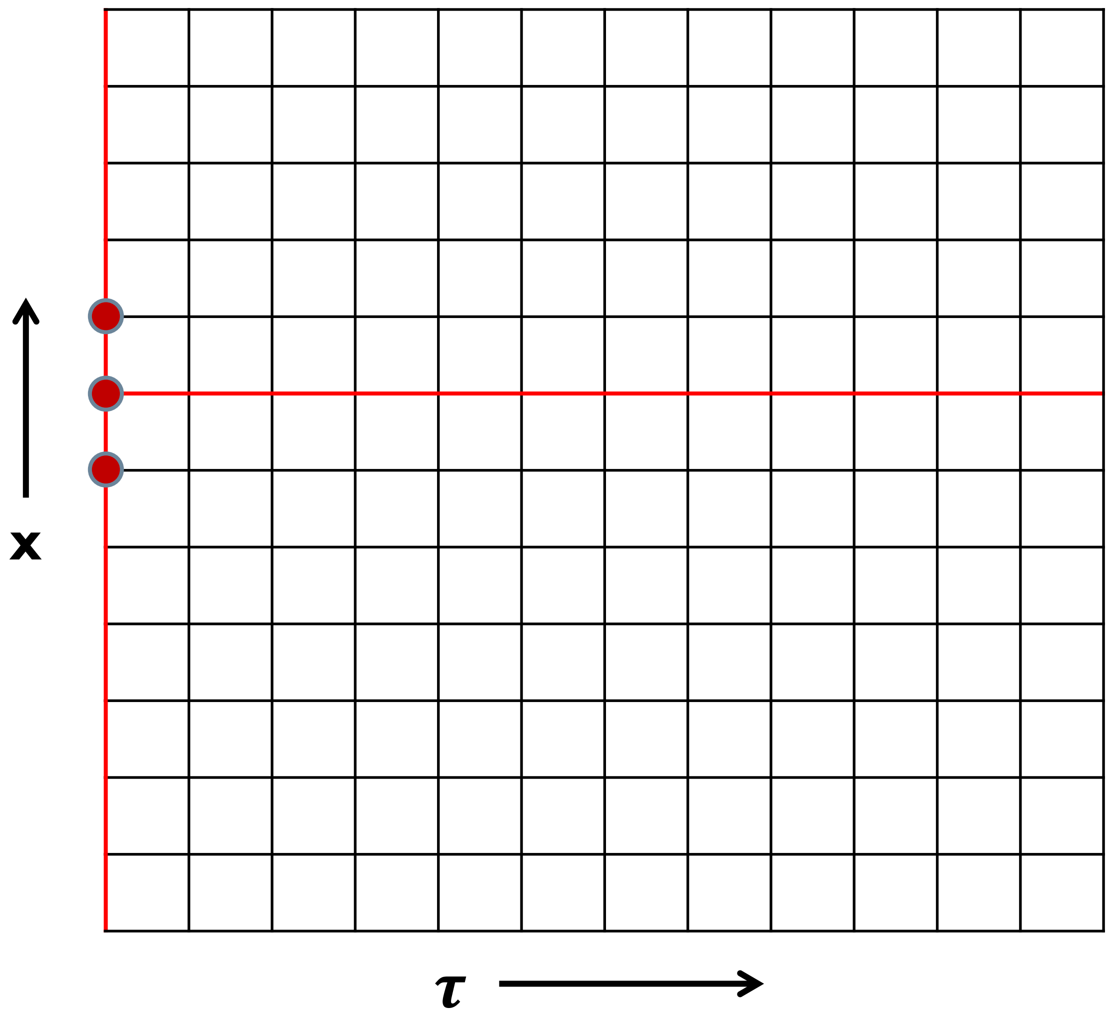
Explicit Finite Difference Method



Two points to compute Delta with central differencing:

$$f_x = \frac{f(x+\Delta x,\tau) - f(x-\Delta x,\tau)}{2\Delta x}$$

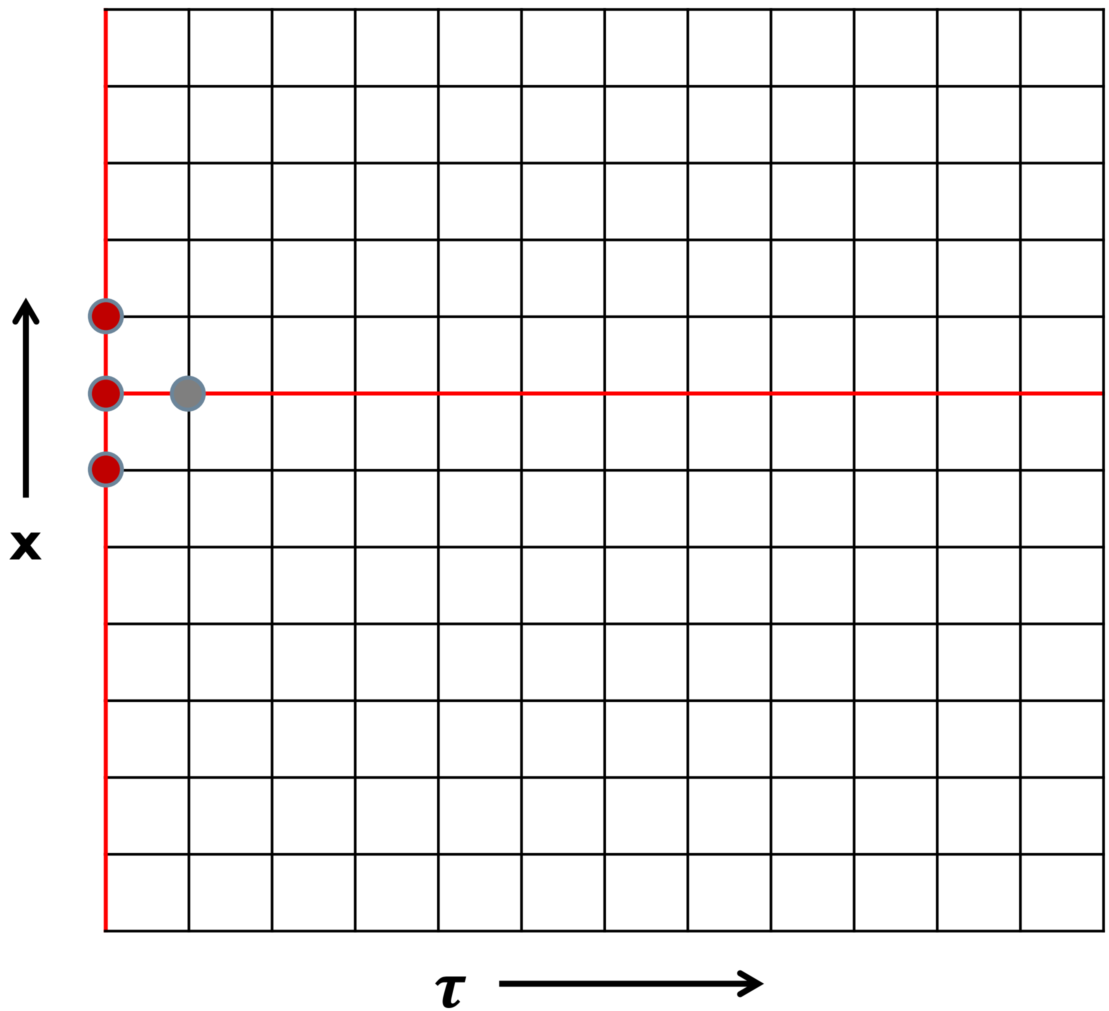
Explicit Finite Difference Method



Three points to compute gamma with central differencing:

$$f_{xx} = \frac{f(x+\Delta x,\tau)-2f(x,\tau)+f(x-\Delta x,\tau)}{\Delta x^2}$$

Explicit Finite Difference Method



Calculation of the point in grey only requires those points in red based on the Black Scholes PDE

All red points are known values as it is values of $f(x, 0)$

Explicit Finite Difference Method

- Using central differencing for Delta and Gamma
- Using forward differencing for Theta
- $V_i^k = f(\tau = \tau_0 + k\Delta\tau, x = x_0 + i\Delta x)$ i.e. price at discretized time k & asset price I
- Similarly, $x_i = x_0 + i\Delta x$
- $f_\tau = rxf_x + \frac{1}{2}\sigma^2 x^2 f_{xx} - rf$
- $\frac{V_i^{k+1} - V_i^k}{\Delta\tau} = rx_i \frac{V_{i+1}^k - V_{i-1}^k}{2\Delta x} + \frac{1}{2}\sigma^2 x_i^2 \frac{V_{i+1}^k - 2V_i^k + V_{i-1}^k}{\Delta x^2} - rV_i^k$

Explicit Finite Difference Method

- Rearrange the equation so that all the derivative prices at the next time step is on the LHS and those in the current time step is on the RHS

- $V_i^{k+1} = \left(\frac{1}{2}\sigma^2 x_i^2 \frac{\Delta\tau}{\Delta x^2} - \frac{1}{2} r x_i \frac{\Delta\tau}{\Delta x}\right) V_{i-1}^k + \left(1 - \sigma^2 x_i^2 \frac{\Delta\tau}{\Delta x^2} - r\Delta\tau\right) V_i^k + \left(\frac{1}{2} r x_i \frac{\Delta\tau}{\Delta x} + \frac{1}{2}\sigma^2 x_i^2 \frac{\Delta\tau}{\Delta x^2}\right) V_{i+1}^k$

Initial Condition

- For vanilla options:
  - Call: $v_0 = f(\tau = 0) = \max(x - K, 0)$
  - Put: $v_0 = f(\tau = 0) = \max(K - x, 0)$

Boundary Conditions

- We also need boundary conditions in x for similar reasons as we did for 2nd order ODEs
- Our boundaries are $x = 0$ and $x = \infty$ (large values)
- Plug in $x = 0$ to the Black Scholes PDE

$$f_\tau = rxf_x + \frac{1}{2}\sigma^2 x^2 f_{xx} - rf$$

- The x derivatives go to zero: $f_\tau = -rf$
  - This is called a natural boundary condition
- At $x = \infty$ we usually just assume $f_{xx} = 0$

Loops or Matrices

- Can be programmed using loops or matrices
- Built in packages with Python or Matlab makes programming with matrices much easier than loops
- Let's try both and see which is easier to programme

In-Class Exercise

- Let's price a vanilla call option using loops
- Assume r = 3%, σ = 30%, K = $100, T = 1
- Let's use 2 times of the strike as the upper bound for the mesh
- Try Δx = 1 i.e. 200 steps
- And $\Delta \tau$ = 0.00025 i.e. 4,000 steps
- Plot the option price v vs x

In-Class Exercise

- Remember that

$$f_\tau = rxf_x + \frac{1}{2}\sigma^2 x^2 f_{xx} - rf$$

$$f_\tau \approx \frac{V^{k+1} - V^k}{\Delta\tau}$$

So

$$V^{k+1} \approx V^k + \Delta\tau f_\tau$$

- At x = 0, we have $V_0^{k+1} = (1 - r\Delta t)V_0^k$
- At large x, we have $V_N^{k+1} = 2V_{N-1}^{k+1} - V_{N-2}^{k+1}$

In-Class Exercise

- Now we try with matrices

$$V_i^{k+1} = \left(\frac{1}{2}\sigma^2 x_i^2 \frac{\Delta\tau}{\Delta x^2} - \frac{1}{2} r x_i \frac{\Delta\tau}{\Delta x}\right) V_{i-1}^k + \left(1 - \sigma^2 x_i^2 \frac{\Delta\tau}{\Delta x^2} - r\Delta\tau\right) V_i^k + \left(\frac{1}{2} r x_i \frac{\Delta\tau}{\Delta x} + \frac{1}{2}\sigma^2 x_i^2 \frac{\Delta\tau}{\Delta x^2}\right) V_{i+1}^k$$

- These equations naturally form a tri-diagonal matrix
- At x = 0, we have $V_0^{k+1} = (1 - r\Delta t)V_0^k$
- At large x, we have $V_N^{k+1} = 2V_{N-1}^{k+1} - V_{N-2}^{k+1}$

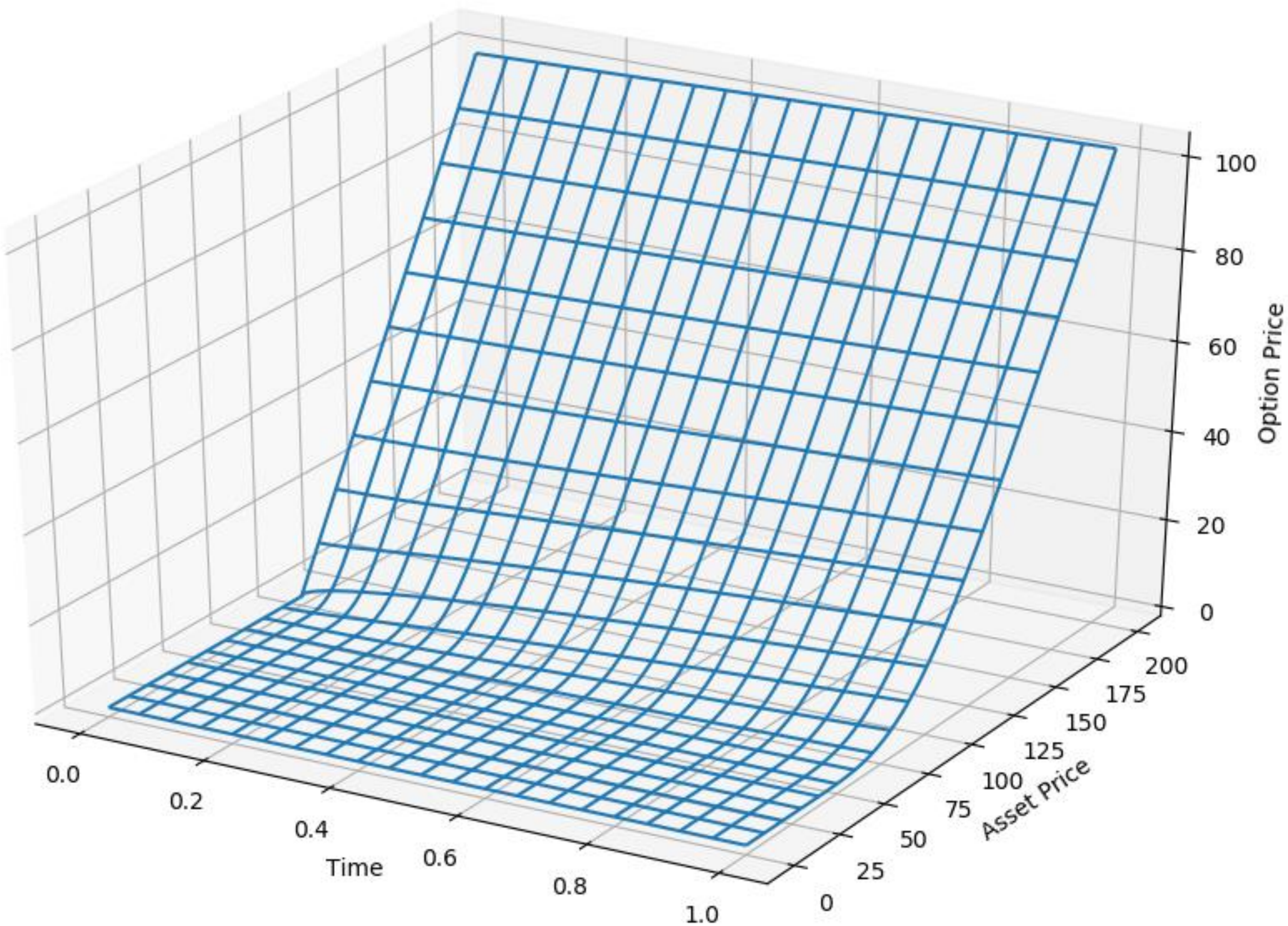
In-Class Exercise

$$\begin{pmatrix} V_0^{k+1} \\ V_1^{k+1} \\ \vdots \\ V_{n-1}^{k+1} \\ V_n^{k+1} \end{pmatrix} = A \begin{pmatrix} V_0^{k} \\ V_1^{k} \\ \vdots \\ V_{n-1}^{k} \\ V_n^{k} \end{pmatrix}$$

$A =$

$$\begin{bmatrix} (1 - r\Delta t) & 0 & 0 & \cdots & 0 \\ \left(\frac{1}{2}\sigma^2 x_1^2 \frac{\Delta\tau}{\Delta x^2} - \frac{1}{2} r x_1 \frac{\Delta\tau}{\Delta x}\right) & \left(1 - \sigma^2 x_1^2 \frac{\Delta\tau}{\Delta x^2} - r\Delta\tau\right) & \left(\frac{1}{2} r x_1 \frac{\Delta\tau}{\Delta x} + \frac{1}{2}\sigma^2 x_1^2 \frac{\Delta\tau}{\Delta x^2}\right) & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \left(\frac{1}{2}\sigma^2 x_{n-1}^2 \frac{\Delta\tau}{\Delta x^2} - \frac{1}{2} r x_{n-1} \frac{\Delta\tau}{\Delta x}\right) & \left(1 - \sigma^2 x_{n-1}^2 \frac{\Delta\tau}{\Delta x^2} - r\Delta\tau\right) & \left(\frac{1}{2} r x_1 \frac{\Delta\tau}{\Delta x} + \frac{1}{2}\sigma^2 x_{n-1}^2 \frac{\Delta\tau}{\Delta x^2}\right) \\ 0 & 0 & \cdots & ? & ? \end{bmatrix}$$

Evolution of Option Price Curve



In-Class Exercise

- Notice we are using much more steps for the time variable than the asset price variable
- What if we try a more balanced ratio?
- Try Δx = 1 i.e. 200 steps
- And $\Delta \tau$ = 0.0025 i.e. 400 steps

Stability of Explicit Method

- There are stability issues with the explicit method where a small error can get magnified with each time step
- The behavior of the error can be analyzed by Von Neumann stability analysis which works for linear PDEs
- We will not go into the mathematics which is quite involved and simply state the condition that provides stability
- The condition is $\Delta\tau < \frac{1}{N_x^2\sigma^2}$ where N_x is number of steps for x
- With $N_x = 200, \sigma = 30\%$, it means we need $\Delta\tau < \frac{1}{3600}$

Stability of Explicit Method

- The stability condition violation produces negative probabilities
- Notice that the condition imposes high computational cost
- If we need to improve accuracy by doubling the number of asset steps then the number of time times must increase by 4 times
- This causes total computational time to increase 8 folds

In-Class Exercise

- Add the appropriate modification to the Explicit FDM from the last exercise to determine a suitable $\Delta\tau$ to use which ensures the stability condition is not violated

Other Boundary Conditions

- The boundary conditions in the preceding example is generic to most derivatives
- There are also other ways to write simpler boundary conditions specific for vanilla call and put options
- For call options, we can also use $V = 0$ when $x = 0$ and $V = S - Ke^{-rt}$ for large x
- Similarly, we can also use $V = 0$ for large x when it comes to put options and $V = Ke^{-rt}$ when $x = 0$

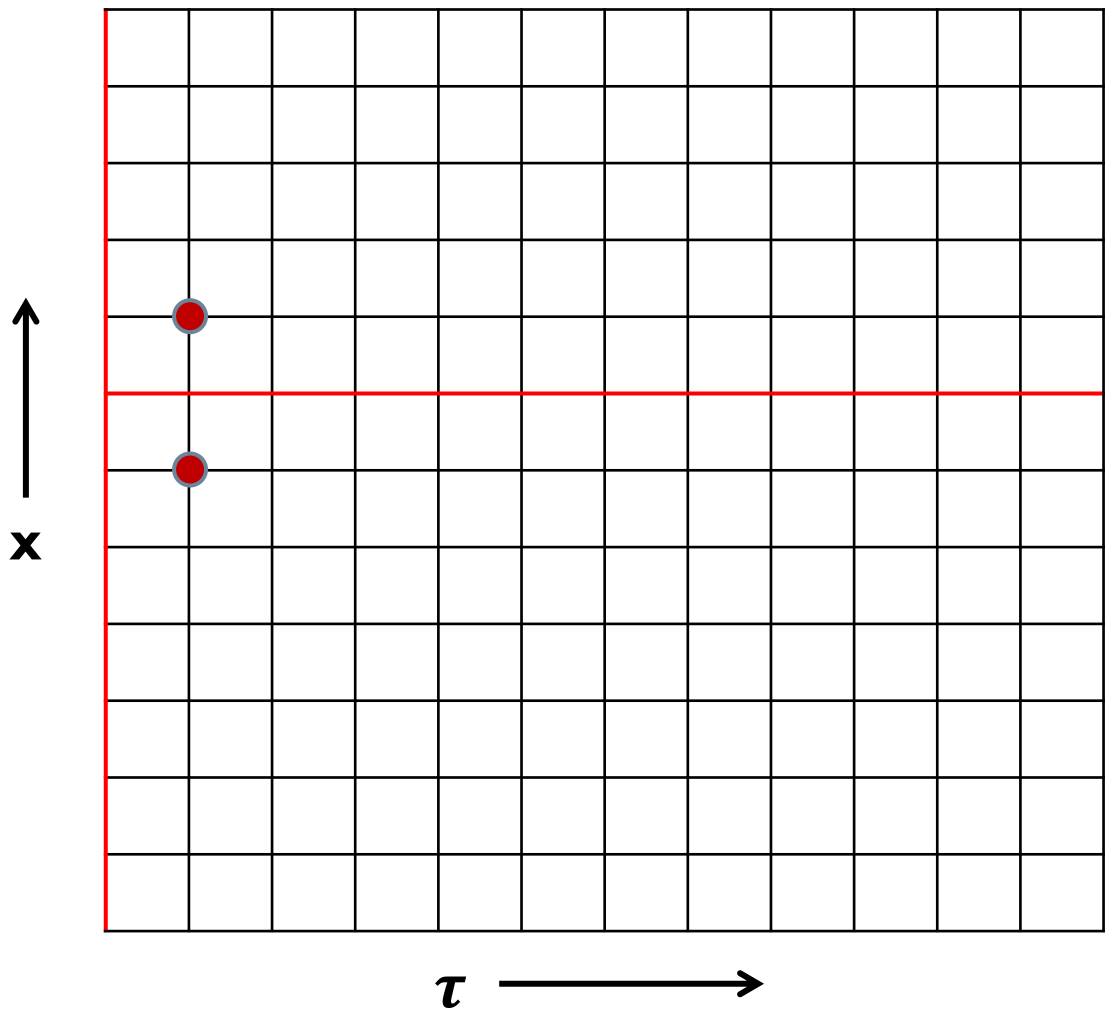
Implicit Finite Difference Method

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Explicit vs. Implicit

- For even a simple problem we need a huge number of time steps with the EXPLICIT method
- We can achieve unconditional stability if we choose to approximate the x derivatives at a forward time step
  - Similar to the homework problem for Euler method where we used $\hat{y}_{i+1}$ instead
- This is called the IMPLICIT method

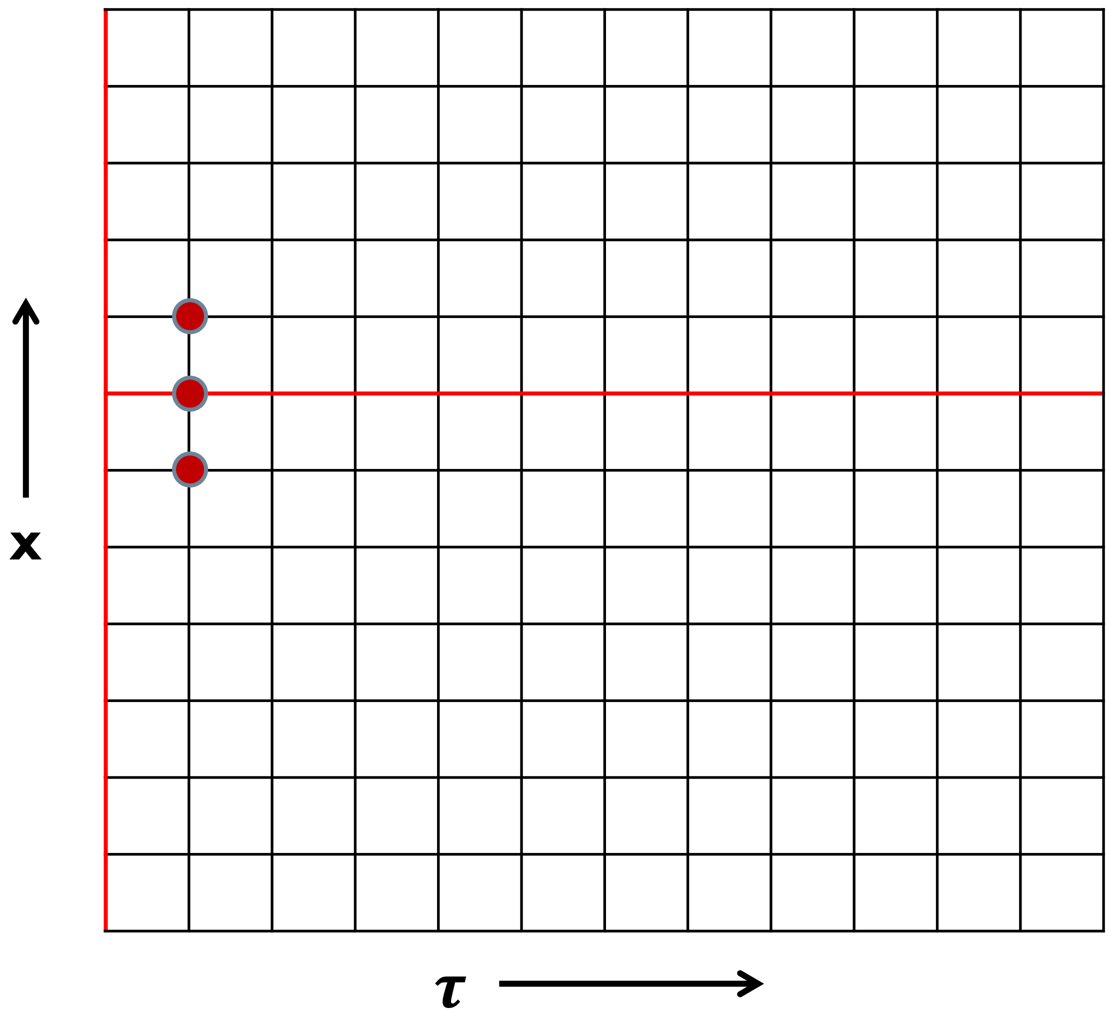
Implicit Finite Difference Method



Two points to compute Delta with central differencing:

$$f_x = \frac{f(x+\Delta x,\tau+\Delta\tau)-f(x-\Delta x,\tau+\Delta\tau)}{2\Delta x}$$

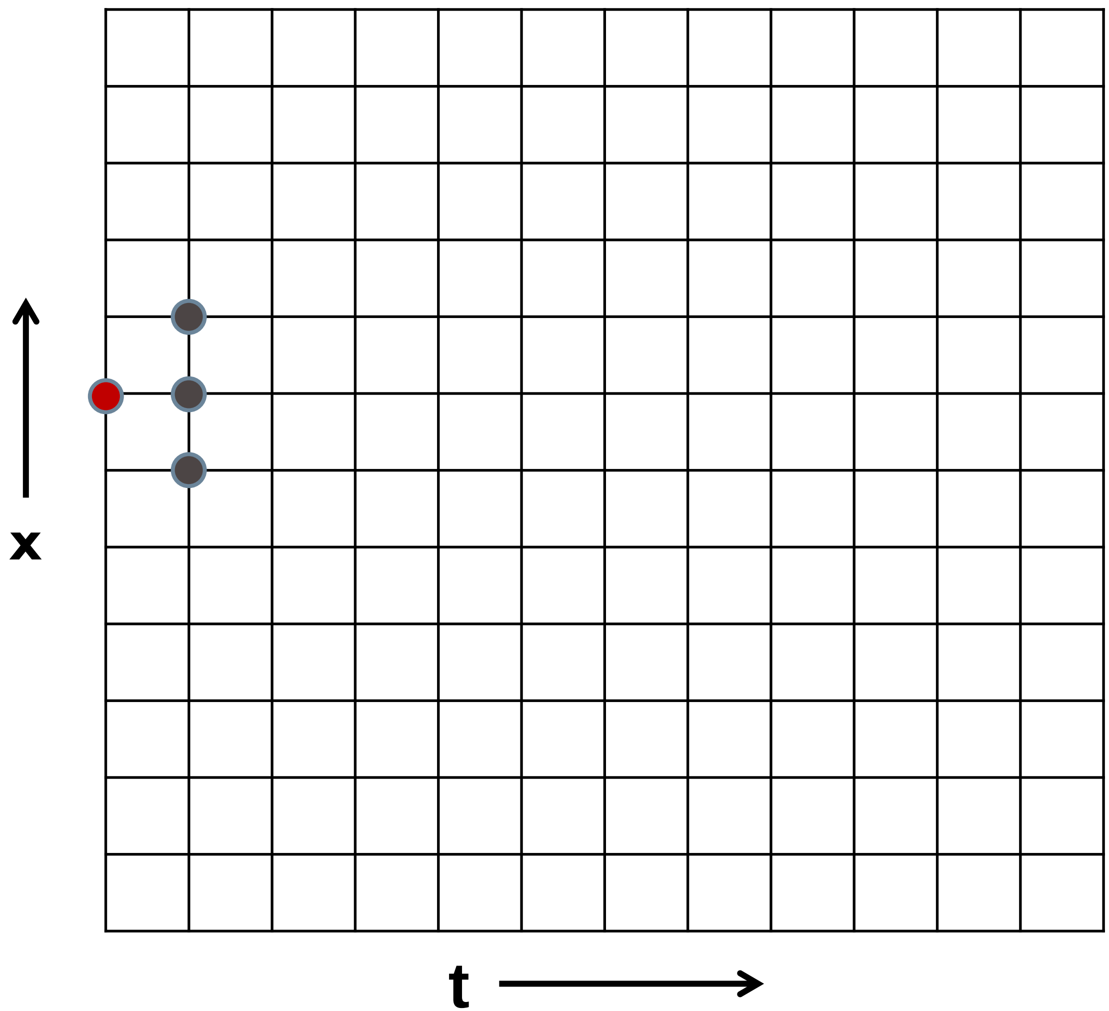
Implicit Finite Difference Method



Three points to compute gamma with central differencing:

$$f_{xx} = \frac{f(x+\Delta x, \tau+\Delta\tau) - 2f(x, \tau+\Delta\tau) + f(x-\Delta x, \tau+\Delta\tau)}{\Delta x^2}$$

Implicit Finite Difference Method



- Form an equation of 3 unknown points in grey using the known point in red
- Boundary conditions supply missing equations for system to have a solution (recall linear algebra)
- Solve system of linear equations
- Recall 2nd Order ODE

Implicit Finite Difference Method

- The main difference is using the values of V at the next time step i+1 to compute the x derivatives

$$f_\tau = rxf_x + \frac{1}{2}\sigma^2 x^2 f_{xx} - rf$$

$$\frac{V_i^{k+1} - V_i^k}{\Delta\tau} = rx_i \frac{V_{i+1}^{k+1} - V_{i-1}^{k+1}}{2\Delta x} + \frac{1}{2}\sigma^2 x_i^2 \frac{V_{i+1}^{k+1} - 2V_i^{k+1} + V_{i-1}^{k+1}}{\Delta x^2} - rV_i^{k+1}$$

$$\left(\frac{1}{2} r x_i \frac{\Delta\tau}{\Delta x} - \frac{1}{2}\sigma^2 x_i^2 \frac{\Delta\tau}{\Delta x^2}\right) V_{i-1}^{k+1} + \left(1 + r\Delta\tau + \sigma^2 x_i^2 \frac{\Delta\tau}{\Delta x^2}\right) V_i^{k+1}$$
$$+ \left(-\frac{1}{2} r x_i \frac{\Delta\tau}{\Delta x} - \frac{1}{2}\sigma^2 x_i^2 \frac{\Delta\tau}{\Delta x^2}\right) V_{i+1}^{k+1} = V_i^k$$

Implicit FDM

- Same conditions at the boundaries of x
- At x = 0, we have $V_0^{k+1} = (1 - r\Delta t)V_0^k$
- At large x, we have $V_N^{k+1} = 2V_{N-1}^{k+1} - V_{N-2}^{k+1}$
- These equations naturally form a matrix equation

$$A\vec{v}_{k+1} = \vec{v}_k$$

- Both conditions can be build into the matrix e.g.

$$\begin{pmatrix} \frac{1}{1 - r\Delta t} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \quad -2 \quad 1 \end{pmatrix} \vec{v}_{k+1} = \begin{pmatrix} v_0^k \\ \vdots \\ v_{N-1}^k \\ 0 \end{pmatrix}$$

Stability of Implicit Method

- The implicit method provides unconditional stability
- Computational cost comes in the form of solving the system of linear equations

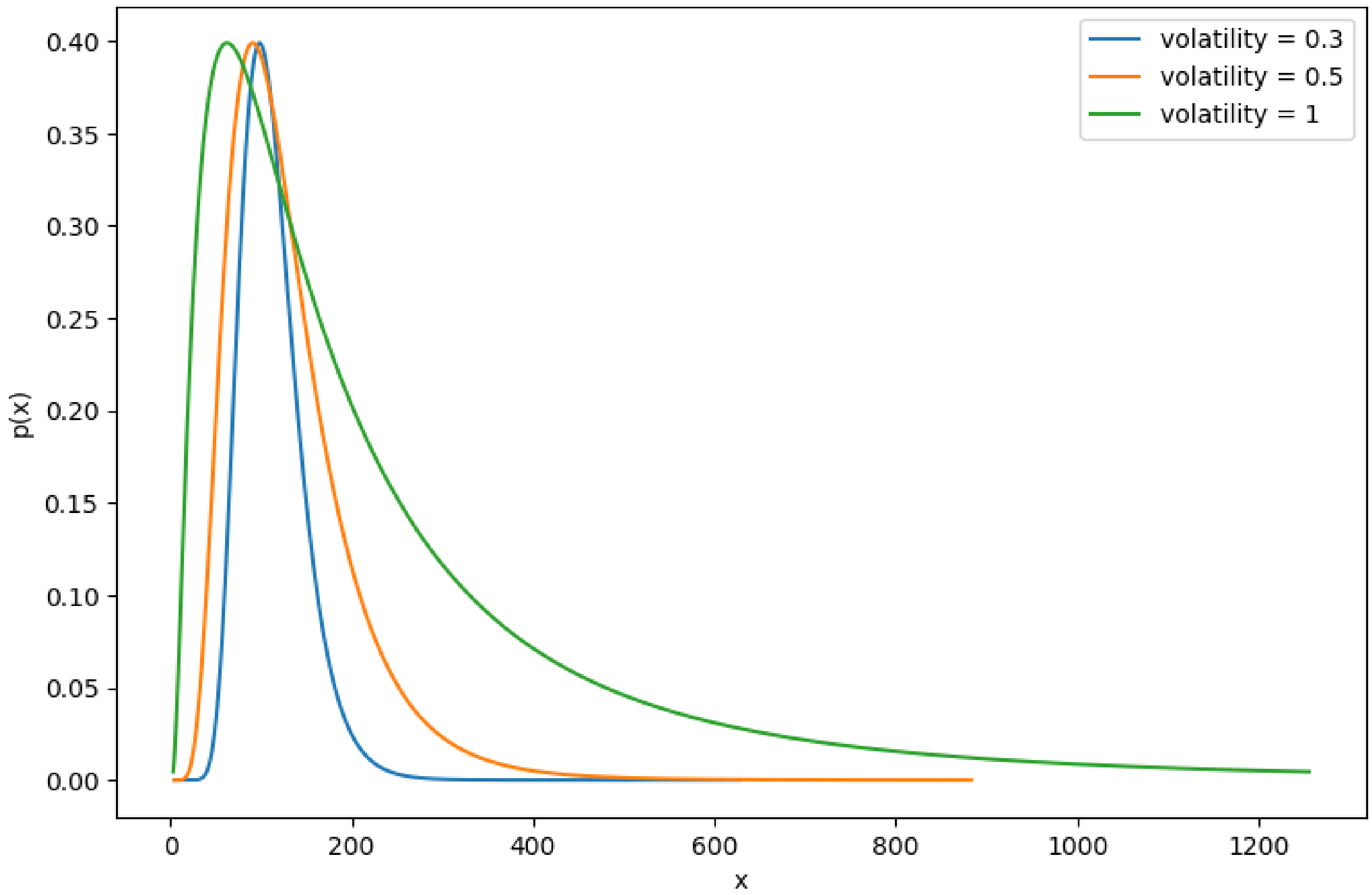
In-Class Exercise

- Same example to implement with the implicit method
- Assume r = 3%, σ = 30%, K = $100, T = 1
- Let's use 2 times of the strike as the upper bound for the mesh
- Try Δx = 1 i.e. 200 steps
- And $\Delta \tau$ = 0.0025 i.e. 400 steps
- Plot the option price v vs x

In-Class Exercise

- Back to the same example
- Try increasing volatility to 70%
- What happens to the error?
- Does the error reduce if we reduce the size of the steps?
- What is wrong?

Lognormal Distribution



Upper Bound for x

- One way to automate the solution is to use the inverse CDF to solve for a suitable upper bound
- E.g. Solve for the x value that would cover 99% of the distribution
- In the example with 70% volatility, the x value will be ~400

Crank Nicolson

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

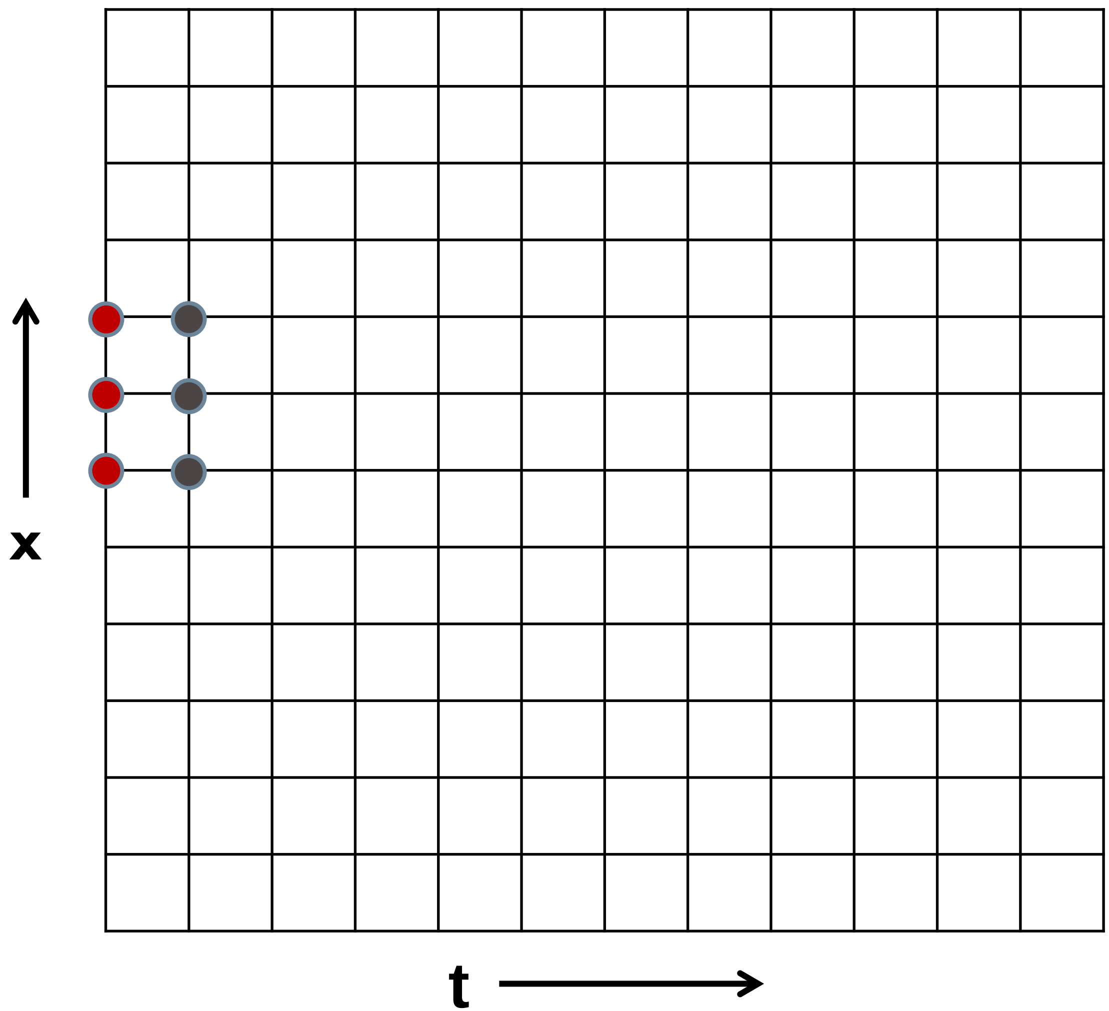
Order of Error

- Both the Explicit and Implicit method have local truncation error of $O(\Delta t^2, \Delta t \Delta x^2)$ and global error of $O(\Delta t, \Delta x^2)$
- The stability requirement for Explicit method requires large number of time steps
- The form of the Implicit method requires solving a linear system of equations
- In other words, both have its costs
- Can we improve the error without much added cost?

Crank Nicholson

- Can be written as a 2nd Order Runge Kutta method
- An average of the implicit and explicit method
- It also provides unconditional stability
- The key benefit is that the error is $O(\Delta t^2, \Delta x^2)$
- Computational cost is only marginally higher than the Implicit method

Crank Nicolson



- We use all 6 points with the Crank Nicolson method

Crank Nicolson

- $f_\tau = rxf_x + \frac{1}{2}\sigma^2 x^2 f_{xx} - rf$
- $\frac{V_i^{k+1} - V_i^k}{\Delta\tau} = rx_i \frac{V_{i+1}^k - V_{i-1}^k}{2\Delta x} + \frac{1}{2}\sigma^2 x_i^2 \frac{V_{i+1}^k - 2V_i^k + V_{i-1}^k}{\Delta x^2} - rV_i^k$
- In matrix form, we write Explicit FDM as $f_\tau = A\overrightarrow{v_\tau}$ which is rearranged to $\overrightarrow{v_{\tau+1}} = (I + \Delta\tau A)\overrightarrow{v_\tau}$ since $f_\tau = \frac{\overrightarrow{v_{\tau+1}} - \overrightarrow{v_\tau}}{\Delta\tau}$ where I is the identity matrix
- For Implicit method, it will be $f_\tau = A\overrightarrow{v_{\tau+1}}$ which is rearranged to $(I - \Delta\tau A)\overrightarrow{v_{\tau+1}} = \overrightarrow{v_\tau}$

Crank Nicolson

- Crank Nicolson is $f_\tau = \frac{1}{2}A\overrightarrow{v_{t+1}} + \frac{1}{2}A\overrightarrow{v_t}$ which rearranges to $(I - \frac{1}{2}\Delta\tau A)\overrightarrow{v_{t+1}} = \ (I + \frac{1}{2}\Delta\tau A)\overrightarrow{v_t}$

Crank Nicolson

- Explicit FDM

$$V_i^{k+1} = \left(\frac{1}{2}\sigma^2 x_i^2 \frac{\Delta\tau}{\Delta x^2} - \frac{1}{2} r x_i \frac{\Delta\tau}{\Delta x}\right) V_{i-1}^k + \left(1 - r\Delta\tau - \sigma^2 x_i^2 \frac{\Delta\tau}{\Delta x^2}\right) V_i^k + \left(\frac{1}{2} r x_i \frac{\Delta\tau}{\Delta x} + \frac{1}{2}\sigma^2 x_i^2 \frac{\Delta\tau}{\Delta x^2}\right) V_{i+1}^k$$

- Implicit FDM

$$\left(\frac{1}{2} r x_i \frac{\Delta\tau}{\Delta x} - \frac{1}{2}\sigma^2 x_i^2 \frac{\Delta\tau}{\Delta x^2}\right) V_{i-1}^{k+1} + \left(1 + r\Delta\tau + \sigma^2 x_i^2 \frac{\Delta\tau}{\Delta x^2}\right) V_i^{k+1} + \left(-\frac{1}{2} r x_i \frac{\Delta\tau}{\Delta x} - \frac{1}{2}\sigma^2 x_i^2 \frac{\Delta\tau}{\Delta x^2}\right) V_{i+1}^{k+1} = V_i^k$$

Crank Nicolson

$$\frac{V_i^{k+1}-V_i^k}{\Delta\tau} = \frac{1}{2}\left[rx_i\frac{V_{i+1}^k-V_{i-1}^k}{2\Delta x} + \frac{1}{2}\sigma^2 x_i^2\frac{V_{i+1}^k-2V_i^k+V_i^k-1}{\Delta x^2} - rV_i^k\right] +$$

$$\frac{1}{2}\left[rx_i\frac{V_{i+1}^{k+1}-V_{i-1}^{k+1}}{2\Delta x} + \frac{1}{2}\sigma^2 x_i^2\frac{V_{i+1}^{k+1}-2V_i^{k+1}+V_{i-1}^{k+1}}{\Delta x^2} - rV_i^{k+1}\right]$$

Rearranges to

$$\frac{1}{2}\left(\frac{1}{2}rx_i\frac{\Delta\tau}{\Delta x} - \frac{1}{2}\sigma^2 x_i^2\frac{\Delta\tau}{\Delta x^2}\right)V_{i-1}^{k+1} + \left(1+\frac{1}{2}\left(r\Delta\tau + \sigma^2 x_i{}^2\frac{\Delta\tau}{\Delta x^2}\right)\right)V_i^{k+1} +$$

$$\frac{1}{2}\left(-\frac{1}{2}rx_i\frac{\Delta\tau}{\Delta x} - \frac{1}{2}\sigma^2 x_i^2\frac{\Delta\tau}{\Delta x^2}\right)V_{i+1}^{k+1} = \frac{1}{2}\left(\frac{1}{2}\sigma^2 x_i^2\frac{\Delta\tau}{\Delta x^2} - \frac{1}{2}rx_i\frac{\Delta\tau}{\Delta x}\right)V_{i-1}^k +$$

$$\left(1-\frac{1}{2}\left(\sigma^2 x_i^2\frac{\Delta\tau}{\Delta x^2} + r\Delta\tau\right)\right)V_i^k + \frac{1}{2}\left(\frac{1}{2}rx_i\frac{\Delta\tau}{\Delta x} + \frac{1}{2}\sigma^2 x_i^2\frac{\Delta\tau}{\Delta x^2}\right)V_{i+1}^k$$

In-Class Exercise

- Same example to implement with the implicit method
- Assume r = 3%, σ = 30%, K = $100, T = 1
- Let's use 2 times of the strike as the upper bound for the mesh
- Try Δx = 1 i.e. 200 steps
- And $\Delta\tau$ = 0.0025 i.e. 400 steps
- Plot the option price v vs x

Choice of FDM Type

- Order of the error
- Stability restrictions
- Ease of programming
- Flexibility to boundary conditions and decision points

In-Class Exercise

- Use the Crank Nicolson Method to price the put option
- Assume $r = 3\%$, $\sigma = 40\%$, $K = \$90$, $T = 2$,
- Create a function to compute the actual put price using the Black Scholes formula to check on the error
- Previously it was an At-The-Money (ATM) option that was being priced i.e. strike price = spot price
- Now assume $x_0 = 100$
- Experiment with the step sizes and boundaries for x

Richardson Extrapolation

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Richardson Extrapolation

- Recall Richardson Extrapolation with 1 dimension
- Suppose we are using an expression $g(h)$ with error ch^n to estimate a quantity G
- We know $g(\cdot)$ at two different step sizes h_1 and h_2:

$$g(h_1) = G + ch_1^n + O(h_1^{n+1})$$
$$g(h_2) = G + ch_2^n + O(h_2^{n+1})$$

- We combine these two equations to eliminate the error term, obtaining:

$$\hat{G} = \frac{\left(\frac{h_1}{h_2}\right)^n g(h_2) - g(h_1)}{\left(\frac{h_1}{h_2}\right)^n - 1}$$

Richardson Extrapolation

- Applying Richardson Extrapolation to the Explicit FDM

$$\hat{V}_1 = V + c_1\Delta t_1 + c_2\Delta x_1^2 + O(\Delta t^2, \Delta x^3)$$
$$= V + \Delta x_1^2\left(c_1\frac{\Delta t_1}{\Delta x_1^2} + c_2\right) + O(\Delta t^2, \Delta x^3)$$
$$\hat{V}_2 = V + c_1\Delta t_2 + c_2\Delta x_2^2 + O(\Delta t^2, \Delta x^3)$$
$$= V + \Delta x_2^2\left(c_1\frac{\Delta t_2}{\Delta x_2^2} + c_2\right) + O(\Delta t^2, \Delta x^3)$$

- Choosing $\frac{\Delta t_1}{\Delta x_1^2} = \frac{\Delta t_2}{\Delta x_2^2}$ we get a better solution with the estimate $\hat{V}_R = \frac{\Delta x_2^2\hat{V}_1 - \Delta x_1^2\hat{V}_2}{\Delta x_2^2 - \Delta x_1^2} + O(\Delta t^2, \Delta x^3)$

The Greeks

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Calculating Greeks with FDM

- The Greeks are essential information in risk management of derivative portfolios
- The Greeks of a portfolio is a linear combination of the Greeks of the individual positions
- Delta: $\frac{\partial C}{\partial x} = N(d_1)$ and $\frac{\partial P}{\partial x} = N(d_1) - 1$
- Gamma: $\frac{\partial^2 C}{\partial x^2} = \frac{\partial^2 P}{\partial x^2} = \frac{N'(d_1)}{x\sigma\sqrt{\tau}}$
- Theta: $\frac{\partial C}{\partial t} = -\frac{xN'(d_1)\sigma}{2\sqrt{\tau}} - rKe^{-r\tau}N(+d_2)$

Delta hedging

- Imagine you are a trader who sold a call option to a client and now you need to hedge the risk exposure
- Ignoring bid/ask spreads, on day 1 you receive the option premium equal to the value of the option
- If the underlying price moves up, the value of the option moves up and you liability goes up
- How do you hedge this risk?

Delta hedging

- If the underlying price moves up 1 unit, the option price should go up by $\frac{\partial c}{\partial x}$
- If you buy $\frac{\partial c}{\partial x}$ units of shares then the value of the shares you own will move up in the same amount as the option
- This is essentially Delta hedging where you try to adjust the portfolio to be Delta neutral
- But there are some things to note

Delta hedging

- Buying the shares will require financing which generates additional cost and that cost is factored into the option premium that you receive
- The Delta does not stay constant so similar to the problem you encounter with Euler method, there will be errors
- For example, as the underlying price moves up, the Delta moves up which requires you to buy more shares to keep up

Delta hedging

- Not a problem in the Black Scholes world but a problem in the real world
- In the case, where you are short the call, you are "chasing" the market by having to buy more shares as prices move up which means the "errors" cost you PnL
- This happens when you are essentially short Gamma and the "errors" that build up must be factored into the option price and is linked to the volatility of the underlying

Delta hedging

- In the reverse case where the client sells you the call then you are selling the shares into rising prices which earns you PnL
- The question is whether the profits that accumulate is enough to offset the decay in option value as Theta is negative (remember you paid for the option)
- This is sometimes called Gamma Scalping as you are long Gamma in this case

In Class Exercise

- FDM provides an easy way to calculate the Greeks with a grid of option prices for differencing technique
- Modify the Crank Nicolson cell to compute the Delta, Gamma and Theta for the same call option
- 1-year call option with strike at 100
- Volatility of the underlying is 30%, interest rates are 3% and current price of the underlying is 100 i.e. $x_0 = 100$

In Class Exercise

- Delta: $f_x = \frac{f(x_0+\Delta x, T) - f(x_0-\Delta x, T)}{2\Delta x}$
- Gamma: $f_{xx} = \frac{f(x_0+\Delta x, T) - 2f(x_0, T) + f(x_0-\Delta x, T)}{\Delta x^2}$
- Theta: $f_\tau = \frac{f(x_0, T) - f(x_0, T-\Delta\tau)}{\Delta\tau}$