Introduction to Numerical Methods

In this lecture. . .

- The justification for pricing by Monte Carlo simulation

- Grids and discretization of derivatives
- The explicit finite-difference method

F.D.M

By the end of this lecture you will be able to

- implement the Monte Carlo method for simulating asset paths and pricing options   Probability – simulations

- implement the explicit finite-difference method for pricing options   Deterministic – PDE sol^n method

Computational Finance

Introduction Numerical Analysis for finance

More often than not we must solve option-pricing problems by numerical means.

It is rare to be able to find closed-form solutions for prices unless both the contract and the model are very simple.

The most useful numerical techniques are Monte Carlo simulations and finite-difference methods.

Monte Carlo Simulations

Relationship between derivative values and (simulations)

Expectations

Theory says: $N(d_{12})$ $N(d_{\pm})$

- The fair value of an option is the present value of the expected payoff at expiry under a *risk-neutral* random walk for the underlying.

density

measure

The risk-neutral random walk for S is

$\mu \longrightarrow r$

- $$dS = rS\,dt + \sigma S\,dX^{\mathbb{Q}}.$$

This is simply our usual lognormal random walk but with the risk-free rate instead of the real growth rate.

Justification: For risk-neutral pricing

- Binomial method ✓
- Black–Scholes Equation similar to backward Kolmogorov equation ✓
- Martingale theory

Step in B.S.E $V(S,t) = e^{-r(T-t)} U(S,t)$

We can therefore write

- option value $= e^{-r(T-t)} E\,[\text{payoff}(S)]$

provided that the expectation is with respect to the risk-neutral random walk, not the *real* one.

$\mathbb{P}(S,t) \longrightarrow (S', T)$

$$V(S,t) = e^{-r(T-t)} \times \underbrace{\frac{1}{\sigma\sqrt{2\pi(T-t)}} \int_0^\infty \exp\left[-\frac{1}{2}\left(\log\left(\frac{S}{S'}\right) + \left(r - \frac{1}{2}\sigma^2\right)(T-t)\right)^2 \Big/ \sigma^2(T-t)\right] \text{Payoff}(S') \frac{dS'}{S'}}_{\mathbb{E}^{\mathbb{Q}}[\text{Payoff}(S')]}$$

The algorithm: Monte Carlo recipe

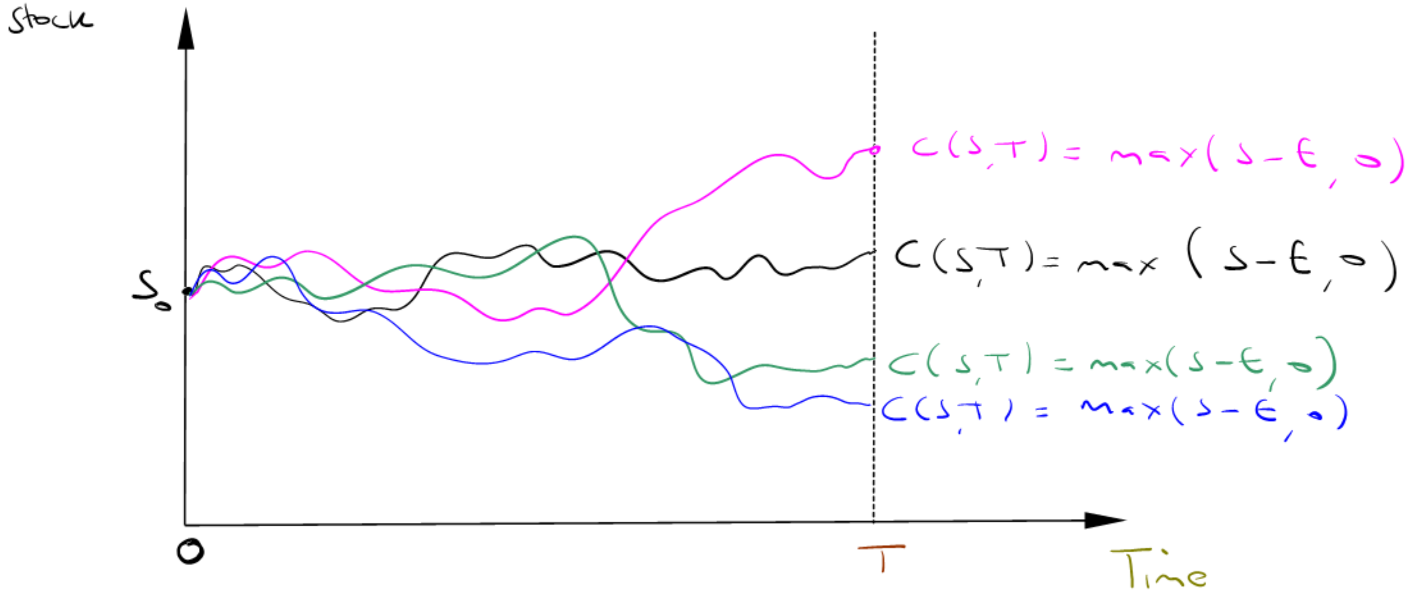
1. Simulate the risk-neutral random walk starting at today's value of the asset S_0 over the required time horizon. This gives one realization of the underlying price path.
   $S(0) = S_0$ T - time horizon $dS = rS\,dt + \sigma S\,dX$

2. For this realization calculate the option payoff, for the derivative security of interest.

3. Perform many more such realizations over the time horizon.

4. Calculate the average payoff over all realizations.

5. Take the present value of this average, this is the option value.



$$\text{Average} = \frac{1}{n}\sum_{i=1}^{n} \max(S_T^{(i)} - E, 0)$$

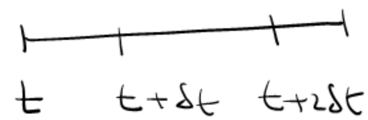
$$\text{Option value} = e^{-r(T-t)} \frac{1}{n}\sum_{i=1}^{n} \max(S_T^{(i)} - E, 0)$$

How do we simulate the asset?

Two ways:

1. **If** the s.d.e. for the asset path is integrable **and** the contract is not path dependent (or American) **then** simulate in 'one giant leap' closed form solution for S_T

2. **Otherwise** you will have to simulate time step by time step, the entire path E-M



One giant leap: A method that works in special cases

For the lognormal random walk we are lucky that we can find a simple, and exact, time stepping algorithm.

We can write the risk-neutral stochastic differential equation for S in the form Itô on log S (Module 1 LS)

$$d(\log S) = \left(r - \tfrac{1}{2}\sigma^2\right) dt + \sigma\, dX.$$

This can be integrated exactly to give

$$S(t) = S(0) \exp\left(\left(r - \tfrac{1}{2}\sigma^2\right) t + \sigma \int_0^t dX\right).$$

i.e.

- $$S(T) = S(0) \exp\left(\left(\underbrace{r - \tfrac{1}{2}\sigma^2}_{\text{drift}}\right) T + \underbrace{\sigma}_{\text{vol}}\sqrt{T}\,\phi\right).$$

$\phi \sim N(0,1)$)

Because this expression is exact and simple it is the best time stepping algorithm to use. . . but only if we have a payoff that only depends on the final asset value, i.e. is European and path independent.

We can then simulate the final asset price in one giant leap, using a time step of T if both of these are true

stock Model

- the s.d.e. is integrable and → Solve by hand. $r, \sigma \in \mathbb{R}$

Contract feature

- the contract is European and not path dependent ϕ

Simulating the entire path: A method that always works

Price paths are simulated using a discrete version of the stochastic differential equation for S.

An obvious choice is to use

$dS = rS\,dt + \sigma S\,dX$ cts time

$$\delta S = rS\,\delta t + \sigma S\sqrt{\delta t}\,\phi, \qquad \text{discrete time}$$

$S_{i+1} = S_i\left(1 + r\delta t + \sigma\sqrt{\delta t}\,\phi\right)$ E-M solution

where ϕ is from a standardized Normal distribution.

- This way of simulating the time series is called the **Euler method**. This method has an error of $O(\delta t^{1/2})$.

Milstein $O(\delta t)$ $S_{t+\delta t} = S_t\, e^{(r-\frac{1}{2}\sigma^2)\delta t + \sigma\phi\sqrt{\delta t}}$

Errors

There are two (at least) sources of error in the Monte Carlo method:

- If the size of the time step is δt then we may introduce errors of $O(\delta t)$ by virtue of the discrete approximation to continuous events

- Because we are only simulating a finite number of an infinite number of possible paths, the error due to using N realizations of the asset price paths is $O(N^{-1/2})$. $\frac{1}{\sqrt{N}}$

Generating Normal variables

Recap M1 L5

- **Quick 'n' dirty**: A useful distribution that is easy to implement on a spreadsheet, and is fast, is the following *approximation* to the Normal distribution:

RAND() Excel

$\psi_i \sim U(0,1)$

rand() C++

$$\left(\sum_{i=1}^{12} \psi_i\right) - 6,$$

$\sqrt{\frac{12}{n}}\left[\sum \psi_i - \frac{n}{2}\right]$

where the ψ_i are independent random variables, drawn from a uniform distribution over zero to one.

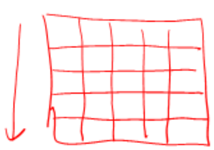
There are other methods such as **Box–Muller**, more later.

Accuracy and computational time

Let's use ϵ to represent the desired accuracy in a MC calculation.

We know that errors are $\boxed{O(\delta t)}$ and $\boxed{O(1/\sqrt{N})}$. It makes sense to have errors due to the time step and to the finite number of simulations to be of the same order (no point in having one link in a chain stronger than another!). So we would choose:

N

$\frac{1}{\delta t}$  $\delta t = \frac{1}{T}$ $T = \frac{1}{\delta t}$

$$\delta t = O(\epsilon) \quad \text{and} \quad N = O(\epsilon^{-2}).$$

$\varepsilon \sim \frac{1}{\sqrt{N}}$

$8 \; \frac{1}{\delta t} = O(\varepsilon^{-1})$

The time taken is then proportional to number of calculations, therefore

$N \times \frac{1}{\delta t}$

$$8 \quad \text{Time taken} = O(\epsilon^{-3}).$$

If you want to halve the error it will take eight times as long.

$\left(\frac{1}{2}\right)^3$

In higher dimensions. . .

Suppose you have a basket option with D underlyings. The time taken now becomes

$$\text{Time taken} = O(D\epsilon^{-3}).$$

(Think of having one Excel spreadsheet per asset.)

This is surprisingly insensitive to dimension!

Other issues

- Greeks

$$\Delta = \frac{\partial V}{\partial S} = \lim \frac{V(S+\delta S) - V(S)}{\delta S}$$

- Early exercise (and other embedded decisions).

Advantages of Monte Carlo simulations

- The mathematics that you need to perform a Monte Carlo simulation can be very basic

- Correlations can be easily modeled, and it is easy to price options on many assets (high-dimensional contracts)

  $\frac{dS_i}{S_i} = r\,\delta t + \sigma\, dX^{(i)}$ $\quad \mathbb{E}\left[dX^{(p)} dX^{(q)} \right] = \rho\, \delta t$

- It is computationally quite efficient in high dimensions $\text{Error} \sim \frac{1}{\sqrt{n}}$

- There is plenty of software available, at the very least there are spreadsheet functions that will suffice for most of the time Monte-Carlo pricing engine,

- To get a better accuracy, just run more simulations

  as $n \longrightarrow \infty$ $\quad \text{Error} \longrightarrow 0$

- The effort in getting *some* answer is very low

- The models can often be changed without much work flexible

Exotics
- Complex path dependency can often be easily incorporated

- Many contracts can be priced at the same time

- People accept the technique, and will believe your answers

Disadvantages of Monte Carlo simulations

- The method is very slow, you need a lot of simulations to get an accurate answer In low dim^ns M-C bad

- Finding the greeks can be hard – extra work

- The method does not cope well with early exercise/embedded decisions – avoid

Finite Difference Methods

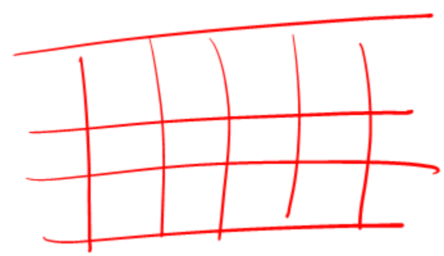
Explicit scheme

Monte Carlo simulations can be very slow to converge to the answer, and they do not give us the greeks without further effort.

There is a method that is very similar to the binomial tree method which is the method of choice for certain types of problem.

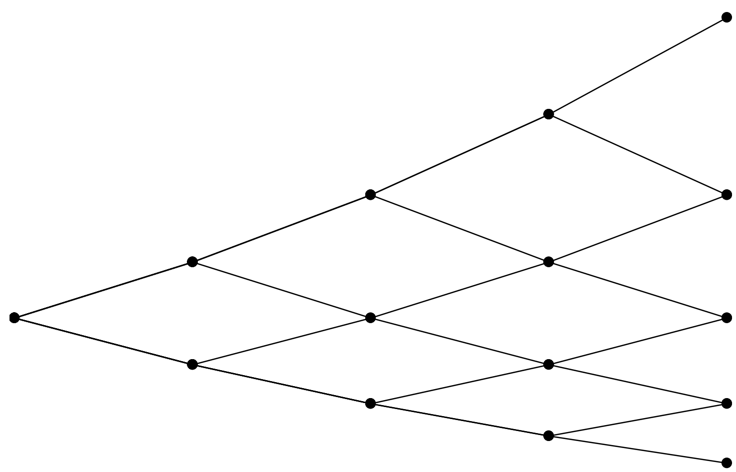
(1) Create a grid

(2) discretise the pde (T.J E.)



Grids

Recall the shape of the binomial tree...

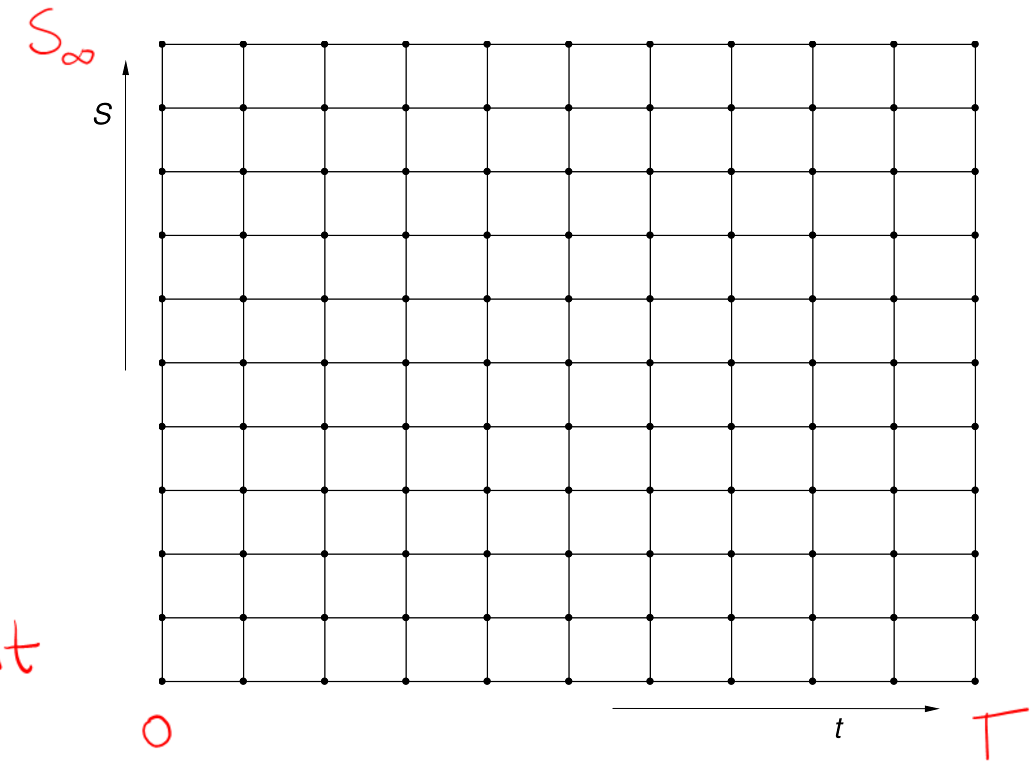


The shape of the tree is determined by the asset volatility.

$V(S, t)$

$S := i\,\delta S$

$t := T - k\,\delta t$



$\delta S = \dfrac{S_\infty}{I}$ $\qquad$ $\delta t = \dfrac{T}{K}$

The finite-difference grid.

The finite-difference grid usually has equal time steps and equal S steps.

Differentiation using the grid

Notation: time step δt and asset step δS. The grid is made up of the points at asset values

$\delta S, \quad \delta t$

$$S = i\,\delta S$$

$$\frac{\partial V}{\partial t} \qquad \frac{\partial V}{\partial S} \qquad \frac{\partial^2 V}{\partial S^2}$$

and times

$t := k\,\delta t$

$$\tau = t = T - k\,\delta t$$

Time to expiry

where $0 \leq i \leq I$ and $0 \leq k \leq K$.

We will be solving for the asset value going from zero up to the asset value $I\,\delta S$.

The Black–Scholes equation is to be solved for $0 \leq S < \infty$ so that $I\,\delta S$ is our approximation to infinity.

Write the option value at each of these grid points as

$$V(S,t) := \quad V_i^k = V(\underbrace{i\,\delta S}_{S}, \underbrace{T - k\,\delta t}_{t}).$$

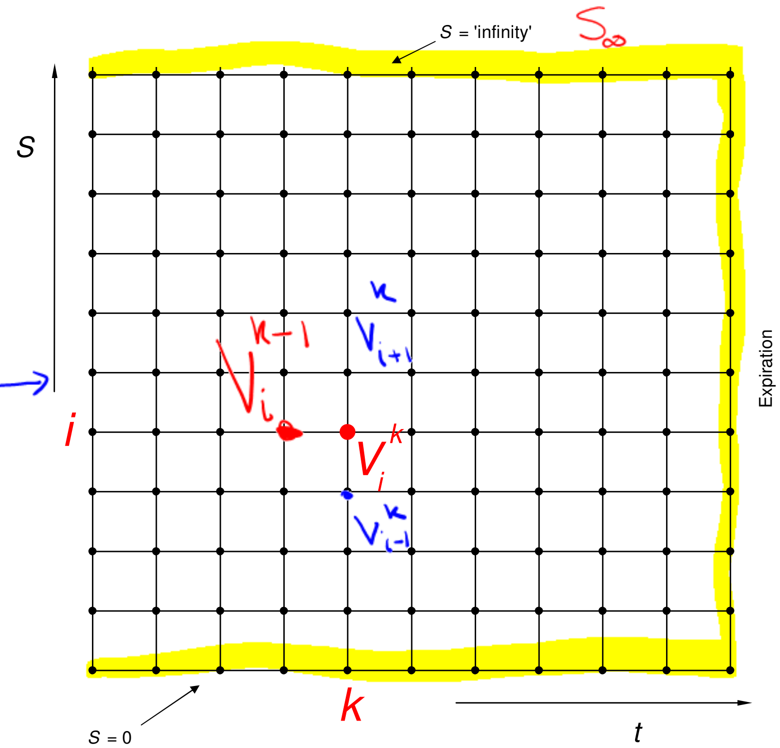
V^k ← Time on its own

i ← asset price(s)

- The superscript is the time variable and the subscript the asset variable.

$V(S,t)$
$\quad\uparrow_0$

This boundary → represents $V(S,0)$



3 boundaries along which option price is known

Payoff

$k \pm 1 \qquad t \pm \delta t$

$i \pm 1 \qquad S \pm \delta S$

Approximating θ

$\tau = T - t$

The definition of the first time derivative of V is simply

V^{k+1} = - - - - fwd

$$\frac{\partial V}{\partial t} = \lim_{h \to 0} \frac{V(S,t) - V(S,t-h)}{h}.$$

Backward time diff.

V^{k-1} = - - - - bwd.

$\frac{\partial V}{\partial t} := -\frac{\partial V}{\partial \tau}$

It follows naturally that we can approximate the time derivative from our grid of values using

forward marching

$k+1$ $\quad$ k $\quad$ $*$

$$\boxed{\frac{\partial V}{\partial t}(S,t) \approx \frac{V_i^k - V_i^{k+1}}{\delta t}.}$$

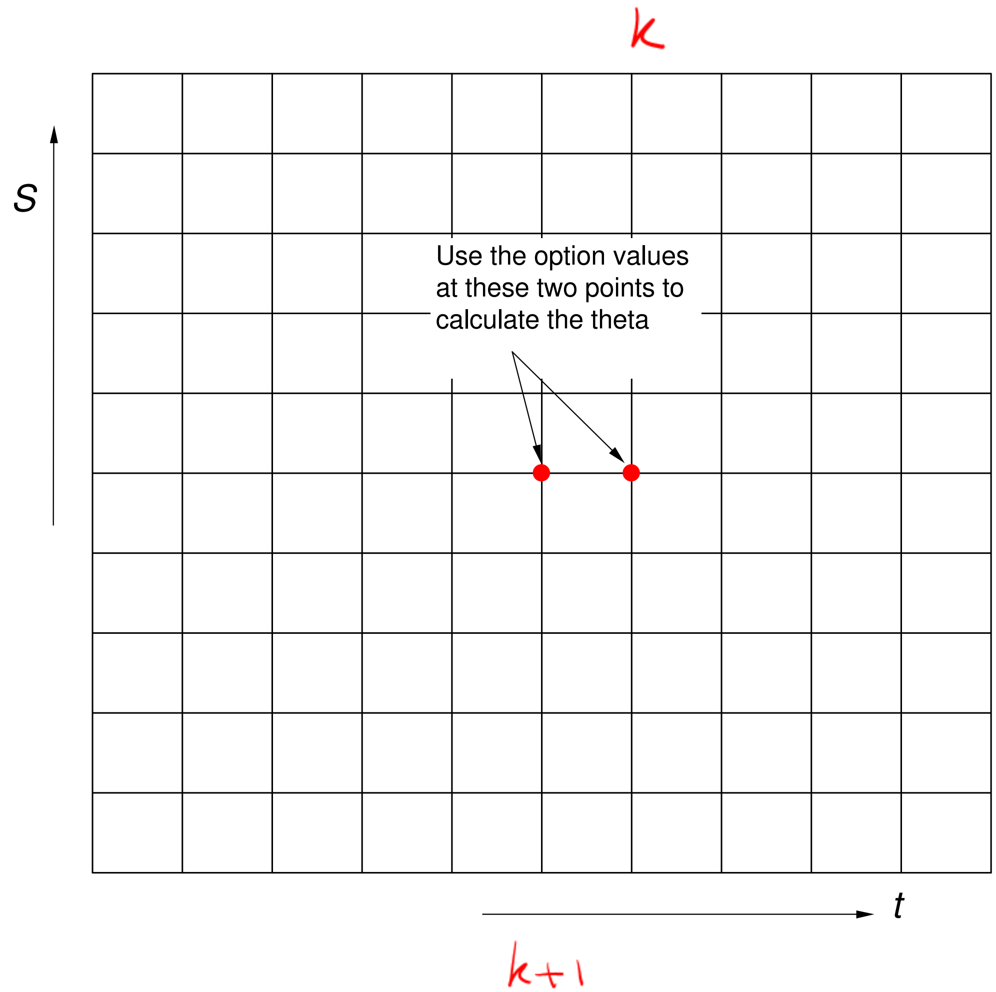
$\tau = T - t$

$* \; 0 \longrightarrow T$

This is our approximation to the option's theta.

$k-1$ $\quad$ k $\quad$ Backward marching

$** \; 0 \longleftarrow T$



Approximating the theta. $\frac{\partial V}{\partial t}$

How accurate is this approximation?

We can expand the option value at asset value S and time $t - \delta t$ in a Taylor series about the point S, t as follows.

$$V(S, t - \delta t) = V(S, t) - \delta t \frac{\partial V}{\partial t}(S, t) + O(\delta t^2).$$

$k-1$ $\quad$ k. $\quad$ $\rightarrow O(\delta t)$

In terms of values at grid points this is just

$$V_i^k = V_i^{k+1} + \delta t \frac{\partial V}{\partial t}(S, t) + O(\delta t^2).$$

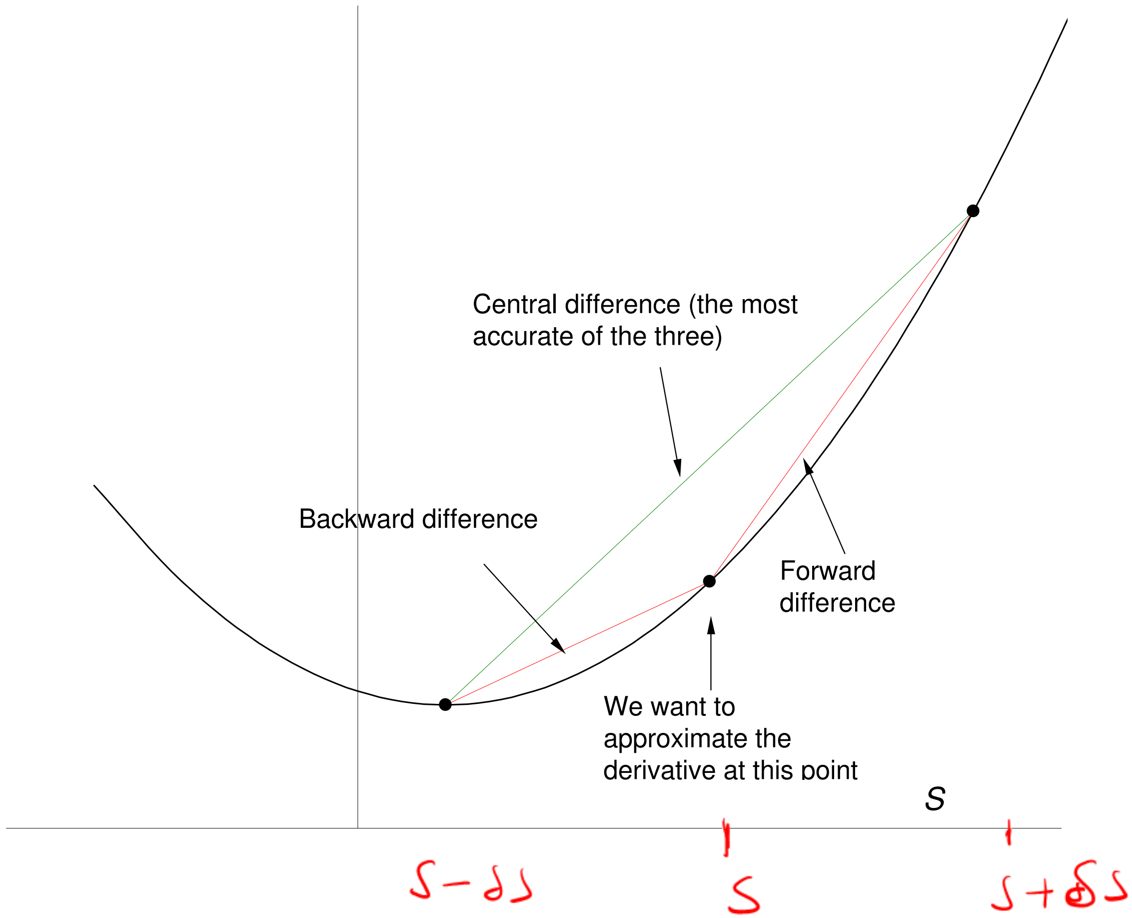
Which, upon rearranging, is

$$\Theta = \frac{\partial V}{\partial t}(S, t) = \frac{V_i^k - V_i^{k+1}}{\delta t} + \underline{O(\delta t)}.$$

- The error is $O(\delta t)$.

Approximating Δ $= \frac{\partial V}{\partial S}$

Examine a cross section of the grid at one of the time steps.



These three approximations are

$$\frac{V_{i+1}^k - V_i^k}{\delta S}, \qquad \text{fwd}$$

$$\frac{V_i^k - V_{i-1}^k}{\delta S} \qquad \text{bwd}$$

$$\text{and} \quad \frac{V_{i+1}^k - V_{i-1}^k}{2\,\delta S}. \qquad \text{centred } \checkmark\checkmark$$

These are called a **forward difference**, a **backward difference** and a **central difference** respectively.

One of these approximations is better than the others.

From a Taylor series expansion of the option value about the point $S + \delta S$, t we have

$$V(S + \delta S, t) = V(S, t) + \delta S \frac{\partial V}{\partial S}(S, t) + \tfrac{1}{2}\delta S^2 \frac{\partial^2 V}{\partial S^2}(S, t) + O(\delta S^3).$$

Similarly,

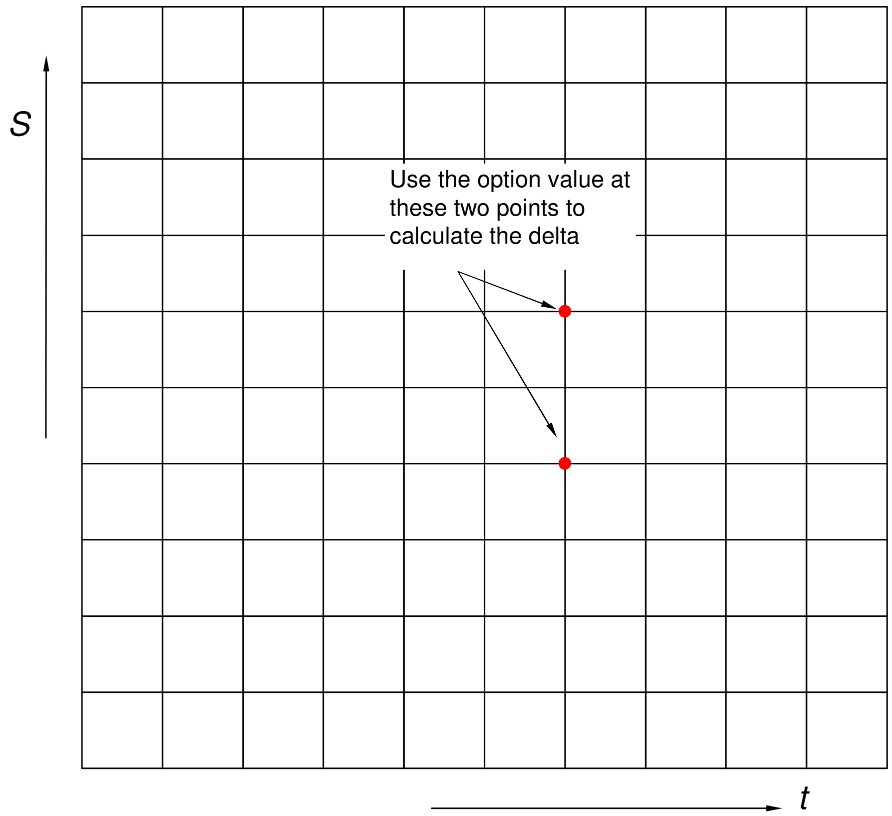
$$V(S - \delta S, t) = V(S, t) - \delta S \frac{\partial V}{\partial S}(S, t) + \tfrac{1}{2}\delta S^2 \frac{\partial^2 V}{\partial S^2}(S, t) + O(\delta S^3).$$

From these we get

$$\frac{\partial V}{\partial S}(S,t) = \frac{V_{i+1}^k - V_{i-1}^k}{2\,\delta S} + O(\delta S^2).$$

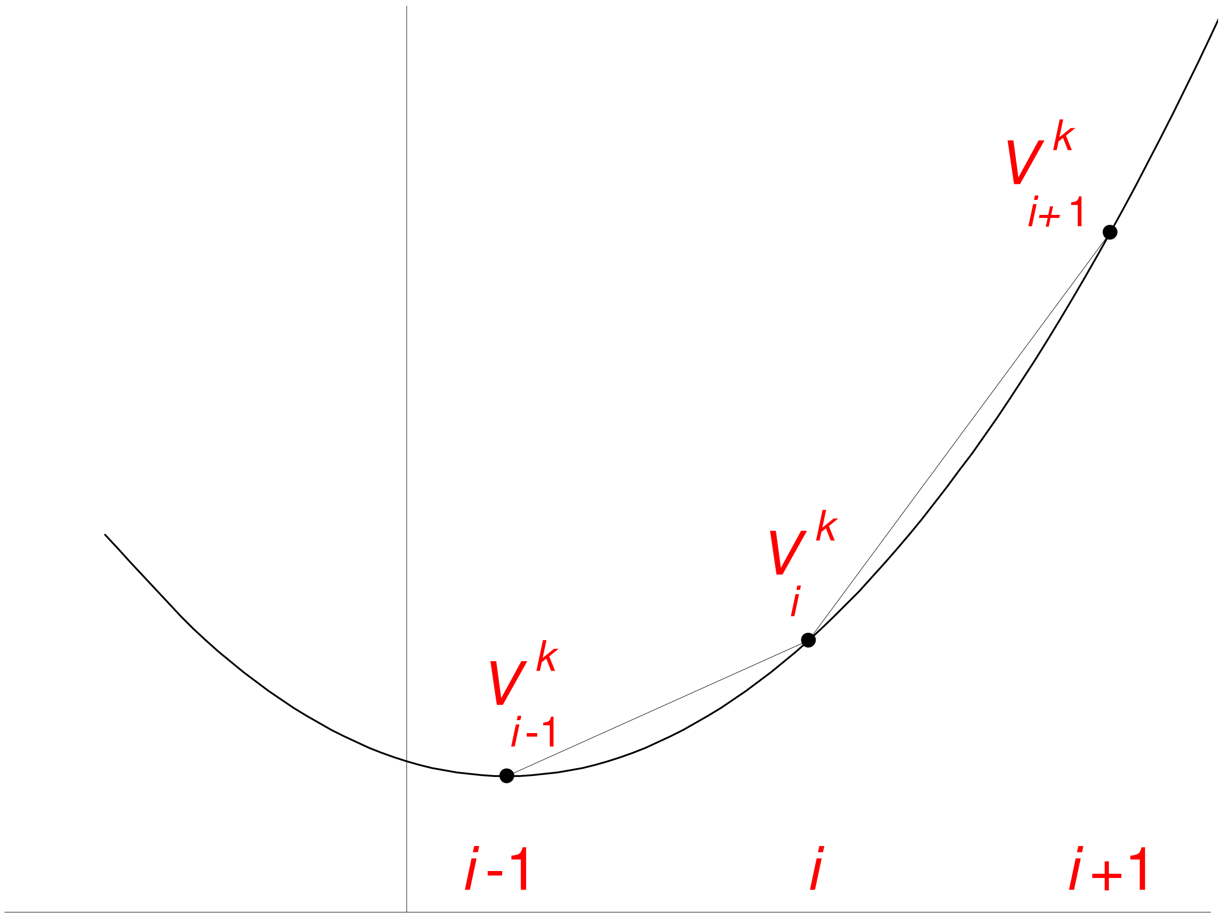
- The central difference has an error of $O(\delta S^2)$, the error in the forward and backward differences are both much larger, $O(\delta S)$.

The central difference calculated at S requires knowledge of the option value at $S + \delta S$ and $S - \delta S$.


S
Use the option value at
these two points to
calculate the delta
t

Approximating Γ

Gamma is the sensitivity of the delta to the underlying.



Calculate the delta half way between i and $i+1$, and the delta half way between $i-1$ and i . . . and difference them!

$$\text{Forward difference } = \frac{V_{i+1}^k - V_i^k}{\delta S}.$$

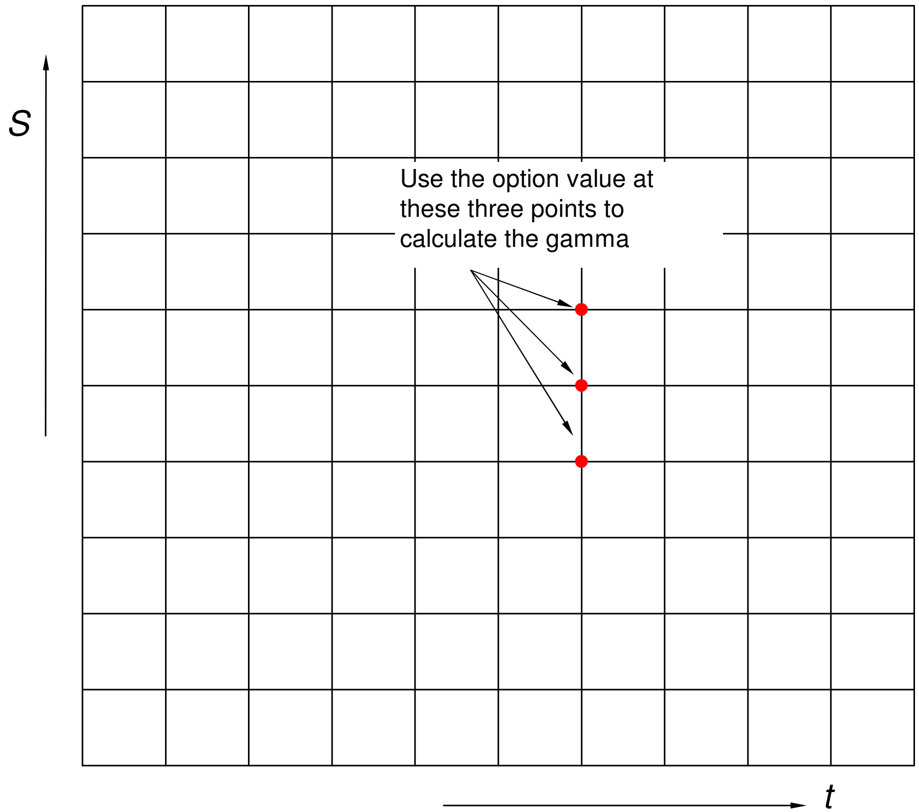
$$\text{Backward difference } = \frac{V_i^k - V_{i-1}^k}{\delta S}.$$

Therefore the natural approximation for the gamma is

$$\frac{\partial^2 V}{\partial S^2}(S,t) \approx \frac{\frac{V_{i+1}^k - V_i^k}{\delta S} - \frac{V_i^k - V_{i-1}^k}{\delta S}}{\delta S}$$

$$= \frac{V_{i+1}^k - 2V_i^k + V_{i-1}^k}{\delta S^2}.$$

The error in this approximation is also $O(\delta S^2)$.



S
Use the option value at
these three points to
calculate the gamma
t

Write 2 T.S.E $\quad V(S+\delta S, t)$; $V(S-\delta S, t)$

$$V(S+\delta S, t) = V(S, t) + \frac{\partial V}{\partial S}\delta S + \frac{1}{2}\frac{\partial^2 V}{\partial S^2}\delta S^2 + \frac{1}{3!}\frac{\partial^3 V}{\partial S^3}\delta S^3 + O(\delta S^4) \qquad (1)$$

$$V(S-\delta S, t) = V(S, t) - \frac{\partial V}{\partial S}\delta S + \frac{1}{2}\frac{\partial^2 V}{\partial S^2}\delta S^2 - \frac{1}{3!}\frac{\partial^3 V}{\partial S^3}\delta S^3 + O(\delta S^4) \qquad (2)$$

(1) − (2) $\quad V(S+\delta S, t) - V(S-\delta S, t) = 2\frac{\partial V}{\partial S}\delta S + O(\delta S^3)$

÷ thro everything by δS

Rearrange: $\dfrac{V(S+\delta S, t) - V(S-\delta S, t)}{2\delta S} = \dfrac{\partial V}{\partial S} + O(\delta S^2)$

i.e $\quad \dfrac{\partial V}{\partial S} = \dfrac{V(S+\delta S, t) - V(S-\delta S, t)}{2\delta S} + O(\delta S^2)$

i.e $\quad \dfrac{\partial V}{\partial S} \sim \dfrac{V^K_{i+1} - V^K_{i-1}}{2\delta S}$ $\quad$ i.e approx^{n} for Δ

Now (1) + (2)

$$V(S+\delta S, t) + V(S-\delta S, t) = 2V(S, t) + \frac{\partial^2 V}{\partial S^2}\delta S^2 + O(\delta S^4)$$

rearrange and divide thro' "everything" by δS^2

$$\frac{\partial^2 V}{\partial S^2} = \frac{V(S-\delta S, t) - 2V(S, t) + V(S+\delta S, t)}{\delta S^2} + O(\delta S^2)$$

i.e $$\frac{\partial^2 V}{\partial S^2} \sim \frac{V_{i-1}^k - 2V_i^k + V_{i+1}^k}{\delta S^2}$$ i.e Γ

To summarise

$$\boxed{\begin{aligned} \frac{\partial V}{\partial t} &\sim \frac{V_i^k - V_i^{k+1}}{\delta t} \\ \frac{\partial V}{\partial S} &\sim \frac{V_{i+1}^k - V_{i-1}^k}{2\delta S} \\ \frac{\partial^2 V}{\partial S^2} &\sim \frac{V_{i-1}^k - 2V_i^k + V_{i+1}^k}{\delta S^2} \\ V &\sim V_i^k \end{aligned}}$$

Which replace all the terms in the B.S.E

Final conditions and payoffs

We know that at expiry the option value is just the payoff function. At expiry we have

$$V(S,T) = \text{Payoff}(S)$$

or, in our finite-difference notation,

V_i^k

$T - k\,\delta t$

$$V_i^0 = \text{Payoff}(i\,\delta S).$$

when $k=0$ time $=T$

The right-hand side is a known function.

For example, if we are pricing a call option we have

$$V_i^0 = \max(i\,\delta S - E, 0).$$

This final condition will get our finite-difference scheme started.

The explicit finite-difference method

The Black–Scholes equation is

$$\frac{\partial V}{\partial t} + \tfrac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0.$$

Write this as

$$\frac{\partial V}{\partial t} + a(S,t)\frac{\partial^2 V}{\partial S^2} + b(S,t)\frac{\partial V}{\partial S} + c(S,t)V = 0$$

so that we can examine more general problems.

Using the above approximations

$\partial V/\partial t$ $\quad\quad\quad\quad$ $\partial^2 V/\partial S^2$ $\quad\quad\quad\quad$ $\partial V/\partial S$

unknown

$$\frac{V_i^k - V_i^{k+1}}{\delta t} + a_i^k \left(\frac{V_{i+1}^k - 2V_i^k + V_{i-1}^k}{\delta S^2} \right) + b_i^k \left(\frac{V_{i+1}^k - V_{i-1}^k}{2\delta S} \right)$$

$$+c_i^k V_i^k = O(\delta t, \delta S^2).$$

Method is 1st order accurate in time, 2nd order accurate in stock.

This can be rearranged...

$y = f(x)$

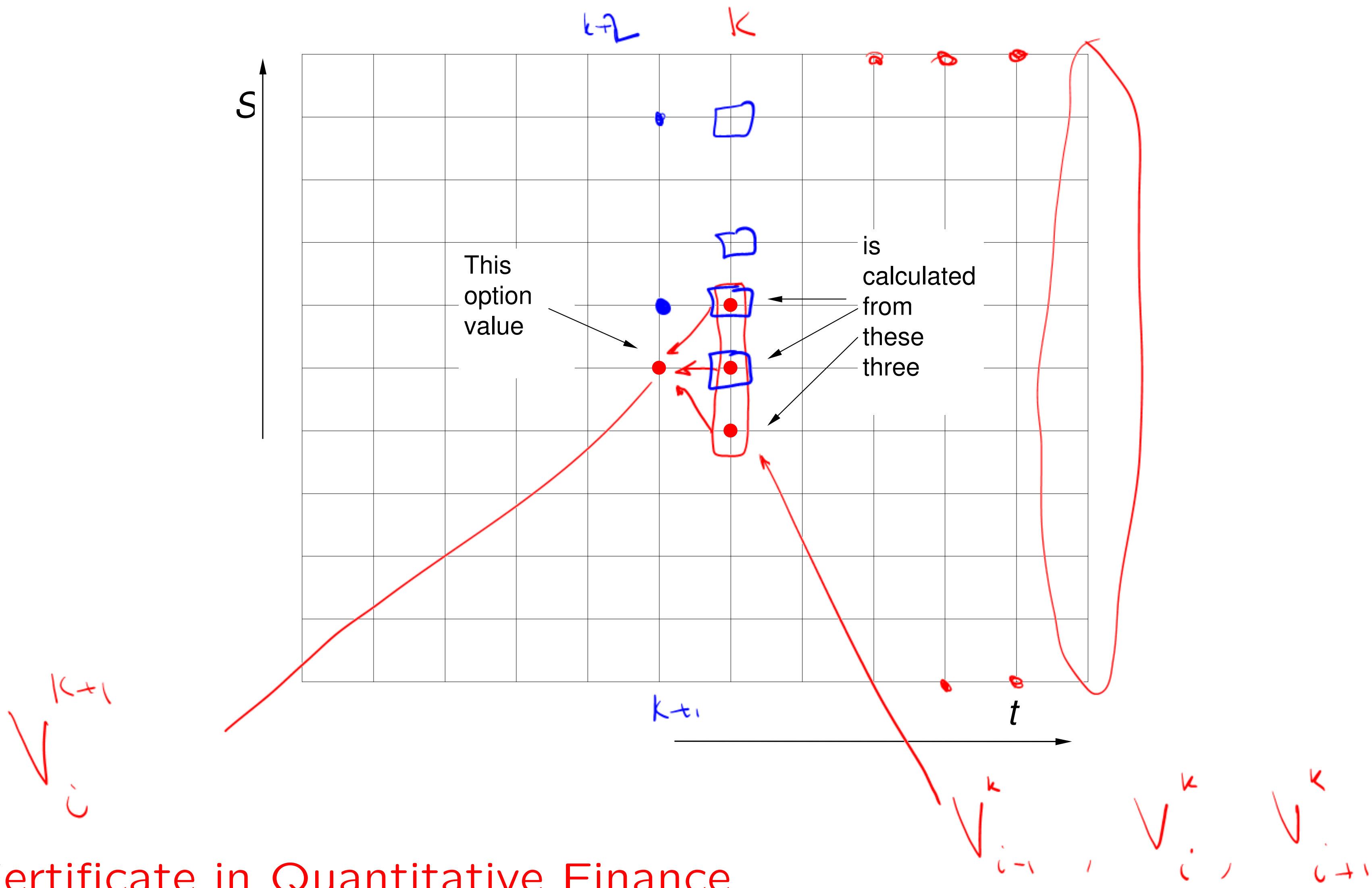
$$V_i^{k+1} = \overset{\gamma_i}{\cdots} V_{i+1}^k + \overset{\beta_i}{\cdots} V_i^k + \overset{\alpha_i}{\cdots} V_{i-1}^k.$$

$V_i^{k+1} = F(V_{i-1}^k, V_i^k, V_{i+1}^k)$ explicit

This is an equation for V_i^{k+1} given three option values at time k.

(That's why this is called the **explicit finite-difference method**.)

The relationship between the option values in the algorithm is shown in the figure below.



Points to note:

- The time derivative uses the option values at 'times' k and $k+1$, whereas the other terms all use values at k.

- The gamma term is a central difference, in practice one never uses anything else.

- The delta term uses a central difference. There are often times when a one-sided derivative is better. We'll see examples later.

- The asset- and time-dependent functions a, b and c have been valued at $S_i = i\,\delta S$ and $t = T - k\,\delta t$ with the obvious notation.

- The error in the equation is $O(\delta t, \delta S^2)$.

$$\frac{V_i^k - V_i^{k+1}}{\delta t} + \frac{1}{2}\sigma^2 i^2 \cancel{\delta S^2}\left(\frac{V_{i-1}^k - 2V_i^k + V_{i+1}^k}{\cancel{\delta S^2}}\right) + (r-D)i\cancel{\delta S}\left(\frac{V_{i+1}^k - V_{i-1}^k}{2\cancel{\delta S}}\right) - rV_i^k = 0$$

$$V_i^{k+1} = V_i^k + \frac{1}{2}\sigma^2 i^2 \delta t\left(V_{i-1}^k - 2V_i^k + V_{i+1}^k\right) + \frac{(r-D)i\,\delta t}{2}\left(V_{i+1}^k - V_{i-1}^k\right) - rV_i^k\,\delta t$$

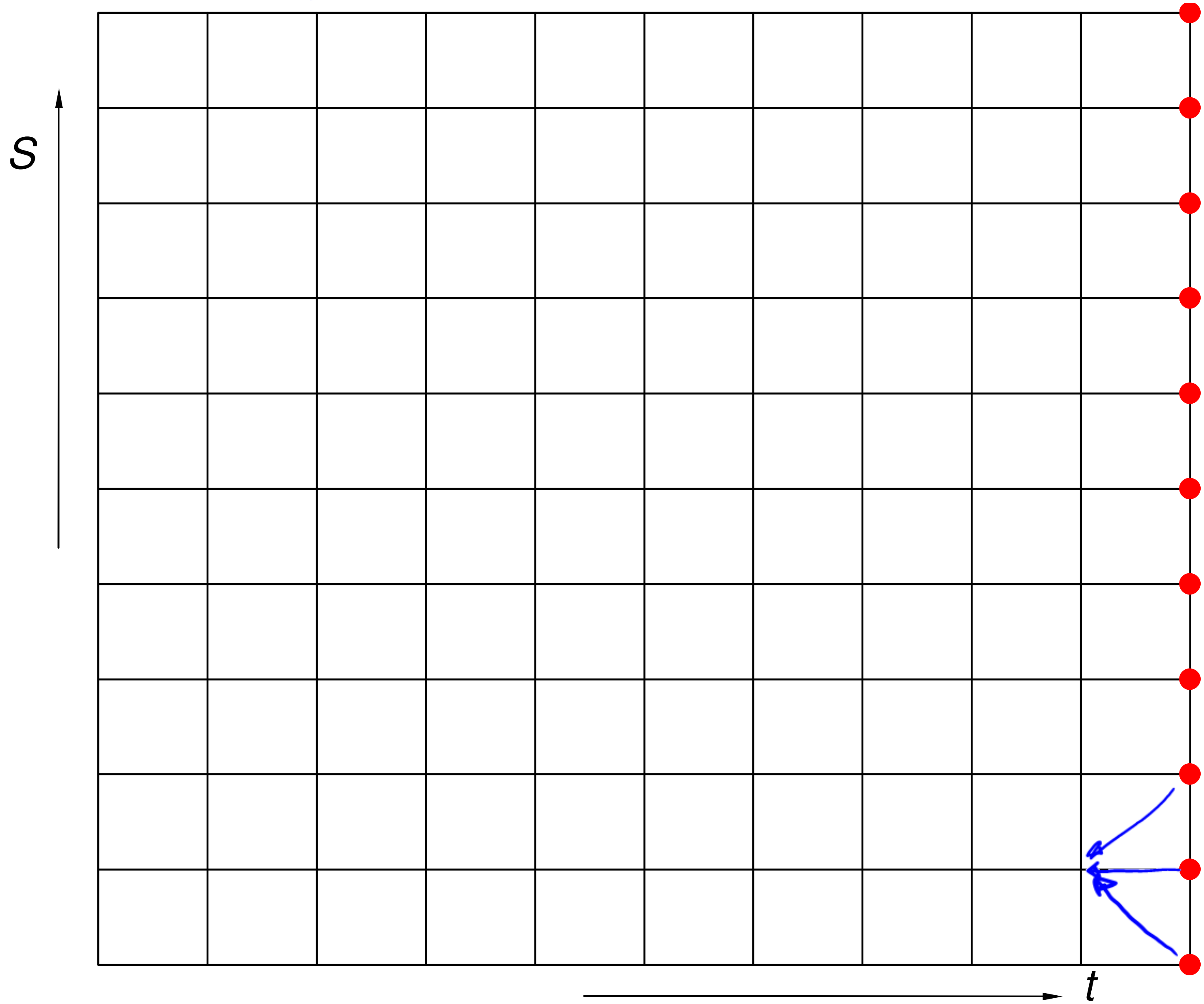
$$V_{i-1}^k : \quad \frac{1}{2}\sigma^2 i^2 \delta t - \frac{1}{2}(r-D)i\,\delta t = \frac{1}{2}\left(\sigma^2 i^2 - (r-D)i\right)\delta t \qquad \alpha_i$$

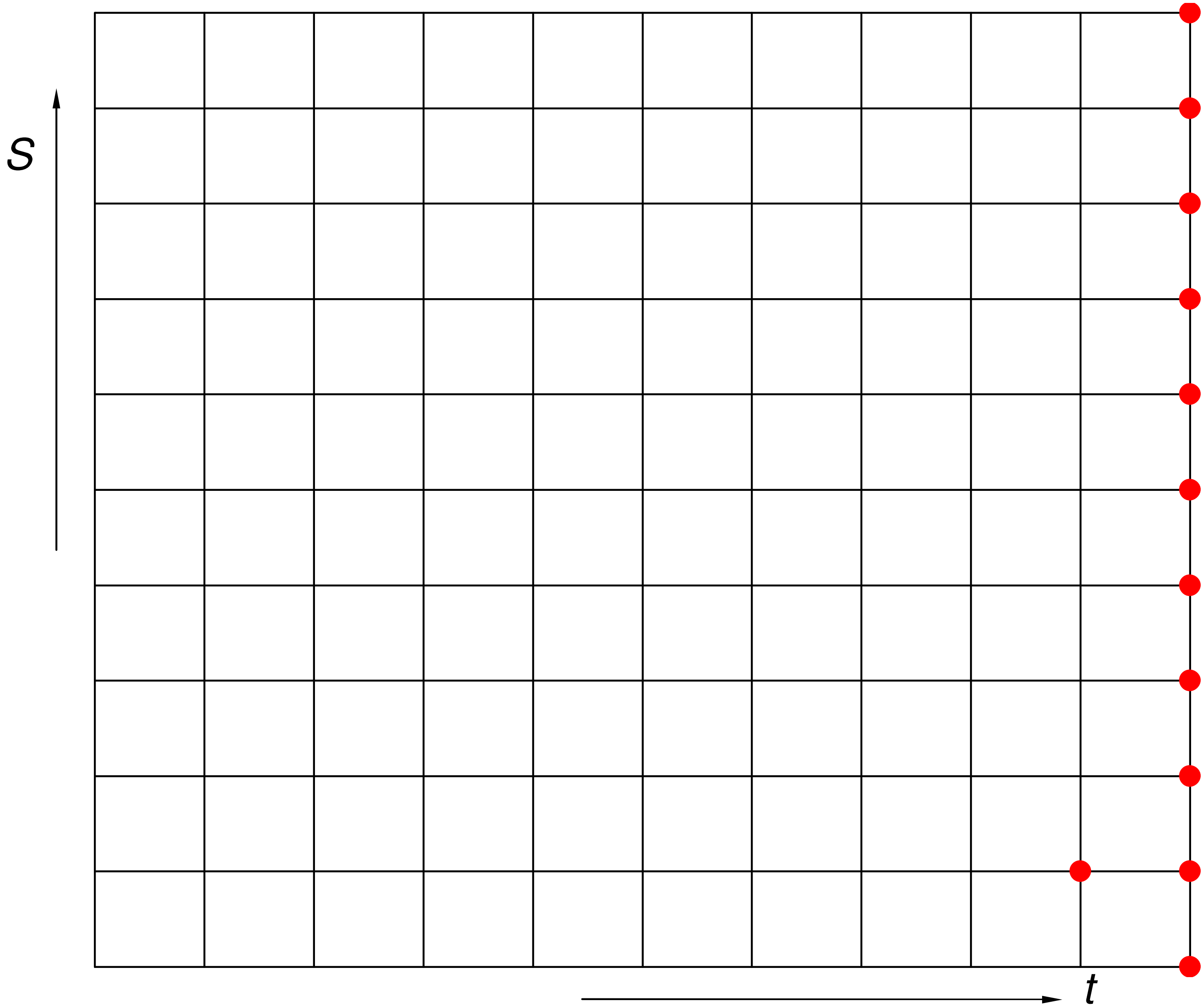
$$V_i^k : \quad 1 - \sigma^2 i^2 \delta t - r\,\delta t = 1 - (\sigma^2 i^2 + r)\,\delta t \qquad \beta_i$$

$$V_{i+1}^k : \quad \frac{1}{2}\left(\sigma^2 i^2 + (r-D)i\right)\delta t \qquad \gamma_i$$

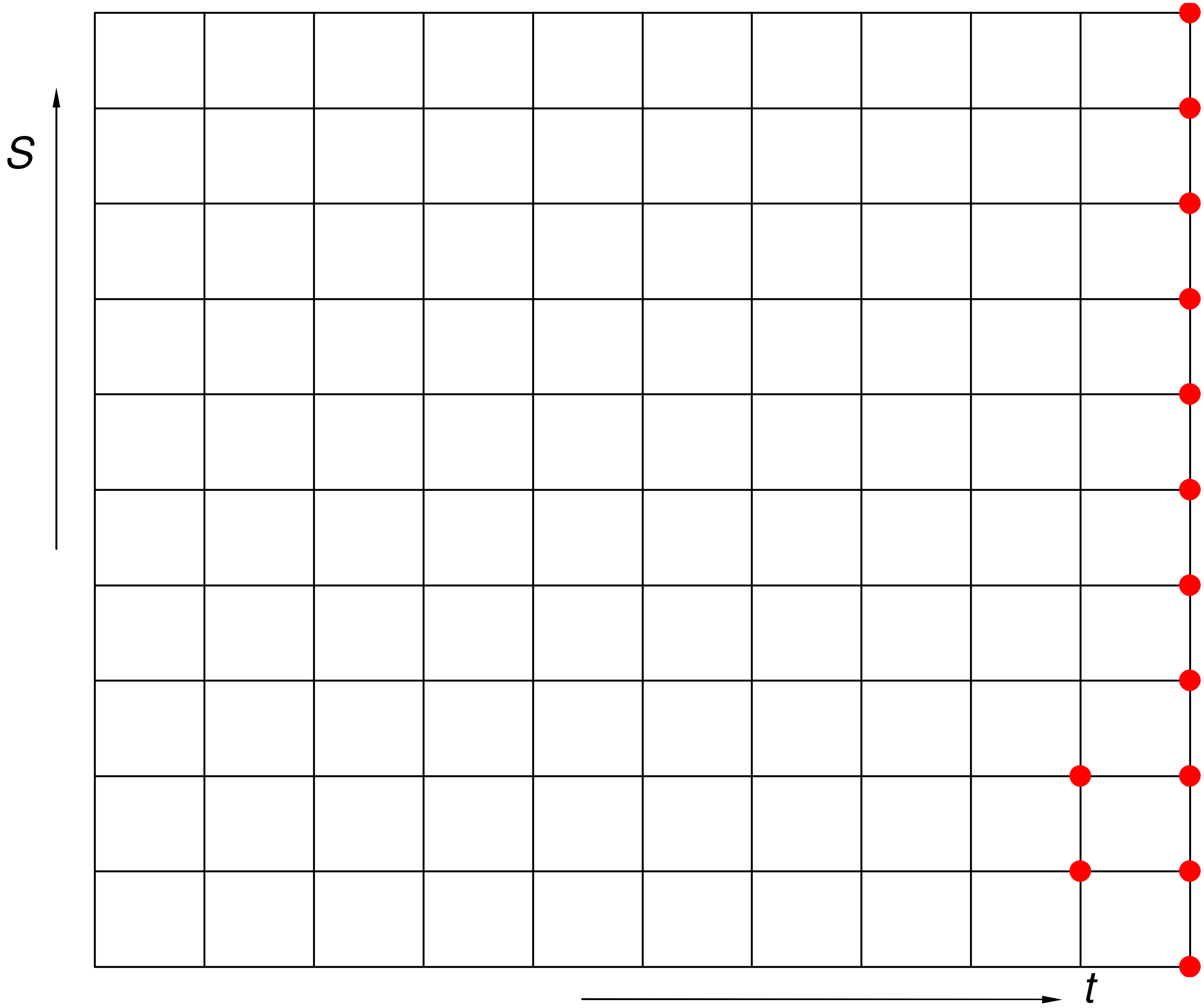
$$V_i^{k+1} = \alpha_i V_{i-1}^k + \beta_i V_i^k + \gamma_i V_{i+1}^k$$

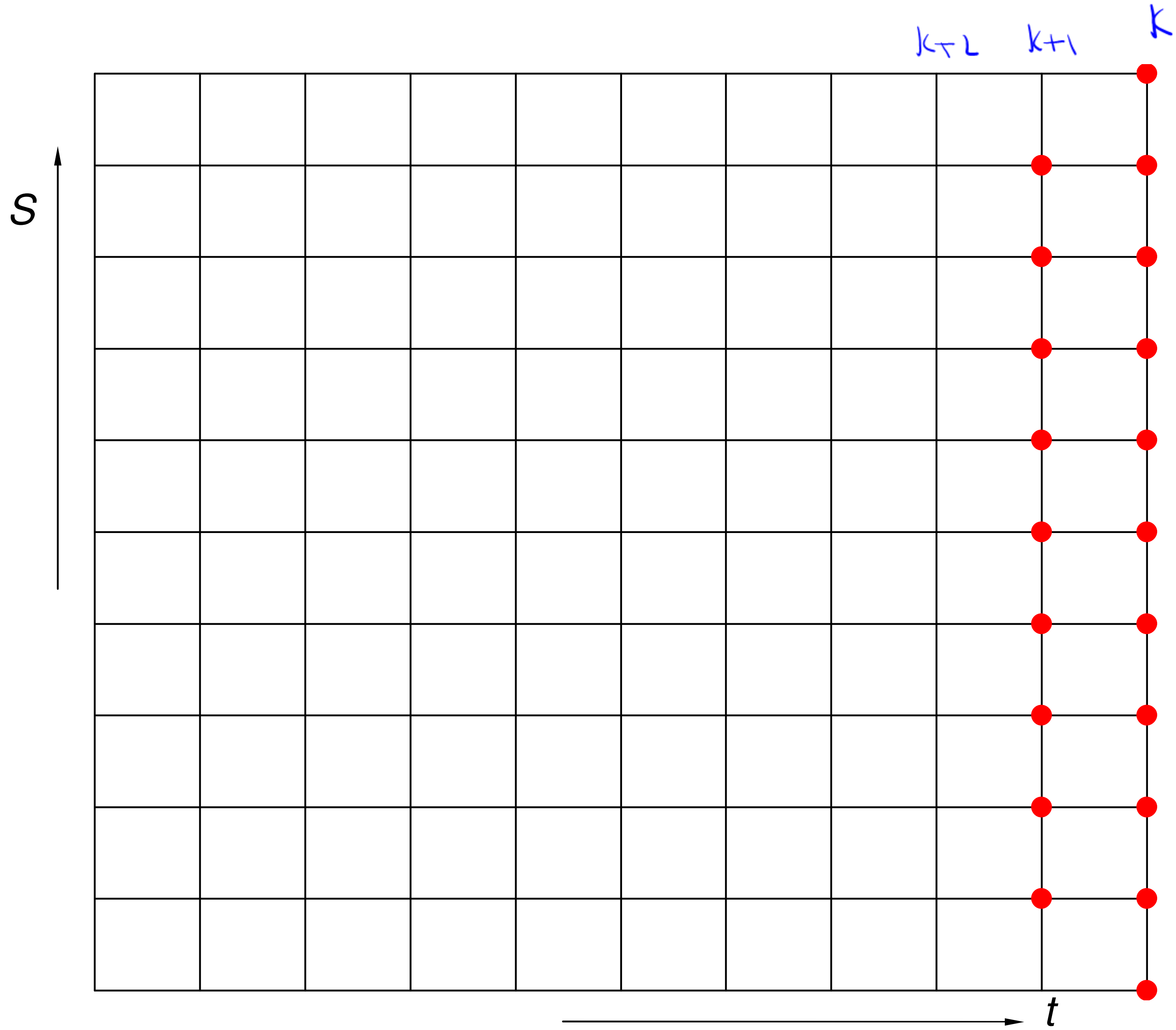
$S = 0 \quad \alpha_0 = 0 \; ; \; \gamma_0 = 0 \quad \beta_0 = 1 - r\,\delta t \quad V_0^{k+1} = (1 - r\,\delta t)\, V_0^k$

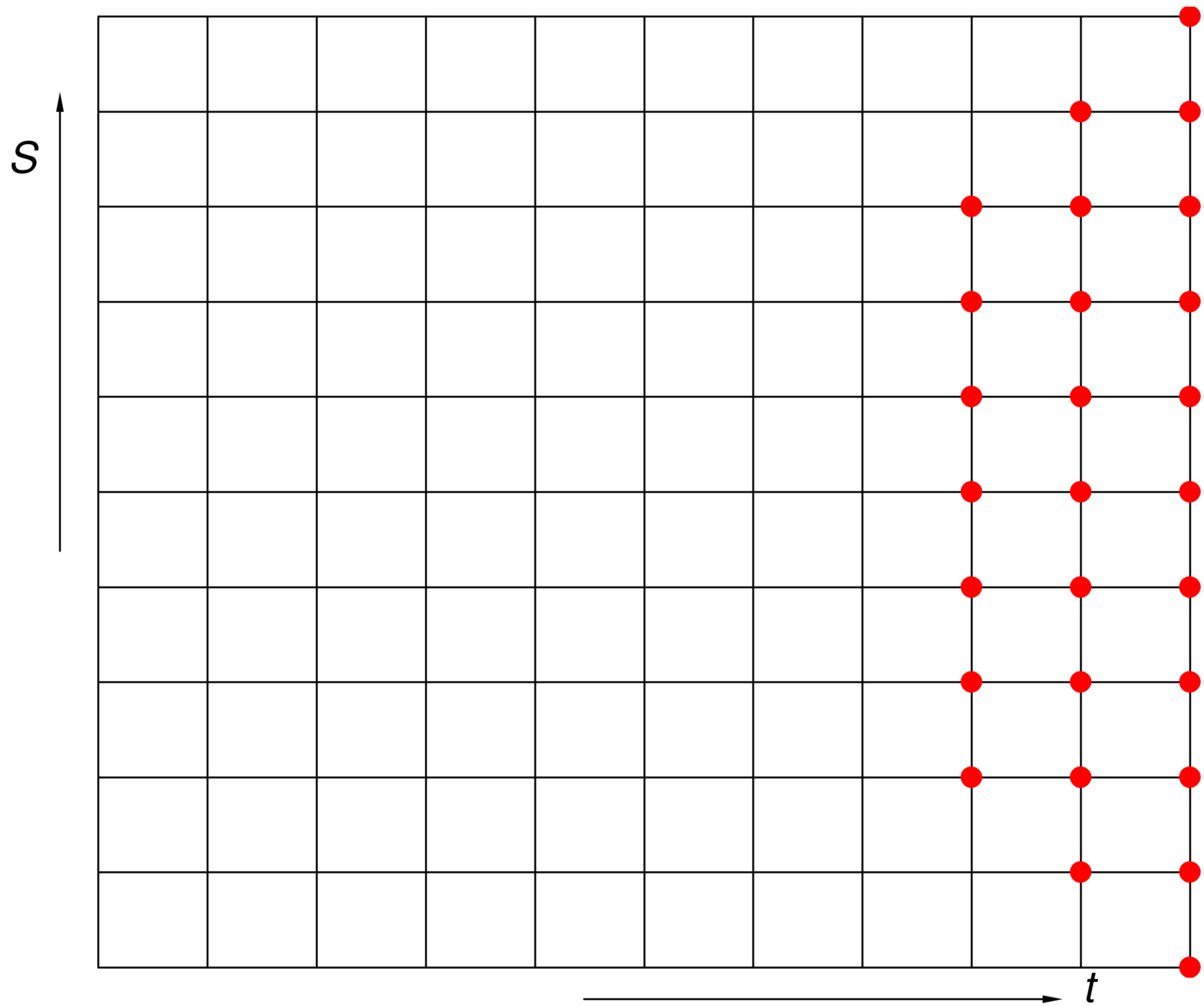


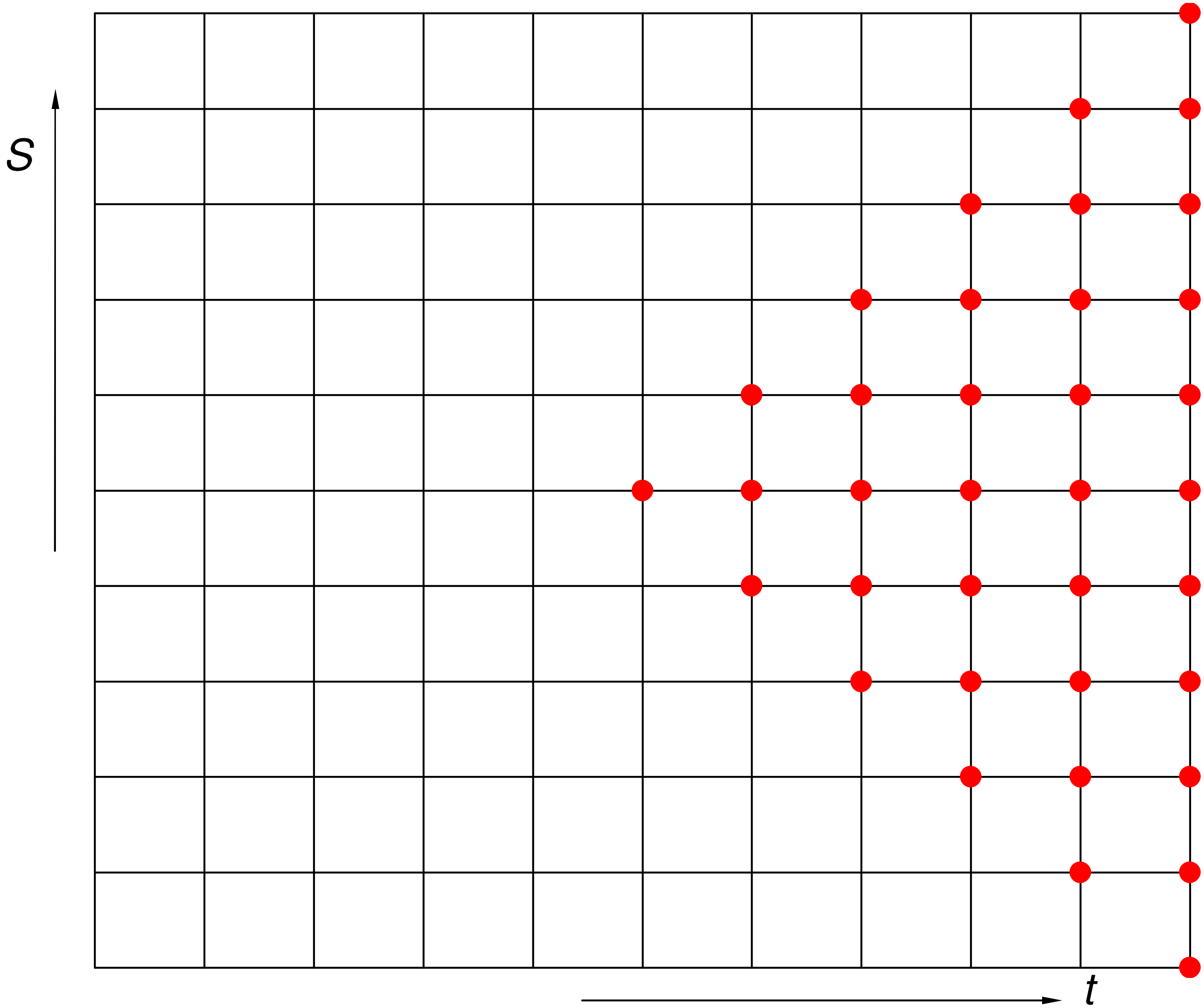


S
t


S
t


k+2
k+1
k
S
t


S
t


S
t

Boundary conditions

We must specify the option values at the extremes of the region, at $S=0$ and at $S=I\,\delta S$. They will depend on our option.

Example 1: Call option at $S=0$

At $S=0$ we know that the value is always zero, therefore

$$V_0^k = 0.$$

Example 2: Call option for large S

For large S the call value asymptotes to $S-Ee^{-r(T-t)}$. Thus

$$V_I^k = I\,\delta S - Ee^{-rk\,\delta t}.$$

Example 3: Put option at $S = 0$

At $S = 0$ $V = Ee^{-r(T-t)}$. I.e.

$$V_0^k = Ee^{-rk\,\delta t}.$$

Example 4: Put option for large S

The put option becomes worthless for large S and so

$$V_I^k = 0.$$

Example 5*: General condition at $S = 0$

A useful boundary condition to apply at $S = 0$ is that the diffusion and drift terms 'switch off.'

$$\frac{\partial V}{\partial t}(0,t) - rV(0,t) = 0 \qquad \overbrace{V_0^k = (1 - r\delta t)V_0^{k-1}}^{\text{bwd marching}}. \quad \text{formula}$$

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 \cancel{S^2}\frac{\partial^2 V}{\partial S^2} + (r-D)\cancel{S}\frac{\partial V}{\partial S} - rV = 0$$

$$\to 0 \qquad \to 0$$

$$\frac{\partial V}{\partial t} = rV \qquad \frac{V_i^k - V_i^{k+1}}{\delta t} = rV_i^k$$

when $S=0$ $\quad i=0$

$$V_0^k - V_0^{k+1} = rV_0^k\,\delta t \qquad V_0^{k+1} = (1 - r\delta t)V_0^k$$

bwd marching

Example 6*: General condition at infinity

$V_i^{k+1} = \alpha_i V_{i-1}^k + \beta_i V_i^k + \gamma_i V_{i+1}^k$

When the option has a payoff that is linear in the underlying for large S then

At $S \longrightarrow S_\infty$ $\quad i = I$ $\quad$ So $V_I^{k+1} = \alpha_I V_{I-1}^k + \beta_I V_I^k + \gamma_I V_{I+1}^k$

NOT DEFINED

Vanishing Γ :

$$\frac{\partial^2 V}{\partial S^2}(S,t) \to 0 \quad \text{as } S \to \infty.$$

The finite-difference representation is

$$\Gamma \sim \frac{V_{I-1}^k - 2V_I^k + V_{I+1}^k}{\delta S^2} \sim 0 \longrightarrow V_I^k = 2V_{I-1}^k - V_{I-2}^k.$$

Call option : $S \longrightarrow \infty$ $\quad S >> E$ $\quad S + \delta S$?

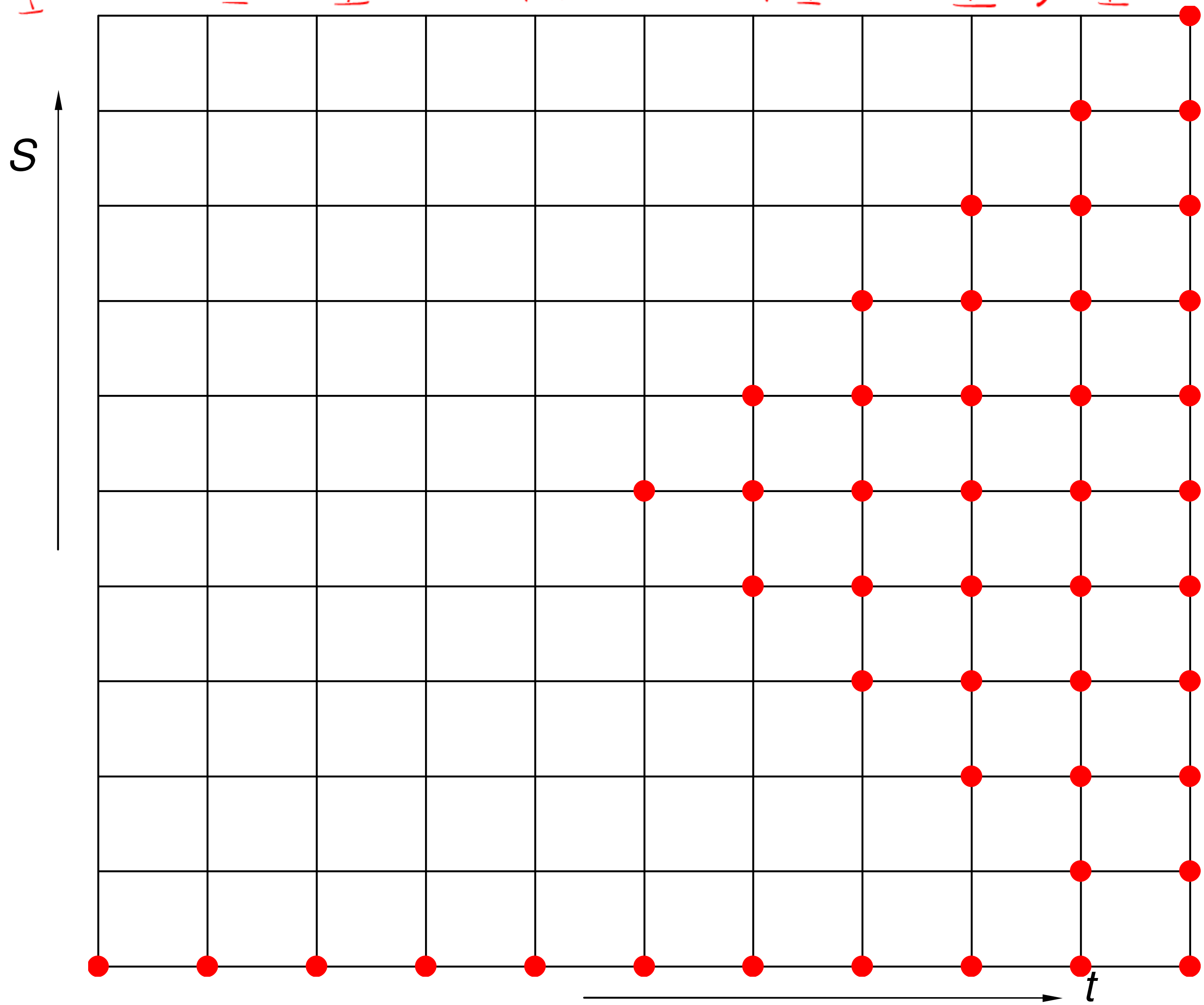
Δ is insensitive to small changes $\quad \Delta(S,t) \longrightarrow \Delta(t)$

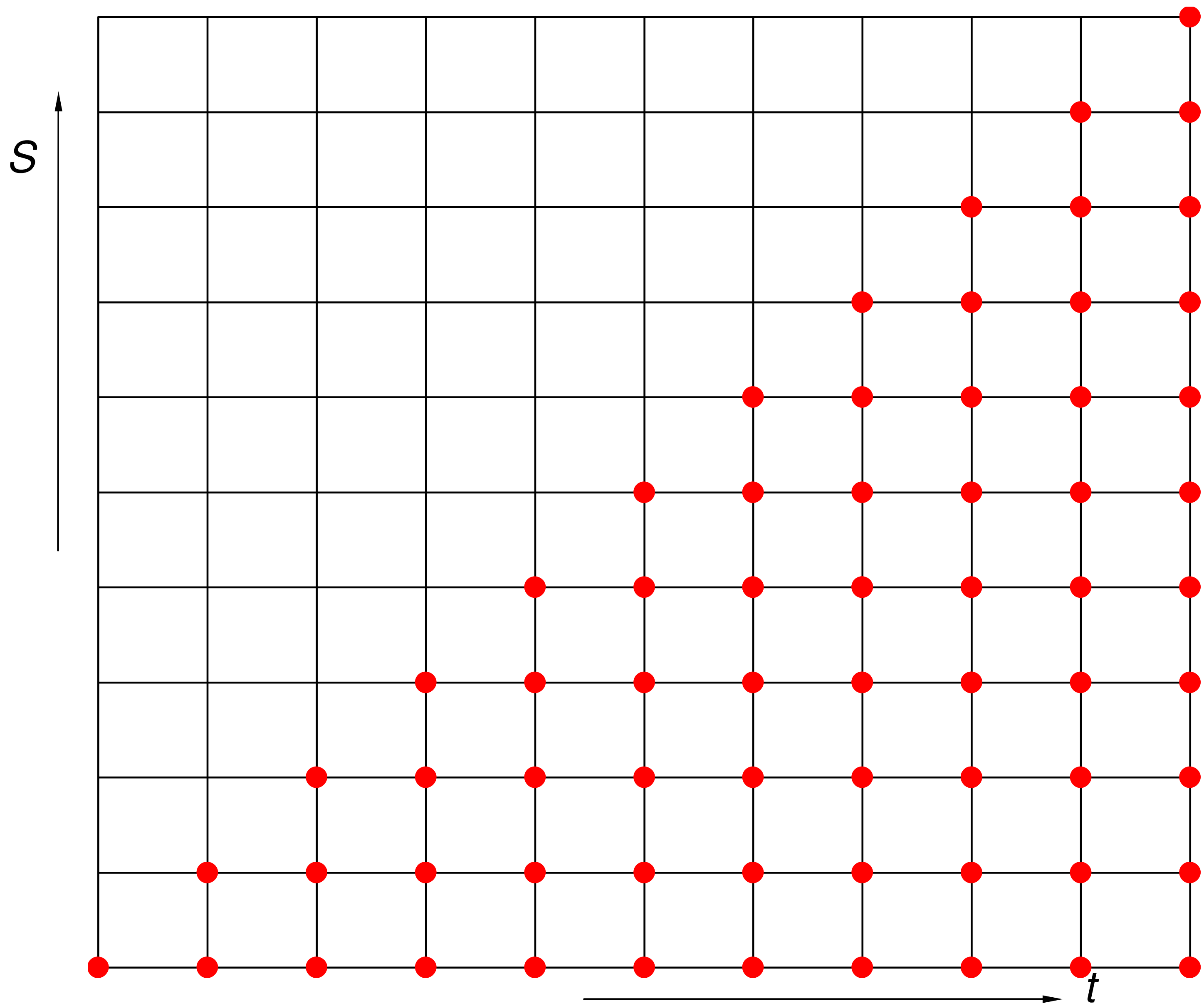
$\Gamma = \frac{\partial}{\partial S}\Delta(t) = 0$

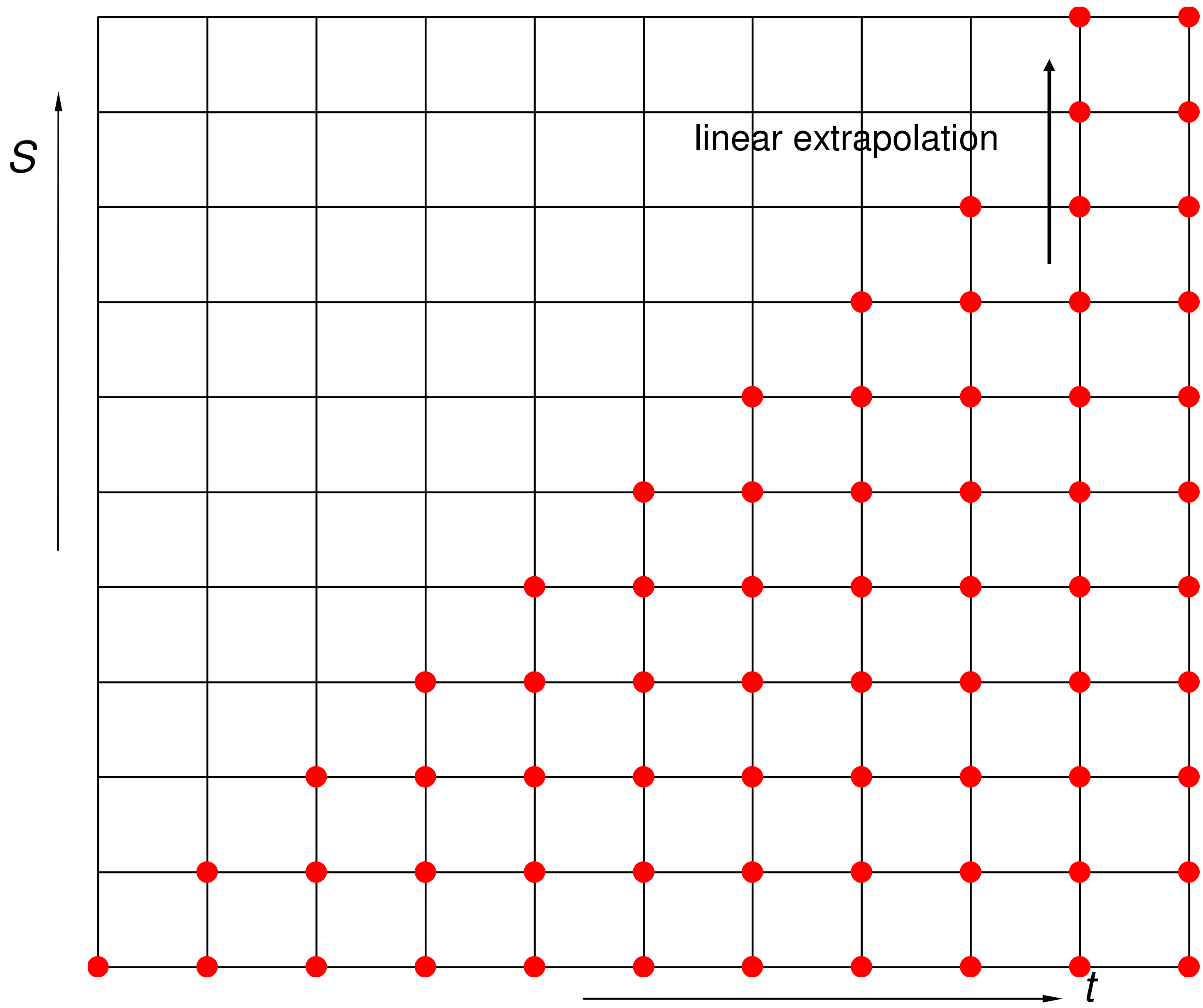
$V_{I+1}^k = 2V_I^k - V_{I-1}^k \longrightarrow V_I^{k+1} = \alpha_I V_{I-1}^k + \beta_I V_I^k + \gamma_I\left(2V_I^k - V_{I-1}^k\right)$

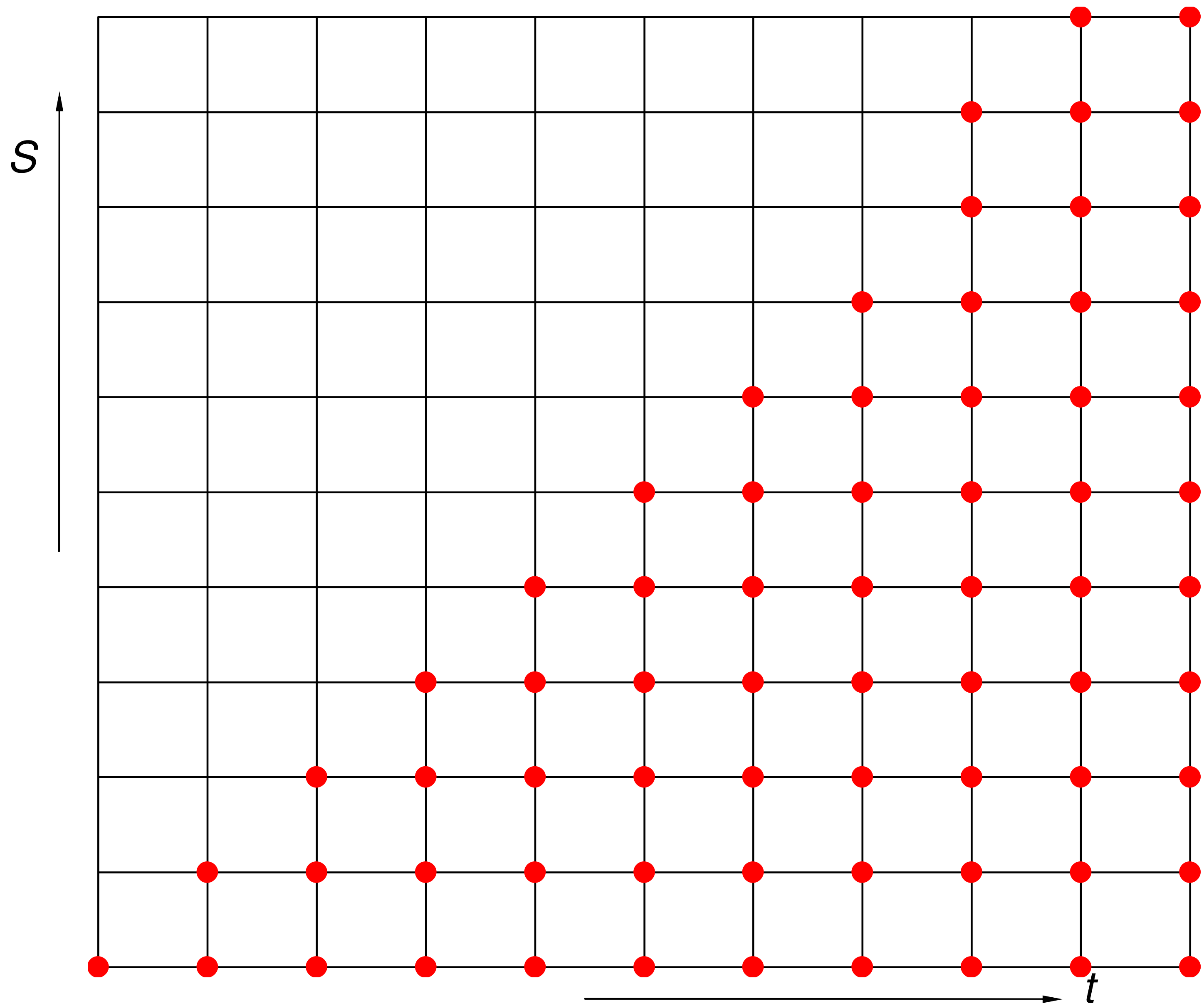
$V_I^{k+1} = (\alpha_I - \gamma_I)V_{I-1}^{k} + (\beta_I + 2\gamma_I)V_I \; \cdot \; ??$

I

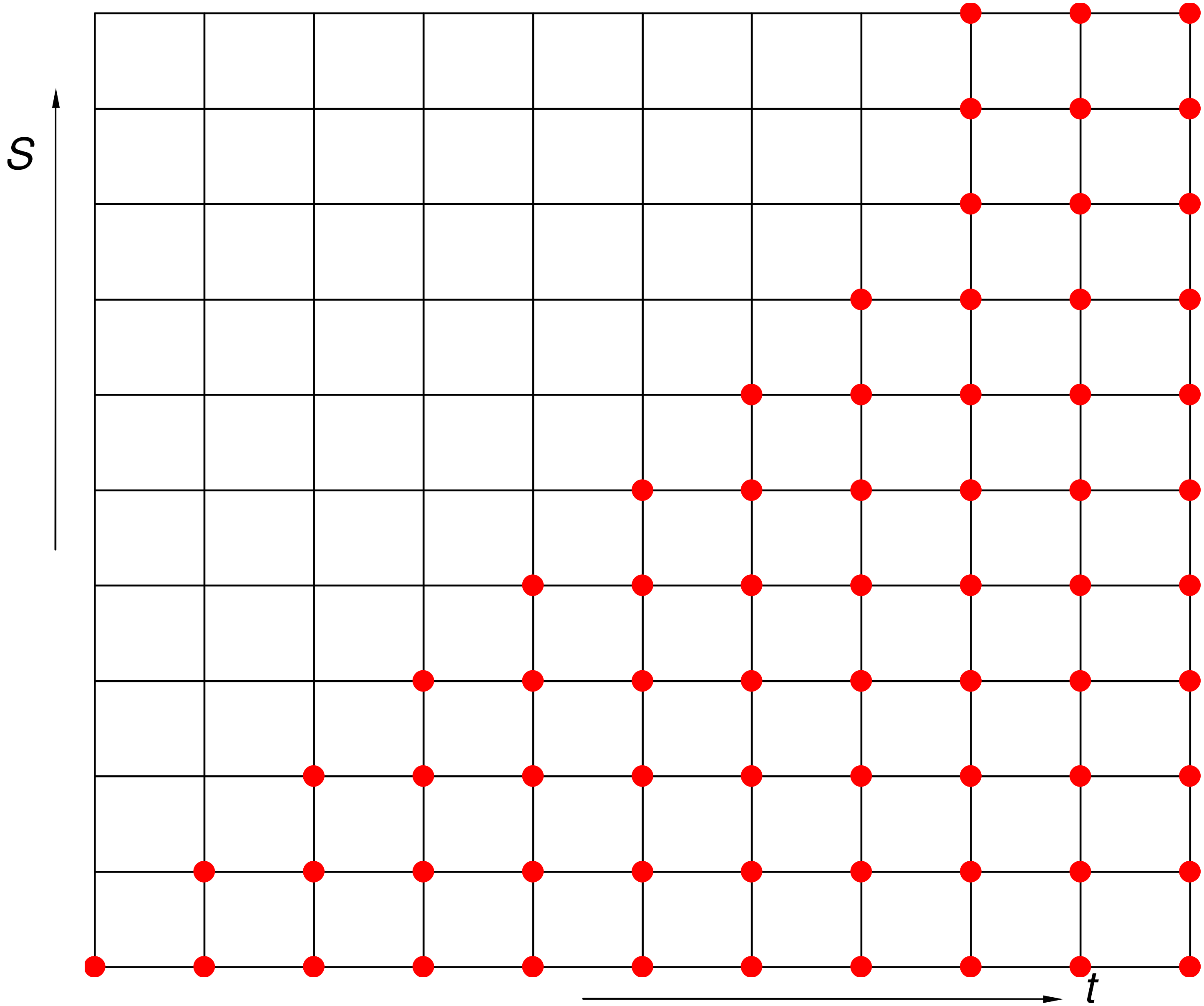


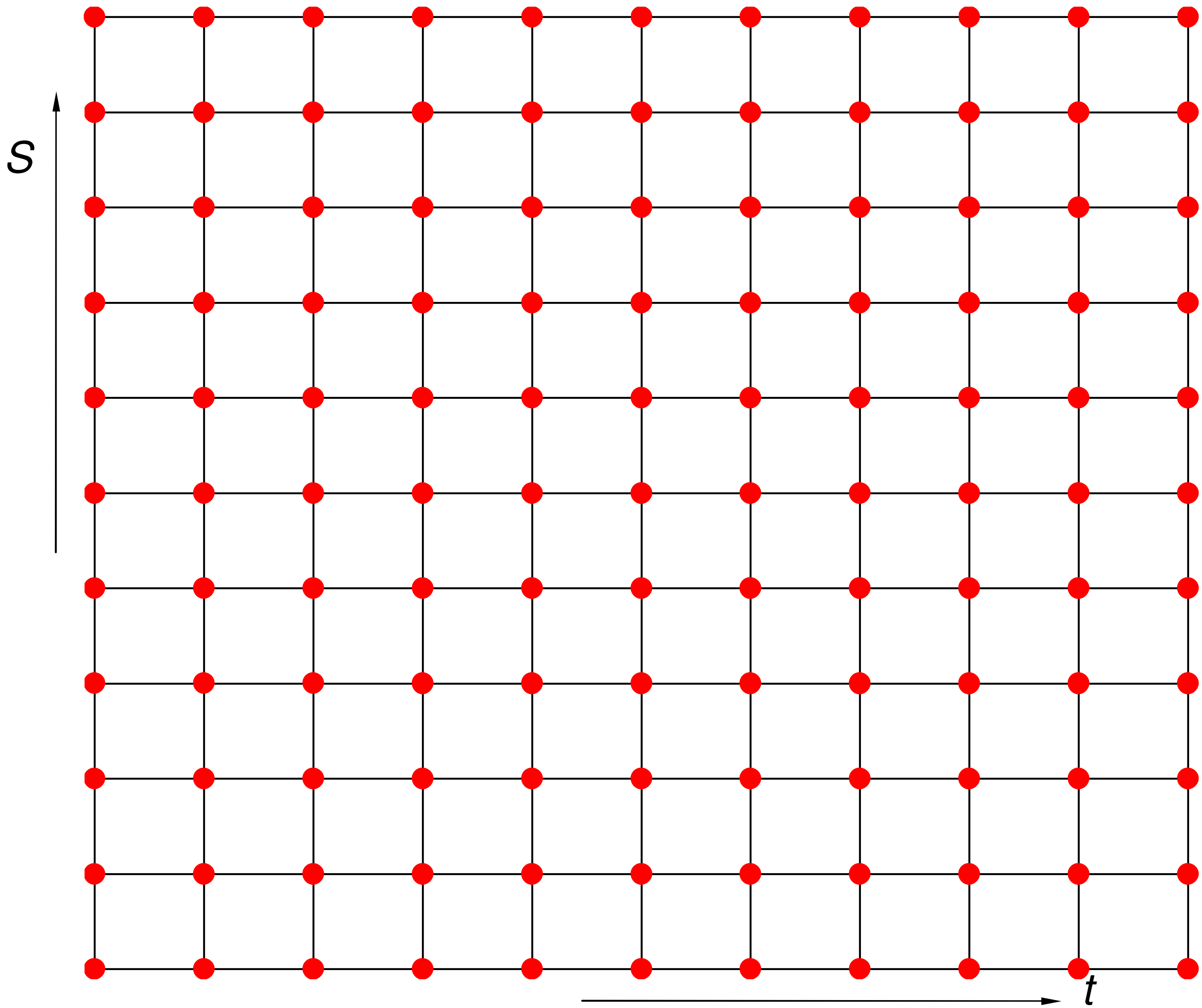

S
t


S
linear extrapolation
t



S
t


S
t



S
t

Accuracy and computational time

Again let's use ϵ to represent the desired accuracy in a calculation.

We know that errors are $O(\delta t)$ and $O(\delta S^2)$. It makes sense to have errors due to the time step and to the finite number of simulations to be of the same order. So we would choose:

$$\delta t = O(\epsilon) \quad \text{and} \quad \delta S = O(\epsilon^{1/2}).$$

The time taken is then proportional to number of calculations, therefore

$$\text{Time taken} = O(\epsilon^{-3/2}).$$

In higher dimensions. . .

Suppose you have a basket option with D underlyings. The time taken now becomes

$$\text{Time taken} = O(\epsilon^{-1-D/2}).$$

This is very sensitive to dimension!

Other issues

- Greeks

- Early exercise (and other decisions)

The advantages of the explicit method

- It is very easy to program and hard to make mistakes

- When it does go unstable it is usually obvious

- It copes well with coefficients that are asset and/or time dependent

- it copes very well with early exercise

- It can be used for modern option-pricing models

The disadvantages of the explicit method

- There are restrictions on the time step

- It is slower than Monte Carlo in high dimensions

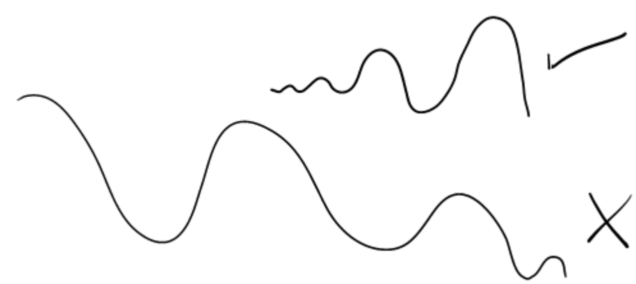
Explicit scheme — simple to use

$S := n\,\delta S$

Problem — Conditionally stable

$t := T - m\,\delta t$

$V(S,t) := V_n^m$



Fourier stability / Von-Neumann's Method

Suppose grid is fixed i.e δS, δt — const.

$V_n^m = \hat{V}_n^m + \epsilon_n^m$ $\quad \epsilon_n^m$ error. $\therefore$ ϵ_n^m is created by

explicit FDM $\Rightarrow$ ϵ_n^m also satisfies the earlier F.D expression

$$\epsilon_n^{m+1} = \alpha_n \epsilon_{n-1}^m + \beta_n \epsilon_n^m + \gamma_n \epsilon_{n+1}^m$$

Error $\propto e^{in\omega}$

ω - freq. of waves

Waves have an amplitude $\underline{\underline{a}}$ $\quad$ Error $\epsilon_n^m = a^m e^{in\omega}$ $\quad i = \sqrt{-1}$

$$\cancel{a^{m+1}} \cancel{e^{in\omega}} = \alpha_n \cancel{a^m} e^{i(n-1)\omega} + \beta_n \cancel{a^m} \cancel{e^{in\omega}} + \gamma_n \cancel{a^m} e^{i(n+1)\omega} \longrightarrow a = \alpha_n e^{-i\omega} + \beta_n + \gamma_n e^{i\omega}$$

Next point is important: In a parabolic PDE, $\frac{\partial V}{\partial t}$, $\frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}$ that control the stability $\Theta \sim -\frac{1}{2}\sigma^2 S^2 \Gamma$ $\therefore$ from $\alpha_n, \beta_n, \gamma_n$

collect terms from Θ, $\frac{1}{2}\sigma^2 S^2 \Gamma$

$$\alpha_n = \gamma_n = \frac{1}{2}\sigma^2 n^2 dt \qquad \beta_n = 1 - \sigma^2 n^2 dt$$

$$a = \sigma^2 n^2 dt \left(\frac{e^{-i\omega} + e^{i\omega}}{2}\right) + 1 - \sigma^2 n^2 dt = \sigma^2 n^2 dt \cos\omega - \sigma^2 n^2 dt + 1$$

$$a = \sigma^2 n^2 \delta t \underbrace{(\cos\omega - 1)}_{-2\sin^2\frac{\omega}{2}} + 1$$

Trig. identity $\cos 2x = c^2 - s^2$

$= 2c^2 - 1$

$\cos 2x = 1 - 2s^2$

$\cos 2x - 1 = -2\sin^2 x$

with $x = \frac{\omega}{2}$

$\cos\omega - 1 = -2\sin^2\frac{\omega}{2}$

$$a = 1 - 2\sigma^2 n^2 \sin^2\frac{\omega}{2}\,\delta t$$

$n = 0, \ldots, N$

Amplitude $|a| < 1$

$$\left|1 - 2\sigma^2 n^2 \sin^2\frac{\omega}{2}\,\delta t\right| < 1$$

$n = N$

$\left|\sin^2\frac{\omega}{2}\right| \leq 1$

$$|1 - 2\sigma^2 N^2 \delta t| < 1$$

This condition cannot be violated

$$-1 < 1 - 2\sigma^2 N^2 \delta t < 1$$

$$-2 < -2\sigma^2 N^2 \delta t < 0$$

$$1 > \sigma^2 N^2 \delta t \geq 0$$

$$\boxed{\delta t < \frac{1}{\sigma^2 N^2}}$$

Conditional stability

This gives the relationship between the time step and no. of assets in stock

$$\delta t \sim O\left(\frac{1}{N^2}\right)$$

Summary

Please take away the following important ideas

- There are two main numerical methods for pricing derivatives

- Monte Carlo methods exploit the relationship between option prices and expectations

- The finite-difference method solved a discretized version of the Black–Scholes equation