FMO6 — Web: https://tinyurl.com/ycaloqk6 Polls: https://pollev.com/johnarmstron561

Lecture 8

Dr John Armstrong

King's College London

August 22, 2020

Finite Difference Methods

Risk neutral pricing

We have learned how to price the following by Monte Carlo

- European Call Options
- European Put Options
- Digital Call Options
- Knockout Options
- Asian Options

But what options can't we price?

Motivation

- Finite difference methods allow us to price American Options
- They given us alternative methods of pricing European Options which is great for testing
- Exchange traded options on stocks are typically American
- We can't price Google options yet!

European Options

In your project you might want to numerically test examples about European options. It helps to know that:

- Exchange traded options on indices are typically European
- *Call options* on non dividend paying stocks have the same price whether they are European or American.

So if you want to find real data to test a theory, you can.

Finite difference methods

- Finite difference methods are a method of solving PDEs (partial differential equations)
- In a Black–Scholes world, the risk neutral price of a derivative on a stock obeys the Black–Scholes PDE.

$$V_t + \frac{1}{2}\sigma^2 S^2 V_{SS} + rSV_S - rV = 0$$

- $$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$
- V is the price of the derivative. S is the stock price. t is time.

How to remember the PDE

- It's a PDE for V
- It's first order in time
- It's second order in S
- It's linear

In other words it looks like this:

$$* \frac{\partial V}{\partial t} + * \frac{\partial^2 V}{\partial S^2} + * \frac{\partial V}{\partial S} + * V = 0$$

Dimensional analysis: The S terms

To remember the coefficients

- WLOG the coefficient of

$$\frac{\partial V}{\partial t} = 1$$

- Each term has units of

$$\$\text{years}^{-1}$$

- So the equation must be

$$\frac{\partial V}{\partial t} + *S^2 \frac{\partial^2 V}{\partial S^2} + *S \frac{\partial V}{\partial S} + *V = 0$$

with the $*$'s having units of years^{-1}

The last coefficient

Any derivative must obey the PDE including

- A derivative with payoff $V = e^{rt}$ (i.e. the cash account). In this case
$$\frac{\partial V}{\partial t} = re^{rt}, \quad \frac{\partial V}{\partial S} = 0, \quad \frac{\partial^2 V}{\partial^2 S} = 0$$
- So PDE must be:

$$\frac{\partial V}{\partial t} + *S^2 \frac{\partial^2 V}{\partial S^2} + *S \frac{\partial V}{\partial S} - rV = 0$$

The second to last coefficient

The stock also obeys the PDE. It satisfies

$$\frac{\partial V}{\partial t} = 0, \quad \frac{\partial V}{\partial S} = 1, \quad \frac{\partial^2 V}{\partial^2 S} = 0$$

We deduce that the PDE must be:

$$\frac{\partial V}{\partial t} + *S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

Conclusion

All you have to remember is that $* = \frac{1}{2}\sigma^2$!

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

Note that the units are dollars/year throughout.

The Explicit Method

- Suppose that I know the payoff at time T as a function of S. This is $V_T(S)$.
- I can then compute $\frac{\partial V}{\partial S}$ and $\frac{\partial^2 V}{\partial S^2}$.
- Plugging this into the Black–Scholes PDE I can now compute

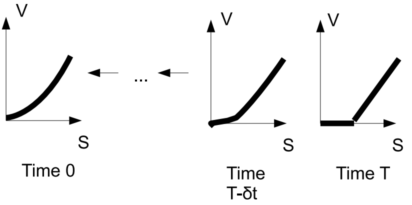
$$\frac{\partial V}{\partial t}$$

- Using

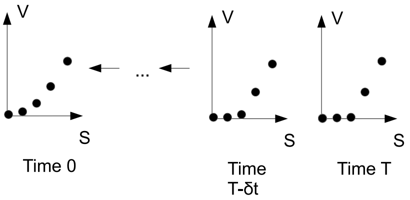
$$V_{T-\delta t} \approx V_t - \frac{\partial V}{\partial t} \delta t$$

  we can now compute $V_{T-\delta t}$
- So given payoff function at time T we can approximate the price at $V_{T-\delta t}$.
- Stepping back in time we can compute the price at time 0.

Picture of algorithm

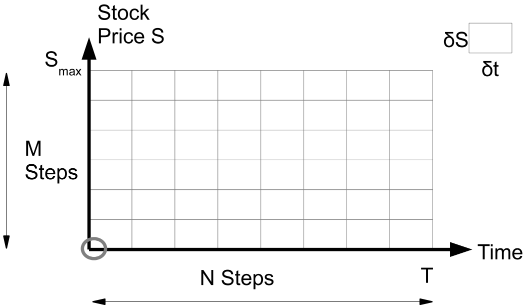


We can't store a function on a computer, so use a grid of points instead



Grid notation

Equivalently we compute a grid of values showing how price V varies with S and t.



$$\delta S = \frac{S_{\max}}{M}, \qquad \delta t = \frac{T}{N}$$

Let $V_{(i,j)}$ be the value at point (i, j) in this grid with i and j integers.

- We have central difference estimate

$$\frac{\partial V}{\partial S}_{(i,j)} \approx \frac{V_{(i,j+1)} - V_{(i,j-1)}}{2\delta S}$$

- Second derivative estimate

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

- By PDE, plus backwards estimate for $\frac{\partial V}{\partial t}$:

$$\begin{aligned} V_{(i-1,j)} &\approx V_{(i,j)} - \delta t \frac{\partial V}{\partial t}_{(i,j)} \\ &\approx V_{(i,j)} + \delta t \left(\frac{1}{2}\sigma^2 S^2 V_{SS} + rSV_S - rV \right) \end{aligned}$$

Recurrence equation

- This gives a recurrence equation

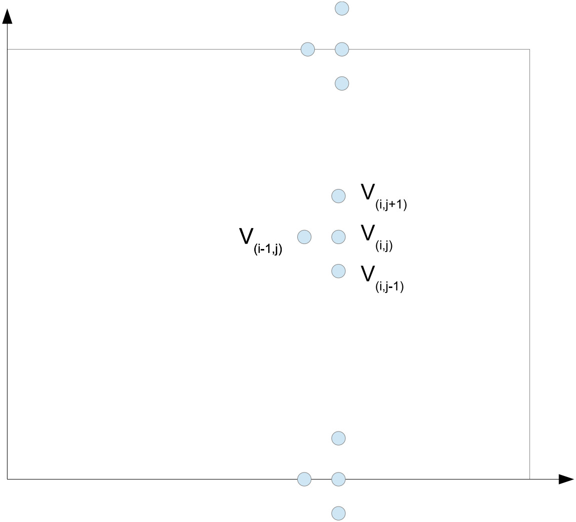
$$V_{(i-1,j)} = \text{some function of } V \text{ at later time } i$$

- with “initial” condition (for a call option)

$$V_{(N,j)} = \text{payoff at maturity} = \max\{j\delta S - K, 0\}$$

- However, we also need to consider the top and bottom boundaries

Boundary conditions



Boundary conditions for European Call Option

- Our recurrence equation breaks down at the top and bottom boundary
- We impose boundary conditions derived using simple heuristic arguments (or more rigorously limiting arguments)
- When $S = 0$, a call option is worthless. So $V(t, 0) = 0$. So $V_{(i,0)} = 0$.
- When $S = S_{\max}$, it is unlikely to end out of the money. So it is worth roughly the same as a portfolio consisting of one unit of stock and $-K$ zero coupon bonds. Hence $V(t, S_{\max}) \approx S_{\max} - e^{-r(T-t)}K$. In detail:

$$
\begin{aligned}
V(t, S_{\max}) &= e^{-r(T-t)} E_Q(\text{payoff} | S_t = S_{\max}) \\
&\approx e^{-r(T-t)} E_Q(S_T - K | S_t = S_{\max}) \\
&= E_Q(e^{-r(T-t)} S_T | S_t = S_{\max}) - e^{-r(T-t)} E_Q(K | S_t = S_{\max}) \\
&= S_{\max} - e^{-r(T-t)} K
\end{aligned}
$$

Choosing boundary conditions

- We do not know the exact price along the top and bottom boundary, we must use approximation arguments.
- If the boundary is far enough away, our solution won't be very sensitive to the boundary.
- What is far enough away? You need to choose $S_{\max}$ so that our approximation on the top boundary is reasonably accurate. Recall that over a time interval δt the change in the log stock price is normally distributed with mean $(r - \frac{1}{2}\sigma^2)\delta t$ and standard deviation $\sigma\sqrt{t}$. So you might choose $S_{\max} = e^{-(r-\frac{1}{2}\sigma^2)T+4\sigma\sqrt{T}}K$ as a 4 standard deviation movement in the log of the stock price is fairly unlikely.

Boundary conditions for European Call Option

- So our boundary conditions are:

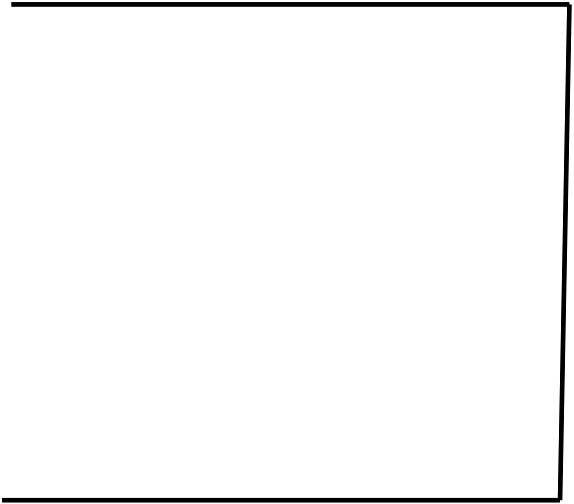
$$V_{(N,j)} = \max\{S_j - K, 0\}$$

$$V_{(i,0)} = 0$$

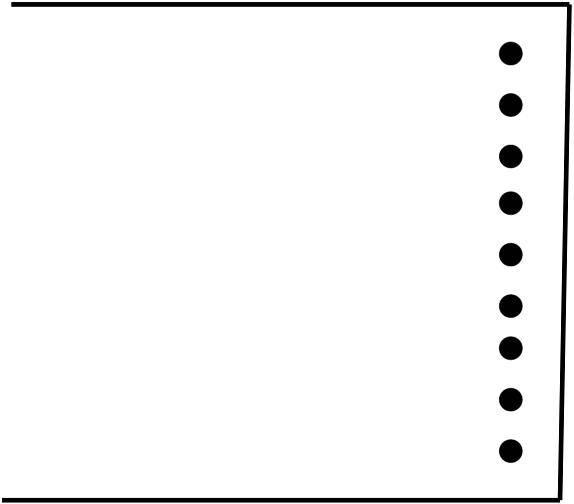
$$V_{(i,M)} = S_{\max} - e^{-r(T-t)}K$$

- We have a recurrence relation for all other $V_{(i,j)}$
- We can now solve this in Matlab

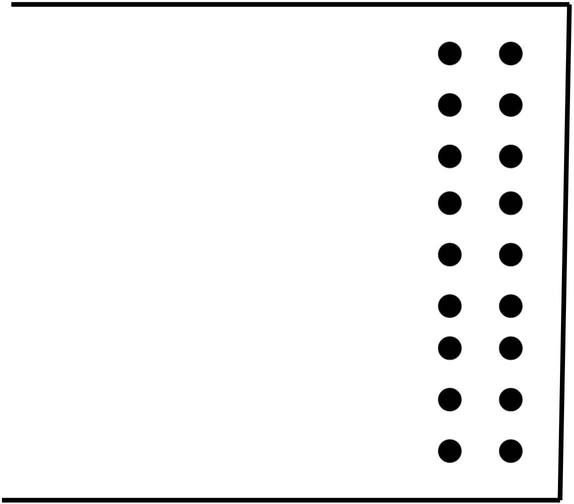
Boundary conditions



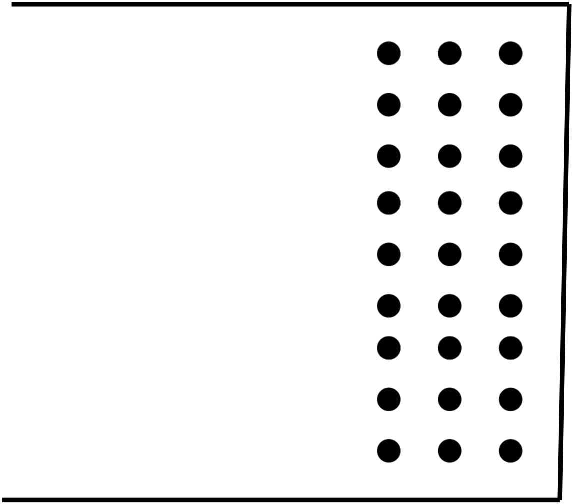
Step 1



Step 2



Step 3



Explicit calculation

- Before proceeding to the Matlab, let's compute the difference equation explicitly.
- Black–Scholes PDE

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

- Finite difference equations

$$\frac{V_{(i,j)} - V_{(i-1,j)}}{\partial t} + \frac{1}{2}\sigma^2 S_j^2 \left(\frac{V_{(i,j+1)} - 2V_{(i,j)} + V_{(i,j-1)}}{(\delta S)^2} \right) + rS_j \left(\frac{V_{(i,j+1)} - V_{(i,j-1)}}{2\delta S} \right) - rV_{(i,j)} = 0$$

- Central difference in S. Backwards difference in t.

Rearrange to get

$$V_{(i-1,j)} = \alpha_j V_{(i,j+1)} + \beta_j V_{(i,j)} + \gamma_j V_{(i,j-1)}$$

where

$$\alpha = \frac{1}{2}\frac{\sigma^2 S_j^2 \delta t}{\delta S^2} + \frac{r S_j}{2\delta S}\delta t$$

$$\beta = 1 - \frac{\sigma^2 S_j^2 \delta t}{\delta S^2} - r\delta t$$

$$\gamma = \frac{1}{2}\frac{\sigma^2 S_j^2 \delta t}{\delta S^2} - \frac{r S_j}{2\delta S}\delta t$$

Check: everything is dimensionless.

Matlab code - slide 1

```
function [ price, V ] = priceCallByExplicitMethod(  ...
    K, T, S0, r, sigma, SMax, N, M )
%PRICECALLBYEXPLICITMETHOD Price a call option by the explicit method
%    using the Black Scholes PDE directly

iMin = 1;
iMax = N+1;
jMin = 1;
jMax = M+1;

dS = SMax/M;
dt = T/N;

t=0:dt:T;
S=0:dS:SMax;
```

Matlab code - slide 2. Boundary conditions.

```
V=zeros(N+1,M+1);
V(iMax,:)=max(S-K,0); % Call option payoff
V(:,jMax)=SMax - exp(-r*(T-t))*K; % Estimate for high stock price
V(:,jMin)=0; % Worthless when stock is zero
```

Matlab code - slide 3. Recurrence relation.

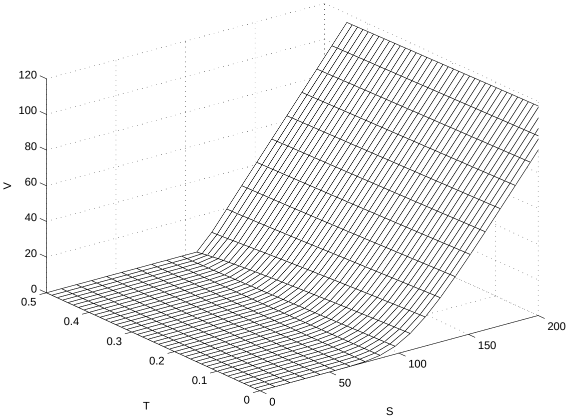
```
for i=iMax:-1:(iMin+1)
    for j=(jMin+1):(jMax-1)
        a = (0.5 * sigma^2 * S(j)^2 * dt)/(dS^2) + r*S(j)*dt/(2*dS);
        b = 1 - sigma^2 * S(j)^2*dt/(dS^2) - r*dt;
        c = (0.5 * sigma^2 * S(j)^2 * dt)/(dS^2) - r*S(j)*dt/(2*dS);
        V(i-1,j) = a*V(i,j+1) + b*V(i,j) + c*V(i,j-1);
    end
end
```

Final touches

```
% We want to extract the actual price at S0, so use
% polynomial interpolation since S0 may not fall on a grid point
s0Index = S0/dS+jMin;
price = linearInterpolate(V(iMin,:),s0Index);

end
```

Result



Numerical Results

Parameter choices

- $\sigma = 0.3$
- $S_0 = 108$
- $T = 0.5$
- $K = 110$
- $r = 0.05$

Taking $S_{\max} = e^{-(r-\frac{1}{2}\sigma^2)T+4\sigma\sqrt{T}}$

Black–Scholes Price	9.455
Finite difference with $M = 100,\ N = 500,$	9.454
Finite difference with $M = 150,\ N = 500,$	8.7×10^{148}

- Something has gone terribly wrong. This is called *instability*.

The Heat Equation

Heat Equation - Motivation

- The Black–Scholes PDE is so complex because we've made a poor choice of variables.
- Clearly life would be easier if eliminated r from the equations. We can do this by working in real terms

$$W = e^{-rt} V$$

$$\tilde{S} = e^{-rt} S$$

- We also know that by working with $\tilde{x} = \log \tilde{S}$ instead of $\tilde{S}$ everything should be easier mathematically.

$$W = e^{-rt} V, \qquad \tilde{x} = -rt + \log(S)$$

- In fact we can make things even easier by taking:

$$W = e^{-rt} V, \qquad x = -(r - \frac{1}{2}\sigma^2)t + \log(S)$$

because x then follows driftless Brownian motion

Change of coordinates

Lemma

Under the change of coordinates

$$W = e^{-rt} V$$

$$x = -(r - \frac{1}{2}\sigma^2)t + \log S$$

$$\tau = -t$$

The Black–Scholes PDE transforms to the heat equation

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

Proof - step 1

- The Black–Scholes PDE is

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

- Writing $V = e^{rt}W$ we get:

$$e^{rt}\frac{\partial W}{\partial t} + re^{rt}W + \frac{1}{2}\sigma^2 S^2 e^{rt}\frac{\partial^2 W}{\partial S^2} + rSe^{rt}\frac{\partial W}{\partial S} - re^{rt}W = 0$$

- Cancelling the e^{rt} terms:

$$\frac{\partial W}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 W}{\partial S^2} + rS\frac{\partial W}{\partial S} = 0 \tag{1}$$

Proof - step 2

$$\frac{\partial W}{\partial t} = \frac{\partial W}{\partial \tau}\frac{\partial \tau}{\partial t} + \frac{\partial W}{\partial x}\frac{\partial x}{\partial t} = -\frac{\partial W}{\partial \tau} - \left(r - \frac{1}{2}\sigma^2\right)\frac{\partial W}{\partial x}$$

Note there is a potential trap here because partial differentiation notation is not that great. It would be better to write

$$\frac{\partial W}{\partial t} =: \frac{\partial^1 W}{\partial^0 S \partial^1 t}$$

to emphasize that we are holding S fixed. And to write

$$\frac{\partial W}{\partial \tau} =: \frac{\partial^1 W}{\partial^0 x \partial^1 \tau}$$

to emphasize that we are now holding x fixed.

So even though $\tau = -t$

$$\frac{\partial W}{\partial t} \neq -\frac{\partial W}{\partial \tau}$$

Proof - step 2, continued

$$\frac{\partial W}{\partial t} = \frac{\partial W}{\partial \tau}\frac{\partial \tau}{\partial t} + \frac{\partial W}{\partial x}\frac{\partial x}{\partial t} = -\frac{\partial W}{\partial \tau} - \left(r - \frac{1}{2}\sigma^2\right)\frac{\partial W}{\partial x}$$

$$\frac{\partial W}{\partial S} = \frac{\partial W}{\partial \tau}\frac{\partial \tau}{\partial S} + \frac{\partial W}{\partial x}\frac{\partial x}{\partial S} = \frac{1}{S}\frac{\partial W}{\partial x}$$

$$\frac{\partial^2 W}{\partial S^2} = -\frac{1}{S^2}\frac{\partial W}{\partial x} + \frac{1}{S}\frac{\partial}{\partial S}\left(\frac{\partial W}{\partial x}\right) = -\frac{1}{S^2}\frac{\partial W}{\partial x} + \frac{1}{S^2}\frac{\partial^2 W}{\partial x^2}$$

So equation (1) two slides ago becomes:

$$-\frac{\partial W}{\partial \tau} - \left(r - \frac{1}{2}\sigma^2\right)\frac{\partial W}{\partial x} + \frac{1}{2}\sigma^2\frac{\partial^2 W}{\partial x^2} - \frac{1}{2}\sigma^2\frac{\partial W}{\partial x} + r\frac{\partial W}{\partial x} = 0$$

which simplifies to:

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

What have we gained?

The advantages of the heat equation are:

- The Heat equation has constant coefficients which makes it much easier to understand. This is the main purpose of the change.
- There is a huge literature on the heat equation which we can exploit
- The coordinates are "better" because they scale correctly. For larger stock values, we should use larger grid sizes. A one dollar step size is a large step size for a one dollar stock, but a small step size for a one hundred dollar stock. Taking logs automatically scales things.

Against this:

- The boundary conditions are a little more awkward
- For pricing barrier options, we have transformed constant barriers (easy) to curved barriers (harder)

Slightly different coordinates

Lemma

Under the change of coordinates

$$W = e^{-rt} V$$

$$x = -(r - \frac{1}{2}\sigma^2)t + \log S$$

The Black–Scholes PDE transforms to the time reversed heat equation

$$\frac{\partial W}{\partial t} = -\frac{1}{2}\sigma^2 \frac{\partial^2 W}{\partial x^2}$$

Hazard Warning

If you try to prove this directly you will almost certainly make the mistake of thinking

$$\frac{\partial^1 W}{\partial^0 S \partial^1 t} = \frac{\partial^1 W}{\partial^0 x \partial^1 t}$$

because in the ambiguous conventional notation this becomes

$$\frac{\partial W}{\partial t} = \frac{\partial W}{\partial t}$$

which looks plausible, but isn't always true!

Feynman–Kac formula approach

There is an alternative, and I think much easier proof:

- Suppose V obeys the Black-Scholes PDE.
- Then by Feynman-Kac $V(S,t)$ is given as the expectation of

  $$E(e^{-r(T-t)}V(S_T,T)|S_t = S)$$

  where S_t follows the process

  $$\mathrm{d}S_t = S_t(r\mathrm{d}t + \sigma \mathrm{d}W_t)$$

- Then by Ito's lemma $x_t = \log(S_t) - (r - \frac{1}{2}\sigma^2)t$ follows driftless Brownian motion with volatility σ.
- Write $W(x_t,t) = e^{-rt}V(S_t,t)$ so

  $$\begin{aligned} W(x,t) &= e^{-rt}V(S_t,t) = e^{-rt}E(e^{-r(T-t)}V(S_T,T)|S_t = S) \\ &= E(e^{-rT}V(x_T,T)|S_t = S) = E(W(x_T,T)|S_t = S) \end{aligned}$$

- So by Feynman–Kac, W obeys the time-reversed heat equation.

Feynman-Kac

The solution to the PDE

$$\frac{\partial u}{\partial t} + \mu(x,t)\frac{\partial u}{\partial x} + \frac{1}{2}\sigma^2(x,t)\frac{\partial^2 u}{\partial x^2} - r(x,t)u(x,t) = 0$$

with final condition $u(x,T) = \psi(T)$ is given by

$$E\left(e^{-\int_t^T r(X_\tau,\tau)}\psi(X_T)|X_t = x\right)$$

where X follows the process:

$$\mathrm{d}X_t = \mu(X_t,t)\mathrm{d}t + \sigma(X_t,t)\mathrm{d}W_t$$

for W_t a Brownian motion.

Boundary conditions in new coordinates

Old boundary conditions for call option

- $V(t,S) \approx 0$ for small S
- $V(t,S) \approx S - e^{-r(T-t)}K$ for large S
- $V(T,S) = \max\{S-K, 0\}$

$$W = e^{-rt}V, \qquad S = e^{(r-\frac{1}{2}\sigma^2)t+x}$$

Boundary conditions for call option

- $W(t, x_{\text{min}}) \approx 0$
- $W(t, x_{\text{max}}) \approx e^{-\frac{1}{2}\sigma^2 t + x_{\text{max}}} - e^{-rT}K$
- $W(T,x) = \max\{e^{-\frac{1}{2}\sigma^2 T + x} - e^{-rT}K, 0\}$

Difference equation

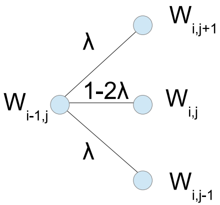
- PDE is
$$\frac{\partial W}{\partial t} = -\frac{1}{2}\sigma^2 \frac{\partial^2 W}{\partial x^2}$$
- Difference equation is:
$$\frac{W_{i,j} - W_{i-1,j}}{\delta t} = -\frac{1}{2}\sigma^2 \left(\frac{W_{i,j+1} - 2W_{i,j} + W_{i,j-1}}{\delta x^2} \right)$$
- Simplifies to:
$$W_{i-1,j} = \lambda W_{i,j+1} + (1 - 2\lambda) W_{i,j} + \lambda W_{i,j-1}$$
where
$$\lambda = \frac{1}{2}\sigma^2 \frac{\delta t}{\delta x^2}$$
- Noticeably simpler than the difference equation for Black-Scholes equation

Observation

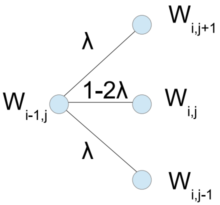
- Equation is

$$W_{i-1,j} = \lambda W_{i,j+1} + (1-2\lambda)W_{i,j} + \lambda W_{i,j-1}$$

- This is a weighted average of $W_{i,j+1}$, $(1-2\lambda)W_{i,j}$ and $\lambda W_{i,j-1}$ so long as $(1-2\lambda) > 0$.
- Physically temperature after a small time is a weighted average of nearby temperatures.



Financially



In terms of x

- Each time period δt, x moves up δx with Q-probability λ
- x moves down with Q-probability λ
- x stays the same with probability $1 - 2\lambda$

or in terms of S

- Stock moves up by a factor $e^{(r-\frac{1}{2}\sigma^2)\delta t+\delta x}$ with Q-probability λ
- Moves up by a factor $e^{(r-\frac{1}{2}\sigma^2)\delta t}$ with Q-probability $1 - 2\lambda$
- Moves down by a factor $e^{(r-\frac{1}{2}\sigma^2)\delta t-\delta x}$ with Q-probability λ

Prices are discounted risk neutral expectations

Confirmation of the interpretation

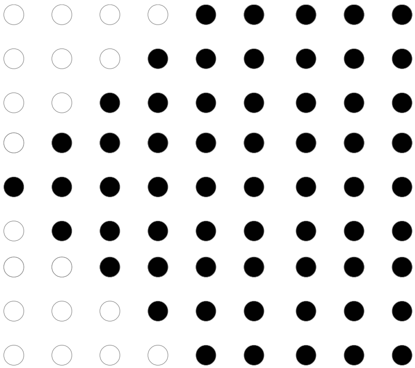
Define a process $\tilde{x}$

- $$\tilde{x}_{t+\delta t} = \begin{cases} \tilde{x}_t + \delta x & \text{with probability } \lambda \\ \tilde{x}_t & \text{with probability } 1 - 2\lambda \\ \tilde{x}_t - \delta x & \text{with probability } \lambda \end{cases}$$
- Expectation $E(\tilde{x}_{t+\delta t}) = E(x_t)$
- Variance

$$\begin{aligned} 2\lambda(\delta x)^2 &= 2\frac{\sigma^2}{2}\frac{\delta t}{\delta x^2}\delta x^2 \\ &= \sigma^2 \delta t \end{aligned}$$

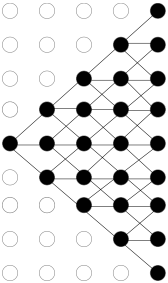
- So in the limit as $\delta t \to 0$ this is Brownian motion with drift 0 and volatility σ.
- $\tilde{x}$ in the limit follows same process as x

Points used



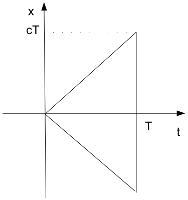
Not all the points on the boundary are used in the calculation. White circles represent data that is not needed to calculate price at black points.

Trinomial Tree



- Pricing using a trinomial tree with risk neutral probabilities λ, $1-2\lambda$, λ is equivalent to a heat equation method.
- Easier to program because there is no need for boundary conditions at top and bottom. This is at the expense of unnecessary computations at unlikely points.

Non-convergence



- Note that if you keep
$$\frac{\delta x}{\delta t}$$
as a constant c as you refine the grid, all data used comes from a fixed triangle
- Therefore the trinomial tree method cannot possibly converge as δt tends to 0 with c fixed. This is because you need to consider payoffs at T outside the triangle to compute the expectation correctly.

Stability

Stability

- We showed earlier that finite differences for the Blsack–Scholes PDE only works if you choose good values of N and M (the step sizes in each direction).
- We have just proved that it cannot converge if you make the seemingly obvious choice of increasing M and N at the same rate
- Finite difference equation for the heat equation is easiest to analyse. Recall it is:

$$W_{i-1,j} = \lambda W_{i,j+1} + (1-2\lambda)W_{i,j} + \lambda W_{i,j-1}$$

where

$$\lambda = \frac{1}{2}\sigma^2 \frac{\delta t}{\delta x^2}$$

- In fact the finite difference equation for the heat equation converges if and only if we keep $(1-2\lambda) > 0$ as $\delta t \to 0$. In other words it converges if and only if we can make the probabilisitic interpretation.

Theorem

If we choose a fixed value for

$$\lambda = \frac{1}{2}\sigma^2 \frac{\delta t}{\delta x^2}$$

with $1 - 2\lambda > 0$ *and consider the sequence of approximate solutions given by letting* $\delta x \to 0$ *in the explicit scheme for the heat equation. This will converge to a solution of the heat equation with the desired boundary conditions (we should add that we need some conditions such as that the boundary is compact and the boundary conditions are continuous).*

Components of proof

- The proof is beyond the scope of this course.
    - Prove the problem is *well-posed*, i.e. there is a unique solution depending continuously on the boundary conditions.
    - Prove the scheme is *consistent*, i.e. the approximate operators used in the scheme are close to the operators in the PDE.
    - Prove the scheme is *stable*, i.e. if you start with a small values W at the final time, they will remain small as you move back in time. This means that if your solution is fairly accurate but has a small rounding ϵ then the effects of that rounding error will remain small as you move back in time. (Note that the equation is linear, so to compute the effects of the rounding error on the scheme we can simply apply the scheme to the rounding error.)
    - Apply the Lax Equivalence Theorem that says that for well posed problems, stability is equivalent to convergence.
- Note that since our process takes weighted averages it will be numerically stable as the average of small things is still small.

Remarks

- Instability shows itself numerically as extreme sensitivity to rounding errors - i.e. numerical instability.
- Our example of a calculation with bad choices of M and N illustrates the instability of the scheme vividly.

Rate of Convergence

- For given boundary conditions, the Explicit finite difference scheme for the heat equation using fixed

$$\lambda = \frac{1}{2}\sigma^2 \frac{\delta t}{\delta x^2}$$

with $1 - 2\lambda > 0$ has an error of $O(\delta t)$.

Implementation

Parameters

- K, T, S_0, r, σ, N, M, nSds.
- N = number of time steps
- M = number of x steps
- nSds = number of standard deviations to include each side of x_0 in the grid. 8 standard deviations would be plenty.

Initialization Code

```
x0 = log( S0 );
xMin = x0 - nSds*sqrt(T)*sigma;
xMax = x0 + nSds*sqrt(T)*sigma;

dt = T/N;
dx = (xMax-xMin)/M;

iMin = 1;
iMax = N+1;
jMin = 1;
jMax = M+1;
x = xMin:dx:xMax;
t = 0:dt:T;

lambda = 0.5*sigma^2 * dt/(dx)^2;
```

Looping Code

```
% Use boundary condition to create vector currW
currW=max(exp(-0.5*sigma^2*T + x) - exp(-(r*T))*K,0);

for i=iMax:-1:iMin+1
    % Use recurrence for all points except jMin and jMax
    currW(jMin+1:jMax-1)=      lambda*currW((jMin):(jMax-2)) ...
                        +(1-2*lambda)*currW((jMin+1):(jMax-1)) ...
                        +      lambda*currW((jMin+2):(jMax));
    % Use boundary conditions for jMin and jMax
    currW(jMin)=0;
    currW(jMax)=exp(-0.5*sigma^2 * t(i) + x(jMax))- exp(-r*T)*K;
end

price = currW(iMin,jMin+M/2);
```

Remarks

The code is written less naively

- We use a single vector `currW` which holds the option price at the current time.
- This saves a lot of memory
- We have no loop across the space dimension. This code is vectorized.
- This makes it much faster.

Convergence

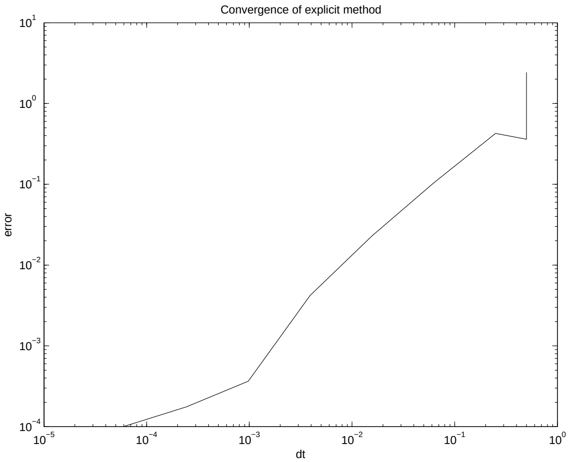


Figure: Convergence of the explicit method (error against δt)

Remarks

- Plot shows convergence to the Black–Scholes price with `nSds=8`.
- It has rate of convergence predicted by theory
- (Note theory says it will converge to solution with our approximate boundary conditions rather than the true solution. Our boundary condition approximation is very accurate when `nSds=8`)

Practicalities

- To price a European put or call use the Heat equation method
- To price a Knock-out option, use Black Scholes PDE method. This is because this has a simpler boundary condition at the barrier.
- To price a Knock-in option, use the fact that Knock Out + Knock In = No Barrier.

American Options

- For an American option, suppose we've computed the price $V_{i,j}$ at time point i.
- Suppose we compute $\tilde{V}_{i-1,j}$ by using the difference equation for the Black–Scholes PDE
- This is the price at time $i-1$ of an option you can only exercise from time i onwards.
- Let $E_{i,j}$ be the value obtained by exercising at time i
- $$V_{i-1,j} = \max\{\tilde{V}_{i-1,j}, E_{i-1,j}\}$$

More explicitly

We have

$$\tilde{V}_{i-1,j} = \alpha_j V_{i,j+1} + \beta_j V_{i,j} + \gamma_j V_{i,j-1}$$

where α_j, β_j and γ_j are as before.

$$E_{i,j} = (S_j - K)^+$$

$$V_{i-1,j} = \max\{\tilde{V}_{i,j}, E_{i,j}\}.$$

The same technique works for the heat equation and trinomial tree approaches.