CHAPTER 28

FINITE-DIFFERENCE METHODS FOR ONE-FACTOR MODELS

1. **Write a program to value European call and put options by solving Black–Scholes equation with suitable final and boundary conditions. Include a constant, continuous dividend yield on the underlying share.**

   We must solve

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

   with final condition

$$V(S, T) = \Lambda(S),$$

   and boundary conditions

$$\frac{\partial^2 V}{\partial S^2} \to 0 \text{ as } S \to \infty,$$
$$\frac{\partial^2 V}{\partial S^2} = 0 \text{ on } S = 0,$$

   where

$$\Lambda(S) = \max(S - E, 0),$$

   for a call and

$$\Lambda(S) = \max(E - S, 0),$$

   for a put.

   The following code solves this problem using an explicit finite difference scheme:

   When param=0, the code values a call option and when param=1, the code values a put option.

```
Function OptionValue(Asset As Double, Strike As Double, Expiry As _
                Double, Volatility As Double, IntRate As Double, Div _
                As Double, param As Integer, NoAssetSteps)

Dim VOld(0 To 100) As Double
Dim VNew(0 To 100) As Double
Dim Delta(0 To 100) As Double
```

```
Dim Gamma(0 To 100) As Double
Dim S(0 To 100) As Double
Dim Ssqd(0 To 100) As Double

halfvolsqd = 0.5 * Volatility * Volatility
AssetStep = 2 * Strike / NoAssetSteps
NearestGridPt = Int(Asset / AssetStep)
dummy = (Asset - NearestGridPt * AssetStep) / AssetStep
Timestep = AssetStep * AssetStep / Volatility / Volatility / _
                      (4 * Strike * Strike)
NoTimesteps = Int(Expiry / Timestep) + 1
Timestep = Expiry / NoTimesteps

For i = 0 To NoAssetSteps

    S(i) = i * AssetStep
    Ssqd(i) = S(i) * S(i)
    If param = 0 Then VOld(i) = max(S(i) - Strike, 0)
    If param = 1 Then VOld(i) = max(Strike - S(i), 0)

Next i

For j = 1 To NoTimesteps

    For i = 1 To NoAssetSteps - 1

        Delta(i) = (VOld(i + 1) - VOld(i - 1)) / (2 * AssetStep)
        Gamma(i) = (VOld(i + 1) - 2 * VOld(i) + VOld(i - 1)) _
                           / (AssetStep * AssetStep)
        VNew(i) = VOld(i) + Timestep * (halfvolsqd * Ssqd(i) *  _
                Gamma(i) + (IntRate - Div) * S(i) * Delta(i) - IntRate _
                           * VOld(i))

    Next i

        VNew(0) = 2 * VNew(1) - VNew(2)
        VNew(NoAssetSteps) = 2 * VNew(NoAssetSteps - 1) - _
                                  VNew(NoAssetSteps - 2)

For i = 0 To NoAssetSteps

    VOld(i) = VNew(i)

Next i

Next j

OptionValue = (1 - dummy) * VOld(NearestGridPt) + _
                       dummy * VOld(NearestGridPt + 1)

End Function
```

2. **Adjust your program to value call options with the forward price as underlying.**

The partial differential equation for the value of an option, $V(F, t)$, with the forward price, F, as underlying is

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

This is just the usual Black–Scholes equation without the delta term. The final and boundary conditions remain unchanged. We therefore alter our OptionValue code to remove the delta term by changing the line

```
VNew(i) = VOld(i) + Timestep * (halfvolsqd * Ssqd(i) * Gamma(i) + _
                (IntRate - Div) * S(i) * Delta(i) - IntRate * VOld(i))
```

to

```
VNew(i) = VOld(i) + Timestep * (halfvolsqd * Ssqd(i) * Gamma(i) - _
                 IntRate * VOld(i))
```

3. **Write a program to value a down-and-out call option, with barrier below the strike price.**

We must solve

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

with final condition

$$V(S, T) = \max(S - E, 0),$$

and boundary conditions

$$\frac{\partial^2 V}{\partial S^2} \to 0 \text{ as } S \to \infty,$$

$$V(X, t) = 0,$$

where the out barrier is at $S = X$.

We must alter our code (from Question 1) to take account of the new lower boundary condition. It is also sensible to change the code so that the program only works out values for $S \geq X$ instead of $S \geq 0$, since we know $V = 0$ for $0 \leq S \leq X$.

The following code solves this problem using an explicit finite difference scheme:

When param=0, the code values a call option and when param=1, the code values a put option.

```
Function BarrierValue(Asset As Double, Strike As Double, Expiry As _
                Double, Volatility As Double, IntRate As Double, Div _
                As Double, param As Integer, Barrier as Double, _
                NoAssetSteps)

Dim VOld(0 To 100) As Double
Dim VNew(0 To 100) As Double
```

```
Dim Delta(0 To 100) As Double
Dim Gamma(0 To 100) As Double
Dim S(0 To 100) As Double
Dim Ssqd(0 To 100) As Double

halfvolsqd = 0.5 * Volatility * Volatility
AssetStep = 2 * Strike / NoAssetSteps
NearestGridPt = Int((Asset - Barrier)/ AssetStep)
dummy = ((Asset - Barrier) - NearestGridPt * AssetStep) / AssetStep
Timestep = AssetStep * AssetStep / Volatility / Volatility / _
                      (4 * Strike * Strike)
NoTimesteps = Int(Expiry / Timestep) + 1
Timestep = Expiry / NoTimesteps

For i = 0 To NoAssetSteps

    S(i) = Barrier + i * AssetStep
    Ssqd(i) = S(i) * S(i)
    If param = 0 Then VOld(i) = max(S(i) - Strike, 0)
    If param = 1 Then VOld(i) = max(Strike - S(i), 0)

Next i

For j = 1 To NoTimesteps

    For i = 1 To NoAssetSteps - 1

        Delta(i) = (VOld(i + 1) - VOld(i - 1)) / (2 * AssetStep)
        Gamma(i) = (VOld(i + 1) - 2 * VOld(i) + VOld(i - 1)) _
                           / (AssetStep * AssetStep)
        VNew(i) = VOld(i) + Timestep * (halfvolsqd * Ssqd(i) * _
                Gamma(i) + (IntRate - Div) * S(i) * Delta(i) - IntRate _
                * VOld(i))

    Next i

        VNew(0) = 0
        VNew(NoAssetSteps) = 2 * VNew(NoAssetSteps - 1) - _
                                  VNew(NoAssetSteps - 2)

For i = 0 To NoAssetSteps

    VOld(i) = VNew(i)

Next i

Next j

BarrierValue = (1 - dummy) * VOld(NearestGridPt) + _
                        dummy * VOld(NearestGridPt + 1)

End Function
```

4. Write a program to value compound options of the following form:

(a) Call on a call,
(b) Call on a put,
(c) Put on a call,
(d) Put on a put.

We must solve

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

with final condition

$$V(S, T) = \Lambda(S),$$

and boundary conditions

$$\frac{\partial^2 V}{\partial S^2} \to 0 \text{ as } S \to \infty,$$

$$\frac{\partial^2 V}{\partial S^2} = 0 \text{ on } S = 0,$$

where

$$\Lambda(S) = \max(C_{BS} - E, 0),$$

for a call on a call,

$$\Lambda(S) = \max(P_{BS} - E, 0),$$

for a call on a put,

$$\Lambda(S) = \max(E - C_{BS}, 0),$$

for a put on a call, and

$$\Lambda(S) = \max(E - P_{BS}, 0),$$

for a put on a put.

We must therefore alter the final condition in the OptionValue code of Question 1, to

```
For i = 0 To NoAssetSteps

    S(i) = i * AssetStep
    Ssqd(i) = S(i) * S(i)
    If param = 0 Then VOld(i) = max(CallValue(S(i), Expiry1) _
                                - Strike, 0)
    If param = 1 Then VOld(i) = max(PutValue(S(i), Expiry1) _
                                - Strike, 0)
    If param = 2 Then VOld(i) = max(Strike - CallValue(S(i), _
                                Expiry1), 0)
    If param = 3 Then VOld(i) = max(Strike - PutValue(S(i), _
                                Expiry1), 0)

Next i
```

where param=0 values a call on a call, param=1 values a call on a put, param=2 values a put on a call and param=3 values a put on a put.

CallValue(S,T) and PutValue(S,T) are functions which give the value of a European call and put option respectively, with current share price S and expiry after a time T. The OptionValue function from Question 1 could be used to calculate these values. Alternatively, we could rewrite the code so that these values are calculated first, since the code to do this is very similar to the code for the main program. A third option would be to use the explicit Black–Scholes formulae for these option values.

5. Alter your compound option program to value a chooser option which allows you to buy a call or a put at expiry.

We must solve

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

with final condition

$$V(S, T) = \max(C_{BS} - E_1, P_{BS} - E_2, 0)$$

and boundary conditions

$$\frac{\partial^2 V}{\partial S^2} \to 0 \text{ as } S \to \infty,$$

$$\frac{\partial^2 V}{\partial S^2} = 0 \text{ on } S = 0.$$

We must therefore alter the final condition in the OptionValue code of Question 1, to

```
For i = 0 To NoAssetSteps

    S(i) = i * AssetStep
    Ssqd(i) = S(i) * S(i)
    VOld(i) = max(CallValue(S(i), Expiry1) - Strike1, _
                       PutValue(S(i), Expiry1) - Strike2, 0)

Next i
```