Transactions on Engineering Sciences vol 15, © 1997 WIT Press, www.witpress.com, ISSN 1743-3533

Symbolic-Numeric Option Valuation

P. Mitic
Department of Mathematics, Rovaniemi Polytechnic, 96300 Rovaniemi, Finland

Abstract

Techniques for option valuation, both analytic and numeric, are well-established. Many of them cannot be used if the nature of the option is such that knowledge of the future is a prerequisite for valuation. For example, the value when the option expires may depend on the price of the underlying asset during the lifetime of the option. This paper describes a method of overcoming this problem by manipulating a symbolic parameter in a numeric scheme. This parameter characterises the option price at any stage in its lifetime as a function of the price history of the underlying asset. Certain options may then be valued by symbolic manipulations involving this parameter, such that it is not necessary to know a numerical value for a parameter in advance. This enables a limited amount of 'path-dependency' to be introduced into option valuation when using established numerical techniques, which has hitherto been very difficult or impossible.

1 Symbolic Computation in a Numeric Process

Options and other derivatives of financial instruments (stocks, bonds, currency exchanges etc) have been designed and traded for many years. They provide the purchaser with a means to stabilise money flows, speculate and hedge their positions. The reader is referred to references such as Willmott [1] or Hull [2] for the technical background and nomenclature of financial derivatives. This paper addresses particular problems in the valuation of an options which is a combination of two well-known types. These are options

with *Cash-or-nothing* and *Ramped* payoff functions. Let $f(S,t)$ be the value of an option on an underlying asset of value S at time t. If X is the exercise price, the value of a Ramped payoff is $Max(0, S - X)$ and the value of a Cash-or-nothing payoff is a fixed amount P if $S > X$ and zero otherwise. In general, valuation of an option with a given payoff will not be possible analytically, and numeric methods must be used. As a result, it has not been possible to use a symbolic parameter in a valuation process. This paper presents a general framework, and a specific example, for propagating a symbolic input parameter through a numeric iterative scheme such that the output is an implicit function of this input parameter. The Mathematica output expressions do not permit an explicit dependence of outputs on input parameters. Nevertheless, this analysis enables us to do three things. First, any decision about a numeric value of a parameter can be deferred until the output expression has been obtained. In this way, only one overall computation is needed. Second, the numeric value of a parameter can be calculated in order to fulfil a given condition. An element of path dependency can thus be introduced without knowing, in advance, the numeric values of independent input variables. In effect, the form of the solution is given by an implicit convolution *if < condition(input)> then output1 else output2* . Third, this method provides an analytic means to test for ill-conditioning of the system with respect to small changes in the numeric values of input parameters. This is better than using an range of input values with a complete recalculation each time.
The problem considered here is to switch between a Cash-or-nothing and a Ramped payoff Eurorean option at a given time during the option's lifetime. The switch depends on the value of the underlying asset at that time. This value cannot be known when the option is valued (hence the path-dependency). A symbolic parameter which represents the value of the cash-or-nothing payout at expiry is introduced, and its value is calculated such that the need to know the value of the underlying asset at switch time is obviated. This option is like a barrier/digital option.

2 Option valuation and the Black-Scholes Equation

It can be shown that $f(S,t)$, is given by a solution of the Black-Scholes equation (1), in which r is the risk-free interest rate, σ is the asset volatility and T is the lifetime of the option, in addition to the symbols already defined. Equation (1) and the given boundary and initial conditions are for a European ramped payoff option and apply for $0 \leq t \leq T$. An analytical solution is available by the method of images (see, e.g., Willmott [1]).

$$\frac{df}{dt} + rS\frac{df}{dS} + \frac{\sigma^2}{2}\frac{d^2 f}{dS^2} = rf \tag{1}$$

Transactions on Engineering Sciences vol 15, © 1997 WIT Press, www.witpress.com, ISSN 1743-3533

The initial and boundary conditions are $f(t,0) = Xe^{-r(T-t)}$, $f(t,\infty) = 0$ and $f(T,S) = Max(0, X-S)$.

Numerical methods are needed if a Black-Scholes model introduces non-linearity into (1) or the option is American. In practice, the principal numerical techniques employed to solve Black-Scholes equations are binary trees and finite difference schemes. Finite differences are used here and the solutions obtained can be compared with analytical and other numeric methods.

Many parabolic linear forms, such as (1) with the stated boundary and initial conditions, can be transformed into a standardised diffusion equation $\frac{dU}{d\tau} = \frac{d^2U}{dx^2}$, where U, x and τ are functions of f, S, t, r, X and σ. If this can be done, a finite difference scheme becomes purely numeric and it is not possible to inject a path dependent element into the analysis. In practice, untransformed Black-Scholes equations are used more often than not because more complicated Black-Scholes models cannot be standardised in this way. The following analysis is therefore used with an untransformed Black-Scholes equation.

3 Finite Difference Approximation

The terms $\frac{df}{dt}$, $\frac{df}{dt}$ and $\frac{d^2f}{dS^2}$ in the Black-Scholes equation can be replaced by standard approximations in an explicit backward finite difference scheme. The explicit nature of such a scheme make it possible to propagate symbolic terms through the scheme. If an implicit scheme is used it would require much more memory and run unacceptably slow. However, the accuracy of the explicit scheme is limited by the constraint $\mid \frac{\Delta x}{(\Delta t)^2} \mid < 0.5$. Using an explicit backward finite difference scheme, the discretised form of the Black-Scholes equation becomes, with, for the time step $0 \le i \le NN$, and for the asset price step $0 \le j \le M$

$$
\begin{aligned}
f_{i+1,j} &= a_{j-1}f_{i,j-1} + b_{j-1}f_{i,j} + c_{j-1}f_{i,j+1} \\
a_j &= (-\frac{rj\Delta t}{2} + \frac{\sigma^2 j^2 \Delta t}{2})/(1 + r\Delta t) \\
b_j &= (1 - \sigma^2 j^2 \Delta t)/(1 + r\Delta t) \\
c_j &= (\frac{rj\Delta t}{2} + \frac{\sigma^2 j^2 \Delta t}{2})/(1 + r\Delta t)
\end{aligned}
$$

An arbitrary payoff function, $PayOff[P, m, v]$ is used in place of the standard ramp payoff, $Max[0, X-S]$. P can be used to specify the numerical value of a parameter such as the strike price after all iterations, m represents the asset price at any stage in the iteration, and v is a parameter list which is reserved for any other use. Here, v is used to specify a cut-off point in time. The used of the arbitrary function $PayOff[P, m, v]$ results in increasingly complex, but unnested, symbolic expressions. The number

Transactions on Engineering Sciences vol 15, © 1997 WIT Press, www.witpress.com, ISSN 1743-3533

of terms in each one is limited by the asset price grid spacing. The cases considered here use an $(M+1)$ by $(NN+1)$ grid with $M = NN = 20$. This gives asset prices ranging from 0 to 100 in steps of 5 and 20 time steps, each of $\frac{1}{48}$ years. At the final time step, the *Valuation Profile*(i.e. the option value at valuation for the complete range of asset price) is obtained.

4 Mathematica Implementation

The following code forms the basis of the analysis, and is a straightforward implementation of the finite difference scheme, similar to the method of Hayes [3]. Outputs are not given.

```
dS:= 5; dt:= N[1/48]; r:= 0.1; sigma:= 0.4; M = 20; NN =
20;
a[j_]:= (-r j dt/2 + sigma^2 j^2 dt/2)/(1+r dt)
b[j_]:= (1 - sigma^2 j^2 dt)/(1+r dt)
c[j_]:= (r j dt/2 + sigma^2 j^2 dt/2)/(1+r dt)
```

The iterations are done by the function `BSBackwardStep`.

```
BSBackwardStep[ {oldF_, oldF_, TimeToExpiry_}, P_]:=
Block[{newF = oldF, j, te = TimeToExpiry},
      Do[ newF[[j]] =
          a[j-1] oldF[[j-1]] + b[j-1] oldF[[j]] +
          c[j-1] oldF[[j+1]] // Expand, {j,2,M} ];
      newF[[M+1]] = 0; (* at large S *)
      newF[[1]] = P E^(-r (dt+te)) ; (* at S=0 *)
      {newF, te+dt} ]
```

To obtain, in symbolic form, the Valuation Profile (at $t = T$), from an arbitrary payoff for a non-dividend paying European Put, the following definitions are made. The ramped payoff function is then defined, using a strike price of 50. The aim is to find valuations for $t = 0$ and $0 \leq S \leq 100$, which is given by `ValuationProfile` , and for the particular case $S = 50$, which is given by `ValuationProfile[[M/2+1]]`. The results agree with standard texts such as Hull [2].

```
POff = Table[PayOff[P,m,v], {m,0,20 }];
OptionPriceHistory = First /@
      NestList[BSBackwardStep[#,P]&, {POff,0},20];
ValuationProfile = Last[OptionPriceHistory];
PayOffRamp[P_,m_]:= P-5m /; 0≤m≤P/5
PayOffRamp[P_,m_]:= 0 /; P/5<m≤20
StrikePrice = 50;
```

```
ValuationProfile[[M/2+1]] /.
        {PayOff[P_,m_,v___]-> PayOffRamp[StrikePrice,m],
        a_. P -> StrikePrice a }
```

In order to value an option with a cash-or-nothing payoff instead, all that is necessary is to define an appropriate payoff function (below), and replace `PayOffRamp` by `PayOffCashOrNothing` .

```
  PayOffCashOrNothing[P_,m_]:= P /; 0≤m≤10
PayOffCashOrNothing[P_,m_]:= 0 /; 10<m≤20
```

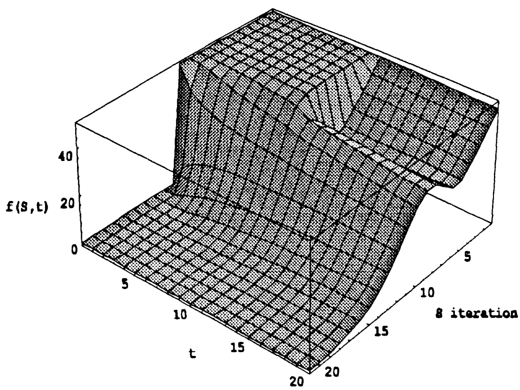
5 The Switch Option

Suppose that an option starts as a cash-or-nothing option. Half way through the option's life (time $\frac{NN}{2}$), the option switches to a ramp option if the asset price is 30 or less ($m \leq 6$), and stays a cash-or-nothing option otherwise. The buyer is betting on the higher payout provided by a cash-or-nothing option, provided that the asset price stays near the money. This option should be cheaper than a pure cash or nothing option but more expensive than a pure ramp option. The task for the writer is to construct the option so that the valuation is not too sensitive to changes to S, particularly near the money. This is good hedging practice. It amounts to choosing the cash or nothing payout, P, such that the valuation profile is monotonic decreasing. Since the writer cannot know how S will vary in the future, the algebra for calculating P must be incorporated into the valuation. This situation is analysed by using the parameter v to mark the position (in terms of time and asset price) of each option valuation in the grid with the option valuation itself. The payoff function is then replaced with a switching payoff function, which depends on the asset price at a given time step. The first step is to use the parameter v to associate the time of each option valuation with the option valuation itself. Next, we define the switch payoff function in terms of `PayOffCashOrNothing` and `PayOffRamp`. Then the arbitrary function `PayOff[P, m, v]` is replaced by the specific function `PayOffSwitch[P, m, v]`. In `PayOffSwitch`, P is expressed as a list. The first element in this list represents the Ramp case strike price and the second element in this list represents the Cash-or-Nothing case fixed payout. The last line produces a plot of the entire history surface of the option.

```
POff = Table[PayOff[P,m,v],m,0,M];
OptionPrices = First /@
   NestList[BSBackwardStep[#,P]&, {POff,0},NN] //Flatten;
OptionPriceTimeIndex =
```

```
   (Table[i, {i,0,NN }, {j,0,M }] //Flatten) ;
OptionPriceHistory = MapThread[f1,
   {OptionPrices,OptionPriceTimeIndex}];
PayOffSwitch[P_List,m_,v_]:=
   PayOffCashOrNothing[P[[2]],m] /; (v<NN/2)
PayOffSwitch[P_List,m_,v_]:=
   PayOffCashOrNothing[P[[2]],m] /; (v>=NN/2 && 6<m<=M )
PayOffSwitch[P_List,m_,v_]:=
   PayOffRamp[P[[1]],m] /; (v>=NN/2 && 0<=m<=6 )
StrikePrice = 50; CashOrNothingValue = 50;
OptionPriceHistory /. { PayOff[P_,m_,v___]->
   PayOffSwitch[{StrikePrice,CashOrNothingValue},m,v]};
OptionPriceHistorySwitched =
   Partition[% /. a_. + b_. P -> a + b StrikePrice, M+1];
VPSwitched = Last[OptionPriceHistorySwitched];
ListPlot[Transpose[{Table[i,{i,0,M}],VPSwitched }]]
ListPlot3D[OptionPriceHistorySwitched]
```

The history surface shows the discontinuity of the cash-or-nothing pay-out (trailing edge) at $t = 0$ and the maximum and minimum at the valuation profile for $t = T$ (leading edge). The switch discontinuity is also apparent.



The next task is to smooth the valuation profile such that it is monotonic decreasing. The effect of variation in S is then minimal.

6 A Price-Insensitive Switch Condition

The rationale behind this move is to minimise transaction costs when hedging.

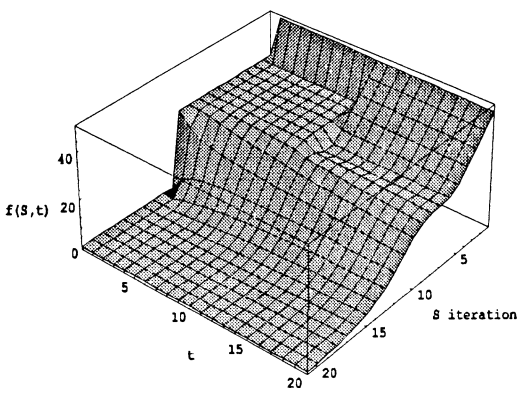
Near a local maximum on the Valuation Profile, if $f(S,t)$ decreases with decreasing S, the equivalent of a long asset position results because of the

Transactions on Engineering Sciences vol 15, © 1997 WIT Press, www.witpress.com, ISSN 1743-3533

decreasing asset price. This is hedged by going short the asset. But if $f(S,t)$ decreases with increasing S, the equivalent of a short asset position results because of the increasing asset price. This is hedged by going long the asset. A similar argument applies near a local minimum. Hence, at a local maximum or mininimum this strategy might mean many changes of position as the asset price changes. It is to be avoided by ensuring that there are no local maxima or minima on the valuation profile. In addition to the previous code we then need the following, in which the cash-or-nothing parameter is Q. From the result we pick the smallest positive Q. This will ensure that all the differences are negative and $f(S,t)$ is monotonic decreasing.

```
differences= (Rest[VPSwitched]-
            Drop[VPSwitched,-1]);
gradients = differences/dS //Expand;
Solve[0 == #]& /@ gradients;
solQ = First /@ %
QOptimal = Min[#[[1,2]]& /@ Cases[solQ,{Q->x_?Positive}]]
VPSwitched[[M/2+1]] /. Q->QOptimal
ListPlot[Transpose[{Table[i,{i,0,M}],VPSwitched /.
    Q->QOptimal}]]
ListPlot3D[OptionPriceHistorySwitched /. Q->QOptimal]
```

The value of `QOptimal` obtained from this analysis is 31.27, and the valuation to use for the option when the asset price is 50 is 16.46. We expect the switched option valuation to be in between the cash-or-nothing and the ramped payoff option. Valuations of 26.98 for the cash-or-nothing option and 4.01 for the ramped payoff option are obtained. The graph of the history surface follows. The relatively smaller cash-or-nothing payout, the inflexion in the valuation profile and switch discontinuity are all clear.



Transactions on Engineering Sciences vol 15, © 1997 WIT Press, www.witpress.com, ISSN 1743-3533

7 Conditioning Analysis

In order to test the stability if the switch option system, we show that the valuation profile is well-conditioned with respect to small changes, dQ, in Q. This analysis is not dependent on particular numeric values for Q or S. For each S and Q, we define a perturbation of the valuation profile and a local error measure $LocalError(S, Q, e) = ValuationProfile(S, Q + dQ) - ValuationProfile(S, Q)$. Next, a measure of the accumulated local errors is computed as follows.

$$TotalError(dQ) = \sqrt{\sum_{i=0}^{M}[LocalError(S_i, Q, dQ)]^2} \qquad (2)$$

```
VPPurturbed = VPSwitched /. Q->Q+dQ;
LocalErrors = Simplify[VPSwitched - VPPurturbed]
TotalError= Sqrt[Simplify[Plus @@ LocalErrors^2]] /.
            Sqrt[e1_^2]->e1
```

The result is $TotalError = 3.91857e$: the absolute change in $TotalError$ is about four times the change in Q. Hence, the system is absolutely well-conditioned with respect to small changes in Q. A similar analysis shows that the system is also relatively well-conditioned with respect to small changes in Q. Consequently, the system will be stable if, in practice, a near-optimal Q is chosen.

8 Conclusion

We have defined an option which can switch from a ramped payoff to a cash-or-nothing payoff under given conditions. It was valued by propagating a symbolic parameter through a finite difference scheme and then defining the meaning of the parameter in terms of the payoff. This single pass method produces an implicit parametric valuation which is stable with respect to small changes in the parameter. The use of symbolic computation in this way makes it unnecessary to carry out many numerical computations.

9 References

1. Willmott, P., Howison, S. and Dewynne, J. *The Mathematics of Financial Derivatives.* CUP. 1995.

2. Hull, J.C. *Futures Options and other Derivative Securities.* Prentice Hall. 1989.

3. Hayes, A. *Experiments in Exposition with Mathematica 3.0, Mathematica in Education and Research, Vol 5, No. 3, pp 3 -51.*Telos. 1997.