Note Title 1/28/2015

Details Binomial for European options

We need only S_{iN} $\quad 0 \le i \le N$

Note: $S_{iN} = S u^i d^{N-i} = S u^{2i-N}$, which can be computed efficiently as follows

Algorithm (to Compute S_{iN})

$A = u^2$

If N is even

$x = N/2$

$\Delta_x = S$

Notation $[x]$ = largest integer less than or equal to x.

Floor of x

$[1.3] = 1$

Comments

$\Delta_i = S_{iN}$

```
else
    x = ⌈N/2⌉ + 1
    s_x = S x
endif
for i = x+1, N
    s_i = s_{i-1} A
end
for i = x-1, 0
    s_i = s_i / A
end
```

N odd, then

$$2\left(\left\lceil \frac{N}{2} \right\rceil + 1\right) = N + 1$$

Output $s_i = S_{iN}$

Complexity $= O(N)$

Space $= O(N)$

If we only want the option value at $t=0$

Algorithm

Input: $F_i = f_{iN}$, $\alpha = e^{-r\delta t}$, p, $q = 1-p$

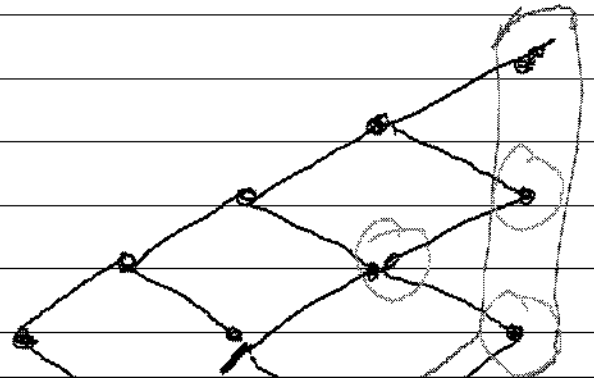
Output = $F_0 = f_{00}$

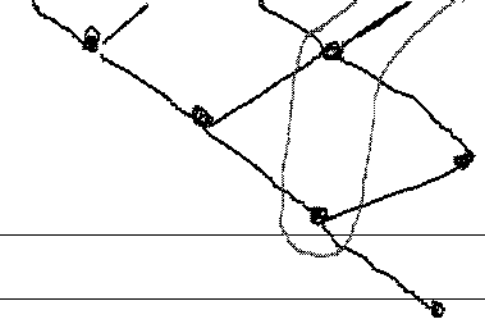
$x = \alpha p$

$y = \alpha q$

for $j = N-1, 0$

  for $i = 0, j$



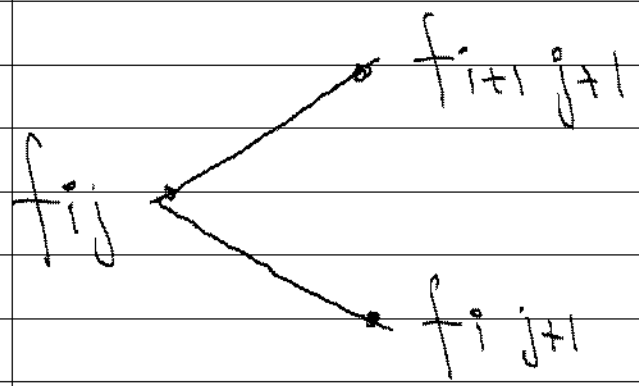


$$F_i = x F_{i+1} + y F_i$$

end

end

<u>American option pricing using binomial</u>



If it is call option

$$f_{ij} = \max\{S_{ij} - K, e^{-r\delta t}(p f_{i+1\,j+1} + q f_{i\,j+1})\}$$

if it is a put option

$$f_{ij} = \max\{K - S_{ij}, e^{-r\delta}(p f_{i+1\,j+1} + q f_{i\,j+1})\}$$

Algorithm to compute f_{ij}

Input: p, f_{iN}, $\alpha = e^{-r\delta t}$, $q = 1-p$, S_{ij}

Output: f_{ij}

$x = \alpha p$

$y = \alpha q$

for $j = N-1, 0$

for $i = 0, j$

If call then

$$f_{ij} = \max\{S_{ij} - K, \; x f_{i+1,j+1} + y f_{i,j+1}\}$$

else

$$f_{ij} = \max\{K - S_{ij}, \; x f_{i+1,j+1} + y f_{i,j+1}\}$$

end

end

Details: 1) If we only want the value of the option at

time $t=0$, proceed with the F_i as before

2) Now we need S_{ij} for all j and i

$$S_{ij} = S\,u^{2i-j} \qquad -N \le 2i-j \le N$$

Algorithm (to compute S_{ij})

Input: S, u

Output: Λ_i $\quad 0 \le i \le 2N$, $\quad S_{ij} = \Lambda_{2i-j+N}$

$x = N$

$\Lambda_x = S$

for $i = x+1, 2N$

Comments

$$\Lambda = \begin{bmatrix} S/u^N \\ S/u^{N-1} \\ \vdots \\ S \\ Su \\ \vdots \\ Su^N \end{bmatrix} = \begin{bmatrix} \Lambda_0 \\ \Lambda_1 \\ \Lambda_2 \\ \vdots \\ \Lambda_N \\ \vdots \\ \Lambda_{2N} \end{bmatrix}$$

Complexity $= 2N$

$s_i = u\, s_{i-1}$

end

for $i = x-1, 0$

$s_i = s_{i+1} / u$

end

Numerical analysis

Roundoff errors

Example : $x = 0.3452194$

$y = 0.3452196$

$y - x = 0.0000002$

Keep only 6 significant digits

$x \approx \hat{x} = 0.345219$ $\qquad \hat{y} - \hat{x} = 0.000001$

$y \approx \hat{y} = 0.345220$

Note: $\hat{y} - \hat{x} = 5(y - x)$

Substracting very close numbers may lead to losing significant digits of precision

Machine precision Let $x = 1 + \varepsilon$, $\varepsilon > 0$. Let $\bar{x}$ be the computer representation of x. If ε is very small, $\bar{x} = 1$. Think of the machine precision as the smallest ε such that

$\bar{x} > 1$.

Numerical problem: Input: x, Output: $y = f(x)$

actual input: $\bar{x}$, where $\bar{x} = x + \delta x$, with δx small

Algorithm A to solve the problem, which gives us $f_A(\bar{x})$

Def: A is stable if

$$\frac{\| f(x) - f_A(\bar{x}) \|}{\| f(x) \|} = O(\text{machine precision})$$

$\| \ \|$ is a norm (measures how big things are)

Def: Conditioning of a problem

$$\frac{\|f(x) - f(\bar{x})\|}{\|f(x)\|} \lesssim \kappa \frac{\|x - \bar{x}\|}{\|x\|}$$

κ = condition number

the problem is well conditioned if $\kappa = O(1)$ (not large)

κ large, then problem ill conditioned.