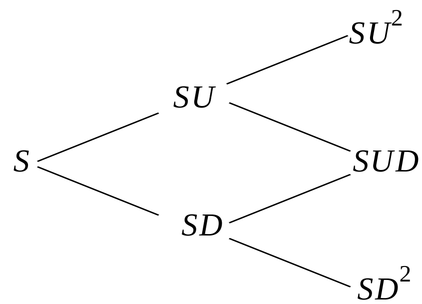
Numerical Solution of Black-Scholes Equation

1.0 MATLAB function BINPRICE (Binomial approach)

BINPRICE implements binomial method (for American options even though not explicitly mentioned in the documents) of the following form :



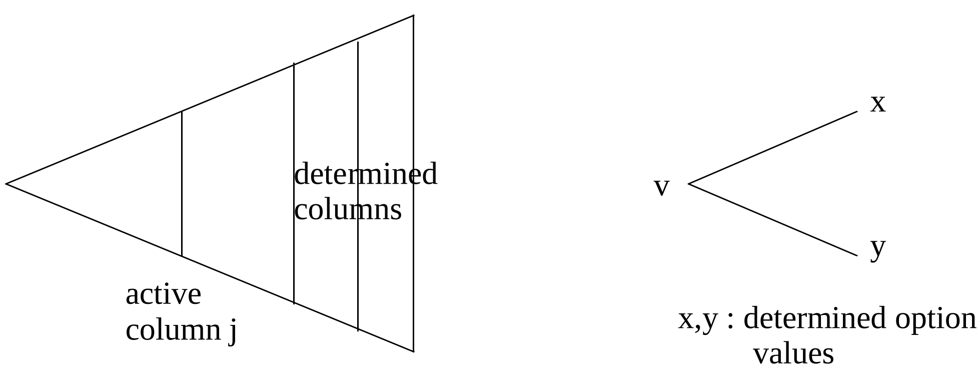
where $u = e^{\sigma\sqrt{\Delta t}}$, $d = \frac{1}{u} = e^{-\sigma\sqrt{\Delta t}}$.

Forward pass requires $O(n^2)$ time and space, but just 1 Matlab statement :

```
pr = triu(fliplr(flipud(so.*u.^jex.*d.^(iex-jex))));.
```

Reverse pass for option pricing goes backward in time, time-step by time-step.

Snap shot:



v is determined by $v = max\{e^{-r\Delta t}(px + (1-p)y), payoff(node(i,j))\}$ where $p = \frac{a-d}{u-d}$ and $a = e^{r\Delta t}$.

In case of American put option, $v = max\{e^{-r\Delta t}(px + (1-p)y), max(K - S(i,j), 0)\}$ since the payoff at (i, j) is $max(K - S(i,j), 0)$.

Matlab determines entire vector of option values corresponding to time j concurrently.

```
opt(:,n) = [max(x-pr(1:n,n),discopt);zeros(npp-n,1)];.
```

A few comments on dividends :

Either dividend rate or a list of dividends and ex-dividend date is taken as an input.

(1) Dividend rate q : $a = e^{(r-q)\Delta t}$

Matlab code

```
a = exp((r-q).*dt);
```

(2) Dividend list

i) Adjust S_0 : $\overline{S_0} = S_0 - \sum e^{-t_i r} div_i$

ii) Generate the stock lattice with $\bar{S}_0$ and add back the discounted dividends

Matlab code

```
pvdiv = div.*exp(-exdiv.*dt.*r);  % Find present value of all dividends
so = so-sum(pvdiv(:));         % Find current price - div present values
% Asset price at nodes, matrix is flipped so tree
% appears correct visually
pr = triu(fliplr(flipud(so.*u.^jex.*d.^(iex-jex))));
dpvtot = zeros(npp);          % Preallocate matrix
for y = 1:lenexdiv
  z = (exdiv(y):-1:0);        % Create vector from 0 to ex-div date
  dpv = div(y)*exp(-z*dt*r);  % Discount dividends nodes
  dpvmat = [dpv(ones(npp,1),:) zeros(npp,npp-length(dpv))];
  dpvtot = dpvtot + dpvmat;   % Add next discounted dividend to total
end
m = find(pr~=0);             % Find nodes where option will have value
pr(m) = pr(m)+dpvtot(m); % combine div pv's and prices to get new prices
end
```

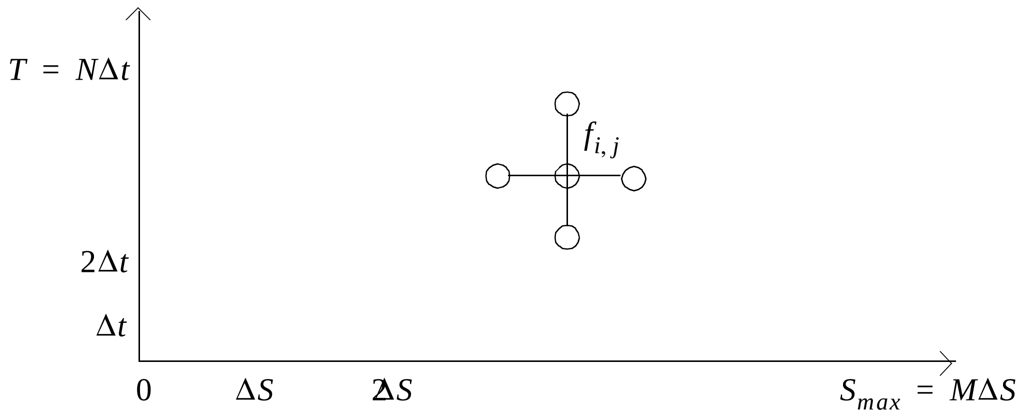
2.0 Finite difference method for Black -Scholes equation

First let's consider an European put option. The diffential equation that the option must satisfy is :

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = rf \qquad \textbf{(2.1)}$$

Suppose T is the maturity of the option and S_{max} is the maximum stock price.

Let $M\Delta S = S_{max}$ and $N\Delta t = \dot{T}$. $f_{i,j}$ denotes the option value at $(i\Delta t, j\Delta S)$.



2.1 Implicit FD

For an interior point (i, j),

Symmetric approximation to $\frac{\partial f}{\partial S}$: $\frac{f_{i,j+1} - f_{i,j-1}}{2\Delta S}$

Forward approximation to $\frac{\partial f}{\partial t}$: $\frac{f_{i+1,j} - f_{i,j}}{\Delta t}$

Approximation to $\frac{\partial^2 f}{\partial S^2}$: $\left(\frac{f_{i,j+1} - f_{i,j}}{\Delta S} - \frac{f_{i,j} - f_{i,j-1}}{\Delta S}\right)/\Delta S = \frac{f_{i,j+1} + f_{i,j-1} - 2f_{i,j}}{\Delta S^2}$

Substituting the above equations into B-S equations gives

$$\frac{f_{i+1,j} - f_{i,j}}{\Delta t} + rj\Delta S\frac{f_{i,j+1} - f_{i,j-1}}{2\Delta S} + \frac{1}{2}\sigma^2 j^2 \Delta S^2 \frac{f_{i,j+1} + f_{i,j-1} - 2f_{i,j}}{\Delta S^2} = rf_{i,j} \qquad \textbf{(2.2)}$$

By rearranging the terms, we have

$$a_j f_{i,j-1} + b_j f_{i,j} + c_j f_{i,j+1} = f_{i+1,j} \quad \text{for} \quad i = 1 : N-1 \text{ and } j = 1 : M-1 \qquad \textbf{(2.3)}$$

where

$$a_j = \frac{1}{2}rj\Delta t - \frac{1}{2}\sigma^2 j^2 \Delta t$$
$$bj = 1 + \sigma^2 j^2 \Delta t + r\Delta t$$
$$c_j = -\frac{1}{2}rj\Delta t - \frac{1}{2}\sigma^2 j^2 \Delta t$$

Now consider the boundary condition for an European put option.

(1) $f_{N,j} = max(K - j\Delta S), \quad j = 0, 1, ..., M$

(2) $f_{i,0} = K, \quad i = 0, 1, ..., N$

(3) Since $f \to 0$ as $S \to \infty$, choose $f_{i,M} = 0, \quad i = 0, 1, ..., N$

We have interior points and 1 boundary to be determined. Therefore, there are $(M-1)(N-1) + N - 1 = M(N-1)$ unknowns. We can solve for unknowns backward in time.

I.e., solve $i = N-1 : -1 : 0$

$$a_j f_{i,j-1} + b_j f_{i,j} + c_j f_{i,j+1} = f_{i+1,j} \quad \text{for} \quad f_{i,j}\ ,\ j = 1 : M-1$$

In matrix form,

$$\begin{bmatrix} b_1 & c_1 & & \\ a_2 & b_2 & c_2 & \\ & & \cdots & \\ & & a_{M-1} & b_{M-1} \end{bmatrix} \begin{bmatrix} f_{i,1} \\ f_{i,2} \\ : \\ f_{i,M-1} \end{bmatrix} = \begin{bmatrix} f_{i+1,1} - a_1 f_{i,0} \\ f_{i+1,2} \\ : \\ f_{i+1,M} - c_{M-1} f_{i,M} \end{bmatrix} \qquad \text{(2.4)}$$

Implicit method requires $O(M)$ work per line since the matrix is tridiagonal.

In case of the American put option, we need to compare the option value and the payoff of the opton and choose the bigger one, i.e., $f_{i,j} = max(f_{i,j}, K - S(i,j))$.

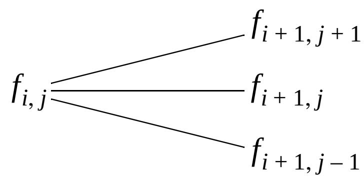
In the implicit finite difference scheme, $T_i \cdot \bar{f}_i = rhs(\bar{f}_{i+1})$ where T_i is tridiagonal matrix which is not necessarily symmetric. Total work needed amounts to $O(MN)$ and the space required is :

(i) $O(MN)$ if the whole surface on the grid is wanted,

(ii) $O(M)$ if only the current option value is wanted.

2.2 Explicit FD approach

To determine $f_{i,j}$, we need the three option values from the previous step.



We can get the approximations to the derivatives in a similar way.

$$\frac{\partial f}{\partial S} \approx \frac{f_{i+1,j+1} - f_{i+1,j-1}}{2\Delta S}$$

$$\frac{\partial^2 f}{\partial S^2} \approx \frac{f_{i+1,j+1} + f_{i+1,j-1} - 2f_{i+1,j}}{\Delta S^2}$$

$$\frac{\partial f}{\partial t} \approx \frac{f_{i+1,j} - f_{i,j}}{\Delta t}$$

After rearranging we have

$$_j = \tilde{a}_j f_{i+1,j-1} + \tilde{b}_j f_{i+1,j} + (\tilde{c} \perp j) f_{i+1,j} \text{ -} \quad \textbf{(2.5)}$$

where

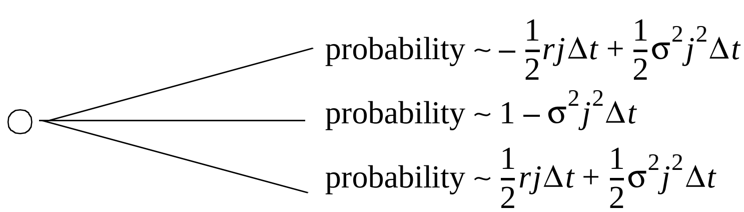
$$\tilde{a}_j = \frac{1}{1+r\Delta t}\left(-\frac{1}{2}rj\Delta t + \frac{1}{2}\sigma^2 j^2 \Delta t\right) \quad \textbf{(2.6)}$$

$$\tilde{b}_j = \frac{1}{1+r\Delta t}(1 - \sigma^2 j^2 \Delta t)$$

$$\tilde{c}_j = \frac{1}{1+r\Delta t}\left(\frac{1}{2}rj\Delta t + \frac{1}{2}\sigma^2 j^2 \Delta t\right)$$

Total work and space needed is the same as the implicit method. We need to compare American option price with its payoff at each grid point.

The explicit method can be viewed as a trinomial tree method :



where the discount factor is $\frac{1}{1+r\Delta t}$. Note that the sum of probabilities is 1.

But we can get negative probabilities unless we impose further restrictions on Δt and ΔS. This corresponds to the fact that the explicit method is unstable unless we impose further restrictions on Δt and ΔS.

2.3 Crank-Nicholson scheme

There is one more FD scheme which has the better convergence results : Crank-Nicholson scheme. It's the average of the explicit and implicit methods.

Recall the implicit method yields,

$$\frac{f_{i+1,j}-f_{i,j}}{\Delta t}+rj\Delta S\frac{f_{i,j+1}-f_{i,j-1}}{2\Delta S}+\frac{1}{2}\sigma^2 j^2\Delta S^2\frac{f_{i,j+1}+f_{i,j-1}-2f_{i,j}}{\Delta S^2} = rf_{i,j} \quad \textbf{(2.7)}$$

and the explicit method yields,

$$\frac{f_{i+1,j}-f_{i,j}}{\Delta t}+rj\Delta S\frac{f_{i+1,j+1}-f_{i+1,j-1}}{2\Delta S}+\frac{1}{2}\sigma^2 j^2\Delta S^2\frac{f_{i+1,j+1}+f_{i+1,j-1}-2f_{i+1,j}}{\Delta S^2} = rf_{i,j} \quad \textbf{(2.8)}$$

After adding and rearranging, we have

$$\left(\frac{\sigma^2 j^2}{2}-\frac{rj}{2}\right)f_{i,j-1}+\left(\frac{-2}{\Delta t}-\sigma^2 j^2-2r\right)f_{i,j}+\left(\frac{\sigma^2 j^2}{2}+\frac{rj}{2}\right)f_{i,j+1} = \quad \textbf{(2.9)}$$
$$\left(\frac{rj}{2}-\frac{\sigma^2 j^2}{2}\right)f_{i+1,j-1}+\left(\frac{-2}{\Delta t}+\sigma^2 j^2\right)f_{i+1,j}+\left(-\frac{rj}{2}-\frac{\sigma^2 j^2}{2}\right)f_{i+1,j+1}$$

Time and space required is of the same order as either the implicit the explicit method.

3.0 Option pricing as complementarity problems

3.1 Linear and nonlinear complementarity problems

Nonlinear complementarity problem is as follows :

Given $F : R^n \to R^n$ where F is continuous

Find $x \in R^n$ s.t. $x \geq 0, F(x) \geq 0, \quad x^T F(x) = 0$

Example - Consider an optimimization problem $min\{f(x) : x \geq 0\}$ where $f : R^n \to R^1$

The (local) optimality condition is $\nabla f(x^*) = \sum_{i,\, x_i^* = 0} \lambda_i e_i, \lambda_i \geq 0, x^* \geq 0,$

which is equivalent to $(x^*)^T \nabla f(x^*) = 0, x^* \geq 0, \nabla f(x^*) \geq 0$.

Note : The above is a necessary, but not a sufficient condition.

The following linear complementarity problem approach is useful for option pricing (especially for American put option).

LCP : Given $M \in M^{n \times n}$ and $q \in R^n$

Find $z \in R^n$ s.t. $q + Mz \geq 0, z \geq 0, z^T(q + Mz) = 0$

Or equivalently, find $v \geq 0, w \geq 0$ s.t. $w = q + Mz, z^T w = 0$.

Example - Consider a QP, i.e., $min\left\{c^T x + \frac{1}{2}x^T Hx : x \geq 0\right\}$ where H is a symmetric matrix.

The (local) optimality condition gives us $c + Hx^* \geq 0, x^* \geq 0, (c + Hx^*)^T x^* = 0$.

Example - The first order necessary condition for x to be a (local) solution of

$$min\left\{c^T x + \frac{1}{2}x^T Hx, Ax \geq b, x \geq 0\right\}$$

$$u = c + Hx - A^T y \geq 0, x \geq 0, x^T u = 0, v = -b + Ax \geq 0, y \geq 0, y^T v = 0$$

The above condition can be viewed as a linear complementarity problem if we let

$$q = \begin{bmatrix} c \\ -b \end{bmatrix}, M = \begin{bmatrix} H & -A^T \\ A & 0 \end{bmatrix}, \text{ and } z = \begin{bmatrix} x \\ y \end{bmatrix}$$

There are several methods to solve LCP.

1. Reformulate as a QP (not common). Note that LCP is not generally equivalent to QP.

2. Specialized simplex-like methods (Lemke’s method).

3. In some restrictive cases, we can reformulate LCP as LP. (M : Z-matrix).

4. Interior methods for LCP (new). See "Linear Complementarity Problem" by Cottle, Pang, and Stone.

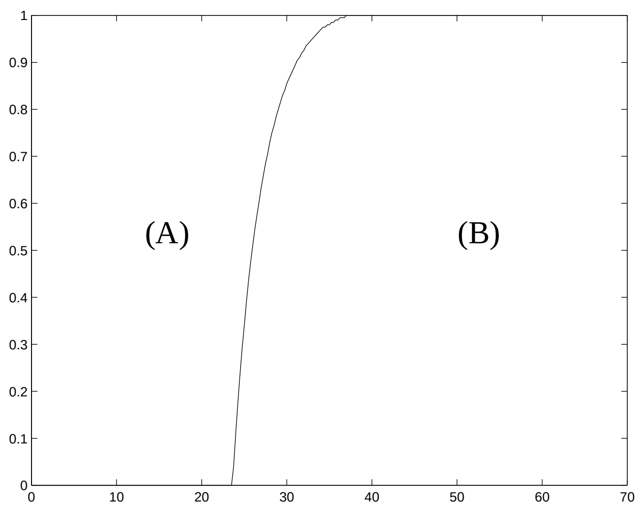
3.2 American put option pricing as LCP

The American put option value, f, satisfies the following with appropriate boundary conditions

$$-\frac{\partial f}{\partial t} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} - rS\frac{\partial f}{\partial S} + rf \geq 0 \quad \textbf{(3.1)}$$

$$f(S, t) - payoff \geq 0 \quad \textbf{(3.2)}$$

$$\left(-\frac{\partial f}{\partial t} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} - rS\frac{\partial f}{\partial S} + rf\right)(f(S, t) - payoff) = 0 \quad \textbf{(3.3)}$$



For example, (3.2) is an equality in (A) of the above figure and (3.1) is an equality in (B). The curve dividing (A) and (B) is called the free boundary.

After discretization and adding the appropriate boundary conditions (i.e. $f(0, t) = K$, $f(Smax, t) = 0$, and $f(S, T) = max(K - S, 0)$) the above can be formulated as LCP :

$$Mz + q \geq 0, z \geq 0, z^T(Mz + q) = 0$$

where $z = \tilde{z} - p, q = \tilde{q} + Mp$.

$\tilde{z}$ is a vector of option values, q is a vector describing the boundary conditon, and p is a vector containing the payoff of the American put option.

M is a block matrix of the form : $M = \begin{bmatrix} A_1 & & & & \\ B_2 & A_2 & & & \\ & B_3 & A_3 & & \\ & & \cdots\cdots & & \\ & & & B_m & A_m \end{bmatrix}$ and contains $O(mn)$ nonzeros.

Note that A_i's are tridiagonal matrices.