Black-Scholes PDE

The Black-Scholes equation for European derivative security with value U(x; t) is described as follows:

$$U_t(x,t) + \frac{1}{2}\sigma^2 x^2 U_{xx}(x,t) + rxU_x(x,t) - rU(x,t) = 0,$$

Where x is the price of the underline stock, t is time, and r is risk-free interest. With maturity time T and a suitable maximum possible price S_{max}, we can subdivide the intervals $0 < t < T$ and $0 < x < S_{max}$ into n and m subintervals, respectively. If K is the strike of a European put, we state additional conditions as

$$U(x,T) = \max(K - x, 0),$$
$$U(0,t) = Ke^{-r(T-t)}, \quad U(S_{\max}, t) = 0$$

With $h = \frac{S_{max}}{m}$ and $k = \frac{T}{n}$ we can form the subdivisions $0 = x_0 < x_1 < \ldots < x_m = S_{max}$ and $0 = t_0 < t_1 < t_2 < \ldots < t_n = T$ to get the following difference equations using the backward difference method.

$$U(x_i, t_{j+1}) = a_i U(x_{i-1}, t_j) + b_i U(x_i, t_j) + c_i U(x_{i+1}, j)$$

Where

$$a_i = \frac{k}{2}(ri - \sigma^2 i^2)$$
$$b_i = 1 + k(\sigma^2 i^2 + r)$$
$$c_i = \frac{k}{2}(-ri - \sigma^2 i^2)$$

$i = 1, 2, \ldots, m-1$ and $j = 0, 1, \ldots, n-1$ =. We may write this relation in matrix form as

$$
\begin{bmatrix} b_1 & c_1 & 0 & \dots & 0 \\ a_2 & b_2 & c_2 & \ddots & \vdots \\ 0 & a_3 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & c_{m-2} \\ 0 & \dots & 0 & a_{m-1} & b_{m-1} \end{bmatrix} \begin{bmatrix} w_{1,j} \\ w_{2,j} \\ \vdots \\ \vdots \\ w_{m-1,j} \end{bmatrix} = \begin{bmatrix} w_{1,j+1} \\ w_{2,j+1} \\ \vdots \\ \vdots \\ w_{m-1,j+1} \end{bmatrix} - \begin{bmatrix} a_1 w_{0,j} \\ 0 \\ \vdots \\ 0 \\ c_{m-1} w_{m,j} \end{bmatrix}
$$

Here is the Matlab function Value=EuroPut(S0,K,r,sigma,T,Smax,n,m)
that estimates the price of a put option using the backward-difference method:

```
function Value=EuroPut(S0,K,r,sigma,T,Smax,n,m)
h=Smax/m;
k=T/n;
% create boundary conditions
x=0:h:Smax;
wxT(1,:)= max(K-x,0);
t=0:k:T;
w0t(:,1)= K*exp(-r*(T-t));
% set up a
i=1:m-1;
a=(k/2)*(r*i-sigma^2*i.^2);
% set up b
b=1+k*(sigma^2*i.^2+r);
% set up c
c=(k/2)*(-r*i-sigma^2*i.^2);
% set up matrix A, Dimension:(m-1)*(m-1)
A=zeros(m-1);
for i=1:m-1
A(i,i)=b(i);
end
for i=1:m-2
A(i+1,i)=a(i+1);
A(i,i+1)=c(i);
end
% set up boundary conditions
S=zeros(n+1,m+1);
S(n+1,:)=wxT;
S(:,1)=w0t;
S(1,m+1)=0;
% set up the estimates
for j=n:-1:1
b=[S(j+1,2), S(j+1,3:m-1), S(j+1,m)]-[a(1)*S(j,1), zeros(1,m-3), c(m-1)*S(j,m+1)];
S(j,2:m)=SolSystemCrout(A,b);
end
% if the initial asset price does not lie on the grid, we must interpolate
```

```
% between the two neighboring points.
Value=interp1(x,S(1,:),S0);
```

The estimation for a European put option price with $S0 = 50$, $K = 50$, $r = 0.1$, $\sigma = 0.4$, $T = 5/12$ using $S_{max} = 100$, $n = 100$ and $m = 50$ is:

Estimate = 4.05447923827109

The price of a put at time 0 in the Black-Scholes formula is

Exact =4.07598098478778