1

Let us denote by $C(E,t_2,S,t_1)$ the value of a call option with exercise price $E$, maturity time $t_2$, stock price at time $t_1 < t_2$ equal to $S$. We want to calculate $C(S_{T_1},T_1+T_2, S_{T_1}, T_1)$. From the Black-Scholes formula we get:

$$C(S_{T_1}, T_1+T_2, S_{T_1}, T_1) = S_{T_1} N(d_1) - S_{T_1} e^{-rT_2} N(d_2) = S_{T_1} (N(d_1) - e^{-rT_2} N(d_2)),$$

where

$$d_1 = \frac{\log(S_{T_1}) + (r + \frac{1}{2}\sigma^2)T_2}{\sigma \sqrt{T_2}} = \frac{r + \frac{1}{2}\sigma^2}{\sigma} \sqrt{T_2},$$

$$d_2 = \frac{r - \frac{1}{2}\sigma^2}{\sigma} \sqrt{T_2}.$$

Thus

$$C(S_{T_1}, T_1+T_2, S_{T_1}, T_1) = \frac{S_{T_1}}{S_0} C(S_0, T_2, S_0, 0),$$

where $C(S_0, T_2, S_0, 0)$ is the value at $t = 0$ of a European call option with strike price $S_0$ (at-the-money option) and expiry at $T_2$.

Denote $c = C(S_0, T_2, S_0, 0)$ and notice that this value is a known constant.

Consider the portfolio consisting of $\frac{c}{S_0}$ shares of the asset $S_t$. At $t = T_1$ the value of this portfolio is $\frac{c}{S_0} S_{T_1}$, which is exactly $C(S_{T_1}, T_1+T_2, S_{T_1}, T_1)$ and is the value of the forward start call option at $t = T_1$.

Thus for $0 \leq t \leq T_1$ the value of the forward start call option must be equal to the value of the portfolio (using the no arbitrage assumption). This value is: $\frac{c}{S_0} S_t$.

Thus for $0 \leq t \leq T_1$:

$$V(S_t, t) = \frac{c}{S_0} S_t = S_t [N(d_1) - e^{-rT_2} N(d_2)],$$

with $d_1, d_2$ given above.
We will use the notation from problem 1. \( C(S, t, E, T) \) will be European call option value with strike \( E \), expiry \( T \), stock price \( S \) at time \( t \). Same notation for the put option: \( P(S, t, E, T) \).

At time \( T \), since the payoff for the call option is \( \text{Payoff}(C) = \max(S_T - E, 0) \), if we consider the case \( S \to \infty \), then we will certainly exercise the put. Now consider holding \( e^{-rT} \) shares of the underlying asset at time \( t = 0 \) and reinvesting the dividend payment into purchasing the underlying asset immediately. Since the asset pays continuous dividend yield \( q \), at time \( T \) we will have one unit of the asset \( S_T \). Taking this into account and also the time-value of money, we get that \( C(S, 0, E, T) = Se^{rT} - Ee^{-rT} \). Also \( C(S, t, E, T) = Se^{(T-t)} - Ee^{-r(T-t)} \). In the case of a put option, the payoff is \( \text{Payoff}(P) = \max(E - S_T, 0) \), so if \( S \to \infty \) we will not exercise the option, so its value at time 0 is 0: \( P(S, t, E, T) = 0 \).

Now let's analyse the situation when \( S = 0 \). Looking at the payoff formula, we see that in this case we will not exercise the call option. This means that its value at time 0 is equal to 0: \( C(0, t, E, T) = 0 \). Also from its corresponding payoff we see that we should exercise the put option, and the payoff at time \( T \) will be equal to \( E \). From time-value of money we get that the value of the put option should be \( P(0, t, E, T) = Ee^{-r(T-t)} \).

This is not the only way to find the boundary values. We observe that we could only calculate the values for the call options and we could get the values of the put options from the Put-Call parity.

We could also use the Black-Scholes formulae:

\[
C(S, t, E, T) = Se^{-\sigma^2(T-t)}N(d_1) - Ee^{-r(T-t)}N(d_2),
\]
\[
P(S, t, E, T) = Ee^{-r(T-t)}N(-d_2) - Se^{-\sigma^2(T-t)}N(-d_1),
\]

where

\[
d_1 = \log\left(\frac{S}{E}\right) + \frac{(r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}},
\]
\[
d_2 = \log\left(\frac{S}{E}\right) + \frac{(r - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}.
\]

We see that as \( S \to \infty \) we get \( d_1 = d_2 = \infty \) and replacing in the Black-Scholes formulae we get the same result as above:

\[
C(S, t, E, T) = Se^{-\sigma^2(T-t)} - Ee^{-r(T-t)}.
\]
\[
P(S, t, E, T) = 0.
\]

If we consider the case \( S = 0 \) we get \( d_1 = d_2 = -\infty \) and Black-Scholes implies:

\[
C(0, t, E, T) = 0,
\]
\[ P(0, t, E, T) = E e^{-r(T - t)}. \]

(b) The Crank-Nicolson scheme for the Black-Scholes equation

\[
\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2(S) S^2 \frac{\partial^2 V}{\partial S^2} + (r - q) S \frac{\partial V}{\partial S} - rV = 0
\]

involves averaging over the implicit and explicit finite difference equations. The implicit finite difference equations are as follows: for \( i = 2, \ldots, N, j = 2, \ldots, M + 1 \)

\[
\sigma(S)^2 S^2 \frac{\partial^2 V}{\partial S^2} \approx (\sigma_i)^2 S^i \left( \frac{V_{i+1}^j - 2V_i^j + V_{i-1}^j}{\delta^2} \right),
\]

\[
S \frac{\partial V}{\partial S} \approx S_i \left( \frac{V_{i+1}^j - V_i^j}{2\delta} \right).
\]

And the explicit difference equations are: for \( i = 2, \ldots, N, j = 2, \ldots, M + 1 \)

\[
\sigma(S)^2 S^2 \frac{\partial^2 V}{\partial S^2} \approx (\sigma_i)^2 S^i \left( \frac{V_{i+1}^{j-1} - 2V_i^{j-1} + V_{i-1}^{j-1}}{\delta^2} \right),
\]

\[
S \frac{\partial V}{\partial S} \approx S_i \left( \frac{V_{i+1}^{j-1} - V_i^{j-1}}{2\delta} \right).
\]

Now taking a backward difference in time and averaging over the explicit and implicit finite differences, we get: for \( i = 2, \ldots, N, j = 2, \ldots, M + 1 \)

\[
\frac{V_i^j - V_{i-1}^{j-1}}{\delta t} + \frac{1}{2} \left( \frac{1}{2} (\sigma_i)^2 S_i^2 \left( \frac{V_{i+1}^j - 2V_i^j + V_{i-1}^j}{\delta^2} \right) + \frac{1}{2} (\sigma_i)^2 S_i^2 \left( \frac{V_{i+1}^{j-1} - 2V_i^{j-1} + V_{i-1}^{j-1}}{\delta^2} \right) \right) +
\]

\[
\frac{1}{2} \left( (r - q) S_i \frac{V_{i+1}^j - V_i^j}{2\delta} + (r - q) S_i \frac{V_{i+1}^{j-1} - V_i^{j-1}}{2\delta} \right) - \frac{1}{2} r (V_i^j + V_i^{j-1}) = 0.
\]

Simplifying and rearranging terms to find the coefficients of the unknowns \( V_{i-1}^j, V_i^j \) and \( V_{i+1}^j \) we get the equation

\[
\left( \frac{1}{4} (r - q) \frac{\delta t}{\delta S} S_i - \frac{1}{4} (\sigma_i)^2 \frac{\delta t}{\delta S^2} S_i^2 \right) V_{i-1}^{j-1} + \left( 1 + \frac{1}{2} r \delta t + \frac{1}{2} (\sigma_i)^2 \frac{\delta t}{\delta S^2} S_i^2 \right) V_i^{j-1} +
\]

\[
\left( -\frac{1}{4} (r - q) \frac{\delta t}{\delta S} S_i - \frac{1}{4} (\sigma_i)^2 \frac{\delta t}{\delta S^2} S_i^2 \right) V_{i+1}^{j-1} = V_i^j + \frac{1}{4} (\sigma_i)^2 S_i^2 \delta t \left( \frac{V_{i+1}^j - 2V_i^j + V_{i-1}^j}{\delta^2} \right) +
\]

\[
\frac{1}{2} (r - q) S_i \delta t \left( \frac{V_{i+1}^j - V_i^j}{2\delta} \right) - \frac{1}{2} r \delta t V_i^j.
\]

For the above equation, let \( e_i, d_i \) and \( f_i \) be the coefficients of \( V_{i-1}^j, V_i^j \) and \( V_{i+1}^j \) respectively (we observe that these coefficients are independent of time, so they do not depend on \( j \), only on \( i \)), and let \( b_i^{j-1} \) denote the RHS. Now for each \( j = 2, \ldots, M + 1, i = 2, \ldots, N \) we have equations of the form:

\[
e_i V_{i-1}^j + d_i V_i^j + f_i V_{i+1}^j = b_i^j.
\]
Note however that for the equations where \( i = 1 \) and \( i = N + 1 \), the quantities \( V^i_j \) and \( V^i_{N+1} \) are known from the boundary conditions and so we adjust the equations accordingly by setting

\[

b^i_1 = V^i_1 + \frac{1}{4}(\sigma_1)^2 S^2 \delta t \left( \frac{V^i_1 - 2V^i_2 + V^i_3}{\delta S^2} \right) - \frac{1}{4}(r - q)S_2 \delta t \left( \frac{V^i_3 - V^i_1}{\delta S} \right) - \frac{1}{2}r \delta t V^i_2 - e_2 V^{i-1}_1.
\]

\[ b^{i-1}_N = V^{i-1}_N + \frac{1}{4}(\sigma_N)^2 S^2 \delta t \left( \frac{V^{i-1}_N - 2V^{i-1}_N + V^{i-1}_{N+1}}{\delta S^2} \right) + \frac{1}{4}(r - q)S_N \delta t \left( \frac{V^{i-1}_{N+1} - V^{i-1}_N}{\delta S} \right) - \frac{1}{2}r \delta t V^{i-1}_N = f_N V^{i-1}_{N+1}.
\]

Now setting up the system of equations in matrix-vector form we get the desired result where \( x^{i-1} = [V^{i-1}_2, \ldots, V^{i-1}_N] \),

\[

e = [e_2, \ldots, e_N], \quad d = [d_2, \ldots, d_N] \quad f = [f_2, \ldots, f_N], \quad b^i = [b^i_1, \ldots, b^i_N]
\]

\[

A = \text{diag}(e(2 : N - 1), -1) + \text{diag}(d, 0) + \text{diag}(f(1 : N - 2), 1)
\]

and solve the system \( Ax^{i-1} = b^i \) for each \( j = M + 1, \ldots, 2 \).

(c)function U=BSPDEuro(S0,r,q,Smax,T,M,N,E,flag,sigfcn,param)

k=length(E);

dt=T/M;

dS=Smax/N;

Tdist=dt*(0:1:M);

Sdist=dS*(0:1:N);

term1=.5*(r-q)*(dt/dS)*Sdist(2:N);

term2=.5*(dt/dS.^2).*Sdist(2:N).^2);

sig=feval(sigfcn,Sdist(2:N),param));

e=term1-term2.*(sig(1:N-1).^2);

d=1+5*r*dt+2*term2.*(sig(1:N-1).^2);

f=-term1-term2.*(sig(1:N-1).^2);

[l,u]=TriDiLU(d, e, f);

for s=1:k

if flag==1

V=max(Sdist-E(s),0);

else

V=max(E(s)-Sdist,0);

end

for j=M+1:-1:2

if flag==1

a=0;

c=Smax*exp(-q*Tdist(j-1))-E(s)*exp(-r*Tdist(j-1));

else

end
\[ a = E(s) \exp(-rT \text{dist}(j-1)) \]
\[ c = 0 \]
end

\[ b_j(1:(N-1)) = V(2:N) * (1-.5*r*dt) + \text{term2} * (\text{sig}(1:N-1) \cdot 2 + (V(1:N-1) - V(2:N)) \cdot \text{term1} \cdot (V(3:N+1) - V(1:N-1))) \]
\[ b_j(1) = b_j(1) \cdot e(1) \cdot a \]
\[ b_j(N-1) = b_j(N-1) \cdot f(N-1) \cdot c \]
\[ x = \text{LBidiSol}(l,b_j) \]
\[ V(2:N) = \text{UBidiSol}(u,f,x) \]
\[ V(1) = a \]
\[ V(N+1) = c \]
end

l = \text{find}(S_0 < S_{\text{dist}}); 
\]
\[ j = l(1); 
\]
\[ U(s) = V(j-1); 
\]
end

function \[ \text{sig} = \text{absoluteDiffusion}(S, \alpha) \]
\[ \text{sig} = \alpha / S \]

function \[ \text{sig} = \text{quadraticVol}(S, x) \]
\[ \text{sig} = x(1) * S \cdot 2 + x(2) * S + x(3) \]

\[ E = 90:2:110; \]
\[ n = \text{length}(E); \]
\[ U = \text{zeros}(n,1); \]
\[ U = \text{BSPDEuro}(100, 03, 02, 200, 1, 100, 200, E, 1, '\text{absoluteDiffusion}', 30) \]

<table>
<thead>
<tr>
<th>Strike Price</th>
<th>Option Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>17.7794</td>
</tr>
<tr>
<td>92</td>
<td>16.5567</td>
</tr>
<tr>
<td>94</td>
<td>15.3831</td>
</tr>
<tr>
<td>96</td>
<td>14.2595</td>
</tr>
<tr>
<td>98</td>
<td>13.1865</td>
</tr>
<tr>
<td>100</td>
<td>12.1647</td>
</tr>
<tr>
<td>102</td>
<td>11.1942</td>
</tr>
<tr>
<td>104</td>
<td>10.2750</td>
</tr>
<tr>
<td>106</td>
<td>9.4069</td>
</tr>
<tr>
<td>108</td>
<td>8.5894</td>
</tr>
<tr>
<td>110</td>
<td>7.8220</td>
</tr>
</tbody>
</table>

(d) function \[ y = \text{myfun}(\text{param}, S_0, r, q, S_{\text{max}}, T, M, N, E, \text{flag}, \text{sigfcn}, U); \]
\[ y = U - \text{BSPDEuro}(S_0, r, q, S_{\text{max}}, T, M, N, E, \text{flag}, \text{sigfcn}, \text{param}); \]
E=90:2:110;
U=BSPDEuro([100,03,,02,200,1,100,200,E,1,'absoluteDiffusion',30]);
x0= [0 0 .3];
x=lsqnonlin('myfun',x0,-inf*ones(3,1),inf*ones(3,1),[],S0,q,Smax,T,M,N,E,flag,'quadraticVol',U);

![Figure 1: Volatility comparison](image)

We observe that our approximation is close; however, the estimated volatility is concave whereas the original volatility function is convex.

<table>
<thead>
<tr>
<th>local volatility</th>
<th>estimated volatility</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.8529</td>
<td>13.9276</td>
</tr>
<tr>
<td>12.5619</td>
<td>12.6596</td>
</tr>
<tr>
<td>11.3403</td>
<td>11.4577</td>
</tr>
<tr>
<td>10.1903</td>
<td>10.3230</td>
</tr>
<tr>
<td>9.1131</td>
<td>9.2563</td>
</tr>
<tr>
<td>8.1097</td>
<td>8.2579</td>
</tr>
<tr>
<td>7.1801</td>
<td>7.3277</td>
</tr>
<tr>
<td>6.3239</td>
<td>6.4652</td>
</tr>
<tr>
<td>5.5400</td>
<td>5.6697</td>
</tr>
<tr>
<td>4.8266</td>
<td>4.9397</td>
</tr>
<tr>
<td>4.1814</td>
<td>4.2737</td>
</tr>
</tbody>
</table>

3

(a) At time $T$ since Payoff($P$) = $\max(E - S_T, 0)$ and $E \to \infty$, we will certainly exercise the put. Now consider holding $e^{-qT}$ units of the underlying asset at time $t = 0$ and reinvesting the dividend payment into purchasing the underlying asset immediately. Since the asset pays a continuous dividend with dividend yield $q$, at time $T$, we have one unit of the asset $S_T$. Taking this and time-value of money into account, we get that $P(S_0, 0, E, T) =$
\[
E e^{-rT} - S_0 e^{-qT}. \text{ In general } P(S_0, t, E, T) = E e^{-r(t-T)} - S_0 e^{-q(t-T)} \text{. From put-call parity (and not only) we get that the value of the call is } C(S_0, t, E, T) = 0. \\

Other conditions are when \( E = 0 \), the put will never be exercised and so it has zero value at all times. Thus \( P(S, t, 0, T) = 0 \) for all \( T \). Also \( C(S_0, 0, E, 0) = \max(S_0 - E, 0) \).

Finally at \( T = 0 \), the value of the put is just its payoff, i.e. \( P(S_0, 0, E, 0) = \max(E - S_0, 0) \) and also the call \( C(S_0, 0, E, 0) = \max(S_0 - E, 0) \).

(b) The Crank-Nicolson scheme for the Black-Scholes equation

\[
\frac{\partial V}{\partial t} = \frac{1}{2} \sigma(E, T)^2 E^2 \frac{\partial^2 V}{\partial E^2} + (r - q) E \frac{\partial V}{\partial E} + qV = 0
\]

involves averaging over the implicit and explicit finite difference equations.

The implicit finite difference equations are as follows: for \( i = 1, \ldots, N - 1, j = 1, \ldots, M \)

\[
\sigma(E, T)^2 E^2 \frac{\partial^2 V}{\partial E^2} \approx (\sigma_i^j)^2 E^2_i \frac{V_{i+1}^j - 2V_i^j + V_{i-1}^j}{\delta E^2}
\]

and

\[
E_i \frac{\partial V}{\partial E} \approx E_i \frac{V_{i+1}^j - V_{i-1}^j}{2\delta E}
\]

And the explicit difference equations are: for \( i = 1, \ldots, N - 1, j = 1, \ldots, M \)

\[
\sigma(E, T)^2 E^2 \frac{\partial^2 V}{\partial E^2} \approx (\sigma_i^{j-1})^2 E^2_i \frac{V_{i+1}^{j-1} - 2V_i^{j-1} + V_{i-1}^{j-1}}{\delta E^2}
\]

and

\[
E_i \frac{\partial V}{\partial E} \approx E_i \frac{V_{i+1}^{j-1} - V_{i-1}^{j-1}}{2\delta E}
\]

Now taking a backward difference in time and averaging over the explicit and implicit finite differences, we get: for \( i = 1, \ldots, N - 1, j = 1, \ldots, M \)

\[
\frac{V_i^j - V_{i-1}^{j-1}}{\delta T} - \frac{1}{2} \left( \frac{1}{2} (\sigma_i^j)^2 E_i^2 \left( \frac{V_{i+1}^j - 2V_i^j + V_{i-1}^j}{\delta E^2} \right) + \frac{1}{2} (\sigma_i^{j-1})^2 E_i^2 \left( \frac{V_{i+1}^{j-1} - 2V_i^{j-1} + V_{i-1}^{j-1}}{\delta E^2} \right) \right) + \frac{1}{2} \left( \frac{r - q}{\delta T} E_i \frac{V_{i+1}^j - V_{i-1}^j}{2\delta E} + \frac{r - q}{\delta T} E_i \frac{V_{i+1}^{j-1} - V_{i-1}^{j-1}}{2\delta E} \right) + \frac{1}{2} q (V_i^j + V_i^{j-1}) = 0.
\]

Simplifying and rearranging terms to find the coefficients of the unknowns \( V_{i-1}^j \), \( V_i^j \) and \( V_{i+1}^j \) we get the equation

\[
\left( -\frac{1}{4} (r - q) \frac{\delta T}{\delta E} E_i - \frac{1}{4} (\sigma_i^j)^2 \frac{\delta T}{\delta E^2} E_i^2 \right) V_i^j + \left( 1 + \frac{1}{2} q \frac{\delta T}{\delta E} + \frac{1}{2} (\sigma_i^j)^2 \frac{\delta T}{\delta E^2} E_i^2 \right) V_{i-1}^j = \frac{1}{2} \left( \frac{r - q}{\delta E^2} E_i \left( \frac{V_{i+1}^j - 2V_i^j + V_{i-1}^j}{\delta E^2} \right) + \frac{r - q}{\delta E^2} E_i \left( \frac{V_{i+1}^{j-1} - 2V_i^{j-1} + V_{i-1}^{j-1}}{\delta E^2} \right) \right) + \frac{1}{2} q (V_i^j + V_i^{j-1}).
\]
$$\left( \frac{1}{4} (r - q) \frac{\delta T}{E} E_i - \frac{1}{4} (\sigma_i^2)^2 \frac{\delta T}{E^2} E_i^2 \right) V_{i+1}^{j} = V_i^{j-1} + \frac{1}{4} (\sigma_i^{j-1})^2 E_i^2 \delta T \left( \frac{V_{i+1}^{j-1} - 2V_i^{j-1} + V_{i-1}^{j-1}}{\delta E} \right)$$

$$= \frac{1}{2} (r - q) E_i \delta T \left( \frac{V_{i+1}^{j-1} - V_i^{j-1}}{2\delta E} \right) - \frac{1}{2} q \delta T V_i^{j-1}.$$ 

For the above equation, let $c_j$, $d_j$ and $f_j$ be the coefficients of $V_{i-1}^{j}$, $V_i^{j}$ and $V_{i+1}^{j}$ respectively, and let $b_i^{j-1}$ denote the RHS. Now for each $j = 1, \ldots, M$, $i = 1, \ldots, N - 1$ we have equations of the form

$$c_i^j V_{i-1}^{j} + d_i^j V_i^{j} + f_i^j V_{i+1}^{j} = b_i^{j-1}.$$ 

Note however that for the equations where $i = 1$ and $i = N - 1$, the quantities $V_0^j$ and $V_N^j$ are known from the boundary conditions and so we adjust the equations accordingly by setting

$$b_1^{j-1} = V_1^{j-1} + \frac{1}{4} (\sigma_1^{j-1})^2 E_1^2 \delta T \left( \frac{V_2^{j-1} - 2V_1^{j-1} + V_0^{j-1}}{\delta E} \right) - \frac{1}{2} (r - q) E_1 \delta T \left( \frac{V_2^{j-1} - V_1^{j-1}}{2\delta E} \right) - \frac{1}{2} q \delta T V_1^{j-1} - c_1^j V_0^j,$$

$$b_{N-1}^{j-1} = V_{N-1}^{j-1} + \frac{1}{4} (\sigma_{N-1}^{j-1})^2 E_{N-1}^2 \delta T \left( \frac{V_N^{j-1} - 2V_{N-1}^{j-1} + V_{N-2}^{j-1}}{\delta E} \right) - \frac{1}{2} (r - q) E_{N-1} \delta T \left( \frac{V_N^{j-1} - V_{N-1}^{j-1}}{2\delta E} \right) - \frac{1}{2} q \delta T V_{N-1}^{j-1} - f_{N-1}^j V_N^j.$$ 

Now setting up the system of equations in matrix-vector form we get the desired result where $x^j = [V_1^j, \ldots, V_{N-1}^j]$,

$$c^j = [c_1^j, \ldots, c_{N-1}^j], d^j = [d_1^j, \ldots, d_{N-1}^j], f^j = [f_1^j, \ldots, f_{N-1}^j] and b^{j-1} = [b_1^{j-1}, \ldots, b_{N-1}^{j-1}]$$

$$A^j = diag(c^j(2 : N - 1), -1) + diag(d^j, 0) + diag(f^j(1 : N - 2), 1),$$

and solve the system $A^j x^j = b^{j-1}$ for each $j = 1, \ldots, M$.

function U=FwdBS(i0,r,q,Emax,Tmax,E,M,N,flag,sigfcn,param)

% Computes European option values with different strike prices and
% expiries given volatility sigfcn, risk-free interest rate (r)
% and dividend yield (q).

dT=Tmax/M;
dE=Emax/N;
Tdist=dT*(0:1:M);
Edist=dE*(0:1:N);
sig=feval(sigfcn,Edist,param);

% Setup boundary and initial conditions from part(b)
if flag==1
  V(1,1:M+1)=S0*exp(-q*Tdist);
  V(1:N+1,M+1)=max(S0-E,0);
  V(N+1,1:M+1)=0;
else
  V(1,1:M+1)=0;
  V(1:N+1,M+1)=max(E-S0),0);
  V(N+1,1:M+1)=Emax*exp(-r*Tdist)-S0*exp(-q*Tdist);
end

% Now find the put option values over the discretized grid

term1=0.25*(r-q)*(dT/dE)*Edist(2:N);
term2=0.25*(dT/dE^2).*Edist(2:N,^2)

e=-term1-(sig(2:N).^2).*term2;
d=1 + 0.5*q*dT + 2*(sig(2:N).^2).*term2;
f=term1-(sig(2:N).^2).*term2;
[l,u]=TriDiLU(d, e, f);
for j=2:1:M+1
  bj(1:N-1)=V(2:N,j-1)*(1-0.5*dT*q)+(sig(2:N,j-1).^2).*term2.*(V(3:N+1,j-1)-...
  2*V(2:N,j-1)+V(1:N-1,j-1)-term1.*(V(3:N+1,j-1)-V(1:N-1,j-1)));
  bj(1)=bj(1)-e(1).*V(1,j);
  bj(N-1)=bj(N-1)-f(N-1).*V(N+1,j);
  x=LBidiSol(l,bj);
  V(2:N,j)=UBidiSol(u,f,x);
end

k=length(E);
U=zeros(k);
for i=1:k
  A=find(Edist > E(i));
  j=A(1);
  U(i)=V(j-1,1);
end

<table>
<thead>
<tr>
<th>Strike Price</th>
<th>Option Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>90,000</td>
<td>7.046</td>
</tr>
<tr>
<td>91,000</td>
<td>7.396</td>
</tr>
<tr>
<td>92,000</td>
<td>7.759</td>
</tr>
<tr>
<td>93,000</td>
<td>8.134</td>
</tr>
<tr>
<td>94,000</td>
<td>8.522</td>
</tr>
<tr>
<td>95,000</td>
<td>8.922</td>
</tr>
<tr>
<td>96,000</td>
<td>9.334</td>
</tr>
<tr>
<td>97,000</td>
<td>9.759</td>
</tr>
<tr>
<td>98,000</td>
<td>10.197</td>
</tr>
<tr>
<td>99,000</td>
<td>10.647</td>
</tr>
</tbody>
</table>
**FINAL REMARK:** We observe that the forward method is more efficient than the backward method because it avoids the loop corresponding to the different options (given by the different strike prices). We have this loop in the BSPDEuro function.

<table>
<thead>
<tr>
<th>Strike Price</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>100.000</td>
<td>11.111</td>
</tr>
<tr>
<td>101.000</td>
<td>11.587</td>
</tr>
<tr>
<td>102.000</td>
<td>12.076</td>
</tr>
<tr>
<td>103.000</td>
<td>12.577</td>
</tr>
<tr>
<td>104.000</td>
<td>13.092</td>
</tr>
<tr>
<td>105.000</td>
<td>13.619</td>
</tr>
<tr>
<td>106.000</td>
<td>14.159</td>
</tr>
<tr>
<td>107.000</td>
<td>14.712</td>
</tr>
<tr>
<td>108.000</td>
<td>15.277</td>
</tr>
<tr>
<td>109.000</td>
<td>15.855</td>
</tr>
<tr>
<td>110.000</td>
<td>16.445</td>
</tr>
</tbody>
</table>