function pvalue = CrankNicolsonB1(TS)
% initialize the parameters for the model from the question

So = 100; % stock price
Smax = 300; % upper limit of PDE mesh
Smin = 0 ; % lower limit of PDE mesh
M_x  = 1; % the fineness of stock grid
N = (Smax - Smin) / M_x ; % this is the horizontal grid number
T = 1; % the time to maturity
% TS = 300; % the time grid size
dt = T/TS; % the fineness of time grid
K  = 90; % strike price
sigma = 0.25; % volatility
r = 0.0525; % interest rate
q = 0.02; % dividend rate

% define our initial condition and stock price matrix
S = zeros(N,1); % the stock price matrix
V = zeros(N,1); % the initial value matrix
p = zeros(N,1); % the boundary condition matrix
% to define our solution matrix
sol = zeros(N,TS + 1);

% initialize the above matrices
for i  = 1:1:N
    S(i) = Smin + (i * M_x); % stock price points
    V(i) = max(K - S(i),0); % the initial condition for a put
end

% defining and building our implicit function matrix
% this matrix has a tridiagonal structure
A = zeros(N,N);
for i = 1:1:N
    term1 = 0.5 * (sigma ^ 2) * (S(i) ^ 2) * (dt/(M_x ^ 2));
    term2 = (r - q) * S(i) * (dt / (2 * M_x));
    % the coefficients of the equation
    alpha = term2 - term1;
    beta  = 1 + (r * dt) + (2 * term1);
    gamma = -term1 - term2;
    % get beta in the first element of the matrix for BC1
    if i == 1
        A(i,i) = beta;
        % get gamma in the second element
        A(i,i + 1) = gamma;
    elseif i == N
        % to fill the last row of the matrix
\begin{verbatim}
A(i,i-1) = alpha;
A(i,i) = beta;
else
    % for rest of the tri-diagonal elements
    A(i, i - 1) = alpha;
    A(i,i) = beta;
    A(i,i + 1) = gamma;
end

% defining and building our explicit function matrix
% this matrix has a tridiagonal structure
B = zeros(N,N);
for i = 1:1:N
    term1 = 0.5 * (sigma ^ 2) * (S(i) ^ 2) * (dt/(M_x ^ 2));
    term2 = (r - q) * S(i) * (dt / (2 * M_x));
    % the coefficients of the equation
    alpha = term1 - term2;
    beta  = 1 - (r * dt) - (2 * term1);
    gamma = term1 + term2;
    % get beta in the first element of the matrix for BC1
    if i == 1
        B(i,i) = beta;
        % get gamma in the second element
        B(i,i + 1) = gamma;
    elseif i == N
        % to fill the last row of the matrix
        B(i,i-1) = alpha;
        B(i,i) = beta;
    else
        % for rest of the tri-diagonal elements
        B(i, i - 1) = alpha;
        B(i,i) = beta;
        B(i,i + 1) = gamma;
    end
end

% include the initial condition in the solution matrix
sol(:,1) = V;
A = A^-1;
% now solving the PDE for each time step
for i = 1:TS
    % solve the next step with the help of the previous one
    tau  = (i - 1) * dt;
    % the boundary conditions
    p1 = 0.5 * dt * (N - 1) * (((sigma ^ 2) * (N - 1)) - (r - q)) * ((K * exp(-r * tau)));
    p2 = 0.5 * dt * (N - 1) * (((sigma ^ 2) * (N - 1)) - (r - q)) * ((K * exp(-r * (tau + \sqrt{dt}))) + p1 + p2);.zeros(N - 1,1))];
    sol(:,i + 1) = A*temp;
end
\end{verbatim}
\[ \text{ans} = \frac{(S_0 - S_{\text{min}})}{M_x}; \]

\[ \text{value} = \text{sol}(\text{ans}, TS + 1); \]

% numerical greeks using finite differences
\[
\text{fdelta} = \frac{-\text{sol}(\text{ans} - 1, TS + 1) + \text{sol}(\text{ans} + 1, TS + 1)}{(2 \times M_x)};
\]

\[
\text{fgamma} = \frac{\text{sol}(\text{ans} + 1, TS + 1) - (2 \times \text{sol}(\text{ans}, TS + 1)) + \text{sol}(\text{ans} - 1, TS + 1)}{(M_x^2)};
\]

% to get the Black Scholes price and greeks
\[
\text{d1} = \frac{(\log(S_0/K) + ((r - q + (0.5 \times \sigma^2)) \times T))/(\sigma \times \sqrt{T})}{(\sigma \times \sqrt{T})};
\]

\[
\text{d2} = \text{d1} - \sigma \times \sqrt{T};
\]

\[
\text{d3} = \text{normcdf}(\text{d1});
\]

\[
\text{d4} = \text{normcdf}(\text{d2});
\]

\[
\text{d5} = \text{normcdf}(\text{-d1});
\]

\[
\text{d6} = \text{normcdf}(\text{-d2});
\]

\[
\text{BSprice} = -\left( S_0 \times \exp(-q \times T) \times \text{d5} \right) + \left( K \times \text{d6} \times \exp(-r \times T) \right);
\]

\[
\text{delta} = \exp(-q \times T) \times (d3 - 1);
\]

\[
\text{gamma} = \frac{(\text{normpdf}(\text{d1}) \times \exp(-q \times T))/(S_0 \times \sigma \times \sqrt{T})}{(S_0 \times \sigma \times \sqrt{T})};
\]

\[
\text{kappa} = \frac{(S_0 \times \exp(-q \times T) \times \text{normpdf}(\text{d1}) \times \sqrt{T})}{(S_0 \times \exp(-q \times T) \times \text{normpdf}(\text{d1}) \times \sqrt{T})};
\]

\[
\text{pvalue} = \text{value}/\text{BSprice};
\]

% the below commented code is to get the Oscillation in time graphs
\[
\text{d1} = \frac{(\log(S_0/K) + ((r - q + (0.5 \times \sigma^2)) \times T))/(\sigma \times \sqrt{T})}{(\sigma \times \sqrt{T})};
\]

\[
\text{d2} = \text{d1} - \sigma \times \sqrt{T};
\]

\[
\text{d3} = \text{normcdf}(\text{d1});
\]

\[
\text{d4} = \text{normcdf}(\text{d2});
\]

\[
\text{d5} = \text{normcdf}(\text{-d1});
\]

\[
\text{d6} = \text{normcdf}(\text{-d2});
\]

\[
\text{for} \ i = 1 : (TS + 1)
\]

\[
\text{tau} = dt \times i;
\]

\[
\text{d}_1 = \frac{(\log(S_0/K) + ((r - q + (0.5 \times \sigma^2)) \times \tau))/(\sigma \times \sqrt{\tau})}{(\sigma \times \sqrt{\tau})};
\]

\[
\text{d}_2 = \text{d}_1 - \sigma \times \sqrt{\tau};
\]

\[
\text{d}_1 = \text{normcdf}(\text{-d}_1);
\]

\[
\text{d}_2 = \text{normcdf}(\text{-d}_2);
\]

\[
\text{BSprice}(i) = -\left( S_0 \times \exp(-q \times \tau) \times \text{d}_1 \right) + \left( K \times \text{d}_2 \times \exp(-r \times \tau) \right);
\]

\[
\text{Error}(i) = \text{sol}(\text{ans},i)/\text{BSprice}(i);
\]

\[
\text{end}
\]

\[
\text{delta} = \exp(-q \times T) \times (d3 - 1);
\]

\[
\text{gamma} = \frac{(\text{normpdf}(\text{d1}) \times \exp(-q \times T))/(S_0 \times \sigma \times \sqrt{T})}{(S_0 \times \sigma \times \sqrt{T})};
\]

\[
\text{kappa} = \frac{(S_0 \times \exp(-q \times T) \times \text{normpdf}(\text{d1}) \times \sqrt{T})}{(S_0 \times \exp(-q \times T) \times \text{normpdf}(\text{d1}) \times \sqrt{T})};
\]

\[
\text{pvalue} = ((\text{value} - \text{BSprice}) / \text{BSprice}) \times 100;
\]