This homework is to be done as a group. Each team will hand in one homework solution, using the blueprint solution provided by our teaching assistant.

The homework has the following parts:
1. Pricing European options using finite differences on a fixed computational domain
2. Pricing American options using finite differences

Pricing European Put Options Using Finite Differences on a Fixed Computational Domain

An underlying asset has lognormal distribution with volatility $\sigma = 0.35$, spot price $S_0 = 41$, and pays dividends continuously at the rate $q = 0.02$. Consider a put option with strike $K = 40$ and maturity $T = 0.75$, i.e., 9 months, The interest rate is assumed to be constant and equal to $r = 0.04$.

We will solve the diffusion equation (the heat equation) on a bounded domain as follows:

$$ u_\tau = u_{xx}, \forall x_{left} < x < x_{right}, \forall 0 < \tau < \tau_{final}, $$

with boundary conditions

$$ u(x,0) = f(x), \forall x_{left} \leq x \leq x_{right}; $$

$$ u(x_{left},\tau) = g_{left}(\tau), \forall 0 \leq \tau \leq \tau_{final}; $$

$$ u(x_{right},\tau) = g_{right}(\tau), \forall 0 \leq \tau \leq \tau_{final}. $$

1. Computational domain:
   We will use a fixed computational domain, where the node

   $$ x_{compute} = \ln \left( \frac{S_0}{K} \right) $$

   will no longer be on the finite difference mesh.

   The upper bound $\tau_{final}$ for $\tau$ is

   $$ \tau_{final} = \frac{T \sigma^2}{2}. $$

   Let the computational domain be

   $$ [x_{left}, x_{right}] = \left[ \ln \left( \frac{S_0}{K} \right) + \left( r - q - \frac{\sigma^2}{2} \right) T - 3\sigma \sqrt{T}, \ln \left( \frac{S_0}{K} \right) + \left( r - q - \frac{\sigma^2}{2} \right) T + 3\sigma \sqrt{T} \right]. $$

   The finite difference discretization is built as follows:
Start with $M$ and $\alpha_{\text{temp}}$ given ($\alpha$ will be slightly smaller than $\alpha_{\text{temp}}$ in the end). Then

$$\delta \tau = \frac{\tau_{\text{final}}}{M}$$

and $\delta x$ will be approximately $\sqrt{\delta \tau / \alpha_{\text{temp}}}$. Let

$$N = \text{floor}\left(\frac{x_{\text{right}} - x_{\text{left}}}{\sqrt{\delta \tau / \alpha_{\text{temp}}}}\right),$$

where floor($y$) is the largest integer smaller than or equal to $y$.

Then,

$$\delta x = \frac{x_{\text{right}} - x_{\text{left}}}{N}$$

and $\alpha < \alpha_{\text{temp}}$ is defined as

$$\alpha = \frac{\delta \tau}{(\delta x)^2}.$$  

2. Boundary conditions:

The following change of variables transforms $x$ and $\tau$ into $S$ and $t$, respectively, and maps $V(S, t)$, the value of the call option, into $u(x, \tau)$, a solution to the heat equation:

$$V(S, t) = \exp(-ax - b\tau)u(x, \tau),$$

where

$$x = \log \frac{S}{K}; \quad \tau = \frac{(T - t)\sigma^2}{2},$$

and the constants $a$ and $b$ are given by

$$a = \frac{r - q}{\sigma^2} - \frac{1}{2},$$

$$b = \left(\frac{r - q}{\sigma^2} + \frac{1}{2}\right)^2 + \frac{2q}{\sigma^2}.$$

For a put option, the corresponding boundary conditions for $u(x, \tau)$ are:

$$f(x) = K \exp(ax) \max(1 - \exp(x), 0), \quad x_{\text{left}} < x < x_{\text{right}};$$

$$g_{\text{left}}(\tau) = K \exp(ax_{\text{left}} + b\tau) \left(\exp\left(\frac{2\tau \sigma^2}{2}\right) - \exp\left(x_{\text{left}} - \frac{2\tau \sigma^2}{2}\right)\right), \quad 0 < \tau < \tau_{\text{final}};$$

$$g_{\text{right}}(\tau) = 0, \quad \forall \ 0 < \tau < \tau_{\text{final}}.$$

3. Finite difference schemes:

Use Forward Euler with $\alpha = 0.45$, and Backward Euler and Crank-Nicolson with $\alpha \in (0.45, 5)$, to solve the diffusion equation for $u(x, \tau)$. For the implicit methods, use both Cholesky and SOR with relaxation parameter $\omega = 1.2$. The stopping criterion for SOR is that the norm of the difference between two consecutive approximations is less than $\text{tol} = 10^{-6}$.

Run each finite difference method for the initial value $M = 4$, and then quadruple the number of points on the $\tau$-axis, i.e., choose $M \in \{4M, 16M, 64M\}$.

To understand the numbers you provide, please include the following: for Forward Euler with $\alpha = 0.45$, for Backward Euler with $\alpha = 0.45$, and for Crank-Nicolson with $\alpha = 0.45$, let $M = 4$. Run your codes and record the values of the finite difference approximations at each node, including at the boundary nodes. For $M = 4$ and $\alpha = 0.45$ the corresponding value of $N$ is $N = 12$. Thus, for each of the three methods above, you will have to fill out a table with five rows (corresponding to time steps from 0 - boundary conditions, to 4) and 13 columns (including the boundary conditions at $x_{\text{left}}$ and $x_{\text{right}}$).

4. Pointwise Convergence:
Identify the interval containing \( x_{\text{compute}} = \log(S_0/K) \), i.e., find \( i \) such that
\[
x_i \leq x_{\text{compute}} < x_{i+1}.
\]

Let
\[
(2) \quad S_i = Ke^{x_i}, \\
(3) \quad S_{i+1} = Ke^{x_{i+1}}
\]
and let
\[
V_i = \exp(-ax_i - 2b\tau_{\text{final}})u(x_i, \tau_{\text{final}}) \\
V_{i+1} = \exp(-ax_{i+1} - 2b\tau_{\text{final}})u(x_{i+1}, \tau_{\text{final}})
\]
be the approximate values of the option corresponding to \( S_i \) and \( S_{i+1} \), respectively.

The approximate value of the option, \( V_{\text{approx}}(S_0, 0) \) is now computed by linear interpolation, i.e.,
\[
V_{\text{approx}}(S_0, 0) = \frac{(S_{i+1} - S_0)V_i + (S_0 - S_i)V_{i+1}}{S_{i+1} - S_i}.
\]

Let \( V_{\text{exact}}(S_0, 0) \) be the value computed from the Black–Scholes Formula. The pointwise relative error is
\[
\text{error}_{\text{pointwise}} = |V_{\text{approx}}(S_0, 0) - V_{\text{exact}}(S_0, 0)|.
\]

Another way of computing an approximate value for the option would be to use linear interpolation to find the value of \( u(x_{\text{compute}}, \tau_{\text{final}}) \), and then use the change of variables to obtain \( V_{\text{approx}, 2}(S_0, 0) \), i.e.,
\[
u(x_{\text{compute}}, \tau_{\text{final}}) = \frac{(x_{i+1} - x_{\text{compute}})u(x_i, \tau_{\text{final}}) + (x_{\text{compute}} - x_i)u(x_{i+1}, \tau_{\text{final}})}{x_{i+1} - x_i},
\]
\[
V_{\text{approx}, 2}(S_0, 0) = \exp(-ax_{\text{compute}} - 2b\tau_{\text{final}})u(x_{\text{compute}}, \tau_{\text{final}}).
\]

Let
\[
\text{error}_{\text{pointwise}, 2} = |V_{\text{approx}}(S_0, 0) - V_{\text{exact}}(S_0, 0)|
\]
be the pointwise relative error corresponding to this method.

For each finite difference method, compute and record \( \text{error}_{\text{pointwise}} \) and \( \text{error}_{\text{pointwise}, 2} \), as well as the ratio of the approximation errors from one discretization level to the next.

5. **Root-Mean-Squared (RMS) Error:**

Let \( x_k = x_{e, f} + kd, k = 0 : N \), be a nodal point on the \( x \)-axis. Let \( S_k = Ke^x \) and let \( V_{\text{approx}}(S_k, 0) \) be the approximate value of a call option with spot price \( S_k \) obtained from the finite difference scheme, i.e.,
\[
V_{\text{approx}}(S_k, 0) = \exp(-ax_k - b\tau_{\text{final}})u(x_k, \tau_{\text{final}}).
\]

Let \( V_{\text{exact}}(S_k, 0) \) be the Black–Scholes value of a call option with spot price \( S_k \). The RMS error \( \text{error}_{\text{RMS}} \) is defined as
\[
\text{error}_{\text{RMS}} = \sqrt{\frac{1}{N+1} \sum_{k=0}^{N} \frac{|V_{\text{approx}}(S_k, 0) - V_{\text{exact}}(S_k, 0)|^2}{|V_{\text{exact}}(S_k, 0)|^2}}
\]

For each finite difference method, compute and record \( \text{error}_{\text{RMS}} \), as well as the ratio of the approximation errors from one discretization level to the next.

6. **Finite Difference Approximation of \( \Delta, \Gamma, \) and \( \Theta \):**

Recall that \( x_i \) and \( x_{i+1} \) are consecutive nodes such that \( x_i \leq x_{\text{compute}} < x_{i+1} \).
Recall the notations from (5–7) and let
\[ S_{i-1} = Ke^{x_{i-1}}; \]
\[ S_{i+2} = Ke^{x_{i+2}}; \]
\[ V_{i-1} = \exp(-ax_{i-1} - 2b\tau_{\text{final}})u(x_{i-1}, \tau_{\text{final}}); \]
\[ V_{i+2} = \exp(-ax_{i+2} - 2b\tau_{\text{final}})u(x_{i+2}, \tau_{\text{final}}). \]

The finite different approximations for the \( \Delta \) and \( \Gamma \) of the option are as follows:
\[
\Delta \approx \frac{V_{i+1} - V_i}{S_{i+1} - S_i}; \\
\Gamma = \frac{V_{i+2} - V_{i+1}}{S_{i+2} - S_{i+1}} - \frac{V_{i+1} - V_i}{S_i - S_{i-1}}.
\]

To compute an approximation for \( \Theta \), note that the next to last time step on the \( \tau \)-axis, \( \tau_{\text{final}} - \delta \tau \) corresponds to time \( \delta t = \frac{2\delta \tau}{\sigma^2} \).

Let
\[
V_{i,\delta t} = \exp(-ax_i - 2b(\tau_{\text{final}} - \delta \tau))u(x_i, \tau_{\text{final}} - \delta \tau); \\
V_{i+1,\delta t} = \exp(-ax_{i+1} - 2b(\tau_{\text{final}} - \delta \tau))u(x_{i+1}, \tau_{\text{final}} - \delta \tau),
\]
and let
\[
V_{\text{approx}}(S_0, \delta t) = \frac{(S_{i+1} - S_0)V_{i,\delta t} + (S_0 - S_i)V_{i+1,\delta t}}{S_{i+1} - S_i}.
\]

Then,
\[
\Theta \approx \frac{V_{\text{approx}}(S_0, 0) - V_{\text{approx}}(S_0, \delta t)}{\delta t}.
\]

Compute the finite difference approximations (4), (4), and (4) for the Greeks for every finite difference scheme.
Pricing American Put Options Using Finite Differences on a Fixed Computational Domain

We now want to price an American Put option with the same parameters, i.e., $S_0 = 41$, $K = 40$, $T = 0.75$, $\sigma = 0.35$, $q = 0.02$, and $r = 0.04$.

There is no closed formula for pricing American put options. To test convergence, use the following value, obtained from an average binomial tree method with 10,000 time steps as the exact value of the American put:

$$P_{amer,bin} = 4.083817051176386.$$

To price the American Put, we solve the same diffusion equation as for the European Put, taking into account the fact that the value of the put option must be greater than the early exercise value $\max(K - S(t), 0)$.

The computational domain and the boundary conditions are the same as those previously considered for the European Put option. Note that the domain is therefore a fixed computational domain and interpolation will be required to compute the finite difference approximation of the value of the American option.

The Forward Euler scheme has to be modified as follows: for European options at each time step $m$ corresponding to $\tau = m\delta\tau$, we computed $U^m$ directly from $U^{m-1}$, without having to solve a linear system. For American Options, the value of the put must be greater than $\max(K - S, 0)$. Using the change of variables, this corresponds to the following condition for $u(x, \tau)$:

$$u(x, \tau) \geq \exp(ax + 2b\tau) \max(1 - \exp(x), 0).$$

The discretized version of this is

$$U^m_{amer}(i) \geq \text{early.ex.premium}(i), \quad \forall \ i = 2 : N,$$

where early.ex.premium is computed as

for $i=1:(N-1)$

$$x = x_{left} + i\delta x; \tau = m\delta\tau;$$

$$\text{early.ex.premium}(i) = \exp(ax + 2b\tau) \max(1 - \exp(x), 0);$$

end

The Forward Euler code for pricing European options needs to be modified in only one place, as follows:

for $i=1:(N-1)$

$$U^m_{amer}(i) = \max(U^m_{euro}(i), \text{early.ex.premium}(i));$$

end

We modify the implicit Crank-Nicolson scheme using an iterative SOR solver to price American put options. For European options, a linear system of the form

$$AU^m_{euro} = b^m,$$

where $b^m$ is computed using $U^{m-1}$, is solved at each time step using the SOR iteration as follows:

for $j = 1:N-1$

$$x_{n+1}(j) = (1 - \omega) x_n(j) + \frac{\omega}{1 + \alpha} \left( \frac{\alpha}{2} x_{n+1}(j - 1) + \frac{\alpha}{2} x_n(j + 1) + b(j) \right);$$

end

until convergence is achieved, and then we set $U^m_{euro}$ to be $x_n$.

For American options, we use projected SOR, i.e., we modify the iteration above as follows:

for $j = 1:N-1$
\[ x_{n+1}(j) = \max(\text{early}_{ex}.\text{premium}(j), (1 - \omega) x_n(j) + \frac{\omega}{1 + \alpha} \left( \frac{\alpha}{2} x_{n+1}(j-1) + \frac{\alpha}{2} x_n(j+1) + b(j) \right) ); \]

Here, early_{ex}.premium is the early exercise vector corresponding to time \( m\delta \tau \) computed before.

**Finite difference schemes**

Use Forward Euler with \( \alpha = 0.45 \), and Backward Euler and Crank-Nicolson with \( \alpha \in \{0.45, 0.5\} \), to solve the diffusion equation for \( u(x, \tau) \). For the implicit methods, use SOR with relaxation parameter \( \omega = 1.2 \). The stopping criterion for SOR is that the norm of the difference between two consecutive approximations is less than \( tol = 10^{-6} \).

Run each finite difference method for the initial value \( M = 4 \), and then quadruple the number of points on the \( \tau \)-axis, i.e., choose \( M \in \{4M, 16M, 64M\} \).

To understand the numbers you provide, please include the following: for Forward Euler with \( \alpha = 0.45 \) and for Crank-Nicolson with \( \alpha = 0.45 \), let \( M = 4 \). Run your codes and record the values of the finite difference approximations at each nodes, including at the boundary nodes. For \( M = 4 \) and \( \alpha = 0.45 \) the corresponding value of \( N \) is \( N = 12 \). Thus, for each of the three methods above, you will have to fill out a table with five rows (corresponding to time steps from 0 - boundary conditions, to 4) and 13 columns (including the boundary conditions at \( x_{left} \) and \( x_{right} \)).

### 1. Pointwise Convergence:

Identify the interval containing \( x_{compute} = \log(S_0/K) \), i.e., find \( i \) such that
\[ x_i \leq x_{compute} < x_{i+1}. \]

Let
\[
\begin{align*}
S_i &= Ke^{x_i} \\
S_{i+1} &= Ke^{x_{i+1}}
\end{align*}
\]

and let
\[
\begin{align*}
V_i &= \exp(-ax_i - 2b\tau_{final})u(x_i, \tau_{final}) \\
V_{i+1} &= \exp(-ax_{i+1} - 2b\tau_{final})u(x_{i+1}, \tau_{final})
\end{align*}
\]

be the approximate values of the option corresponding to \( S_i \) and \( S_{i+1} \), respectively.

The approximate value of the option, \( V_{approx}(S_0, 0) \) is now computed by linear interpolation, i.e.,
\[
V_{approx}(S_0, 0) = \frac{(S_{i+1} - S_0)V_i + (S_0 - S_i)V_{i+1}}{S_{i+1} - S_i}.
\]

Let \( V_{exact}(S_0, 0) = P_{amer.bin} = 0.038317051176386 \) be the value computed using the average binomial tree method. The pointwise relative error is
\[
\text{error}_{pointwise} = \frac{|V_{approx}(S_0, 0) - V_{exact}(S_0, 0)|}{|V_{exact}(S_0, 0)|}.
\]

For each finite difference method, compute and record \( \text{error}_{pointwise} \) and \( \text{error}_{pointwise, 2} \), as well as the ratio of the approximation errors from one discretization level to the next.

### 2. Finite Difference Approximation of the Greeks

For each finite difference method, compute and record approximate values of \( \Delta, \Gamma, \) and \( \Theta \) as above for the American Put option.
3. Variance Reduction for American Option pricing

For any finite difference scheme used to price an American option, you can use the same scheme to price the European version of the same option. If you also know the exact value of the European option, as is the case for a plain vanilla put, where the Black–Scholes value can be computed, then you have the following three pieces of information:

- $P_{\text{Amer}}^{\text{approx}}(S_0, 0)$, the finite difference approximation of the price of the American put option;
- $P_{\text{Eur}}^{\text{approx}}(S_0, 0)$, the finite difference approximation of the price of the European put option;
- $P_{\text{BS}}(S_0, 0)$, the Black–Scholes price of the European put option.

The variance reduction technique generates a new approximation for the price of the American put, by adding the finite difference error corresponding to the European put to the finite difference approximation of the American put, i.e.,

$$P_{\text{Amer}}^{\text{Var Red}}(S_0, 0) = P_{\text{Amer}}^{\text{approx}}(S_0, 0) + (P_{\text{BS}}(S_0, 0) - P_{\text{Eur}}^{\text{approx}}(S_0, 0)).$$

The corresponding approximation error is

$$\text{error pointwise}_{\text{Var Red}} = |P_{\text{Amer}}^{\text{Var Red}}(S_0, 0) - P_{\text{Amer bin}}|.$$

For each finite difference method, compute and record:

1. $P_{\text{Amer}}^{\text{Var Red}}(S_0, 0)$ as “Var Red”
2. $\text{error pointwise}_{\text{Var Red}}$ as “Var Red Pointwise Error”

4. Early exercise domain

Look only at Crank-Nicolson SOR with $M = 16$ and $\alpha = 0.45$. For each time step $m$, identify the interval where early exercise becomes optimal, i.e., find $N_{\text{opt}}$ such that

$$U^m(N_{\text{opt}}) = \text{early ex premium}(N_{\text{opt}})$$
and

$$U^m(N_{\text{opt}} + 1) > \text{early ex premium}(N_{\text{opt}} + 1).$$

Note that this corresponds to

$$\tau = m\delta \tau,$$

which, in $t$-space, is

$$t = T - \frac{2\tau}{\sigma^2} = T - \frac{2m\delta \tau}{\sigma^2}.$$

Compute the corresponding values for the spot price, i.e.,

$$S_{N_{\text{opt}}} = K \exp(x_{N_{\text{opt}}})$$
$$S_{N_{\text{opt}} + 1} = K \exp(x_{N_{\text{opt}} + 1})$$

and average them:

$$S_{\text{opt}}(t) = \frac{S_{N_{\text{opt}}} + S_{N_{\text{opt}} + 1}}{2},$$

where $t$ is given by

$$t = T - \frac{2m\delta \tau}{\sigma^2}.$$

Record $(t, S_{\text{opt}}(t))$ for all $m = 0 : 16$. 
Pricing Bermudan Options using Finite Differences

We now want to price a Bermudan Put option with the same parameters, i.e., $S_0 = 41$, $K = 40$, $T = 0.75$, $\sigma = 0.35$, $q = 0.02$, and $r = 0.04$. This option can also be exercised at time $t_{ex,1} = 0.4$, i.e., in four months.

Let

$$\tau_{ex,1} = \frac{(T - t_{ex,1})\sigma^2}{2}$$

$$\tau_{final} = \frac{T\sigma^2}{2}$$

To price the Bermudan Put, we solve the same diffusion equation as for the European Put, between time $\tau = 0$ and $\tau_{ex,1}$. Then, the values obtained corresponding to $\tau_{ex,1}$ are checked for early exercise optimality and modified accordingly. These values become boundary conditions for a diffusion equation between time $\tau_{ex,1}$ and $\tau_{final}$.

We only use Forward Euler with $\alpha = 0.45$.

We will also use a computational domain where node $x_{compute} = \ln \left( \frac{S_0}{K} \right)$ is a discretization node.

**Computational domain on the interval $[0, \tau_{ex,1}]$**

In the $(x, \tau)$ space one of the nodal values will be

$$x_{compute} = \log \left( \frac{S_0}{K} \right).$$

To choose the domain on the $x$-axis, fix the Courant constant $\alpha$. If $M_1$ is the number of time steps on the $\tau$-axis, then $\delta \tau_1 = \frac{\tau_{ex,1}}{M_1}$. Therefore

$$\delta x = \sqrt{ \frac{\delta \tau_1}{\alpha} };$$

Here,

$$\alpha = \frac{\delta \tau_1}{(\delta x)^2}.$$  

Choose temporary left and right end points as follows:

$$\tilde{x}_{left} = \ln \left( \frac{S_0}{K} \right) + \left( r - q - \frac{\sigma^2}{2} \right) T - 3\sigma \sqrt{T};$$

$$\tilde{x}_{right} = \ln \left( \frac{S_0}{K} \right) + \left( r - q - \frac{\sigma^2}{2} \right) T + 3\sigma \sqrt{T}.$$  

Let

$$N_{left} = \text{ceil} \left( \frac{x_{compute} - \tilde{x}_{left}}{\delta x} \right) \quad \text{and} \quad N_{right} = \text{ceil} \left( \frac{\tilde{x}_{right} - x_{compute}}{\delta x} \right),$$

where $\text{ceil}(y)$ is the smallest integer larger than or equal to $y$. Then

$$N = N_{left} + N_{right}$$

and

$$x_{left} = x_{compute} - N_{left}\delta x \quad \text{and} \quad x_{right} = x_{compute} + N_{right}\delta x.$$  

The nodal points are

$$x_k = x_{left} + k\delta x, \ \forall k = 0 : N.$$  

Note that $N$, the number of intervals in the $x$-axis discretization will not change when we discretize the interval $[\tau_{ex,1}, \tau_{final}]$. 
Boundary Conditions on the interval $[0, \tau_{ex, 1}]$

The boundary conditions for a put options are:

\begin{align*}
    f_1(x) &= \exp(ax) \max(1 - \exp(x), 0), \quad \forall \ x_{left} < x < x_{right}; \\
    g_{1, left}(\tau) &= \exp(ax_{left} + 2b\tau) \left(\exp(-2r\tau/\sigma^2) - \exp(x_{left})\right), \quad \forall \ 0 < \tau < \tau_{ex, 1}; \\
    g_{1, right}(\tau) &= 0, \quad \forall \ 0 < \tau < \tau_{ex, 1}.
\end{align*}

Choose $M_1 = 4$ to begin with. Repeat the algorithm for $M_1 \in \{4M_1, 16M_1, 64M_1\}$.

Use Forward Euler with $\alpha = 0.45$ to solve the diffusion equation for $u(x, \tau_{ex, 1})$. Let $U_1^{M_1}$ be the corresponding nodal vector values. Change the approximation values by requiring they are larger than the early exercise values, i.e.,

\[ U_1^{M_1}(i) = \max(U_1^{M_1}(i), \text{early}_{\text{ex, premium}}(i)), \quad \forall \ i = 1 : N - 1, \]

where $\text{early}_{\text{ex, premium}}$ is computed as before.

Computational domain on the interval $[\tau_{ex, 1}, \tau_{final}]$

The nodal points on the $x$-axis stay the same, i.e., $x_{left}, x_{right}, \delta x$ and $N$ will not change. What will change is $\alpha$, which will become slightly lower, and we will choose $M_2$ accordingly.

Let

\[ \delta \tau_2 = \frac{\tau_{final} - \tau_{ex, 1}}{M_2}. \]

If we assume $\alpha = \delta \tau_2/(\delta x)^2$, then

\[ M_2 = \frac{\tau_{final} - \tau_{ex, 1}}{\alpha(\delta x)^2}. \]

This, however, may not be a positive integer, so we choose

\[ M_2 = \text{ceil} \left( \frac{\tau_{final} - \tau_{ex, 1}}{\alpha(\delta x)^2} \right), \]

and therefore

\[ \alpha_2 = \frac{\delta \tau_2}{(\delta x)^2} < \alpha. \]

Boundary Conditions on the interval $[\tau_{ex, 1}, \tau_{final}]$

The boundary conditions are the same on the left and right side of the interval, i.e.,

\begin{align*}
    g_{2, left}(\tau) &= \exp(ax_{left} + 2b\tau) \left(\exp(-2r\tau/\sigma^2) - \exp(x_{left})\right), \quad \forall \ 0 < \tau < \tau_{ex, 1}; \\
    g_{2, right}(\tau) &= 0, \quad \forall \ 0 < \tau < \tau_{ex, 1}.
\end{align*}

The nodal values corresponding to time $\tau_{ex, 1}$ are set equal to $U_1^{M_1}$, i.e.,

\[ U_2^0(i) = U_1^{M_1}(i), \quad \forall \ i = 1 : N - 1. \]

Use Forward Euler with $\alpha_2$ to solve the diffusion equation for $u(x, \tau_{final})$.

Finite Difference Solution:

Let $U_2^{M_2}$ be the vector of length $N - 1$ which gives the finite difference solution after $M_2$ time steps on the interval $[\tau_{ex, 1}, \tau_{final}]$. Recall that $x_{\text{compute}} = x_{N_{left}}$. Then, $U_2^{M_2}(N_{left})$ is the finite difference approximation to $u(x_{\text{compute}}, \tau_{final})$. The following change of variables computes the finite difference approximate value of the option, $P_{\text{Bern approx}}(S_0, 0)$:

\[ P_{\text{Bern approx}}(S_0, 0) = K \exp(-ax_{\text{compute}} - 2b\tau_{final}) U_2^{M_2}(N_{left}). \]

Let $P_{\text{exact}}(S_0, 0)$ be the value computed from the Black–Scholes formula.

Computational Domain and Finite Difference Solution
For $M_1 \in \{4, 16, 64, 256\}$, and $\alpha = 0.4$, record $M_2$, $\alpha_2$, $N_{left}$, $N_{right}$, $N$, $x_{left}$, $x_{right}$, $x_{compute}$, $\delta x$, $\tau_{ex,1}$, $\tau_{final}$, $\delta \tau_1$ and $\delta \tau_2$.

For each finite difference method, compute and record:

1. $U_2^{M_2}(N_{left})$ as “u value”
2. $P_{approx}(S_0, 0)$ as “Bermudan Put Value”