Pricing of American Style Options

Ishaq Hamza, Rohit Jorige, Balaji R

1 Inroduction

A Bermudan option is characterized by strike K, time to maturity T, and a set of exercise dates $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$, where $t_i < t_{i+1}$. The option can be exercised at any of the exercise dates t_i. The payoff at exercise date t_i is given by:

$$V(t_i) = \max(S(t_i) - K, 0) \tag{1}$$

where $S(t_i)$ is the stock price at time t_i. The continuation value is defined as the expected value of holding the option until the next exercise date:

$$C(t_i) = e^{-r(T-t_i)}\mathbb{E}[V(t_{i+1})|S(t_i)] \tag{2}$$

where r is the risk-free interest rate. The option is exercised at time t_i if the payoff at that time is greater than the continuation value, i.e., $V(t_i) > C(t_i)$. Otherwise, the option is held until the next exercise date.

An American option is a special case of the Bermudan option where $\mathcal{T} = [0, T]$. The option can be exercised at any time before or at maturity. The payoff at exercise date t_i is given by:

$$V(t_i) = \max(S(t_i) - K, 0) \tag{3}$$

We will discuss pricing of the Bermudan options using longstaff and Schwartz model (simulation based approach) and binomial tree model. We extend the binomial tree model to price American options in two ways: (1) using $\mathcal{T} = \{0, T/N, \ldots, (N-1)T/N\}$, where we take N to be a large value, and (2) using Richardson extrapolation. We also explore a finite difference scheme assuming Black-Scholes model but with *open boundary conditions*.

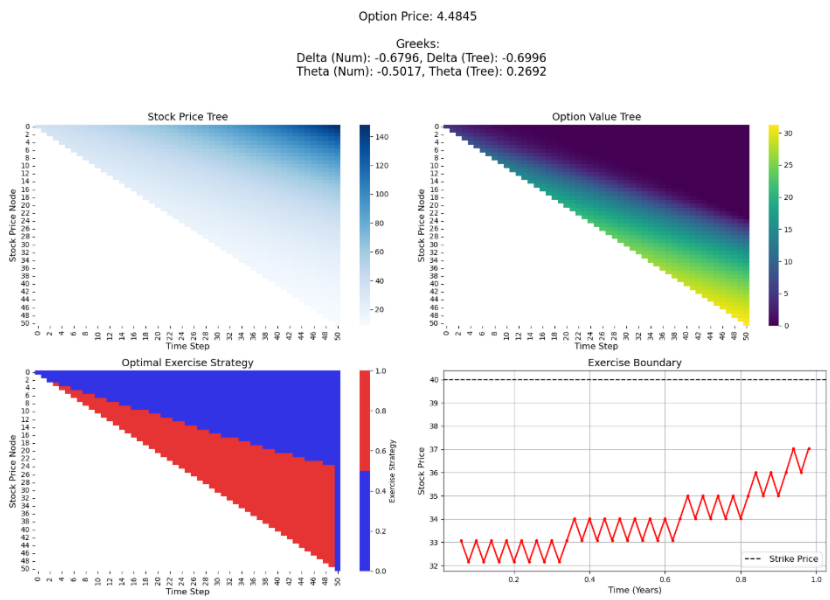


Figure 1: Bermudean pricing using Binomial tree

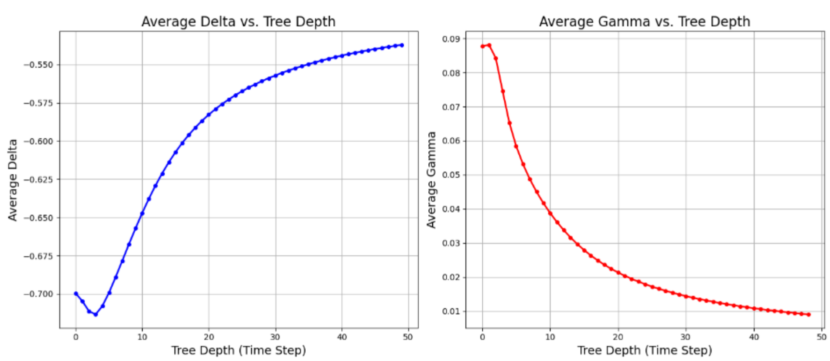


Figure 2: Variation of greeks with tree depth

2 Longstaff-Schwartz Method For Bermudan Option Pricing

2.1 Methodology For Calculating Option Price

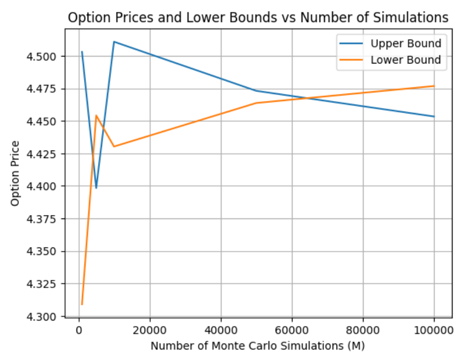
Consider a Bermudan Put option with the following parameters:

- **Initial stock price**: $S_0 = 36$
- **Strike price**: $K = 40$
- **Risk-free rate**: $r = 0.06$
- **Volatility**: 20%
- **Maturity**: $T = 1$ year
- **Number of exercise dates**: $N = 50$
- **Number of Monte Carlo simulations**: $M = 10{,}000$

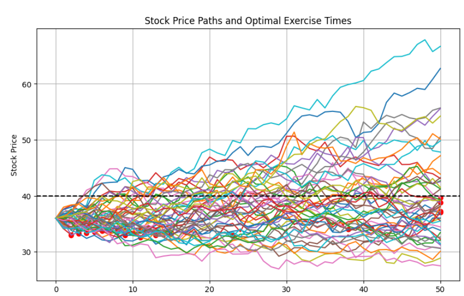
- Sample M monte carlo paths of geometric brownian motion of the stock under the risk neutral distribution. Assume no dividends.
- For each path, initialize cashflow as the payoff at maturity. Virtually, this is the same as a European option.
- Iterate over the exercise times backward, from the $(N-1)^t h$ to the first timestep.
- At each timestep, the question here is whether to exercise or not. One exercises when the immediate payoff is grater than the continuation value. The goal is to estimate the continuation value as a function of stock price. From Bellman's optimality principle of dynamic programming, one can estimate the continuation value as the mean of the future cashflows.
- Do a regression(least squares, fitted with a quadratic polynomial) of the mean above, with the stock price. This will give the continuation value. Store the regression coefficients.
- Finally update the in the money cashflows with the immediate payoff value if it is greater than the continuation value, else discount the existing cashflow for the path. Discount all the out of money paths' existing cashflows.
- Convergence plot is drawn for different number of paths.

2.2 Methodology For Calculating Lower Bound

- Use the regression coefficients to get the continuation value function at each timestep.
- Sample new paths, and for each path determine the optimal policy just from the continuation value corresponding to the stock price calculated from the regression coefficients.
- Finally from the optimal policy, discount the cashflows for each path. Take the mean of all cashflows to get the lower bound.



Finally, using the regression coefficients, one can plot the optimal exercise policy for a few(50) sample paths:



3 Binomial Tree Model for Bermudan Call Options

3.1 Backward Induction Algorithm

Let $\mathcal{T} = \{t_1, ..., t_n\}$ be exercise dates. For each node (i, j) in the binomial tree:

$$
\begin{aligned}
\text{Stock Price:} \quad & S_{i,j} = S_0 u^j d^{i-j} \\
\text{Payoff:} \quad & \Phi(S_{i,j}) = \max(S_{i,j} - K, 0) \quad \text{(Call)} \\
\text{Continuation Value:} \quad & V_{i,j}^{\text{cont}} = e^{-r\Delta t}[pV_{i+1,j} + (1-p)V_{i+1,j+1}] \\
\text{Option Value:} \quad & V_{i,j} = \begin{cases} \max(\Phi(S_{i,j}), V_{i,j}^{\text{cont}}) & \text{if } t_i \in \mathcal{T} \\ V_{i,j}^{\text{cont}} & \text{otherwise} \end{cases}
\end{aligned}
$$

3.2 Exercise Boundary

For each exercisable time $t_i \in \mathcal{T}$, the critical stock price $S_{t_i}^*$ is:

$$S_{t_i}^* = \min\{S_{i,j} \mid \Phi(S_{i,j}) \geq V_{i,j}^{\text{cont}}\}$$

This forms the early exercise boundary where immediate exercise becomes profitable in expectation.

3.3 Greeks Computation

We compute greeks at each node (i, j) using finite difference approximations, with step sizes pertaining to that of the tree itself, making it computationally very efficient. For true values, we should run a numerical differentiation limiting scheme which computes the node's paramters with slightly shifted values and uses a central difference formula. But as discussed in the first assignment, as long as the step sizes of the tree are small enough for option value to converge, the finite-difference approximation of the greeks with the tree's step size is as almost the same as that with the numerical differentiation limiting scheme which requires the running of the whole algorithm at each node with perturbed values (computationally infavourable). Following are the greeks:

Delta Measures sensitivity to underlying price (positive for calls):

$$\Delta_{i,j} = \frac{\partial V}{\partial S} \approx \frac{V_{i+1,j} - V_{i+1,j+1}}{S_{i,j}(u-d)}$$

Gamma Measures convexity of option value with respect to underlying price:

$$\Gamma_{i,j} = \frac{\partial^2 V}{\partial S^2} \approx \frac{\Delta_{i,j} - \Delta_{i,j+1}}{S_{i,j}(u-d)}$$

Theta Measures time decay (typically negative):

$$\Theta_{i,j} = \frac{\partial V}{\partial t} \approx \frac{V_{i+1,j}^{\text{cont}} - V_{i,j}}{\Delta t}$$

We also have Vega = $\frac{\partial V}{\partial \sigma}$ and Rho = $\frac{\partial V}{\partial r}$, which measure sensitivity to volatility and interest rate respectively, but since these are taken to be constants in our modeling, computing them cannot be simplified through the tree as we have done for the other greeks. We can compute them using numerical schemes but they won't be used in hedging as they are not the varying parameters in the model.

3.4 Implementation and Metrics of Interest

Essentially, we have a normal binomial tree which we used for European options in the first assignment, but now we have boundary condition at all depths $\in \mathcal{T}$ which are computed by comparing the payoff at that stage with the continuation value (which is computed from the future layer). Hence we compute $V_{i,j}$ as defined in the backward induction algorithm.

1. Construct binomial tree for underlying asset
2. Initialize terminal payoffs at maturity: $V_{N,j} = \max(S_{N,j} - K, 0)$
3. Backward induction with early exercise check at $\mathcal{T}$
4. Compute Greeks at each node using finite differences
5. Identify exercise boundaries through critical price comparison

It is natural that a Bermudan option is worth more than a European option and less than an American option with the same strike and maturity, as the holder has the flexibility to exercise at multiple times but not at any time. By increasing the number of exercise dates, we can test this convergence (and we did, in the implementation). This is also a naive way to price an American option.

One also should consider how many steps in the binomial tree n are needed to obtain reasonable convergence of the option price as a function of the number of exercise dates $|\mathcal{T}|$. We would expect this to grow, as more boundary conditions are added. This too, is explored in the implementation.

Besides these, the usual metrics of the convergence of the option price with number of steps and the early exercise boundary are included.

3.5 Results

We priced a put option with the given parameters assuming equally spaced exercise dates. The results are as follows

The other plots are for the extra metrics which we discussed.

Figure 3: convergence plot

4 Estimating American Option price using numerical methods

Pricing American options, which permit exercise at any time up to maturity T, is challenging due to the optimal early exercise decision. A common approach involves approximating the American price using Bermudan options, which restrict exercise to discrete dates $\mathcal{T} = \{t_1, ..., t_{N_e}\}$. As the number of exercise dates N_e increases, the Bermudan price $V(N_e)$ converges to the American price V_{American}. This convergence allows us to use extrapolation techniques based on Bermudan prices computed via methods like the Binomial Tree model.

4.1 Binomial Tree for Bermudan Prices

The Binomial Tree model discretizes time into N steps. At each step i, the stock price $S_{j,i}$ can move up to $S_{j,i}u$ or down to $S_{j,i}d$ with risk-neutral probability p, where $u = e^{\sigma\sqrt{\Delta t}}$, $d = 1/u$, and $p = (e^{r\Delta t} - d)/(u - d)$. The option value $V_{j,i}$ is found via backward induction. At allowed exercise steps i, the value is $V_{j,i} = \max(\text{Intrinsic Value}, \text{Continuation Value})$, where the continuation value is $C_{j,i} = e^{-r\Delta t}[pV_{j,i+1} + (1-p)V_{j+1,i+1}]$. Otherwise, $V_{j,i} = C_{j,i}$. We compute $V(N)$ assuming exercise is possible at all steps $i = 1, ..., N$.

4.2 Richardson Extrapolation

This method leverages the error expansion of the Binomial Tree price:

$$V(N) = V_{\text{American}} + \frac{c_1}{N} + \frac{c_2}{N^2} + \mathcal{O}\left(\frac{1}{N^3}\right)$$

By computing $V(N)$ and $V(2N)$, the combination $2V(2N) - V(N)$ eliminates the leading $O(1/N)$ error term:

$$V_{\text{Richardson}} = 2V(2N) - V(N) = V_{\text{American}} - \frac{c_2}{2N^2} + \mathcal{O}\left(\frac{1}{N^3}\right)$$

This provides a more accurate estimate $V_{\text{Richardson}}$ converging at a rate of $O(1/N^2)$.

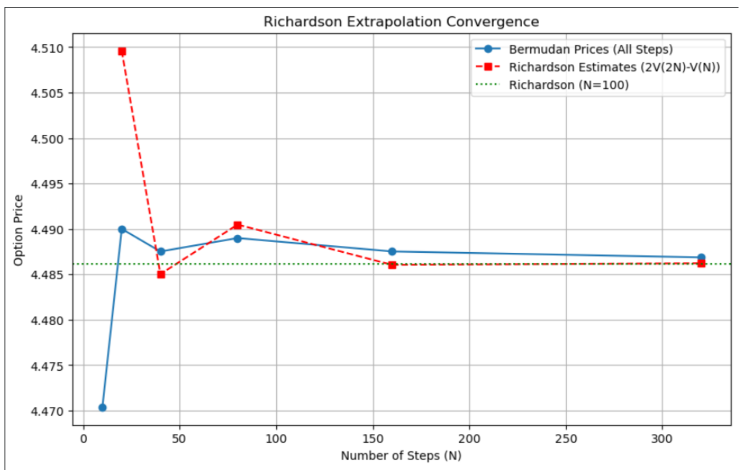


Figure 4: Convergence of Binomial Tree prices and Richardson extrapolation estimates for the American put option.

4.3 Cubic Polynomial Extrapolation

This technique fits a cubic polynomial to $V(N)$ viewed as a function of $h = 1/N$. Using the error expansion:

$$V(N) = V(h) = V_{\text{American}} + a_1 h + a_2 h^2 + a_3 h^3 + \mathcal{O}(h^4)$$

We compute $V(N_i)$ for several N_i values, let $h_i = 1/N_i$, and fit the model:

$$\hat{V}(h) = \beta_3 h^3 + \beta_2 h^2 + \beta_1 h + \beta_0$$

The extrapolated American price estimate is the constant term β_0, corresponding to $h = 0$:

$$V_{\text{Cubic}} = \hat{V}(0) = \beta_0$$

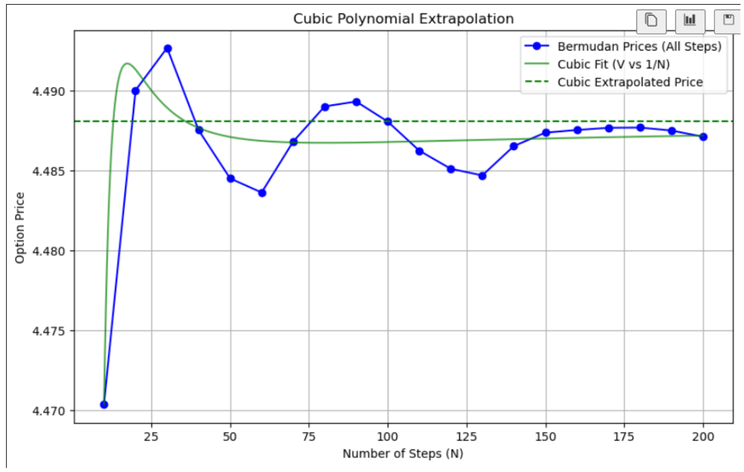


Figure 5: Cubic polynomial fit to Binomial Tree prices for American put option extrapolation.

4.4 Romberg Extrapolation

Romberg extrapolation systematically applies Richardson-like steps to eliminate higher-order error terms. We compute $V(N_k)$ for a sequence $N_k = N_0 \cdot 2^k$. Let $R_{k,0} = V(N_k)$. The Romberg table entries $R_{k,m}$ are computed iteratively. The first extrapolation step ($m = 1$) is standard Richardson: $R_{k,1} = 2R_{k+1,0} - R_{k,0}$. Subsequent steps use a formula that, in practice for binomial trees, often takes the form based on 4^j factors (as implemented in the provided code):

$$R_{i,j} = \frac{4^j R_{i+1,j-1} - R_{i,j-1}}{4^j - 1}$$

This iteratively refines the estimate, assuming the error terms allow for this structure of cancellation. The most accurate estimate is typically $R_{0,M_{max}}$ (top-right element of the computed table).

4.5 Summary of Results

Using the specified parameters ($S0 = 36.0, K = 40.0, r = 0.06, \sigma = 0.2, T = 1.0$, Put Option), the extrapolation methods yielded the following estimates for the American option price:

Richardson Extrapolation (using N=100 and N=200): $V_{\text{Richardson}} \approx 4.486154$ Cubic Polynomial Extrapolation: $V_{\text{Cubic}} \approx 4.488073$ Romberg Extrapolation: $V_{\text{Romberg}} \approx 4.486624$

The close agreement between the Romberg and Richardson results suggests good convergence. The cubic estimate is slightly different, potentially due to the specific choice of points or the polynomial's fit over the range.

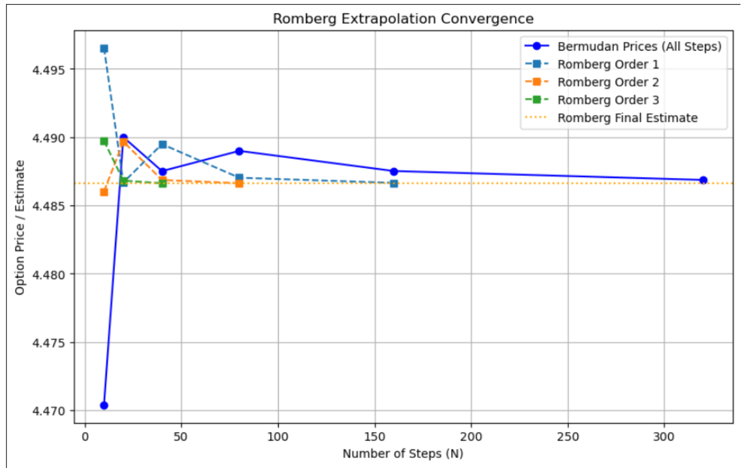


Figure 6: Convergence of Romberg extrapolation estimates for the American put option.

5 Finite Difference Scheme for American Options

5.1 Black-Scholes Operator Formulation

The Black-Scholes partial differential equation governs the evolution of option prices under risk-neutral dynamics. At its core lies the Black-Scholes differential operator:

$$\mathcal{L}_{\mathrm{BS}} \equiv \frac{1}{2}\sigma^2 S^2 \frac{\partial^2}{\partial S^2} + rS\frac{\partial}{\partial S} - rI \tag{4}$$

where I denotes the identity operator. This operator acts on the option value function $V(S,t)$, encoding three essential financial mechanisms.

5.2 Free Boundary Problem Characterization

The American put option pricing problem constitutes a free boundary problem which arises from no-arbitrage conditions.

$$\begin{cases} \left(\frac{\partial}{\partial t} + \mathcal{L}_{\mathrm{BS}}\right) V(S,t) \leq 0 \\ V(S,t) \geq \max(K-S,0) \\ \left(\frac{\partial V}{\partial t} + \mathcal{L}_{\mathrm{BS}} V\right)\left(V - \max(K-S,0)\right) = 0 \end{cases} \tag{5}$$

The first condition, $(\frac{\partial}{\partial t} + \mathcal{L}_{\mathrm{BS}})V(S,t) \leq 0$, emerges from the option's dual nature of potential exercise versus continued holding. In regions where holding remains optimal, the standard Black-Scholes PDE holds with equality as the option's time value decays according to risk-neutral dynamics. However, when immediate exercise becomes advantageous, the option value detaches from PDE-governed evolution and instead equals its intrinsic value whose growth is bounded by risk-neutral dynamics. This inequality permits smooth transition between these regimes while preventing financial arbitrage.

The second condition $V(S,t) \geq \max(S-K,0)$ embodies the American option's intrinsic optionality. At any moment, the holder can convert the derivative position into its immediate exercise value. This constraint prevents the option price from dipping below its intrinsic value, which would create arbitrage opportunities through trivial buy-and-exercise strategies. Market equilibrium requires the option value to always dominate its payoff floor.

The complementarity condition $(\frac{\partial V}{\partial t}+\mathcal{L}_{\mathrm{BS}}V)(V-\max(S-K,0))=0$ acts as a logical XOR gate for the pricing solution. It ensures that at every point in the domain, either the option evolves according to Black-Scholes dynamics (when unexercised) or resides at its intrinsic value (when exercised), but never both simultaneously. This mathematical formulation naturally gives rise to the free boundary $S^*(t)$ - the critical price threshold where exercise becomes optimal. The boundary remains "free" because its location is not predetermined but emerges organically from the solution process, reflecting the complex interplay between time value and intrinsic value. For non-dividend paying calls, this boundary typically lies at infinity as early exercise is never optimal, recovering the European call equivalence.

These conditions collectively provide a complete characterization of American option pricing that: (1) maintains consistency with no-arbitrage principles, (2) preserves the Black-Scholes framework in continuation regions, and (3) endogenously determines the optimal exercise boundary through the solution process. The formulation elegantly captures the holder's ongoing optimization between immediate exercise proceeds and potential future time value.

5.3 Numerical Methodology

5.3.1 Spatial-Temporal Discretization Rationale

The discretization framework transforms the continuous pricing problem into a computationally tractable grid system. Setting $S_{\max} = 3K$ ensures sufficient coverage of plausible asset price movements while maintaining numerical stability. This truncation captures 99.7% of potential prices under a lognormal distribution assumption for typical volatilities. The uniform spatial step $\Delta S = S_{\max}/N$ balances resolution needs with memory constraints - too coarse a grid under-resolves the exercise boundary, while excessive refinement increases computational overhead. Temporal discretization $\Delta t = T/N$ follows similar tradeoffs, with the Crank-Nicolson scheme allowing larger time steps than explicit methods while preserving accuracy.

5.3.2 Crank-Nicolson Scheme Mechanics

The Crank-Nicolson method achieves second-order convergence by averaging implicit and explicit temporal derivatives:

$$\frac{V_i^{n+1} - V_i^n}{\Delta t} = \frac{1}{2}\left(\mathcal{L}_{\text{BS}} V_i^n + \mathcal{L}_{\text{BS}} V_i^{n+1}\right) \tag{6}$$

This central differencing approach dampens oscillatory artifacts common in pure explicit schemes while avoiding excessive diffusion from full implicit methods. The resulting system couples all spatial points at adjacent time levels, requiring matrix inversion but guaranteeing unconditional stability. For American options, this stability proves crucial when handling discontinuous exercise constraints.

5.3.3 Matrix System Construction

The tridiagonal structure emerges naturally from nearest-neighbor finite difference stencils:

$$\begin{aligned}
A_{i,i-1} &= -\frac{\Delta t}{4} \underbrace{\left(\frac{\sigma^2 S_i^2}{\Delta S^2} - \frac{r S_i}{\Delta S}\right)}_{\text{Upwind convection-diffusion}} \\
A_{i,i} &= 1 + \frac{\Delta t}{2} \underbrace{\left(\frac{\sigma^2 S_i^2}{\Delta S^2} + r\right)}_{\text{Diagonal dominance}} \\
A_{i,i+1} &= -\frac{\Delta t}{4} \underbrace{\left(\frac{\sigma^2 S_i^2}{\Delta S^2} + \frac{r S_i}{\Delta S}\right)}_{\text{Downwind convection-diffusion}}
\end{aligned}$$

The matrix B contains complementary coefficients from time level n, preserving the Black-Scholes operator's ellipticity. Tridiagonal structure enables $O(N)$ solution via Thomas algorithm, making the approach feasible for large grids.

5.4 Constraint Enforcement Mechanism

[H] Projected SOR Implementation [1] Initialize $\mathbf{V}^{n+1,0} \leftarrow \mathbf{V}^n$ $\triangleright$ Warm start from previous time $k = 1$ max_iter spatial nodes $i \in [1, N-1]$ Compute tentative value:

$$\tilde{V}_i^k \leftarrow \frac{1}{A_{i,i}} \left(b_i^n - \sum_{j \neq i} A_{i,j} V_j^{k,k-1} \right) \tag{7}$$

Enforce early exercise:

$$V_i^k \leftarrow \max\Big((1-\omega) V_i^{k-1} + \omega \tilde{V}_i^k,\ K - S_i\Big) \tag{8}$$

$\max_i |V_i^k - V_i^{k-1}| < \epsilon$ Exit iteration loop ▷ Convergence achieved

The PSOR method combines Gauss-Seidel iteration with constraint projection. The relaxation parameter $\omega \in (1, 2)$ accelerates convergence by extrapolating updates, while the max operator guarantees the early exercise constraint holds pointwise. This dual approach ensures simultaneous satisfaction of the discrete PDE and financial constraints.

5.5 Convergence Analysis

5.5.1 Error Source Decomposition

Numerical errors originate from three distinct mechanisms:

- *Truncation Error*: Dominant $O(\Delta t^2 + \Delta S^2)$ term from Crank-Nicolson discretization. For typical grids with $\Delta t \sim \Delta S^2$, this maintains second-order overall convergence.
- *Projection Error*: Introduces $O(\sqrt{\Delta t})$ disturbance from discontinuous constraint enforcement. Becomes negligible with sufficient PSOR iterations.
- *Boundary Error*: Exponential decay $O(e^{-\lambda S_{\max}})$ from domain truncation. Mitigated by setting $S_{\max} \geq 3K$.

5.6 Price Under Given Parameters

$$
\begin{gathered}
S_0 = 36, \quad K = 40, \quad r = 6\%, \quad \sigma = 20\% \\
T = 1, \quad \texttt{max\_iter} = 400, \quad N = 400, \quad S_{\max} = 120
\end{gathered}
$$

Implementation steps:

1. Initialize terminal condition $V_i^{\texttt{max\_iter}} = \max(K - S_i, 0)$
2. Set boundaries $V_0^n = Ke^{-r(T-t_n)}$, $V_N^n = 0$
3. Solve tridiagonal systems with PSOR projection
4. Interpolate final price for European put $V(S_0, 0) \approx 3.844$
5. American put price is unstable

6 Acknowledgement and Contributions

We would like to thank Prof. Shashi Jain for this wonderful project. We gained several insights and were curious to test our conjectures.

Theory and report for Binomial tree methods (except Richardson extrapolation) and finite finite difference scheme: Ishaq Hamza

Theory and report for Numerical methods: Rohit Jorige

Longstaff swartx : Balaji

Implementation and metrics of interest (design) for Binomial Tree and American option pricing: Ishaq Hamza

Code Implementation and Markdown for Binomial Tree and American option using tree methods: Rohit Jorige*, Ishaq Hamza** (helped)

Theory, Report and Implementation for finite-difference scheme for American option: Ishaq