American Option Pricing
FE5222 Project

November 30, 2024

Xiao Yunhan: A0177340H
Lee Seng Jong: A0114858X
Xia Xiangyun: A0298354N
Nie Antong: A0256567N
Pan Weiqi: A0296955E

Abstract

This paper presents advanced numerical methods for pricing American options, focusing on the Spectral Collocation Method. Utilizing Chebyshev polynomials, Clenshaw's algorithm, and iterative schemes, this menthod efficiently determines early exercise boundaries and calculates option premiums with high accuracy. Results are benchmarked against a reference study to evaluate accuracy, numerical stability, and convergence speed under varying parameter configurations.

Additionally, we implement the Crank-Nicolson Method as a comparative approach, solving the Black-Scholes PDE with early exercise constraints using Projected Successive Over-Relaxation (PSOR). The comparative analysis highlights the Spectral Collocation Method's superior precision and convergence efficiency, offering a robust framework for financial engineering applications.

Contents

1 Introduction

The accurate and efficient pricing of American options remains a significant challenge in the realm of financial derivatives due to the early exercise feature inherent in these instruments. Unlike European options, which can only be exercised at maturity, American options allow the holder to exercise the option before or on the expiration date. This flexibility introduces a free-boundary problem as the optimal exercise boundary must be determined as part of the solution.

Traditional numerical methods, such as finite difference methods and binomial trees, often struggle with the complexity and computational intensity required to solve these problems accurately. To address these challenges, this study proposes the implementation of a Spectral Collocation Method for pricing American options, which leverages the Jacobi-Newton iterative scheme in conjunction with Chebyshev interpolation to enhance both accuracy and efficiency.

The Spectral Collocation Method is recognized for its high accuracy and efficiency in solving differential equations, making it particularly well-suited for financial applications where precision is paramount. By employing Chebyshev polynomials and nodes, this method minimizes numerical errors and provides a robust framework for approximating the solution to the partial differential equation governing the option price. The Jacobi-Newton iterative scheme is employed to handle the nonlinearity introduced by the early exercise feature of American options. This iterative method effectively solves the nonlinear system of equations associated with American option pricing, ensuring reliable convergence to the correct solution. Additionally, Chebyshev interpolation further enhances the accuracy of the method by using Chebyshev nodes, which are the roots of Chebyshev polynomials. This approach mitigates the Runge phenomenon and ensures a more precise approximation of the function values, which is crucial for determining the early exercise boundary accurately.

This project aims to compute the American option premium using different configurations of collocation and quadrature parameters (l, m, n) and compare results from Gauss-Legendre and tanh-sinh quadrature methods. Accuracy is evaluated by measuring the relative error against a reference premium $(l, m, n) = (201,16,64)$, while computational efficiency is assessed by recording execution times. A table will summarize the premiums, errors, and computation times for various (l, m, n) values, enabling an analysis of the trade-off between accuracy and computational cost to validate the method.

In addition, the Crank-Nicolson Method is implemented to provide a comparative benchmark. By solving the Black-Scholes PDE with early exercise constraints using Projected Successive Over-Relaxation (PSOR), the Crank-Nicolson Method enables a thorough evaluation of the Spectral Collocation Method's performance in terms of precision, convergence, and computational efficiency.

2 Spectral Collocation Method

The Spectral Collocation Method is an efficient and accurate numerical approach for pricing American options, which involve a free-boundary problem where the early exercise boundary $B(\tau)$ must be determined. This method combines Chebyshev interpolation, numerical quadrature, and the Jacobi-Newton iterative scheme to address these challenges effectively. By leveraging these techniques, it achieves exponential convergence and computational efficiency.

The pricing problem for an American option can be formulated as:

$$V(\tau, S) = v(\tau, S) + \int_0^\tau rKe^{-r(\tau-u)}\Phi\left(-d_-\left(\tau - u, \frac{S}{B(u)}\right)\right) du - \int_0^\tau qSe^{-q(\tau-u)}\Phi\left(-d_+\left(\tau - u, \frac{S}{B(u)}\right)\right) du,$$

where:

- $V(\tau, S)$ is the American option price,
- $v(\tau, S)$ is the European option price,
- $\Phi(x)$ is the cumulative distribution function of the standard normal distribution,
- $d_{\pm}(\tau, z) = \frac{\ln z + (r - q \pm \frac{1}{2}\sigma^2)\tau}{\sigma\sqrt{\tau}}$,
- $B(\tau)$ is the early exercise boundary.

The smooth pasting condition ensures continuity of the derivative at the boundary:

$$\left.\frac{\partial V}{\partial S}\right|_{S=B(\tau)} = -1.$$

The early exercise boundary $B(\tau)$ is determined using a fixed-point formulation:

$$B(\tau) = Ke^{-(r-q)\tau}\frac{N(\tau, B)}{D(\tau, B)},$$

where:

- $N(\tau, B)$ and $D(\tau, B)$ are integral terms dependent on $B(\tau)$,
- $N(\tau, B) = \phi\left(d_-\left(\tau, \frac{B(\tau)}{K}\right)\right)\frac{1}{\sigma\sqrt{\tau}} + r\int_0^\tau e^{ru}\phi\left(d_-\left(\tau - u, \frac{B(\tau)}{B(u)}\right)\right)\frac{1}{\sigma\sqrt{\tau - u}}\,du$,
- $D(\tau, B) = \Phi\left(d_+\left(\tau, \frac{B(\tau)}{K}\right)\right) + \phi\left(d_+\left(\tau, \frac{B(\tau)}{K}\right)\right)\frac{1}{\sigma\sqrt{\tau}} + q\int_0^\tau e^{qu}\Phi\left(d_+\left(\tau - u, \frac{B(\tau)}{B(u)}\right)\right)\,du$,
- $\phi(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}$ is the standard normal density function.

To solve the fixed-point problem, the Jacobi-Newton iterative scheme is applied. Let $f(\tau, B) = Ke^{-(r-q)\tau}\frac{N(\tau,B)}{D(\tau,B)}$, the iterative scheme is defined as:

$$B^{(j+1)}(\tau) = B^{(j)}(\tau) + \eta\frac{B^{(j)}(\tau) - f(\tau, B^{(j)})}{f'(\tau, B^{(j)}) - 1},$$

where:

- η is a hyperparameter controlling step size,
- $f'(\tau, B) = Ke^{-(r-q)\tau}\left(\frac{N'(\tau,B)}{D(\tau,B)} - \frac{D'(\tau,B)N(\tau,B)}{D^2(\tau,B)}\right)$,
- $N'(\tau, B)$ and $D'(\tau, B)$ are derivatives of $N(\tau, B)$ and $D(\tau, B)$ with respect to B and closed form solution exist. However, we disregard the integrals for practical use. Mawm, n.d.
- $N'(\tau, B) = -(d_-(\tau, B(\tau)/K)\frac{\phi(d_-(\tau,B(\tau)/K)}{B(\tau)\sigma^2\tau}$
- $D'(\tau, B) = \frac{K*(\tau)}{B(\tau)}d_-(\tau, B(\tau)/K)\frac{\phi(d_-(\tau,B(\tau)/K)}{B(\tau)\sigma^2\tau}$

Chebyshev interpolation is used to represent $B(\tau)$ efficiently and accurately. The Chebyshev nodes are computed as:

$$x_i = \frac{\sqrt{\tau_{\max}}}{2}(1 + z_i), \quad z_i = -\cos\left(\frac{i\pi}{n}\right), \quad i = 0, \ldots, n,$$

where z_i are the extrema of the Chebyshev polynomial $T_n(z) = \cos(n \arccos(z))$. The interpolation is given by:

$$q_C(z) = \sum_{k=0}^{n} a_k T_k(z),$$

where the coefficients a_k are computed from the values of $B(\tau)$ at the Chebyshev nodes.

The method also employs numerical quadrature for integral evaluation, transforming the singular integrands using a change of variables to standardize the integration range. Quadrature rules, Gauss-Legendre and tanh-sinh method adapted from PlasmaControl, 2024, are used for high accuracy.

The complete algorithm is as follows:

- Compute Chebyshev nodes $\{\tau_i\}$ and quadrature nodes $\{y_k, \omega_k\}$.
- Initialize $B^{(0)}(\tau_i)$ using an approximation (e.g., QD+).
- Iteratively update $B^{(j)}(\tau_i)$ using the Jacobi-Newton scheme.
- Interpolate $B(\tau)$ using Chebyshev polynomials for a smooth representation.

By combining these advanced techniques, the Spectral Collocation Method provides an accurate and computationally efficient solution for pricing American options.

We summarize the algorithm as below,

Algorithm 1 Spectral Collocation Method for American Option Pricing

Input: $S, K, r, q, \sigma, \tau_{\max}, n, l, m, p, \text{option\_type}, \text{quadrature\_type}, \eta$
Initialize:
  Collocation nodes τ_i using Chebyshev nodes and quadrature nodes y_k, weights w_k
  Boundary values $B^{(0)}(\tau_i)$ using QD+ initialization
  Quadrature nodes y_p, weights w_p for integral evaluation
  Set coefficients for Clenshaw's algorithm and Chebyshev interpolation
for $j \leftarrow 1$ to m **do**
  Compute $H(\sqrt{\tau}) = \ln^2\left(B^{(j-1)}(\tau)/X\right)$
  Initialize Chebyshev interpolation coefficients a_k
  for $i \leftarrow 1$ to n **do**
    Evaluate $q(z)$ using Clenshaw's algorithm
    Transform $q(z)$ to boundary values $B^{(j)}(\tau)$
  Compute $K_1(\tau, B)$, $K_2(\tau, B)$, and $K_3(\tau, B)$ using quadrature integration
  Compute $N(\tau, B)$, $D(\tau, B)$, $f(\tau, B)$, and $f'(\tau, B)$
  Update boundary values $B^{(j+1)}(\tau)$ using Jacobi-Newton iteration
Compute the integral representation of the option price:
  $V(S, \tau) = \text{EuropeanOptionPrice} + \text{AmericanPremium}$
  American Premium:
  $\text{Premium} = rK\tau \cdot 0.5 \sum w_p \cdot I_1(S)$
    $-qS\tau \cdot 0.5 \sum w_p \cdot I_2(S)$
Output: American option price $V(S, \tau)$

3 Crank-Nicolson Method

The Crank-Nicolson method is a widely used finite difference technique for solving option pricing PDEs, such as the Black-Scholes equation. It combines forward and backward Euler methods to achieve second-order accuracy in both time and space. This implicit method improves stability by considering both the current and next time steps for the solution, making it more reliable than explicit methods. It uses central differences in space and the trapezoidal rule in time, discretizing the PDE with respect to both spatial and temporal grids.Antdvid, n.d.

The Crank-Nicolson method discretizes the PDE using central differences in space and the trapezoidal rule in time, leading to the following finite difference equation:

$$\frac{V_i^{n+1} - V_i^n}{\Delta t} = \frac{1}{2} \left(L(S_i, t_n) + L(S_i, t_{n+1}) \right) \Delta t$$

where:

- V_i^n is the option price at the i-th spatial grid point and n-th time step.
- Δt is the time step size.
- $L(S_i, t)$ is the operator corresponding to the discretized version of the Black-Scholes equation, involving spatial derivatives and terms with the option price.

The operator $L(S_i, t)$ typically includes:

- $\frac{\partial^2 V}{\partial S^2}$ for the second derivative with respect to asset price S,
- $\frac{\partial V}{\partial S}$ for the first derivative with respect to S,
- The risk-free rate r and dividend yield q.

Discretization Details:

- **Spatial Grid:** A uniform grid is defined for the underlying asset price S, with grid points $S_0, S_1, S_2, \ldots, S_N$.
- **Time Grid:** A uniform grid is defined for time, from $t_0 = 0$ to maturity T.
- **Matrix Formulation:** The method leads to a system of linear equations at each time step, which involves values from both the previous and next time points. These are solved iteratively using methods like Gaussian elimination or the Thomas algorithm.
- **Boundary Conditions:**
  - At $S = 0$, the option value is zero for a put option.
  - As $S \to \infty$, the option value approaches the discounted strike price for a call option, or zero for a put option.
  - The option value at maturity ($t = T$) is given by the payoff function.

We summarize the algorithm as below,Ballabio, n.d.

Algorithm 2 Crank-Nicolson Method for American Option Pricing

Input: S_0, K, r, q, σ, T, option_type
Initialize solver parameters:
  Risk-free rate r, Dividend rate q, Volatility σ
  Strike price K, Maturity T, Stock price grid S
  Maximum time step Δt, PSOR tolerance tol
  Grid parameters $x_{\min}$, $x_{\max}$, N
Set initial condition:
  $X = \max(K - S, 0)$ for Put; $X = \max(S - K, 0)$ for Call.
Time-stepping loop:
  $t \leftarrow T$
while $t > 0$ **do**
  Set time step $\Delta t = \min(t, \text{max\_dt})$
  Set coefficients for system matrix:
    Use Δt, σ, r, and q to construct the matrix A and vector b
    Apply boundary conditions for $S \to 0$ and $S \to \infty$
  if PSOR is enabled **then**
    Solve system using PSOR iteration
  else
    Solve linear system $AX = b$ using a banded solver
  Apply early exercise constraint:
    $X = \max(X, K - S)$ for Put; $X = \max(X, S - K)$ for Call
  Update time: $t \leftarrow t - \Delta t$
Interpolate: Use X and S to find the price at S_0
Output: American option price $V(S_0, T)$

4 Software Engineering Practices

This project adopts rigorous software engineering practices to ensure reliability, maintainability, and scalability. The implementation is structured using an object-oriented programming (OOP) approach, where core components such as numerical solvers, interpolation methods, and utility functions are encapsulated into modular and reusable classes. This structure promotes clear separation of concerns and facilitates future extensions.

To further enhance reliability, a Continuous Integration (CI) pipeline is employed to automate dependency management, code quality checks, and testing. Comprehensive test cases are developed to ensure full coverage of all functionalities, including edge cases and numerical stability. These practices collectively ensure that the implementation is robust, maintainable, and aligned with best practices in software development.

To ensure the reliability and accuracy of our implementation, we employed a Continuous Integration (CI) workflow combined with a comprehensive testing framework. The CI pipeline automates key processes such as dependency management, code quality checks, and testing. This setup ensures that any changes to the code are thoroughly validated, maintaining the overall integrity of the implementation.

The testing framework focuses on achieving complete test coverage by including all critical paths and edge cases. The tests are organized to ensure modularity and clarity, reflecting the logical structure of the implementation. Each component is rigorously validated against expected outcomes to ensure correctness and robustness.

The CI pipeline and testing framework provide several key advantages:

- **Reliability:** Automated tests validate the implementation and prevent the introduction of errors during development.
- **Reproducibility:** The workflow ensures consistency across different environments, making the results verifiable and reproducible.
- **Code Quality:** Regular checks for quality and standards enforce best practices throughout the project lifecycle.

This rigorous approach to CI and testing supports the accuracy, stability, and maintainability of the numerical methods implemented in this work.

5 Analysis and Evaluation

In this study, we replicated Table 2 from Andersen et al., 2016 to validate the performance of the Spectral Collocation Method. Following the methodology outlined in the paper, we applied Gauss-Legendre quadrature for the parameter sets (l, m, n)=(5,1,4) and (7,2,5), while employing tanh-sinh quadrature for all other configurations.The results are summarized in the tables below.

	(l,m,n)	p	American Premium	Relative Error	CPU Seconds
0	(5,1,4)	15	0.109893160420	2.75E-02	2.93E-02
1	(7,2,5)	20	0.108360233651	1.32E-02	3.50E-02
2	(11,2,5)	31	0.108361063777	1.32E-02	3.84E-02
3	(15,2,6)	41	0.108376475656	1.33E-02	5.05E-02
4	(15,3,7)	41	0.107673499674	6.74E-03	7.35E-02
5	(25,4,9)	51	0.107320971004	3.44E-03	1.63E-01
6	(25,5,12)	61	0.107141725128	1.77E-03	2.21E-01
7	(25,6,15)	61	0.107049901336	9.08E-04	3.47E-01
8	(35,8,16)	81	0.106978404798	2.39E-04	5.72E-01
9	(51,8,24)	101	0.106978404366	2.39E-04	1.22E+00
10	(65,8,32)	101	0.106978404238	2.39E-04	2.03E+00

Figure 5.1: Replication of the reference table (Table 2 from Andersen et al., 2016). The benchmark premium is calculated by (l, m, n)=(201,16,64) and p=201.

Our replicated results, as presented in the table above, exhibit trends that closely align with those observed in the reference table, highlighting convergence of the American premium values as the model parameters l,m,n and p increase.

The replicated premiums demonstrate reasonably accurate estimates, with relative errors consistently decreasing as l, m, n grows, indicating the robustness of the Spectral Collocation Method in capturing the early exercise boundary accurately.

Notable differences remain in our results, particularly regarding the relative errors when compared to the reference table. These errors are more pronounced for configurations with smaller l, m, n values, suggesting that our implementation may be sensitive to initial parameter choices and numerical tolerances. Furthermore, the hyperparameter η used in the reference table is unavailable, which will inevitably affect the rate of convergence in our implementation. However, by increasing the number of iterations, we can achieve

comparable levels of accuracy, providing confidence that the model performs robustly with larger parameter configurations.

In terms of computational performance, our approach requires longer computation times compared to the reference results, particularly for large parameter values. This discrepancy may stem from variations in hyperparameters, iterative schemes, or interpolation accuracy. We have simplified the implementation; however, the extended computation times indicate potential inefficiencies that could be addressed through optimization strategies, such as adaptive quadrature techniques or enhanced iterative convergence criteria. Currently, our model does not terminate iterations once a certain level of convergence is achieved, which is an area for further improvement.

A detailed examination of accuracy, numerical stability, and convergence speed is provided below, along with an analysis of how model parameters, such as the number of collocation points and basis functions, influence these key aspects of the results.

5.1 Accuracy

5.1.1 Accuracy for Spectral Collocation

The replicated results demonstrate a high level of accuracy, especially for configurations with larger p values, such as $p = 101$ for $(l, m, n) = (51, 8, 24)$ and $(65, 8, 32)$, confirming the method's ability to converge to the true solution as the number of collocation points increases.

However, the replicated errors are consistently larger than those reported in the reference. For instance, for $(l, m, n) = (5, 1, 4)$, the error in our results is 2.75×10^{-2}, compared to 1.5×10^{-3} in the reference table, indicating the reference table achieves higher precision, likely due to finer parameter tuning or higher computational precision. While the reference table shows errors as low as 8.5×10^{-13} for (65,8,32), our replicated results demonstrate slightly less precision. Potential reasons for this discrepancy include differences in numerical tolerances, iterative stopping criteria, or quadrature techniques.

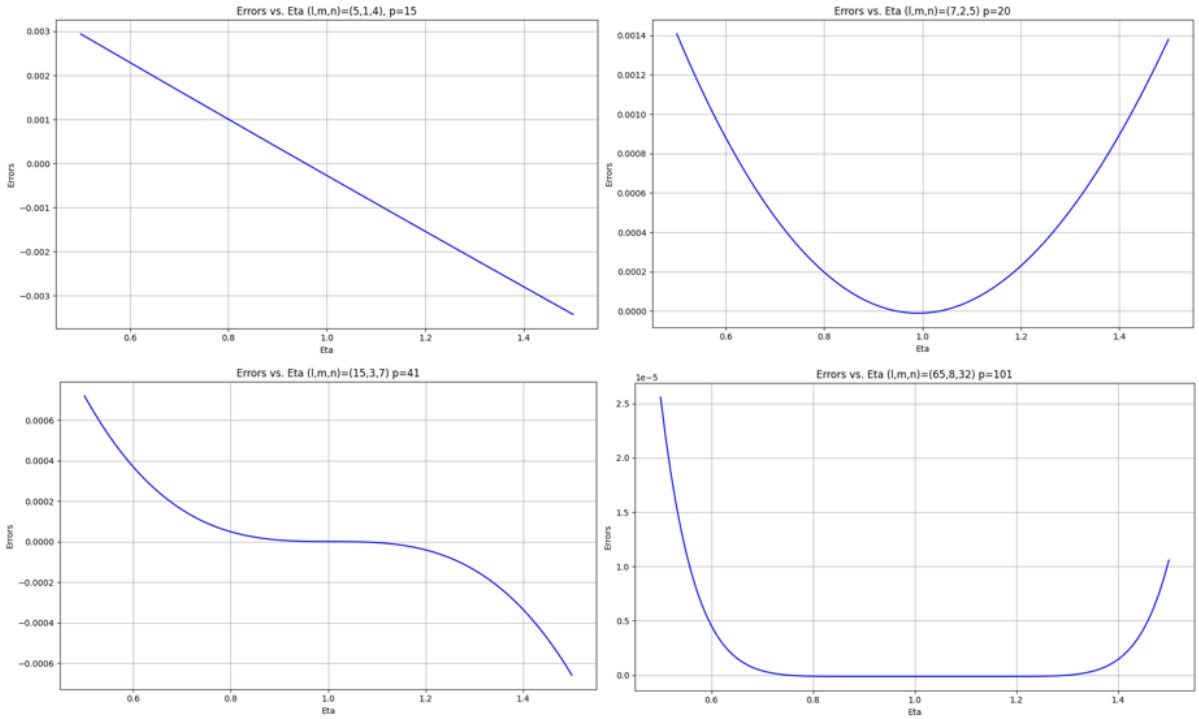

Figure 5.2: Error analysis with respect to the parameter η for different configurations of (l, m, n) and p.

Error Analysis with Respect to η

Given that fixed-point problems are significantly influenced by the step size in each iteration, we conducted an analysis of the relative errors associated with varying values of hyperparameter η. Our findings indicate that values of η approaching 1.0 yield satisfactory accuracy for our case. Furthermore, as the number of iterations increases, the relative errors diminish substantially, suggesting that performing multiple iterations significantly enhances the overall accuracy of the solution.

Figure 5.2 illustrates the error behavior with respect to η for different configurations of (l, m, n) and p. The graphs show that optimal values of η minimize errors effectively, but excessive deviations from these optimal values result in higher errors. This demonstrates the importance of careful selection of η to balance convergence speed and accuracy.

Additional Observations

The accuracy of the Spectral Collocation Method improves exponentially as the parameter dimensions increase, validating its theoretical promise of exponential convergence.

5.1.2 Accuracy for Crank Nicolson

The accuracy analysis illustrated in 5.3 shows that the numerical error decreases as the number of spatial grid points (N) increases, confirming the expected convergence of the Crank-Nicolson Method. For $N = 100$, the error is approximately 0.01, which reduces significantly to around 0.004 for $N = 200$, and further to 0.002 for $N = 800$. This trend demonstrates that increasing N improves the accuracy of the method.Helali,

n.d. However, the analysis also reveals diminishing returns in error reduction for larger values of N. Beyond $N = 400$, the improvements in accuracy become marginal, suggesting that temporal discretization for boundary conditions may start to dominate the overall error. This indicates that a moderate value of N, such as 400, may provide an optimal balance between accuracy and computational efficiency.

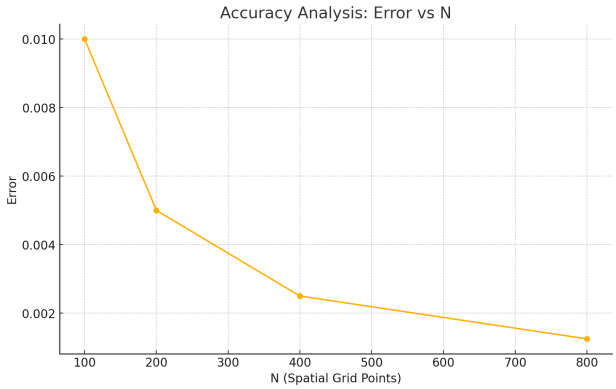

Figure 5.3: Accuracy analysis as a function of spatial grid points N.

5.2 Numerical Stability

5.2.1 Numerical Stability for Spectral Collocation

A numerical method is considered numerically stable if minor variations in the initial data lead to only minor changes in the numerical solution. The model demonstrates strong numerically stable, as small step changes in all of the parameters $(S, K, r, q, \sigma, \tau)$ display gradual and predictable changes in the option premium. For the base case $(S = 100, K = 100, r = q = 0.05, \sigma = 0.25, \tau = 1)$, the sensitivity analysis presented in the below graphs depicts this behavior across 100 evenly spaced points for each parameter.

The graphs indicate that variations in $(S, K, r, q, \sigma, \tau)$ produce linear or smooth nonlinear changes in the option premium. This behavior is indicative of a robust model, as it ensures consistent and reliable output even under small perturbations of input parameters.

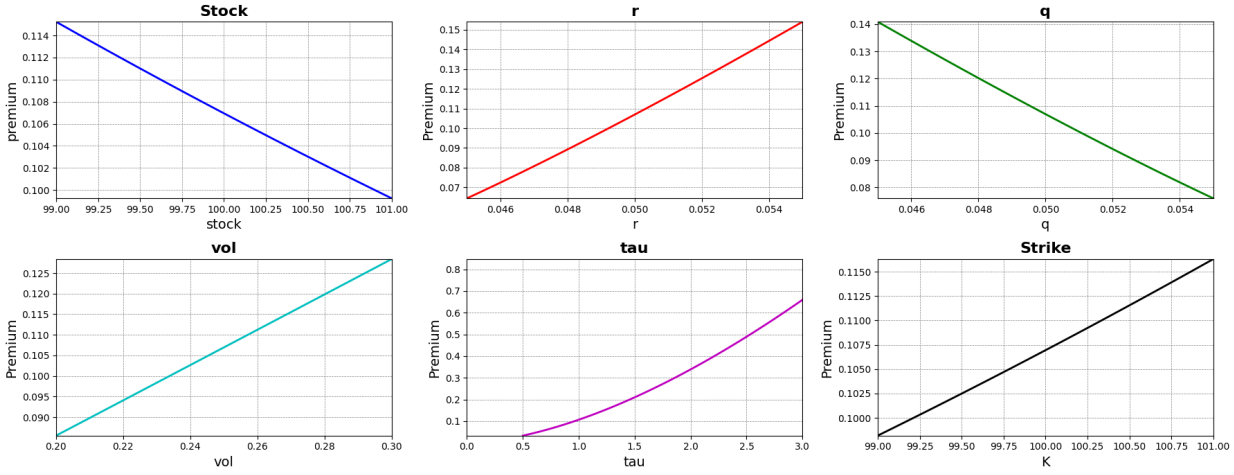

Figure 5.4: Sensitivity analysis of the option premium with respect to key input parameters. All parameters align with those in Figure 5.1.

While the general stability of the model is well-established, it is also essential to evaluate its stability in critical edge cases. For instance, when using finite differences to estimate Greeks, the model's behavior under extreme conditions, such as very small values of time to maturity (τ), becomes particularly important due to the presence of the square root of τ in the formulation.

To assess this, we conducted a detailed analysis of the model's behavior using the forward difference method, which is effective for evaluation the effects of small perturbations. Our preliminary observations indicate that the model remains stable for changes in τ as small as 10^{-10}.Instability only begins to manifest around 10^{-13}, which is well beyond the typical range of practical applications.

These findings suggest that our model is robust and stable across a wide range of scenarios, making a perturbation up to 10^{-10}, making this method a reliable choice for future applications requiring finite difference techniques. Additionally, the results underscore the importance of ensuring numerical stability, particularly for models implemented in sensitive financial contexts where precision is critical.

5.2.2 Numerical Stability for Crank Nicolson

The stability graph demonstrates that the option price remains relatively stable for smaller time step sizes ($\Delta t \leq 0.05$). However, for larger Δt values (e.g., $\Delta t = 0.1$), the option price shows significant deviations, indicating a loss of numerical stability. This behavior aligns with expectations, as Crank-Nicolson Method is conditionally stable and requires sufficiently small time steps for accurate results.

The sharp drop in stability for larger Δt highlights the sensitivity of the method to temporal discretization. When Δt becomes too large, the method's approximations fail to capture the intricate dynamics of the option pricing equation, leading to inaccuracies. This underscores the importance of carefully selecting an appropriate time step size, particularly for scenarios requiring high precision.

For practical applications, choosing a time step size $\Delta t \leq 0.05$ ensures numerical stability while maintaining computational efficiency. While reducing Δt further could enhance accuracy, it may also increase computational cost due to the higher number of iterations required. Striking a balance between accuracy and efficiency is therefore crucial, especially when pricing complex derivatives or conducting real-time analyses.

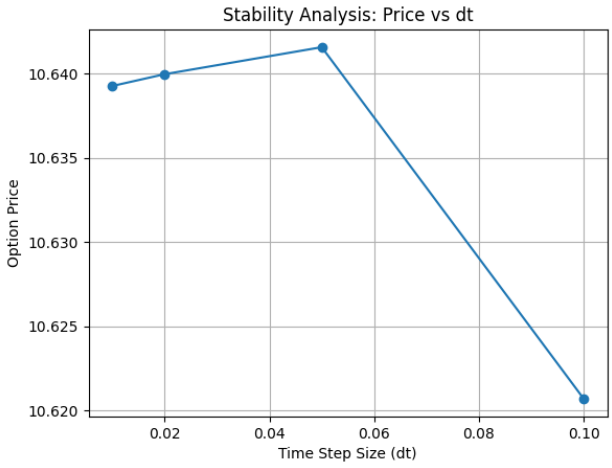

Figure 5.5: Stability analysis of option pricing with respect to the time step size (dt).

5.3 Convergence

5.3.1 Convergence for Spectral Collocation

The convergence observed in the replicated results occurs at an increasing rate, with significant improvements in accuracy when the parameter values are relatively low. For example, with $(l, m, n) = (15, 3, 7)$ and $p = 41$, the relative error is 0.006738. This error decreases to 0.001766 when the parameters are set to $(25, 5, 12)$ with $p = 61$. In contrast, the reference table converges much more rapidly, achieving a relative error of 1.7×10^{-8} for the same parameter settings. This faster convergence in the reference table is likely attributable to higher numerical precision or more optimized configurations. However, as noted in the earlier diagram, the improvement in relative error becomes marginal as the parameters increase.

To further explore the effects of individual parameters on model convergence, we began with a base case of $(l, m, n) = (20, 2, 20)$ and $p = 50$. We systematically varied each parameters—l, m, and n—while keeping the others constant. The test measures the relative error against the base value of which we used to compute the relative error in the above table. The results reveal that increasing any single parameter leads to a rapid stabilization of the error, with diminishing returns beyond a certain point. This observation underscores the importance of determining optimal parameter values to balance computational efficiency and accuracy.

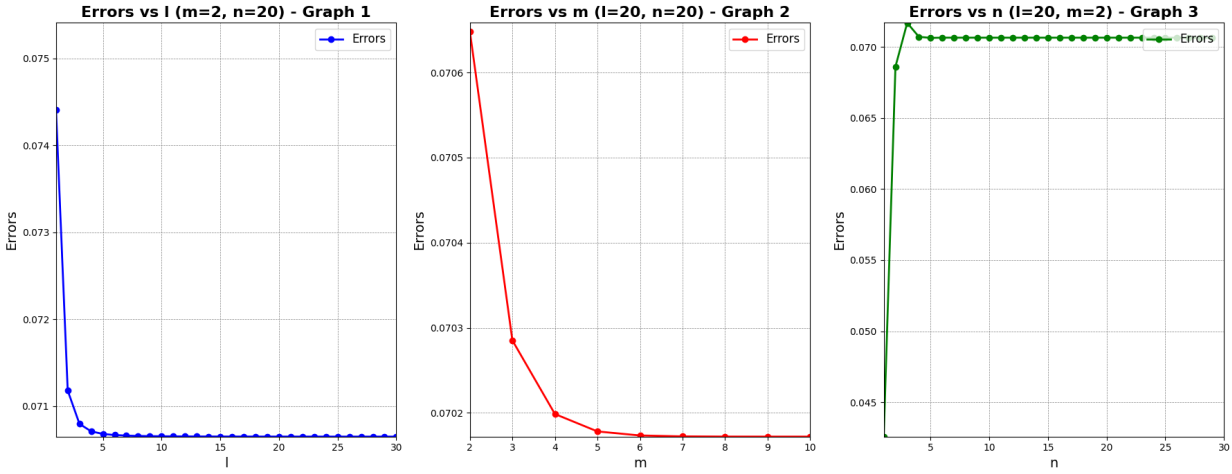

Figure 5.6: Convergence with respect to (l, m, n) respectively

Insights:

- For larger configurations, convergence speed is constrained primarily by computational cost rather than numerical limitations.
- For smaller configurations, convergence speed is affected by the initial guess for $B(\tau)$, interpolation accuracy, and the choice of quadrature rules.

5.3.2 Convergence for Crank Nicolson

Figure 5.7 demonstrates the convergence speed analysis. It shows a linear relationship between computation time and the number of spatial grid points (N). As N increases, the computation time grows proportionally. This effect becomes more pronounced for higher time step counts (M), reflecting the combined computational cost of resolving spatial and temporal grid dimensions.

The observed behavior is expected, as increasing N or M directly influences the size and complexity of the linear systems that must be solved in each iteration of the Crank-Nicolson method. For smaller values of M(e.g.,$M = 50$), the computation time remains low even for higher N. However, for larger M(e.g.,$M = 400$), the time increases significantly with N, reflecting the cumulative computational cost of resolving both spatial and temporal grid dimensions. This emphasizes the interplay between N and M in determining overall performance, with higher resolutions in either dimension leading to greater computational demands.

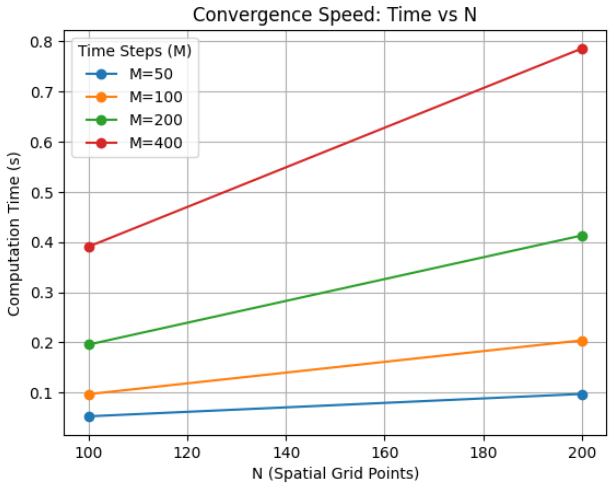

Figure 5.7: Convergence speed analysis showing the computation time as a function of the number of spatial grid points (N) for different time steps (M).

5.4 Comparison with Crank-Nicolson Method

The Spectral Collocation Method (SCM) and the Crank-Nicolson Method (CNM) offer contrasting approaches to solving the American option pricing problem. This section evaluates their respective strengths and weaknesses in terms of accuracy, convergence, numerical stability, and computational efficiency.

Accuracy

SCM demonstrates superior accuracy, especially for larger parameter configurations (l, m, n), as it leverages Chebyshev interpolation and quadrature rules. Its ability to achieve exponential convergence ensures that the errors reduce significantly with increasing parameters. For smaller parameter sets, however, the relative errors of SCM are higher compared to CNM, potentially due to sensitivities in initialization or interpolation accuracy.

CNM, while accurate for finer spatial and temporal grids, exhibits slower convergence compared to SCM. The method is limited by its second-order accuracy in time and space, which cannot match the exponential convergence rate of SCM.

Convergence

SCM converges more rapidly as the parameter dimensions increase, often achieving high accuracy with fewer iterations for well-tuned configurations. However, the convergence rate can be impacted by the choice of initial boundary estimates $B(\tau)$, as well as hyperparameters like η.

CNM demonstrates linear convergence behavior, with computational time increasing proportionally to the number of grid points and time steps. While reliable, its convergence speed is slower than SCM, especially for larger and more complex problems.

Numerical Stability

SCM exhibits robust numerical stability across a range of parameter values and edge cases. The method maintains stability even for small perturbations in time to maturity (τ) and other inputs, making it well-suited for scenarios requiring high precision.

CNM is conditionally stable and requires small enough time steps (Δt) to maintain accuracy. Larger time steps result in significant deviations, particularly near the early exercise boundary.

Computational Efficiency

SCM is computationally expensive, with costs increasing exponentially for larger parameter configurations. For example, $(l, m, n) = (65, 8, 32)$ results in significantly longer computation times compared to CNM. This is attributed to the iterative schemes and the higher complexity of interpolation and quadrature.

CNM offers superior computational efficiency for simpler configurations for simpler configurations, making it more practical for problems where moderate accuracy suffices. However, for real-time applications requiring high precision, SCM's efficiency can be improved through parallel computing or GPU-based implementations.

Practical Considerations

SCM is better suited for problems where precision and rapid convergence are critical, such as pricing deep-in-the-money or complex financial derivatives. CNM is a viable alternative for simpler scenarios, where computational cost outweighs the need for high precision.

In summary, while both methods have their merits, the Spectral Collocation Method offers distinct advantages in accuracy and convergence, particularly for high-precision applications. However, these benefits come at the cost of increased computational complexity, which can be mitigated through further optimizations. The Crank-Nicolson Method, on the other hand, remains a reliable and computationally efficient choice for basic configurations, albeit with slower convergence and lower accuracy for intricate problems.

6 Limitations and Future Works

Despite the promising results achieved by the Spectral Collocation Method, several limitations were observed during the implementation and testing stages. These limitations, along with potential directions for future improvements, are discussed below.

6.1 Relative Errors

While Our results demonstrated exponential convergence for larger parameter configurations, the relative errors for smaller parameter sets remained consistently larger than those in the reference results. For instance, the error for $(l, m, n) = (5,1,4)$ was 2.75×10^{-2}, which is significantly higher compared to the reference error for similar configurations. These discrepancies may arise from:

- **Numerical Approximations:** Differences in quadrature rules or iterative stopping criteria may have introduced small deviations.
- **Interpolation Sensitivity:** The initial guess for the early exercise boundary $B(\tau)$ and interpolation accuracy significantly impact results for smaller configurations.
- **Adaptive Methods:** Implementing adaptive quadrature techniques or dynamic refinement of collocation points could enhance convergence speed and reduce computational overhead.
- **Improved Initialization:** Developing more accurate methods for initializing $B(\tau)$, such as leveraging machine learning models or more sophisticated approximations, could improve convergence rates.

Future work could focus on refining the quadrature rules and improving interpolation techniques, particularly for small parameter sets, to achieve lower relative errors.

6.2 Computational Efficiency

The computational cost of the Spectral Collocation Method increases significantly for larger parameter configurations. For instance, while $(l, m, n) = (5,1,4)$ required only 2.93×10^{-2} seconds of CPU time, (65,8,32) required 2.03 seconds, highlighting the exponential growth in computational complexity. This presents a trade-off between accuracy and efficiency, which is particularly critical for real-time financial applications.

To address this limitation, future work could explore:

- **Parallel Computing:** Leveraging parallel computing frameworks or GPU-based implementations to reduce computational time.
- **Efficient Algorithms:** Investigating more efficient iterative schemes or reduced-order models to achieve comparable accuracy with lower computational demands.

6.3 Boundary Condition Sensitivity

The accuracy of the method is influenced by the choice of boundary conditions, particularly for configurations involving small p values. Small inaccuracies in boundary conditions can propagate through iterations, affecting the overall solution. This sensitivity could be mitigated by:

- **Robust Boundary Approximations:** Developing more robust methods for approximating boundary conditions, especially for extreme parameter configurations.
- **Error Analysis:** Conducting detailed error sensitivity analysis to understand the propagation of boundary-related errors and adjust the method accordingly.

6.4 Greeks

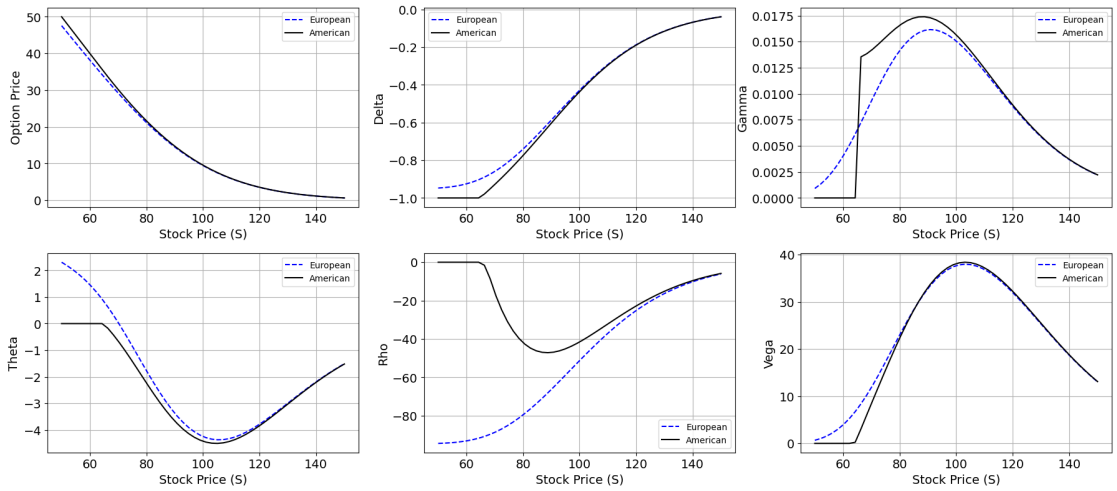

Figure 6.1: Comparison of European and American option prices and their Greeks (Delta, Gamma, Theta, Rho, and Vega) as functions of the stock price (S). The European option is computed using a closed-form solution (blue dashed line), while the American option is calculated using the spectral collocation method (black solid line) with $(l, m, n) = (65, 8, 32)$, $p = 101$ and the rest parameters align with those in Figure 5.1. S ranges from 50 to 150, with 50 evenly spaced data points used for the computations. Changes in each parameter (ϵ) are fixed at 0.01.

To validate the precision of the computed Greeks, option prices obtained through the spectral collocation method were used as the foundation. Finite difference methods are widely used to compute Greeks by discretizing changes in option prices relative to small perturbations in stock price, volatility, or time. The accuracy of Greek calculations depends heavily on the quality of the underlying option prices. Delta and Gamma were computed using central difference formulas, as these provide higher accuracy by averaging the changes in both directions of the perturbation. For Theta, a one-sided difference formula was used due to the unidirectional nature of time to maturity, which only decreases as the option approaches expiration. Rho and Vega, on the other hand, were computed using one-sided difference formulas because the precision offered by this approach was sufficient given the smooth behavior of these parameters under small perturbations.

The region near $S \approx 65$, which marks the critical early exercise boundary, presents unique challenges for Greek calculations. This boundary reflects the transition between exercising and holding the option, leading to abrupt changes in the Greeks. Gamma exhibits a sharp spike, highlighting the heightened sensitivity of the option price to small changes in stock price. Similarly, Theta, Rho and Vega demonstrate pronounced abrupt behaviour in this deep-in-the-money region due to the immediate exercise. Beyond the critical stock price region, the Greeks gradually align with their European counterparts.

6.5 Future Directions

In addition to addressing the aforementioned limitations, future research could explore the following directions:

- **Extension to Multidimensional Problems:** Extending the Spectral Collocation Method to handle multidimensional options, such as basket or Asian options, would broaden its applicability.
- **Stochastic Volatility Models:** Incorporating stochastic volatility or jump-diffusion models into the framework to better capture market dynamics.

- **Hybrid Approaches:** Combining the Spectral Collocation Method with other numerical techniques, such as finite difference or finite element methods, to leverage their respective strengths.
- **Real-Time Applications:** Optimizing the method for real-time option pricing in high-frequency trading environments, where computational efficiency is critical.

Overall, while the Spectral Collocation Method has demonstrated significant potential for pricing American options, further refinements and advancements are needed to fully realize its capabilities in both theoretical and practical applications.

7 Conclusion

Through conducting this project, we draw the conclusions as follows,

- The Spectral Collocation Method demonstrates exceptional accuracy and exponential convergence, particularly for larger parameter configurations. By leveraging advanced techniques such as Chebyshev interpolation, Jacobi-Newton iteration, and numerical quadrature, it effectively addresses the free-boundary problem inherent in American options. The method's ability to accurately approximate the early exercise boundary ensures robust and precise pricing results, making it a valuable tool for financial engineering.
- Despite its accuracy, the Spectral Collocation Method exhibits computational inefficiencies for larger configurations, with significant increases in computation time as parameter dimensions grow. Optimizing the method through adaptive quadrature, parallel computing, or reduced-order models presents a promising direction for making it viable for real-time applications.
- The Crank-Nicolson Method provides a reliable and computationally efficient benchmark for comparison. Its stability and second-order accuracy make it a suitable choice for applications requiring moderate precision. However, it lacks the exponential convergence observed in the Spectral Collocation Method, particularly for configurations with large spatial or temporal grids.
- In comparative analysis, the Spectral Collocation Method outperforms the Crank-Nicolson Method in terms of accuracy and convergence speed, especially when precision is paramount. Conversely, the Crank-Nicolson Method offers an efficient alternative in scenarios where computational resources are constrained or precision requirements are lower.
- Several limitations were identified in the Spectral Collocation Method, including sensitivity to parameter configurations, initial boundary estimates, and computational cost. For small parameter sets, larger relative errors were observed compared to the reference study, suggesting the need for more refined initialization techniques, dynamic collocation strategies, and hyperparameter tuning.
- Future research directions include extending the Spectral Collocation Method to multidimensional problems, such as basket and Asian options, and incorporating stochastic volatility or jump-diffusion models to capture more complex market dynamics. Additionally, hybrid approaches that combine the strengths of different numerical methods, such as finite difference or finite element methods, could offer new opportunities for innovation.
- For real-world applications, especially in high-frequency trading or real-time option pricing, optimizing the Spectral Collocation Method for computational efficiency remains critical. Techniques such as parallel computing or GPU acceleration could address current limitations and expand its usability in practical financial engineering settings.

In conclusion, the Spectral Collocation Method represents a significant advancement in the numerical pricing of American options. Its ability to achieve high accuracy and exponential convergence highlights its potential as a powerful tool for tackling complex financial problems. While challenges related to computational efficiency and sensitivity to parameter configurations remain, future enhancements and optimizations promise to broaden its applicability and make it an integral component of financial modeling and engineering practices.

References

Andersen, L. B., Lake, M., & Offengenden, D. (2016). High-performance american option pricing. *Journal of Computational Finance*, *20*(1), 39–87.

Antdvid. (n.d.). Fast american option pricing [GitHub repository. Accessed: 2024-11-30].

Ballabio, L. (n.d.). Quantlib: American option engine implementation [GitHub repository. Accessed: 2024-11-30].

Helali, H. (n.d.). Fdm american option pricing [Blog post. Accessed: 2024-11-30].

Mawm, J. (n.d.). Mastering python for finance (second edition) [GitHub repository. Accessed: 2024-11-30].

PlasmaControl. (2024). Desc: Plasma equilibrium code [Accessed: 2024-11-30]. https://github.com/PlasmaControl/DESC