MTH 9871 Final Project Report

Xinyi(Krystal) Lin

December 16, 2024

Introduction

In this project, we consider the pricing of an Asian Basket option using Monte Carlo (MC) methods. We have N uncorrelated lognormal assets following Geometric Brownian Motion (GBM), each with a time-dependent volatility structure. Our goal is to estimate the option price and the sensitivities (Greeks) of the payoff with respect to various parameters. We implement two main approaches for computing sensitivities: the finite difference method (FDM) and Adjoint Algorithmic Differentiation (AAD).

Model Setup

We consider N assets $S_i(t)$, $i = 1, \ldots, N$, following:

$$\frac{dS_i(t)}{S_i(t)} = (r - \delta_i)\, dt + \sigma_i(t)\, dW_i(t),$$

where r is the risk-free rate, δ_i is the continuous dividend yield of the i-th asset, and $\sigma_i(t)$ is the (piecewise constant) volatility. Under the risk-neutral measure, the discounted asset prices are martingales.

We price an Asian Basket call option with payoff at T_M:

$$\text{Payoff} = \left(\frac{1}{M} \sum_{m=1}^{M} B(T_m) - K \right)^+ ,$$

where

$$B(T_m) = \frac{1}{N} \sum_{i=1}^{N} \frac{S_i(T_m)}{S_i(T_0)}.$$

The value of the option at time 0 is the discounted expected payoff:

$$\text{Value} = e^{-rT_M} \mathbb{E} \left[\max \left(\frac{1}{M} \sum_{m=1}^{M} B(T_m) - K,\, 0 \right) \right].$$

Monte Carlo Simulation

We discretize the time interval $[0, T_M]$ into M steps. For each simulated path:

1. Generate the increments of the underlying assets under the risk-neutral measure.
2. Compute the normalized asset values $S_i(T_m)/S_i(T_0)$ at each observation time T_m.
3. Form the basket $B(T_m)$ by averaging over the N assets.
4. Compute the average over $m = 1, \ldots, M$ to obtain the Asian basket value.
5. Evaluate the payoff and discount it by e^{-rT_M}.

Repeat this procedure for a large number of paths and estimate the value as the sample mean, with the standard error indicating statistical uncertainty.

Sensitivity Analysis

We aim to compute sensitivities (Greeks) with respect to parameters such as $S_i(0)$, δ_i, and $\sigma_{i,m}$. We consider two methods:

Finite Difference Method (FDM)

FDM approximates derivatives by perturbing a parameter θ by a small ϵ:

$$\frac{\partial V}{\partial \theta} \approx \frac{V(\theta + \epsilon) - V(\theta)}{\epsilon}.$$

This method is conceptually simple but requires multiple re-runs of the MC simulation (one for each parameter perturbation), becoming computationally expensive as the number of parameters grows.

Adjoint Algorithmic Differentiation (AAD)

AAD computes all sensitivities in a single backward pass after the forward simulation. After computing all paths and payoffs, we define adjoint variables and apply the chain rule in reverse. This process yields sensitivities for all parameters from a single simulation, thus greatly reducing computational cost when the number of parameters is large.

Mathematics of the Adjoint Method

We write the increments for each asset:

$$X_{i,m} = (r - \delta_i - \tfrac{1}{2}\sigma_{i,m}^2)\Delta t + \sigma_{i,m}\sqrt{\Delta t}\, Z_{i,m},$$

where $Z_{i,m} \sim N(0,1)$ and $\Delta t = T_M/M$. Then:

$$\frac{S_i(T_m)}{S_i(0)} = \exp\left(\sum_{k=1}^{m} X_{i,k}\right).$$

The basket at time T_m is:

$$B(T_m) = \frac{1}{N}\sum_{i=1}^{N} \exp\left(\sum_{k=1}^{m} X_{i,k}\right),$$

and the average basket:

$$\overline{B} = \frac{1}{M}\sum_{m=1}^{M} B(T_m).$$

The payoff is:

$$\text{Payoff} = \max(\overline{B} - K, 0).$$

Discounting by e^{-rT_M} gives the value at time 0. To apply AAD:

1. Perform a forward sweep computing all $X_{i,m}$, $B(T_m)$, and $\overline{B}$.
2. Start the backward sweep by defining
$$\lambda_{\overline{B}} = \frac{\partial \text{Payoff}}{\partial \overline{B}} = \begin{cases} e^{-rT_M}, & \text{if } \overline{B} > K, \\ 0, & \text{otherwise.} \end{cases}$$
3. Propagate adjoints back through $B(T_m)$:
$$\lambda_{B(T_m)} = \frac{\partial \text{Payoff}}{\partial B(T_m)} = \lambda_{\overline{B}} \frac{1}{M}.$$
4. Further propagate adjoints to $X_{i,m}$:
$$\frac{\partial B(T_m)}{\partial X_{i,k}} = \frac{1}{N} \exp\left(\sum_{j=1}^{m} X_{i,j}\right) \mathbf{1}_{k \le m}.$$
Summing over m:
$$\frac{\partial \text{Payoff}}{\partial X_{i,k}} = \sum_{m=k}^{M} \lambda_{B(T_m)} \frac{\partial B(T_m)}{\partial X_{i,k}}.$$
5. Finally, use the chain rule to obtain:
$$\frac{\partial \text{Payoff}}{\partial \delta_i} = \sum_{k=1}^{M} \frac{\partial \text{Payoff}}{\partial X_{i,k}} \frac{\partial X_{i,k}}{\partial \delta_i}, \qquad \frac{\partial X_{i,k}}{\partial \delta_i} = -\Delta t,$$
and similarly for $\sigma_{i,m}$:
$$\frac{\partial X_{i,m}}{\partial \sigma_{i,m}} = -\sigma_{i,m}\Delta t + \sqrt{\Delta t} Z_{i,m}.$$

All sensitivities w.r.t. δ_i and $\sigma_{i,m}$ are thus obtained with a single backward pass. Sensitivities w.r.t. $S_i(0)$ are zero here due to the normalization $S_i(T)/S_i(0)$.

Results and Discussion

Example with Simulated Inputs

To demonstrate the implementation, we simulate an example with $N = 2$ assets, $M = 4$ observation points, and the following parameters:

- Time to maturity: $T = 1$ year,
- Risk-free rate: $r = 5\%$,
- Strike price: $K = 0.9$,
- Number of Monte Carlo paths: 100000

The simulated inputs for the initial spot prices, dividend yields, and volatility paths are as follows:

```
Simulated Inputs:
Initial Spot Prices (S0):
[116.302148 108.351082]

Dividend Yields:
[0.01424518 0.01844776]

Volatility Paths:
[[0.10047189 0.13380522 0.16713855 0.20047189]
 [0.11215691 0.14549025 0.17882358 0.21215691]]
```

Monte Carlo Simulation Results

The results of the Monte Carlo simulation for the Asian Basket option are summarized in Table 1.

Metric	**Value**
Option Price	0.115979
Standard Error	0.000194

Table 1: Monte Carlo Simulation Results for Asian Basket Option.

Finite Difference Method (FDM) Results

The results of the Finite Difference Method (FDM) for the Asian Basket option are summarized in Tables 2 and 3.

Metric	**Value**
Option Price (FDM)	0.115979

Table 2: FDM Option Price for Asian Basket Option.

Sensitivity	**Asset 1**	**Asset 2**
Initial Spot Prices (S_0)	0.000000	0.000000
Dividend Yields (δ_i)	-0.299136	-0.298286
Volatility (Step 1)	0.005210	0.006972
Volatility (Step 2)	0.003275	0.003502
Volatility (Step 3)	0.002047	0.001827
Volatility (Step 4)	0.000934	0.000194

Table 3: Sensitivities w.r.t. Initial Spot Prices, Dividend Yields, and Volatilities.

Automatic Adjoint Differentiation (AAD) Results

The results of the Automatic Adjoint Differentiation (AAD) method for the Asian Basket option are summarized in Tables 4 and 5.

Metric	**Value**
Option Price (AAD)	0.115979
Standard Error of Price	0.000194

Table 4: AAD Option Price for Asian Basket Option.

Sensitivity	**Asset 1**	**Std. Error (Asset 1)**	**Asset 2**	**Std. Error (Asset 2)**
Initial Spot Prices (S_0)	0.000000	0.000000	0.000000	0.000000
Dividend Yields (δ_i)	-0.299136	0.000180	-0.298286	0.000184
Volatility (Step 1)	0.005210	0.000758	0.006972	0.000759
Volatility (Step 2)	0.003275	0.000576	0.003502	0.000573
Volatility (Step 3)	0.002046	0.000387	0.001827	0.000387
Volatility (Step 4)	0.000934	0.000197	0.000194	0.000196

Table 5: AAD Sensitivities w.r.t. Initial Spot Prices, Dividend Yields, and Volatilities.

Comparison of FDM and AAD Sensitivities

The differences between the sensitivities computed using Finite Difference Method (FDM) and Automatic Adjoint Differentiation (AAD) are summarized below. These differences are within numerical tolerances, confirming the consistency between the two methods.

Sensitivity Type	Asset 1	Asset 2
Initial Spot Prices (S_0)	0.00000000e+00	0.00000000e+00
Dividend Yields (δ_i)	4.06208932e-08	9.38501469e-08

Table 6: Differences Between FDM and AAD Sensitivities for Initial Spot Prices and Dividend Yields.

Step	Asset 1	Asset 2
1	2.69046053e-07	-1.65340122e-07
2	2.13011490e-07	-1.05076423e-07
3	3.48248109e-07	1.32936454e-07
4	-8.77658673e-08	-8.00462081e-08

Table 7: Differences Between FDM and AAD Volatility Sensitivities.

Comparison of Accuracy and Computational Cost

Accuracy: As expected, both FDM and AAD yield similar sensitivity estimates. Since AAD and FDM both converge to the true sensitivities as the number of paths increases, the differences become negligible with sufficient computational effort.

CPU Time Ratios: We record CPU time for both methods over varying M (for fixed $N = 1$) and varying N (for fixed $M = 1$). Finite difference calculations require multiple Monte Carlo runs (one per parameter perturbation), making it expensive when the dimensionality of parameters (N and M) increases. In contrast, AAD computes all sensitivities in a single pass after the forward simulation. The computational overhead for AAD is typically only a small multiple of the single MC run time, making it much more efficient for large N and M. For instance:

- With $N = 1$ and increasing M, FDM CPU Ratio increase linearly, while AAD remains almost constant relative to the number of sensitivities.
- With $M = 1$ and increasing N, FDM CPU Ratio again increase linearly, while AAD remains more cost-effective.

The CPU ratios for Finite Difference Method (FDM) and Automatic Adjoint Differentiation (AAD) were computed as the time required to calculate the price and sensitivities divided by the time required to calculate only the price. The results are shown in Figures 1 and 2.

Conclusion

We have implemented a Monte Carlo simulation to price an Asian Basket option and computed sensitivities using both finite difference and AAD methods. While both methods produce comparable results in terms of accuracy, AAD significantly outperforms FDM as

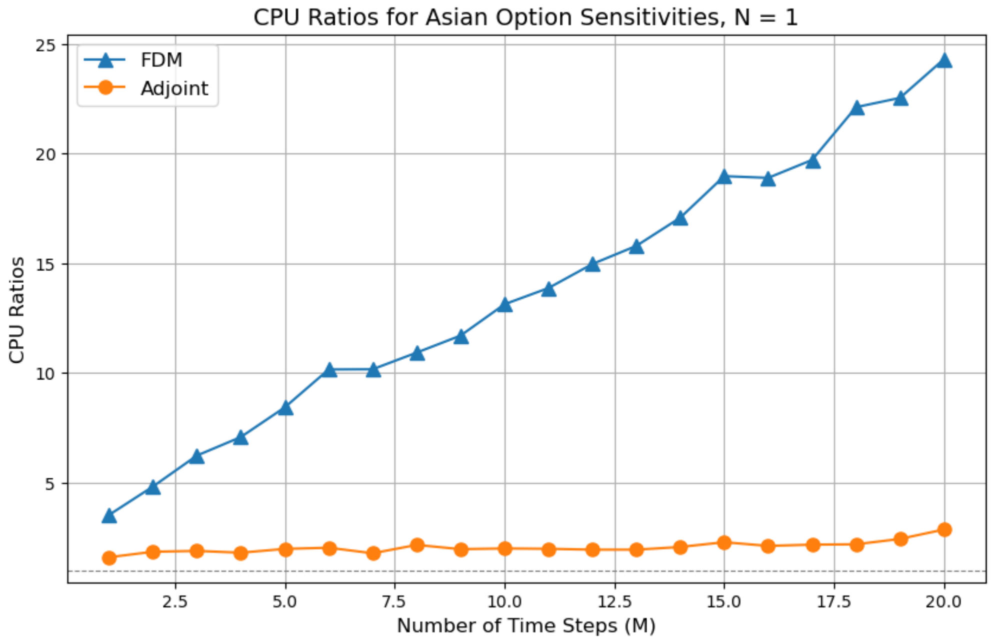

Figure 1: CPU Ratios with Fixed $N = 1$.

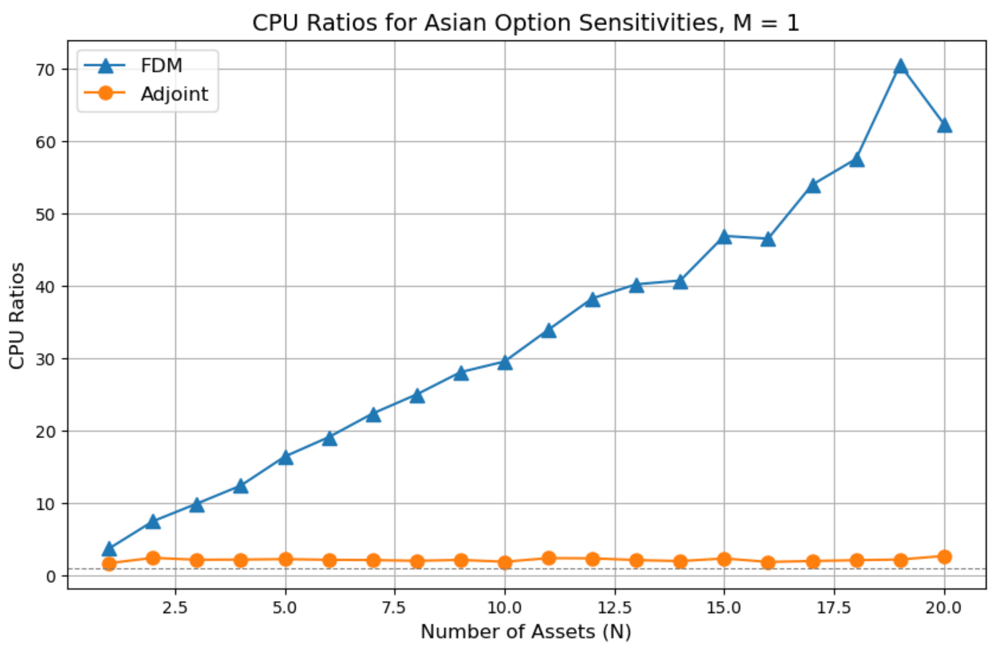

Figure 2: CPU Ratios with Fixed $M = 1$.

the dimensionality of the parameter space grows. The AAD approach provides a more computationally efficient way to obtain a large number of sensitivities simultaneously, making it preferable for high-dimensional or large-scale problems commonly encountered in practice.