On adaptive stratification

Adrien Cortes, Louis-Marie Lovichi

May 10, 2024

Abstract

This project is based on the article by **Pierre Etore et al.** titled "On adaptive stratification". Stratified sampling is a well-known variance reduction method. However, its performance can be significantly improved depending on the choice of space partitions called ***strata***. Good execution of the stratified sampling technique thus depends on both:

- Strata where the function to be integrated is nearly constant.
- Optimal allocation of samples in each of the strata.

This problem is far from trivial when the dimension of the space is large. The authors of the paper therefore propose an algorithm for adapting the strata and the associated allocation. The variance reduction is done at the level of the direction of the strata, and it is the directions of the strata that are updated in this algorithm.

1 Introduction

The objective of this digital project is twofold:

- Implement the adaptive stratification estimator proposed in the article. This involves choosing the number of realizations to take in each stratum adaptively by estimating the variances of each stratum "online".
- Test the robustness of the approach in high dimensions (Gaussian case).

In this report, we will explore the theoretical justifications of the algorithm and conduct a comprehensive numerical study in the case of path-dependent option pricing [EFJM11]. Furthermore, we have had the opportunity to directly engage with the authors of the paper. In 2009, they had implemented the algorithm in Matlab. However, we have made the personal choice to develop the algorithm in **Python**.

2 Stratified Sampling

Introduction to Stratification The goal of stratification is to estimate the expectation $\mathrm{E}[\phi(Y)]$ where $\phi : R^d \to R$ is a measurable function and Y is a random variable in R^d.

Throughout the article, $Y = (Y_1, \ldots, Y_d)$ follows a multivariate normal distribution. We will divide R^d into mutually exclusive *strata* and evaluate our function on each *strata*. The application will be to price high-dimensional options with Gaussian vectors.

To define a stratum, we only need the data of a partition of the space, $\{S_{\mathbf{i}}, \mathbf{i} \in \mathcal{I}\}$ and an orthogonal matrix μ of dimensions $d \times m$, $m \leq d$ allowing us to do the projection.

The space is then divided into strata:

$$\mathrm{S}_{\mu,\mathbf{i}} \stackrel{\text{def}}{=} \left\{x \in R^d, \mu^T x \in \mathrm{S}_{\mathbf{i}}\right\}, \quad \mathbf{i} \in \mathcal{I}$$

likewise, we define by definition $p_{\mathbf{i}}(\mu) \stackrel{\text{def}}{=} P\left(Y \in \mathrm{S}_{\mu,\mathbf{i}}\right) = P\left(\mu^T Y \in \mathrm{S}_{\mathbf{i}}\right)$. We assume, for the sake of simplicity, that $p_{\mathbf{i}}(\mu) > 0$, for all $\mathbf{i} \in \mathcal{I}$. In this paper, we suppose that $p_{\mathbf{i}}(\mu) = \frac{1}{I}$ with $I > 0$.

Let M be the total number of draws that we are going to make and $\mathcal{Q} = \{q_{\mathbf{i}}, \mathbf{i} \in \mathcal{I}\}$ the allocation vector. This allows us to calculate the total number of draws that we are going to make for the i-th stratum,

$$M_{\mathbf{i}} \stackrel{\text{def}}{=} \left\lfloor M \sum_{\mathbf{j} \leq \mathbf{i}} q_{\mathbf{j}} \right\rfloor - \left\lfloor M \sum_{\mathbf{j} < \mathbf{i}} q_{\mathbf{j}} \right\rfloor, \quad \mathbf{i} \in \mathcal{I}$$

where $\lfloor \cdot \rfloor$ denotes the integer part and $\sum_{\emptyset} q_{\mathbf{j}} = 0$ by convention.

We also have:

$$\sigma_{\mathbf{i}}^2(\mu) \stackrel{\text{def}}{=} \mathrm{E}\left[\phi^2(Y) \mid \mu^T Y \in \mathrm{S}_{\mathbf{i}}\right] - \left(\mathrm{E}\left[\phi(Y) \mid \mu^T Y \in \mathrm{S}_{\mathbf{i}}\right]\right)^2$$

and

$$q_{\mathbf{i}}^{\star}(\mu) \stackrel{\text{def}}{=} \frac{p_{\mathbf{i}}(\mu)\sigma_{\mathbf{i}}(\mu)}{\sum_{\mathbf{j} \in \mathcal{I}} p_{\mathbf{j}}(\mu)\sigma_{\mathbf{j}}(\mu)}$$

Given a stratum, an allocation, and a number of draws, we obtain *the stratified estimator of* $\mathrm{E}[\phi(Y)]$:

$$\sum_{\mathbf{i} \in \mathcal{I}: M_{\mathbf{i}} > 0} p_{\mathbf{i}}(\mu) \left\{ \frac{1}{M_{\mathbf{i}}} \sum_{j=1}^{M_{\mathbf{i}}} \phi\left(Y_{\mathbf{i},j}\right) \right\}$$

where the $\{Y_{\mathbf{i},j}, j \leq M_{\mathbf{i}}, \mathbf{i} \in \mathcal{I}\}$ are independent with $Y_{\mathbf{i},j}$ distributed according to the conditional distribution $\mathrm{P}\left[Y \in \cdot \mid \mu^T Y \in \mathrm{S}_{\mathbf{i}}\right]$ for $j \leq M_{\mathbf{i}}$.

Here we have an unbiased estimator of $\mathrm{E}[\phi(Y)]$ if all the $M_{\mathbf{i}}$ are positive.

Practical Calculation of the Stratified Estimator The first part consists of calculating the strata. We fix a tuple $\mathbf{i} = (i_1, \ldots, i_m) \in \{1, \cdots, I\}^m$, a positive integer I, and the dimension d of the multivariate law. Noting G as the cumulative distribution functions of centered reduced Gaussians:

$$\mathrm{S}_{\mathbf{i}} \stackrel{\text{def}}{=} \prod_{k=1}^{m} \left(G_k^{-1}\left(\frac{i_k - 1}{I}\right), G_k^{-1}\left(\frac{i_k}{I}\right) \right]$$

We take $m = 1$. We then have enough elements to describe the initialization of the algorithm:

Algorithm 1 Initialization of the Algorithm

Input: Initial stratification directions $\mu^{(0)}$, total number of draws M
Output: Initial number of draws in each stratum $M_i^{(0)}$, probabilities $p_i^{(0)}$
procedure INITIALIZATION
  Choose initial stratification directions $\mu^{(0)}$
  Define initial number of draws for each stratum $\{M_i^{(0)}\}_{i=1}^m$ such that $\sum_{i=1}^m M_i^{(0)} = M$
  Compute probabilities $p_i^{(0)} = \frac{1}{I}$ for each stratum
  Compute $q_i^*(\mu^{(0)})$ for each stratum using variance calculations:

$$q_i^*(\mu^{(0)}) = \frac{p_i(\mu^{(0)})\sigma_i(\mu^{(0)})}{\sum_{j \in I} p_j(\mu^{(0)})\sigma_j(\mu^{(0)})}$$

  Compute M_i using:

$$M_{\mathbf{i}} \stackrel{\text{def}}{=} \left\lfloor M \sum_{\mathbf{j} \leq \mathbf{i}} q_{\mathbf{j}} \right\rfloor - \left\lfloor M \sum_{\mathbf{j} < \mathbf{i}} q_{\mathbf{j}} \right\rfloor, \quad \mathbf{i} \in \mathcal{I}$$

end procedure

Introduction of a Key Quantity When M is large and I is fixed, the article shows that minimizing the variance of the stratified estimator amounts to minimizing $V(\mu)$ given by:

$$V(\mu) \stackrel{\text{def}}{=} \sum_{\mathbf{i}=1}^{\mathcal{I}} p_{\mathbf{i}}(\mu)\sigma_{\mathbf{i}}(\mu) = \sum_{\mathbf{i}=1}^{\mathcal{I}} \left(\nu_{\mathbf{i}}(f,\mu)\nu_{\mathbf{i}}\left(f\phi^2,\mu\right) - \nu_{\mathbf{i}}^2(f\phi,\mu)\right)^{1/2}$$

where we have $\nu_{\mathbf{i}}(h,\mu) \stackrel{\text{def}}{=} \int_{\mathrm{S}_{\mu,\mathbf{i}}} hd\lambda = \int \prod_{k=1}^{m} 1_{\left\{y, G_k^{-1}((\mathbf{i}_k-1)/I) \leq \langle \mu_k, y\rangle \leq G_k^{-1}(\mathbf{i}_k/I)\right\}} hd\lambda,\ h \in \left\{f, \phi f, \phi^2 f\right\}$ and $\sigma_{\mathbf{i}}^2(\mu) = \frac{\nu_{\mathbf{i}}\left(f\phi^2,\mu\right)}{\nu_{\mathbf{i}}(f,\mu)} - \left(\frac{\nu_{\mathbf{i}}(f\phi,\mu)}{\nu_{\mathbf{i}}(f,\mu)}\right)^2$. $p_{\mathbf{i}}(\mu) = \nu_{\mathbf{i}}(f,\mu)$.

The gradient of $V(\mu)$ has the good taste of being explicitly computable. More precisely, we find:

$$\nabla_\mu V(\mu) = \sum_{\mathbf{i}=1}^{\mathcal{I}} \frac{\nabla_\mu \nu_{\mathbf{i}}(f,\mu)\nu_{\mathbf{i}}\left(f\phi^2,\mu\right) + p_{\mathbf{i}}(\mu)\nabla_\mu \nu_{\mathbf{i}}\left(f\phi^2,\mu\right) - 2\nu_{\mathbf{i}}(f\phi,\mu)\nabla_\mu \nu_{\mathbf{i}}(f\phi,\mu)}{2p_{\mathbf{i}}(\mu)\sigma_{\mathbf{i}}(\mu)} 1_{\{p_{\mathbf{i}}(\mu)\sigma_{\mathbf{i}}(\mu)\neq 0\}}$$

A significant contribution of the algorithm is then to choose the direction of stratification optimally (from the calculation of this gradient). The optimization involves orienting the orthogonal matrix μ of stratification in the following manner:

$$\tilde{\mu} = \mu^{(t)} - \gamma_t \nabla \bar{V}\left(\mu^{(t)}\right)$$

A large part of the work is therefore to compute $\widehat{\nabla V}\left(\mu^{(t)}\right)$.

We can then describe the first part of the algorithm, which precisely consists of calculating this gradient.

Algorithm 2 Iteration process

1: **Iteration**: At iteration $t+1$, given $\mu^{(t)}$, $M^{(t)}$ and $\{p_i(\mu^{(t)})\}, i \in \{1,\ldots,I\}$
2: **Compute** $\nabla V(\mu^{(t)})$:
3: (i) For $i \in \{1,\ldots,I^m\}$, draw $M_i^{(t)}$ realizations of i.i.d. random variables $\{Y_{i,k}^{(t)} | k \leq M_i^{(t)}\}$
4: with distribution $P(Y \mid Y \in S_{\mu^{(t)},i})$ and evaluate $v_i^{(t+1)}(h) = \frac{p_i(\mu^{(t)})}{M_i^{(t)}} \sum_{k=1}^{M_i^{(t)}} h(Y_{i,k}^{(t)})$ for $h \in \{\phi, \phi^2\}$.
5: (ii) For $k \in \{1,\ldots,m\}, s \in \{G_k^{-1}(1/I),\ldots,G_k^{-1}((I-1)/I)\}$, draw $\tilde{M}_{k,s}^{(t)}$ realizations of i.i.d.
6: random variables with distribution $P(Y \mid \mu_k^{(t)T} Y = s)$. Compute a Monte Carlo estimate
7: of $\nabla \nu_i(h,\mu^{(t)})$ for $h \in \{f, f\phi, f\phi^2\}$.
8: (iii) Compute a Monte Carlo estimate of $\nabla V(\mu^{(t)})$.

End of the Algorithm After computing the gradient of $V(\mu)$, we can modify the direction of stratification $\tilde{\mu} = \mu^{(t)} - \gamma_t \nabla \bar{V}\left(\mu^{(t)}\right)$. This necessarily impacts the allocation vector, which we must update.

$$q_{\mathbf{i}}^{(t+1)} = \frac{p_{\mathbf{i}}\left(\mu^{(t)}\right)\hat{\sigma}_{\mathbf{i}}^{(t+1)}}{\sum_{\mathbf{j}\in\{1,\ldots,I\}^m} p_{\mathbf{j}}\left(\mu^{(t)}\right)\hat{\sigma}_{\mathbf{j}}^{(t+1)}}$$

taking into account the deviation on stratum i, $\hat{\sigma}_{\mathbf{i}}^{(t+1)} = \left(\frac{\hat{\nu}_{\mathbf{i}}^{(t+1)}(\phi^2)}{p_{\mathbf{i}}\left(\mu^{(t)}\right)} - \left(\frac{\hat{\nu}_{\mathbf{i}}^{(t+1)}(\phi)}{p_{\mathbf{i}}\left(\mu^{(t)}\right)}\right)^2\right)^{1/2}$. We also necessarily need to update the number of draws we will make per stratum $M_{\mathbf{i}} \stackrel{\text{def}}{=} \left\lfloor M\sum_{\mathbf{j}\leq\mathbf{i}} q_{\mathbf{j}}\right\rfloor - \left\lfloor M\sum_{\mathbf{j}<\mathbf{i}} q_{\mathbf{j}}\right\rfloor$.

Enfin, la dernière partie consiste à calculer un estimateur des quantités que nous calculons. À savoir l'espérance. $\left[\varsigma^2\right]^{(t+1)} = \frac{1}{M}\left(\sum_{\mathbf{i}\in\{1,\cdots,I\}^m} p_{\mathbf{i}}\left(\mu^{(t)}\right)\hat{\sigma}_{\mathbf{i}}^{(t+1)}\right)^2$ To make the algorithm work, we must apply the classic conditions for the convergence of the gradient descent, namely

$$\sum_{t\geq 0}\gamma_t = +\infty, \quad \sum_{t\geq 0}\gamma_t^2 < +\infty$$

Algorithm 3 Update and Compute Steps of the Algorithm

Update the direction of stratification:

Set $\tilde{\mu} = \mu^{(t)} - \gamma_t \nabla V(\mu^{(t)})$; define $\mu^{(t+1)}$ using SVD of $\tilde{\mu}$, keeping the m left singular vectors.

Update the allocation policy:

(i) Compute an estimate $\hat{\sigma}_i^{(t+1)}$ of the standard deviation within stratum i:

$$\hat{\sigma}_i^{(t+1)} = \left(\frac{\nu_i^{(t+1)}(\phi^2)}{\pi_i(\mu^{(t)})} - \left(\frac{\nu_i^{(t+1)}(\phi)}{\pi_i(\mu^{(t)})} \right)^2 \right)^{1/2}$$

(ii) Update the allocation vector $q_i^{(t+1)}$ and number of draws $M_i^{(t+1)}$:

$$q_i^{(t+1)} = \frac{\pi_i(\mu^{(t)})\hat{\sigma}_i^{(t+1)}}{\sum_{j=1}^{I} \pi_j(\mu^{(t)})\hat{\sigma}_j^{(t+1)}}$$

$$M_i^{(t+1)} = \left\lfloor M \sum_{j \leq i} q_j^{(t+1)} \right\rfloor - \left\lfloor M \sum_{j < i} q_j^{(t+1)} \right\rfloor$$

Update probabilities $\pi_i(\mu^{(t+1)})$ for each stratum.

Compute an averaged stratified estimate:

Estimate the Monte Carlo variance $s^2(t+1)$ and current fit $\epsilon^{(t+1)}$ of the stratified estimator:

$$s^2(t+1) = \frac{1}{M} \left(\sum_{i=1}^{I} \pi_i(\mu^{(t)})\hat{\sigma}_i^{(t+1)} \right)^2$$

$$\epsilon^{(t+1)} = \left(\sum_{\tau=1}^{t+1} \frac{1}{s^2(\tau)} \right)^{-1} \sum_{\tau=1}^{t+1} \frac{\nu_i^{(\tau)}(\phi)}{s^2(\tau)}$$

Before giving a complete description of the algorithm, we would like to address a potential point of difficulty, namely the calculation of $\mathrm{P}\left[Y \in \cdot \mid \mu^T Y \in \mathrm{S}_{\mathbf{i}}\right]$.

Generation of Data Conditioned on a Specific Value Suppose we want to generate a variable $\xi \sim \mathcal{N}(\mu, \Sigma)$ in R^d and we want to stratify ξ along a certain vector $v \in R^d$. If the matrix Σ of dimension $d \times d$ is full rank, we can reduce the problem to stratifying ξ along $v^T \xi = 1$. This means we are looking to generate ξ such that $X = v^T \xi$ is a standard normal variable $\mathcal{N}(0, 1)$. [Gla04].

To generate ξ conditionally on X, we start by noting that ξ and X are jointly normal:

$$\begin{pmatrix} \xi \\ X \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \Sigma & v\Sigma \\ v^T\Sigma & v^T\Sigma v \end{pmatrix}\right).$$

Using the conditioning formula for multivariate normal distributions, we find that the conditional distribution of ξ given $X = x$ is:

$$\xi | X = x \sim \mathcal{N}(\Sigma v x, \Sigma - \Sigma v v^T \Sigma).$$

It is crucial to note that the conditional covariance matrix does not depend on x, which implies that only one factorization, for example via Cholesky decomposition, is necessary to sample from this conditional distribution.

The following algorithm allows generating K samples of $\mathcal{N}(0, \Sigma)$ stratified along v:

Algorithm 4 Generation of K Stratified Samples

```
for i = 1, ..., K do
    Generate U ~ Uniform(0, 1)
    V ← (i − 1 + U)/K
    X ← Φ^{-1}(V)                        ▷ Φ^{-1} is the quantile function of the standard normal
    Generate Z ~ N(0, I_d)               ▷ I_d is the identity matrix in dimension d
    ξ ← ΣvX + (A − Σvv^T A)Z             ▷ A such that AA^T = Σ
end for
```

3 Applications

Asian Option We suppose the asset price to follow a classical Black-Scholes dynamic. The asset price is discretized on a regular grid $0 = t_0 < t_1 < \cdots < t_d = T$, with $t_i \overset{\text{def}}{=} iT/d$. The increment of the Brownian motion on $[t_{i-1}, t_i)$ is simulated as $\sqrt{T/d} Y_i$ for $i \in \{1, \cdots, d\}$ where $Y = (Y_1, \cdots, Y_d) \sim \mathcal{N}_d(0, \text{ Id })$. The discounted payoff of a discretely monitored arithmetic average Asian option with strike price K is given by $\Xi(Y)$,

$$\Xi(y) = \exp(-rT)\left(\frac{s_0}{d}\sum_{k=1}^{d}\exp\left(\left(r - 0.5v^2\right)\frac{kT}{d} + v\sqrt{\frac{T}{d}}\sum_{j=1}^{k} y_j\right) - K\right)_+ \quad , \quad y = (y_1, \cdots, y_d) \in R^d$$

We found the following graph for Asian options. We observed that the algorithm is able to decrease the variance.

We also propose a new configuration for the Asian options.

Barrier Options A knock-out barrier option is a path-dependent option that expires worthless if the underlying reaches a specified barrier level. The payoff of this option is given by

$$\Xi(y) = \exp(-rT)\left(\frac{s_0}{d}\sum_{k=1}^{d}\exp\left(\left(r - 0.5\sigma^2\right)\frac{kT}{d} + \sigma\sqrt{\frac{T}{d}}\sum_{j=1}^{k} y_j\right) - K\right)_+ 1_{\{S_T(y) \leq B\}},$$

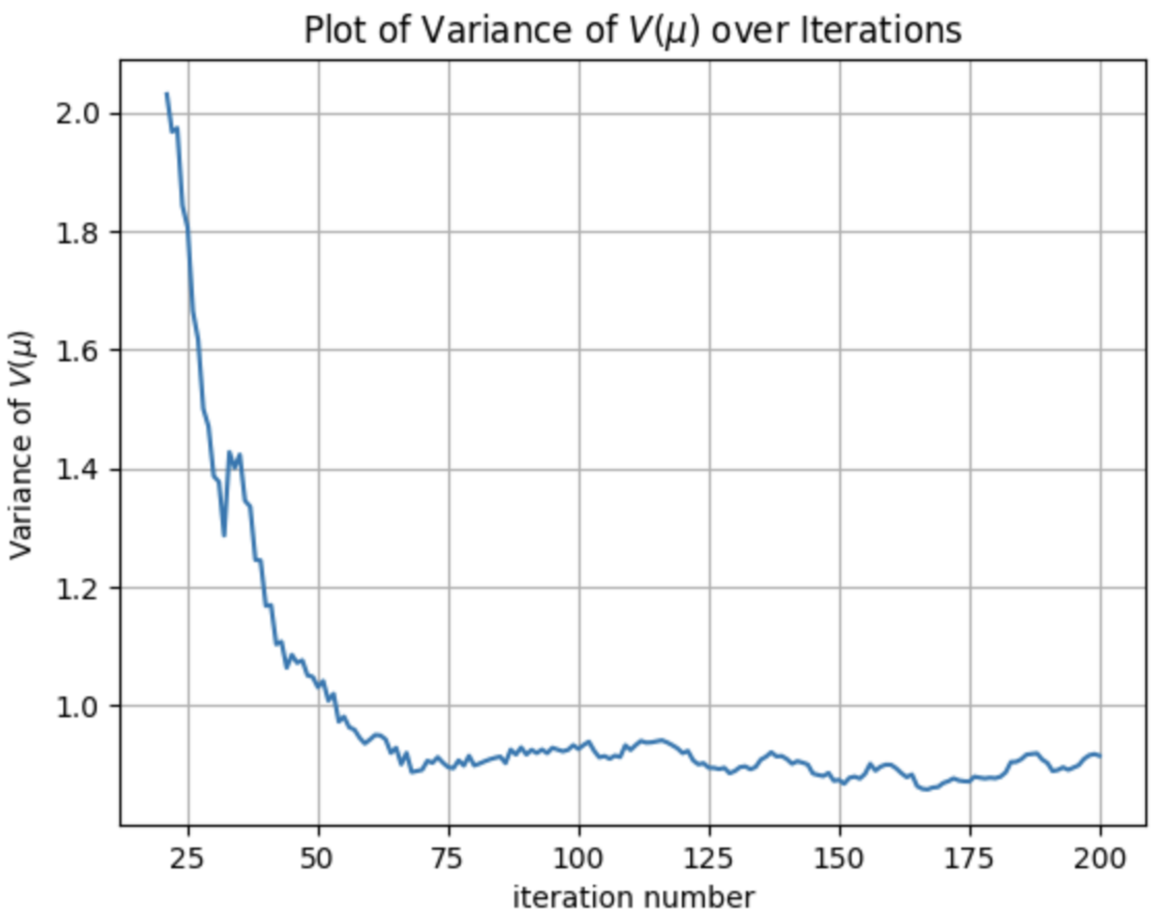

Figure 1: Asian Option, first set of parameters.

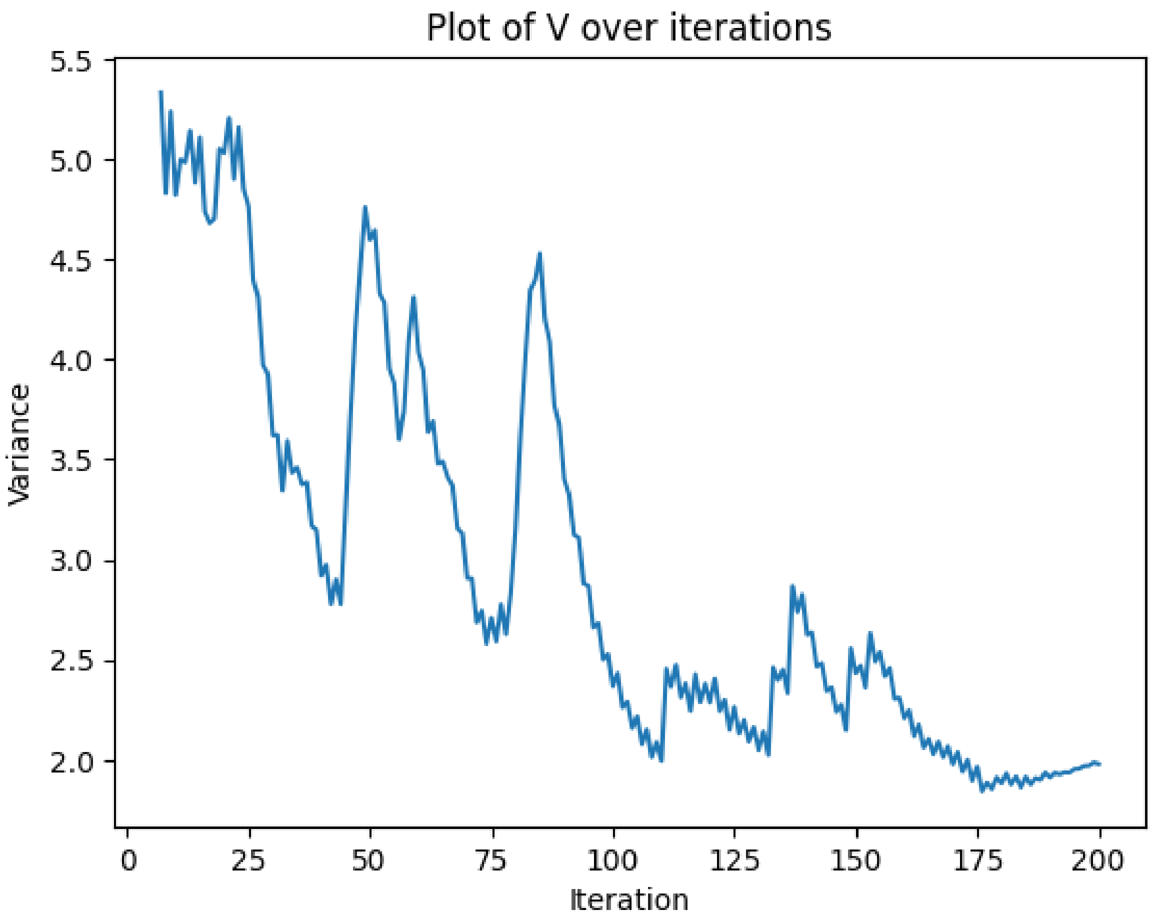

Figure 2: For this new comparison of Asian Options- instability observed.

Parameter	**Value**
d	16
T	1
r	0.05
S_0	50
K	45
v	0.1
I	100
M	20000
N	200

Table 1: Parameters Used for Asian Option

Parameter	**Value**
d	16
T	1
r	0.02
S_0	50
K	40
v	0.1
I	100
M	20000
N	200

Table 2: New parameters Used for Asian Option (for comparison)

where K is the strike price, B is the barrier and $S_T(y)$ is the underlier price modeled as

$$S_T(y) = s_0 \exp\left(\left(r - 0.5\sigma^2\right)T + \sigma\sqrt{\frac{T}{d}}\sum_{j=1}^{d} y_j\right)$$

Parameter	Value
d	16
T	1
r	0.05
S_0	50
K	45
B	60
v	0.1

Table 3: Parameters for Barrier Option

Basket options We consider a basket option composed of $d = 40$ different assets. The dynamics of each asset's price $S_t^{(k)}$ under the risk-neutral probability measure are modeled by a geometric Brownian motion:

$$\frac{dS_t^{(k)}}{S_t^{(k)}} = r dt + v_k dW_t^{(k)}, \quad k = 1, \dots, d$$

where:

- $r = 0.05$ is the risk-free rate.
- v_k is the volatility of the k-th asset, which linearly increases from 0.1 to 0.4 across the assets.

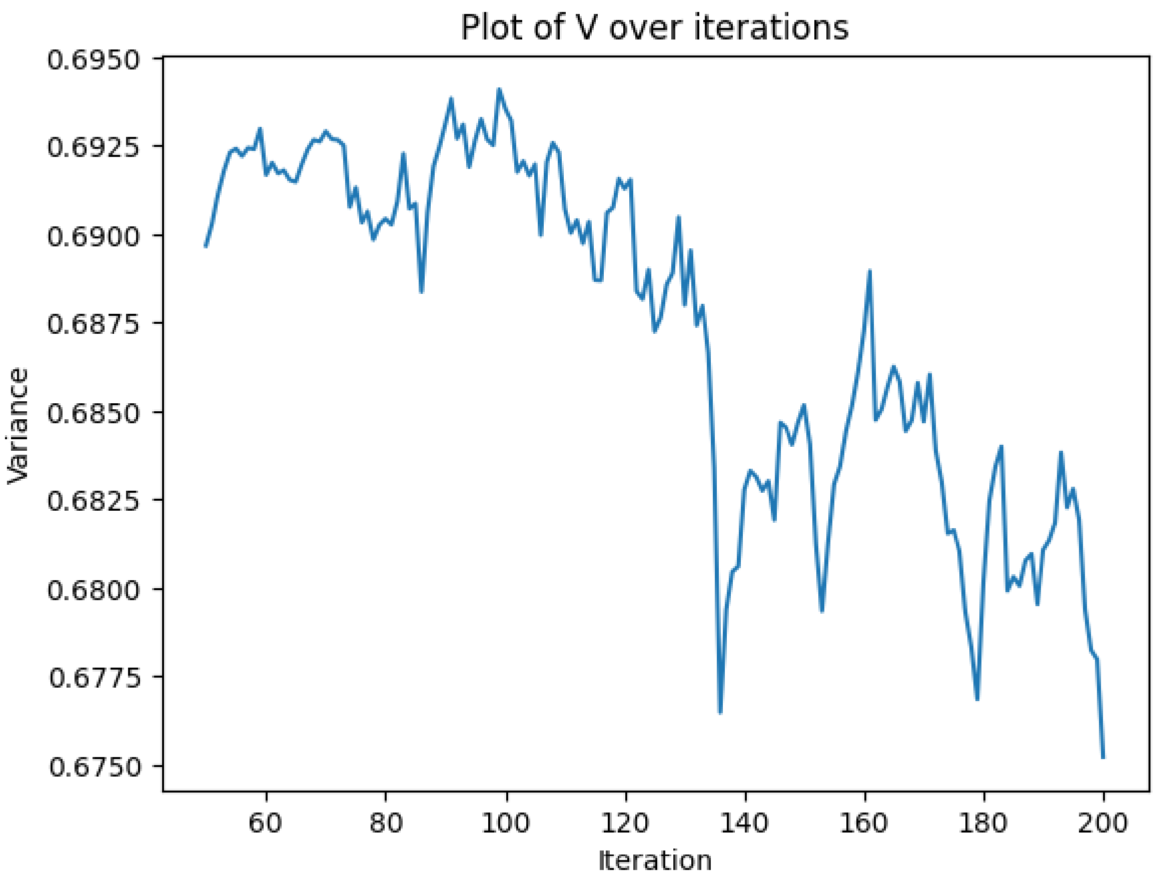

Figure 3: Barrier Option

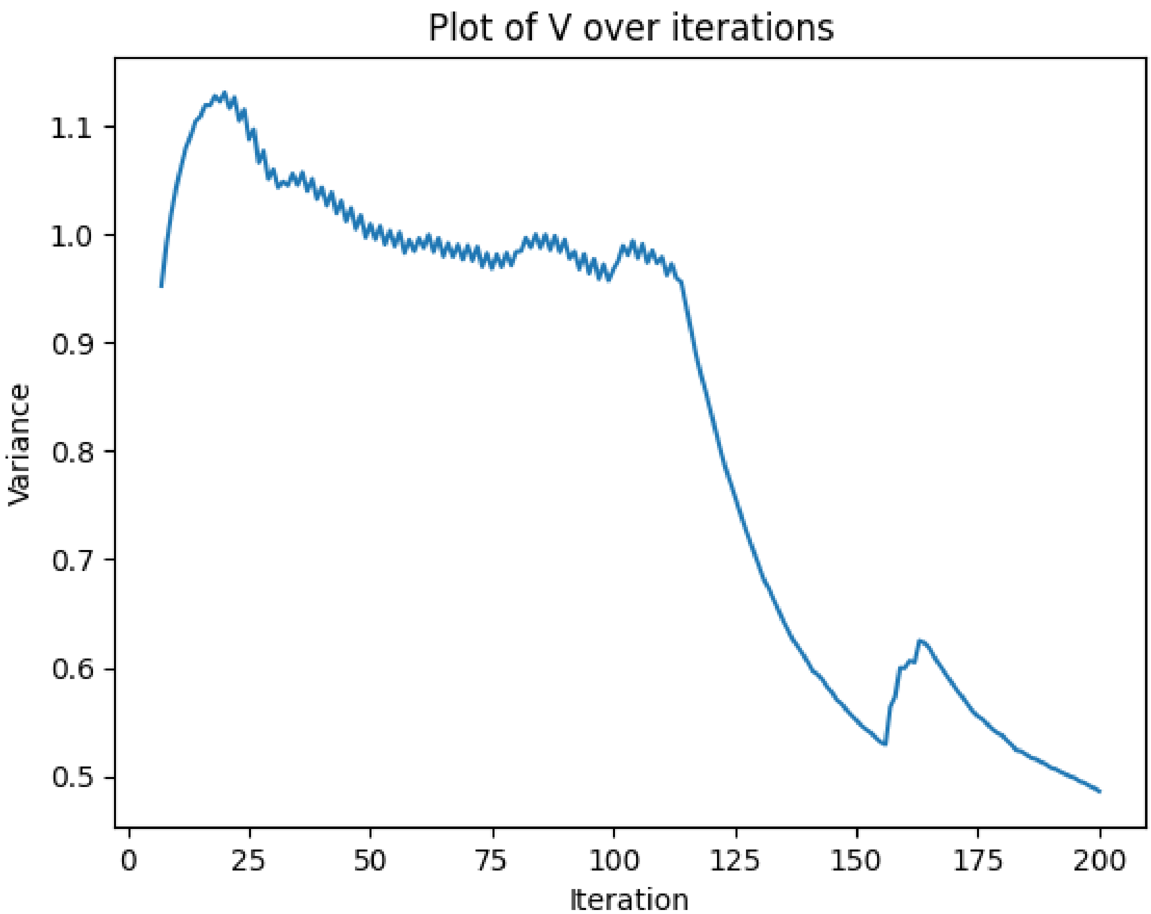

Figure 4: Basket Option

- $W_t^{(k)}$ are possibly correlated Brownian motions.

The correlation structure among the asset prices is given by the covariance matrix Σ, defined as:

$$\Sigma = c \cdot \mathbf{1}_{d\times d} + (1-c) \cdot \mathbf{I}_d$$

where $c = 0.1$, $\mathbf{1}_{d\times d}$ is a matrix of all ones, and $\mathbf{I}_d$ is the identity matrix. This setup implies a base correlation of 0.1 among all pairs of different assets, with a variance of 1 for each asset.

- Initial stock prices $S_0[i]$ are drawn from a uniform distribution between 20 and 80.
- The proportion of each asset in the basket, $\alpha[i]$, is equal, with $\alpha[i] = \frac{1}{d}$.
- The strike price of the basket option, K, is set at 45.

The price at time 0 of a European call option on this basket, with strike price K and maturity $T = 1$ year, is calculated using the expected payoff under the risk-neutral measure, discounted at the risk-free rate.

The price at time 0 of a European call option with strike price K and exercise time T is given by $\mathrm{E}[\Xi(Y)]$ where

$$\Xi(y) = \exp(-rT) \left(\sum_{k=1}^{d} \alpha_k s_0^{(k)} \exp\left(\left(r - 0.5v_k^2\right) T + v_k \sqrt{T} \tilde{y}_k \right) - K \right)_+$$

where $(x)_+ = \max(x, 0)$ denotes the positive part of x.

Parameter	**Description/Value**
d	40
T	1
r	0.05
S_0	`np.random.uniform(20, 80, d)` (Initial stock prices)
K	45
α	$\frac{\texttt{np.ones}(d)}{d}$ (Weight vector)
c	0.1
σ	$c \cdot \texttt{np.ones}((d,d)) + (1-c) \cdot \texttt{np.eye}(d)$ (Covariance matrix)
v	`np.linspace`(0.1, 0.4, d) (Volatility vector)

Table 4: Parameters for a Complex Barrier Option Model

4 Remarks

We tried using the application cases mentioned in the article, and we have a few observations. Even though we followed the steps described in the paper, the decrease in variance isn't as significant. Specifically, we observe some instabilities, which could be due to either a too large gradient descent step or an inaccurate estimation of the ν gradient. Additionally, we couldn't replicate the variance reduction factors described in the paper (we're more in the range of a factor of 2 to 3).

5 Conclusion

In conclusion, our group project has explored and implemented the principles described in the paper regarding the use of stratified sampling as a variance reduction technique for approximating integrals over large dimensional spaces. Through this project, we have gained insights into the role of selecting appropriate partitions (strata) and the allocation of sample sizes within these strata to optimize the estimator's efficiency. Our implementation focused on adapting the stratification and sample allocation dynamically, which proved to be a effective method for managing complex, high-dimensional integrations typical in financial models.

The application of this adaptive stratified sampling technique to the pricing of path-dependent options, particularly in models with both constant and stochastic volatilities, demonstrated a variance reduction—sometimes by factors exceeding 2—compared to traditional Monte Carlo estimators. This substantial improvement underscores the method's theoretical robustness, as discussed in the academic paper, but also highlights its practical relevance and in real-world financial modeling.

This project encourages further exploration and application of adaptive stratification techniques in other complex quantitative fields.

References

[EFJM11] Pierre Etore, Gersende Fort, Benjamin Jourdain, and Eric Moulines. On adaptive stratification. *Annals of operations research*, 189:3, 2011.

[Gla04] Paul Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer, 2004.