

American Option Pricing via Multilevel Monte Carlo

Incorporating Mean Field Feedback

Dominic Marchese

June 2, 2025

1 Prologue

I started this project while taking a course in Monte Carlo Methods (credit to Dr. Jeffery LaComb for teaching in such an engaging way) and just kept expanding it as an exercise in writing. Much of the mathematics I use is just thrown in with maybe a day or two of reading and a few exercises, meaning that much of it may seem very out of place.

Disregard the discussion of the numerical solver for the PDE system, while they are accurate (too my knowledge), my knowledge of numeric solvers is a lot better know so I will be redoing all of it.

2 Theoretic Foundations

American Options are a type of option in which the option holder can exercise the option at any time between the date of purchase and the expiration. Thus, this asset lends itself to a optimal control type of problem. In solving this kind of problem there are numerous tools that can be employed in estimating optimal stopping conditions. As a result of my own interests, I decided to use the mathematical tools I will discuss in this section.

2.1 Stochastic Asset Dynamics

In the framework I've decided upon, we model the joint evolution of d underlying assets whose prices are influenced by random factors and have some underlying correlations. The modeling of asset price paths is done under the risk-neutral measure \mathbb{Q} , in order to maintain consistency with a no arbitrage condition. Under this measure, the discounted asset prices are martingales.

Each asset price $S_t^{(i)}$, for $i = 1, 2, \dots, d$, is assumed to follow correlated Geometric Brownian Motion (GBM). I used Geometric Brownian Motion for the following reasons; It is easy to work with, computational inexpensive to simulate, the process remain strictly positive (meaning price paths remain positive), and returns are log-normally distributed (a property which makes analysis far easier).

Mathematically, the continuous time dynamics of the i -th asset price path under the measure \mathbb{Q} can be expressed with the following stochastic differential equations:

$$dS_t^{(i)} = rS_t^{(i)} dt + \sigma S_t^{(i)} dW_t^{(i)}, \quad i = 1, \dots, d,$$

where:

- r is the constant risk-free interest rate, representing the deterministic drift term under the risk-neutral measure \mathbb{Q} . Intuitively, the asset must grow at least as fast as interest would accrue in a bank account.
- $\sigma > 0$ is the constant volatility parameter, which I decided to make uniform across all assets for simplicity, which which represents the deviations of asset price pathes from the constant growth test r .
- $\{W_t^{(i)}\}_{i=1}^d$ is a d -dimensional vector of correlated standard Brownian Motions adapted to the filtration $\{\mathcal{F}_t\}$ (given by the Multilevel Monte Carlo).

The Brownian motions $W_t^{(i)}$ capture the continuous stochastic shocks driving uncertainty in asset prices. Unlike independent Brownian motions, here each realization of the Brownian Motion is correlated, reflecting

realistic market movements, which often tend to move together. The instantaneous covariance structure of the random Brownian motion increments is characterized by the correlation matrix $\boldsymbol{\rho} = [\rho_{ij}]$, a symmetric and positive definite matrix with unit diagonal entries (required for Cholesky decomposition):

$$\mathbb{E}[dW_t^{(i)} dW_t^{(j)}] = \rho_{ij} dt, \quad \text{with } \rho_{ii} = 1 \quad \forall i, \quad (1)$$

where $\rho_{ij} \in [-1, 1]$ denotes the instantaneous Pearson correlation coefficient between the i -th and j -th asset Brownian increments.

Solution to the SDE system:

Using Itô's lemma and standard results for GBM, the explicit solution to the SDE (2.1) can be written as:

$$S_t^{(i)} = S_0^{(i)} \exp \left\{ \left(r - \frac{\sigma^2}{2} \right) t + \sigma W_t^{(i)} \right\}, \quad i = 1, \dots, d, \quad (2)$$

where $S_0^{(i)}$ is the initial known price at time zero. The term $-\frac{\sigma^2}{2}t$ accounts for the convexity adjustment due to Itô's correction and ensures that the process is a martingale under \mathbb{Q} .

Because the Brownian motions are correlated, the joint distribution of $\mathbf{W}_t = (W_t^{(1)}, \dots, W_t^{(d)})^\top$ at any fixed t is multivariate normal:

$$\mathbf{W}_t \sim \mathcal{N}(\mathbf{0}, t\boldsymbol{\rho}),$$

which implies that the vector of log-returns $\ln(S_t^{(i)}/S_0^{(i)})$ is also jointly normally distributed with mean vector $(r - \frac{\sigma^2}{2})t$ and covariance matrix $\sigma^2 t \boldsymbol{\rho}$.

Basket price definition:

In multi-asset pricing, it is usually the case that payoffs depend on a basket of underlying assets rather than on an individual asset. A standard definition of a basket price B_t is the average of the component asset prices:

$$B_t = \frac{1}{d} \sum_{i=1}^d S_t^{(i)}. \quad (3)$$

This aggregation captures the collective behavior of the portfolio of assets and will serve as our definition of our basket price going forward.

Additional considerations:

- The uniform volatility assumption σ simplifies the model but can be relaxed to asset-specific volatilities σ_i , leading to the general form:

$$dS_t^{(i)} = rS_t^{(i)} dt + \sigma_i S_t^{(i)} dW_t^{(i)}.$$

- The correlation matrix $\boldsymbol{\rho}$ must be positive semi-definite to ensure the existence of a valid multivariate Brownian motion. Numerically, a Cholesky decomposition $\boldsymbol{\rho} = LL^\top$ is used to simulate correlated Brownian increments as:

$$\mathbf{W}_t = L\mathbf{Z}_t,$$

where \mathbf{Z}_t is a vector of independent standard Brownian motions.

- I've assumed both continuous trading and frictionless markets, which are idealizations.
- Extensions of this framework may include stochastic volatility, jump processes, or local volatility models, but the correlated GBM seems to be an excellent starting point.

2.2 Pricing and Optimal Stopping

An American basket option grants the holder the right to exercise the option at any chosen stopping time τ within the allowable time horizon $[0, T]$. Formally, the set of admissible exercise times is the collection of all \mathcal{F}_t -adapted stopping times:

$$\mathcal{T} = \{\tau : \tau \text{ is an } \{\mathcal{F}_t\}\text{-stopping time with } 0 \leq \tau \leq T\}.$$

This is what distinguishes American options from European options, which can be exercised only at maturity. Consequently, the valuation of American options involves solving an optimal stopping problem. The value of the American basket put option at the initial time $t = 0$ is expressed as:

$$V_0 = \sup_{\tau \in \mathcal{T}} \mathbb{E}^{\mathbb{Q}} [e^{-r\tau} (K - B_\tau)^+], \quad (4)$$

where:

- K is the fixed strike price of the option.
- B_τ is the basket value at stopping time τ , defined as in equation (3).
- $(x)^+ = \max(x, 0)$ denotes the positive part function, representing the option payoff if exercised at time τ .
- r is the constant risk-free interest rate used for discounting future payoffs back to present value.
- The expectation $\mathbb{E}^{\mathbb{Q}}[\cdot]$ is taken under the risk-neutral measure \mathbb{Q} .

The supremum over stopping times τ reflects the holder's optimization problem: to select the exercise time that maximizes the expected discounted payoff. The fundamental problem here is that exercising early yields immediate value, but waiting may yield a higher payoff from potential future movements in the basket price.

Mathematically, the American option pricing problem can be formulated as an optimal stopping problem for the Markov process $\{B_t\}_{t \in [0, T]}$. Denote the value function as:

$$V(t, B_t) = \sup_{\tau \in \mathcal{T}_{t, T}} \mathbb{E}^{\mathbb{Q}} [e^{-r(\tau-t)} (K - B_\tau)^+ | \mathcal{F}_t],$$

where $\mathcal{T}_{t, T}$ is the set of stopping times taking values in $[t, T]$. The dynamic programming principle (DPP) states that for any $s \in [t, T]$:

$$V(t, B_t) = \max \left\{ (K - B_t)^+, \mathbb{E}^{\mathbb{Q}} [e^{-r(s-t)} V(s, B_s) | \mathcal{F}_t] \right\}.$$

This expresses that, at time t , the holder chooses between immediate exercise, with payoff $(K - B_t)^+$, or continuation, which is the discounted expected value of holding the option until, at least, time s .

Under suitable regularity conditions, the value function $V(t, x)$ satisfies the following inequality, which characterizes the free boundary problem:

$$\max \left\{ \frac{\partial V}{\partial t} + \mathcal{L}V - rV, (K - x)^+ - V \right\} = 0, \quad (t, x) \in [0, T] \times \mathbb{R}^+, \quad (5)$$

with terminal condition

$$V(T, x) = (K - x)^+.$$

Here, \mathcal{L} is the infinitesimal generator associated with the underlying basket price process, B_t . For example, if the basket dynamics were approximated by a diffusion process with drift $\mu(t, x)$ and volatility $\eta(t, x)$, then \mathcal{L} acts on functions f as:

$$\mathcal{L}f(t, x) = \mu(t, x) \frac{\partial f}{\partial x} + \frac{1}{2} \eta^2(t, x) \frac{\partial^2 f}{\partial x^2}.$$

Consequently, the value function V must be above the payoff, and the option is exercised optimally when the value function and payoff are equivalent (call this the exercise region).

We assume that the exact solution is not known, thus we must use numerical methods. Common methods such as finite difference methods, least squares Monte Carlo, and dynamic programming algorithms can be used to approximate the value function and the optimal exercise strategy. The optimal stopping time τ^* is characterized by the hitting time of the stopping region:

$$\tau^* = \inf\{t \geq 0 : V(t, B_t) = (K - B_t)^+\}.$$

The above formulation describes the fundamental complexity of American option pricing; Balancing the immediate exercise payoff and the value of waiting subject to the uncertainty of the underlying basket.

2.3 Regression-Based Least Squares Monte Carlo (LSMC)

The Longstaff-Schwartz algorithm is a Monte Carlo method for numerically solving optimal stopping problems, in high-dimensional settings where the usual finite difference methods become computationally expensive or unreasonable. The method approximates the continuation value at discrete exercise dates using regression techniques on simulated paths, to estimate the optimal exercise policy.

Discretizing the time interval $[0, T]$ into M equally spaced points $t_m = m\Delta t$, $m = 0, 1, \dots, M$, where $\Delta t = T/M$ is the uniform timestep. At each exercise date t_m , the option value along each simulated path i is denoted by $V_{t_m}^{(i)}$. Applying backward induction, yielding the following recursive formula:

$$V_{t_m}^{(i)} = \max \left((K - B_{t_m}^{(i)})^+, \mathbb{E}^{\mathbb{Q}} \left[e^{-r\Delta t} V_{t_{m+1}}^{(i)} \mid \mathcal{F}_{t_m} \right] \right), \quad (6)$$

where the first term corresponds to immediate exercise value and the second term corresponds to the continuation value — the expected discounted future payoff if the option is not exercised at t_m .

Direct evaluation of the conditional expectation

$$C(t_m, B_{t_m}^{(i)}) = \mathbb{E}^{\mathbb{Q}} [e^{-r\Delta t} V_{t_{m+1}} \mid B_{t_m} = x]$$

is generally not reasonable to calculate, especially in higher dimensions. The key aspect of the Longstaff-Schwartz method is to approximate this continuation value by regressing the realized discounted payoffs $e^{-r\Delta t} V_{t_{m+1}}^{(i)}$ on some set of basis functions evaluated on the current price path at the current time step, $B_{t_m}^{(i)}$.

Formally, let $\{\phi_k(x)\}_{k=0}^K$ denote a family of basis functions; In this case I used Laguerre polynomials. The continuation value is approximated as

$$C(t_m, x) \approx \sum_{k=0}^K \beta_k^{(m)} \phi_k(x), \quad (7)$$

where the regression coefficients $\beta^{(m)} = (\beta_0^{(m)}, \dots, \beta_K^{(m)})^\top$ are approximated by ordinary least-squares regression over all simulated paths, that are in-the-money, at time t_m :

$$\beta^{(m)} = \arg \min_{\beta \in \mathbb{R}^{K+1}} \sum_{i \in \mathcal{I}_m} \left| e^{-r\Delta t} V_{t_{m+1}}^{(i)} - \sum_{k=0}^K \beta_k \phi_k(B_{t_m}^{(i)}) \right|^2, \quad (8)$$

where $\mathcal{I}_m = \{i : (K - B_{t_m}^{(i)})^+ > 0\}$ denotes the indices of paths that are in-the-money at exercise date t_m , since out-of-the-money paths have zero immediate payoff and will not be exercised.

As mentioned, a common choice for basis functions ϕ_k are Laguerre polynomials $\{L_k(x)\}$, which form an orthogonal polynomial system on the positive real number line, $[0, \infty)$ with respect to the weight e^{-x} . I decided on these both because of their common use (I saw them in a number of places), and the desirable properties they have, like orthogonality and numerical stability properties which makes them well-suited for my regression approach. For example, the first few Laguerre polynomials are:

$$L_0(x) = 1, \quad L_1(x) = 1 - x, \quad L_2(x) = 1 - 2x + \frac{x^2}{2}, \quad \dots$$

So, the conditional expectation at time t_m can be approximated with:

$$\mathbb{E}^{\mathbb{Q}} [V_{t_{m+1}} \mid B_{t_m} = x] \approx \sum_{k=0}^K \beta_k^{(m)} L_k(x). \quad (9)$$

To improve regression accuracy and stability, the basket values $B_{t_m}^{(i)}$ can be normalized or transformed before evaluating the basis functions, a method I've considered is called, Tikhonov regularization. The algorithm proceeds backwards in time, from maturity time $t_M = T$, where $V_{t_M}^{(i)} = (K - B_{t_M}^{(i)})^+$, and uses regression at each exercise date to estimate continuation values and determine the optimal exercise policy pathwise by comparing immediate exercise payoffs to continuation values.

The output of the Longstaff-Schwartz algorithm includes the estimated option price at $t = 0$, computed as the average discounted payoff across all simulated paths when following the optimal exercise strategy determined by the regression-based continuation values.

2.4 Multilevel Monte Carlo (MLMC)

The Multilevel Monte Carlo (MLMC) method, is a variance reduction technique that significantly accelerates convergence of Monte Carlo simulations for estimating expectations of stochastic processes. Consider a sequence of increasingly refined time discretizations indexed by the level $\ell = 0, 1, \dots, L$, where the number of time steps at level ℓ is $N_\ell^{\text{steps}} = 2^{\ell+1}$. Let P_ℓ denote the approximate payoff estimator computed using the discretization at level ℓ .

The MLMC estimator exploits the telescoping sum decomposition:

$$\mathbb{E}[P_L] = \mathbb{E}[P_0] + \sum_{\ell=1}^L \mathbb{E}[P_\ell - P_{\ell-1}], \quad (10)$$

where; $\mathbb{E}[P_L]$ is the expectation of the payoff at the finest discretization level L , which closely approximates the true continuous-time quantity of interest, $\mathbb{E}[P_0]$ is the expectation at the coarsest level (level 0) with fewest time steps, and the sum of expectations of differences $\mathbb{E}[P_\ell - P_{\ell-1}]$ corrects the bias introduced by coarse discretizations, using the fact that finer and coarser approximations are correlated.

The key insight is that each difference $P_\ell - P_{\ell-1}$ can be estimated by simulating paired sample paths driven by the same underlying Brownian increments (using the same random numbers) so that the variance of their difference is substantially reduced compared to estimating P_ℓ or $P_{\ell-1}$ individually.

For each level ℓ , an independent Monte Carlo estimator is constructed using N_ℓ samples:

$$\hat{Y}_\ell = \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \left(P_\ell^{(i)} - P_{\ell-1}^{(i)} \right), \quad (11)$$

where we define $P_{-1} = 0$, at the coarsest level $\ell = 0$. The overall MLMC estimator is then the sum over levels:

$$\hat{P}_{\text{MLMC}} = \sum_{\ell=0}^L \hat{Y}_\ell = \frac{1}{N_0} \sum_{i=1}^{N_0} P_0^{(i)} + \sum_{\ell=1}^L \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \left(P_\ell^{(i)} - P_{\ell-1}^{(i)} \right). \quad (12)$$

The MLMC estimator has two fundamental benefits we care about:

1. **Variance Reduction via Coupling:** Because P_ℓ and $P_{\ell-1}$ are computed on the same underlying Brownian path, their difference will have much smaller variance than the payoffs themselves. This allows fewer samples on finer level, which are more expensive to compute.
2. **Computational Efficiency via Sample Allocation:** Since coarse levels are computationally cheap but have higher variance, and fine levels are expensive but have lower variance in differences, an optimal allocation of the number of samples N_ℓ per level can be determined to reduce total cost.

To achieve a root mean square error (RMSE) tolerance ε , we balance bias (discretization error) and statistical error by choosing level L and sample sizes $\{N_\ell\}_{\ell=0}^L$ appropriately.

Assuming that rate of convergence for bias and variance are well behaved, usually that they converge monotonically, we can say they scale by:

$$|\mathbb{E}[P - P_L]| = \mathcal{O}(2^{-\alpha L}), \quad \text{and} \quad \text{Var}[P_\ell - P_{\ell-1}] = \mathcal{O}(2^{-\beta \ell}),$$

for some $\alpha, \beta > 0$, and with the cost of one sample at level ℓ proportional to, $C_\ell = \mathcal{O}(2^{\gamma \ell})$, where $\gamma > 0$, the optimal number of samples per level, has the following property:

$$N_\ell \propto \sqrt{\frac{\text{Var}[P_\ell - P_{\ell-1}]}{C_\ell}} \propto 2^{-\frac{\beta+\gamma}{2} \ell}. \quad (13)$$

In particular, if we let $\beta > \gamma$, the total computational cost to achieve root mean squared error ε scales by:

$$\mathcal{C}_{\text{MLMC}} = \mathcal{O}(\varepsilon^{-2}),$$

which matches the cost of standard Monte Carlo but with some reduced constant multiplying factors due to our variance reduction efforts. In contrast, a naive Monte Carlo simulation using only the finest level L requires computational cost which scales with, $\mathcal{C}_{\text{MC}} = \mathcal{O}(\varepsilon^{-3})$.

This computational gain makes the use of MLMC quite advantageous for this kind of problem, where the payoff depends on discretized path-dependent processes and fine time resolution is required. Overall, MLMC provides a way to reduce variance, by balancing discretization error and variance, and optimizing computational constraints.

2.5 Higher-Order Moment Estimation and Sparse Tensor Approximation

A common way to consider an assets risk profile, beyond considering its expected value, it is often useful to estimate higher-order moments like variance (spread), skewness (asymmetry), and kurtosis (tailedness). These moments provide a description of the variability, asymmetry, and tailedness behavior of the payoff distribution. Given the payoff random variable P , the moments are defined by:

$$\text{Mean} = \mu = \mathbb{E}[P] \quad (14)$$

$$\text{Variance} = \sigma^2 = \mathbb{E}[P^2] - \mu^2 \quad (15)$$

$$\text{Skewness} = \gamma = \frac{\mathbb{E}[(P - \mu)^3]}{\sigma^3} \quad (16)$$

$$\text{Kurtosis} = \kappa = \frac{\mathbb{E}[(P - \mu)^4]}{\sigma^4} \quad (17)$$

The direct estimation of moments such as $\mathbb{E}[P^k]$ for $k > 1$ can be extraordinarily computational expensive, especially in high-dimensional settings where the payoff $P = P(\mathbf{X})$ depends on a vector of random inputs $\mathbf{X} = (X_1, \dots, X_d)$. However, to efficiently approximate these expectations, I decided to utilize a sparse tensor product technique, particularly Smolyak grids, which mitigate the curse of dimensionality by selecting a subset of quadrature points in the multi-dimensional input space. **For a brief discussion of tensors and some relevant operations, see [here](#)**

To estimate the k -th moment $\mathbb{E}[P^k]$, define the function $f_k(\mathbf{x}) = P(\mathbf{x})^k$, which maps the input parameters \mathbf{x} to the k -th power of the payoff. The sparse tensor quadrature approximation is then given by:

$$\mathbb{E}[P^k] \approx A^{(q,d)}[f_k] = \sum_{\alpha \in \Lambda(q,d)} c_\alpha \sum_{\mathbf{x}_\alpha \in \mathcal{X}_\alpha} w_{\mathbf{x}_\alpha} f_k(\mathbf{x}_\alpha). \quad (18)$$

This approach drastically reduces the number of required function evaluations compared to a full tensor product grid, which scales exponentially with dimension d .

The use of sparse tensor approximations allowed me to maintain high-accuracy integration and moment estimation in high-dimensional spaces while computational costs grow with a polynomial, rather than exponentially, in d . This allowed me to estimate the variance, skewness, kurtosis, and potentially higher moments or joint moment tensors.

2.6 Mean Field Game (MFG) Feedback

Mean Field Games (MFGs) provide a way to model collective behavior of a large population of interacting agents without an the obscenely large computation cost of solving for each agent individually. Each agent is optimizing their own objective while responding to aggregate distribution conditions. In the context of financial markets, MFGs allow for the incorporation of endogenous feedback effects.

Consider a continuum of agents characterized by their state variable $x \in \mathbb{R}$ at time $t \in [0, T]$. The distribution of agents' states is described by a density function $m(x, t)$, which evolves over time as agents adjust their positions in response to both individual incentives and the overall population distribution. The dynamics of m are governed by the Fokker–Planck equation:

$$\partial_t m(x, t) = -\partial_x(v(x, t)m(x, t)) + \frac{\sigma^2}{2} \partial_{xx} m(x, t), \quad (19)$$

where $v(x, t)$ is the control or drift velocity field representing agents' optimal actions, and $\sigma > 0$ is the diffusion coefficient representing 'noise' or what may be more accurately categorized as uncertainty or other agent dynamics.

Each agent simultaneously seeks to maximize the expected payoff of some value function $u(x, t)$, which satisfies the backward Hamilton–Jacobi–Bellman (HJB) equation:

$$-\partial_t u(x, t) = H(x, \partial_x u(x, t), m(\cdot, t)) + \frac{\sigma^2}{2} \partial_{xx} u(x, t). \quad (20)$$

Where, H denotes the Hamiltonian, which encapsulates the agent's optimization problem. This is done by relating the state x , the gradient $\partial_x u$ (representing marginal value with respect to state), and the distribution m . Typically, the Hamiltonian takes the form

$$H(x, p, m) = \sup_{v \in \mathcal{A}} \{-vp - L(x, v, m)\},$$

where $p = \partial_x u$ and L is a running cost or Lagrangian that may depend on state, control, and the population distribution.

The optimal feedback control is obtained as the argmax of the Hamiltonian, and in the quadratic case (e.g., $L(x, v, m) = \frac{1}{2}v^2 + F(x, m)$), it simplifies to

$$v(x, t) = -\partial_x u(x, t). \quad (21)$$

The coupled system of (19)–(20) defines a forward-backward type of PDE system.

Closing the Feedback Loop in Asset Dynamics Incorporating the mean field game framework into asset price modeling allows the collective investor state to influence market dynamics endogenously. Let the mean agent state at time t be

$$\bar{x}(t) = \int_{\mathbb{R}} x m(x, t) dx, \quad (22)$$

which represents an some sort of aggregate factor, like sentiment, position, or macroeconomic which represents the distribution of agents. In order to use these result, we must somehow incorporate this feedback. Thus, the asset price dynamics for each asset S_t^i is modified to additively include this feedback effect in the drift term:

$$dS_t^i = (r + \phi(\bar{x}(t))) S_t^i dt + \sigma S_t^i dW_t^i, \quad (23)$$

where r is the risk-free rate, σ the volatility, and $\phi : \mathbb{R} \rightarrow \mathbb{R}$ is a function modeling how the mean field $\bar{x}(t)$ influences the asset price behavior. By endogenizing this approximation of market behavior, it allows for analysis via comparative statics.

3 Structure for MLLSMC without Mean Field Feedback

The Multilevel Least Squares Monte Carlo (MLLSMC) algorithm combines the strengths of the Longstaff–Schwartz regression method with the variance reduction of Multilevel Monte Carlo. Below is a thorough description of each step involved in the implementation of the MLLSMC algorithm:

0. Optimal Sample Size Allocation

An important factor of the MLMC algorithm is the efficient allocation of computation time across levels to minimize overall computation cost for a desired level of accuracy. Let,

$$V_\ell = \text{Var}[P_\ell - P_{\ell-1}] \quad \text{and} \quad C_\ell = \text{average computational cost per sample at level } \ell.$$

The total mean squared error (MSE) of the MLMC estimator can be decomposed into two elements, bias and variance components, and to achieve some RMSE tolerance ϵ , the optimal number of samples at each level is:

$$N_\ell = \left\lceil \frac{1}{\epsilon^2} \sqrt{\frac{V_\ell}{C_\ell}} \sum_{k=0}^L \sqrt{V_k C_k} \right\rceil.$$

Note that the ceiling is used because we need an integer number of paths. This formula balances the variance contributions and the cost per sample $\sqrt{(V_\ell/C_\ell)}$ so that more samples are drawn at coarser levels (where samples are cheaper but variance is higher) and fewer at finer levels (more costly samples but lower variance).

1. Path Simulation and Level Coupling

For each discretization level $\ell = 0, 1, \dots, L$, we define a temporal grid on the interval $[0, T]$ with uniform time step,

$$\Delta t_\ell = \frac{T}{2^{\ell+1}}$$

and exercise times, where $M_\ell = 2^{\ell+1}$ is the total number of steps for discretization level ℓ , $t_m^\ell = m\Delta t_\ell$, $m = 0, 1, \dots, M_\ell$. We simulate N_ℓ independent sample paths of each of the underlying assets,

or equivalently, N_ℓ independent basket price process $\{B_t^{(i,\ell)} : t \in \{t_m^\ell\}_{m=0}^{M_\ell}\}_{i=1}^{N_\ell}$ modeled by correlated geometric Brownian motions. The stochastic differential equation (SDE) governing the asset dynamics for the d -dimensional basket is:

$$dS_t = \text{diag}(S_t) \left((r\mathbf{1} + \boldsymbol{\mu})dt + \Sigma^{1/2}dW_t \right),$$

where:

- $S_t = (S_t^1, \dots, S_t^d)^\top$ is the vector of asset prices at time t ,
- r is the risk-free interest rate,
- $\boldsymbol{\mu}$ is the drift vector,
- $\Sigma \in \mathbb{R}^{d \times d}$ is the covariance matrix of asset returns,
- W_t is a standard d -dimensional Brownian motion.

To enforce strong coupling between consecutive levels ℓ and $\ell-1$, we generate the Brownian increments on the finer grid Δt_ℓ and aggregate them to form increments on the coarser grid $\Delta t_{\ell-k} = 2^k \Delta t_\ell \rightarrow \Delta t_{\ell-1} = 2\Delta t_\ell$. For example, if $\Delta W_m^{(\ell)}$ denotes the Brownian increment over $[t_{m-1}^\ell, t_m^\ell]$, then the increment at level $\ell-1$ over $[t_{m-1}^{\ell-1}, t_m^{\ell-1}]$ is given by:

$$\Delta W_m^{(\ell-1)} = \Delta W_{2m-1}^{(\ell)} + \Delta W_{2m}^{(\ell)}.$$

This ensures that the paths at levels ℓ and $\ell-1$ are coupled together to the same underlying Brownian motion.

2. Backward Induction and Continuation Value Estimation via Regression

For each path i and discretization level ℓ , we perform backward induction starting at maturity time $t_{M_\ell}^\ell = T$. The payoff at maturity for the American basket put option is:

$$V_{t_{M_\ell}^\ell}^{(i,\ell)} = \max \left(K - B_{t_{M_\ell}^\ell}^{(i,\ell)}, 0 \right).$$

Moving backward through each exercise time t_m^ℓ , $m = M_\ell - 1, \dots, 0$, we want to estimate the continuation value:

$$C_{t_m^\ell}^{(i,\ell)} = \mathbb{E}^\mathbb{Q} \left[e^{-r\Delta t_\ell} V_{t_{m+1}^\ell}^{(i,\ell)} \mid \mathcal{F}_{t_m^\ell} \right],$$

where $\mathcal{F}_{t_m^\ell}$ is the filtration representing the information available up to time t_m^ℓ . Since the conditional expectation cannot or is too difficult to be computed exactly, it is approximated via OLS regression. The discounted continuation payoffs, $\{e^{-r\Delta t_\ell} V_{t_{m+1}^\ell}^{(i,\ell)}\}_{i=1}^{N_\ell}$, serve as the dependent variable, while the basket prices, $\{B_{t_m^\ell}^{(i,\ell)}\}_{i=1}^{N_\ell}$, serve as the regressors.

The regression model uses Laguerre polynomials, $\{L_k(\cdot)\}_{k=0}^K$. The continuation value is modeled as:

$$C_{t_m^\ell}^{(i,\ell)} \approx \sum_{k=0}^K \beta_k^{(\ell,m)} L_k \left(B_{t_m^\ell}^{(i,\ell)} \right).$$

The coefficient vector $\boldsymbol{\beta}^{(\ell,m)} = (\beta_0^{(\ell,m)}, \dots, \beta_K^{(\ell,m)})^\top$ is obtained by minimizing the least squares criterion:

$$\boldsymbol{\beta}^{(\ell,m)} = \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^{K+1}} \sum_{i=1}^{N_\ell} \left(e^{-r\Delta t_\ell} V_{t_{m+1}^\ell}^{(i,\ell)} - \sum_{k=0}^K \beta_k L_k \left(B_{t_m^\ell}^{(i,\ell)} \right) \right)^2.$$

This regression step produces an estimator of the conditional expectation operator defining the continuation value.

3. Optimal Exercise Policy Determination

At each time step t_m^ℓ for each path i , the option holder must choose between exercising immediately and continuing to hold the option. The immediate exercise payoff (intrinsic value) is:

$$I_{t_m^\ell}^{(i,\ell)} = \max \left(K - B_{t_m^\ell}^{(i,\ell)}, 0 \right).$$

Comparing the immediate exercise value to the estimated continuation value $\hat{C}_{t_m^\ell}^{(i,\ell)}$, the optimal stopping rule is:

$$V_{t_m^\ell}^{(i,\ell)} = \max \left(I_{t_m^\ell}^{(i,\ell)}, \hat{C}_{t_m^\ell}^{(i,\ell)} \right).$$

If $I_{t_m^\ell}^{(i,\ell)} > \hat{C}_{t_m^\ell}^{(i,\ell)}$, the option is exercised at t_m^ℓ , and no further continuation value is considered. Otherwise, the option is held, and the payoff is the discounted continuation value. Therefore, $V_{t_m^\ell}^{(i,\ell)}$ can be defined peicewise as:

$$V_{t_m^\ell}^{(i,\ell)} = \begin{cases} I_{t_m^\ell}^{(i,\ell)} & \text{if } I_{t_m^\ell}^{(i,\ell)} > \hat{C}_{t_m^\ell}^{(i,\ell)} \\ \hat{C}_{t_m^\ell}^{(i,\ell)} & \text{otherwise} \end{cases}$$

By iterating backward through all exercise times, a sequence of stopping decisions $\tau^{(i,\ell)}$ for each path is determined, defining the approximate optimal stopping policy at level ℓ .

4. Computation of Discounted Payoffs

For each path i at level ℓ , once the stopping time $\tau^{(i,\ell)}$ is identified, the realized discounted payoff is computed as:

$$P_\ell^{(i)} = e^{-r\tau^{(i,\ell)}} \max \left(K - B_{\tau^{(i,\ell)}}^{(i,\ell)}, 0 \right).$$

If the option is never exercised before maturity (i.e., $\tau^{(i,\ell)} = T$), then the payoff reduces to the terminal payoff.

This collection $\{P_\ell^{(i)}\}_{i=1}^{N_\ell}$ forms the empirical basis for estimating expected option values and their differences between levels.

5. MLMC Estimator

We estimate each term $\mathbb{E}[P_\ell - P_{\ell-1}]$ by the sample mean over N_ℓ asset-price paths:

$$\hat{Y}_\ell = \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \left(P_\ell^{(i)} - P_{\ell-1}^{(i)} \right).$$

The overall MLMC estimator for the option price, at inception, is:

$$\hat{V}_0 = \sum_{\ell=0}^L \hat{Y}_\ell = \sum_{\ell=0}^L \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \left(P_\ell^{(i)} - P_{\ell-1}^{(i)} \right).$$

Due to the coupling of the Brownian increments across levels, the variance of the difference $P_\ell - P_{\ell-1}$ is reduced, yielding a better rate of convergence as compared to standard Monte Carlo estimation.

3.1 Moment Estimation and Risk Profile Extraction

A key component of risk analysis and quantitative finance is not only the estimation of the expected option payoff (as is assumed under \mathbb{Q}) but also the quantification of higher-order moments (variance, skewness, and kurtosis) that characterize the payoff distribution's shape and inform about risk.

MLMC Estimation of Central Moments: For the k -th raw moment of the payoff P , we use the telescoping decomposition:

$$\mathbb{E}[P^k] = \mathbb{E}[P_0^k] + \sum_{\ell=1}^L \mathbb{E}[P_\ell^k - P_{\ell-1}^k]. \quad (24)$$

The MLMC estimator for $\mathbb{E}[P^k]$ is then

$$\widehat{\mathbb{E}}[P^k] = \sum_{\ell=0}^L \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \left((P_\ell^{(i)})^k - (P_{\ell-1}^{(i)})^k \right), \quad (25)$$

where $P_\ell^{(i)}$ is the payoff from the i -th sample at level ℓ , and by convention $P_{-1}^{(i)} := 0$.

This estimator uses the same coupled path simulations used for pricing to compute moments.

Sparse Tensor Approximation for High-Dimensional Moment Computation: In high-dimensional basket options or when dealing with multiple risk factors, directly computing moments via brute force averaging becomes computationally challenging or even infeasible due to the curse of dimensionality.

To avoid such a situation, we use sparse grid tensor product techniques, such as the Smolyak algorithm, to approximate moment functionals in an effective manner. Formally, we express the moment as

$$\mathbb{E}[P^k] \approx \sum_{\alpha \in \Lambda} w_\alpha f_k(x_\alpha),$$

where:

- Λ is a sparse index set generated by the Smolyak construction
- w_α are quadrature weights associated with sparse grid nodes x_α
- $f_k(x) = (P(x))^k$ is the moment function evaluated at quadrature points x_α .

This sparse tensor approach dramatically reduces the number of function evaluations necessary for accurate moment estimation, thus allowing the reasonable computation of variance, skewness, kurtosis, and higher-order moments for complex payoff distributions.

Interpretation and Use in Risk Profiles:

- The *variance* $\text{Var}[P] = \mathbb{E}[P^2] - \mathbb{E}[P]^2$ quantifies payoff dispersion.

- The *skewness* captures asymmetry, computed as:

$$\text{Skew}[P] = \frac{\mathbb{E}[(P - \mu)^3]}{\sigma^3},$$

where $\mu = \mathbb{E}[P]$ and $\sigma^2 = \text{Var}[P]$.

- The *kurtosis* measures tail heaviness:

$$\text{Kurt}[P] = \frac{\mathbb{E}[(P - \mu)^4]}{\sigma^4} - 3.$$

These higher moments provide critical insights into tail risk and the likelihood of extreme outcomes, which are important for a variety of application, including stress testing, something I plan on doing a little project on.

4 Results

Redoing this

5 Conclusion

Interesting project, learned a lot.

6 Key Terms:

Most of these terms are for my own notes and there may be errors, particularly in the mathematical definitions as I tried to do all of them without referring to outside resources. These are here if anyone else viewing this document would like to access them.

6.1 Common Terminology

These are terms or ideas I feel comfortable saying I understand, on some level whether introductory or in-depth. I am familiar as a result of either previous work (coursework, other projects, readings) or me exploring project ideas.

- **Martingale:** A stochastic process $\{X_t\}_{t \geq 0}$ adapted to a filtration $\{\mathcal{F}_t\}_{t \geq 0}$ is called a martingale with respect to a probability measure \mathbb{P} if, for all $s \leq t$, it satisfies $\mathbb{E}[|X_t|] < \infty$ and $\mathbb{E}[X_t | \mathcal{F}_s] = X_s$.

Intuitively, a martingale represents what might be thought of as a "fair game". More generally, given the present value and all historic information, the expected future value discounted into current terms, is equivalent to the current value. In our instance, I use the term $e^{-r\tau}$ to represent that instead of purchasing an option, a buyer might rather put their money into a bank account and receive the accrued interest.

- **Brownian Motion:** Brownian Motion is a stochastic process W_t such that:

1. $W_0 = 0$
2. W_t is almost certainly continuous. Formally,

$$P \left(\sup_{s, t \in [0, T], |s-t| \leq \delta} |W_s - W_t| \rightarrow 0 \text{ as } \delta \rightarrow 0 \right) = 1$$

3. W_t has independent increments, meaning that for two intervals, (a, b) and (c, d) such that $b \leq c$, $Cov[(W_b - W_a), (W_d - W_c)] = 0$
4. For some interval (a, b) such that $0 \leq a \leq b$, $(W_b - W_a) \sim \mathcal{N}(0, b - a)$

Geometric Brownian Motion: For our purposes, I will refer to Geometric Brownian Motion as Brownian Motion such that each step is taken from $\mathcal{N}(0, 1)$.

- **Filtration:** A filtration is an increasing sequence of σ -algebras $\{\mathcal{F}_t\}_{t \geq 0}$. This sequence represents the accumulation of information over time. Formally, it satisfies $\mathcal{F}_s \subseteq \mathcal{F}_t$ for all $0 \leq s \leq t$. In the context of stochastic processes, a filtration models what is known at each time t and is used to define adapted processes and conditional expectations. Filtrations represent the idea of having “no foresight”, ensuring that decisions or events depend only on currently available information, as would occur in the real world, mostly. If it is not a familiar concept, as it wasn’t for me, view the σ -algebras term.
- **Probability Space**
- **Markov Property**
- **Probability Measure**
- **Optimal Stopping Problem (Markov Process Formulation):** An optimal stopping problem is a type of problem whose objective is to determine the optimal time to stop a stochastic process in order to maximize (or minimize) the expected product of some function of system. When the underlying process is a Markov process $(X_t)_{t \geq 0}$ on some filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t), \mathbb{P})$, the objective is to find a maximizing stopping time τ^* .

Mathematically, the general form of the objective function is:

$$V(x) = \sup_{\tau \in \mathcal{T}} \mathbb{E}_x [g(X_\tau)],$$

where:

1. X_t is a Markov process with initial state $X_0 = x$.
2. $g : \mathbb{R}^d \rightarrow \mathbb{R}$ is a measurable function.
3. \mathcal{T} is the set of all stopping times with respect to the filtration (\mathcal{F}_t) .
4. $\mathbb{E}_x[\cdot]$ denotes expectation conditional on $X_0 = x$.

In our context, the function $V(x)$ represents the expected reward from stopping the process, and the optimal stopping time τ^* is the one that achieves this supremum (meaning any time at which the expected value is largest).

- **Dynamic Programming Principle (DPP):** The Dynamic Programming Principle asserts that the value of a stochastic control or optimal stopping problem at an initial time depends (recursively) on the value of the same problem at future times. Essentially, it states that we must start at the final step, taking all previous steps as given, and solve for the optimal decision, then move to the next previous step, doing the same thing all the way to the beginning (backward induction).

In the context of optimal stopping for a Markov process $(X_t)_{t \geq 0}$ with the value function $V(x)$, the DPP states that for any admissible stopping time θ ,

$$V(x) = \sup_{\tau \in \mathcal{T}} \mathbb{E}_x [g(X_\tau)] = \sup_{\tau \in \mathcal{T}} \mathbb{E}_x [\mathbb{1}_{\{\tau \leq \theta\}} g(X_\tau) + \mathbb{1}_{\{\tau > \theta\}} V(X_\theta)].$$

This recursive formula states that if stopping is not optimal at time θ , then the optimal strategy must continue, and the value at x equals the expected value of applying the optimal strategy starting from the state X_θ .

This principle is a core idea in many of the mathematical ideas I used, so it is an important idea to keep in mind.

- **Laguerre Polynomials:** Laguerre polynomials are a sequence of orthogonal polynomials defined on the interval $[0, \infty)$ with respect to the weight function e^{-x} . They arise in solving a variety of types of equations. The n -th Laguerre polynomial, denoted $L_n(x)$, can be defined by the formula (Rodrigues' formulation with $\alpha = 0$):

$$L_n(x) = \frac{e^x}{n!} \frac{d^n}{dx^n} (x^n e^{-x}).$$

These polynomials satisfy the relation

$$\int_0^\infty e^{-x} L_m(x) L_n(x) dx = 0 \quad \text{for } m \neq n.$$

Laguerre polynomials are commonly used as basis functions in regression methods, the example that inspired the use of these was an application of the Longstaff-Schwartz algorithm.

- **Telescoping Sum Identity:** A telescoping sum is a series in which consecutive terms cancel out, resulting in a simplified expression involving only the first and last terms. The general form of a telescoping sum is:

$$\sum_{k=1}^n (a_k - a_{k+1}) = a_1 - a_{n+1}.$$

A useful little identity.

- **Strong Coupling Between Consecutive Levels ℓ :** In the context of Multilevel Monte Carlo (MLMC), this refers to the construction of paired estimators on levels ℓ and $\ell-1$ such that the difference between their outputs is minimized in variance. This is achieved by correlating the simulations across levels through shared randomness or co-simulation strategies. Let P_ℓ and $P_{\ell-1}$ be approximations of some quantity of interest (in this case price paths) at levels ℓ and $\ell-1$, respectively. A strong coupling implies that the variance of their difference decays rapidly:

$$\text{Var}(P_\ell - P_{\ell-1}) \ll \text{Var}(P_\ell), \quad \text{as } \ell \rightarrow \infty.$$

This variance decay is the key idea behind the efficiency of the telescoping sum identity used in MLMC:

$$\mathbb{E}[P_L] = \mathbb{E}[P_0] + \sum_{\ell=1}^L \mathbb{E}[P_\ell - P_{\ell-1}].$$

Strong coupling ensures that fewer samples are needed on finer levels (which are more computationally expensive) because the difference estimators $P_\ell - P_{\ell-1}$ exhibit low variance. This underpins the optimal sample allocation strategy:

$$M_\ell \propto \sqrt{\frac{\text{Var}(P_\ell - P_{\ell-1})}{C_\ell}},$$

where C_ℓ is the computational cost per sample at level ℓ .

6.2 Complicated Terms

These are terms that, while I understand them on some level, I could not either prove nor expand upon these concepts. Ideas I either threw in because these were required for other things I was interested in or because I felt ambitious and went too far exploring some idea.

- **Itô's Lemma:** Provides the differential of a function of a stochastic process. If W_t is an Itô process of the form $dX_t = \mu_t dt + \sigma_t dW_t$, and $f(t, X_t) \in C^2$, then Itô's Lemma states the following:

$$df(t, X_t) = \left(\frac{\partial f}{\partial t} + \mu_t \frac{\partial f}{\partial x} + \frac{1}{2} \sigma_t^2 \frac{\partial^2 f}{\partial x^2} \right) dt + \sigma_t \frac{\partial f}{\partial x} dW_t.$$

Itô's Lemma is waaaaay beyond my understand of stochastics and so I refer to it only when necessary.

- **Algebra (of sets):** An algebra \mathcal{A} over a set Ω is a collection of subsets of Ω that is closed under finite unions, finite intersections, and complements. That is, if $A, B \in \mathcal{A}$, then $A \cup B$, $A \cap B$, and A^c (the complement of A with respect to Ω) are also in \mathcal{A} . Algebras provide a foundational structure for measurable spaces and are a precursor to σ -algebras, which extend closure to countable operations and are used to define filtrations and probability measures.
- **σ -algebra:** A σ -algebra \mathcal{F} over a set Ω is a collection of subsets of Ω that includes the empty set and is closed under complements and countable unions. Formally, if $A_1, A_2, \dots \in \mathcal{F}$, then $\bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$ and $A^c \in \mathcal{F}$ for any $A \in \mathcal{F}$. σ -algebras are essential in probability theory for defining measurable spaces and ensuring that random variables and events are well-defined under a probability measure.
- **Cholesky Decomposition:** A matrix factorization method that expresses a symmetric, positive-definite matrix A as the product $A = LL^\top$, where L is a lower triangular matrix with real and positive diagonal entries. From my exploration, it is used for efficient numerical solutions to linear systems. The three places I saw it used in practice was when the matrix represented a covariance structure.

Mathematically, for $A \in \mathbb{R}^{n \times n}$ with $A = A^\top$ and $x^\top A x > 0$ for all non-zero $x \in \mathbb{R}^n$, there exists a unique lower triangular matrix L such that: $A = LL^\top$. A *Simplified algorithm* (for A of size $n \times n$), is given by:

1. For $i = 1$ to n :
 - (a) $L_{ii} = \sqrt{A_{ii} - \sum_{k=1}^{i-1} L_{ik}^2}$
 - (b) For $j = i + 1$ to n :

$$L_{ji} = \frac{1}{L_{ii}} \left(A_{ji} - \sum_{k=1}^{i-1} L_{jk} L_{ik} \right)$$

2. Set $L_{ij} = 0$ for all $j > i$

- **Suitable Regularity Conditions:** I refer to these a few times, and while my formulation are pretty simple, I thought it important to include some description of exactly what I meant when I said "Suitable Regularity Conditions".

These refer to the mathematical assumptions I've imposed on the functions, processes, or spaces I used to ensure well-posedness, existence, uniqueness, and stability (stability was a major concern for my first few attempts since I was much newer to numerics when I started this project).

Regularity conditions include continuity, boundedness, and differentiability, for functions such as the payoff functions, drift and diffusion coefficients of my stochastic differential equations, and the value functions. As an example of why I made note of these (I can't find the source for the assertion and proof for this statement):

For functions $f : \mathbb{R}^d \rightarrow \mathbb{R}$, there exists a constant $L > 0$ such that for all $x, y \in \mathbb{R}^d$,

$$|f(x) - f(y)| \leq L \|x - y\|.$$

This Lipschitz condition ensures stability and convergence in the stochastic model.

- **Sparse Tensor Products**

I hadn't done any linear algebra for a number of months so I wrote a bit of a review sort of thing for myself here. Then I talk a bit about

1. Basic Definitions

- **Vector Space:** A set V with two operations — addition (+) and scalar multiplication (\times) — such that for any vectors $u, v \in V$ and any scalars $a \in \mathbb{R}$, the following hold:

$$u + v \in V, \quad a \times u \in V,$$

and these operations satisfy properties like associativity, commutativity, existence of zero vector, and distributivity.

- **Bilinear Combination:** Given two vectors u and v from vector spaces V and W , a bilinear combination is an expression that is linear in each argument separately, such as

$$a \times u + b \times v,$$

where a, b are scalars. More generally, a bilinear map $B : V \times W \rightarrow X$ satisfies linearity in each argument:

$$B(au_1 + bu_2, w) = aB(u_1, w) + bB(u_2, w), \quad B(v, cw_1 + dw_2) = cB(v, w_1) + dB(v, w_2).$$

- **Basis:** A set of vectors $\{b_1, b_2, \dots, b_n\}$ in a vector space V such that any vector $v \in V$ can be written uniquely as a linear combination

$$v = a_1b_1 + a_2b_2 + \dots + a_nb_n,$$

where the a_i are scalars.

Example: In $V = \mathbb{R}^2$, the vectors $b_1 = (1, 0)$ and $b_2 = (0, 1)$ form a basis because any vector (x, y) can be written as $xb_1 + yb_2$.

2. What is a Tensor?

A **tensor** is a multidimensional generalization of scalars, vectors, and matrices:

- A *scalar* is a single number (0th-order tensor).
- A *vector* is a list of numbers (1st-order tensor).
- A *matrix* is a grid of numbers (2nd-order tensor).
- Higher-order tensors are arrays of numbers with three or more indices.

Mathematically, a tensor of order d is an element of the tensor product of d vector spaces:

$$T \in V_1 \otimes V_2 \otimes \dots \otimes V_d.$$

3. Tensor Product of Vector Spaces

Given two vector spaces V and W , the **tensor product** $V \otimes W$ is a new vector space whose elements represent all combinations formed by multiplying vectors from V and W . This operation formalizes the idea of pairing elements from each space.

If $\{v_i\}$ is a basis for V and $\{w_j\}$ is a basis for W , then the tensor product space $V \otimes W$ has a basis consisting of all vectors of the form

$$v_i \otimes w_j,$$

where the symbol \otimes denotes the tensor product.

Generalized Example Using Elementary Operators:

- Consider vectors $v = (v_1, v_2)$ and $w = (w_1, w_2)$.
- Their tensor product can be thought of as the set of all products of elements from v and w :

$$\{v_1 \times w_1, \quad v_1 \times w_2, \quad v_2 \times w_1, \quad v_2 \times w_2\}.$$

- We can arrange these as a matrix:

$$\begin{bmatrix} v_1 w_1 & v_1 w_2 \\ v_2 w_1 & v_2 w_2 \end{bmatrix}.$$

This illustrates how tensor products combine information from two vectors into a higher-order object (here, $V, W \in \mathbb{R}^2$ and $X \in \mathbb{R}^3$, and the product $\otimes : V \times W \rightarrow X$).

For functions $f(x)$ and $g(y)$, the tensor product corresponds to:

$$(f \otimes g)(x, y) = f(x) \times g(y).$$

This principle generalizes to d vector spaces or functions of d variables.

4. Direct Sum of Vector Spaces

The **direct sum** of vector spaces V and W , denoted $V \oplus W$, is the space containing all pairs (v, w) with $v \in V$ and $w \in W$. Vector addition and scalar multiplication are applied component-wise as:

$$\begin{aligned} (v_1, w_1) + (v_2, w_2) &= (v_1 + v_2, w_1 + w_2), \\ a \times (v, w) &= (a \times v, a \times w). \end{aligned}$$

Dimensionally, the direct sum adds the sizes of the spaces:

$$\dim(V \oplus W) = \dim(V) + \dim(W).$$

Example: Consider the xy-plane, which is the direct sum of the two, single dimensional, real vector spaces X and Y .

5. Tensor Products in Function Approximation

To approximate a function of multiple variables, $f(x_1, x_2, \dots, x_d)$, one approach is to build bases for each variable individually and then combine them using tensor products.

For example, suppose we approximate functions of one variable by polynomials of degree up to n_j , forming spaces V_{n_j} . The space of approximations in d variables is the tensor product:

$$V_{\mathbf{n}} = V_{n_1} \otimes V_{n_2} \otimes \dots \otimes V_{n_d}.$$

The dimension grows as:

$$\dim(V_{\mathbf{n}}) \approx \prod_{j=1}^d (n_j + 1).$$

This exponential growth with d is commonly known as the *curse of dimensionality*.

6. Sparse Tensor Products

Sparse tensor products reduce this complexity by selecting only those combinations where the sum of approximation levels across dimensions is bounded. Formally, using *detail spaces* W_{n_j} representing increments in approximation, the sparse space at level L is:

$$V_L^{\text{sparse}} = \bigoplus_{\substack{\mathbf{n} \in \mathbb{N}^d \\ \sum_{j=1}^d n_j \leq L+d-1}} \bigotimes_{j=1}^d W_{n_j}.$$

This restriction reduces the number of basis functions from scaling by $\mathcal{O}(n!)$ to $\mathcal{O}(n^x)$ (I need to fact check this, I think I got something wrong here).

- **Smolyak Grids:** A sparse, adaptive set of points used for high-dimensional function approximation and integration. It efficiently combines one-dimensional quadrature or interpolation points across multiple dimensions, selecting only those points that contribute most to the approximation accuracy. This reduces the exponential growth in the number of points (the curse of dimensionality) typical of

full tensor grids. The Smolyak construction uses a combination of hierarchical incremental levels to form the grid and balances accuracy and computational cost.

Mathematically, for dimension d and level L , the Smolyak operator $A^{(L)}$ applied to a function f is defined as

$$A^{(L)}(f) = \sum_{|\mathbf{i}| \leq L+d-1} \left(\bigotimes_{j=1}^d \Delta_{i_j} \right) (f),$$

where $\mathbf{i} = (i_1, \dots, i_d)$ is a multi-index, $|\mathbf{i}| = \sum_{j=1}^d i_j$, and Δ_{i_j} denotes the hierarchical difference operator at level i_j in the j -th dimension.

7 Ideas for Expansion:

7.1 Improvements:

- Rename or just changing all the MLMC-LSMC, and MLMC-LSMC-MFG with something more reasonable, kind of a lot of letters.
- Replace we/our with I/my, less professional but more appropriate
- Some things are passed without any justification, give atleast a little something for each thing. (Δt_ℓ size on page 7)
- Add hyperlinks to key terms, previous definition etc.

7.2 Ideas:

Either interesting ideas to add or things to improve upon:

- Tikhonov Regularization
- Multi-Factor inputs, not just prices
- Laguerre Polynomials - Added to regression step, justify and find some optimal number
- Mean Field Game Theory - Added the fundamentals, expand and adapt
- Higher Moments - Used sparse tensor products to approximate, review and adjust
- Key Terms section - Proofed, always worth a re-proof