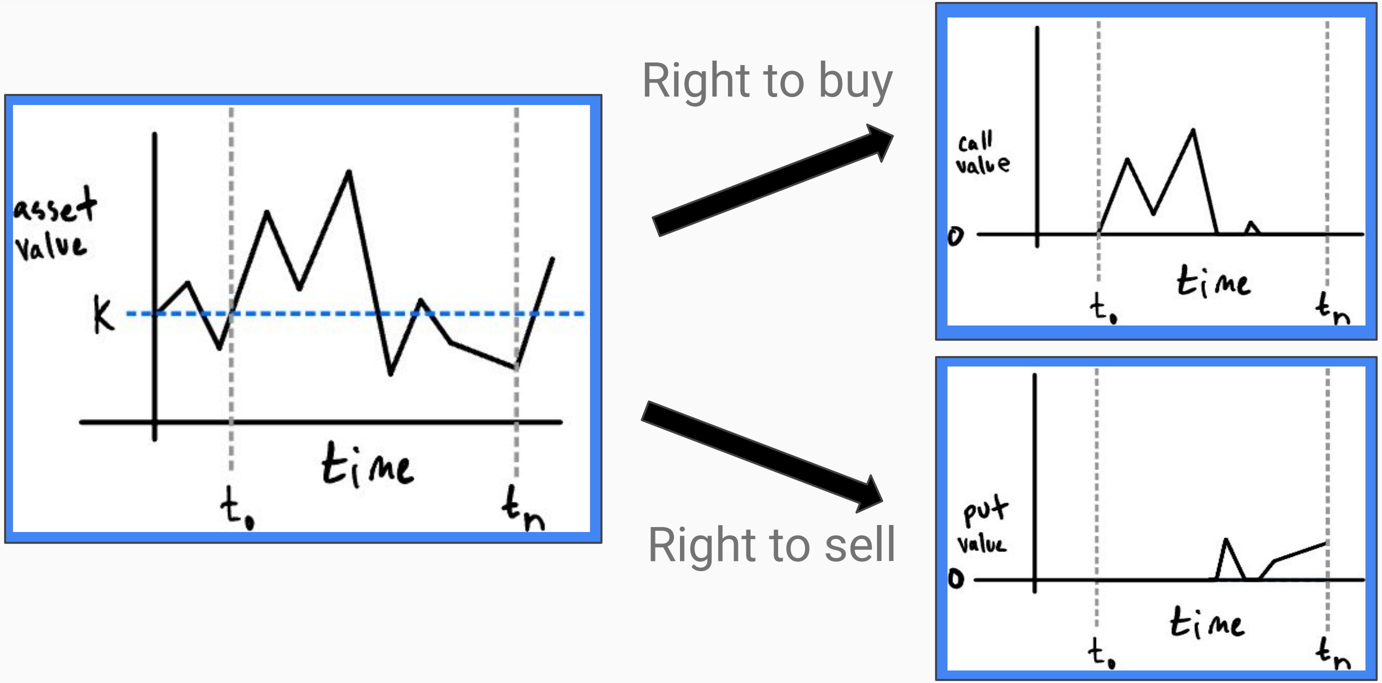
Accelerating American-style Options

By: Cormac Taylor

Background

Options Basics (long)



Options Types

European:

- Can exercise only at t_n

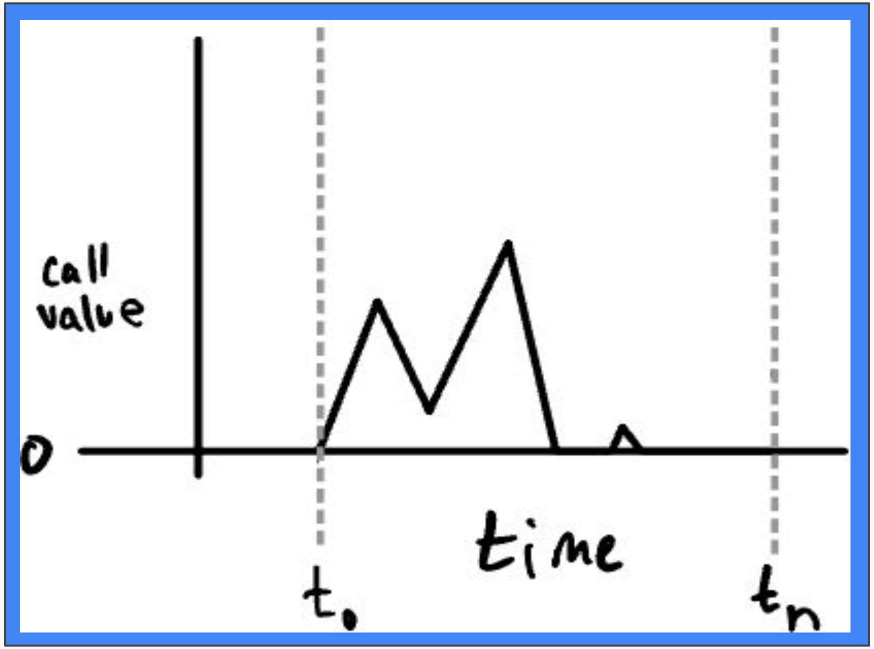
American:

- Can exercise once any time between t_0 and t_n

Swing:

- Can exercise multiple times between t_0 and t_n

Many more ...



Goal

Calculate Expected Value Given Optimal Behavior

Simple arbitrage:

```
if (expected_value > current_value)
    hold for now
else
    exercise right
```

Traditional Approach

Least-Square Monte Carlo (Longstaff-Schwartz)

$$V = \text{esssup}\Big\{\mathbb{E}\big(\varphi_\tau(X_\tau)\big|\mathcal{F}_0\big) : \tau \text{ is a } (\mathcal{F}_k)\text{-stopping time}\Big\}$$

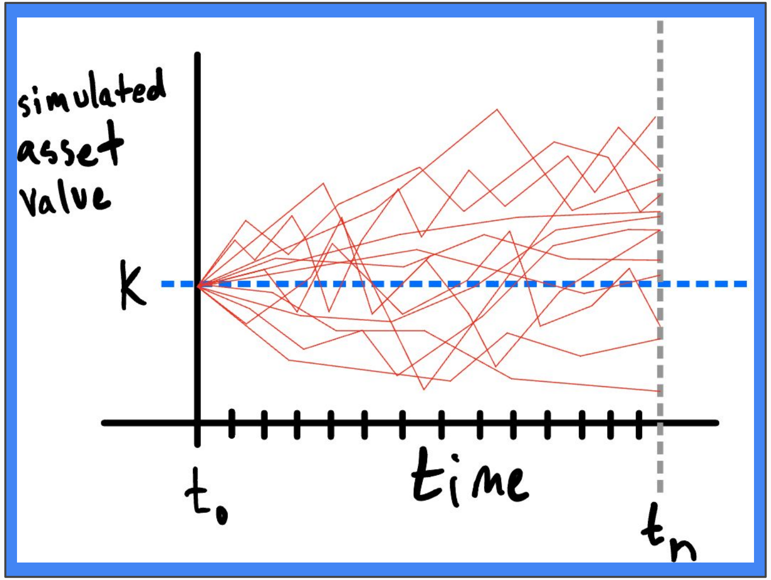
$$V_n = \varphi_{t_n}(X_n)$$

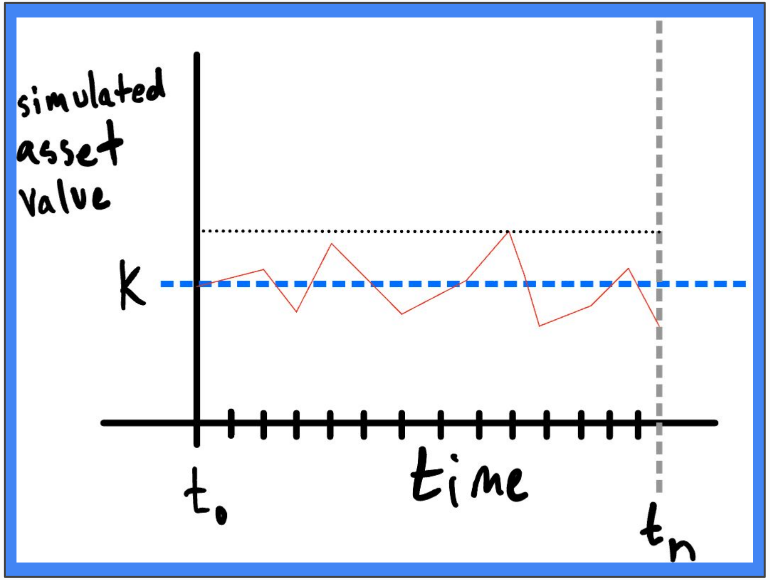
$$V_k = \max\Big(\varphi_{t_k}(X_k);\ \mathbb{E}\big(V_{k+1}\big|\mathcal{F}_k\big)\Big),\ 0 \le k \le n-1.$$

Simulating M paths of (X_k) (forward step)

Starting at $k = n-1$, approximate $f_k(x) = \mathbb{E}(V_{k+1}|X_k = x)$ by a Least Squares regression and proceed backwards to 0. (backward step)

Least-Square Monte Carlo Intuition





Motivation

Need to make decisions NOW! A lot can happen in a few seconds.

Due to the use of backward dynamic programming, the Least-Square Monte Carlo approach is inherently sequential.

Need a different approach in order to allow for parallelization.

Parallel-optimized Approach

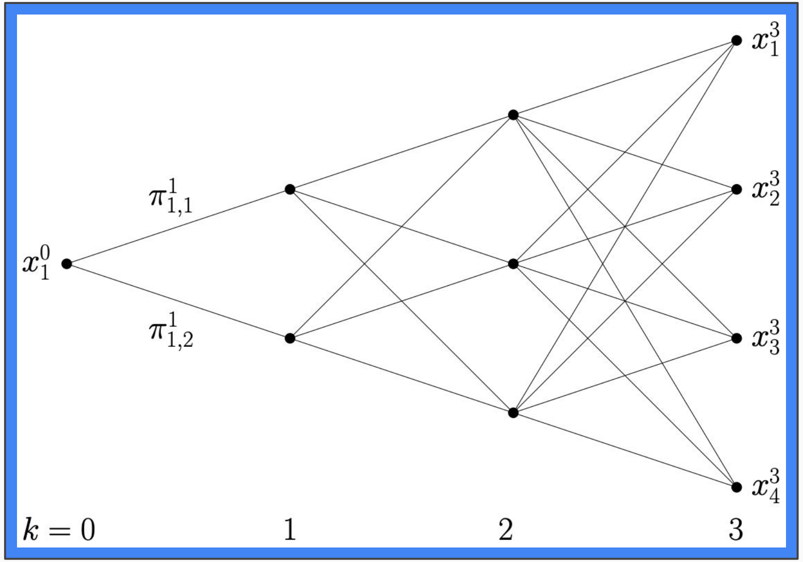
Tree Quantization

$$P(Q) = \operatorname{esssup}\Bigg\{\mathbb{E}\Bigg(\sum_{k=0}^{n-1} q_k v_k(X_k)\Bigg|\mathcal{F}_0\Bigg) : \forall k = 0, \ldots, n-1 : \\ q_k : (\Omega, \mathcal{F}_k) \to [0,1], \sum_{k=0}^{n-1} q_k \in [Q_{\min}, Q_{\max}]\Bigg\},$$

$$\begin{aligned} \widehat{P}_n &\equiv 0 \\ \widehat{P}_k(Q^k) &= \max\Big\{x v_k(\widehat{X}_k^{\Gamma_k}) \\ &\qquad + \mathbb{E}\big(\widehat{P}_{k+1}(\chi^{n-k-1}(Q^k, x))\big|\widehat{X}_k^{\Gamma_k}\big), x \in \{0,1\} \cap I_{Q^k}^{n-k-1}\Big\}, \end{aligned}$$

$$|P(Q) - \widehat{P}_0(Q)| \leq C \sum_{k=0}^{n-1} \Big(\mathbb{E}\|X_k - \widehat{X}_k^{\Gamma_k}\|^2\Big)^{1/2}.$$

Tree Quantization Intuition



$$\begin{aligned}\pi_{ij}^k &= \mathbb{P}\big(\widehat{X}_k^{\Gamma_k} = x_j^k \mid \widehat{X}_{k-1}^{\Gamma_{k-1}} = x_i^{k-1}\big)\\ &= \mathbb{P}\big(X_k \in C_j(\Gamma_k) \mid X_{k-1} \in C_i(\Gamma_{k-1})\big).\end{aligned}$$

Implementation

Algorithms

- CPU only Least Square Monte Carlo (lsm_cpu)
- Partially accelerated Least Square Monte Carlo (lsm_gpu)
- Path-wise tree quantization (qt2)
- Time-wise tree quantization (qt3)

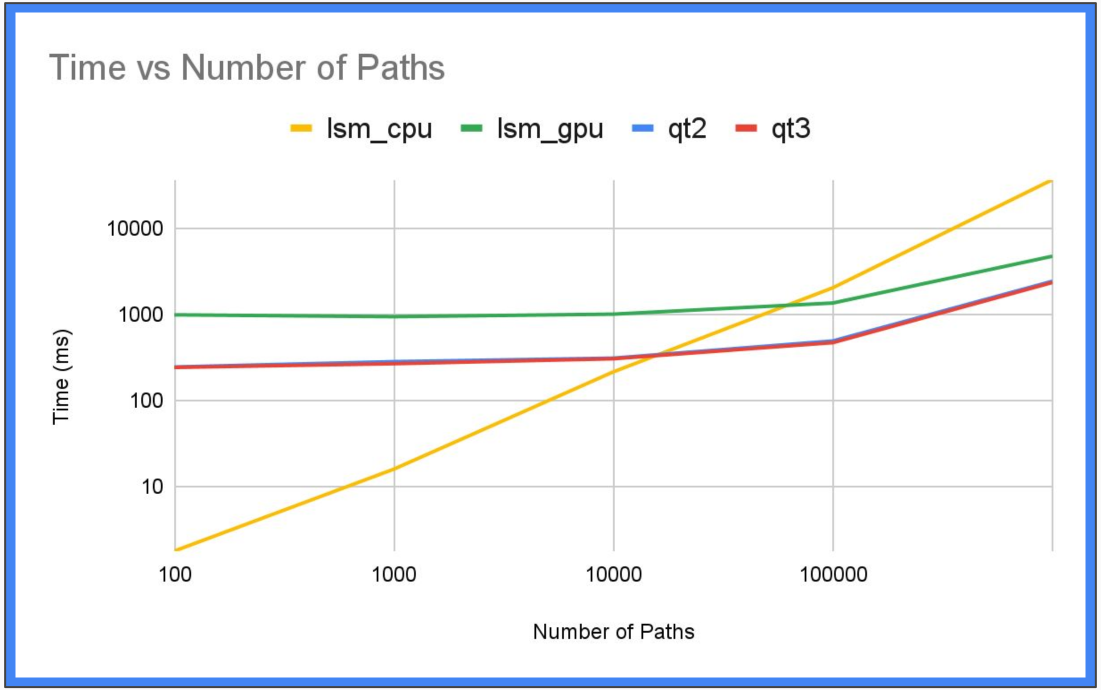
Tree quantization pseudocode

```
Algorithm II
for m = 1,...,M do in parallel
  # Initialization
  x ← x_0, i ← 0, p_1^i ← 1
  for k = 1,...,n do
    Simulate ε_k
    x ← A_k x + T_k ε_k
    Find NN-Index j of x in Γ_k

    atomic increment p_ij^k
    atomic increment p_j^{k+1}
    i ← j
  end for
end for in parallel
Synchronize threads
Set in parallel π_ij^k ← p_ij^k / p_i^k,   1 ≤ i,j ≤ N_k, 1 ≤ k ≤ n.
```

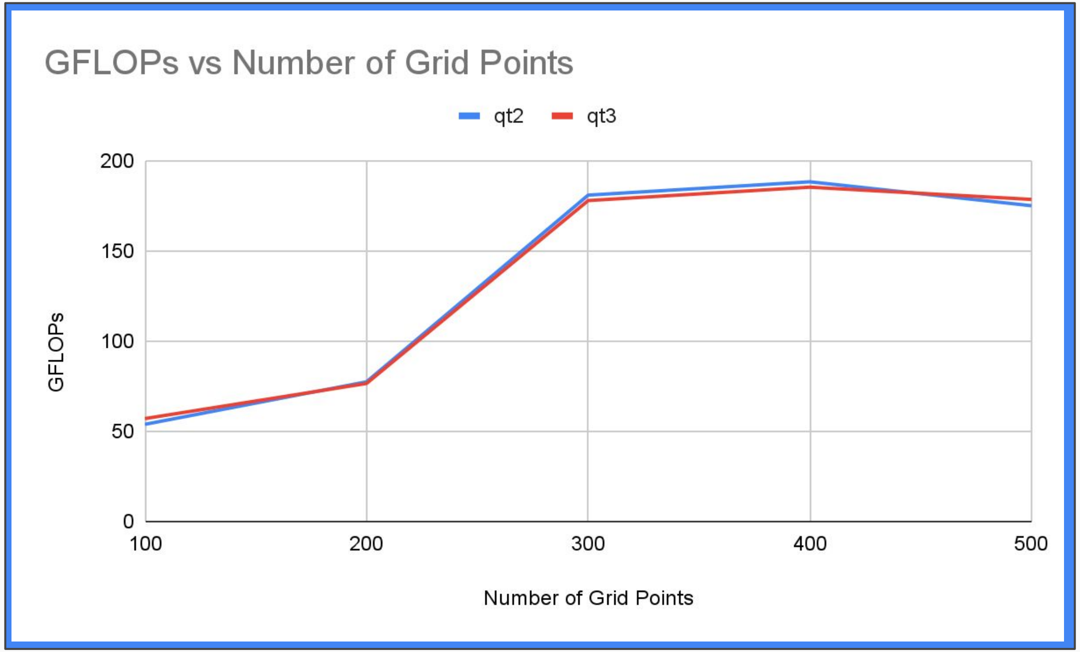
```
Algorithm III
for k = 1,...,n do in parallel
  for m = 1,...,M do in parallel
    Simulate X_k, ε_k

    Find NN-Index i of X_k in Γ_k
    Find NN-Index j of A_k X_k + T_k ε_k in Γ_{k+1}

    atomic increment p_ij^k
    atomic increment p_i^k
  end for in parallel
  Synchronize
  Set in parallel π_ij^k ← p_ij^k / p_i^k,   1 ≤ i,j ≤ N_k
end for in parallel
```

Results

Time for All Algs.



GFLOPs for Tree Quantization Algs.



Sources

https://arxiv.org/abs/1101.3228