Stochastic mesh methods for quadratic hedging with transaction costs

Pierre-A. Tremblay and Pierre L'Ecuyer
DIRO, Université de Montréal, Canada
tremblap@iro.umontreal.ca

MCQMC, Warsaw, August 2010

Hedging problem

Hedging aims to reduce the risk of a derivative by trading in a more liquid security.

We want to compute an optimal dynamic hedging policy in discrete time given a non-tradeable derivative with price process $\{h_k : \ k = 0, \ldots, K\}$.

State of the system at step k : $x_k = (c_k, u_k, S_k) \in \mathbb{R}^{2M+1}$, where $S_k \in \mathbb{R}^M$ is the stock price vector, $u_k \in \mathbb{R}^M$ the stock quantity held and $c_k \in \mathbb{R}$ is the value of the cash account.

Portfolio value : $\Pi_k = c_k + u_k^T S_k - h_k$, for $k = 0, \ldots, K$.

Problem to solve

$$\min_{\mu} J_0^{\mu},$$

where

- μ is a trading policy, i.e. $u_k = \mu_k(x_{k-1})$, and
- J_0^{μ} is a risk measure on the portfolio value Π_k^{μ} under policy μ.

An example

Hedging a call option with strike X on a basket of M stocks, with payoff

$$h_K(S_K) = \max(0, \overline{S_K} - X),$$

where $\overline{S_K} = \prod_{j=1}^{M} S_{K,j}^{1/M}$.

Assume

- $S_k \sim M$-dim GBM process with fixed correlation ρ between stocks.
- we work with discounted price processes $S_{k,j} \Rightarrow e^{-rt_k} S_{k,j}$.
- expectations under a pricing measure $\mathbb{P}$ (so that $h_k = \mathbb{E}_k[h_K]$).

Basic policy : Delta hedging

$$\mu_{k,j}^{DH} = \partial h_k(S_k) / \partial S_{k,j}.$$

Risk function definition

Transaction costs per unit of stock j : $\phi_{k,j} = a + bS_{k,j}$, where $a, b \geq 0$.

Risk function : Let $J_k^\mu = \mathbb{E}_k \left[\sum_{l=k}^{K} g_{l+1}(x_l, \mu_{l+1}, \omega) \right]$, where

$$g_{k+1}(x_k, u_{k+1}, \omega) = (u_{k+1}^T \psi_{k+1} - Z_{k+1})^2 + (u_{k+1} - u_k)^T \Phi_k (u_{k+1} - u_k),$$

for $\psi_{k+1} := S_{k+1} - S_k$, $Z_{k+1} := h_{k+1} - h_k$ and Φ_k a $M \times M$ diagonal matrix with $\Phi_{k,jj} = \phi_{k,j}^2$.

- Linear quadratic (LQ) problem.
- This is a 'gentle' way to include costs, to see how far we can go with a simple case.
- The optimal risk J_k^* will be a known (quadratic) function of stock quantities u_k, but unknown in the stock prices S_k.
- More generally, dependence of J_k^* on u_k would need to be approximated.

Recursive matrix equations for the optimal solution

$$J_k^*(x_k) = u_{k-1}^T \Phi_k A_k u_{k-1} - 2u_{k-1}^T \Phi_k B_k + C_k,$$

where $A_k \in \mathbb{R}^{M\times M}$, $B_k \in \mathbb{R}^M$ and $C_k \in \mathbb{R}$ are defined recursively via

$$\begin{aligned}
A_k &= D_k^{-1}\mathbb{E}_k[\Phi_{k+1}A_{k+1} + \psi_{k+1}\psi_{k+1}^T], \\
B_k &= D_k^{-1}\mathbb{E}_k[\Phi_{k+1}B_{k+1} + \psi_{k+1}Z_{k+1}], \\
C_k &= \mathbb{E}_k[C_{k+1} + Z_{k+1}^2] \\
&\quad -\mathbb{E}_k[\Phi_{k+1}B_{k+1} + \psi_{k+1}Z_{k+1}]^T D_k^{-1}\mathbb{E}_k[\Phi_{k+1}B_{k+1} + \psi_{k+1}Z_{k+1}], \\
D_k &= \mathbb{E}_k[\Phi_{k+1}A_{k+1} + \psi_{k+1}\psi_{k+1}^T] + \Phi_k.
\end{aligned}$$

Optimal policy :

$$\mu_k^*(x_k) = B_k + (\mathbb{I} - A_k)u_{k-1}$$

Notes :

- Dependence of J_k^* on the stock quantities u_k is explicit.
- A_k, B_k, C_k are functions of stock price vector $S_k = (S_{k,j})_{j=1,\ldots,M}$.

Stochastic mesh methods

Try computing A_k, B_k, C_k using a stochastic mesh :

- Simulate N independent price paths $\{S_k^i : k = 0, \ldots, K\}$.
- Associate weights $w_{k,i,j}$ to all pairs of states (S_k^i, S_{k+1}^j).
- Approximate risk functions J_k^* recursively at each state S_k^i using

$$\mathbb{E}_k[G_{k+1}|S_k^i]$$

$$\approx \frac{1}{N}\sum_{j=1}^{N} w_{k,i,j} G_{k+1}(S_{k+1}^j).$$

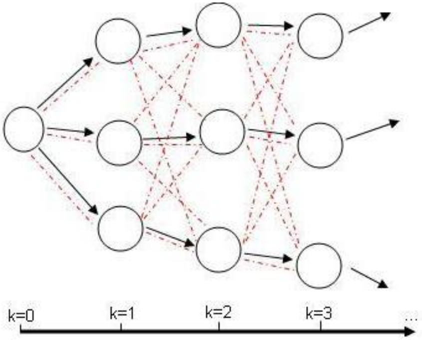

Figure: Mesh illustration : circles are the states for three sample paths, with weights between states indicated by dotted lines.

Stochastic mesh (continued)

Likelihood ratio (LR) weights : $w_{k,i,j} = f_{k,k+1}(S_k^i, S_{k+1}^j)/g_{k+1}(S_{k+1}^j)$.

Average density (AD) weights :

$$w_{k,i,j} = \frac{f_{k,k+1}(S_k^i, S_{k+1}^j)}{\frac{1}{N}\sum_{l=1}^{N} f_{k,k+1}(S_k^l, S_{k+1}^j)}.$$

Stochastic mesh and AD method introduced by Broadie and Glasserman (1997) to price high-dimensional American options.

- Requires a known transition density $f_{k,k+1}$.
- For $u_0 = 0$, $\widehat{C_0}$ yields a low biased estimator for J_0^*.
- Similarly, applying the resulting policy $\hat{\mu}$ over new independent sample paths yields a high biased estimator $\widehat{J_0}$.
- For American option pricing, linear regression methods are also known to do well. Here, the structure is different, but regression could be applied to define element of A_k, B_k and C_k as functions of S_k.

Policy computation time

Computing a policy over a mesh with K time steps and N points per step takes time in $O(KN^2)$. This includes

1. computing the mesh states and weights, and
2. computing the A_k, B_k, C_k (Dynamic Programing step).

In what follows, we decompose the constant C implicit in the $O(KN^2)$ notation into the computation times for the mesh and the DP steps

$$C = C_T(\text{Mesh}) + C_T(\text{DP})$$

and introduce two techniques that reduce each component :

1. Using a single grid, to reduce $C_T(\text{Mesh})$, and
2. Russian roulette, to reduce $C_T(\text{DP})$.

Efficiency of methods will be determined by looking at out-of-sample performance and computing times of mesh based policies.

Using a single grid (SG)

Consider mesh points defined by inversion from a single distribution G

$$S_k^j = G^{-1}(u_j), \text{ where } u_j \sim U(0,1),\ j = 1, \ldots, N, \forall k \geq 0.$$

Here we take $G(S) = F_{0,K}(S)$, the unconditional distribution of the price vector at (the final) step K.

Using LR weights $w_{k,i,j}$ leads to high variance, so replace them by normalized weights

$$w_{k,i,j}^N = \frac{w_{k,i,j}}{\frac{1}{N}\sum_{l=1}^N w_{k,i,l}}.$$

- Assuming $f_{k,i,j}$ is independent of k, $w_{k,i,j} = w_{i,j}, \forall i, j$ and $k > 0$ $\Rightarrow$ can reduce $C_T(\text{Mesh})$ by a factor of K.
- Conditional expectation estimators may have more variance with SG compared to AD method.

Related works : Rust (1997), Boyle, Kolkiewiecz and Tan (2003).

Use of randomized QMC (RQMC)

1. SG method: Take u_j from a low discrepancy (LD) sequence in dim M to generate $Z \sim N(0, I)$. Compute S_K based on $W = \Gamma Z$, where Γ is the eigenvector matrix given by PCA for the covariance matrix Σ.
2. AD method : Take u_j from a LD sequence in dimension $M \times K$ to generate M standard Brownian paths $\{B_k : k = 0, \ldots, K\}$ using a Brownian bridge. Then compute the S_k based on the transformed Brownian paths $W_k = \Gamma B_k$, for Γ as above.
3. Out-of-sample performance : Given a (mesh-based) policy, estimate the performance $\widehat{J_0}$ based on n paths as in point 2.

- Here, u_j taken as N initial points from Sobol sequence.
- To obtain error estimates, a random shift is applied to the u_j for each new mesh construction.
- For out-of-sample performance, a new (randomized) mesh is computed $n_{rep} = 1000$ times, and for each mesh, the policy is evaluated for $n = 1000$ paths.

Single grid performance comparison

$u_0 = 0, S_0 = 10, X = 10, \sigma = 25\%, T = 0.1, \rho = 0.5, a = 0.5\%, b = 1.5\%.$

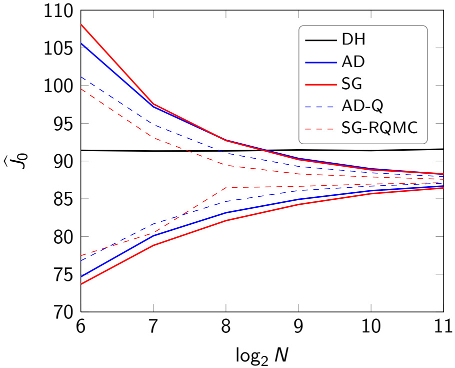

Figure: High and low biased estimators of J_0^* for $M = 3$ and $K = 8$

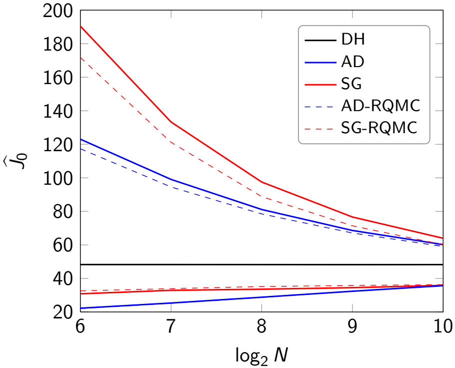

Figure: High and low biased estimators of J_0^* for $M = 5$ and $K = 16$

Error of estimates

Detail of results for the case $M = 5$, $K = 16$, $N = 512$.

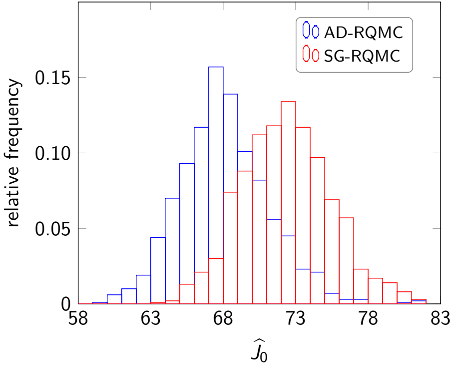

Figure: Histogram of (high biased) out-of-sample estimator $\widehat{J_0}$

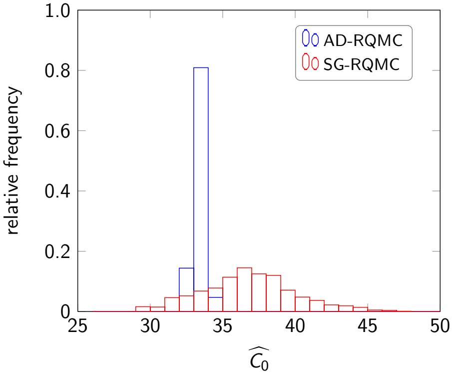

Figure: Histogram of (low biased) inner mesh estimator $\widehat{C_0}$

Russian roulette technique

As M and K increase, most consecutive states (S_k^i, S_{k+1}^j) will tend to be "far apart" $\Rightarrow$ associated weight $w_{k,i,j}$ will be small.

We would like to discart negligible weights to reduce $C_T(\text{DP})$.

Russian roulette : Fix a threshold $\gamma > 0$. If $w_{k,i,j} < \gamma$, set

- $w_{k,i,j} = 0$ with probability $1 - w_{k,i,j}/\gamma$,
- $w_{k,i,j} = \gamma$ with probability $w_{k,i,j}/\gamma$.

Notes :

1. Does not add bias to mesh based conditional expectations..
2. ...But variance is increased, depending on γ.

Ideally, we would like less weights to be rejected, since they take time to compute!

Efficiency comparison of mesh constructions

Out-of-sample estimates $\widehat{J}_0$ of J_0^* as a function of $\log_2 C$, where $C = C_T(\text{Mesh}) + C_T(\text{DP})$ (in sec.), for various number of points N.

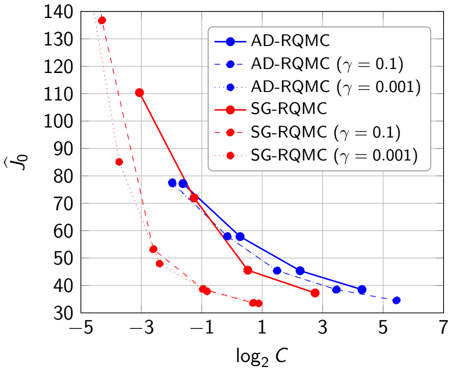

Figure: $M = 3$ and $K = 64$

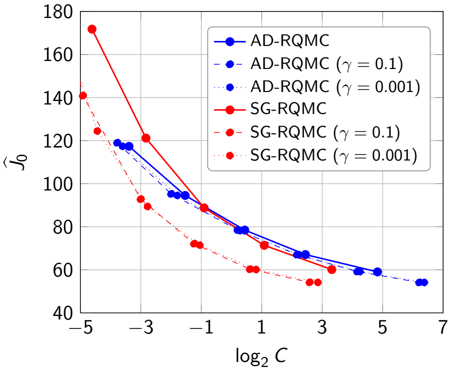

Figure: $M = 5$ and $K = 16$

Concluding remarks

Summary :

- Stochastic mesh methods can be applied to solve a (simplified) quadratic hedging problem with costs.
- The policy computation time can be reduced substantially when using both 1) a single grid and 2) a Russian roulette procedure.
- These methods increase the variance and bias of mesh based policies, especially for small N and larger dimensions M and K, but we do observe efficiency gains for higher N.