Multi-level Algorithms for Infinite-dimensional Integration on $\mathbb{R}^N$

Ben Niu

Department of Applied Mathematics
Illinois Institute of Technology
Email: hickernell@iit.edu, Web: www.math.iit.edu

Joint work with Fred Hickernell (IIT), Thomas Müller-Gronbach (U Passau) and Klaus Ritter (TU Kaiserslautern).
This work is supported by grant NSF-DMS-0713848

August 19, 2010
Pricing continuously monitored options is well modeled as the expectation of a functional of a continuous time (infinite dimensional) Brownian motion.

Shrewdly defining the Hilbert space of the payoff functional splits the error into the finite approximation error, $O(d^{-q})$, and the finite sampling error, $O(n^{-p})$.

The multilevel Monte Carlo algorithm with low discrepancy points is used to improve the convergence rate of the worst case error.

The goal is to minimize the worst case error as a function of each level’s sample size $n_l$ and truncated dimension $d_l$ given the computational cost $N = n_1d_1 + \cdots + n_Ld_L$. 
Option Pricing

\[
\text{option price} = E[\text{payoff}(S(\cdot))] \\
\text{E.g., payoff} = \max \left( \frac{1}{T} \int_0^T S(t) \, dt - K, 0 \right) e^{-rT} \quad \text{arithmetic mean Asian call}
\]

\[
r = \text{risk-less interest rate} \quad K = \text{strike price} \quad T = \text{time to expiry}
\]
Option Pricing

\[
\text{option price} = E[\text{payoff}(S(\cdot))] = E[g(\sqrt{B(\cdot)})] \\
\text{E.g., payoff} = \max \left( \frac{1}{T} \int_0^T S(t) \, dt - K, 0 \right) e^{-rT} \quad \text{arithmetic mean Asian call}
\]

\[
r = \text{risk-less interest rate} \quad K = \text{strike price} \quad T = \text{time to expiry}
\]

\[
\text{E.g., } S(t) = S(0)e^{(r-\sigma^2/2)t+\sigma B(t)} \quad \text{geometric Brownian motion}
\]

\[
\sigma = \text{asset volatility}
\]

\[
B(t_j; X_{1:j}) = B(t_{j-1}; X_{1:j-1}) + \sqrt{t_j - t_{j-1}}X_j. \quad \text{Time Differencing}
\]

\[
j = 1, \ldots, s, \quad X_1, \ldots, X_s \text{ i.i.d. } N(0, 1), \quad 0 = t_0 < t_1 < \cdots < t_s \leq T.
\]
Option Pricing

$$\text{option price} = E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(\underbrace{X_1, X_2, \ldots}_{\text{Gaussian random variables}})]$$

$$B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t),$$

$j$ large means $e_j$ is high frequency, small magnitude. *Karhunen-Loève expansion*:

$$e_j(t) = \sqrt{2T} \frac{\sin \left(\left(j - \frac{1}{2}\right) \pi t / T\right)}{\left(j - \frac{1}{2}\right) \pi}$$
Option Pricing

option price = $E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(V, X_1, X_2, \ldots)]$

$B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t),$ 

$j$ large means $e_j$ is high frequency, small magnitude. **Karhunen-Loève expansion**:

$e_j(t) = \sqrt{2T} \frac{\sin \left( (j - \frac{1}{2}) \pi t / T \right)}{(j - \frac{1}{2}) \pi}$
Option Pricing

option price = $E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(\underbrace{X_1, X_2, \ldots}_{\text{Gaussian random variables}})]$

$B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t)$,

$j$ large means $e_j$ is high frequency, small magnitude. Karhunen-Loève expansion:

$e_j(t) = \sqrt{2T} \frac{\sin \left( (j - \frac{1}{2}) \pi t / T \right)}{(j - \frac{1}{2}) \pi}$
Option Pricing

\[ \text{option price} = E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(\underbrace{X_1, X_2, \ldots}_n)] \]

\[ B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t), \]

\( j \) large means \( e_j \) is high frequency, small magnitude. Karhunen-Loève expansion:

\[ e_j(t) = \sqrt{2T} \frac{\sin((j - \frac{1}{2}) \pi t/T)}{(j - \frac{1}{2}) \pi} \]
Introduction

Multi-Level Algorithm

Numerical Experiments

Conclusion

Summary

Option Pricing

option price = \( E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(\begin{array}{c} X_1, X_2, \ldots \end{array})] \)

\[ B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t), \]

\( j \) large means \( e_j \) is high frequency, small magnitude. Karhunen-Loève expansion:

\[ e_j(t) = \sqrt{2T} \frac{\sin \left( (j - \frac{1}{2}) \frac{\pi t}{T} \right)}{(j - \frac{1}{2}) \pi} \]
option price = $E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(\underbrace{X_1, X_2, \ldots}_{\text{Gaussian random variables}})]$

$$B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t),$$

$j$ large means $e_j$ is high frequency, small magnitude. Karhunen-Loève expansion:

$$e_j(t) = \sqrt{2T} \frac{\sin \left( \left( j - \frac{1}{2} \right) \pi t / T \right)}{\left( j - \frac{1}{2} \right) \pi}$$
Option Pricing

$$\text{option price} = E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(\overset{\sim}{X_1, X_2, \ldots})]$$

$$B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t),$$

$j$ large means $e_j$ is high frequency, small magnitude. Karhunen-Loève expansion:

$$e_j(t) = \sqrt{2T} \frac{\sin \left( \frac{(j - 1/2) \pi t}{T} \right)}{(j - 1/2) \pi}$$
Option Pricing

\[
\text{option price} = E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(X_1, X_2, \ldots)]
\]

\[
B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t),
\]

\(j\) large means \(e_j\) is high frequency, small magnitude. Karhunen-Loève expansion:

\[
e_j(t) = \sqrt{2T} \frac{\sin \left( (j - \frac{1}{2}) \frac{\pi t}{T} \right)}{(j - \frac{1}{2}) \pi}
\]
Option Pricing

option price = \( E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(X_1, X_2, \ldots)] \)

\[
B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t),
\]

\( j \) large means \( e_j \) is high frequency, small magnitude. Karhunen-Loève expansion:

\[
e_j(t) = \sqrt{2T} \frac{\sin \left( \left( j - \frac{1}{2} \right) \pi t / T \right)}{\left( j - \frac{1}{2} \right) \pi}
\]
Option Pricing

\[
\text{option price} = E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(X_1, X_2, \ldots)]
\]

\[
B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t),
\]

\( j \) large means \( e_j \) is high frequency, small magnitude. Karhunen-Loève expansion:

\[
e_j(t) = \sqrt{2T} \frac{\sin \left( \left( j - \frac{1}{2} \right) \pi t / T \right)}{\left( j - \frac{1}{2} \right) \pi}
\]
Option Pricing

\[
\text{option price} = E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(\underbrace{X_1, X_2, \ldots}_{\text{Gaussian random variables}})]
\]

\[
B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t),
\]

\(j\) large means \(e_j\) is high frequency, small magnitude. Karhunen-Loève expansion:

\[
e_j(t) = \sqrt{2T} \frac{\sin \left( \left( j - \frac{1}{2} \right) \pi t / T \right)}{\left( j - \frac{1}{2} \right) \pi}
\]
**Option Pricing**

The option price is given by:

\[ \text{option price} = E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(X_1, X_2, \ldots)] \]

Where:

\[
B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t),
\]

Here, \( j \) large means \( e_j \) is high frequency, small magnitude. **Brownian bridge:**

\[
e_1(t) = \frac{t}{\sqrt{T}}
\]

\[
e_{j+1}(t) = \sqrt{\frac{T}{2m_j+2}} \text{hat} \left( \frac{2m_j+1(t-t_j)}{T} \right),
\]

where \( t_j/T = 1 - \text{van der Corput}_2(j) \) and \( m_j = \lfloor \log_2(t_j) \rfloor \).
Option Pricing

option price = $E[\text{payoff}(S(\cdot))]) = E[g(B(\cdot))] = E[f(\ X_1, X_2, \ldots\ )]$\\

\[ B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t), \]

$j$ large means $e_j$ is high frequency, small magnitude. Brownian bridge:

\[ e_1(t) = \frac{t}{\sqrt{T}} \]

\[ e_{j+1}(t) = \sqrt{\frac{T}{2m_j+2 \text{ hat}}} \left( \frac{2^{m_{j+1}}(t - t_j)}{T} \right), \]

where $t_j/T = 1 - \text{van der Corput}_2(j)$ and $m_j = \lfloor \log_2(t_j) \rfloor$. 

```
d = 2
```

![Graph](image-url)
Option Pricing

\[
\text{option price} = E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(\underbrace{X_1, X_2, \ldots}_{\text{Gaussian random variables}})]
\]

\[
B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t),
\]

\(j\) large means \(e_j\) is high frequency, small magnitude. Brownian bridge:

\[
e_1(t) = \frac{t}{\sqrt{T}}
\]

\[
e_{j+1}(t) = \sqrt{\frac{T}{2m_j+2}} \text{hat} \left( \frac{2m_j+1(t-t_j)}{T} \right),
\]

where \(t_j/T = 1 - \text{van der Corput}_2(j)\) and \(m_j = \lfloor \log_2(t_j) \rfloor\).
Option Pricing

\[
\text{option price } = E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(\hat{X}_1, \hat{X}_2, \ldots)]
\]

\[
B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t),
\]

\(j\) large means \(e_j\) is high frequency, small magnitude. Brownian bridge:

\[
e_1(t) = \frac{t}{\sqrt{T}}
\]

\[
e_{j+1}(t) = \sqrt{\frac{T}{2m_j+2}} \hat{\text{van der Corput}_2(j)} \left( \frac{2m_j+1(t-t_j)}{T} \right),
\]

where \(t_j/T = 1 - \text{van der Corput}_2(j)\) and \(m_j = \lfloor \log_2(t_j) \rfloor\).
Option Pricing

\[
\text{option price } = E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(X_1, X_2, \ldots)]
\]

\[
B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t),
\]

\(j\) large means \(e_j\) is high frequency, small magnitude. Brownian bridge:

\[
e_1(t) = \frac{t}{\sqrt{T}}
\]

\[
e_{j+1}(t) = \sqrt{\frac{T}{2m_{j+2}}} \hat{\text{vander Corput}_2}(\frac{2m_{j+1}(t - t_j)}{T}),
\]

where \(t_j/T = 1 - \text{van der Corput}_2(j)\) and \(m_j = \lfloor \log_2(t_j) \rfloor\).
Option Pricing

option price = $E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(X_1, X_2, \ldots)]$

$B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t)$,

$j$ large means $e_j$ is high frequency, small magnitude. Brownian bridge:

$e_1(t) = \frac{t}{\sqrt{T}}$

$e_{j+1}(t) = \sqrt{\frac{T}{2m_j+2}} \hat{\text{van der Corput}}_2(j) \left( \frac{2m_j+1(t - t_j)}{T} \right)$,

where $t_j/T = 1 - \text{van der Corput}_2(j)$ and $m_j = \lfloor \log_2(t_j) \rfloor$. 
Option Pricing

\[
\text{option price} = E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(X_1, X_2, \ldots)]
\]

\[
B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t),
\]

\(j\) large means \(e_j\) is high frequency, small magnitude. Brownian bridge:

\[
e_1(t) = \frac{t}{\sqrt{T}}
\]

\[
e_{j+1}(t) = \sqrt{\frac{T}{2m_j+2}} \text{hat} \left( \frac{2m_j+1(t - t_j)}{T} \right),
\]

where \(t_j/T = 1 - \text{van der Corput}_2(j)\) and \(m_j = \lceil \log_2(t_j) \rceil\).
Option Pricing

option price = $E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(\underbrace{X_1, X_2, \ldots}_{\text{Gaussian random variables}})]$

$B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t),$

$j$ large means $e_j$ is high frequency, small magnitude. Brownian bridge:

$e_1(t) = \frac{t}{\sqrt{T}}$

$e_{j+1}(t) = \sqrt{\frac{T}{2m_j+2}} \text{hat} \left( \frac{2m_j+1(t-t_j)}{T} \right),$

where $t_j/T = 1 - \text{van der Corput}_2(j)$ and $m_j = \lfloor \log_2(t_j) \rfloor.$
Option Pricing

\[
\text{option price} = E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(\underbrace{X_1, X_2, \ldots}_d)]
\]

\[
B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t),
\]

\(j\) large means \(e_j\) is high frequency, small magnitude. Brownian bridge:

\[
e_1(t) = \frac{t}{\sqrt{T}}
\]

\[
e_{j+1}(t) = \sqrt{\frac{T}{2m_j+2}} \hat{\text{van der Corput}}_2(j) \left(2^{m_j+1} (t - t_j)\right),
\]

where \(t_j/T = 1 - \text{van der Corput}_2(j)\) and \(m_j = \lfloor \log_2(t_j) \rfloor\). 

\[
\begin{array}{c}
0 & 0.2 & 0.4 & 0.6 & 0.8 & 1 \\
-1.5 & -1 & -0.5 & 0 & 0.5 & 1 & 1.5
\end{array}
\]

\(d = 256\)
Option Pricing

\[ \text{option price} = E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(\underbrace{X_1, X_2, \ldots}_d)] \]

\[ B(t) = \sum_{j=1}^{\infty} X_je_j(t) \approx \sum_{j=1}^{d} x_je_j(t), \]

\( j \) large means \( e_j \) is high frequency, small magnitude. Brownian bridge:

\[ e_1(t) = \frac{t}{\sqrt{T}} \]

\[ e_{j+1}(t) = \sqrt{\frac{T}{2m_{j+2}}} \hat{\text{van der Corput}}_2(j) \left( \frac{2m_{j+1}(t - t_j)}{T} \right), \]

where \( t_j/T = 1 - \text{van der Corput}_2(j) \) and \( m_j = \lfloor \log_2(t_j) \rfloor \).
Option Pricing

\[
\text{option price} = E[\text{payoff}(S(\cdot))] = E[g(B(\cdot))] = E[f(\underbrace{X_1, X_2, \ldots})]
\]

\[
B(t) = \sum_{j=1}^{\infty} X_j e_j(t) \approx \sum_{j=1}^{d} x_j e_j(t),
\]

\(j\) large means \(e_j\) is high frequency, small magnitude. Brownian bridge:

\[
e_1(t) = \frac{t}{\sqrt{T}}
\]

\[
e_{j+1}(t) = \sqrt{\frac{T}{2m_j+2}} \hat{\text{van der Corput}_2(j)} \left( \frac{2m_j+1(t-t_j)}{T} \right),
\]

where \(t_j/T = 1 - \text{van der Corput}_2(j)\) and \(m_j = \lfloor \log_2(t_j) \rfloor\).
The Problem

Efficiently approximate

\[
\int_{\mathbb{R}^d} f(x) \rho(x) \, dx
\]

using function values \( f(x_1), \ldots, f(x_n) \).

Recent work on this problem has been done by Hickernell and Wang (2002); Creutzig et al. (2009); Kuo et al. (2010); Niu et al. (2010); Niu and Hickernell (2010); Hickernell et al. (2010); Gnewuch (2010); Plaskota and Wasilkowski (2010).
Multilevel Monte Carlo Simulation

**Single Level: A unique truncated dimension** $d$.

$$
\mu = E[f(X_1, X_2, \ldots)] = \mu_d + (\mu - \mu_d),
$$

$$
\mu_d = E[f(X_1, \ldots, X_d, 0, \ldots)] \approx \hat{\mu} := \frac{1}{n} \sum_{i=1}^{n} f(x_{i1}, \ldots, x_{id}, 0, \ldots),
$$

$$
\mu - \mu_d \approx 0.
$$

The computational cost is

$$
N = nd.
$$
Multilevel Monte Carlo Simulation

Multilevel: An increasing sequence of truncated dimensions \( d_1 < \cdots < d_L \).

\[
\mu = \sum_{l=1}^{L+1} E \left[ f(X_1, \ldots, X_{d_l}, 0, \ldots) - f(X_1, \ldots, X_{d_{l-1}}, 0, \ldots) \right]
\]

\[
\hat{\mu} = \sum_{l=1}^{L} \frac{1}{n_l} \sum_{i=1}^{n_l} \left[ f(x_{i,1:d_l}^{(l)}, 0, \ldots) - f(x_{i,1:d_{l-1}}^{(l)}, 0, \ldots) \right]
\]

- Conveniently set \( d_0 = 0 \) and \( d_{L+1} = \infty \).
- The level \( L + 1 \) is approximated by 0 for large \( L \).
- The computations cost of \( \hat{\mu} \) is proportional to \( N = \sum_{l=1}^{L} n_l d_l \).

See Heinrich (2001); Giles (2008); Giles and Waterhouse (2009) for previous work using multilevel methods.
Hilbert Space of Functionals

The functional $f : \mathcal{X} \to \mathbb{R}$ is assumed to reside in a Hilbert space $\mathcal{H}(K)$ with reproducing kernel

$$K(x, y) = \prod_{j=1}^{\infty} [1 + \gamma_j K_1(x_j, y_j)], \quad K_1(0, 0) = 0 \ \forall x,$$

$$\gamma_1 \geq \gamma_2 \geq \cdots \geq 0, \quad \sum_{j=d}^{\infty} \gamma_j = O(d^{-2q}),$$

$$\mathcal{X} = \{ x \in \mathbb{R}^N : K(x, x) < \infty \} \supseteq \ell_\infty.$$
Worst Case Error Bound (tight)

\[
\sup_{\|f\|_{\mathcal{H}(K)} \leq 1} |\mu - \hat{\mu}|^2 = \prod_{j=1}^{d_L} [1 + \gamma_j \alpha] \left[ \prod_{j=d_L+1}^{\infty} [1 + \gamma_j \alpha] - 1 \right]
\]

\[
\leq d_L^{2q} \text{worst-bias}^2 (d_L; K) \leq d_L^{-2q}
\]

\[
+ \sum_{l=1}^{L} \left[ D^2 (\{ x_i^{(l)} \}_{i=1}^{m_l}; K^{(d_l)}) - D^2 (\{ x_i^{(l)} \}_{i=1}^{m_l}; K^{(d_l-1)}) \right],
\]

where \( \alpha = \int_{\mathbb{R}} K_1(x, y) \rho_1(x) \rho_1(y) \, dx \, dy. \)

The goal is to minimize the worst-case error given the cost:

\[
N = \sum_{l=1}^{L} n_l d_l.
\]
Worst Case Error Bound (loose)

Introduce another sequence of coordinate weights, $\gamma' = (\gamma'_1, \gamma'_2, \ldots)$ satisfying $\gamma_j \leq \gamma'_j$ and $\gamma'_j \asymp j^{2(q-q')} \gamma_j$. The corresponding kernel function $K'$ is defined by using $\gamma'_j$ instead of $\gamma_j$.

Strong tractability results show that $D \left( P; K'(d) \right) \preceq n^{-p}$, where $p$ depends on $q'$.

Theorem

Consider $p, q, q' > 0$ with $q \geq q'$ and $p + q' \neq q$. Let the computational cost of the multilevel algorithm is defined as $N = \sum_{l=1}^{L} n_l d_l$, then (Niu et al., 2010)

$$\sup_{\|f\|_{\mathcal{H}(K)} \leq 1} |\mu - \hat{\mu}| \leq N^{-p} \min\left(1, \frac{q}{p+q'}\right).$$
Two specific examples

Find an optimal $q'$ such that $\sup_{\|f\|_{\mathcal{H}(K)} \leq 1} |\mu - \hat{\mu}| \leq N^{-\tau}$, where

$$\tau = \max_{q'} p \min \left(1, \frac{q}{p + q'} \right) \leq \min (p, q) .$$

**Rank-1 Lattice**

**Niederreiter (T,d)-net**

Not as good as Plaskota and Wasilkowski (2010).
Algorithm

- For the multilevel algorithm,

\[
\hat{\mu} = \sum_{l=1}^{L} \hat{\mu}_l = \sum_{l=1}^{L} \frac{1}{n_l} \sum_{i=1}^{n_l} \left[ f(x^{(l)}_{i,1:d_l}, 0, \ldots) - f(x^{(l)}_{i,1:d_{l-1}}, 0, \ldots) \right].
\]

- Denote \( S_j = \sum_{i=1}^{m_j} h(x_{i+m_j,1:d_l}) \), define \( m_j = m_0 2^j \), \( j = 0, 1, 2, \ldots \) and \( n_l = m_J_l \), then

\[
\hat{\mu}_l = \frac{1}{m_1} S_1 \quad \text{(enough?)}
\]
Algorithm

For the multilevel algorithm,

$$\hat{\mu} = \sum_{l=1}^{L} \hat{\mu}_l = \sum_{l=1}^{L} \frac{1}{n_l} \sum_{i=1}^{n_l} \left[ f(x_{i,1:d_l}, 0, \ldots) - f(x_{i,1:d_{l-1}}, 0, \ldots) \right].$$

Denote $S_j = \sum_{i=1}^{m_j} h(x_{i+m_j,1:d_l})$, define $m_j = m_0 2^j$, $j = 0, 1, 2, \ldots$ and $n_l = m_J_l$, then

$$\hat{\mu}_l = \frac{1}{m_1} S_1 \quad (\text{enough?})$$

$$+ \frac{1}{m_2} (S_2 - S_1) \quad (\text{enough?})$$
Algorithm

For the multilevel algorithm,

$$
\hat{\mu} = \sum_{l=1}^{L} \hat{\mu}_l = \sum_{l=1}^{L} \frac{1}{n_l} \sum_{i=1}^{n_l} \left[ f(\mathbf{x}_i^{(l)}, 0, \ldots) - f(\mathbf{x}_i^{(l)}, 0, \ldots) \right].
$$

Denote $S_j = \sum_{i=1}^{m_j} h(\mathbf{x}_{i+m_j}, 1:d_l)$, define $m_j = m_0 2^j$, $j = 0, 1, 2, \ldots$ and $n_l = m_{J_l}$, then

$$
\hat{\mu}_l = \frac{1}{m_1} S_1 \quad (\text{enough?})
+ \frac{1}{m_2} (S_2 - S_1) \quad (\text{enough?})
+ \cdots
+ \frac{1}{m_{J_l}} (S_{J_l} - S_{J_l-1}).
$$
Measure the Error: Replications

Run $R$ replications at each level. For each level $l$

\[
\hat{\mu}_l = \frac{1}{R} \sum_{r=1}^{R} \frac{1}{n_l} \sum_{i=1}^{n_l} h(x^{(r)}_{i,1:d_l}).
\]

The estimated error at each level $l$ is computed as

\[
\hat{\text{err}}_{l,n_l} = 1.96 \sqrt{\frac{1}{R} \sum_{r=1}^{R} (\hat{\mu}_l - \hat{\mu}_{l,r})^2}
\]

The algorithm dynamically searches for the optimal levels $L$. Given the tolerance $\varepsilon$, the algorithm will stop when

\[
\hat{\text{err}}_{1,n_1} + \hat{\text{err}}_{2,n_2} + \cdots + \hat{\text{err}}_{L,n_L} + \hat{\mu}_L < \varepsilon.
\]
Sechmatic for for the Multilevel Algorithm.
Geometric Mean Asian Call Option

\[ \text{price} = E[\text{payoff}(B(\cdot))], \text{ where } B(t) \text{ is a Brownian motion} \]

\[ \text{payoff} = \max \left( \exp \left( \frac{1}{T} \int_0^T \log \left( S(0) e^{(r-\sigma^2/2)t+\sigma B(t)} \right) \, dt \right) - K, 0 \right) e^{-rT} \]

for the geometric mean Asian option

\[ r = \text{risk-less interest rate} \quad \sigma = \text{asset volatility} \]

\[ K = \text{strike price} \quad T = \text{time to expiry} \]

A closed form for the option price allows us to compute error exactly.

\[ \hat{\text{price}} = \frac{1}{n} \sum_{i=1}^{n} \text{payoff}(\hat{B}_i(\cdot)) \]

\[ \text{price} = \$7.09 \quad S(0) = \text{initial asset price} = \$100 \]
\[ r = \text{risk-less interest rate} = 3\% \quad \sigma = \text{asset volatility} = 30\% \]
\[ K = \text{strike price} = \$100 \quad T = \text{time to expiry} = 1 \text{ year} \]
Geometrical Mean Asian Call Option

Figure: $R = 30$, $m_0 = 32$, $d_l = 2^l$, $l = 0, 1, 2, \ldots$
Barrier and Lookback Option

\[
payoff_{\text{Barrier}} = \max(S(T) - K, 0) \text{ if } \max_{0 \leq t \leq T} S(t) > B, \quad \text{Up and In},
\]

\[
payoff_{\text{Lookback}} = \max(S(t) - \min_{0 \leq t \leq T} S(t), 0),
\]

\[
\begin{align*}
& r = \text{risk-less interest rate} & \sigma = \text{asset volatility} \\
& K = \text{strike price} & T = \text{time to expiry} & B = \text{Barrier}
\end{align*}
\]

There are no closed form for both options

\[
\hat{\text{price}} = \frac{1}{n} \sum_{i=1}^{n} \text{payoff}(\hat{B}_i(\cdot))
\]

\[
\begin{align*}
S(0) &= \text{initial asset price} = $100 & T &= \text{time to expiry} = 1 \text{ year} \\
r &= \text{risk-less interest rate} = 5\% & \sigma &= \text{asset volatility} = 30\% \\
K &= \text{strike price} = $120 & B &= \text{Barrier} = $110
\end{align*}
\]
Barrier Option

Figure: \( R = 30, m_0 = 32, d_l = 2^{\frac{l}{2}}, l = 0, 1, 2, \ldots \)
Lookback Option

Figure: $R = 30, m_0 = 32, d_l = 2^{\frac{l}{2}}, l = 0, 1, 2, \ldots$
Summary and Future Work

Summary

- Pricing path dependent options requires the expectation of a payoff functional of a continuous time (infinite dimensional) Brownian motion.
- The Karhunen-Loéve expansion or the Brownian bridge focuses sampling effort on low frequency components of the Brownian motion.
- A multilevel (in dimension) algorithm computes a numerical approximation to the difference between $d_l$ and $d_{l-1}$ dimensional approximation to the functional. Can get nearly ideal rates of convergence for worst case error.

Future work

- Search for an appropriate error estimate to quantify the approximation error.
- Implement more numerical experiments in computational finance.
## Food for Thought

To approximate \( \int_{\mathbb{R}^d} f(x) \rho(x) \, dx \) with error \( \leq N^{-\tau} \) for known p.d.f. you need . . .

<table>
<thead>
<tr>
<th>( d )</th>
<th>( \tau )</th>
<th>( \var(f) &lt; \infty ), i.i.d. random sample points</th>
</tr>
</thead>
<tbody>
<tr>
<td>fixed</td>
<td>( \frac{1}{2} )</td>
<td>var(( f )) &lt; ( \infty ), i.i.d. random sample points</td>
</tr>
</tbody>
</table>
Food for Thought

To approximate \[ \int_{\mathbb{R}^d} f(x) \rho(x) \, dx \] with error \( \leq N^{-\tau} \) you need...

<table>
<thead>
<tr>
<th>Condition</th>
<th>Error Rate</th>
<th>Additional Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>fixed (\tau)</td>
<td>(\frac{1}{2})</td>
<td>(\text{var}(f) &lt; \infty), i.i.d. random sample points</td>
</tr>
<tr>
<td>small (\tau)</td>
<td>(1 - \epsilon)</td>
<td>(|f'|_2 &lt; \infty), low discrepancy, (D), sample points</td>
</tr>
</tbody>
</table>
### Food for Thought

To approximate \( \int_{\mathbb{R}^d} f(x) \rho(x) \, dx \) with error \( \leq N^{-\tau} \) for known \( \rho \) you need . . .

<table>
<thead>
<tr>
<th>( d )</th>
<th>( \tau )</th>
<th>( \text{you need} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>fixed</td>
<td>( \frac{1}{2} )</td>
<td>( \var(f) &lt; \infty ), i.i.d. random sample points</td>
</tr>
<tr>
<td>small</td>
<td>( 1 - \epsilon )</td>
<td>( | f' |_2 &lt; \infty ) low discrepancy, ( D ), sample points</td>
</tr>
<tr>
<td>arbitrary</td>
<td>( \frac{1}{2} )</td>
<td>( \sup_d \var(f) &lt; \infty ), i.i.d. random sample points</td>
</tr>
</tbody>
</table>
## Food for Thought

To approximate \( \int_{\mathbb{R}^d} f(x) \rho(x) \, dx \) with error \( \leq N^{-\tau} \) for a known p.d.f. you need . . .

<table>
<thead>
<tr>
<th>( d )</th>
<th>with ( \tau )</th>
<th>you need . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>fixed</td>
<td>( \frac{1}{2} )</td>
<td>( \text{var}(f) &lt; \infty ), i.i.d. random sample points</td>
</tr>
<tr>
<td>small</td>
<td>( 1 - \epsilon )</td>
<td>( |f'|_2 &lt; \infty ) low discrepancy, ( D ), sample points</td>
</tr>
<tr>
<td>arbitrary</td>
<td>( \frac{1}{2} )</td>
<td>( \sup_d \text{var}(f) &lt; \infty ), i.i.d. random sample points</td>
</tr>
<tr>
<td>arbitrary</td>
<td>( 1 - \epsilon )</td>
<td>( \sup_d |f'|<em>{2,\gamma} &lt; \infty ) low weighted discrepancy, ( D</em>\gamma ), sample points</td>
</tr>
</tbody>
</table>
### Food for Thought

To approximate
\[
\int_{\mathbb{R}^d} f(x) \rho(x) \, dx
\]
with error \( \leq N^{-\tau} \)

<table>
<thead>
<tr>
<th>Case</th>
<th>Parameter</th>
<th>Condition</th>
<th>Sample Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed</td>
<td>( \frac{1}{2} )</td>
<td>( \text{var}(f) &lt; \infty ), i.i.d. random sample points</td>
<td></td>
</tr>
<tr>
<td>Small</td>
<td>( 1 - \epsilon )</td>
<td>( |f'|_2 &lt; \infty ), low discrepancy, ( D ), sample points</td>
<td></td>
</tr>
<tr>
<td>Arbitrary</td>
<td>( \frac{1}{2} )</td>
<td>( \sup_d \text{var}(f) &lt; \infty ), i.i.d. random sample points</td>
<td></td>
</tr>
<tr>
<td>Arbitrary</td>
<td>( 1 - \epsilon )</td>
<td>( \sup_d |f'|<em>{2,\gamma} &lt; \infty ), low weighted discrepancy, ( D</em>\gamma ), sample points</td>
<td></td>
</tr>
<tr>
<td>( \infty )</td>
<td>( \min(1, q) - \epsilon )</td>
<td>( |f'|<em>{2,\gamma} &lt; \infty ), low weighted discrepancy, ( D</em>\gamma ), sample points</td>
<td></td>
</tr>
</tbody>
</table>

For values of \( \gamma_j \) or their decay rate needed to design the multi-level algorithm, you need . . .

\[\text{nben@iit.edu} \quad \text{MCQMC 2010}\]
Thank you!
References I


