Parallel Multilevel Monte Carlo Simulation

Thomas Gerstner
Mathematisches Institut
Goethe-Universität Frankfurt am Main

Conference and Training Event on Computational Methods and Technologies in Finance
Tampere, Finland
May 13-17, 2013
Outline

1. Monte Carlo
   - Approach
   - Error Estimates
   - Examples

2. Multilevel Monte Carlo
   - Approach
   - Error Estimates
   - Examples

3. Adaptive Multilevel Monte Carlo
   - Approach
   - Examples

4. Parallel Multilevel Monte Carlo
   - Approach
   - Algorithm
   - Numerical Results
Outline

1. Monte Carlo
   - Approach
   - Error Estimates
   - Examples

2. Multilevel Monte Carlo
   - Approach
   - Error Estimates
   - Examples

3. Adaptive Multilevel Monte Carlo
   - Approach
   - Examples

4. Parallel Multilevel Monte Carlo
   - Approach
   - Algorithm
   - Numerical Results
**Option pricing**

**Model:** Black-Scholes

\[ dS(t) = \mu S(t)dt + \sigma S(t)dW(t) \]

**Euler-Maruyama discretization:**

\[ \hat{S}(t_{j+1}) = \hat{S}(t_j) + \mu \hat{S}(t_j)h + \sigma \hat{S}(t_j)z_j \quad \text{with} \quad z_j \sim N(0, h) \]

**Martingale approach:**

\[ V(S, 0) = e^{-rT} E^* [V(S, T)] \]

**Monte Carlo simulation:**

\[ \hat{V}(S, 0) = e^{-rT} \frac{1}{N} \sum_{i=1}^{N} \hat{V}(\{\hat{S}^{(i)}(t_1), \ldots, \hat{S}^{(i)}(t_d)\}, T) \]

where \( \hat{V} \) is the discretized payoff, e.g. for a lookback option

\[ \hat{V}(\{\hat{S}(t_1), \ldots \hat{S}(t_d)\}, T) = \hat{S}(t_d) - \min_{1 \leq j \leq d} \hat{S}(t_j) \]
Discretization Error

Shortcut notation:

\[
Y = E[f(S)] \quad \text{and} \quad \hat{Y} = \frac{1}{N} \sum_{i=1}^{N} f(\hat{S}(i))
\]

Mean square error:

\[
MSE = E[|E[f(S)] - \hat{E}[f(\hat{S})]|^2] = (E[\hat{Y} - Y])^2 + Var[\hat{Y}]
= O(h^{2\alpha} + N^{-1})
\]

if the approximation \( f(\hat{S}) \) converges with weak order \( \alpha \)

\[
\max_{1 \leq j \leq d} |E[f(\hat{S}(t_j))] - E[f(S(t_j))]| \leq ch^\alpha.
\]
Error balancing: To achieve an RMSE of $O(\epsilon)$ it is necessary to select

$$N = O(\epsilon^{-2}) \quad \text{and} \quad h = O(\epsilon^{1/\alpha})$$

The corresponding cost $C$ is then

$$C = N \cdot d = O(\epsilon^{-2-1/\alpha}).$$

Solving for $\epsilon$ yields the optimal rate of convergence

$$RMSE = O(\epsilon) = O(C^{-\alpha/(2\alpha+1)}).$$

Optimal refinement rule:

$$d \to 2d \quad \Rightarrow \quad N \to 4\alpha N$$
Asian Option

Parameters: $S_0 = 1$, $r = 0.05$, $\sigma = 0.2$, $K = 1$, $T = 1$, $d_1 = 1$, $N_1 = 5$, 100 repetitions $\Rightarrow$ all rates $1/3$
RMSE of various options

- European option
- Asian option
- Barrier option

Parameters: $S_0 = 1$, $r = 0.05$, $\sigma = 0.2$, $K = 1$, $T = 1$, $d_1 = 1$, $N_1 = 5$, 100 repetitions
Convergence rates

- European option: $\alpha = 1$
- Asian option: $\alpha = 1$
- Barrier option: $\alpha = 1/2$

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>RMSE rate $\alpha/(2\alpha + 1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/3</td>
</tr>
<tr>
<td>1/2</td>
<td>1/4</td>
</tr>
</tbody>
</table>

Improvement: discrete minimum correction for Barrier options:

$$\hat{S}_{min} = \min_{j=1,\ldots,d} \hat{S}(t_j) - k^* \sigma \sqrt{T/d}$$

with $k^* = 0.5826$ recovers $\alpha = 1$ (Kou, 2003)
Monte Carlo
Multilevel Monte Carlo
Adaptive Multilevel Monte Carlo
Parallel Multilevel Monte Carlo

Approach
Error Estimates
Examples

RMSE of various options

Parameters: $S_0 = 1$, $r = 0.05$, $\sigma = 0.2$, $K = 1$, $T = 1$, $d_1 = 1$, $N_1 = 5$, 100 repetitions

Thomas Gerstner
Parallel Multilevel Monte Carlo Simulation
Outline

1. Monte Carlo
   - Approach
   - Error Estimates
   - Examples

2. Multilevel Monte Carlo
   - Approach
   - Error Estimates
   - Examples

3. Adaptive Multilevel Monte Carlo
   - Approach
   - Examples

4. Parallel Multilevel Monte Carlo
   - Approach
   - Algorithm
   - Numerical Results
Multilevel Approach

Approach (Giles, 2008): simulate asset prices for different mesh widths $h_1, \ldots, h_L$ in time

- small mesh width $\implies$ low discretization error, but large costs
- large mesh width $\implies$ high discretization error, but small costs

Rewrite the payoff on the finest level $L$ as a telescope sum

$$E[\hat{P}_L] = E[\hat{P}_0] + \sum_{l=1}^{L} E[\hat{P}_l - \hat{P}_{l-1}].$$

where $\hat{P}_l$ is the approximation for mesh width $h_l = M^{-l} T$. 
Multilevel Approach

Computation: From the estimate of the expectation $E[\hat{P}_l - \hat{P}_{l-1}]$:

$$\hat{Y}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} (\hat{P}_l^{(i)} - \hat{P}_{l-1}^{(i)}), \quad \text{for} \quad l = 1, \ldots, L,$$

and the estimate of $E[\hat{P}_0]$:

$$\hat{Y}_0 = \frac{1}{N_0} \sum_{i=1}^{N_0} \hat{P}_0^{(i)}$$

results the multilevel estimate of $E[\hat{P}_L]$:

$$\hat{Y} = \sum_{l=0}^{L} \hat{Y}_l,$$

with $N_l$ being the number of simulations for level $l$. 
Multilevel Error Estimates

Conditions

- weak convergence

\[ \mathbb{E} \left[ \hat{P}_l - P \right] \leq c_1 h_1^{\alpha} \]

- strong convergence

\[ \mathbb{V} \left[ \hat{Y}_l \right] \leq c_2 N_l^{-1} h_1^{\beta} \]

which corresponds to the strong order of the SDE discretization

\[ \max_{1 \leq j \leq d} \mathbb{E} \left( |f(\hat{S}(t_j)) - f(S(t_j))|^p \right)^{1/p} \leq c_3 h^{\beta/2} \]

Multilevel Complexity (Giles, 2008):

\[ \text{RMSE} \approx \begin{cases} O(C^{-1/2}) & \text{for } \beta \geq 1, \\ O(\frac{-\alpha}{2\alpha+1-\beta}) & \text{for } \beta < 1. \end{cases} \]
Examples

Alpha of various options

Parameters: $S_0 = 1$, $r = 0.05$, $\sigma = 0.2$, $K = 1$, $T = 1$, $N_0 = 100$, 100 repetitions
Examples

Beta of various options

Parameters: $S_0 = 1$, $r = 0.05$, $\sigma = 0.2$, $K = 1$, $T = 1$, $N_0 = 100$, 100 repetitions
Examples

RMSE of various options

Parameters: $S_0 = 1$, $r = 0.05$, $\sigma = 0.2$, $K = 1$, $T = 1$, $N_0 = 100$, 100 repetitions
### Summary

<table>
<thead>
<tr>
<th>Option</th>
<th>expected</th>
<th>computed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\alpha$</td>
<td>$\beta$</td>
</tr>
<tr>
<td>Asian</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Lookback</td>
<td>1/2</td>
<td>1</td>
</tr>
<tr>
<td>Barrier</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td>Digital</td>
<td>1</td>
<td>1/2</td>
</tr>
</tbody>
</table>

RMSE = $\alpha/(2\alpha + 1 - \beta)$ if $\beta < 1$ and RMSE = 1/2 if $\beta \geq 1$
Adaptive Path Discretization

- MLMC not optimal for barrier options
- one idea: rewrite payoff as a product of probabilities

\[ P = e^{-r(S(t_N) - K)^+} \prod_{i=0}^{N-1} p_i \]

with
\[ p_i = 1 - \exp \left( \frac{-2(S_n - B)^+(S_{n+1} - B)^+}{\sigma^2 S_n^2 T/N} \right) \]

- also possible for Double Barrier Options but getting more complicated with more conditions in the payoff
Adaptive Path Discretization

- more general idea: use adaptive path discretization close to the barrier
- adaptively refine time intervals if the barrier-crossing probability is large
  \[ \Psi(S_{i-1}, S_i) := \mathbb{P}(S_{i-1/2} < B) > w \]
- no additional complexity for more complicated options
Adaptive Path Discretization

- How to construct the mid points after refinement?
- version 1: Brownian bridge using the mean of forward and 'backward' Euler-Maruyama estimates

\[
S_{i-1/2} = \frac{1}{2} \left( S_{i-1} + S_{i-1}r(t_{i-1/2} - t_{i-1}) + S_{i-1}\sigma(W_{i-1/2} - W_{i-1}) \right) + \frac{1}{2} \left( \frac{S_i}{1 + r(t_i - t_{i-1/2}) + \sigma(W_i - W_{i-1/2})} \right)
\]

- version 2: Brownian bridge using an arithmetic Brownian motion

\[
S_{i-1/2} = S_{i-1} + r(t_{i-1/2} - t_{i-1}) + \sigma(W_{i-1/2} - W_{i-1})
\]
Example

Down and Out Call

- $S(t)$
- $W(t)$
- Barrier

Thomas Gerstner

Parallel Multilevel Monte Carlo Simulation
Example

Down and Out Call

\[ S(t) \]

\[ W(t) \]

\[ \text{Barrier} \]
Example

Down and Out Call

\[ S(t), W(t) \]

Barrier

Thomas Gerstner
Parallel Multilevel Monte Carlo Simulation
Example

10^4 paths for a Barrier option

Refinement steps with w=−1.28 and d=1

Number

Steps

0 5 10 15 20 25

0
2
4
6
8
10

log(Number)

Steps

0 5 10 15 20 25

0
2
4
6
8
10
Again rewrite the payoff on the finest level $L$ as a telescope sum

$$E[\hat{P}_L^{w_L}] = E[\hat{P}_0^{w_0}] + \sum_{l=1}^{L} E[\hat{P}_l^{w_l} - \hat{P}_{l-1}^{w_{l-1}}].$$

where $\hat{P}_l^{w_l}$ is the approximation for mesh width $h_l = M^{-l} T$ and adaptive path discretization parameter $w_l$.

Calculate the expectations on the right side with the corresponding Monte Carlo estimator.
Alpha of various methods for Barrier Options

Parameters: $S_0 = 1, r = 0.05, \sigma = 0.2, K = 1, T = 1, N_0 = 100, 100 repetitions$
Beta of various methods for Barrier Options

Parameters: $S_0 = 1$, $r = 0.05$, $\sigma = 0.2$, $K = 1$, $T = 1$, $N_0 = 100$, 100 repetitions
Approach

Examples

Monte Carlo
Multilevel Monte Carlo
Adaptive Multilevel Monte Carlo
Parallel Multilevel Monte Carlo

RMSE of various methods for Barrier Options

- + - transformed Barrier
- + - adaptive refining
+ - Barrier

<table>
<thead>
<tr>
<th>Method</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barrier</td>
<td>0.13</td>
</tr>
<tr>
<td>transf. Barrier</td>
<td>0.29</td>
</tr>
<tr>
<td>adapt. refining</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Parameters: $S_0 = 1$, $r = 0.05$, $\sigma = 0.2$, $K = 1$, $T = 1$, $N_0 = 100$, 100 repetitions
Outline

1. Monte Carlo
   - Approach
   - Error Estimates
   - Examples

2. Multilevel Monte Carlo
   - Approach
   - Error Estimates
   - Examples

3. Adaptive Multilevel Monte Carlo
   - Approach
   - Examples

4. Parallel Multilevel Monte Carlo
   - Approach
   - Algorithm
   - Numerical Results
Parallel programming

Properties

- Computer with more than one processor/kernel needed
- Linear speed-up in the number of processors possible
- Efficient for loops that do the same recurring tasks
- Very useful for Monte Carlo simulation

Example: Parallel Monte Carlo with $M$ processors

$$
\hat{V}_j(S, 0) = e^{-rT} \frac{1}{N/M} \sum_{i=1}^{N/M} \hat{V} \left( \{ \hat{S}^{(i,j)}(t_1), \ldots, \hat{S}^{(i,j)}(t_d) \}, T \right)
$$

and the aggregated estimator for the option price is

$$
\hat{V}(S, 0) = \sum_{j=1}^{M} \hat{V}_j(S, 0)
$$
Parallel Multilevel Monte Carlo

Idea

- Parallelize each sum of the MLMC estimator

\[ \hat{Y} = \sum_{l=0}^{L} \hat{Y}_l \]

with

\[ \hat{Y}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} (\hat{P}_l^{(i)} - \hat{P}_{l-1}^{(i)}) \]

- Do not overcharge the memory
  - do not save the whole path of \( S \) but only the necessary part for the option value
  - do not save all option values but calculate the expectation and variance recursively
Algorithm

1. set $L := 0$, $N_{M_0} = \lceil 1000/M \rceil$

2. determine the variances $V_{k,l}$ for $k = 1, \ldots, M$ and $l = 0, \ldots, L$ on each processor such that $V_l = \left( \sum_{k=1}^{M} V_{k,l} \right) / M$ and $V := \sum_{l=0}^{L} V_l$

3. define optimal $N_l$, $l = 0, \ldots, L$ as in the standard algorithm and if $N_l$ has increased calculate $N_{M_l} = \lceil (N_{l}^{\text{new}} - N_{l}^{\text{old}}) / M \rceil$ extra samples on each processor

4. stop if $\text{RMSE} < \epsilon$ and $L \geq 2$

5. else set $L := L + 1$, $N_{M_l} = \lceil 1000/M \rceil$ and go to step 2.
**European Option**

**Figure:** Convergence rates in time for a European option using 1, 10 and 50 threads.
European Option

<table>
<thead>
<tr>
<th>$\epsilon$</th>
<th>$2 \cdot 10^{-3}$</th>
<th>$1 \cdot 10^{-3}$</th>
<th>$5 \cdot 10^{-4}$</th>
<th>$1 \cdot 10^{-4}$</th>
<th>$5 \cdot 10^{-5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 kernels</td>
<td>8.1</td>
<td>8.6</td>
<td>9.3</td>
<td>10.0</td>
<td>10.0</td>
</tr>
<tr>
<td>50 kernels</td>
<td>13.4</td>
<td>16.5</td>
<td>24.3</td>
<td>45.4</td>
<td>48.8</td>
</tr>
</tbody>
</table>

Table: Time factor improvements for a European option with 10 and 50 kernels compared to 1 kernel.

- linear speed-up in the number of kernels if the program is running for more than 1 second
- code written in C++ using MPI
Conclusions

Summary:

- Monte Carlo: RMSE rate 1/3
- Multilevel Monte Carlo: RMSE rate 1/2 in best case
- Adaptive Multilevel Monte Carlo: RMSE rate 1/2 also for barrier options
- Parallel Multilevel Monte Carlo: linear speed-up

Extensions:

- MLMC for Milstein and higher order schemes (Giles, 2007)
- Multilevel Quasi-Monte Carlo (Giles, Waterhouse, 2009; G., Noll 2012)
- Multilevel (adaptive) sparse grid integration (G., Heinz, 2012)