

Multilevel Monte Carlo Methoden

Hendrik Kleikamp

Hauptseminar Wissenschaftliches Rechnen

5. Juli 2017



Inhaltsübersicht

- 1 Einführung
- 2 Multilevel Monte Carlo Methoden
- 3 Hauptresultat
- 4 Implementierung
- 5 Multi-Index Monte Carlo

Kapitel 1

Einführung

Beispiel: Kurssimulationen¹

Aktienkurse in Abhängigkeit der Zeit sind Lösungen der Stochastischen Differentialgleichung

$$dS(t) = a(S, t)dt + b(S, t)dW(t), \quad 0 < t < T.$$

¹Weiterf. Informationen bspw. in „Stochastic Differential Equations and Financial Mathematics“ von Thomas Ønskog (2016, Vorlesungsunterlagen)

Beispiel: Kurssimulationen¹

Aktienkurse in Abhängigkeit der Zeit sind Lösungen der Stochastischen Differentialgleichung

$$dS(t) = a(S, t)dt + b(S, t)dW(t), \quad 0 < t < T.$$

Gesucht in Anwendungen: Erwarteter Wert $\mathbb{E}[P(S)]$ des Kurses zum Zeitpunkt T .

Wir schreiben von nun an P statt $P(S)$.

¹Weiterf. Informationen bspw. in „Stochastic Differential Equations and Financial Mathematics“ von Thomas Ønskog (2016, Vorlesungsunterlagen)

Beispiel: Kurssimulationen¹

Aktienkurse in Abhängigkeit der Zeit sind Lösungen der Stochastischen Differentialgleichung

$$dS(t) = a(S, t)dt + b(S, t)dW(t), \quad 0 < t < T.$$

Gesucht in Anwendungen: Erwarteter Wert $\mathbb{E}[P(S)]$ des Kurses zum Zeitpunkt T .

Wir schreiben von nun an P statt $P(S)$.

✚ **Problem:** $\mathbb{E}[P]$ kann (oft) nicht analytisch bestimmt werden.

¹Weiterf. Informationen bspw. in „Stochastic Differential Equations and Financial Mathematics“ von Thomas Ønskog (2016, Vorlesungsunterlagen)

Beispiel: Kurssimulationen

Idee der Monte Carlo Methode:

- 1 Bestimmen (approximativ) die Lösung S der SDGL.

Beispiel: Kurssimulationen

Idee der Monte Carlo Methode:

- 1 Bestimmen (approximativ) die Lösung S der SDGL.
- 2 Simulieren N Kursverläufe $S^{(1)}, \dots, S^{(N)}$ bezüglich S .

Beispiel: Kurssimulationen

Idee der Monte Carlo Methode:

- 1 Bestimmen (approximativ) die Lösung S der SDGL.
- 2 Simulieren N Kursverläufe $S^{(1)}, \dots, S^{(N)}$ bezüglich S .
- 3 Berechnen mit $P^{(i)} := P(S^{(i)})$ den Monte Carlo Schätzer

$$X := \frac{1}{N} \sum_{i=1}^N P^{(i)} \quad (1)$$

als Näherung an $\mathbb{E}[P]$.

Beispiel: Kurssimulationen

Idee der Monte Carlo Methode:

- 1 Bestimmen (approximativ) die Lösung S der SDGL.
- 2 Simulieren N Kursverläufe $S^{(1)}, \dots, S^{(N)}$ bezüglich S .
- 3 Berechnen mit $P^{(i)} := P(S^{(i)})$ den Monte Carlo Schätzer

$$X := \frac{1}{N} \sum_{i=1}^N P^{(i)} \quad (1)$$

als Näherung an $\mathbb{E}[P]$.

Es ist (Erwartungstreue von X)

$$\mathbb{E}[X] = \mathbb{E}[P]$$

und

$$\mathbb{V}[X] = \frac{1}{N} \mathbb{V}[P].$$

Beispiel: Kurssimulationen

Als root mean squared error ($RMSE$) dieses Ansatzes ergibt sich:

$$\begin{aligned} RMSE &= \sqrt{\mathbb{E}[(X - \mathbb{E}[P])^2]} \\ &= \sqrt{\mathbb{V}[X] + (\mathbb{E}[X] - \mathbb{E}[P])^2} \\ &= \sqrt{\frac{1}{N} \mathbb{V}[P]} \\ &= \mathcal{O}(N^{-1/2}). \end{aligned}$$

Beispiel: Kurssimulationen

Als root mean squared error ($RMSE$) dieses Ansatzes ergibt sich:

$$\begin{aligned} RMSE &= \sqrt{\mathbb{E}[(X - \mathbb{E}[P])^2]} \\ &= \sqrt{\mathbb{V}[X] + (\mathbb{E}[X] - \mathbb{E}[P])^2} \\ &= \sqrt{\frac{1}{N} \mathbb{V}[P]} \\ &= \mathcal{O}(N^{-1/2}). \end{aligned}$$

Für eine Genauigkeit von $\epsilon > 0$ werden also $N = \mathcal{O}(\epsilon^{-2})$ Samples benötigt.

Beispiel: Kurssimulationen

Als root mean squared error ($RMSE$) dieses Ansatzes ergibt sich:

$$\begin{aligned} RMSE &= \sqrt{\mathbb{E}[(X - \mathbb{E}[P])^2]} \\ &= \sqrt{\mathbb{V}[X] + (\mathbb{E}[X] - \mathbb{E}[P])^2} \\ &= \sqrt{\frac{1}{N} \mathbb{V}[P]} \\ &= \mathcal{O}(N^{-1/2}). \end{aligned}$$

Für eine Genauigkeit von $\epsilon > 0$ werden also $N = \mathcal{O}(\epsilon^{-2})$ Samples benötigt.

✚ **Problem:** Hoher Rechenaufwand, falls ϵ klein ist.

Kapitel 2

Multilevel Monte Carlo Methoden

Idee der Multilevel Monte Carlo Methode

Ziel: Gleichbleibende Genauigkeit bei geringerem Rechenaufwand.

Im Allgemeinen:

Gegeben: Wahrscheinlichkeitsraum $(\Omega, \mathcal{A}, \mathbb{P})$,
(reelle) Zufallsvariable P

Gesucht: $\mathbb{E}[P]$

Idee der Multilevel Monte Carlo Methode

Ziel: Gleichbleibende Genauigkeit bei geringerem Rechenaufwand.

Im Allgemeinen:

Gegeben: Wahrscheinlichkeitsraum $(\Omega, \mathcal{A}, \mathbb{P})$,
(reelle) Zufallsvariable P

Gesucht: $\mathbb{E}[P]$

Ansatz: Approximieren P durch eine Folge $\{P_l\}_{l \in \mathbb{N}}$ für die gilt:

- ⊕ Approximationsfehler von P_l sinkt mit wachsendem l .
- ⊖ Rechenaufwand für Simulationen steigt mit wachsendem l monoton.

Idee der Multilevel Monte Carlo Methode

Ziel: Gleichbleibende Genauigkeit bei geringerem Rechenaufwand.

Im Allgemeinen:

Gegeben: Wahrscheinlichkeitsraum $(\Omega, \mathcal{A}, \mathbb{P})$,
(reelle) Zufallsvariable P

Gesucht: $\mathbb{E}[P]$

Ansatz: Approximieren P durch eine Folge $\{P_l\}_{l \in \mathbb{N}}$ für die gilt:

- ⊕ Approximationsfehler von P_l sinkt mit wachsendem l .
- ⊖ Rechenaufwand für Simulationen steigt mit wachsendem l monoton.

Idee: Vorteile von kleinen und großen Leveln l kombinieren, um $\mathbb{E}[P_L]$ für ein L als Näherung an $\mathbb{E}[P]$ zu berechnen.

Idee der Multilevel Monte Carlo Methode

Wir nutzen folgende Teleskopsumme:

$$\mathbb{E}[P_L] = \mathbb{E}[P_0] + \sum_{l=1}^L \mathbb{E}[P_l - P_{l-1}]. \quad (2)$$

Idee der Multilevel Monte Carlo Methode

Wir nutzen folgende Teleskopsumme:

$$\mathbb{E}[P_L] = \mathbb{E}[P_0] + \sum_{l=1}^L \mathbb{E}[P_l - P_{l-1}]. \quad (2)$$

Einsetzen eines Schätzers wie in Formel (1) für jeden Erwartungswert ergibt den Schätzer Y ,

$$Y := \frac{1}{N_0} \sum_{i=1}^{N_0} P_0^{(0,i)} + \sum_{l=1}^L \left(\frac{1}{N_l} \sum_{i=1}^{N_l} P_l^{(l,i)} - P_{l-1}^{(l,i)} \right), \quad (3)$$

für $\mathbb{E}[P_L] \approx \mathbb{E}[P]$.

Fehler der Multilevel Monte Carlo Methode

Es gilt

$$\mathbb{E}[Y] = \mathbb{E}[P_L]$$

und

$$\mathbb{V}[Y] = \sum_{l=0}^L \frac{1}{N_l} \mathbb{V}[P_l - P_{l-1}]$$

wobei $P_{-1} \equiv 0$ gesetzt wird.

Fehler der Multilevel Monte Carlo Methode

Es gilt

$$\mathbb{E}[Y] = \mathbb{E}[P_L]$$

und

$$\mathbb{V}[Y] = \sum_{l=0}^L \frac{1}{N_l} \mathbb{V}[P_l - P_{l-1}]$$

wobei $P_{-1} \equiv 0$ gesetzt wird.

Dies ergibt einen mean squared error (MSE) von

$$\begin{aligned} MSE &= \mathbb{E}[(Y - \mathbb{E}[P])^2] \\ &= \mathbb{V}[Y] + (\mathbb{E}[Y] - \mathbb{E}[P])^2 \\ &= \underbrace{\sum_{l=0}^L \frac{1}{N_l} \mathbb{V}[P_l - P_{l-1}]}_{\text{Stichproben-Fehler}} + \underbrace{(\mathbb{E}[P_L - P])^2}_{\text{Approximationsfehler}}. \end{aligned} \quad (4)$$

Konkreter: Zwei-Level Monte Carlo Methode

Zwei-Level Schätzer:

$$Y := \frac{1}{N_0} \sum_{i=1}^{N_0} P_0^{(0,i)} + \frac{1}{N_1} \sum_{i=1}^{N_1} \underbrace{P_1^{(1,i)} - P_0^{(1,i)}}_{\text{Differenz von } P_1 \text{ und } P_0 \text{ bei gleichen Samples}}$$

Konkreter: Zwei-Level Monte Carlo Methode

Zwei-Level Schätzer:

$$Y := \frac{1}{N_0} \sum_{i=1}^{N_0} P_0^{(0,i)} + \frac{1}{N_1} \sum_{i=1}^{N_1} \underbrace{P_1^{(1,i)} - P_0^{(1,i)}}_{\text{Differenz von } P_1 \text{ und } P_0 \text{ bei gleichen Samples}}$$

Fragen:

- Wie verhalten sich Berechnungskosten und Varianz des Schätzers?
- Wie sollten N_0 und N_1 am besten gewählt werden?

Berechnungskosten und Varianz des Zwei-Level-Schätzers

Seien C_0 und C_1 die Berechnungskosten für ein Sample von P_0 und $P_1 - P_0$.

Gesamtkosten: $C = N_0 C_0 + N_1 C_1$

Berechnungskosten und Varianz des Zwei-Level-Schätzers

Seien C_0 und C_1 die Berechnungskosten für ein Sample von P_0 und $P_1 - P_0$.

Gesamtkosten: $C = N_0 C_0 + N_1 C_1$

Seien V_0 und V_1 die Varianzen von P_0 und $P_1 - P_0$.

Varianz des Schätzers: $V = \frac{V_0}{N_0} + \frac{V_1}{N_1}$

Berechnungskosten und Varianz des Zwei-Level-Schätzers

Seien C_0 und C_1 die Berechnungskosten für ein Sample von P_0 und $P_1 - P_0$.

Gesamtkosten: $C = N_0 C_0 + N_1 C_1$

Seien V_0 und V_1 die Varianzen von P_0 und $P_1 - P_0$.

Varianz des Schätzers: $V = \frac{V_0}{N_0} + \frac{V_1}{N_1}$

Sei V fest. Wir minimieren C , wobei N_0, N_1 reelle Variablen seien.

⇒ **Lagrange-Multiplikatoren-Methode**

Lagrange-Multiplikatoren-Methode

Lagrange-Funktion:

$$\mathcal{L}((N_0, N_1), \mu) = N_0 C_0 + N_1 C_1 + \mu \cdot \left(\frac{V_0}{N_0} + \frac{V_1}{N_1} - V \right)$$

Lagrange-Multiplikatoren-Methode

Lagrange-Funktion:

$$\mathcal{L}((N_0, N_1), \mu) = N_0 C_0 + N_1 C_1 + \mu \cdot \left(\frac{V_0}{N_0} + \frac{V_1}{N_1} - V \right)$$

Karush-Kuhn-Tucker-Bedingungen:

$$\begin{aligned} 0 &\stackrel{!}{=} \nabla_{(N_0, N_1)} \mathcal{L} = \begin{bmatrix} C_0 - \mu V_0 / N_0^2 \\ C_1 - \mu V_1 / N_1^2 \end{bmatrix} \\ \Leftrightarrow \quad N_0 &= \sqrt{\frac{\mu V_0}{C_0}} \quad \wedge \quad N_1 = \sqrt{\frac{\mu V_1}{C_1}} \end{aligned} \tag{5}$$

Varianz

Ziel: Gesamtvarianz $V = \epsilon^2/2$

Mit Gleichung (5) folgt:

$$\begin{aligned}\epsilon^2/2 &= V \\ \Leftrightarrow \epsilon^2/2 &= V_0/N_0 + V_1/N_1 \\ \Leftrightarrow \epsilon^2/2 &= \sqrt{V_0 C_0/\mu} + \sqrt{V_1 C_1/\mu} \\ \Leftrightarrow \epsilon^2/2 &= \frac{1}{\sqrt{\mu}} (\sqrt{V_0 C_0} + \sqrt{V_1 C_1}) \\ \Leftrightarrow \sqrt{\mu} &= \frac{2(\sqrt{V_0 C_0} + \sqrt{V_1 C_1})}{\epsilon^2} \quad (6)\end{aligned}$$

Rechenaufwand

Mit (5) und (6) gilt für den Rechenaufwand C nun:

$$\begin{aligned} C &= N_0 C_0 + N_1 C_1 \\ &= \sqrt{\frac{\mu V_0}{C_0}} C_0 + \sqrt{\frac{\mu V_1}{C_1}} C_1 \\ &= \sqrt{\mu} (\sqrt{V_0 C_0} + \sqrt{V_1 C_1}) \\ &= 2 \frac{(\sqrt{V_0 C_0} + \sqrt{V_1 C_1})^2}{\epsilon^2} \end{aligned} \tag{7}$$

Optimale Anzahl von Samples

Zusammenfassend ergibt sich:

$$N_0 = \sqrt{\mu} \sqrt{\frac{V_0}{C_0}} = 2 \left(\frac{\sqrt{V_0 C_0} + \sqrt{V_1 C_1}}{\epsilon^2} \right) \sqrt{\frac{V_0}{C_0}}$$

$$N_1 = \sqrt{\mu} \sqrt{\frac{V_1}{C_1}} = 2 \left(\frac{\sqrt{V_0 C_0} + \sqrt{V_1 C_1}}{\epsilon^2} \right) \sqrt{\frac{V_1}{C_1}}$$

↪ Alle Resultate lassen sich analog auf $L > 1$ übertragen.

Kapitel 3

Hauptresultat

Multilevel Monte Carlo Theorem

Wichtig: Die Multilevel Monte Carlo Methode benötigt im Allgemeinen keine geometrische Level-Sequenz!

Grundlagen für das Theorem:

- 1 Geometrische Level-Sequenz
- 2 Kosten wachsen exponentiell mit dem Level
- 3 Fehler $|\mathbb{E}[P_l - P]|$ und Varianz $\mathbb{V}[P_l - P_{l-1}]$ sinken exponentiell mit dem Level

Ziele:

- 1 $\mathbb{V}[Y] < \epsilon^2/2$
 - 2 $(\mathbb{E}[P_L - P])^2 < \epsilon^2/2$
- $$\stackrel{(4)}{\implies} MSE < \epsilon^2$$

Theorem (MLMC [Giles, 2015])

Sei P eine Zufallsvariable und P_l eine numerische Approximation an P im Level l .

Angenommen, es existieren unabhängige Schätzer Y_l basierend auf N_l Monte Carlo Samples, jeweils mit erwarteten Kosten C_l und einer Varianz V_l , sowie positive Konstanten $\alpha, \beta, \gamma, c_1, c_2, c_3$ mit $\alpha \geq \frac{1}{2} \min(\beta, \gamma)$ und

$$\mathbf{1} \quad |\mathbb{E}[P_l - P]| \leq c_1 2^{-\alpha l}$$

$$\mathbf{2} \quad \mathbb{E}[Y_l] = \begin{cases} \mathbb{E}[P_0], & l = 0 \\ \mathbb{E}[P_l - P_{l-1}], & l > 0 \end{cases}$$

$$\mathbf{3} \quad V_l \leq c_2 2^{-\beta l}$$

$$\mathbf{4} \quad C_l \leq c_3 2^{\gamma l}.$$

Theorem (MLMC [Giles, 2015])

Dann gibt es eine Konstante $c_4 > 0$, sodass für alle $\epsilon < e^{-1}$ Zahlen L und N_l existieren, für die der Multilevel-Schätzer

$$Y := \sum_{l=0}^L Y_l$$

einen Fehler von

$$MSE = \mathbb{E} [(Y - \mathbb{E}[P])^2] < \epsilon^2$$

besitzt und der Rechenaufwand C insgesamt beschränkt ist durch

$$\mathbb{E}[C] \leq \begin{cases} c_4 \epsilon^{-2}, & \beta > \gamma \\ c_4 \epsilon^{-2} (\log \epsilon)^2, & \beta = \gamma \\ c_4 \epsilon^{-2 - (\gamma - \beta)/\alpha}, & \beta < \gamma. \end{cases}$$

Hinweise zum MLMC-Theorem I²

Wofür werden die vielen Annahmen benötigt?

Bedingung....

- 1 sichert, dass $|\mathbb{E}[P_l - P]| \xrightarrow{l \rightarrow \infty} 0$,
- 2 sichert, dass $\mathbb{E}[Y] = \mathbb{E}[P_L]$,
- 3 sichert, dass $V_l \xrightarrow{l \rightarrow \infty} 0$,
- 4 sichert, dass $C_l < \infty \quad \forall l \geq 0$.

²Ein Beweis des Theorems findet sich in „Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients“ von Cliffe et al. (2011)

Hinweise zum MLMC-Theorem II

Erinnerung:

$$C = 2 \frac{\left(\sum_{l=0}^L \sqrt{V_l C_l} \right)^2}{\epsilon^2} \stackrel{\text{Ann. 3+4}}{\leq} 2 \frac{\left(\sum_{l=0}^L c_2 c_3 2^{l(\gamma-\beta)/2} \right)^2}{\epsilon^2}$$

Hinweise zum MLMC-Theorem II

Erinnerung:

$$C = 2 \frac{\left(\sum_{l=0}^L \sqrt{V_l C_l} \right)^2}{\epsilon^2} \stackrel{\text{Ann. 3+4}}{\leq} 2 \frac{\left(\sum_{l=0}^L c_2 c_3 2^{l(\gamma-\beta)/2} \right)^2}{\epsilon^2}$$

Fall 1: $\beta > \gamma$

→ Größter Rechenaufwand im größten Level

Hinweise zum MLMC-Theorem II

Erinnerung:

$$C = 2 \frac{\left(\sum_{l=0}^L \sqrt{V_l C_l} \right)^2}{\epsilon^2} \stackrel{\text{Ann. 3+4}}{\leq} 2 \frac{\left(\sum_{l=0}^L c_2 c_3 2^{l(\gamma-\beta)/2} \right)^2}{\epsilon^2}$$

Fall 1: $\beta > \gamma$

→ Größter Rechenaufwand im größten Level

Fall 2: $\beta < \gamma$

→ Größter Rechenaufwand im feinsten Level

Hinweise zum MLMC-Theorem II

Erinnerung:

$$C = 2 \frac{\left(\sum_{l=0}^L \sqrt{V_l C_l} \right)^2}{\epsilon^2} \stackrel{\text{Ann. 3+4}}{\leq} 2 \frac{\left(\sum_{l=0}^L c_2 c_3 2^{l(\gamma-\beta)/2} \right)^2}{\epsilon^2}$$

Fall 1: $\beta > \gamma$

→ Größter Rechenaufwand im größten Level

Fall 2: $\beta < \gamma$

→ Größter Rechenaufwand im feinsten Level

Fall 3: $\beta = \gamma$

→ Rechenaufwand in jedem Level ungefähr gleich (Produkt aus Varianz und Kosten unabhängig vom Level)

Hinweise zum MLMC-Theorem III

- Schätzer Y_l kann beliebig konstruiert werden, solange die Voraussetzungen des MLMC-Theorems erfüllt sind.
- Aufteilung des Fehlers ϵ^2 gleichermaßen auf $(\mathbb{E}[P_L - P])^2$ und $\mathbb{V}[Y]$ ist nicht immer optimal.

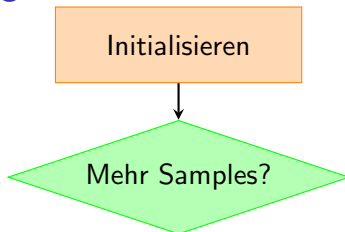
Kapitel 4

Implementierung

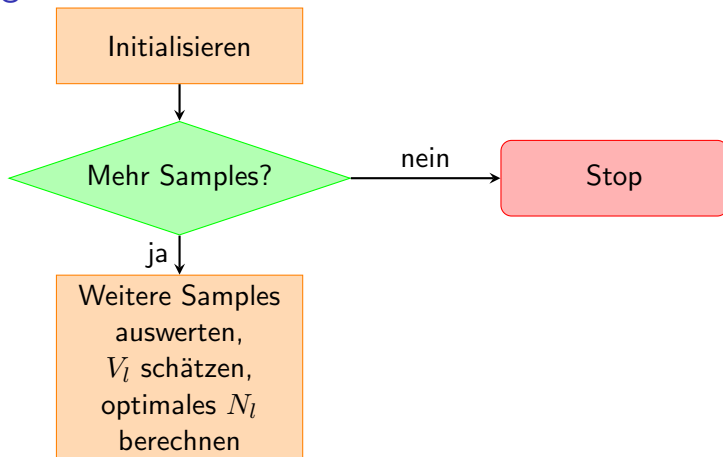
Der Algorithmus

Initialisieren

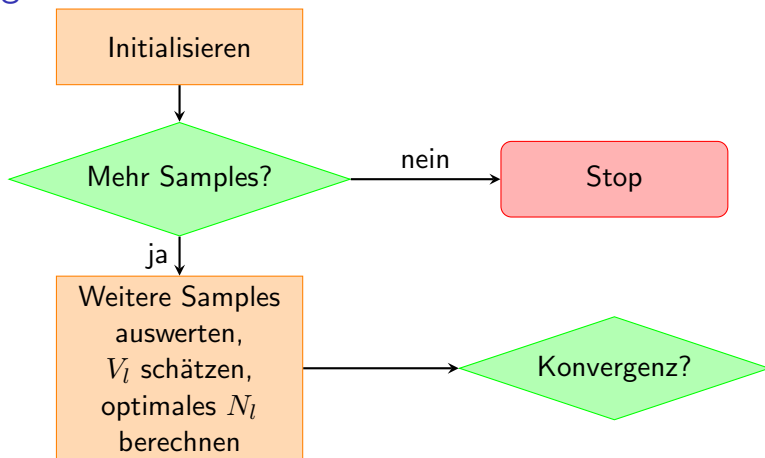
Der Algorithmus



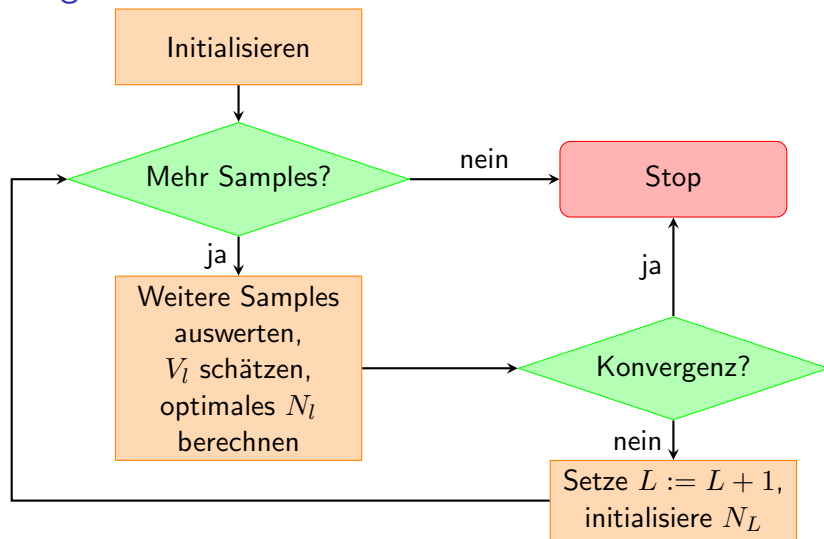
Der Algorithmus



Der Algorithmus



Der Algorithmus



Hinweise zur Implementierung

- Die Konstanten c_1 und c_2 sind meist nicht bekannt.
- Die Genauigkeit der Schätzer für V_l ($l = 0, \dots, L$) hängt von der Anzahl der verwendeten Samples ab.
- **Wichtig:** Der Algorithmus ist heuristisch, ein MSE von ϵ^2 ist nicht garantiert!

Besonderheiten bei der Implementierung³

- Der Konvergenz-Test überprüft, ob $|\mathbb{E}[P_L - P]| < \epsilon/\sqrt{2}$ um den gewünschten *MSE* zu erhalten:

$$|\mathbb{E}[P - P_L]| = \left| \sum_{l=L+1}^{\infty} \mathbb{E}[P_l - P_{l-1}] \right| = \left| \frac{\mathbb{E}[P_L - P_{L-1}]}{(2^\alpha - 1)} \right| \stackrel{!}{<} \epsilon/\sqrt{2}$$

- Sind α und β nicht bekannt, so können diese mittels linearer Regression geschätzt werden.

³Für eine konkrete Umsetzung des MLMC-Algorithmus mit sämtlichen Besonderheiten siehe [1] und <http://people.maths.ox.ac.uk/gilesm/mlmc/> (C/C++ und MATLAB-Code)

Kapitel 5

Multi-Index Monte Carlo

Erweiterung: Multi-Index Monte Carlo⁴

Idee: Mehrdimensionale Levelindizes

⇒ Ermöglicht bspw. unterschiedliche Diskretisierungsweiten in unterschiedliche Koordinatenrichtungen.

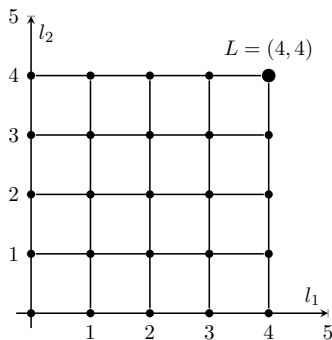


Abbildung: Zwei-dimensionale Indizes

⁴Weiterf. Informationen in „Multi index Monte Carlo: when sparsity meets sampling“ von Haji-Ali et al. (2014)

Multilevel Teleskopsumme im Mehrdimensionalen

Wir definieren für $l = (l_i)_{i=1}^d \in \mathbb{N}^d$ die Differenz

$$\Delta_i P_l = \begin{cases} P_l - P_{l-e_i}, & l_i > 0 \\ P_l, & l_i = 0 \end{cases}$$

wobei e_i den Einheitsvektor in Koordinatenrichtung i bezeichnet.

Multilevel Teleskopsumme im Mehrdimensionalen

Wir definieren für $l = (l_i)_{i=1}^d \in \mathbb{N}^d$ die Differenz

$$\Delta_i P_l = \begin{cases} P_l - P_{l-e_i}, & l_i > 0 \\ P_l, & l_i = 0 \end{cases}$$

wobei e_i den Einheitsvektor in Koordinatenrichtung i bezeichnet.

Sei zudem

$$\Delta P_l = \left(\prod_{i=1}^d \Delta_i \right) P_l.$$

Multilevel Teleskopsumme im Mehrdimensionalen

Wir definieren für $l = (l_i)_{i=1}^d \in \mathbb{N}^d$ die Differenz

$$\Delta_i P_l = \begin{cases} P_l - P_{l-e_i}, & l_i > 0 \\ P_l, & l_i = 0 \end{cases}$$

wobei e_i den Einheitsvektor in Koordinatenrichtung i bezeichnet.

Sei zudem

$$\Delta P_l = \left(\prod_{i=1}^d \Delta_i \right) P_l.$$

Es ergibt sich mit $I = \{l \in \mathbb{N}^d : l \geq 0\}$ der Schätzer

$$Y := \sum_{l \in I} \mathbb{E}[\Delta P_l]$$

für $\mathbb{E}[P]$.

Beispiel: Natürliche Wahl von I

Sei

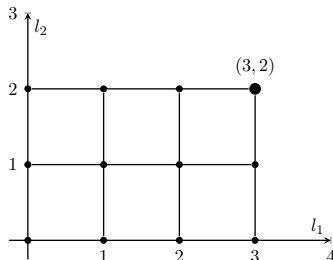
$$I = \{(l_1, l_2) \in \mathbb{N}^2 : 0 \leq l_1 \leq 3 \text{ und } 0 \leq l_2 \leq 2\}.$$

Beispiel: Natürliche Wahl von I

Sei

$$I = \{(l_1, l_2) \in \mathbb{N}^2 : 0 \leq l_1 \leq 3 \text{ und } 0 \leq l_2 \leq 2\}.$$

I entspricht nun einem Rechteck mit Eckpunkt $(3, 2)$.



Dann ist $\Delta P_{(3,2)} = \Delta_1 \Delta_2 P_{(3,2)} = P_{(3,2)} - P_{(2,2)} - P_{(3,1)} + P_{(2,1)}$
und $Y = \mathbb{E}[P_{(3,2)}]$.

Wie sollte I gewählt werden?

Die optimale Wahl ist (ähnlich zu dünnen Gittern)

$$I = \{l \in \mathbb{N}^d : l \geq 0, \langle l, n \rangle \leq L\}$$

für eine Richtung $n \in \mathbb{N}^d, n > 0$, und ein $L \in \mathbb{N}$.

Wie sollte I gewählt werden?

Die optimale Wahl ist (ähnlich zu dünnen Gittern)

$$I = \{l \in \mathbb{N}^d : l \geq 0, \langle l, n \rangle \leq L\}$$

für eine Richtung $n \in \mathbb{N}^d, n > 0$, und ein $L \in \mathbb{N}$.

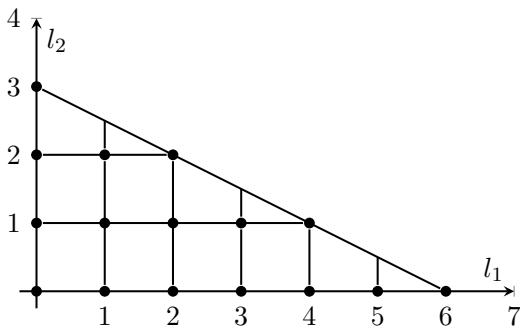


Abbildung: Beispiel mit $L = 6$ und $n = [1, 2]^T$

Wie sollte I gewählt werden?

Die optimale Wahl ist (ähnlich zu dünnen Gittern)

$$I = \{l \in \mathbb{N}^d : l \geq 0, \langle l, n \rangle \leq L\}$$

für eine Richtung $n \in \mathbb{N}^d, n > 0$, und ein $L \in \mathbb{N}$.

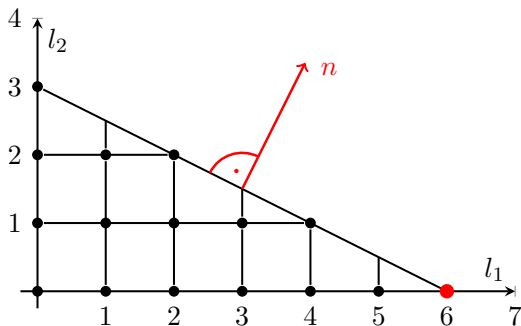


Abbildung: Beispiel mit $L = 6$ und $n = [1, 2]^T$

Optimale Wahl von I in 3D

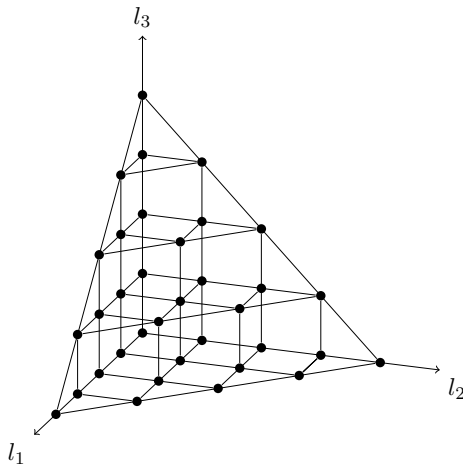


Abbildung: Beispiel mit $L = 4$ und $n = [1, 1, 1]^T$

Zusammenfassung - Multilevel Monte Carlo Methoden

- **Ziel:** $\mathbb{E}[P]$ berechnen.
- **Idee:** $\mathbb{E}[P_L]$ als Näherung an $\mathbb{E}[P]$ berechnen,
 $\mathbb{E}[P_L]$ mittels Teleskopsumme aufspalten:

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

- Erwartungswerte $\mathbb{E}[P_0]$ und $\mathbb{E}[P_l - P_{l-1}]$ ($l = 1, \dots, L$) werden bspw. mit Monte Carlo Schätzern approximiert.
- Hauptresultat liefert bei einer geometrischen Level-Sequenz eine Abschätzung für den Rechenaufwand.

Ausblick I - Weiterführende Forschungsthemen

- Verbesserung des Verfahrens⁵:
Durch welche Wahl von Y_l wird die Varianz noch geringer?
- Nicht geometrische Levelsequenz⁶:
Wie können die Levels und die Anzahl der Samples pro Level nun ausgewählt werden?
- Multilevel Richardson-Romberg Extrapolation⁷:
In welchem Fall bringt dieses Verfahren eine weitere Verbesserung der MLMC-Methode?

⁵ „A new approach to unbiased estimation for SDE's“ von Rhee und Glynn (2012)

⁶ „A model and variance reduction method for computing statistical outputs of stochastic elliptic partial differential equations“ von Giles et al. (2014)

⁷ „Multi-level Richardson-Romberg extrapolation“ von Lemaire und Pagès

Ausblick II - Weiterführende Forschungsthemen

- Mehrdimensionale Zufallsvariable P^8 :
Wie überträgt sich das MLMC-Theorem auf diesen Fall?
- Multilevel Quasi Monte Carlo⁹:
Wie verhält sich hierbei der Rechenaufwand?
- **Kommende Vorträge:** Wie sieht die MLMC-Methode in konkreten Anwendungen aus?

⁸Beispielsweise zu finden in „Complexity of Banach Space Valued and Parametric Integration“ von Daun und Heinrich (2013)

⁹„Multilevel quasi-Monte Carlo path simulation“ von Giles und Waterhouse (2009)

Literaturverzeichnis I



Michael B. Giles: „Multilevel Monte Carlo methods“.
Cambridge University Press, 2015



Sascha Pezoldt: „Multi-Level Monte-Carlo-Methoden und ihre
Anwendung in der Finanzmathematik“.
Masterarbeit am Mathematischen Institut der Universität
Bayreuth, 2016



Stefan Walter Heinz: „Multilevel Monte Carlo Methoden und
deren Anwendung“.
Diplomarbeit am Institut für Computerorientierte Mathematik
der Johann Wolfgang Goethe Universität in Frankfurt am
Main, 2009

Literaturverzeichnis II



Marco Noll: „Multilevel Monte-Carlo Simulationsverfahren mit Anwendung auf Multi-Asset Optionen“.
2010



Patrick Wolff: „Das Multilevel Monte Carlo Verfahren und seine Anwendung im Libor Marktmodell“.
Masterarbeit am Institut für Mathematische Statistik der Westfälischen Wilhelms-Universität Münster, 2013



Abdul-Lateef Haji-Ali, Fabio Nobile, Raúl Tempone: „Multi index Monte Carlo: when sparsity meets sampling“.
Mathematics Institute of Computational Science and Engineering, Lausanne, 2014