# Multilevel Monte-Carlo algorithms for Lévy-driven SDE's

Steffen Dereich

U Marburg / WWU Münster

http://www.mathematik.uni-marburg.de/~dereich/

CREST and 4th Ritsumeikan-Florence Workshop, 8/03/2012

# Outline of the talk

# I Introduction

**SDE:** $Y = (Y_t)_{t\in[0,1]}$ solution to

$$Y_t = y_0 + \int_0^t b(Y_{s-})\,\mathrm{d}s + \int_0^t a(Y_{s-})\,\mathrm{d}X_s,$$

where $\sigma : \mathbb{R}^{d_Y} \to \mathbb{R}^{d_Y \times d_X}$ and $b : \mathbb{R}^{d_Y} \to \mathbb{R}^{d_Y}$ are Lipschitz continuous $C^\infty$-coefficients and $X$ is a Wiener (or more generally a Lévy process).

# I Introduction

**SDE:** $Y = (Y_t)_{t\in[0,1]}$ solution to

$$Y_t = y_0 + \int_0^t b(Y_{s-})\,\mathrm{d}s + \int_0^t a(Y_{s-})\,\mathrm{d}X_s,$$

where $\sigma : \mathbb{R}^{d_Y} \to \mathbb{R}^{d_Y \times d_X}$ and $b : \mathbb{R}^{d_Y} \to \mathbb{R}^{d_Y}$ are Lipschitz continuous $C^\infty$-coefficients and $X$ is a Wiener (or more generally a Lévy process).

**Quadrature problem:** Computation of expectations

$$S(f) := \mathbb{E}[f(Y)],$$

where $f : D[0,1] \to \mathbb{R}$ satisfies certain smoothness assumptions ($D[0,1]$ denotes the Skorokhod space of càdlàg functions on $[0,1]$).

**E.g.:** $f(x) = (\int_0^1 x_t\,\mathrm{d}t - K)_+$ or $f(x) = \max_{t\in[0,1]} x_t$

# I Motivation

**Approximations:** Take a family of approximate solutions $\{Y^{(m)} : m \in \mathbb{N}\}$ with exponentially increasing complexity, for instance Euler with stepsize $2^{-m}$.

# I Motivation

**Approximations:** Take a family of approximate solutions $\{Y^{(m)} : m \in \mathbb{N}\}$ with exponentially increasing complexity, for instance Euler with stepsize $2^{-m}$.

**Classical Monte Carlo:**

$$S(f) = \mathbb{E}[f(Y)] \approx \mathbb{E}[f(Y^{(m)})] \approx \frac{1}{n}\sum_{i=1}^{n} f(Y^{(m,i)}),$$

where $Y^{(m,1)}, Y^{(m,2)}, \ldots$ are independent copies of $Y^{(m)}$.

**Monte Carlo with extrapolation:** (Talay, Tubaro '90, Bally, Talay '96)

$$\mathbb{E}[f(Y)] \approx \mathbb{E}[2f(Y^{(m+1)}) - f(Y^{(m)})] \approx \frac{1}{n}\sum_{i=1}^{n} 2f(Y^{(m+1,i)}) - f(Y^{(m,i)}),$$

where $(Y^{(m+1,1)}, Y^{(m,1)}), \ldots$ are independent copies of $(Y^{(m+1)}, Y^{(m)})$.

# I Motivation

**Approximations:** Take a family of approximate solutions $\{Y^{(m)} : m \in \mathbb{N}\}$ with exponentially increasing complexity, for instance Euler with stepsize $2^{-m}$.

**Classical Monte Carlo:**

$$S(f) = \mathbb{E}[f(Y)] \approx \mathbb{E}[f(Y^{(m)})] \approx \frac{1}{n}\sum_{i=1}^{n} f(Y^{(m,i)}),$$

where $Y^{(m,1)}, Y^{(m,2)}, \ldots$ are independent copies of $Y^{(m)}$.

**Monte Carlo with extrapolation:** (Talay, Tubaro '90, Bally, Talay '96)

$$\mathbb{E}[f(Y)] \approx \mathbb{E}[2f(Y^{(m+1)}) - f(Y^{(m)})] \approx \frac{1}{n}\sum_{i=1}^{n} 2f(Y^{(m+1,i)}) - f(Y^{(m,i)}),$$

where $(Y^{(m+1,1)}, Y^{(m,1)}), \ldots$ are independent copies of $(Y^{(m+1)}, Y^{(m)})$.

**Approximation error:** Typically one gets for non-path dependent options for classical Monte Carlo

$$\text{root mean squared error} \approx N^{-1/3}$$

and for Monte Carlo with extrapolation

$$\text{root mean squared error} \approx N^{-2/5}$$

in the computational time $N$ as $N \to \infty$.

# I Motivation

**Alternatives for approximate soutions:** Kusuoka-Lyons-Victoir methods, splitting methods, . . .

**Strong research activity** since the 90s with contribution by (e.g.) Kusuoka, Kohatsu-Higa and many others

**Focus of the talk:**

I Lower bounds for quadrature problems

II Multilevel methods with a focus on Lévy-driven SDEs

In [I] we will follow the ideas of information based complexity (Novak, Plaskota, Ritter, Sloan, Wasilkowski, Wozniakowski, . . . ) and specify an algorithmic problem and provide lower bounds.

In [II] we briefly introduce multilevel Monte Carlo and provide upper bounds for their efficiency for Lévy-driven SDEs

# II Information based complexity point of view

**Question:** Can one prove lower bounds in the quadrature problem?

We restrict attention to continuous diffusions.

For doing so we need to specify

(a) a class $\mathcal{F}$ of functions on which we test the algorithm

(b) the computational cost of algorithms and

(c) an error criterion.

# II Information based complexity point of view

**Question:** Can one prove lower bounds in the quadrature problem?

We restrict attention to continuous diffusions.

For doing so we need to specify

(a) a class $\mathcal{F}$ of functions on which we test the algorithm

(b) the computational cost of algorithms and

(c) an error criterion.

**(a):** We consider $\mathcal{F}$ as the 1-Lipschitz continuous functions $f : C[0,1] \to \mathbb{R}$ (w.r.t. supremum norm)

**(b):** Specify a nested sequence $\{V_n : n \in \mathbb{N}_0\}$ of linear subsets of $C[0,1]$ with dimension $\dim(V_n) = n$ and charge each evaluation of $f$ for a path $x \in V_n \backslash V_{n-1}$ with cost $n$. All real number operations are allowed.

**(c):** An algorithm $\widehat{\mathcal{S}}$ produces a possibly random output $\widehat{S}(f)$ when applied to $f$ and we consider the worst-case-error

$$\mathrm{err}(\widehat{\mathcal{S}}) := \sup_{f \in \mathcal{F}} \mathbf{E}[(S(f) - \widehat{S}(f))^2]^{1/2}.$$

# II Information based complexity point of view

**Question:** Can one prove lower bounds in the quadrature problem?

We restrict attention to continuous diffusions.

For doing so we need to specify

(a) a class $\mathcal{F}$ of functions on which we test the algorithm

(b) the computational cost of algorithms and

(c) an error criterion.

**(a):** We consider $\mathcal{F}$ as the 1-Lipschitz continuous functions $f : C[0,1] \to \mathbb{R}$ (w.r.t. supremum norm)

**(b):** Specify a nested sequence $\{V_n : n \in \mathbb{N}_0\}$ of linear subsets of $C[0,1]$ with dimension $\dim(V_n) = n$ and charge each evaluation of $f$ for a path $x \in V_n \backslash V_{n-1}$ with cost $n$. All real number operations are allowed.

**(c):** An algorithm $\widehat{\mathcal{S}}$ produces a possibly random output $\widehat{S}(f)$ when applied to $f$ and we consider the worst-case-error

$$\mathrm{err}(\widehat{\mathcal{S}}) := \sup_{f \in \mathcal{F}} \mathbf{E}[(S(f) - \widehat{S}(f))^2]^{1/2}.$$

**Task:** Prove lower bounds for the minimal error

$$e(N) := \inf\{\mathrm{err}(\widehat{\mathcal{S}}) : \mathrm{cost}(\widehat{\mathcal{S}}) \leq N\}.$$

# II Comparison of algorithmic concepts

**Ref.:** Creutzig, D, Müller-Gronbach, Ritter '09

**Variable subspace sampling:**

$$e^{\mathrm{variable}}(N) \geq \mathrm{const}\, N^{-1/2}$$

# II Comparison of algorithmic concepts

**Ref.:** Creutzig, D, Müller-Gronbach, Ritter '09

**Variable subspace sampling:**

$$e^{\text{variable}}(N) \geq \text{const}\, N^{-1/2}$$

**Fixed subspace sampling:** Algorithm needs to specify one subspace $V_n$ at first and sampling is only possible in $V_n$ at cost $n$.

$$e^{\text{fixed}}(N) \geq \text{const}\, N^{-1/4}\, (\log N)^{-3/4}$$

# II Comparison of algorithmic concepts

**Ref.:** Creutzig, D, Müller-Gronbach, Ritter '09

**Variable subspace sampling:**

$$e^{\mathrm{variable}}(N) \geq \mathrm{const}\, N^{-1/2}$$

**Fixed subspace sampling:** Algorithm needs to specify one subspace $V_n$ at first and sampling is only possible in $V_n$ at cost $n$.

$$e^{\mathrm{fixed}}(N) \geq \mathrm{const}\, N^{-1/4}\, (\log N)^{-3/4}$$

**Full space sampling:** Algorithm is allowed to sample arbitrarily and each evaluation of $f$ costs one unit.

$$e^{\mathrm{free}}(N) \geq \mathrm{const}\, N^{-1/2} (\log N)^{-3/2}$$

# II Comparison of algorithmic concepts

**Ref.:** Creutzig, D, Müller-Gronbach, Ritter '09

**Variable subspace sampling:**

$$e^{\mathrm{variable}}(N) \geq \mathrm{const}\, N^{-1/2}$$

**Fixed subspace sampling:** Algorithm needs to specify one subspace $V_n$ at first and sampling is only possible in $V_n$ at cost $n$.

$$e^{\mathrm{fixed}}(N) \geq \mathrm{const}\, N^{-1/4}\, (\log N)^{-3/4}$$

**Full space sampling:** Algorithm is allowed to sample arbitrarily and each evaluation of $f$ costs one unit.

$$e^{\mathrm{free}}(N) \geq \mathrm{const}\, N^{-1/2} (\log N)^{-3/2}$$

**Comments:** The first two lower bounds are sharp up to logarithms and there exist algorithms that reach the lower bounds up to logarithms under a reasonable notion of "cost" (runtime):

- Fixed subspace sampling: classical Monte Carlo with Euler
- Variable subspace sampling: multilevel Monte Carlo with Euler

# II Remarks on the proofs

The proofs are based on

- average Kolmogorov widths of diffusions in $V = (C[0,1], \|\cdot\|_\infty)$ that is the asymptotics of

$$d_n := \inf\{\mathbb{E}[d(Y, V_0)] : V_0 \subset V, \dim(V_0) \le 2^{n-1}\}$$

- an idea taken from Bakhvalov ('59) for proving lower bounds
- a connection to quantization complexity

# II Remarks on the proofs

The proofs are based on

- average Kolmogorov widths of diffusions in $V = (C[0,1], \|\cdot\|_\infty)$ that is the asymptotics of

$$d_n := \inf\{\mathbb{E}[d(Y, V_0)] : V_0 \subset V, \dim(V_0) \le 2^{n-1}\}$$

- an idea taken from Bakhvalov ('59) for proving lower bounds
- a connection to quantization complexity

**Comments:**

- General Gaussian measures are included in the analysis
- The results are sharp up to powers of logarithms
  $\rightsquigarrow$ Multilevel Monte Carlo algorithms

# III Multilevel Monte Carlo algorithms

**Ref.:** Heinrich '98, Giles '08

**Aim:** Computation of $S(f) := \mathbb{E}[f(Y)]$ for an implicit random element $Y$ attaining values in a normed space $(V, \|\cdot\|)$
(e.g., $Y = (Y_t)_{t\in[0,1]}$ solution of SDE)

# III Multilevel Monte Carlo algorithms

**Ref.:** Heinrich '98, Giles '08

**Aim:** Computation of $S(f) := \mathbb{E}[f(Y)]$ for an implicit random element $Y$ attaining values in a normed space $(V, \|\cdot\|)$ (e.g., $Y = (Y_t)_{t\in[0,1]}$ solution of SDE)

**Approximate solutions:** $Y^{(1)}, Y^{(2)}, \dots$
(e.g., $Y^{(k)}$ Euler approximation with step size $2^{-k}$)

# III Multilevel Monte Carlo algorithms

**Ref.:** Heinrich '98, Giles '08

**Aim:** Computation of $S(f) := \mathbb{E}[f(Y)]$ for an implicit random element $Y$ attaining values in a normed space $(V, \|\cdot\|)$ (e.g., $Y = (Y_t)_{t\in[0,1]}$ solution of SDE)

**Approximate solutions:** $Y^{(1)}, Y^{(2)}, \dots$
(e.g., $Y^{(k)}$ Euler approximation with step size $2^{-k}$)

**Classical Monte Carlo:** $Y \approx Y^{(m)}$
Approximate $Y$ by $Y^{(m)}$ and use Monte Carlo to approximate $\mathbb{E}[f(Y^{(m)})]$.

# III Multilevel Monte Carlo algorithms

**Ref.:** Heinrich '98, Giles '08

**Aim:** Computation of $S(f) := \mathbb{E}[f(Y)]$ for an implicit random element $Y$ attaining values in a normed space $(V, \|\cdot\|)$
(e.g., $Y = (Y_t)_{t\in[0,1]}$ solution of SDE)

**Approximate solutions:** $Y^{(1)}, Y^{(2)}, \dots$
(e.g., $Y^{(k)}$ Euler approximation with step size $2^{-k}$)

**Classical Monte Carlo:** $Y \approx Y^{(m)}$
Approximate $Y$ by $Y^{(m)}$ and use Monte Carlo to approximate $\mathbb{E}[f(Y^{(m)})]$.

**Multilevel scheme:** $Y \approx Y^{(m)} \longleftrightarrow Y^{(m-1)} \longleftrightarrow \dots \longleftrightarrow Y^{(2)} \longleftrightarrow Y^{(1)}$
Use Monte Carlo to approximate $\mathbb{E}[f(Y^{(k)}) - f(Y^{(k-1)})]$.

# III Multilevel Monte Carlo algorithms

**Ref.:** Heinrich '98, Giles '08

**Aim:** Computation of $S(f) := \mathbb{E}[f(Y)]$ for an implicit random element $Y$ attaining values in a normed space $(V, \|\cdot\|)$
(e.g., $Y = (Y_t)_{t\in[0,1]}$ solution of SDE)

**Approximate solutions:** $Y^{(1)}, Y^{(2)}, \dots$
(e.g., $Y^{(k)}$ Euler approximation with step size $2^{-k}$)

**Classical Monte Carlo:** $Y \approx Y^{(m)}$
Approximate $Y$ by $Y^{(m)}$ and use Monte Carlo to approximate $\mathbb{E}[f(Y^{(m)})]$.

**Multilevel scheme:** $Y \approx Y^{(m)} \longleftrightarrow Y^{(m-1)} \longleftrightarrow \dots \longleftrightarrow Y^{(2)} \longleftrightarrow Y^{(1)}$
Use Monte Carlo to approximate $\mathbb{E}[f(Y^{(k)}) - f(Y^{(k-1)})]$.

**Features:**
- scheme very efficient for diffusions, Gaussian processes, ...
- often errors of order $\mathcal{O}(N^{-1/2})$
- in the setting of part II: $\mathcal{O}(N^{-1/2}(\log N)^{3/2})$
- scheme is robust: applicable under weak assumptions, good performance for discontinuous $f$

# III The MLMC algorithm

**Telescoping sum:**

$$\mathbb{E}[f(Y^{(m)})] = \sum_{k=2}^{m} \mathbb{E}[f(Y^{(k)}) - f(Y^{(k-1)})] + \mathbb{E}[f(Y^{(1)})].$$

**Approximate** each expectation $\mathbb{E}[f(Y^{(k)}) - f(Y^{(k-1)})]$, resp. $\mathbb{E}[f(Y^{(1)})]$, by $n_k$ independent Monte-Carlo simulations.

**Output:** $\widehat{S}(f)$ = sum of the individual random approximations.

# III The MLMC algorithm

**Telescoping sum:**

$$\mathbb{E}[f(Y^{(m)})] = \sum_{k=2}^{m} \mathbb{E}[f(Y^{(k)}) - f(Y^{(k-1)})] + \mathbb{E}[f(Y^{(1)})].$$

**Approximate** each expectation $\mathbb{E}[f(Y^{(k)}) - f(Y^{(k-1)})]$, resp. $\mathbb{E}[f(Y^{(1)})]$, by $n_k$ independent Monte-Carlo simulations.

**Output:** $\widehat{S}(f) =$ sum of the individual random approximations.

**Error estimate (mse):**

$$\mathbb{E}\left[\left(S(f) - \widehat{S}(f)\right)^2\right] = \left|\mathbb{E}[f(Y)] - \mathbb{E}[f(Y^{(m)})]\right|^2 + \sum_{k=2}^{m} \frac{1}{n_k} \mathrm{var}(f(Y^{(k)}) - f(Y^{(k-1)})) + \frac{1}{n_1} \mathrm{var}(f(Y^{(1)})).$$

**Advantage:** only few of the expensive simulations are necessary.

# III Error estimates on Lip-class

If $f$ is Lipschitz continuous with coefficient one, then

$$\mathbb{E}\big[\big(S(f)-\widehat{S}(f)\big)^2\big] \leq \mathcal{W}(\mathbb{P}_Y,\mathbb{P}_{Y^{(m)}})^2 + \mathrm{const}\sum_{k=1}^{m}\frac{1}{n_k}\mathbb{E}[\|Y-Y^{(k)}\|^2],$$

where $\mathcal{W}$ denotes the Wasserstein metric

$$\mathcal{W}(Q_1,Q_2)=\inf\Big\{\int \|x-y\|\,\mathrm{d}\xi(x,y) : Q_1=\pi_1(\xi), Q_2=\pi_2(\xi)\Big\}$$

# III Error estimates on Lip-class

If $f$ is Lipschitz continuous with coefficient one, then

$$\mathbb{E}\left[\left(S(f)-\widehat{S}(f)\right)^2\right] \leq \mathcal{W}(\mathbb{P}_Y, \mathbb{P}_{Y^{(m)}})^2 + \mathrm{const} \sum_{k=1}^{m} \frac{1}{n_k} \mathbb{E}[\|Y - Y^{(k)}\|^2],$$

where $\mathcal{W}$ denotes the Wasserstein metric

$$\mathcal{W}(Q_1, Q_2) = \inf\left\{\int \|x-y\| \,\mathrm{d}\xi(x,y) : Q_1 = \pi_1(\xi), Q_2 = \pi_2(\xi)\right\}$$

**Note:** The error estimate does not depend on $f$ and we need estimates for

(W) $\mathcal{W}(\mathbb{P}_Y, \mathbb{P}_{Y^{(m)}})$ (weak approximation)

(S) $\mathbb{E}[\|Y - Y^{(k)}\|^2]$ (strong approximation)

# III Complexity theorem

**Assumptions:** There exist $0 < \beta \leq 2\alpha$ and a real constant $c$ s.th.

(W) $\mathcal{W}(\mathbb{P}_Y, \mathbb{P}_{Y^{(m)}}) \leq c(2^{-m})^{\alpha}$

(S) $\mathbb{E}[\|Y - Y^{(k)}\|^2] \leq c(2^{-m})^{\beta}$

Further we assign each joint simulation of $f(Y^{(k)}) - f(Y^{(k-1)})$ the cost $c\,2^k$.

# III Complexity theorem

**Assumptions:** There exist $0 < \beta \leq 2\alpha$ and a real constant $c$ s.th.

(W) $\mathcal{W}(\mathbb{P}_Y, \mathbb{P}_{Y^{(m)}}) \leq c(2^{-m})^{\alpha}$

(S) $\mathbb{E}[\|Y - Y^{(k)}\|^2] \leq c(2^{-m})^{\beta}$

Further we assign each joint simulation of $f(Y^{(k)}) - f(Y^{(k-1)})$ the cost $c\,2^k$.

**Theorem:** (Giles '08) Fixing the parameters (highest level $m$ and iteration numbers $n_1, \dots, n_m$) appropriately one obtains a sequence of MLMC algorithms $(\widehat{\mathcal{S}}_N : N \in \mathbb{N})$ each having cost less than or equal to $N$ and satisfying

$$\mathrm{err}(\widehat{\mathcal{S}}_N) = \sup_{f \in \mathrm{Lip}_1} \mathbf{E}[(S(f) - \widehat{S}_N(f))^2]^{1/2} \leq \mathrm{const} \begin{cases} N^{-1/2}, & \beta > 1 \\ N^{-1/2}(\log N)^{1/2}, & \beta = 1 \\ N^{-\frac{\alpha}{1+2\alpha-\beta}}, & \beta < 1, \end{cases}$$

where $\mathrm{const}$ is a constant only depending on $\alpha, \beta$ and $c$.

# III Complexity theorem

**Assumptions:** There exist $0 < \beta \leq 2\alpha$ and a real constant $c$ s.th.

(W) $\mathcal{W}(\mathbb{P}_Y, \mathbb{P}_{Y^{(m)}}) \leq c(2^{-m})^\alpha$

(S) $\mathbb{E}[\|Y - Y^{(k)}\|^2] \leq c(2^{-m})^\beta$

Further we assign each joint simulation of $f(Y^{(k)}) - f(Y^{(k-1)})$ the cost $c\,2^k$.

**Theorem:** (Giles '08) Fixing the parameters (highest level $m$ and iteration numbers $n_1, \dots, n_m$) appropriately one obtains a sequence of MLMC algorithms $(\widehat{\mathcal{S}}_N : N \in \mathbb{N})$ each having cost less than or equal to $N$ and satisfying

$$\mathrm{err}(\widehat{\mathcal{S}}_N) = \sup_{f \in \mathrm{Lip}_1} \mathbf{E}[(S(f) - \widehat{S}_N(f))^2]^{1/2} \leq \mathrm{const} \begin{cases} N^{-1/2}, & \beta > 1 \\ N^{-1/2}(\log N)^{1/2}, & \beta = 1 \\ N^{-\frac{\alpha}{1+2\alpha-\beta}}, & \beta < 1, \end{cases}$$

where $\mathrm{const}$ is a constant only depending on $\alpha, \beta$ and $c$.

**Conversely:** Appropriate classical Monte Carlo algorithms $(\widehat{\mathcal{S}}_N : N \in \mathbb{N})$ lead to

$$\mathrm{err}(\widehat{\mathcal{S}}_N) \leq \mathrm{const}\, N^{-\frac{\alpha}{1+2\alpha}}.$$

# IV Lévy process $X = (X_t)_{t\in[0,1]}$ ($L^2$-integrable)

**Now:** MLMC for Lévy-driven stochastic differential equations!

**Lévy process:** discontinuous extension of the Wiener process in the sense that

- $X_{t_1}, \dots, X_{t_n} - X_{t_{n-1}}$ are independent $(0 \le t_1 \le \dots \le t_n)$
  independent increments
- $X_t - X_s \overset{\mathcal{L}}{=} X_{t-s}$ $(0 \le s \le t)$
  stationary increments
- $X$ is a.s. càdlàg (right cont. with left hand limits)

# IV Lévy process $X = (X_t)_{t\in[0,1]}$ ($L^2$-integrable)

**Now:** MLMC for Lévy-driven stochastic differential equations!

**Lévy process:** discontinuous extension of the Wiener process in the sense that

- $X_{t_1}, \dots, X_{t_n} - X_{t_{n-1}}$ are independent ($0 \le t_1 \le \dots \le t_n$)
  independent increments
- $X_t - X_s \stackrel{\mathcal{L}}{=} X_{t-s}$ ($0 \le s \le t$)
  stationary increments
- $X$ is a.s. càdlàg (right cont. with left hand limits)

**Lévy-Itô decompositio:** The Lévy process $X$ is a combination of

- a Wiener process, parameterized via its symmetric covariance $\Sigma\Sigma^* \in \mathbb{R}^{d_X \times d_X}$
- a drift, parameterized via the trend $b \in \mathbb{R}^{d_X}$, and
- a compensated pure jump process ($L^2$-martingale), parameterized via the jump intensity $\nu$ (Lévy measure), a measure on $\mathbb{R}^{d_X}\backslash\{0\}$ with

$$\int |x|^2 \, \nu(\mathrm{d}x) < \infty.$$

# IV Lévy driven SDEs

**SDE:** $Y = (Y_t)_{t\in[0,1]}$ solution to

$$Y_t = y_0 + \int_0^t a(Y_{s-})\,\mathrm{d}X_s,$$

where $a : \mathbb{R}^{d_Y} \to \mathbb{R}^{d_Y \times d_X}$ is Lipschitz continuous.

**Simulation** of Lévy processes is based on the Lévy-Itô decomposition:

$$X_t = \Sigma W_t + bt + L_t$$

where $W$ is an $\mathbb{R}^{d_X}$-dimensional standard Wiener process and $L$ ist the compensated pure jump part.

# IV Lévy driven SDEs

**SDE:** $Y = (Y_t)_{t\in[0,1]}$ solution to

$$Y_t = y_0 + \int_0^t a(Y_{s-})\,\mathrm{d}X_s,$$

where $a : \mathbb{R}^{d_Y} \to \mathbb{R}^{d_Y\times d_X}$ is Lipschitz continuous.

**Simulation** of Lévy processes is based on the Lévy-Itô decomposition:

$$X_t = \Sigma W_t + bt + L_t$$

where $W$ is an $\mathbb{R}^{d_X}$-dimensional standard Wiener process and $L$ ist the compensated pure jump part.

# IV Lévy driven SDEs

**SDE:** $Y = (Y_t)_{t\in[0,1]}$ solution to

$$Y_t = y_0 + \int_0^t a(Y_{s-})\,\mathrm{d}X_s,$$

where $a : \mathbb{R}^{d_Y} \to \mathbb{R}^{d_Y \times d_X}$ is Lipschitz continuous.

**Simulation** of Lévy processes is based on the Lévy-Itô decomposition:

$$X_t = \Sigma W_t + bt + L_t$$

where $W$ is an $\mathbb{R}^{d_X}$-dimensional standard Wiener process and $L$ ist the compensated pure jump part.

# IV Lévy driven SDEs

**SDE:** $Y = (Y_t)_{t\in[0,1]}$ solution to

$$Y_t = y_0 + \int_0^t a(Y_{s-})\,\mathrm{d}X_s,$$

where $a : \mathbb{R}^{d_Y} \to \mathbb{R}^{d_Y \times d_X}$ is Lipschitz continuous.

**Simulation** of Lévy processes is based on the Lévy-Itô decomposition:

$$X_t = \Sigma W_t + bt + L_t$$

where $W$ is an $\mathbb{R}^{d_X}$-dimensional standard Wiener process and $L$ ist the compensated pure jump part.

# IV Lévy driven SDEs

**SDE:** $Y = (Y_t)_{t\in[0,1]}$ solution to

$$Y_t = y_0 + \int_0^t a(Y_{s-})\,\mathrm{d}X_s,$$

where $a : \mathbb{R}^{d_Y} \to \mathbb{R}^{d_Y \times d_X}$ is Lipschitz continuous.

**Simulation** of Lévy processes is based on the Lévy-Itô decomposition:

$$X_t = \Sigma W_t + bt + L_t$$

where $W$ is an $\mathbb{R}^{d_X}$-dimensional standard Wiener process and $L$ ist the compensated pure jump part.

**Problem:** If $\nu$ is an infinite measure, then the discontinuities of $L$ are dense in $[0,1]$. One can only simulate the jumps being larger than a threshold and perfect simulation of increments is typically not feasible!

# IV Lévy driven SDEs

**SDE:** $Y = (Y_t)_{t\in[0,1]}$ solution to

$$Y_t = y_0 + \int_0^t a(Y_{s-})\, \mathrm{d}X_s,$$

where $a : \mathbb{R}^{d_Y} \to \mathbb{R}^{d_Y \times d_X}$ is Lipschitz continuous.

**Simulation** of Lévy processes is based on the Lévy-Itô decomposition:

$$X_t = \Sigma W_t + bt + L_t$$

where $W$ is an $\mathbb{R}^{d_X}$-dimensional standard Wiener process and $L$ ist the compensated pure jump part.

**Problem:** If $\nu$ is an infinite measure, then the discontinuities of $L$ are dense in $[0, 1]$. One can only simulate the jumps being larger than a threshold and perfect simulation of increments is typically not feasible!

**Idea:** Compensate the "small jumps" by a Gaussian correction (Asmussen, Rosiński '01)

# IV Approximations for $L$

**Thresholds:** We choose thresholds $h_1 > h_2 > \cdots > 0$ that are small and satisfy

$$\nu(B(0, h_k)^c) \leq 2^k.$$

Further, let $\varepsilon_k = 2^{-k}$.

# IV Approximations for $L$

**Thresholds:** We choose thresholds $h_1 > h_2 > \cdots > 0$ that are small and satisfy

$$\nu(B(0, h_k)^c) \leq 2^k.$$

Further, let $\varepsilon_k = 2^{-k}$.

**Point process representation for $L$:**

$$L_t = L^2\text{-}\lim_{h \downarrow 0} \int_{B(0,h)^c \times (0,t]} x \, \mathrm{d}(\xi - \bar{\nu})(\mathrm{d}x, \mathrm{d}s)$$

where $\xi$ is a Poisson point process with intensity $\bar{\nu} = \nu \otimes \lambda$. As approximation we choose

$$L^{(k)} = \int_{B(0,h_k)^c \times (0,t]} x \, \mathrm{d}(\xi - \bar{\nu})(\mathrm{d}x, \mathrm{d}s)$$

# IV Approximations for $L$

**Thresholds:** We choose thresholds $h_1 > h_2 > \cdots > 0$ that are small and satisfy

$$\nu(B(0, h_k)^c) \leq 2^k.$$

Further, let $\varepsilon_k = 2^{-k}$.

**Point process representation for $L$:**

$$L_t = L^2\text{-}\lim_{h \downarrow 0} \int_{B(0,h)^c \times (0,t]} x \, \mathrm{d}(\xi - \bar{\nu})(\mathrm{d}x, \mathrm{d}s)$$

where $\xi$ is a Poisson point process with intensity $\bar{\nu} = \nu \otimes \lambda$. As approximation we choose

$$L^{(k)} = \int_{B(0,h_k)^c \times (0,t]} x \, \mathrm{d}(\xi - \bar{\nu})(\mathrm{d}x, \mathrm{d}s)$$

**Joint simulation of two levels $L^{(k)}, L^{(k-1)}$:** Simulate the restricted point process $\xi|_{B(0,h_k)^c \times (0,1]}$ which consists in the average of

$$\bar{\nu}(B(0, h_k)^c \times (0, 1]) = \int_{B(0,h_k)} \nu(\mathrm{d}x)$$

points.

# IV Euler type approximate solutions

**Updates:** We denote by $\mathbb{T}_k$ the random set of times that contains 0 and 1 and all times $t \in (0, 1)$ with

$$\Delta L_t := L_t - L_{t-} \in B(0, h_k)^c \quad \text{or} \quad [t - \varepsilon_k, t) \cap \mathbb{T}_k = \{t - \varepsilon_k\}$$

$\mathbb{T}_k$ contains all discontinuities of $L^{(k)}$ and further points are added to ensure that updates are not ore than $\varepsilon_k$ time units apart.

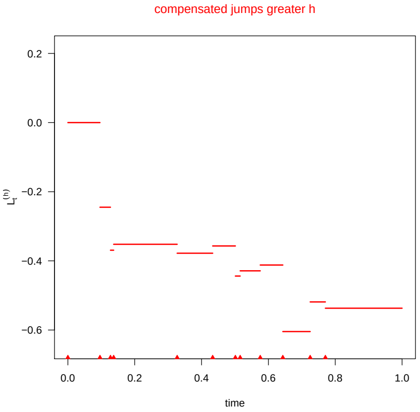
**Approximate solutions:** we set $Y_0^{(k)} = y_0$ and for neighboring times $t < t'$ in $\mathbb{T}_k$, we set

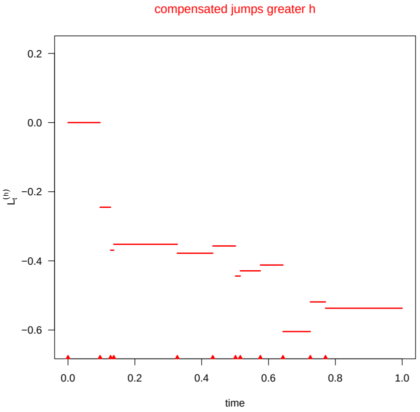
$$Y_{t'}^{(k)} = Y_t^{(k)} + a(Y_t^{(k)})(X_{t'}^{(k)} - X_t^{(k)})$$

where $X_t^{(k)} = W_t + bt + L_t^{(k)}$

# IV Visualisation

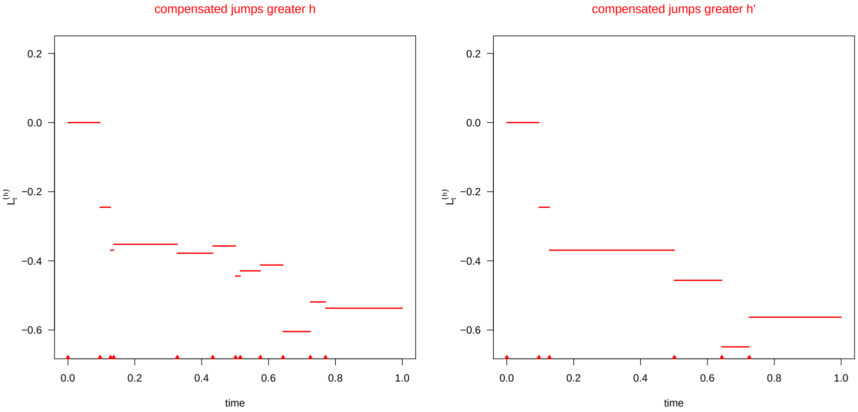
Simulation of $(L^{(k)}, L^{(k-1)})$



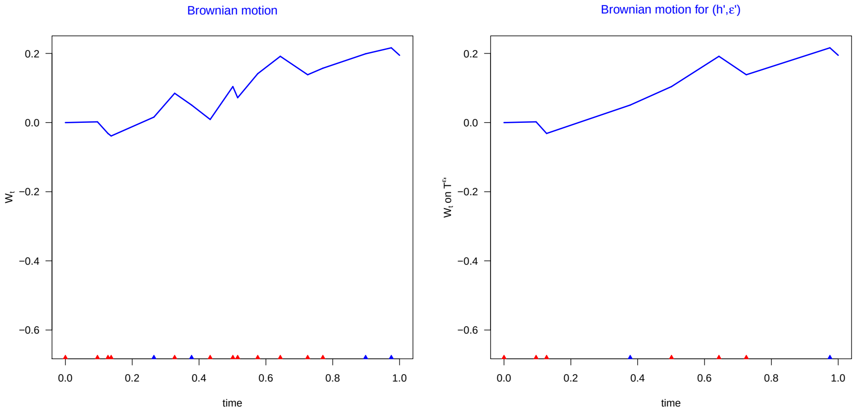
# IV Visualisation

Simulation of $(L^{(k)}, L^{(k-1)})$



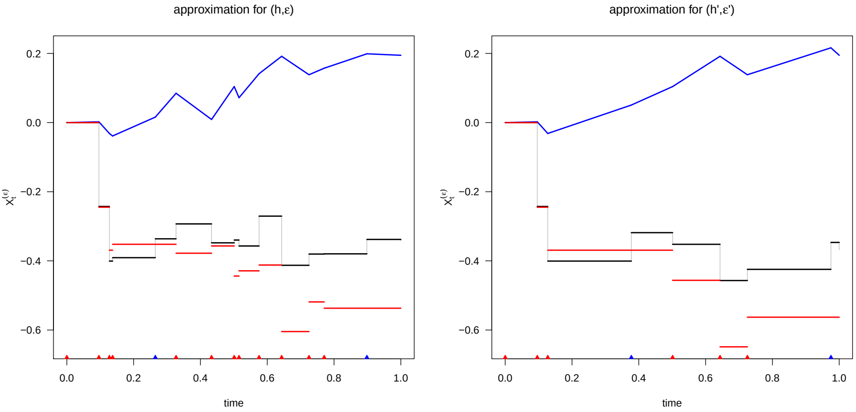
# IV Visualisation

Simulation of $(L^{(k)}, L^{(k-1)}),\ W$



# IV Visualisation

Simulation of $(L^{(k)}, L^{(k-1)})$, $W$ and $(X^{(k)}, X^{(k-1)})$.



# IV Classes of algorithms

**Algorithms** $\widehat{\mathcal{S}}$ are specified via the parameters:

- $m \in \mathbb{N}$ : # of levels
- $n_1, \ldots, n_m$ : # of simulations of pairs $(Y^{(k)}, Y^{(k-1)})$

# IV Classes of algorithms

**Algorithms** $\widehat{S}$ are specified via the parameters:

- $m \in \mathbb{N}$ : # of levels
- $n_1, \dots, n_m$ : # of simulations of pairs $(Y^{(k)}, Y^{(k-1)})$

**Class** $\mathcal{A}_0$**:** (MLMC0, neglect small jumps)
Approximation $Y^{(k)}$ is obtained via $\mathbb{T}_k$-Euler scheme with driving process

$$X_t^{(k)} = \Sigma W_t + L_t^{(k)} + bt,$$

where $L^{(k)}$ is constituted by the compensated jumps of $L$ larger than $h_k$.

## IV Classes of algorithms

**Algorithms** $\widehat{S}$ are specified via the parameters:

- $m \in \mathbb{N}$ : # of levels
- $n_1, \ldots, n_m$ : # of simulations of pairs $(Y^{(k)}, Y^{(k-1)})$

**Class $\mathcal{A}_0$:** (MLMC0, neglect small jumps)
Approximation $Y^{(k)}$ is obtained via $\mathbb{T}_k$-Euler scheme with driving process

$$X_t^{(k)} = \Sigma W_t + L_t^{(k)} + bt,$$

where $L^{(k)}$ is constituted by the compensated jumps of $L$ larger than $h_k$.

**Class $\mathcal{A}_1$:** (MLMC1, Gaussian compensation)
Approximation $Y^{(k)}$ is obtained via $\mathbb{T}_k$-Euler scheme with driving process

$$X_t^{(k)} = \Sigma W_t + \Sigma^{(m)} B_t + L_t^{(k)} + bt,$$

where $B$ is an independent Wiener process and

$$\Sigma^{(m)}(\Sigma^{(m)})^* = \int_{B(0,h_m)} x \otimes x \, \nu(\mathrm{d}x)$$

# IV Error estimates

We express the asymptotic estimates in terms of the

**Blumenthal-Getoor index:**

$$\alpha := \inf\left\{ p > 0 : \int_{B(0,1)} |x|^p \, \nu(\mathrm{d}x) < \infty \right\} \in [0, 2]$$

# IV Error estimates

We express the asymptotic estimates in terms of the

**Blumenthal-Getoor index:**

$$\alpha := \inf\left\{p > 0 : \int_{B(0,1)} |x|^p \, \nu(\mathrm{d}x) < \infty\right\} \in [0,2]$$

**Main Result:** (D, Heidenreich '11, D '11)
For $i = 0, 1$, there exist multilevel Monte Carlo algorithms $\{\widehat{\mathcal{S}}_N^i : N \in \mathbb{N}\}$ in $\mathcal{A}_i$ with $\mathrm{cost}(\widehat{\mathcal{S}}_N^i) \leq N$ and

$$\mathrm{err}(\widehat{\mathcal{S}}_N^i) \leq N^{-(1+o(1))\varphi_i(\alpha)}$$

# IV Error estimates

We express the asymptotic estimates in terms of the

**Blumenthal-Getoor index:**

$$\alpha := \inf\left\{p > 0 : \int_{B(0,1)} |x|^p \, \nu(\mathrm{d}x) < \infty\right\} \in [0,2]$$

**Main Result:** (D, Heidenreich '11, D '11)
For $i = 0, 1$, there exist multilevel Monte Carlo algorithms $\{\widehat{\mathcal{S}}^i_N : N \in \mathbb{N}\}$ in $\mathcal{A}_i$ with $\mathrm{cost}(\widehat{\mathcal{S}}^i_N) \le N$ and

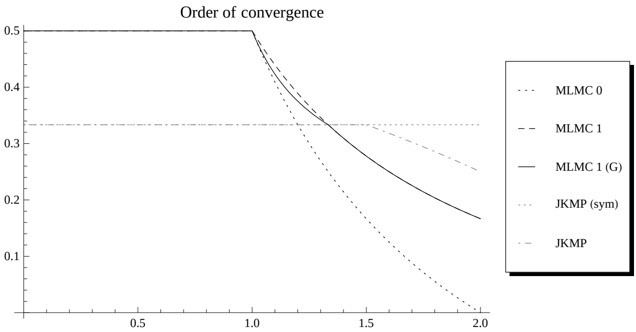
$$\mathrm{err}(\widehat{\mathcal{S}}^i_N) \le N^{-(1+o(1))\varphi_i(\alpha)}$$

for

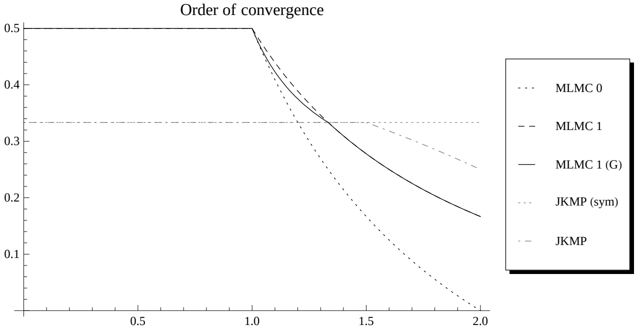
- $\varphi_0(\alpha) = (\frac{1}{\alpha} - \frac{1}{2}) \wedge \frac{1}{2}$
- $\varphi_1(\alpha) = \frac{4-\alpha}{6\alpha} \wedge \frac{1}{2}$ if $\Sigma = 0$ or $\alpha \notin [1, \frac{4}{3}]$
- $\varphi_1(\alpha) = \frac{\alpha}{6\alpha-4}$ if $\Sigma \neq 0$ and $\alpha \in [1, \frac{4}{3}]$.

**Note:** The analysis of $\mathcal{A}_1$ requires a uniform ellipticity assumption on $\nu$.

# IV Error estimates



# IV Error estimates




**Related work on quadrature of marginals:**

- Jacod, Kurtz, Méléard, and Protter '05
- Tanaka and Kohatsu-Higa '09

# IV Remarks on the proofs

Recall that we need estimates for

(W) $\mathcal{W}(\mathbb{P}_Y, \mathbb{P}_{Y^{(m)}})$ (weak approximation)

(S) $\mathbb{E}[\|Y - Y^{(k)}\|^2]$ (strong approximation)

**Proof for class $\mathcal{A}_0$:**

- Control (S) of Euler scheme (as for classical diffusions)
- This gives also an upper bound for (W).
- Balance errors.

# IV Remarks on the proofs

Recall that we need estimates for

(W) $\mathcal{W}(\mathbb{P}_Y, \mathbb{P}_{Y^{(m)}})$ (weak approximation)

(S) $\mathbb{E}[\|Y - Y^{(k)}\|^2]$ (strong approximation)

**Proof for class $\mathcal{A}_0$:**

- Control (S) of Euler scheme (as for classical diffusions)
- This gives also an upper bound for (W).
- Balance errors.

**Proof for class $\mathcal{A}_1$:**

- New estimate for (W) by applying a KMT-like coupling (Zaitsev '98) for small jump part, say $L'$.

Problem: Coupling yields small error in the supremum norm; however this does not allow to control the error in the differential equation directly.

Remedy: Apply independent couplings on consecutive intervals and ignore the impact of small jumps at most update times.

# IV Consequences of Zaitsev's result (KMT)

**Notation:**

- $L$: compensated pure jump process with intensity $\nu$ being supported on $B(0,h)$
- $\Sigma\Sigma^* = \int x \otimes x \, \nu(\mathrm{d}x)$
- $B$: Wiener process

**Theorem:** One can couple $(L_t)_{t\in[0,T]}$ and $(\Sigma B_t)_{t\in[0,T]}$ such that

$$\mathbb{E}[\sup_{t\in[0,T]} |L_t - \Sigma B_t|^2]^{1/2} \le \sqrt{\gamma} h \Big( c_1 \log\Big(\frac{\sigma^2 T}{h^2} \vee e\Big) + c_2 \Big),$$

where

- $\sigma^2 = \int_{B(0,h)} |x|^2 \, \nu(\mathrm{d}x)$ and
- $\gamma \ge 1$ is such that $\int \langle y', x\rangle^2 \, \nu(\mathrm{d}x) \le \gamma \int \langle y, x\rangle^2 \, \nu(\mathrm{d}x)$ for $|x| = |y| = 1$ ($\to$ uniform ellipticity assumption)

**Consequence:** For quadrature of Lévy processes, one has algorithms $(\widehat{\mathcal{S}}_N : N \in \mathbb{N})$ with

$$\mathrm{err}(\widehat{\mathcal{S}}_N) \le \mathrm{const}\, N^{-(1+o(1))\frac{1}{2\alpha}}$$

# IV Comments

- Worst case error bounds over the class of Lipschitz functions $f$ w.r.t. supremum norm
- Weak assumptions on coefficient $a$
- Explicit representation for thresholds $h_k$ in terms of the Lévy measure $\nu$
- Improved rates can be proved if $f$ depends only on marginals
- Numerical implementation have been conducted by F. Heidenreich (TU Kaiserslautern)
- Information retrieved from Monte Carlo on low levels can be used to interpolate and to improve the performance.
- One gets fast convergence rates for the quadrature of Lévy processes.

# V Numerical experiments

In the numerical test we consider

- a one dimensional Lévy process $X$ with characteristics $\Sigma = b = 0$ and

$$\frac{\mathrm{d}\nu}{\mathrm{d}x}(x) = \mathbb{1}_{(0,1]}(|x|)\frac{0.1}{|x|^{1+\alpha}},$$

  where $\alpha \in (0,2)$ denotes the Blumenthal-Getoor index
- the SDE

$$Y_t = 1 + \int_0^t Y_{s-}\,\mathrm{d}X_s$$

- a *lookback option* with strike 1, that is

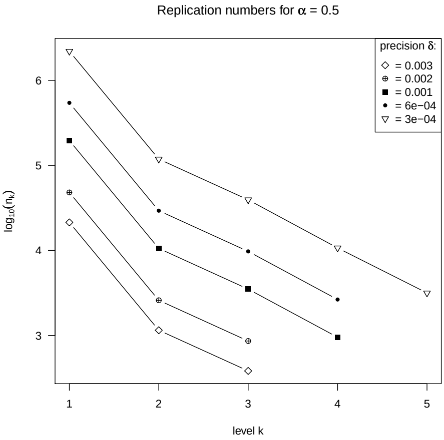
$$f(Y) = (\sup_{t\in[0,1]} Y_t - 1)^+.$$

So far only results for multilevel without Gaussian compensation are available.

# V Adaptive choice of $m$ and $n_1, \ldots, n_m$

Expample of $n_1, \ldots, n_m$ with $\alpha = 0.5$.

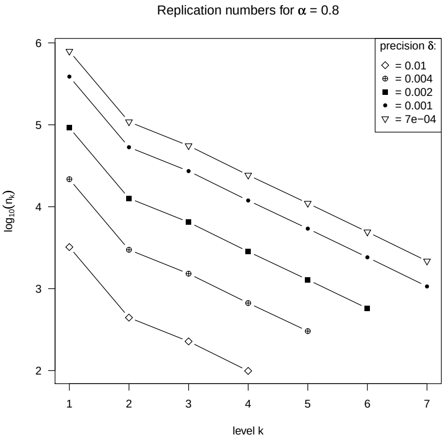
- Precisions $\delta = (0.003, 0.002, 0.001, 0.0006, 0.0003)$.
- Highest levels $m = (3, 3, 4, 4, 5)$.



# V Adaptive choice of $m$ and $n_1, \ldots, n_m$

Expample of $n_1, \ldots, n_m$ with $\alpha = 0.8$.

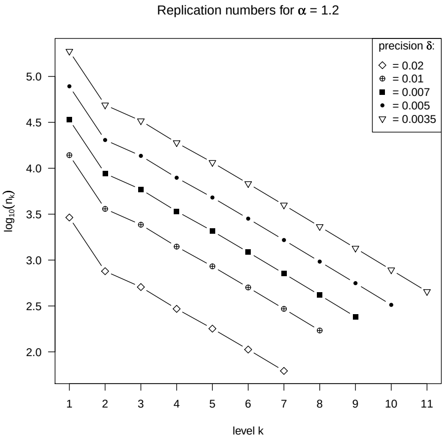
- Precisions $\delta = (0.01, 0.004, 0.002, 0.001, 0.0007)$.
- Highest levels $m = (4, 5, 6, 7, 7)$.



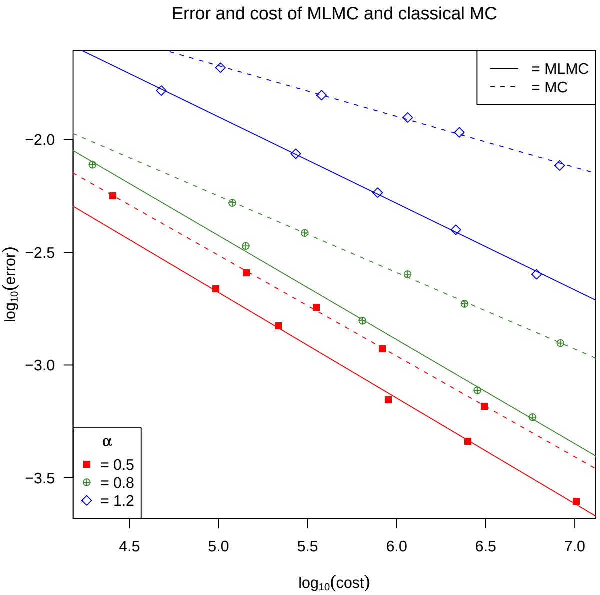
# V Adaptive choice of $m$ and $n_1, \dots, n_m$

Expample of $n_1, \dots, n_m$ with $\alpha = 1.2$.

- Precisions $\delta = (0.02, 0.01, 0.007, 0.005, 0.0035)$.
- Highest levels $m = (7, 8, 9, 10, 11)$.



# V Error versus cost



# V Empirical versus theoretical findings

**Comparison of the empirical findings**

| BG index $\alpha$ | 0.5 | 0.8 | 1.2 |
|---|---|---|---|
| Theoretical order (MLMC) | 0.5 | 0.5 | 0.33 |
| Empirical order (MLMC) | 0.47 | 0.46 | 0.38 |
| Empirical order (MC) | 0.45 | 0.34 | 0.23 |

# V Bias/variance estimates

**Problem:** The theoretic bias estimates are often too big which means that too many pairs of levels are included in the multilevel algorithm.

# V Bias/variance estimates

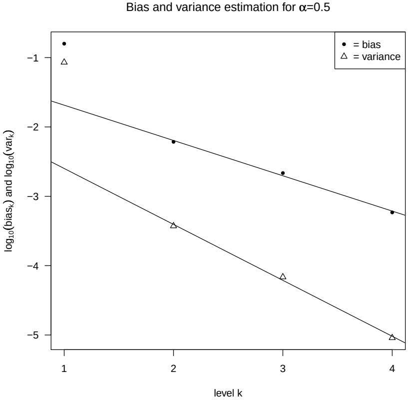
**Problem:** The theoretic bias estimates are often too big which means that too many pairs of levels are included in the multilevel algorithm.

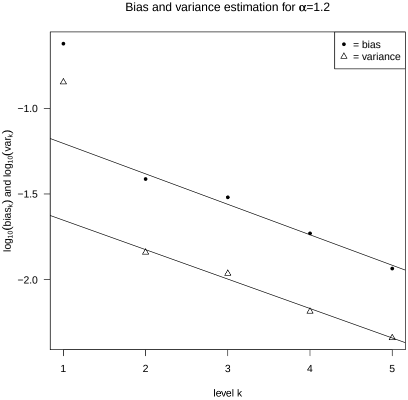
**Remedy:** The coarse levels have high iteration numbers so that we have good estimates for

$$\mathrm{bias}_k := \mathbb{E}[f(Y^{(k)}) - f(Y^{(k-1)})]$$

for small $k$, say for $k = 1, \ldots, 4$. Now we do a linear regression on a log-plot through the first 4 empirically observed bias estimates and extrapolate on the biases of the higher levels.

# V Bias/variance estimates



# Main references

J. Creutzig, S. Dereich, Th. Müller-Gronbach and Klaus Ritter, "Infinite-dimensional quadrature and approximation of distributions", *Found. Comput. Math.* **9**(4):391-429 (2009)

S. Dereich, "Multilevel Monte Carlo algorithms for Lévy-driven SDEs with Gaussian correction", Ann. Appl. Probab. **21**(1):283-311 (2011)

S. Dereich, F. Heidenreich, "A multilevel Monte Carlo algorithm for Lévy driven stochastic differential equations", Stochastic Process. Appl. **121**(7), 1565-1587 (2011)

M.B. Giles "Multilevel Monte Carlo path simulation", *Operations Research* **56**(3):607-617 (2008).

Thank you very much for your attention