Monte Carlo-based methods for the BENCHOP project

The BENCHmarking project in Option Pricing

Á. Leitao, S. Jain and C. W. Oosterlee

Delft University of Technology - Centrum Wiskunde & Informatica

Leiden, September 22, 2017

The BENCHOP project

- The purpose and aim of BENCHOP is to provide sets of benchmark problems in option pricing.
- Facilitating comparison and evaluation of different methods.
- Expecting that future papers in the financial field will compare method performances with the methods in BENCHOP.
- Contributing to a more uniform comparison and understanding of different methods' pros and cons.
- Results published in a journal articles.
- This is the second edition. The results of the first edition have been already published[1].

[1] Lina von Sydow et al. "BENCHOP – The BENCHmarking project in option pricing". In: *International Journal of Computer Mathematics* 92.12 (2015), pp. 2361–2379.

Aspects 2nd edition

- Implementation should be in Matlab.
- Preferable, use of high-performance features: parallel computing toolbox.
- Two categories:
  - Basket options.
  - Stochastic and local volatility models.
- Benchmark: Error (accuracy) in the solution as a function of CPU time.

Basket options - Problem formulation

- Underlying prices modelled by a multidimensional Merton model:

$$\frac{\mathrm{d}S_i(t)}{S_i(t)} = (r - \lambda\kappa_i)\mathrm{d}t + \mathrm{d}B_i(t) + \left(\mathrm{e}^{J_i(t)} - 1\right)\mathrm{d}P(t).$$

- $\mathrm{d}B_i(t), \quad i = 1, \ldots, d$ is a multidimensional Brownian motion with covariance matrix $\Sigma_{ij}^B = \sigma_i^B(S_i, t)\sigma_j^B(S_j, t)\rho_{ij}^B$.
- $P(t)$ is a Poisson process with the arrival rate λ.
- $J_i(t), \quad i = 1, \ldots, d$ follows a multivariate normal distribution with mean values μ_i^J and covariance matrix $\Sigma_{ij}^J = \sigma_i^J(S_i, t)\sigma_j^J(S_j, t)\rho_{ij}^J$.
- The expected jump of the ith component is

$$\kappa_i = \mathbb{E}\left[\mathrm{e}^{J_i(t)} - 1\right] = \exp\left(\mu_i^J + \frac{1}{2}\sum_{j=1}^{d}\sigma_i^J\sigma_j^J\rho_{ij}^J\right) - 1.$$

- When $\lambda = 0$ and σ_i constant: multi Black-Scholes model.

Basket options - Problems

- For all the problems: Price u.
- For some problems also: $\Delta = \frac{\partial u}{\partial S_i}$ and $\mathcal{V} = \frac{\partial u}{\partial \sigma_i}$

1. **European spread option**

$$g(S) = \max\{S_1 - S_2 - K, 0\},$$

with settings: GBM, $S_i = 100$, $r = 0.03$, $T = 1$, $\rho = 0.5$ and $K = 5$.
Two problems: constant volatility ($\sigma_i = 0.15$) or given by the function

$$\sigma_i(S_i, t) = 0.15 + 0.15(0.5 + 2t)\frac{(S_i/100 - 1.2)^2}{(S_i/100)^2 + 1.44}.$$

Basket options - Problems

2. **American put on the minimum of two assets**

$$g(S) = \max\left\{K - \min\left\{S_1, S_2\right\}, 0\right\},$$

with settings: $S_i = 40$, $r = 0.05$, $\sigma_i = 0.3$ $T = 0.5$, $\rho = 0.5$ and $K = 40$. Two problems: without jumps (Black-Scholes) or with jumps ($\mu_i^J = -0.5$, $\sigma_i^J = 0.4$, $\rho_{ij}^J = 0.5$ and $\lambda = 0.4$).

3. **Arithmetic basket options on 3 and 10 assets**

$$g(S) = \max\left\{K - \frac{1}{d}\sum_{i=1}^{d} S_i, 0\right\},$$

with settings: GBM, $S_i = 40$, $r = 0.06$, $\sigma_i = 0.2$, $T = 1$ and $K = 40$. Four problems: European/American and low constant correlation ($\rho = 0.25$), European/American high variable correlations ($\rho_{ij} = 0.9^{|i-j|}$).

Basket options - Problems

4. **European arithmetic basket options on four assets**

$$g(S) = \max\left\{K - \frac{1}{d}\sum_{i=1}^{d} S_i, 0\right\},$$

with settings: GBM, $S_i = 40$, $r = 0.06$, $\sigma_i = 0.3$, $T = 1$ and $K = 40$. Correlation matrix:

$$\rho = \begin{pmatrix} 1 & 0.3 & 0.4 & 0.5 \\ 0.3 & 1 & 0.2 & 0.25 \\ 0.4 & 0.2 & 1 & 0.3 \\ 0.5 & 0.25 & 0.3 & 1 \end{pmatrix}.$$

5. **European/American arithmetic basket options on five assets**

$$g(S) = \max\left\{K - \sum_{i=1}^{d} w_i S_i, 0\right\},$$

with settings: GBM, $S_i = 1$, $r = 0.05$,
$\sigma = [0.518, 0.648, 0.623, 0.570, 0.530]$,
$w = [0.381, 0.065, 0.057, 0.270, 0.227]$, $T = 1$ and $K = 1$. Correlation matrix:

$$\rho = \begin{pmatrix} 1 & 0.79 & 0.82 & 0.91 & 0.84 \\ 0.79 & 1 & 0.73 & 0.80 & 0.76 \\ 0.82 & 0.73 & 1 & 0.77 & 0.72 \\ 0.91 & 0.80 & 0.77 & 1 & 0.90 \\ 0.84 & 0.76 & 0.72 & 0.90 & 1 \end{pmatrix}.$$

Stochastic and local volatility - Problems

- European call options.
- Three prices: in-the-money, at-the-money and out-the-money.

1. **SABR model**

   The formal definition of the SABR model reads

$$\begin{aligned} \mathrm{d}S(t) &= \sigma(t)S^{\beta}(t)\mathrm{d}W_S(t), & S(0) &= S_0 \exp\left(rT\right), \\ \mathrm{d}\sigma(t) &= \alpha\sigma(t)\mathrm{d}W_\sigma(t), & \sigma(0) &= \sigma_0, \end{aligned}$$

   where $S(t) = \bar{S}(t)\exp\left(r(T-t)\right)$. Correlation between the Brownian motions, ρ. Two parameter sets:
   $T = 2,\ r = 0.0,\ S_0 = 0.5,\ \sigma_0 = 0.5,\ \alpha = 0.4,\ \beta = 0.5,\ \rho = 0.$
   $T = 10,\ r = 0.0,\ S_0 = 0.07,\ \sigma_0 = 0.4,\ \alpha = 0.8,\ \beta = 0.5,\ \rho = -0.6.$
   European call option payoff $(\max(S(T) - K_i(T), 0))$ with

$$K_i(T) = S(0)\exp(0.1 \times \sqrt{T} \times \delta_i),$$
$$\delta_i = -1.0, 0.0, 1.0.$$

Stochastic and local volatility - Problems

2. **Quadratic local stochastic volatility model**

$$\mathrm{d}S(t) = rS(t)\mathrm{d}t + \sqrt{V(t)}f(S(t))\mathrm{d}W_S(t),$$
$$\mathrm{d}V(t) = \kappa(\eta - V(t))\mathrm{d}t + \sigma\sqrt{V(t)}\mathrm{d}W_V(t),$$

   with $f(s) = \frac{1}{2}\alpha s^2 + \beta s + \gamma$. Two models:
   Heston($\alpha = 0, \beta = 1, \gamma = 0$) and QLSV($\alpha = 0.02, \beta = 0, \gamma = 0$).

3. **Heston-Hull-White model**

$$\mathrm{d}S(t) = R(t)S(t)\mathrm{d}t + \sqrt{V(t)}S(t)\mathrm{d}W_S(t),$$
$$\mathrm{d}V(t) = \kappa(\eta - V(t))\mathrm{d}t + \sigma_1\sqrt{V(t)}\mathrm{d}W_V(t),$$
$$\mathrm{d}R(t) = a(b - V(t))\mathrm{d}t + \sigma_2\mathrm{d}W_R(t).$$

Our contribution

- We propose Monte Carlo-based methods.
- For Basket options:
  - Stochastic Grid Bundling method (SGBM)[2].
- For SABR model:
  - The mSABR simulation scheme[3].
  - Multi-Level Monte Carlo (MLMC) SABR (BONUS).

[2] Shashi Jain and Cornelis W. Oosterlee. "The Stochastic Grid Bundling Method: Efficient pricing of Bermudan options and their Greeks". In: *Applied Mathematics and Computation* 269 (2015), pp. 412–431.

[3] Álvaro Leitao, Lech A. Grzelak, and Cornelis W. Oosterlee. "On an efficient multiple time step Monte Carlo simulation of the SABR model". In: *Quantitative Finance* 17.10 (2017), pp. 1549–1565.

Stochastic Grid Bundling Method

- Early-exercise option pricing method, particularly Bermudan.
- Dynamic programming approach.
- Simulation and regression-based method.
- Forward in time: Monte Carlo simulation.
- Backward in time: Early-exercise policy computation.
- Step I: Generation of stochastic grid points

$$\{\mathbf{S}_{t_0}(n), \ldots, \mathbf{S}_{t_M}(n)\}, \; n = 1, \ldots, N.$$

- Step II: Option value at terminal time $t_M = T$

$$V_{t_M}(\mathbf{S}_{t_M}) = \max(h(\mathbf{S}_{t_M}), 0).$$

Stochastic Grid Bundling Method

- Backward in time, t_m, $m \leq M$,:
- Step III: Bundling into ν non-overlapping sets or partitions

$$\mathcal{B}_{t_{m-1}}(1), \ldots, \mathcal{B}_{t_{m-1}}(\nu)$$

- Step IV: Parameterizing the option values

$$Z(\mathbf{S}_{t_m}, \alpha_{t_m}^{\beta}) \approx V_{t_m}(\mathbf{S}_{t_m}).$$

- Step V: Computing the continuation and option values at t_{m-1}

$$\widehat{Q}_{t_{m-1}}(\mathbf{S}_{t_{m-1}}(n)) = \mathbb{E}[Z(\mathbf{S}_{t_m}, \alpha_{t_m}^{\beta})|\mathbf{S}_{t_{m-1}}(n)].$$

The option value is then given by:

$$\widehat{V}_{t_{m-1}}(\mathbf{S}_{t_{m-1}}(n)) = \max(h(\mathbf{S}_{t_{m-1}}(n)), \widehat{Q}_{t_{m-1}}(\mathbf{S}_{t_{m-1}}(n))).$$

Stochastic Grid Bundling Method

- Given basis functions $\phi_1, \phi_2, \dots, \phi_K$.
- In our case, $Z\left(\mathbf{S}_{t_m}, \alpha^{\beta}_{t_m}\right)$ depends on $\mathbf{S}_{t_m}$ only through $\phi_k(\mathbf{S}_{t_m})$:

$$Z\left(\mathbf{S}_{t_m}, \alpha^{\beta}_{t_m}\right) = \sum_{k=1}^{K} \alpha^{\beta}_{t_m}(k)\phi_k(\mathbf{S}_{t_m}).$$

- Computation of $\alpha^{\beta}_{t_m}$ (or $\widehat{\alpha}^{\beta}_{t_m}$) by least squares regression.
- The $\alpha^{\beta}_{t_m}$ determines the early-exercise policy.
- The continuation value:

$$\begin{aligned}\widehat{Q}_{t_{m-1}}(\mathbf{S}_{t_{m-1}}(n)) &= D_{t_{m-1}}\mathbb{E}\left[\left(\sum_{k=1}^{K} \widehat{\alpha}^{\beta}_{t_m}(k)\phi_k(\mathbf{S}_{t_m})\right) |\mathbf{S}_{t_{m-1}}\right] \\ &= D_{t_{m-1}}\sum_{k=1}^{K} \widehat{\alpha}^{\beta}_{t_m}(k)\mathbb{E}\left[\phi_k(\mathbf{S}_{t_m})|\mathbf{S}_{t_{m-1}}\right].\end{aligned}$$

Stochastic Grid Bundling Method

- Choosing ϕ_k: the expectations $\mathbb{E}\left[\phi_k(\mathbf{S}_{t_m})|\mathbf{S}_{t_{m-1}}\right]$ should be easy to calculate.
- The intrinsic value of the option, $h(\cdot)$, is usually an important and useful basis function.
- For $\mathbf{S}_t$ following a GBM, some expectations analytically available:
  - Geometric average payoff.
  - Arithmetic average payoff.
  - Min/Max payoff.
  - Spread payoff.
- For European options, the intermediate steps produce a variance reduction in the solution.
- For American options, Richardson interpolation stage is required.

Stochastic Grid Bundling Method

- SGBM has been developed as *duality-based method*.
- Provide two estimators (confidence interval).
- *Direct estimator* (high-biased estimation):

$$\widehat{V}_{t_{m-1}}(\mathbf{S}_{t_{m-1}}(n)) = \max\left(h\left(\mathbf{S}_{t_{m-1}}(n)\right), \widehat{Q}_{t_{m-1}}\left(\mathbf{S}_{t_{m-1}}(n)\right)\right),$$

$$\mathbb{E}[\widehat{V}_{t_0}(\mathbf{S}_{t_0})] = \frac{1}{N}\sum_{n=1}^{N}\widehat{V}_{t_0}(\mathbf{S}_{t_0}(n)).$$

- *Path estimator* (low-biased estimation):

$$\widehat{\tau}^*\left(\mathbf{S}(n)\right) = \min\{t_m \,:\, h\left(\mathbf{S}_{t_m}(n)\right) \geq \widehat{Q}_{t_m}\left(\mathbf{S}_{t_m}(n)\right),\ m = 1, \ldots, M\},$$

$$v(n) = h\left(\mathbf{S}_{\widehat{\tau}^*(\mathbf{S}(n))}\right),$$

$$\underline{V}_{t_0}(\mathbf{S}_{t_0}) = \lim_{N_{\mathrm{L}}}\frac{1}{N_{\mathrm{L}}}\sum_{n=1}^{N_{\mathrm{L}}} v(n).$$

Stochastic Grid Bundling Method

- The bundling stage allows local regression.
- Bundling techniques:
  - K-means clustering.
  - Recursive bifurcation.
  - Recursive bifurcation in a reduced space.
  - Equal partitioning (suitable for parallel versions).
- Important for the accuracy in the regression:
  - Quality of the paths within a bundle.
  - Amount of paths in a bundle.

Stochastic Grid Bundling Method

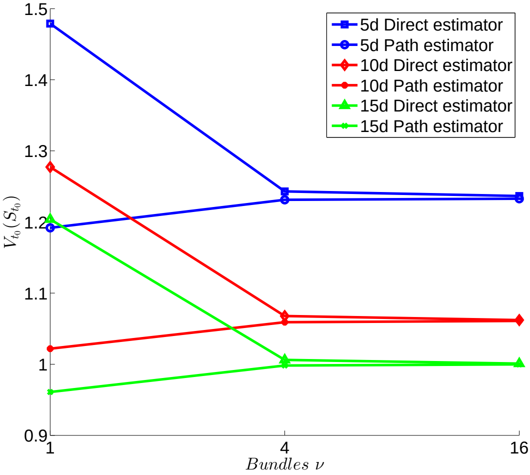


Figure: SGBM on arithmetic basket option - Convergence in bundles.

BENCHOP results - SGBM

- European options.

	10^{-1}	10^{-2}	10^{-3}
BSeuCallspreadU	0.12	0.40	18.53
BSeuCallspreadDelta	0.10	0.42	18.48
BSeuCallspreadVega	0.22	1.75	81.41
BSeuCallspreadVega (parfor)	0.17	0.63	26.57
BSeuPut3DbasketLCCU	0.08	0.08	0.81
BSeuPut10DbasketLCCU	0.10	0.09	0.95
BSeuPut3DbasketHVCU	0.08	0.10	0.79
BSeuPut10DbasketHVCU	0.09	0.10	1.81
BSeuPut4DbasketU	0.08	0.10	1.71
BSeuPut5DbasketU	0.09	0.09	0.15
BSeuPut5DbasketDelta	0.09	0.09	0.17
BSeuPut5DbasketVega	0.37	0.42	0.91

Table: SGBM times(s).

BENCHOP results - SGBM

- Influence of parfor.

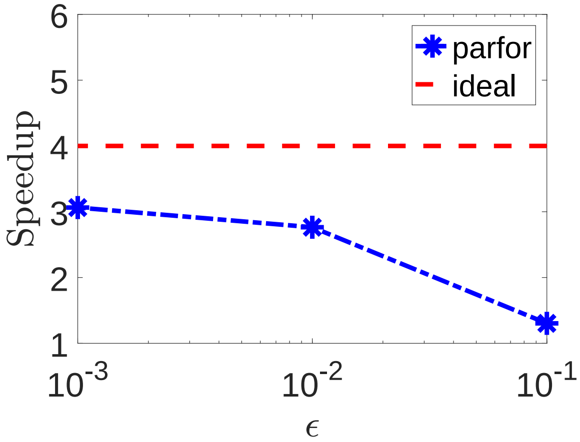


Figure: Speedup

BENCHOP results

- American options (no reference value).

	Value ($\epsilon = 10^{-3}$)	Serial	Parallel
BSamPutminU	4.3511	110.29	38.11
BSamPutminDelta	-0.2753	111.86	38.34
BSamPutminVega	7.889	221.15	74.94
BSamPut3DbasketLCCU	1.4695	3.33	1.35
BSamPut10DbasketLCCU	1.0828	4.27	1.94
BSamPut3DbasketHVCU	2.1878	3.10	1.30
BSamPut10DbasketHVCU	1.8873	8.86	4.30

Table: SGBM.

Efficient simulation of SABR model

- Simulation of the volatility process, $\sigma(t)|\sigma(s)$:

$$\sigma(t) \sim \sigma(s) \exp\left(\alpha \hat{W}_\sigma(t) - \frac{1}{2}\alpha^2(t-s)\right),$$

  where $\hat{W}_\sigma(t)$ is a independent Brownian motion.
- Simulation of the integrated variance process, $\int_s^t \sigma^2(z)\mathrm{d}z|\sigma(t), \sigma(s)$.
- Simulation of the forward process, $S(t)|S(s), \int_s^t \sigma^2(z)\mathrm{d}z, \sigma(t), \sigma(s)$.
- The conditional integrated variance is a challenging part. We propose:
  - Approximate the conditional distribution by using Fourier techniques and copulas.
  - Marginal distribution based on COS method.
  - Conditional distribution based on copulas.
  - Improvements in performance and efficiency (SCMC).

Distribution of the integrated variance

- For notational convenience, we will use $Y(s,t) := \int_s^t \sigma^2(z)\mathrm{d}z$.
- Discrete equivalent, M monitoring dates:

$$Y(s,t) := \int_s^t \sigma^2(z)\mathrm{d}z \approx \sum_{j=1}^{M} \Delta t \sigma^2(t_j) =: \hat{Y}(s,t)$$

  where $t_j = s + j\Delta t$, $j = 1, \dots, M$ and $\Delta t = \frac{t-s}{M}$.
- In the logarithmic domain, where we aim to find an approximation of $F_{\log \hat{Y} | \log \sigma(s)}$:

$$F_{\log \hat{Y} | \log \sigma(s)}(x) = \int_{-\infty}^{x} f_{\log \hat{Y} | \log \sigma(s)}(y)\mathrm{d}y,$$

  where $f_{\log \hat{Y} | \log \sigma(s)}$ is the *probability density function* (PDF) of $\log \hat{Y}(s,t) | \log \sigma(s)$.

PDF of the integrated variance

- Equivalent: Characteristic function and inversion (Fourier pair).
- Recursive procedure to derive an approximated $\phi_{\log \hat{Y} | \log \sigma(s)}$.
- By defining the logarithmic increment of $\sigma^2(t)$:

$$R_j = \log\left(\frac{\sigma^2(t_j)}{\sigma^2(t_{j-1})}\right), j = 1, \ldots, M.$$

- And introducing the iterative process

$$\begin{aligned} Y_1 &= R_M, \\ Y_j &= R_{M+1-j} + Z_{j-1}, \quad j = 2, \ldots, M. \end{aligned}$$

with $Z_j = \log(1 + \exp(Y_j))$.

- The quantity $\hat{Y}(s,t)$ can be expressed:

$$\hat{Y}(s,t) = \sum_{i=1}^{M} \sigma^2(t_i)\Delta t = \Delta t \sigma^2(s) \exp(Y_M).$$

CDF of the integrated variance

- And, we compute $\phi_{\log \hat{Y}|\log\sigma(s)}(u)$, as follows:

$$\phi_{\log \hat{Y}|\log\sigma(s)}(u) = \exp\left(iu\log(\Delta t\sigma^2(s))\right)\phi_{Y_M}(u).$$

- By applying COS method in the support $[\hat{a},\hat{b}]$:

$$f_{\log \hat{Y}|\log\sigma(s)}(x) \approx \frac{2}{\hat{b}-\hat{a}}\sum_{k=0}^{N-1'} C_k \cos\left((x-\hat{a})\frac{k\pi}{\hat{b}-\hat{a}}\right),$$

with

$$C_k = \Re\left(\phi_{\log \hat{Y}|\log\sigma(s)}\left(\frac{k\pi}{\hat{b}-\hat{a}}\right)\exp\left(-i\frac{\hat{a}k\pi}{\hat{b}-\hat{a}}\right)\right).$$

- The CDF of $\log \hat{Y}(s,t)|\log\sigma(s)$:

$$\begin{aligned} F_{\log \hat{Y}|\log\sigma(s)}(x) &= \int_{-\infty}^{x} f_{\log \hat{Y}|\log\sigma(s)}(y)\mathrm{d}y \\ &\approx \int_{\hat{a}}^{x} \frac{2}{\hat{b}-\hat{a}}\sum_{k=0}^{N-1'} C_k \cos\left((y-\hat{a})\frac{k\pi}{\hat{b}-\hat{a}}\right)\mathrm{d}y. \end{aligned}$$

Copula-based simulation of $\int_s^t \sigma^2(z)\mathrm{d}z | \sigma(t), \sigma(s)$

- In order to apply copulas, we need (logarithmic domain):
  - $F_{\log \hat{Y} | \log \sigma(s)}$.
  - $F_{\log \sigma(t) | \log \sigma(s)}$.
  - Correlation between $\log Y(s,t)$ and $\log \sigma(t)$.
- The distribution of $\log \sigma(t) | \log \sigma(s)$ is known ($\sigma(t)$ follows a log-normal distribution).
- Approximated Pearson's correlation coefficient:

$$\mathcal{P}_{\log Y, \log \sigma(t)} \approx \frac{t^2 - s^2}{2\sqrt{\left(\frac{1}{3}t^4 + \frac{2}{3}ts^3 - t^2 s^2\right)}}.$$

- For some copulas, like Archimedean, Kendall's τ is required:

$$\mathcal{P} = \sin\left(\frac{\pi}{2}\tau\right).$$

Sampling $\int_s^t \sigma^2(z)\mathrm{d}z|\sigma(t), \sigma(s)$

- It forms the basis of the mSABR method.
- Steps:
  1. Determine $F_{\log\sigma(t)|\log\sigma(s)}$ and $F_{\log\hat{Y}|\log\sigma(s)}$.
  2. Determine the correlation between $\log Y(s,t)$ and $\log\sigma(t)$.
  3. Generate correlated uniform samples, $U_{\log\sigma(t)|\log\sigma(s)}$ and $U_{\log\hat{Y}|\log\sigma(s)}$ by means of copula.
  4. From $U_{\log\sigma(t)|\log\sigma(s)}$ and $U_{\log\hat{Y}|\log\sigma(s)}$ invert original marginal distributions.
  5. The samples of $\sigma(t)|\sigma(s)$ and $Y(s,t) = \int_s^t \sigma^2(z)\mathrm{d}z|\sigma(t), \sigma(s)$ are obtained by taking exponentials.

Simulation of $S(t)|S(s), \int_s^t \sigma^2(z)\mathrm{d}z, \sigma(t), \sigma(s)$

- In the original paper, we use numerical inversion of the asset CDF.
- For the BENCHOP project, we consider an alternative scheme that allows the use of the mSABR basis.
- Discretization scheme Log-Euler+ (time step Δt):

$$\begin{aligned}\log S(t+\Delta t) = \log S(t) &- \frac{1}{2} S^{2(\beta-1)}(t) \int_t^{t+\Delta t} \sigma^2(z)\mathrm{d}z \\ &+ S^{\beta-1}(t)\frac{\rho}{\alpha}\left(\sigma(t+\Delta t) - \sigma(t)\right) \\ &+ S^{\beta-1}(t)\sqrt{1-\rho^2}\int_t^{t+\Delta t} \sigma(z) dW_S(z),\end{aligned}$$

where $\int_t^{t+\Delta t} \sigma(z) dW_S(z) \sim \mathcal{N}\left(0, \int_t^{t+\Delta t} \sigma^2(z)\mathrm{d}z\right)$.

BONUS - MLMC SABR

- Well establish method introduced by Mike Giles in 2008.
- Suppose a generic multi-dimensional SDE

$$\mathrm{d}S(t) = a(S,t)\mathrm{d}t + b(S,t)\mathrm{d}W(t).$$

- A simple Euler discretization

$$\hat{S}_{n+1} = \hat{S}_n + a(\hat{S}_n, t_n)h + b(\hat{S}_n, t_n)\Delta W_n.$$

- If you want to estimate $\mathbb{E}[f(S_T)]$, the simplest estimator

$$\hat{X} = \frac{1}{N}\sum_{i=1}^{N} f(S^i_{T/h})$$

BONUS - MLMC SABR

- Consider Monte Carlo path simulations with different time step size

$$h_l = \frac{T}{M^l}, \quad l = 1, 2, \dots, L.$$

- For a given Brownian path $W(t)$, let P denote the payoff $f(S(T))$.
- And let $\hat{P}_l$ denote the approximation to P using the numerical discretisation with time step h_l.
- Then, it is true that

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

- The simplest estimator of $\mathbb{E}[\hat{P}_l - \hat{P}_{l-1}]$ is

$$\hat{X}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} \left(\hat{P}_l^i - \hat{P}_{l-1}^i \right)$$

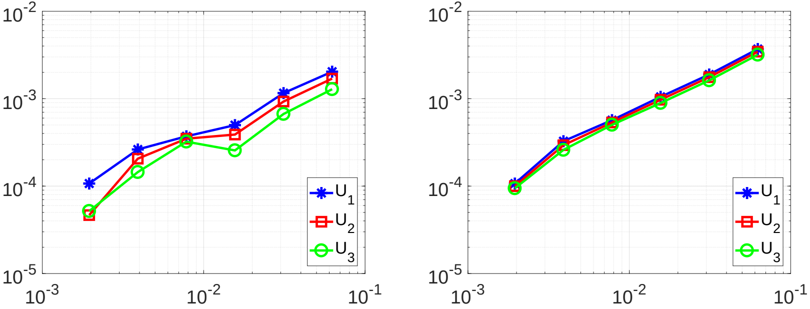
BONUS - MLMC SABR

- A key point here is that the quantity $\hat{P}_l^i - \hat{P}_{l-1}^i$ comes from two discrete approximations with different time steps but the same Brownian path.
- It can be seen as a very coarse estimation + different levels of corrections.
- It is shown in the paper[4] that this reduces the variance and the computational complexity of the final estimator.

[4] Michael B. Giles. "Multi-level Monte Carlo path simulation". In: *Operations Research* 56.3 (2008), pp. 607–617.

Convergence of the MLMC - SABR model

- As usual for MLMC methods, we test the convergence of the correction estimators.



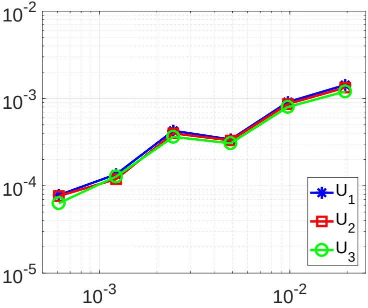
(a) Set I - Mean corrections.

(b) Set I - Var. corrections.

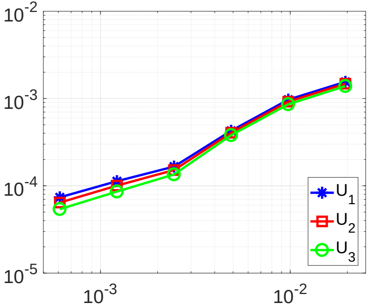
Figure: Convergence of the MLMC implementation for the SABR model.

Convergence of the MLMC - SABR model

- Similar results for Set II.



(a) Set II - Mean corrections.



(b) Set II - Var. corrections.

Figure: Convergence of the MLMC implementation for the SABR model.

BENCHOP results - SABR

- Time vs. accuracy.

	10^{-1}	10^{-2}	10^{-3}
SABReuCallI_mSABR	0.48	0.66	10.46
SABReuCallI_MLMC	0.01	0.07	25.64
SABReuCallII_mSABR	0.36	0.55	10.45
SABReuCallII_MLMC	0.01	0.07	25.55

Table: The mSABR and MLMC methods

Ongoing work

- Implementation of the remaining basket problems.
- Improve the bundling when two assets.
- Parallel version of SABR.
- MLMC + mSABR (if possible).
- Address the other stochastic local volatility models.

Suggestions, comments & questions

E-mail: aleitao85@gmail.com
Webpage: alvaroleitao.github.io

Thank you for your attention