The BENCHOP project

The BENCHmarking project in Option Pricing

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

Delft University of Technology - Centrum Wiskunde & Informatica

Reading group, September 8, 2017

The BENCHOP project

- The purpose and aim of BENCHOP is to provide sets of benchmark problems.
- 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 can be found in [vSHL+15].

Aspects 2nd edition

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

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

Basket options - Problems

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:

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

- 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

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

   where $S(t) = \bar{S}(t)\exp(r(T-t))$. 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

$$\begin{aligned} K_i(T) &= S(0)\exp(0.1 \times \sqrt{T} \times \delta_i), \\ \delta_i &= -1.5, -1.0, -0.5, 0.0, 0.5, 1.0, 1.5. \end{aligned}$$

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 $f(s) = \frac{1}{2}\alpha s^2 + \beta s + \gamma$.

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).
- For SABR model:
  - The mSABR simulation scheme [LGO17].
  - Multi Level Monte Carlo, MLMC, to exploit parallel features.

Stochastic Grid Bundling Method

- Early-exercise pricing method [JO15].
- 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

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

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

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

  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

- Basis functions $\phi_1, \phi_2, \ldots, \phi_K$.
- 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 example:
  - Geometric basket Bermudan:

$$h(\mathbf{S}_t) = \left(\prod_{\delta=1}^{d} S_t^{\delta}\right)^{\frac{1}{d}}$$

  - Arithmetic basket Bermudan:

$$h(\mathbf{S}_t) = \frac{1}{d}\sum_{\delta=1}^{d} S_{t_m}^{\delta}$$

- For $\mathbf{S}_t$ following a GBM: expectations analytically available.

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):

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

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

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

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).

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 to take advantage of the parallel features.
- But we desire to take advantage of mSABR.
- 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 $\int_t^{t+\Delta t} \sigma(z) dW_S(z) \sim \mathcal{N}\left(0, \int_t^{t+\Delta t} \sigma^2(z)\mathrm{d}z\right)$.

Numerical results

- Computational time vs. prescribed accuracy.
- Relative error (RE).
- For SGBM: only sequential times.
- For mSABR and MLMC: sequential times and parallel (parfor + GPU array) times.
- Computer system: Intel Core i7-4720HQ 2.6 GHz, RAM 16 Gb.

Basket options

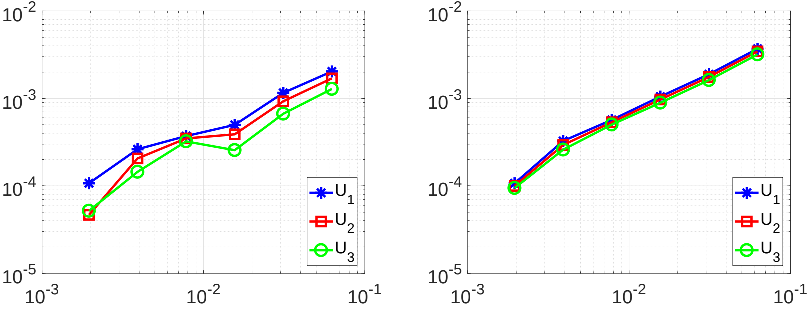
- Reference values only for Problem 5 and European options.
- Targeted precision: $< 10^{-3}$.

	Price u
3D European low corr.	28.4988
3D European high corr.	28.6025
10D European low corr.	68.7701
10D European high corr.	66.2690

Table: SGBM times(s).

Convergence of the MLMC - SABR model

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



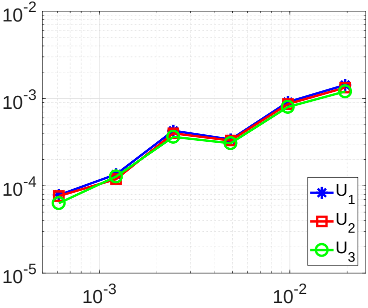
(a) Set I - Mean corrections.

(b) Set I - Var. corrections.

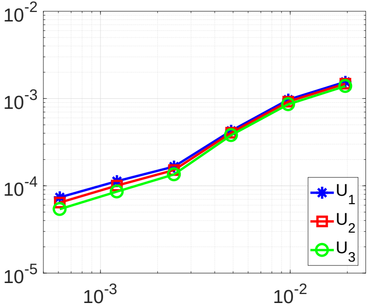
Figure: Convergence of the MLMC implementation for the SABR model.

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.

SABR model

- Computational time in seconds for the considered approaches.
- Targeted precision: $< 10^{-3}$.

	Serial		Parallel	
	mSABR	MLMC	mSABR	MLMC
Set I	11.833	1.737	9.805	1.296
Set II	10.378	27.216	9.628	16.847

Table: Time (s).

Ongoing work

- Implementation of the remaining basket problems.
- Parallel version of SGBM.
- Improved parallel version of mSABR.
- MLMC + mSABR (if possible).
- Other stochastic local volatility models?

References

Shashi Jain and Cornelis W. Oosterlee.

The Stochastic Grid Bundling Method: Efficient pricing of Bermudan options and their Greeks.

Applied Mathematics and Computation, 269:412–431, 2015.

Álvaro Leitao, Lech A. Grzelak, and Cornelis W. Oosterlee.

On an efficient multiple time step Monte Carlo simulation of the SABR model.

Quantitative Finance, 2017.

Lina von Sydow, Lars Josef Höök, Elisabeth Larsson, Erik Lindström, Slobodan Milovanović, Jonas Persson, Victor Shcherbakov, Yuri Shpolyanskiy, Samuel Sirén, Jari Toivanen, Johan Waldén, Magnus Wiktorsson, Jeremy Levesley, Juxi Li, Cornelis W. Oosterlee, Maria J. Ruijter, Alexander Toropov, and Yangzhang Zhao.

Benchop – the benchmarking project in option pricing.

International Journal of Computer Mathematics, 92(12):2361–2379, 2015.

Suggestions, comments & questions

Thank you for your attention