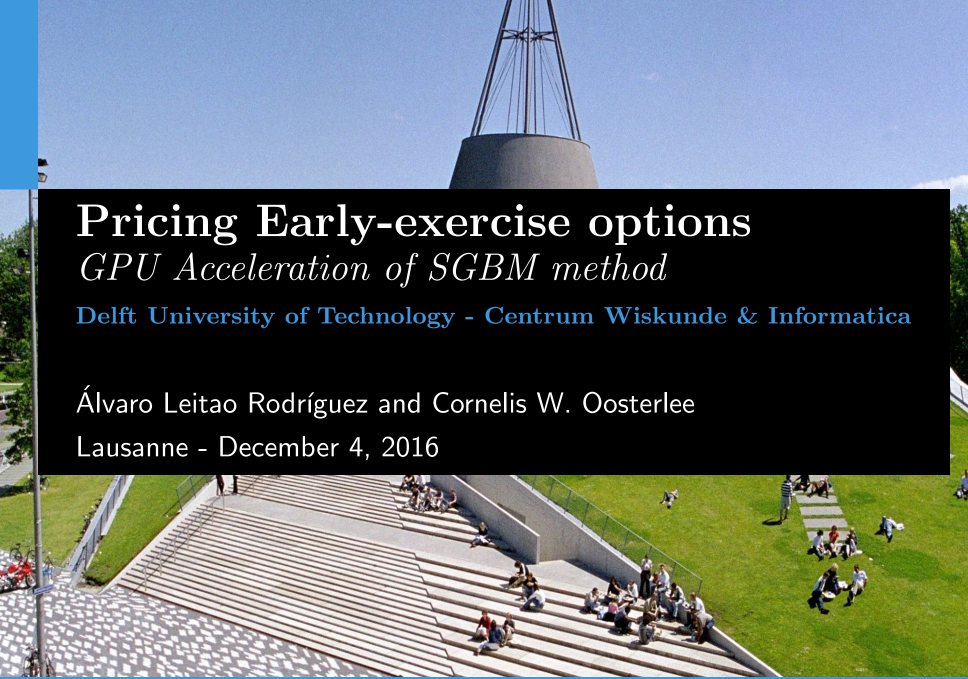


Pricing Early-exercise options

GPU Acceleration of SGBM method

Delft University of Technology - Centrum Wiskunde & Informatica

Álvaro Leitao Rodríguez and Cornelis W. Oosterlee

Lausanne - December 4, 2016

Outline

1. Definitions
2. Basket Bermudan Options
3. Stochastic Grid Bundling Method
4. Parallel GPU Implementation
5. Results
6. Conclusions

Definitions

Option

A contract that offers the buyer the right, but not the obligation, to buy (call) or sell (put) a financial asset at an agreed-upon price (the strike price) during a certain period of time or on a specific date (exercise date). Investopedia.

Option price

The fair value to enter in the option contract. In other words, the (discounted) expected value of the contract.

$$V_t = D_t \mathbb{E}\left[f(S_t)\right]$$

where f is the *payoff* function, S the underlying asset, t the exercise time and D_t the discount factor.

Definitions - cont.

Pricing techniques

- Stochastic process, S_t.
- Simulation: Monte Carlo method.
- PDEs: Feynman-Kac theorem.

Types of options - Exercise time

- European: End of the contract, $t = T$.
- American: Anytime, $t \in [0, T]$.
- Bermudan: Some predefined times, $t \in \{t1, \ldots, t_M\}$
- Many others: Asian, barrier, ...

Definitions - cont.

Early-exercise option price

- American:

$$V_t = \sup_{t \in [0,T]} D_t \mathbb{E}\left[f(S_t)\right].$$

- Bermudan:

$$V_t = \sup_{t \in \{t1,\dots,t_M\}} D_t \mathbb{E}\left[f(S_t)\right].$$

Pricing early-exercise options

- PDEs: Hamilton-Jacobi-Bellman equation.
- Simulation:
  - Least-squares method (LSM), Longstaff and Schwartz.
  - Stochastic Grid Bundling method (SGBM) [JO15].

TUDelft

Basket Bermudan Options

- Right to exercise at a set of times: $t \in \{t_0 = 0, \ldots, t_m, \ldots, t_M = T\}$.
- d-dimensional underlying process: $\mathbf{S}_t = (S_t^1, \ldots, S_t^d) \in \mathbb{R}^d$.
- Intrinsic value of the option: $h_t := h(\mathbf{S}_t)$.
- The value of the option at the terminal time T:

$$V_T(\mathbf{S}_T) = f(\mathbf{S}_T) = \max(h(\mathbf{S}_T), 0).$$

- The conditional continuation value Q_{t_m}, i.e. the discounted expected payoff at time t_m:

$$Q_{t_m}(\mathbf{S}_{t_m}) = D_{t_m}\mathbb{E}\left[V_{t_{m+1}}(\mathbf{S}_{t_{m+1}})|\mathbf{S}_{t_m}\right].$$

- The Bermudan option value at time t_m and state $\mathbf{S}_{t_m}$:

$$V_{t_m}(\mathbf{S}_{t_m}) = f(\mathbf{S}_T) = \max(h(\mathbf{S}_{t_m}), Q_{t_m}(\mathbf{S}_{t_m})).$$

- Value of the option at the initial state $\mathbf{S}_{t_0}$, i.e. $V_{t_0}(\mathbf{S}_{t_0})$.

Basket Bermudan options scheme

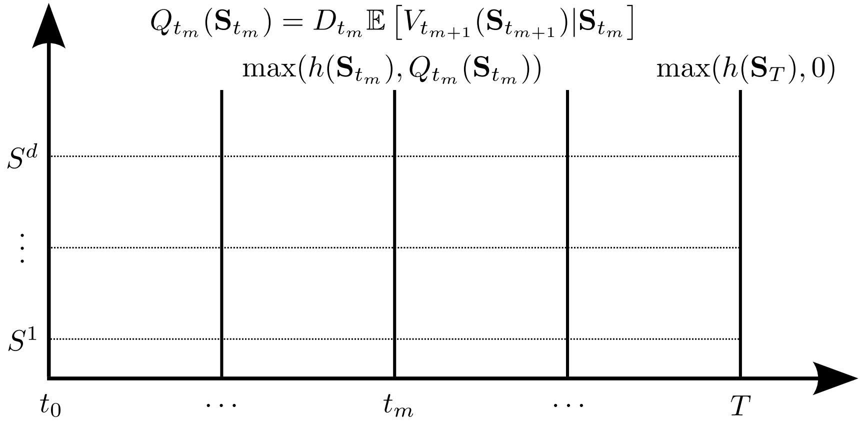


Figure: d-dimensional Bermudan option

Stochastic Grid Bundling Method

- 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 (II)

- 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}

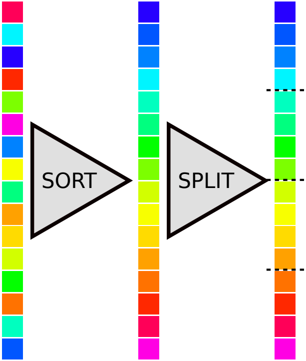
$$\widehat{Q}_{t_{m-1}}(\mathbf{S}_{t_{m-1}}(n)) = \mathbb{E}[Z(\mathbf{S}_{t_m}, \alpha^{\beta}_{t_m})|\mathbf{S}_{t_{m-1}}(n)].$$

  The option value is then given by:

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

Bundling

- Original: Iterative process (K-means clustering).
- Problems: Too expensive (time and memory) and distribution.
- New technique: Equal-partitioning. Efficient for parallelization.
- Two stages: sorting and splitting.



Parametrizing the option value

- 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:

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

TUDelft

Basis functions

- 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.

Estimating the option value

- 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):

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

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

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

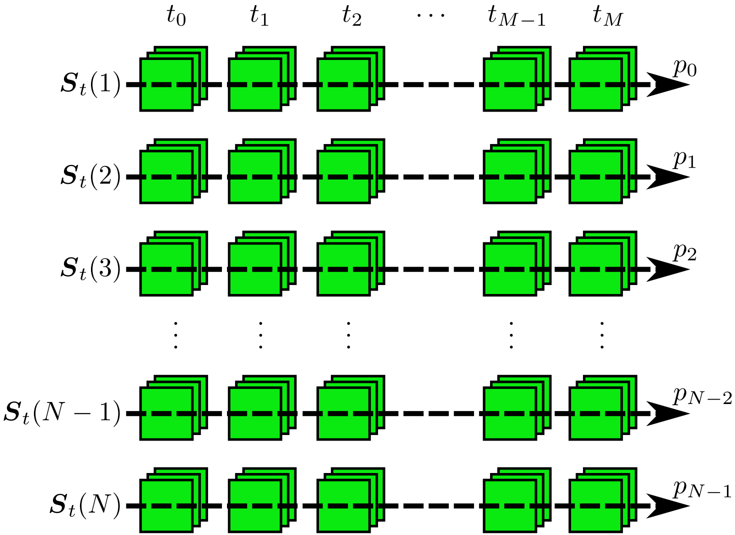
Parallel SGBM on GPU

- NVIDIA CUDA platform.
- Parallel strategy: two parallelization stages:
    - Forward: Monte Carlo simulation.
    - Backward: Bundles $\rightarrow$ Oportunity of parallelization.
- Novelty in early-exercise option pricing methods.
- Two implementations $\rightarrow$ K-means vs. Equal-partitioning:
    - K-means: sequential parts.
    - K-means: transfers between CPU and GPU cannot be avoided.
    - K-means: all data need to be stored.
    - K-means: Load-balancing.
    - Equal-partitioning: fully parallelizable.
    - Equal-partitioning: No transfers.
    - Equal-partitioning: efficient memory use.

Parallel SGBM on GPU - Forward in time

- One GPU thread per Monte Carlo simulation.
- Random numbers “on the fly”: cuRAND library.
- Compute intermediate results:
  - Expectations.
  - Intrinsic value of the option.
  - Equal-partitioning: sorting criterion calculations.
- Intermediate results in the registers: fast memory access.
- Original bundling: all the data still necessary.

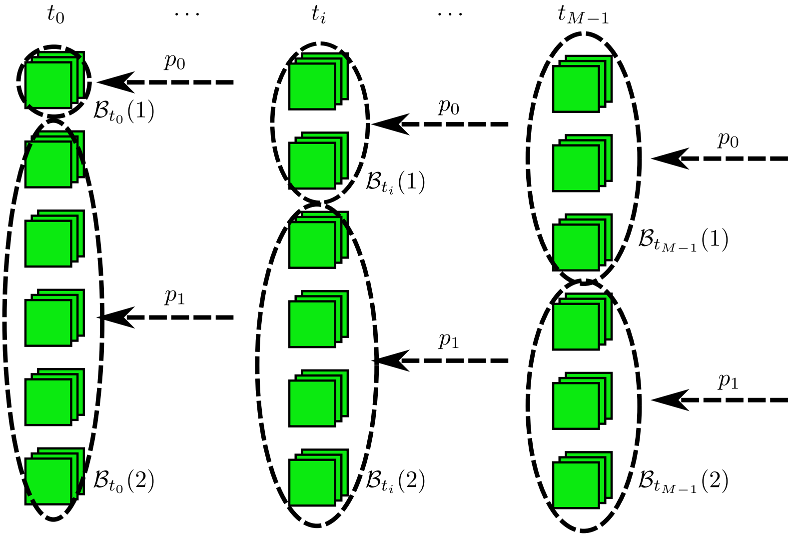
Parallel SGBM on GPU - Forward in time



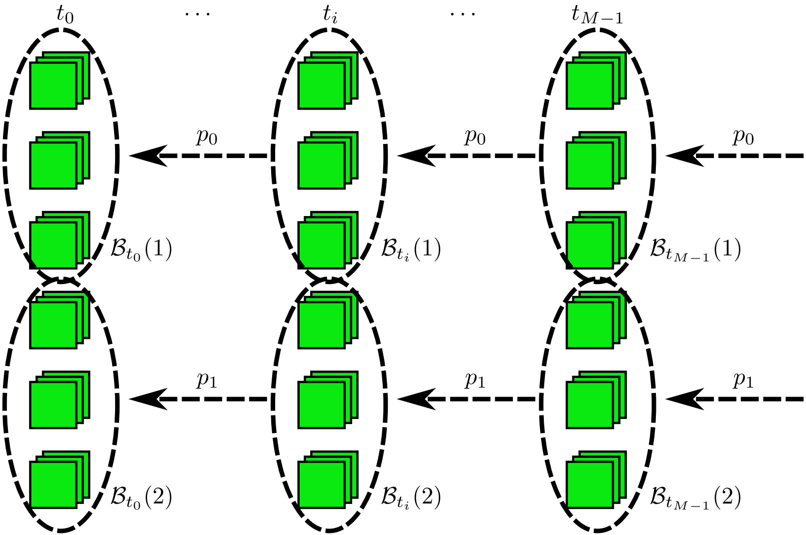
Parallel SGBM on GPU - Backward in time

- One parallelization stage per exercise time step.
- Sort w.r.t bundles: efficient memory access.
- Parallelization in bundles.
- Each bundle calculations (option value and early-exercise policy) in parallel.
- All GPU threads collaborate in order to compute the continuation value.

Parallel SGBM on GPU - Backward in time



Parallel SGBM on GPU - Backward in time

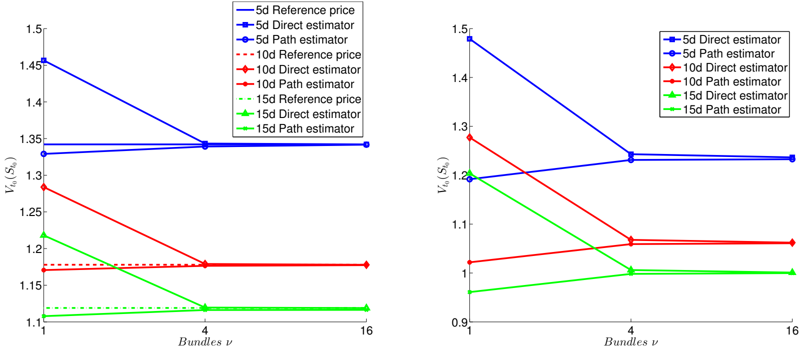


TUDelft

Results

- Accelerator Island system of Cartesius Supercomputer.
  - Intel Xeon E5-2450 v2.
  - NVIDIA Tesla K40m.
  - C-compiler: GCC 4.4.7.
  - CUDA version: 5.5.
- Geometric and arithmetic basket Bermudan put options: $\mathbf{S}_{t_0} = (40, \ldots, 40) \in \mathbb{R}^d$, $X = 40$, $r_t = 0.06$, $\sigma = (0.2, \ldots, 0.2) \in \mathbb{R}^d$, $\rho_{ij} = 0.25$, $T = 1$ and $M = 10$.
- Basis functions: $K = 3$.
- Multi-dimensional Geometric Brownian Motion.
- Euler discretization, $\delta t = T/M$.

Equal-partitioning: convergence test



(a) Geometric basket put option (b) Arithmetic basket put option

Figure: Convergence with equal-partitioning bundling technique. Test configuration: $N = 2^{18}$ and $\Delta t = T/M$.

Speedup

	Geometric basket Bermudan option					
	k-means			equal-partitioning		
	$d = 5$	$d = 10$	$d = 15$	$d = 5$	$d = 10$	$d = 15$
C	604.13	1155.63	1718.36	303.26	501.99	716.57
CUDA	35.26	112.70	259.03	8.29	9.28	10.14
Speedup	17.13	10.25	6.63	36.58	54.09	70.67
	Arithmetic basket Bermudan option					
	k-means			equal-partitioning		
	$d = 5$	$d = 10$	$d = 15$	$d = 5$	$d = 10$	$d = 15$
C	591.91	1332.68	2236.93	256.05	600.09	1143.06
CUDA	34.62	126.69	263.62	8.02	11.23	15.73
Speedup	17.10	10.52	8.48	31.93	53.44	72.67

Table: SGBM total time (s) for the C and CUDA versions. Test configuration: $N = 2^{22}$, $\Delta t = T/M$ and $\nu = 2^{10}$.

Speedup - High dimensions

	Geometric basket Bermudan option					
	$\nu = 2^{10}$			$\nu = 2^{14}$		
	$d = 30$	$d = 40$	$d = 50$	$d = 30$	$d = 40$	$d = 50$
C	337.61	476.16	620.11	337.06	475.12	618.98
CUDA	4.65	6.18	8.08	4.71	6.26	8.16
Speedup	72.60	77.05	76.75	71.56	75.90	75.85
	Arithmetic basket Bermudan option					
	$\nu = 2^{10}$			$\nu = 2^{14}$		
	$d = 30$	$d = 40$	$d = 50$	$d = 30$	$d = 40$	$d = 50$
C	993.96	1723.79	2631.95	992.29	1724.60	2631.43
CUDA	11.14	17.88	26.99	11.20	17.94	27.07
Speedup	89.22	96.41	97.51	88.60	96.13	97.21

Table: SGBM total time (s) for a high-dimensional problem with equal-partitioning. Test configuration: $N = 2^{20}$ and $\Delta t = T/M$.

TU Delft

Conclusions

- Efficient parallel GPU implementation.
- Extend the SGBM's applicability: Increasing dimensionality.
- New bundling technique.
- Future work:
  - Explore the new CUDA 7 features: cuSOLVER (QR factorization).
  - CVA calculations.

TUDelft

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 and Cornelis W. Oosterlee.
GPU Acceleration of the Stochastic Grid Bundling Method for Early-Exercise options.
International Journal of Computer Mathematics, 92(12):2433–2454, 2015.

Acknowledgements

This research was supported by the European Union in the FP7-PEOPLE-2012-ITN Program under Grant Agreement Number 304617 (FP7 Marie Curie Action, Project Multi-ITN STRIKE – Novel Methods in Computational Finance).

Thank you for your attention

Appendix

- Geo. basket Bermudan option - Basis functions:

$$\phi_k(\mathbf{S}_{t_m}) = \left(\left(\prod_{\delta=1}^{d} S_{t_m}^{\delta}\right)^{\frac{1}{d}} \right)^{k-1}, \; k = 1, \dots, K,$$

- The expectation can directly be computed as:

$$\mathbb{E}\left[\phi_k(\mathbf{S}_{t_m}) | \mathbf{S}_{t_{m-1}}(n)\right] = \left(P_{t_{m-1}}(n) e^{\left(\bar{\mu} + \frac{(k-1)\bar{\sigma}^2}{2}\right)\Delta t} \right)^{k-1},$$

where,

$$P_{t_{m-1}}(n) = \left(\prod_{\delta=1}^{d} S_{t_{m-1}}^{\delta}(n)\right)^{\frac{1}{d}}, \; \bar{\mu} = \frac{1}{d}\sum_{\delta=1}^{d}\left(r - q_\delta - \frac{\sigma_\delta^2}{2}\right), \; \bar{\sigma}^2 = \frac{1}{d^2}\sum_{p=1}^{d}\left(\sum_{q=1}^{d} C_{pq}^2\right)^2$$

TUDelft

Appendix

- Arith. basket Bermudan option - Basis functions:

$$\phi_k(\mathbf{S}_{t_m}) = \left(\frac{1}{d}\sum_{\delta=1}^{d} S_{t_m}^{\delta}\right)^{k-1}, k = 1, \ldots, K.,$$

- The summation can be expressed as a linear combination of the products:

$$\left(\sum_{\delta=1}^{d} S_{t_m}^{\delta}\right)^{k} = \sum_{k_1+k_2+\cdots+k_d=k} \binom{k}{k_1, k_2, \ldots, k_d} \prod_{1\leq\delta\leq d} \left(S_{t_m}^{\delta}\right)^{k_\delta},$$

- And the expression for Geometric basket option can be applied.