C. R. Acad. Sci. Paris, Ser. I 354 (2016) 1132–1138

Contents lists available at ScienceDirect

C. R. Acad. Sci. Paris, Ser. I

www.sciencedirect.com

Numerical analysis

The parareal algorithm for American options

La méthode pararéelle pour les options américaines

Gilles Pagès [a], Olivier Pironneau [b], Guillaume Sall [a,b]

[a] *Laboratoire de probabilités et modèles aléatoires, UMR 7599, case 188, 4, place Jussieu, 75252 Paris cedex 05, France*
[b] *Laboratoire Jacques-Louis-Lions, UMR 7598, case 187, 4, place Jussieu, 75252 Paris cedex 05, France*

ARTICLE INFO

Article history:
Received 23 May 2016
Accepted after revision 27 September 2016
Available online 5 October 2016

Presented by Alain Bensoussan

ABSTRACT

This note provides a description of the parareal method for American contracts, a numerical section to assess its performance. The scalar case is investigated. Least-Square Monte Carlo (LSMC) and parareal time decomposition with two or more levels are used, leading to an efficient parallel implementation. It contains also a convergence argument for the two-level parareal Monte Carlo method when the time step used for the Euler scheme at each level is appropriate. This argument provides also a tool for analyzing the multilevel case.

RÉSUMÉ

Dans cette note, la méthode pararéelle est introduite pour le calcul d'options américaines. L'algorithme LSMC (*Least-Square Monte Carlo*) de Longstaff–Schartz est parallélisé grâce à une décomposition en temps multi-niveaux. Dans une section numérique, les performances de la méthode sont données dans deux cas scalaires. Un résultat partiel de convergence est énoncé lorsque la méthode d'Euler explicite est utilisée avec des pas de temps appropriés sur chaque niveau. Une estimation est obtenue, qui permet d'analyser la méthode pararéelle multi-niveaux.

1. Introduction

In quantitative finance, risk assessment is computer intensive and expensive, and there is a market for cheaper and faster methods, as seen from the large literature on parallelism and GPU implementation of numerical methods for option pricing [1,6,7,11,12,14–16,21].

American contracts are not easy to compute on a parallel computer; even if a large number of them have to be computed at once, an embarrassingly parallel problem, still the cost of the transfer of data makes parallelism at the level of one contract attractive. But the task is not easy, especially when the number of underlying assets is large [3,5,22], ruling out the PDE approach [2]. Furthermore the most popular sequential algorithm is the Least-Square Monte Carlo (LSMC) method of

E-mail addresses: gilles.pages@upmc.fr (G. Pagès), olivier.pironneau@upmc.fr (O. Pironneau), guillaume.sall@upmc.fr (G. Sall).

http://dx.doi.org/10.1016/j.crma.2016.09.010

Longstaff and Schwartz [19]. Exploiting parallelism by allocating blocks of Monte Carlo paths to different processors is not convincingly efficient [7], because the backward regression is essentially sequential and needs all Monte Carlo paths in the same processor.

In this note, we investigate the parareal method, introduced in [18], for the task. An earlier study by Bal and Maday [4] has paved the way but it is restricted to Stochastic Differential Equations (SDE) without LSMC. Yet it contains a convergence proof for the two-level method in the restricted case where the solution is computed exactly at the lowest level [4].

This note provides a description of the parareal method for American contracts, a numerical section to assess its performance. The scalar case is investigated. Least-Square Monte Carlo (LSMC) and parareal time decomposition with two or more levels are used, leading to an efficient parallel implementation. It contains also a convergence argument for the two-level parareal Monte Carlo method when the time step used for the Euler scheme at each level is appropriate.

Convergence of LSMC for American contracts has been proved by Clément, Lamberton and Protter [9]; it is not unreasonable to expect an extension of their estimates for the parareal method, but this note does not contain such a result, only a numerical assessment.

2. The problem

With the usual notations [17], consider a probability space $(\Omega, \mathcal{A}, \mathbb{P})$, and functions $b, \sigma, f : [0, T] \times \mathbb{R} \mapsto \mathbb{R}$, uniformly Lipschitz continuous in x, t.

Let $W = (W_t)_{t\in[0,T]}$ be a standard Brownian motion on $(\Omega, \mathcal{A}, \mathbb{P})$. Let $X = (X_t)_{t\in[0,T]},\ X_t \in \mathbb{R}$, be a diffusion process, strong solution to the SDE

$$\mathrm{d}X_t = b(t, X_t)\,\mathrm{d}t + \sigma(t, X_t)\,\mathrm{d}W_t, \quad X(0) = X_0 \in \mathbb{R}. \tag{1}$$

A (vanilla) European contract on X is defined by its maturity T and its payoff $\mathbb{E}[f(T, X_T)]$, typically $f(t,x) = \mathrm{e}^{r(T-t)}(\kappa - x)^+$ in the case of a put of strike price κ and interest rate r. An American style contract allows the owner to claim the payoff $f(t, X_t)$ at any time $\in [0, T]$. So a rational strategy to maximize the average profit $\mathcal{V}$ at time t is to find the $[t, T]$-valued $\mathcal{F}$-stopping time solution to the *Snell envelope* problem:

$$\mathcal{V}(t, X_t) := \mathbb{E}[\mathrm{e}^{-r(\tau_t - t)} f(\tau_t, X_{\tau_t})|\mathcal{F}_t] = \mathbb{P}\text{-ess sup}_{\tau\in\mathcal{T}_t^{\mathcal{F}}}\mathbb{E}[\mathrm{e}^{-r(\tau-t)} f(\tau, X_\tau)|\mathcal{F}_t]$$

where $\mathcal{F} = (\mathcal{F}_t)_{t\in(0,T)}$ is the (augmented) filtration of W and $\mathcal{T}_t^{\mathcal{F}}$ denotes the set of $[t, T]$-valued $\mathcal{F}$-stopping times. Such an optimal stopping time exists (see [8,20]). We do not specify b, σ or f to stay in a general Optimal Stopping framework. In practice, American style options are replaced by the so-called *Bermuda* options where the exercise instants are restricted to a time grid $t_k = kh$, $k = 0, \dots, K$, where $h = \frac{T}{K}$ ($K \in \mathbb{N}^*$). Owing to the Markov property of $\{X_{t_k}\}_{k=0}^K$, the corresponding Snell envelope reads $(V(t_k, X_{t_k}))_{k=0,\dots,K}$ and satisfies a Backward Dynamic Programming recursion on k:

$$V(T, X_T) = f(T, X_T), \quad V(t_k, X_{t_k}) = \max\{f(t_k, X_{t_k}), \mathrm{e}^{-rh}\mathbb{E}[V(t_{k+1}, X_{t_{k+1}})|X_{t_k}]\}, \; k = K-1, \dots, 0. \tag{2}$$

The optimal stopping times τ_k (from time t_k) are given by a similar backward recursion:

$$\tau_K = T, \;\; \tau_k = t_k \text{ if } f(t_k, X_{t_k}) > \mathrm{e}^{-rh}\mathbb{E}[V(t_{k+1}, X_{t_{k+1}})|X_{t_k}], \;\; \tau_k = \tau_{k+1} \text{ otherwise}, \quad k = K-1, \dots, 0. \tag{3}$$

When $(X_{t_k})_{k=0,\dots,K}$ cannot be simulated at a reasonable computational cost, it can be approximated by the Euler scheme with step h, denoted $(\bar{X}^h_{t_k})_{k=0,\dots,K}$, which is a simulable Markov chain recursively defined by

$$\bar{X}^h_{t_{k+1}} = \bar{X}^h_{t_k} + b(t_k, \bar{X}^h_{t_k})h + \sigma(t_k, \bar{X}^h_{t_k})\Delta W_k, \;\; \bar{X}^h_0 = X_0, \; k = 0, \dots, K-1, \tag{4}$$

where $\Delta W_k := W_{t_{k+1}} - W_{t_k} = \sqrt{h}\,Z_k$ so that $\{Z_k\}_{k=0}^{K-1}$ are i.i.d. $\mathcal{N}(0,1)$-distributed random variables. From now on we switch to the Euler scheme, its Snell envelope, etc.

In LSMC, for each k, the conditional expectation $\mathbb{E}[V(t_{k+1}, \bar{X}^h_{t_{k+1}})|\bar{X}^h_{t_k}]$ as a function of $x = \bar{X}^h_{t_k}$, is approximated by its L^2-projection on the linear space spanned by the monomials $\{x^p\}_{p=0}^P$ from the values $\{\mathrm{e}^{-rh}V(t_{k+1}, \bar{X}^{h,(m)}_{t_{k+1}})\}_{m=1}^M$ generated by M Monte Carlo paths using (4); then each path has its own optimal stopping time at each $k \in \{0, \dots, K-1\}$ given by (for the stopping problem starting at k)

$$\tau_K^{(m)} = T, \;\; \tau_k^{(m)} = t_k \text{ if } f(t_k, \bar{X}^{h,(m)}_{t_k}) > \mathrm{e}^{-rh}\sum_{p=0}^{P}\bar{a}_k^p\,(\bar{X}^{h,(m)}_{t_k})^p, \;\; \tau_k^{(m)} = \tau_{k+1}^{(m)} \text{ otherwise}$$

where

$$\{\bar{a}_k^o, \dots, \bar{a}_k^P\} = \arg\min_{\{(a_k^0,\dots,a_k^P)\in\mathbb{R}^{P+1}\}}\sum_{m=1}^{M}\Big(V\big(t_{k+1}, \bar{X}^{h,(m)}_{t_{k+1}}\big) - \sum_{p=0}^{P} a_k^p\big(\bar{X}^{h,(m)}_{t_k}\big)^p\Big)^2.$$

Finally the price of the American contract is computed by

$$V(0, X_0) = \max\{f(0, X_0), \frac{1}{M}\sum_{m=1}^{M} \mathrm{e}^{-r\tau_1^{(m)}} f(\tau_1^{(m)}, \bar{X}^{h,(m)}_{\tau_1^{(m)}})\}.$$

Note that $\sum_0^P \bar{a}_k^p\ (\bar{X}^h_{t_k})^p$ is the best approximation of $\mathbb{E}[V(t_{k+1}, \bar{X}^h_{t_{k+1}})|\bar{X}^h_{t_k}]$ in the least-square sense (e.g., Longstaff and Schwartz's paper [19]) in the vector subspace $\langle(\bar{X}^h_{t_k})^p,\ p = 0 : P\rangle$ of $L^2(\mathbb{P})$.

3. A two-level parareal algorithm

3.1. The parareal method

Consider an ODE

$$\dot{x} = f(x, t),\ x(0) = x_0,\ t \in [t_0, t_K] = \cup_0^{K-1}[t_k, t_{k+1}].$$

Assume that $G_\delta(x_k, t_k)$ is a high-precision integrator that computes x at t_{k+1} from x_k at t_k. Assume $G_\triangle$ is a similar integrator but of low precision. The parareal algorithm is an iterative process with $n = 0, \ldots, N-1$ above a forward loop in time, $k = 0, \ldots, K-1$

$$x^{n+1}_{k+1} = G_\triangle(x^{n+1}_k, t_k) + G_\delta(x^n_k, t_k) - G_\triangle(x^n_k, t_k). \tag{5}$$

So the coarse-grid solution is corrected by the difference between the fine-grid prediction computed from the old value on that interval and the coarse-grid old solution. In this analysis, G_δ and $G_\triangle$ are Euler explicit schemes with time steps $\delta t < \Delta t$ respectively.

The same method can be applied to an SDE in the context of the Monte Carlo method provided the random variables $\{Z^m_{k,j}\}^{m=1,\ldots,M}_{j:=1,\ldots,J-1,k:=1,\ldots,K}$ defining ΔW_k in (4) are sampled once and for all in the initial phase of the algorithm and reused for all n (see the initialization step in Algorithm 3.2 for the notations).

The iterative process (5) is applied on each sample path with $G_\triangle$ a single step of (4) with $h = \Delta t$ and G_δ the result of J steps of (4) with $h = \delta t$. An error analysis is available in [4] for the stochastic case in the limit case $\delta t = 0$, i.e. when the fine integrator is the exact solution. For further results of the parareal method applied to deterministic ODEs and PDEs, see [18] and [13]. In this note, we also extend the result of [4] to the case $0 < \delta t < \Delta t$.

3.2. Algorithm

We denote by V_k a realization of $V(t_k, X_{t_k})$, $k = 0, \ldots, K = \frac{T}{\Delta t}$; consider a refinement of each interval $(t_k, t_{t_{k+1}})$ by a uniform sub-partition of time step $\delta t = \frac{\Delta t}{J}$, for some integer $J > 1$. Then

$$[t_k, t_{k+1}] = \cup_{j=0}^{J-1}[t_{k,j}, t_{k,j+1}] \text{ with } t_{k,j+1} = t_{k,j} + \delta t,\ j = 0, \ldots, J-1, \text{ so that } t_k = t_{k,0} = t_{k-1,J}.$$

Denote by $\mathfrak{P}f$ the L^2-projection of f on the monomials $1, x, \ldots, x^P$.

Let $n = 0, \ldots, N-1$ be the iteration index of the parareal algorithm.

```
Initialization Generate $\{Z^m_{k,j}\}^{m=1,\ldots,M}_{k=1,\ldots,K,j=1,\ldots,J}$ for the $M$ paths of the Monte Carlo method with the coarse and fine mesh.
  Compute recursively forward all Monte Carlo paths $\{\hat{X}^0_{t_k}(\omega^m)\}^M_{m=1}$ from $\hat{X}^0_0 = X_0$ by using (4) with $h = \Delta t$ and then
    recursively backward $\hat{V}^0_k = \max\{f(t_k, \hat{X}^0_{t_k}), \mathrm{e}^{-r\Delta t}\mathfrak{P}\,\mathbb{E}[\hat{V}^0_{k+1}|\hat{X}^0_{t_k}]\}$, $k = K-1, \ldots, 0$ from $\hat{V}^0_K(\omega^m) = \mathrm{e}^{-rT} f(T, \hat{X}^0_T(\omega^m))$,
    $m = 1 \ldots, M$.

for $n = 0, \ldots, N-1$
 for all $M$ paths,
  for $k = 0, \ldots, K-1$ (forward loop):
   (i) compute the fine-grid solution $\{\tilde{X}^{\delta,n}_{t_{k,j}}\}^J_{j=0}$ of (4) with refined step $h = \delta t = \frac{\Delta t}{J}$, started at $t_{k,0} = t_k$ from $\hat{X}^n_{t_k}$.
   (ii) compute the coarse-grid solution at $t_{k+1}$: $\bar{X}^\triangle_{t_{k+1}} = \hat{X}^{n+1}_{t_k} + b(t_k, \hat{X}^{n+1}_{t_k})\Delta t + \sigma(t_k, \hat{X}^{n+1}_{t_k})\Delta W_k$;
   (iii) set $\hat{X}^{n+1}_{t_{k+1}} = \bar{X}^\triangle_{t_{k+1}} + \tilde{X}^{\delta,n}_{t_{k,J}} - \hat{X}^n_{t_{k+1}}$.
  end k-loop
 end M-loop.
initialization: Compute $\bar{V}^{n+1}_K = \hat{V}^{n+1}_K = f(T, \hat{X}^{n+1}_T)$
for $k = K-1, \ldots, 0$ (backward loop):
  (i) on each $(t_k, t_{k+1})$, from $\tilde{V}^{\delta,n}_{k,J} = \mathfrak{P}\,\mathbb{E}(\hat{V}^n_{k+1} \mid \tilde{X}^{\delta,n}_{k,J})$, compute by a backward loop in $j$

        $\tilde{V}^{\delta,n}_{k,j} = \max\left\{f(t_{k,j}, \tilde{X}^{\delta,n}_{t_{k,j}}), \mathrm{e}^{-r\delta t}\mathfrak{P}\,\mathbb{E}[\tilde{V}^{\delta,n}_{k,j+1}|\tilde{X}^{\delta,n}_{t_{k,j}}]\right\},\ j = J-1, \ldots, 0;$
        (ii) compute $\bar{V}_k^{n+1} = \max\left\{f(t_k, \hat{X}_{t_k}^{n+1}), \mathrm{e}^{-r\Delta t}\mathfrak{P}\mathbb{E}[\bar{V}_{k+1}|\hat{X}_{t_k}^{n+1}]\right\}$;
        (iii) set $\hat{V}_k^{n+1} = \bar{V}_k^{n+1} + \tilde{V}_{k,0}^{\delta,n} - \hat{V}_k^n$.
    end backward k-loop
    end n-loop
```

Remark 1. Note that all fine-grid computations are local and can be allocated to a separate processor for each k, for parallelization.

The following partial results can be established for Algorithm 3.2bis obtained from 3.2 by changing the first step into $\tilde{V}_{k,J}^{\delta,n} = \mathfrak{P}\mathbb{E}(\bar{V}_{k+1}^n \,|\, \tilde{X}_{k,J}^{\delta,n})$ and the last step into: $\hat{V}_k^{n+1} = \bar{V}_k^{n+1} + \tilde{V}_{k,0}^{\delta,n} - \bar{V}_k^n$.

Theorem 3.1. *Assume $b, \sigma : [0,T] \times \mathbb{R}$ continuous, C^2 in x with spatial derivatives uniformly Lipschitz in $t \in [0,T]$. Then there exist C, independent of k, Δt and n, such that for $k = 0, \dots, K$, $n = 0, \dots, N$:*

$$\|\hat{X}_{t_k}^n - \bar{X}_{t_k}^\delta\|_{L^2(\mathbb{P})} \leq (C\Delta t)^n \sqrt{\binom{k}{n}} \|\bar{X}_{t_k}^\Delta - \bar{X}_{t_k}^\delta\|_{L^2(\mathbb{P})} \leq (C\Delta t)^n \sqrt{\binom{k}{n}} \sqrt{\Delta t}. \tag{6}$$

Furthermore, $\hat{X}_{t_k}^n = \bar{X}_{t_k}^\delta$ for all $n \geq k$ (e.g., definition (iii)).

Corollary 3.2. *For a fixed δt and n parareal iterations, the final and uniform errors satisfy*

$$\|\hat{X}_T^n - \bar{X}_T^\delta\|_{L^2(\mathbb{P})} \leq (C\Delta t)^{\frac{n}{2}} \sqrt{\frac{\Delta t}{n!}} \quad \text{and} \quad \left\| \max_{k=0,\dots,K} |\hat{X}_{t_k}^n - \bar{X}_{t_k}^\delta| \right\|_{L^2(\mathbb{P})} \leq \frac{(C\Delta t)^{\frac{n}{2}}}{\sqrt{(n+1)!}} \tag{7}$$

respectively where C only depends on the Lipschitz constants and norms of $b, b', b'', \sigma, \sigma', \sigma''$ and on T.

This estimate shows that when $\Delta t << C$, the method converges exponentially in n and geometrically in Δt.

Remark 2. The estimate (6) indicates that a recursive use of parareal with each sub-interval redivided into $J = O(\Delta t^{-1})$ smaller intervals, the so-called multilevels parareal, is better than many iterations at the second level only. Indeed, as the error decreases proportionally to $(\Delta t)^{\frac{n}{2}}$ at each level and as Δt becomes Δt^2 at the next grid level, the error after L levels is decreased by $(\Delta t)^{\frac{nL}{2}}$.

Proposition 3.3. (*a*) *Let*

$$\tilde{V}_{t_k}^{\Delta,n} = \mathbb{P}\text{-ess sup}_{\tau \in \mathcal{T}_{t_k}^{\mathcal{F}}} \mathbb{E}[\mathrm{e}^{-r(\tau - t_k)} f(\tau, \hat{X}_\tau^n) | \mathcal{F}_{t_k}], \quad \bar{V}_{t_k}^{\Delta,\delta} = \mathbb{P}\text{-ess sup}_{\tau \in \mathcal{T}_{t_k}^{\mathcal{F}}} \mathbb{E}[\mathrm{e}^{-r(\tau - t_k)} f(\tau, \bar{X}_\tau^\delta) | \mathcal{F}_{t_k}]$$

where $\mathcal{T}_{t_k}^{\mathcal{F}}$ denotes the set of $\{t_k, t_{k+1}, \dots, t_K\}$-valued $\mathcal{F}$-stopping times. Then, for some constant C,

$$\left\| \max_{k=0,\dots,K} \left|\tilde{V}_{t_k}^{\Delta,n} - \bar{V}_{t_k}^{\Delta,\delta}\right| \right\|_{L^2(\mathbb{P})} \leq [f]_{\text{Lip}} \frac{(C\Delta t)^{\frac{n}{2}}}{\sqrt{(n+1)!}}.$$

(Note that $(\bar{V}_{t_k}^{\Delta,\delta})_{k=0,\dots,K}$ is but the coarse Snell envelope of the refined Euler scheme.) At a fixed time t_k, we have the better estimate

$$\|\tilde{V}_{t_k}^{\Delta,n} - \bar{V}_{t_k}^{\Delta,\delta}\|_2 \leq [f]_{\text{Lip}} (C\Delta t)^{n+\frac{1}{2}} \sqrt{\binom{K+1}{n+1} - \binom{k}{n+1}}. \tag{8}$$

(*b*) *Let $(\bar{V}_{t_k}^\delta)_{k=0,K}$ denote the "fine" Snell envelope of the refined Euler scheme at times t_k defined by*

$$\bar{V}_{t_k}^\delta = \mathbb{P}\text{-ess sup}_{\tau \in \mathcal{T}_{t_k}^{\mathcal{F}}} \mathbb{E}[\mathrm{e}^{-r(\tau - t_k)} f(\tau, \bar{X}_\tau^\delta) | \mathcal{F}_{t_k}].$$

Then, for some constant C',

$$\|\tilde{V}_{t_k,0}^{\delta,n} - \bar{V}_{t_k}^\delta\|_{L^2(\mathbb{P})} \leq C\sqrt{\Delta t}.$$

Remark 3. A result similar to (*a*) can be obtained for $(\bar{V}_{t_k}^n)_{k=0,\dots,K}$, i.e. when $\mathcal{F}_{t_k}$ is replaced by $\hat{X}_{t_k}^n$ in the expectation defining $V_{t_k}^n$ at the cost of losing a $\sqrt{\Delta t}$ in the error estimate. Both quantities $\tilde{V}_{t_k}^{\Delta,n}$ and $\bar{V}_{t_k}^n$ do not coincide as $\hat{X}^n$ is not Markovian (it also depends on $\hat{X}^{n-1}$).

Table 1
Absolute error from the American payoff computed on the fine grid by a sequential LSMC standard algorithm and the same computed using the parareal iterative Algorithms 3.2 and 3.2bis. The coarse grid has K intervals; the coarse time step is $\Delta t/K$; the fine grid has a fixed number of points, hence each interval $(t_k, t_{t_{k+1}})$ has J sub-intervals. The top 4 lines of numbers correspond to Algorithm 3.2, while the last 4 lines correspond to Algorithm 3.2bis, for which a partial convergence estimate can be obtained, but which does not work as well numerically.

K	J	Δt	$n=1$	$n=2$	$n=3$	$n=4$
2	16	0.666667	0.60338	0.152339	0.0171122	0.000833293
4	8	0.4	0.237451	0.0437726	0.00217885	0.000725382
8	4	0.222222	0.0854814	0.0156243	0.000735309	0.000515332
16	2	0.117647	0.0257407	0.00120513	0.000439038	0.000262921
2	16	0.666667	0.5912463	0.1434691	0.0418341	0.0414722
4	8	0.4	0.2245711	0.0743709	0.0225051	0.0224303
8	4	0.222222	0.0740923	0.0205441	0.0072178	0.0072066
16	2	0.117647	0.0194701	0.0021758	0.0021592	0.0021509

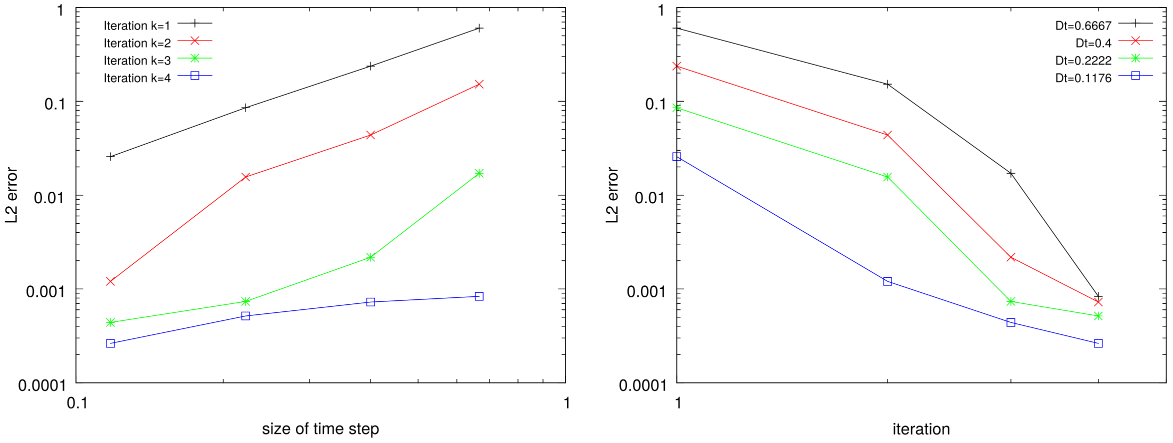

Fig. 1. Black–Scholes case: errors on the payoff versus Δt on the left for several values of n and versus n on the right for several values of Δt. Both graphs are for Algorithm 3.2 in log–log scales and indicate a general behavior of the error ϵ not incompatible with (3.1).

4. Numerical tests

The payoff is $f(t,x) = \mathrm{e}^{r(T-t)}(\kappa - x)^+$ with $X_0 = 36$, $r = 0.05$, $\kappa = 40$, $T = 2$. We have chosen $M = 100.000$ as in Longstaff and Schwartz's work [19]. The interpolation used in the LSMC is on the basis $\{1,\ x,\ x^2\}$, i.e. $P = 2$. The American payoff is then 4.478 at an early exercise $\tau = 0.634$.

4.1. *Convergence of the parareal algorithm*

4.1.1. *The Black–Scholes case*

Here the underlying asset is given by the Black–Scholes SDE, $\sigma(x,t) = \sigma_0 x$, $b(x,t) = rx$. In the test, $\sigma_0 = 0.2$. We have chosen a fine grid with $\delta t = T/32$. The free parameters are Δt, which governs the number of points on the coarse grid and n the number of parareal algorithms. The error between the American payoff computed on the fine grid by LSMC and the same computed by the parareal algorithm is displayed on Table 1 for both Algorithms 3.2 and 3.2bis.

The same information about convergence is now displayed in the two graphs in Fig. 1 for the errors versus Δt and the errors versus n. We were not able to decrease Δt to smaller values because the computing time becomes too large.

4.1.2. *The constant elasticity case*

The volatility is now a function of price [10]: $\sigma(x,t) = \sigma_0 x^{0.7}$. All parameters have the same values as above. The results are shown in Fig. 2.

4.2. *Multilevel parareal algorithm*

The previous construction being recursive, one can again apply the two-level parareal algorithm to LSMC on each interval $[t_k, t_{t_{k+1}}]$. The problem of finding the optimal strategy for parallelism and computing time is complex, because there are so many parameters; in what follows, the number of levels is $L = 4$; furthermore, when an interval with $J+1$ points is divided into sub-intervals, each one is endowed with a partition using $J+1$ points as well. So, if the coarse grid has K intervals,

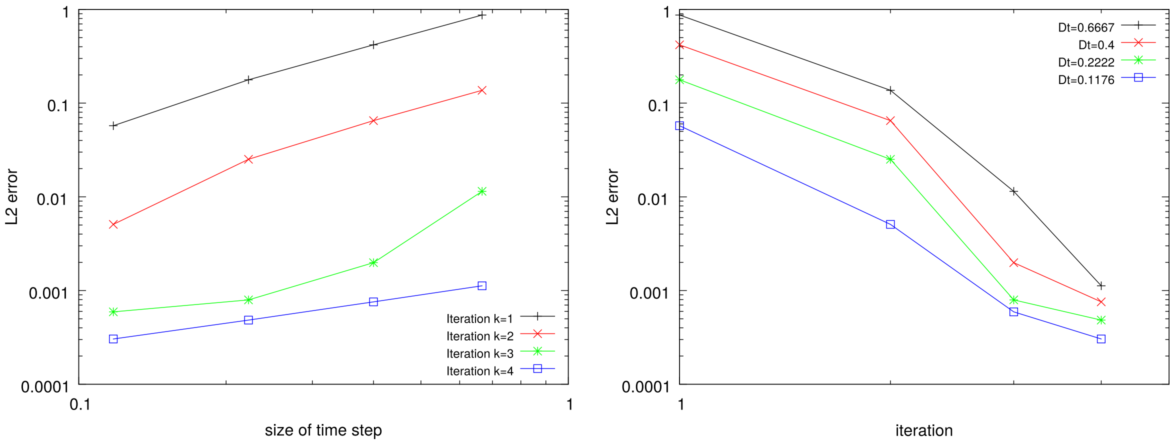

Fig. 2. Constant elasticity case: same legend as in Fig. 1.

Table 2
Absolute error between the computed payoff with the multilevels parareal method and the reference value of Longstaff–Schwartz. The number of levels is $L = 4$, each level is subdivided into K intervals; K^4 is the number of intervals at the deepest level.

K	K^4	$n = 1$	$n = 2$	$n = 3$	$n = 4$
2	16	0.353319	0.118118	0.0477764	0.0276106
3	81	0.208997	0.0390674	0.0225218	0.0186619
4	256	0.141692	0.0259727	0.0192504	0.0136437
5	625	0.105196	0.0220218	0.0179706	0.0129702

Table 3
Absolute error between the computed payoff with the multilevel parareal method and the reference value of Longstaff–Schwartz. There are $L = 4$ levels; at level $l - 1$, to obtain level l each interval is divided into J_l intervals. The errors are given versus the number of parareal iterations $n = 1, 2, 3, 4$. Note that all subdivisions give more or less the same precision; computationally and for parallelism the last one is the best.

Time-step				Total	Absolute-error			
J_1	J_2	J_3	J_4		$n = 1$	$n = 2$	$n = 3$	$n = 4$
6	5	4	3	360	0.108593	0.0305688	0.0202016	0.0142071
3	4	5	6	360	0.35365	0.0316707	0.0167488	0.0135151
20	2	2	2	160	0.0231221	0.0163731	0.0155624	0.013314
2	20	2	2	160	0.354477	0.0835047	0.0231243	0.0121775
2	2	20	2	160	0.351854	0.115285	0.015826	0.0137373
2	2	2	20	160	0.355166	0.119577	0.0444797	0.0110232

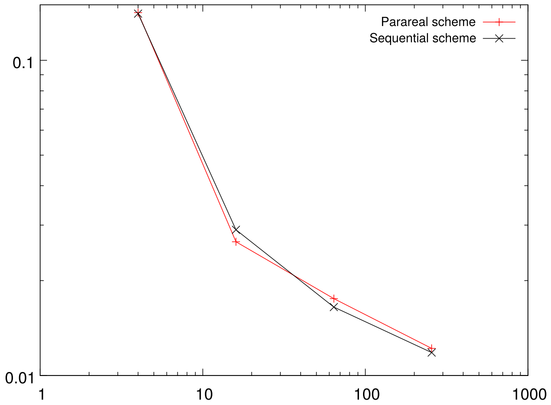

Fig. 3. Comparison between a standard LSMC solution and the parareal solution for the same number of time intervals at the finest level. The 4 points have respectively 1,2,3,4 levels; the first data point has one level and 4 intervals, the second has 2 levels and 16 intervals, the third one 3 levels and 64 intervals, the fourth one 4 levels and 256 intervals. The total number of time steps is on the horizontal axis, in log scale, and the error at $n = 2$ is on the vertical axis, in log scale as well.

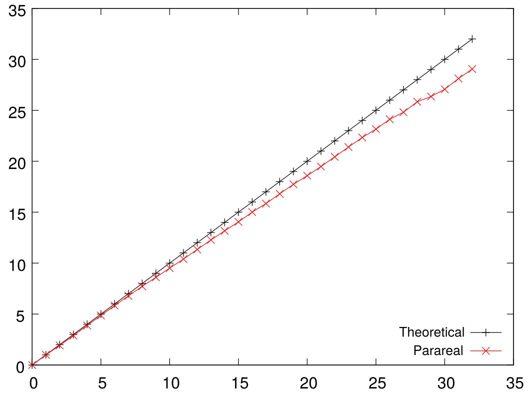

Fig. 4. Speed-up versus the number of processors, i.e. the parareal CPU time on a parallel machine divided by the parareal CPU time on the same machine but running on one processor. There are two levels only; the parameters are $K = 1, 2.., 32$, $n = 2$ and $J = 100$ so as to keep each processor fully busy.

the 4th grid has K^4 intervals. The results are compared with the reference value of Longstaff–Schwartz, 4.478, shown in Table 2 and in Fig. 3.

The number of parareal iterations is 4 and the error is displayed at each n. We have also carried out some tests with sub-partitions using $J \neq K$. Thus each level has its own number of points per interval, J_l. Errors shown on Table 3 are also shown in Fig. 3 for $n = 2$. It seems to be $O(K^4)$ for K small and $O(K^2)$ for K bigger. The method was implemented in parallel; each interval is allocated to a processor, at each level in a round-robin order. Almost perfect parallelism is obtained in our tests on a machine with 32 processors, as shown in Fig. 4 and Table 3.

References

[1] L. Abbas-Turki, B. Lapeyre, American option pricing on multi-core graphic cards, in: Proceedings of the Second International Conference on Business Intelligence and Financial Engineering, BIFE 2009, Beijing, China, 24–26 July 2009, 2009.
[2] Y. Achdou, O. Pironneau, Computation Methods for Option Pricing, Frontiers in Applied Mathematics, SIAM, Philadelphia, ISBN 0-89871-573-3, 2005, xviii + 297 p.
[3] F. Aitsahlia, P. Carr, American options: a comparison of numerical methods, in: L.C.G. Rogers, D. Talay (Eds.), Numerical Methods in Finance, in: Publications of the Newton Institute, Cambridge University Press, Cambridge, UK, 1997, pp. 67–87.
[4] G. Bal, Y. Maday, A “parareal” time discretization for non-linear PDE’s with application to the pricing of an American put, in: L.F. Pavarino, A. Toselli (Eds.), Recent Developments in Domain Decomposition Methods, Workshop on Domain Decomposition, Zurich, Switzerland, 7–8 June 2001, vol. 23, Springer-Verlag, Berlin, Heidelberg, New York, 2002, pp. 189–202.
[5] V. Bally, G. Pagès, A quantization algorithm for solving discrete time multidimensional optimal stopping problems, Bernoulli 6 (2003) 1003–1049.
[6] M. Benguigui, Valorisation d’options américaines et Value-At-Risk de portefeuille sur cluster de GPUs\CPUs, Ph.D. thesis, Université de Nice Sophia Antipolis, France, 2015.
[7] M. Benguigui, F. Baude, Towards parallel and distributed computing on GPU for American basket option pricing, in: Proc. International Workshop on GPU Computing in Cloud in conjunction with 4th IEEE International Conference on Cloud Computing Technology and Science, Taipei, Taiwan, 3–6 December 2012.
[8] A.P. Caverhill, N. Webber, American options: theory and numerical analysis, in: S. Hodges (Ed.), Options: Recent Advances in Theory and Practise, Manchester University Press, Manchester, UK, 1990, pp. 80–94.
[9] E. Clément, D. Lamberton, P. Protter, An analysis of a least squares regression method for American option pricing, Finance Stoch. 6 (2002) 449–471.
[10] J.C. Cox, J.E. Ingersoll, S.A. Ross, A theory of the term structure of interest rates, Econometrica 53 (1985) 385–407.
[11] D.M. Dang, C.C. Christara, K.R. Jackson, An efficient graphics processing unit-based parallel algorithm for pricing multi-asset American options, Concurr. Comput. 24 (18) (2012) 849–866.
[12] M. Fatica, E. Phillips, Pricing American options with least squares Monte Carlo on GPUs, in: Proc. 6th Workshop on High-Performance Computational Finance, Denver, CO, USA, 17–22 November 2013,.
[13] M. Gander, S. Vandewalle, Analysis of the parareal time-parallel, time-integration method, SIAM J. Sci. Comput. 29 (2) (2007) 556–578.
[14] J.-Y. Girard, I.M. Toke, Monte Carlo valuation of multidimensional American option through grid computing, in: Lect. Notes Comput. Sci., vol. 3743, 2006, pp. 462–469.
[15] Y. Hu, Q. Lu, Z. Cao, J. Wang, Parallel simulation of high-dimensional American option pricing based on CPU versus MIC, Concurr. Comput. 27 (15) (2015) 1110–1121.
[16] L. Khodja, J.-Y. Girard, R. Couturier, P. Spitéri, Parallel solution of American option derivatives on GPU clusters, Comput. Math. Appl. 65 (111) (2013) 1830–1848.
[17] P. Kloeden, E. Platen, Numerical Solution of Stochastic Differential Equations, Applications of Mathematics, vol. 23, Springer-Verlag, Berlin, 1999.
[18] J.-L. Lions, Y. Maday, G. Turinici, Résolution d’edp par un schéma en temps “pararéel”, C. R. Acad. Sci. Paris, Ser. I 332 (2000) 661–668.
[19] F.A. Longstaff, E.S. Schwartz, Valuing American options by simulation: a simple least squares approach, Rev. Financ. Stud. 14 (2001) 113–148.
[20] G. Pagès, Introduction to numerical probability for finance, available at: http://www.proba.jussieu.fr/dw/lib/exe/fetch.php?media=users:pages:probnum_gilp_pf16.pdf, 2016, 354 p.
[21] G. Pagès, B. Wilbertz, GPGPUs in computational finance: massive parallel computing for American style options, Concurr. Comput. 24 (18) (2012) 837–848.
[22] J. Wan, K. Lai, A. Kolkiewicz, K. Tan, A parallel quasi-Monte Carlo approach to pricing American options on multiple assets, Int. J. High. Perform. Comput. Networking 4 (2006) 321–330.