Pricing High-Dimensional American Options Using Local Consistency Conditions

Berridge, S.J.; Schumacher, J.M.

Publication date:
2004

No. 2004–19

PRICING HIGH-DIMENSIONAL AMERICAN OPTIONS USING LOCAL CONSISTENCY CONDITIONS

By S.J. Berridge, J.M. Schumacher

February 2004

ISSN 0924-7815

Pricing high-dimensional American options using local consistency conditions

S.J. Berridge* and J.M. Schumacher
Department of Econometrics and Operations Research
and Center for Economic Research (CentER),
Tilburg University, Postbus 90153, 5000 LE Tilburg, The Netherlands.
{s.j.berridge,j.m.schumacher}@uvt.nl

February 6, 2004

Abstract

We investigate a new method for pricing high-dimensional American options. The method is of finite difference type but is also related to Monte Carlo techniques in that it involves a representative sampling of the underlying variables. An approximating Markov chain is built using this sampling and linear programming is used to satisfy local consistency conditions at each point related to the infinitesimal generator or transition density. The algorithm for constructing the matrix can be parallelised easily; moreover once it has been obtained it can be reused to generate quick solutions for a large class of related problems. We provide pricing results for geometric average options in up to ten dimensions, and compare these with accurate benchmarks.

Keywords: American options, high-dimensional problems, free boundary problems, optimal stopping, variational inequalities, numerical methods, unstructured mesh, Markov chain approximation

MSC 2000: 35R35, 60G40, 65D15, 90C33

JEL Codes: C15, C61, C63

*Research supported by the Netherlands Organisation for Scientific Research (NWO).

1 Introduction

The pricing of American options is a problem that has remained inaccessible to closed form solution. It was also long assumed to be inaccessible to Monte Carlo techniques, but Tilley quashed this belief in his 1993 paper [22]. Simulation techniques are of particular importance for higher dimensional problems where conventional discretisation methods become intractable.

Methods for solving American and Bermudan option pricing problems have become increasingly important with the widespread use of options and the development of more and more complex contracts. Examples of potentially high-dimensional options include basket options, swaptions and real options. We consider "high-dimensional" problems to be those where the number of stochastic factors is at least three or four, and thus conventional grid techniques become unmanageable.

Much progress has been seen in the past decade in the area of Monte Carlo techniques, through the work of Barraquand and Martineau [1], Broadie and Glasserman [6] and more recently Longstaff and Schwartz [17], Tsitsiklis and Van Roy [23], Rogers [20], Haugh and Kogan [12], Boyle et al. [5], and through the method proposed in Berridge and Schumacher [2, 3, 4].

Most techniques proposed have centred around path generations of the process. This has the advantage that the points sampled are well adapted to the process, but the disadvantage that it is difficult to determine the expected value of continuation at each point. It is important to know the latter in order to make a stopping decision, and thus determine the early exercise premium.

The last method in the above list is the only one to consider a constant sampling of the state space over time. Since the method centres around an approximating Markov chain, it is simple to estimate continuation values on the grid using an appropriate Markov transition matrix. This method is thus more like a finite difference method, as opposed to the methods in [6, 17, 23, 5] which are more tree-like.

An important advantage of the irregular grid method proposed here is that the number of tuning parameters is small. Furthermore, convergence requires increasing only the number of grid points and the number of time steps, as with finite difference methods. In particular the method does not involve approximation of the value function or exercise region by basis functions.

We also note that using a constant grid allows implicit solutions to be easily obtained; for finite difference techniques this represents an increase in convergence speed from δt to δt^2 when considering European problems.

We proceed along the lines of [2, 3, 4] in that we approximate the value function on an irregular grid. We use a stable and more tractable method however for

approximating the transition probabilities; instead of taking a root of a transition matrix, we directly construct the transition probabilities using local consistency conditions presented in Kushner and Dupuis [16] in the parabolic case and similar conditions to construct the infinitesimal generator in the elliptic case. This allows us to use much larger grids, and thus obtain more accurate solutions.

Using the root method of [2, 3, 4] the grid size was limited to 3000 on a desktop computer, and averaging was needed over several experiments to obtain accurate solutions. We can now deal with grid sizes in the hundreds of thousands, and solutions from a single experiment are of sufficient accuracy that randomisation is no longer required.

The paper continues in Section 2 with a formulation of the problem of interest. Section 3 presents the proposed methodology, refinements are presented in Section 4 and experiments are carried out in Section 5. Section 6 concludes.

2 Formulation

2.1 The market

As in [2, 3, 4], we consider a complete and arbitrage-free market described by state variable $X(s) \in \mathbb{R}^d$ for $s \in [t, T]$ which follows a Markov diffusion process

$$dX(s) = \mu(X(s), s)ds + \sigma(X(s), s)dW(s) \tag{2.1}$$

with initial condition $X(t) = x_t$, and a derivative product on $X(s)$ with exercise value $\psi(X(s), s)$ at time s and value $V(s) = v(X(s), s)$ for some pricing function $v(x, s)$. The process $V(s)$ satisfies

$$dV(s) = \mu_V(X(s), s)ds + \sigma_V(X(s), s)dW(s) \tag{2.2}$$

where μ_V and σ_V can be expressed in terms of μ and σ by means of Itô's lemma. The terminal value is given by $V(\cdot, T) = \psi(\cdot, T)$, and intermediate values satisfy $V(\cdot, s) \geq \psi(\cdot, s)$, $s \in [t, T]$.

In such a market there exists a unique equivalent martingale measure under which all price processes are martingales. The risk-neutral process in this case is given by

$$dX(s) = \mu_{RN}(X(s), s)ds + \sigma(X(s), s)dW(s) \tag{2.3}$$

where μ_{RN} is the risk-neutral drift.

Our objective is to determine the current value $V(X(t), t)$ of the derivative product and the accompanying adapted exercise and hedging strategies τ and H:

$$\tau : \mathbb{R}^d \times [t, T] \rightarrow \{0, 1\} \tag{2.4}$$

$$H : \mathbb{R}^d \times [t, T] \rightarrow \mathbb{R}^d. \tag{2.5}$$

Supposing that one has an estimate $\hat{V}(t)$ of the derivative price, it is often important to specify an exercise rule $\hat{\tau}$ or a hedging strategy $\hat{H}$ in order for the buyer or seller respectively to be able to realise the estimated price.

2.2 Pricing

2.2.1 The primal formulation

The value of the derivative product is formulated in the primal problem as a supremum over stopping times

$$v(x_t, t) = \sup_{\tau \in \mathcal{T}} \mathbb{E}^{\mathbb{Q}}_{x_t} \left(e^{-r(\tau - t)} \psi(X(\tau)) \right) \tag{2.6}$$

where $\mathcal{T}$ is the set of stopping times on $[t, T]$ with respect to the natural filtration, the expectation is taken with respect to the risk-neutral measure $\mathbb{Q}$, and the initial value is $X(t) = x_t$.

2.2.2 The dual formulation

The dual formulation (see Rogers [20] or Haugh and Kogan [12]) forms a price by minimising the cost of the hedging strategy over martingales. Theorem 1 of [20] implies that the price is given by

$$v(x_t, t) = \inf_{M \in H_0^1} \mathbb{E}^{\mathbb{Q}}_{x_t} \left[\sup_{s \in [t,T]} \left(e^{-r(s-t)} \psi(X(s)) - M(s) \right) \right] \tag{2.7}$$

where H_0^1 is the space of martingales with $M(0) = 0$ and $\sup_{s \in [t,T]} |M(s)| \in L^1$. The infimum is attained at a certain martingale $M = M^*$.

2.2.3 The variational inequality formulation

Formulating the problem as a variational inequality invites implications from the large number of results that have been developed in this field, for example the work of Glowinski et al. [11]. Jaillet et al. [15] applied this approach to the analysis of American option pricing.

One must first define an elliptic operator $\mathcal{L}$ giving the diffusion of the process. This is given by

$$\mathcal{L} = \tfrac{1}{2} \mathrm{tr} \sigma \sigma' \frac{\partial^2}{\partial x^2} + \mu_{RN} \frac{\partial}{\partial x} - r \tag{2.8}$$

where r is the risk-free rate.

One must also specify a function space in which to work. Briefly one defines an inner product $\langle \cdot, \cdot \rangle$ and a bilinear form $a(\cdot, \cdot)$ on the Hilbert space H^1 satisfying

$$a(v, u) = \langle u, \mathcal{L}v \rangle \,. \tag{2.9}$$

The equivalent variational inequality formulation is then to find $v(x,t)$ such that

$$\begin{cases} v(x,s) - \psi(x,s) & \geq & 0 \\ u \geq \psi \text{ a.e. } \Rightarrow a(v, u-v) - \left\langle u - v, \frac{\partial v}{\partial t} \right\rangle & \geq & 0 \quad \text{ a.e. } [t,T] \end{cases} \tag{2.10}$$

for $(x,s) \in \mathbb{R}^d \times [t,T]$ with the terminal condition $v(\cdot,T) \equiv \psi(\cdot,T)$.

2.2.4 The complementarity formulation

The variational inequality formulation is not directly amenable to computation. For this reason it is convenient to reformulate it as a complementarity problem. Let $\mathcal{L}$ be the related diffusion operator; then the option value is found by solving the complementarity problem

$$\begin{cases} \frac{\partial v}{\partial t} + \mathcal{L}v & \leq & 0 \\ v - \psi & \geq & 0 \\ \left(\frac{\partial v}{\partial t} + \mathcal{L}v\right)(v - \psi) & = & 0 \end{cases} \tag{2.11}$$

for $(x,s) \in \mathbb{R}^d \times [t,T]$ with the terminal condition $v(\cdot,T) \equiv \psi(\cdot,T)$.

Such a problem can be solved using standard PDE discretisation techniques, with some modifications to account for the inequalities.

2.3 Consequences

In solving the pricing problem we divide the time-state space into two complementary regions: the continuation region where it is optimal to hold the option and the stopping region where it is optimal to exercise. In the continuation region the first line of (2.11) is active and the stopping rule says not to exercise. In the stopping region the second line of (2.11) is active and the stopping rule says to exercise.

In all formulations presented, high dimensionality poses a practical problem since functional approximation in a high-dimensional space is called for.

3 Methodology

The basic methodology presented is similar to that of [2, 3, 4], with the exception of the manner in which the transition matrix is constructed. This is done in the sequel using the local consistency conditions presented in Kushner and Dupuis [16], and a modification of these conditions is used to find an approximation to the infinitesimal generator. These conditions ensure that the approximating Markov chain has a local mean and variance that match those of the continuous process.

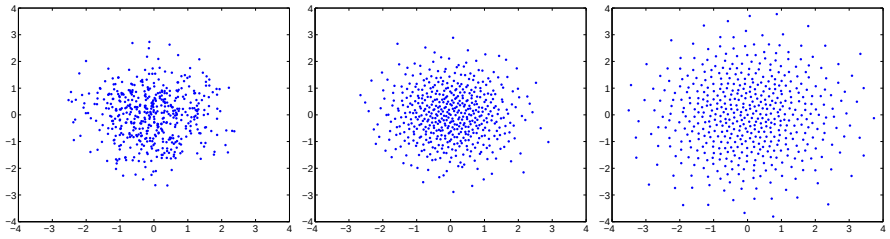
3.1 Irregular grid

We first briefly review the irregular grid methodology presented in [2, 3, 4]. We define an irregular grid to be a representative sampling of the state space

$$\mathcal{X} = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d. \tag{3.1}$$

The method of sampling is to be specified at a later stage, but one can think of it as a low discrepancy or low distortion set (see for example Bally and Pagès [19]) which is dense in the entire state space as $n \to \infty$.

Examples of possible grids in two dimensions are presented in Figure 3.1. As in the case of Monte Carlo integration, it is expected that low discrepancy (e.g. Sobol') and low distortion grids will lead to faster convergence than random grids. For results regarding integration see Evans and Swartz [10] and Pagès [19].



(a) Random (b) Sobol' (c) Low Distortion

Figure 3.1: Grids with 500 points adapted to the normal density.

In order to simplify the analysis we now make the assumption that the risk-neutral process is a d-dimensional time homogeneous diffusion process

$$dX(s) = \mu_{RN}(X(s))ds + R(X(s))dW(s) \tag{3.2}$$

where $R'R$ is the Cholesky decomposition of the state-dependent covariance matrix $\Sigma(X(s))$ and X and W are of the same length d. This assumption is not necessary for the method to work; it merely simplifies some aspects and allows for a clearer exposition.

3.2 Approximation of Markov chain

We consider approximating the risk-neutral process (3.2) using a discrete state, discrete time Markov chain where the states are exactly the points in our irregular grid $\mathcal{X}$ and the time step is δt.

The Markov transition matrix P is constructed in such a way as to satisfy the local consistency conditions given in Kushner and Dupuis [16]. We require[1] for each state $i = 1, \ldots, n$

$$\begin{aligned} \Sigma(x_i)\delta t &= \sum_{j=1}^{n}(x_j - x_i - \mu_{RN}(x_i)\delta t)(x_j - x_i - \mu_{RN}(x_i)\delta t)' p_{i,j} \\ \mu_{RN}(x_i)\delta t &= \sum_{j=1}^{n}(x_j - x_i)p_{i,j} \\ 1 &= \sum_{j=1}^{n} p_{i,j} \\ p_{i,j} &\geq 0 \end{aligned} \tag{3.3}$$

where $p_{i,j}$ is the (i,j)th entry of P.

One must solve for each state i a feasibility problem over the $p_{i,j}$. The number of equality constraints in the problem is given by $\eta_d + 1$ where

$$\eta_d = \frac{1}{2}d(d+3) \tag{3.4}$$

and the number of variables is n. In the problems we consider, η_d is much smaller than n.

In practise one can impose the extra condition that the transitions should only be allowed to close neighbours of each point. Computationally this means that we only need to consider a small number of transitions k where $\eta_d + 1 < k \ll n$, thus dramatically reducing the complexity of the problem.

It is also useful to specify a linear objective function to optimise the proximity of transitions. That is, to satisfy the local consistency conditions using points as close as possible to the mean. The linear objective function, to be minimised, should have a coefficient relating to point j which is an increasing function of the distance $||x_i - x_j||$. Let us denote the objective function by $f_i \cdot p_i$ where p_i is the ith row of P.

We thus pose for each point i a linear program $\min f_i \cdot p_i$ subject to (3.3). In experiments we found that a convenient specification for f is $f_j = k^3$ where x_j is the kth nearest neighbour of $x_i + \mu_{RN}\delta t$.

We note that the solution to the linear program will in general be a corner solution using as many zero variables as possible; the number of nonzero transition

[1] The formulation in [16] is more general in that it allows $o(\delta t)$ terms to be added on the RHS of the first two conditions.

probabilities per point is the minimum number, $\eta_d + 1$. This is a consequence of Corollary 7.11 in Schrijver [21], and of the fact that the constraint matrix has $\eta_d + 1$ rows. Note that the points with positive weights are not necessarily the $\eta_d + 1$ nearest neighbours of $x_i + \mu_{RN}\delta t$, since these may not satisfy the feasibility conditions; the points form rather the closest possible feasible set (with respect to the objective function).

3.3 Approximation of infinitesimal generator

Rather than approximating transition probabilities, one may attempt to approximate the infinitesimal generator directly. This amounts to constructing a discrete space, continuous time approximation to the problem.

Constructing an approximation to the infinitesimal generator allows quick reconstruction of transition probabilities for arbitrary time steps δt, or for scaling the effect of the diffusion operator, through a first order approximation. Consequently this method is preferred over that of Section 3.2, provided we do not have a large state-dependent drift. We assume the latter in this section.

In the case of a non-state dependent drift, we refer the reader to Section 4.5 where we introduce a simple transformation of the continuous process to eliminate a risk-neutral drift that depends deterministically on time.

We start with the problem (3.3), and define

$$a_{i,j} = \frac{1}{\delta t}(p_{i,j}(\delta t) - \delta_{ij}) \tag{3.5}$$

where δ_{ij} is the Kronecker delta. As $\delta t \to 0$ in (3.5) we obtain elements of the infinitesimal generator matrix A.

Substituting (3.5) into (3.3) and letting $\delta t \to 0$ yields the new feasibility problem

$$\begin{aligned} \Sigma(x_i) &= \textstyle\sum_{j \neq i}(x_j - x_i)(x_j - x_i)' a_{i,j} \\ \mu_{RN}(x_i) &= \textstyle\sum_{j \neq i}(x_j - x_i) a_{i,j} \\ a_{i,j} &\geq 0 \end{aligned} \tag{3.6}$$

and $a_{i,i} = -\sum_{j \neq i}^{n} a_{i,j}$. Note that (3.6) now contains only η_d equality constraints, one less than (3.3).

The same considerations as in Section 3.2 are also applied in this case. We solve for each point i a linear program $\min f \cdot a_i$ subject to (3.6) and $a_{i,j} \geq 0$ where a_i is the ith row of A with the diagonal entry omitted. Following from the observation at the end of Section 3.2, we again expect a maximum of $\eta_d + 1$ nonzero entries per row of A.

We note that when a large drift term is present, one may be able to satisfy the local consistency conditions (3.6), but this may require using points x_j which are nonlocal to x_i. This method differs from the usual method of lines in that here we produce a stable system before checking for localness of the neighbours, whereas in the usual method one selects the neighbours a priori before building the equations and finally considering stability (see for example Hundsdorfer and Verwer [14]).

3.4 Time stepping

Given a transition matrix P, corresponding to time step δt, the option pricing problem can be solved using dynamic programming on the discretised Markov chain. Namely, one solves the problem

$$\begin{aligned} v(T) &= \psi \\ v(t_k) &= \max\left(\psi, e^{-r\delta t} P v(t_{k+1})\right) \end{aligned}$$

for $t_k = k\delta t$ and $k = K-1, \ldots, 0$ where K is the number of time steps considered, v is a vector of values at grid points and ψ is a vector of payoffs at grid points. The resulting solution $v(x, 0)$ is an exact solution to the approximating Markov chain.

Given the infinitesimal generator A, one can form a first order approximation to the transition matrix $P \simeq I + A\delta t$ and proceed as above. Alternatively, it is possible to solve the problem to a higher order using the matrix exponential

$$\begin{aligned} v(T) &= \psi \\ v(t_k) &= \max\left(\psi, e^{-r\delta t} e^{A\delta t} v(t_{k+1})\right) \end{aligned}$$

for $k = K-1, \ldots, 0$. Since A is sparse, the effect of the matrix exponential can be calculated efficiently using Krylov subspace methods, see for example Druskin and Knizhnerman [9] or Hochbruck and Lubich [13].

The above time stepping methods are suitable for Bermudan pricing problems with δt being the period between exercise possibilities. We expect convergence to the Bermudan solution as $n \to \infty$, and convergence to the American solution as $\delta t \to 0$.

When considering a truly American problem, it is useful to consider Crank-Nicolson and implicit solutions. In particular the Crank-Nicolson method is known to converge at a rate δt^2 for the European problem as opposed to δt for the explicit and implicit methods, and implicit methods are known to be unconditionally stable for solving sequences of LCPs (see Glowinski et al. [11]).

The Crank-Nicolson method corresponding to the truly American problem is the following system with $\theta = \frac{1}{2}$

$$\begin{aligned} v(T) &= \psi \\ 0 \leq (v(t_k) - \psi) &\perp \left(e^{-\theta A \delta t} v(t_k) - e^{-r\delta t} e^{(1-\theta)A\delta t} v(t_{k+1}) \right) \geq 0 \end{aligned} \tag{3.7}$$

for $k = K-1, \ldots, 0$. The second line is a linear complementarity problem (LCP). There are many methods available for solving LCPs, including the projected successive overrelaxation (PSOR) method proposed in Cryer [7]. Another possible candidate is linear programming, which is used for example by Dempster and Hutton [8] to solve the one-dimensional American option pricing problem.

3.5 Summary of the algorithm

We present a concise statement of the proposed algorithm as Algorithm 1. The generation of the matrices A can be done in advance for a given grid $\mathcal{X}$, with obvious changes to the algorithm.

Algorithm 1 Proposed algorithm for solving high-dimensional American option pricing problems.

Choose the grid size n
Generate a QMC grid $\mathcal{X}$
Compute the generator matrix A
Choose the time step $\delta t > 0$ and implicitness $\theta \in [0, 1]$
Solve the linear complementarity problems (3.7)

4 Fine tuning and extensions

We now mention some implementation issues and refinements of the method. These issues are not essential to the method, but may improve performance and allow quicker execution for a given required accuracy.

4.1 Grid specification

In the presentation so far, we have taken the grid $\mathcal{X}$ to be given; we now consider ways one might specify the grid.

Taking inspiration from the literature on MC and QMC integration, we first suggest that the grid be constructed using low discrepancy (Niederreiter [18]) or low distortion (Pagès [19]) methods. Just as in the regular grid case, we expect the

error to be related to the separation of grid points, more specifically the separation of grid points having positive weights in the generator matrix.

Importance sampling considerations tell us that the most efficient grid density is given by the density of the process itself. Given our suggestion of a constant grid (for efficiency reasons), we cannot provide the most efficient importance sampling at all times. However, given the restriction to a constant grid, we can still provide an acceptable importance sampling.

As outlined in Evans and Swartz [10], the rate of convergence for importance sampling of normal densities using normal importance sampling functions is most damaged when the variance of the importance sampling function is less than that of the true density. Conversely, convergence rates are not greatly affected when the variance of the importance sampling function is greater than that of the true density. The situation we should try to avoid is that the process has a significant probability of lying in the "tails" of the grid density.

A further consideration is the minimisation of boundary effects on the solution. This suggests that the grid covariance should be larger than the covariance of the process.

In [2, 3, 4], where a root method was used to construct transition probabilities, and the process considered was a five-dimensional Brownian motion with drift, a grid covariance of 1.5 times the process covariance at expiry was found to give the best convergence rate when tested against grids with covariances of 1.0 and 2.0 times the covariance at expiry.

4.2 Boundary region and boundary conditions

It is clear that (3.3) and (3.6) may be infeasible for some i. In such a case we say that x_i is an implied boundary point, otherwise it is an implied interior point. Given nondegenerate Σ and a well-adapted grid, one expects that the implied boundary points will indeed lie at the extremities of the grid, and the implied interior points away from the extremities.

One may specify appropriate boundary conditions in this region to reflect the behaviour of the process. In the experiments we let these points be absorbing, which is appropriate for value functions having a linear behaviour at the boundary. One may also apply Dirichlet, Neumann or mixed conditions using neighbours in the grid.

It would be useful to know a priori which points are likely to be in the implied boundary, since we would like to avoid trying to solve infeasible linear programming problems. In practise however it is difficult to do this even for simple cases.

A plot of the boundary behaviour for a 500-point low distortion grid in two dimensions is given in Figure 4.1. Notice in this case that the number of infeasible

points is 21, this being about 4% of the total.

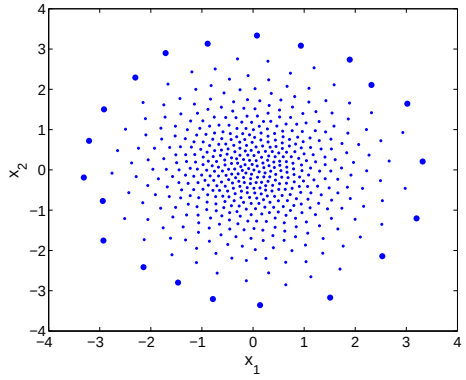


Figure 4.1: Interior points (small) and boundary points (large) on a normal low distortion grid for $d = 2$, $n = 500$.

If one assumes a distribution for the neighbours over which (3.3) or (3.6) is to be solved, then one can quantify the probability of feasibility. Near the boundary of the grid, there may be a low density of points on the boundary side, and thus the probability of feasibility changes.

For example, if our grid consists of n independent standard normal draws, we can calculate the expected number of grid points in a halfspace away from the centre of the grid at some radius r. One can then say what the minimum number of points n is where the expected number of grid points in the halfspace away from the grid centre at radius r is less than some bound.

Let us set this bound to be $\frac{1}{2}\eta_d$, where η_d is given in (3.4), a very optimistic bound but useful to illustrate the approximate behaviour of the boundary. Requiring an expected number of $\frac{1}{2}\eta_d$ points in the halfspace away from the center implies a boundary radius of

$$r = \Phi^{-1}\left(1 - \frac{1}{2n}\eta_d\right) \tag{4.1}$$

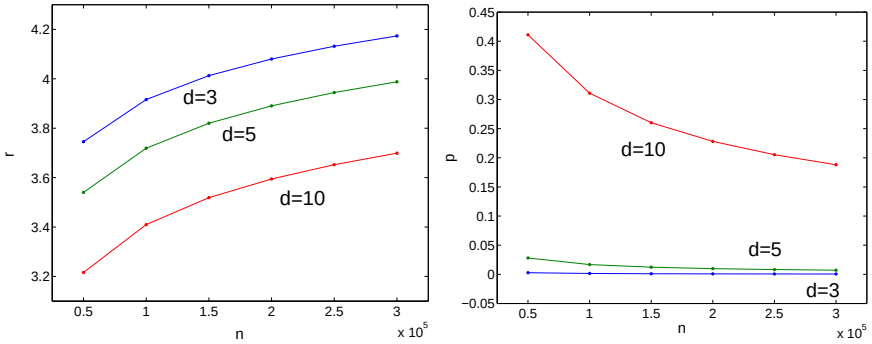
where Φ is the cumulative normal distribution function. In order to find the expected number of boundary points we then note that the squared norm of a standard normal variable in d dimensions is a chi square random variable with d degrees of freedom. Thus, if the boundary region is defined by $\left\{x : \|x\|^2 \geq r^2\right\}$, then the expected numbers of interior and boundary points are

$$\mathbb{E}N_i = n\Psi(r^2, d) \tag{4.2}$$

$$\mathbb{E}N_b = n\left(1 - \Psi(r^2, d)\right) \tag{4.3}$$

respectively where Ψ is the chi square cumulative distribution function.

Plots of the radius and expected number of boundary points are presented in Figure 4.2 for $d = 3, 5, 10$ and n up to $300,000$.



(a) Radius (b) Proportion

Figure 4.2: Naive prediction for the radius of the boundary and the proportion of points which are in the boundary region for a standard normal grid.

Experimentally we find that (4.1) underestimates the implied radius for lower dimensions and overestimates it for higher dimensions (see Section 5 for numerical results). The latter is not surprising since one generally requires more than the minimum number of points η_d to satisfy the feasibility conditions (3.3) and (3.6).

Finally we mention that in estimating the boundary, we prefer an underestimate to an overestimate. An overestimate of the boundary may lead us to waste a considerable amount of computing time trying to solve infeasible linear programs. An underestimate on the other hand just results in the grid having extra boundary points. The latter does not add a significant amount of overhead to the method, the effect being limited to a slight increase in complexity of the nearest neighbour problem and extra zero rows to the sparse generator matrix.

4.3 Parallelism

In the language of computer science, problems (3.3) and (3.6) are said to be embarrassingly parallel. This refers to the fact that a speedup linear in the number of processors can be trivially achieved. For example, having a large number n of linear programs to solve and m computers, we can reduce the time by a factor $1/m$ by solving n/m linear programs per computer, assuming $n > m$ and that the communication time between the computers is negligible. We make use of this point when conducting the experiments, using a distributed computing environment to solve the linear programs.

4.4 Control variates and Richardson extrapolation

To obtain more accurate solutions we consider variance reduction and extrapolation techniques.

Variance reduction is already used in the method in that the grids are constructed using points designed to cover the state space evenly according to the process density at expiry. In the current context the idea of control variates is also very easy to apply since the European solution is usually highly correlated with the Bermudan and American solutions. Since the European price is easy to determine to a high degree of accuracy, it constitutes an ideal control variate.

The concept of extrapolation is also useful once we have an idea of how the error behaves with increasing n. In Section 5.5 below, experimental evidence is given which implies the estimates behave asymptotically as

$$\hat{v}_n = v + c_1 n^{c_2/d} \tag{4.4}$$

for some constants c_1, c_2, which may be estimated. Here we assume that the error is always of the same sign, which may be indicated for example by a monotone behaviour of the approximations.

4.5 Matrix reuse

Given that generating the transition and infinitesimal generator matrices is an expensive operation compared to the final time stepping procedure, it is of interest to know under which conditions these matrices can be reused for related problems. It is clear that a single matrix can be reused for as many different payoff functions as required; it can also be reused for processes with different risk-neutral drifts and covariances as follows.

Suppose that a transition or infinitesimal generator matrix has been constructed for a process with covariance matrix I and zero risk-neutral drift on the grid $\mathcal{X}$. Let us construct the grid $\mathcal{Y}$ where $y_i = R'x_i$, R being a Cholesky factor of the covariance matrix Σ. The implied covariance of the transition or infinitesimal generator matrix on $\mathcal{Y}$ is now Σ.

Suppose now that our process has covariance Σ, and constant (nonzero) risk-neutral drift μ. Consider now the time dependent grid $\mathcal{Y}_k$ where the subscript k corresponds to time $k\delta t$ and $y_k = x + k\mu\delta t$. The implied covariance of the transition or infinitesimal generator matrix remains Σ, but the implied drift is now μ.

Two simple extensions to the time homogeneous problem are those in which the risk-neutral drift is deterministically time-dependent and the covariance matrix

is scaled over time,

$$dX(s) = \mu_{RN}(s)ds + \alpha(s)RdW(s). \tag{4.5}$$

The most convenient way to deal with the drift term is to incorporate the drift in the payoff function. This amounts to the change of variables

$$X_0(s) = X(s) - \int_0^s \mu_{RN}(u)du, \tag{4.6}$$

the new process having zero drift

$$dX_0(s) = \alpha(s)RdW(s) \tag{4.7}$$

and the payoff being

$$\psi_0(x_i, s) = \psi\left(x_i + \int_0^s \mu_{RN}(u)du\right). \tag{4.8}$$

The scaled covariance term can be accommodated by manipulating the time step. By using time step $\alpha(s)\delta t$ at time s in place of δt, we achieve a covariance of $\alpha(s)^2\Sigma$ as required.

4.6 Grid expansion

Grid expansion relates the size of the grid to the variance of the process. A convenient way to generate a grid is to sample the process at expiry; one thus obtains a grid $\mathcal{X}$ that becomes dense in the state space as $n \to \infty$. For a finite n however one can ask how well the process can be represented on $\mathcal{X}$. For example if we consider a standard d-dimensional Brownian motion on $s \in [0, 1]$, the process density at expiry is $\mathcal{N}(0, I)$. If the implied boundary begins at $r < 2$ for example, there is a nonnegligible chance of the discrete Markov process hitting the absorbing boundary before expiry, thus reducing the accuracy of the solution.

In this case we can set a lower limit r_0 for the implied boundary, for example $r_0 = 4$ for which the process has a negligible chance of hitting the boundary. This limit can be achieved by expanding the grid; to do this, one scales the grid points by a factor r_0/r and the generator matrix entries by a factor r/r_0, thus removing the boundary effects while preserving local consistency.

The grid expansion factor allows us to make a tradeoff between errors caused by the boundary and errors related to the discretisation. The higher the factor applied in the grid expansion, the lower the effect from the boundaries but the coarser the grid becomes and hence the higher the discretisation error.

4.7 Partially absorbing boundaries

Infeasibility of points in the boundary region is usually caused by a lack of points in the halfspace away from the center of the grid. If the grid boundary looks locally linear, as in a spherical grid, it is possible that the infeasibility is only in this direction, and not "along" the boundary.

In this case it may be useful to consider partially absorbing boundaries in which one only tries to satisfy local consistency conditions in the direction tangent to the boundary. In the case of a normal grid this amounts to requiring a zero variance along lines through the grid center for points in the boundary layer. This type of boundary condition has not been employed in the current study.

5 Experiments

A major hurdle in testing algorithms for pricing high-dimensional American options is the difficulty of verifying results. One common method is using out-of-sample paths to estimate the value of the exercise and hedging strategies implied by the model. Another, which we use here, is to use benchmark results from a special case that can be solved accurately. In the following we introduce benchmark results and then test the proposed method against those results.

5.1 Geometric average options

We choose to focus on geometric average options, since the pricing problem for these options can be reduced to a one-dimensional problem. The one-dimensional problems can be solved to a high degree of accuracy, thus providing benchmark results for the algorithm.

A geometric average put option written on d assets following the risk-neutral process (2.3) has payoff function

$$\psi(s) = \left(K - \left(\prod s_i \right)^{1/d} \right)^+ \tag{5.1}$$

where s is the asset value and K is the strike price of the option. Assuming a complete and arbitrage free market with the log asset prices following a multivariate Brownian motion with constant covariance Σ, we have a constant risk-neutral drift

$$\mu_{RN} = r\mathbb{1} - \frac{1}{2}\text{diag}\Sigma. \tag{5.2}$$

5.2 Benchmarks

Using Itô's lemma with $Y = f(X) = \overline{X}$, we find that Y follows the risk-neutral process

$$dY(s) = \frac{1}{d}\sum_{i=1}^{d} dX_i(s) \tag{5.3}$$

$$= \tilde{\mu}ds + \tilde{\sigma}dW(s), \tag{5.4}$$

the parameters of the diffusion being given by

$$\tilde{\mu} = r - \frac{1}{2d}\sum_{i=1}^{d}\sigma_i^2 \tag{5.5}$$

$$\tilde{\sigma}^2 = \frac{1}{d^2}\sum_{i=1}^{d}\left(\sum_{j=1}^{d} R_{ij}\right)^2. \tag{5.6}$$

The option is thus equivalent to a standard put option on an asset with starting value $\exp\{\overline{X}_0\}$, strike price K, risk-free rate r and continuous dividend stream

$$\delta = \frac{1}{2}\left(\frac{1}{d}\sum\sigma_i^2 - \tilde{\sigma}^2\right). \tag{5.7}$$

In Table 5.1 we provide benchmark results for geometric put options written on up to ten assets, with starting asset values $S_i = 40$, for all i and strike price 40. The risk-free rate is taken as 0.06, the volatilities $\sigma_i = 0.2$ for all i, and correlations $\rho_{ij} = 0.25$, $i \neq j$.

5.3 Experimental details

Using the methodology proposed in Section 3, we conducted experiments to find the value of the geometric average put options given above.

We used six different grid sizes ranging from $50,000$ to $300,000$, and two types of grids consisting of normal Sobol' points and normal low distortion points with a covariance corresponding to 1.5 times the process covariance at expiry. The transition matrices were generated using distributed computing software in a Matlab environment. A maximum of $20\eta_d$ nearest neighbours were considered when trying to satisfy the local consistency conditions, where η_d is defined in (3.4).

We consider the pricing problem for European options, Bermudan with ten exercise opportunities and true American where the option can be exercised at

d	$\tilde{\sigma}^2 \times 10^2$	$\delta \times 10^2$	European	Bermudan	American
1	4.000	0.000	2.0664	2.2930	2.3196
2	2.500	0.750	1.5553	1.7557	1.7787
3	2.000	1.000	1.3468	1.5380	1.5597
4	1.750	1.125	1.2318	1.4193	1.4392
5	1.600	1.200	1.1585	1.3421	1.3625
6	1.500	1.250	1.1077	1.2893	1.3094
7	1.429	1.286	1.0703	1.2504	1.2703
8	1.375	1.313	1.0416	1.2207	1.2404
9	1.333	1.333	1.0189	1.1971	1.2167
10	1.300	1.350	1.0004	1.1779	1.1974

Table 5.1: Benchmark results for geometric average options in dimensions 1-10. Also displayed are the variance $\tilde{\sigma}^2$ and continuous dividend δ for the equivalent one dimensional problem.

any time up to expiry. For the European and Bermudan problems we used the Crank-Nicolson method with 100 time steps. For solving the linear systems we used the conjugate gradients squared (CGS) and generalised minimum residual (GMRES) methods, the latter being slower but more robust. For the American problems we used projected successive overrelaxation (PSOR) to solve the linear complementarity problems, with 1000 time steps. While it is not necessary to use such a large number of time steps in practise, we wanted to focus on the error with respect to the space discretisation. Having a small enough δt causes the error resulting from time discretisation to be negligible in comparison, and thus allows a more accurate assessment of the error resulting from space discretisation.

5.4 Experimental results

We present results in Tables 5.2–5.4 for prices obtained using normal Sobol' grids for the Bermudan, American and European cases respectively. The results for low distortion grids are presented in Tables 5.7–5.9.

Tables 5.5 and 5.6 show the results on normal Sobol' grids for Bermudan and American options when the European is used as control variate. Tables 5.10 and 5.11 show the same for low distortion grids.

Figures 5.1 and 5.2 present the results graphically for normal Sobol' grids. The results for low distortion grids are shown in Figures 5.3 and 5.4. We see that the error increases with dimension to about 5–10% for $d = 10$. The control variate improves the results dramatically, the error for $d = 10$ being now less than 1%.

When using the European control variate we see that the results are biased upwards, whereas the raw results are biased downwards. This is probably due to the upward bias introduced by the convexity of the max operator which appears in the Bermudan and American problems, but not in the European problem.

In one and two dimensions the generator matrix became numerically unstable for the grid sizes we consider; we have thus not presented results for these low dimensions here. This lack of convergence is due to the finite precision arithmetic, and not to instability in the sense that the generator matrix has unstable eigenvalues (i.e. eigenvalues having positive real part). The method has been found to work very well in one and two dimensions, but for smaller grid sizes.

5.5 Error behaviour

Drawing a parallel with regular grid methods, we expect the error to be related to δx, the distance between grid points with positive weights in A. In a regular grid with the same number of points N in each dimension we have $n = N^d$ points in total, and the distance to the nearest point is simply $n^{-1/d}$. The error when using a standard finite difference method is of order δx^2, or $n^{-2/d}$.

We thus propose modelling irregular grid errors as in the regular grid case, allowing for a scalar factor in the exponent as well as a multiplicative factor:

$$\log|\varepsilon| = c_1 + c_2 \frac{\log n}{d}. \tag{5.8}$$

In Figures 5.5 and 5.6 we present plots of the log absolute error versus $\log(n)/d$, and in Tables 5.12 and 5.13 the regression results. Referring to our assumption of error behaviour (5.8) we find that the complexity is accurately modelled by the given relationship in all three cases (for suitable c_1, c_2). The linear relationships observed, on the log scale, strongly suggest that the algorithm has exponential complexity. We note that the behaviour in the Sobol' and low distortion cases is very similar, with the European and Bermudan prices showing about the same asymptotic relationship, and with American errors showing a slightly faster rate in the Sobol' case, although this is barely significant.

The convergence rate for finite difference methods used to solve PDE problems on regular grids is $1/\delta x^2$, or $n^{-2/d}$ which here translates to $c_2 = -2$. From this point of view our method seems to be slightly slower in convergence than the regular grid method, although this is barely significant. This may be due to the average δx being larger as a function of the grid size in the irregular grid case.

The given model for errors implies that the amount of work required to obtain solutions to a certain accuracy increases exponentially with dimension. This may seem pessimistic in that the curse of dimensionality is not broken; however

d	5×10^4	10×10^4	15×10^4	20×10^4	25×10^4	30×10^4
3	1.5370	1.5375	1.5376	1.5376	1.5377	1.5377
4	1.4135	1.4147	1.4155	1.4161	1.4163	1.4166
5	1.3300	1.3329	1.3345	1.3360	1.3365	1.3371
6	1.2532	1.2630	1.2667	1.2757	1.2766	1.2780
7	1.1981	1.2133	1.2137	1.2305	1.2311	1.2313
8	1.1489	1.1664	1.1672	1.1891	1.1938	1.1807
9	1.1116	1.1255	1.1351	1.1530	1.1514	1.1612
10	1.0901	1.1080	1.1078	1.1129	1.1242	1.1218

Table 5.2: Results for Bermudan geometric average put options in dimensions 3-10 using normal Sobol' grids.

d	5×10^4	10×10^4	15×10^4	20×10^4	25×10^4	30×10^4
3	1.5584	1.5588	1.5590	1.5591	1.5592	1.5592
4	1.4332	1.4347	1.4357	1.4362	1.4365	1.4369
5	1.3489	1.3522	1.3537	1.3551	1.3557	1.3563
6	1.2721	1.2818	1.2858	1.2940	1.2951	1.2965
7	1.2182	1.2325	1.2331	1.2482	1.2491	1.2492
8	1.1693	1.1864	1.1870	1.2071	1.2114	1.1993
9	1.1316	1.1460	1.1549	1.1715	1.1700	1.1802
10	1.1102	1.1281	1.1267	1.1324	1.1433	1.1414

Table 5.3: Results for American geometric average put options in dimensions 3-10 on normal Sobol' grids.

the method we use has definite advantages over regular grid methodology in high dimensions. In particular we note that the number of grid points n can be chosen freely, the grid points can be adapted to the process density and the number of boundary points can be substantially reduced for unbounded problems. Regarding the last point, Section 5.7 provides a comparison between the number of boundary points found in regular and normally distributed grids. The results suggest that the proposed method can handle option pricing problems up to dimension ten, which sets it aside from traditional finite difference methods which start to become unwieldy in dimension three or four.

d	5×10^4	10×10^4	15×10^4	20×10^4	25×10^4	30×10^4
3	1.3461	1.3463	1.3465	1.3465	1.3465	1.3465
4	1.2274	1.2286	1.2293	1.2302	1.2304	1.2304
5	1.1482	1.1505	1.1520	1.1541	1.1545	1.1549
6	1.0716	1.0813	1.0849	1.0977	1.0984	1.0993
7	1.0156	1.0275	1.0318	1.0527	1.0541	1.0545
8	0.9624	0.9792	0.9848	1.0123	1.0151	0.9943
9	0.9231	0.9406	0.9507	0.9735	0.9755	0.9802
10	0.8966	0.9203	0.9277	0.9340	0.9418	0.9424

Table 5.4: Results for European geometric average put options in dimensions 3-10 on normal Sobol' grids.

d	5×10^4	10×10^4	15×10^4	20×10^4	25×10^4	30×10^4
3	1.5382	1.5382	1.5382	1.5379	1.5379	1.5379
4	1.4179	1.4178	1.4180	1.4177	1.4177	1.4179
5	1.3403	1.3409	1.3410	1.3404	1.3405	1.3407
6	1.2892	1.2893	1.2894	1.2857	1.2858	1.2863
7	1.2527	1.2560	1.2521	1.2481	1.2473	1.2470
8	1.2281	1.2288	1.2240	1.2184	1.2203	1.2279
9	1.2074	1.2038	1.2033	1.1984	1.1947	1.1999
10	1.1940	1.1881	1.1805	1.1793	1.1829	1.1799

Table 5.5: Results for Bermudan geometric average put options in dimensions 3-10 on normal Sobol' grids, using the European price as a control variate.

d	5×10^4	10×10^4	15×10^4	20×10^4	25×10^4	30×10^4
3	1.5595	1.5596	1.5596	1.5594	1.5594	1.5594
4	1.4376	1.4378	1.4382	1.4378	1.4379	1.4382
5	1.3592	1.3602	1.3603	1.3595	1.3597	1.3599
6	1.3082	1.3082	1.3085	1.3041	1.3044	1.3048
7	1.2728	1.2752	1.2716	1.2658	1.2653	1.2649
8	1.2484	1.2487	1.2437	1.2364	1.2379	1.2465
9	1.2274	1.2242	1.2231	1.2169	1.2133	1.2189
10	1.2141	1.2082	1.1994	1.1988	1.2020	1.1994

Table 5.6: Results for American geometric average put options in dimensions 3-10 on normal Sobol' grids, using the European price as a control variate.

d	5×10^4	10×10^4	15×10^4	20×10^4	25×10^4	30×10^4
3	1.5372	1.5375	1.5376	1.5377	1.5377	1.5378
4	1.4141	1.4155	1.4160	1.4163	1.4165	1.4166
5	1.3309	1.3338	1.3360	1.3364	1.3370	1.3371
6	1.2695	1.2729	1.2751	1.2777	1.2779	1.2796
7	1.2139	1.2249	1.2255	1.2292	1.2319	1.2321
8	1.1628	1.1773	1.1850	1.1898	1.1899	1.1863
9	1.1234	1.1397	1.1428	1.1548	1.1514	1.1588
10	1.1177	1.1008	1.1131	1.1103	1.1170	1.1242

Table 5.7: Results for Bermudan geometric average put options in dimensions 3-10 using low distortion grids.

d	5×10^4	10×10^4	15×10^4	20×10^4	25×10^4	30×10^4
3	1.5583	1.5587	1.5589	1.5590	1.5590	1.5591
4	1.4341	1.4355	1.4361	1.4364	1.4367	1.4369
5	1.3500	1.3528	1.3550	1.3554	1.3561	1.3564
6	1.2875	1.2912	1.2935	1.2961	1.2965	1.2981
7	1.2319	1.2432	1.2433	1.2474	1.2496	1.2502
8	1.1813	1.1952	1.2032	1.2082	1.2080	1.2042
9	1.1412	1.1580	1.1615	1.1730	1.1689	1.1774
10	1.1390	1.1206	1.1315	1.1288	1.1365	1.1434

Table 5.8: Results for American geometric average put options in dimensions 3-10 on low distortion grids.

d	5×10^4	10×10^4	15×10^4	20×10^4	25×10^4	30×10^4
3	1.3460	1.3463	1.3464	1.3465	1.3465	1.3466
4	1.2287	1.2295	1.2299	1.2301	1.2304	1.2305
5	1.1501	1.1520	1.1535	1.1540	1.1544	1.1546
6	1.0904	1.0947	1.0965	1.0982	1.0987	1.0994
7	1.0394	1.0474	1.0497	1.0523	1.0545	1.0553
8	0.9877	1.0015	1.0078	1.0122	1.0137	1.0131
9	0.9405	0.9605	0.9654	0.9726	0.9729	0.9779
10	0.9080	0.9100	0.9247	0.9291	0.9322	0.9393

Table 5.9: Results for European geometric average put options in dimensions 3-10 using low distortion grids.

d	5×10^4	10×10^4	15×10^4	20×10^4	25×10^4	30×10^4
3	1.5379	1.5379	1.5379	1.5379	1.5380	1.5380
4	1.4172	1.4178	1.4179	1.4179	1.4179	1.4179
5	1.3393	1.3403	1.3410	1.3409	1.3410	1.3410
6	1.2867	1.2859	1.2863	1.2871	1.2868	1.2878
7	1.2448	1.2478	1.2461	1.2472	1.2477	1.2471
8	1.2167	1.2174	1.2187	1.2191	1.2178	1.2147
9	1.2017	1.1980	1.1964	1.2011	1.1974	1.1998
10	1.2101	1.1913	1.1888	1.1817	1.1853	1.1853

Table 5.10: Results for Bermudan geometric average put options in dimensions 3-10 using low distortion grids, using the European price as a control variate.

d	5×10^4	10×10^4	15×10^4	20×10^4	25×10^4	30×10^4
3	1.5590	1.5592	1.5592	1.5592	1.5593	1.5593
4	1.4372	1.4378	1.4380	1.4380	1.4381	1.4381
5	1.3584	1.3593	1.3601	1.3599	1.3602	1.3603
6	1.3048	1.3042	1.3047	1.3056	1.3054	1.3064
7	1.2628	1.2661	1.2639	1.2654	1.2654	1.2652
8	1.2352	1.2353	1.2369	1.2376	1.2359	1.2327
9	1.2196	1.2164	1.2150	1.2193	1.2149	1.2184
10	1.2315	1.2111	1.2072	1.2002	1.2048	1.2045

Table 5.11: Results for American geometric average put options in dimensions 3-10 on low distortion grids, using the European price as a control variate.

Option type	c_1	c_2	R^2
European	$-0.35(\pm 0.23)$	$-1.91(\pm 0.10)$	0.971
Bermudan	$-0.42(\pm 0.14)$	$-1.85(\pm 0.06)$	0.988
American	$-0.55(\pm 0.13)$	$-1.74(\pm 0.05)$	0.989

Table 5.12: Regression coefficients for the error behaviour on normal Sobol' grids (95% CI in parentheses).

Option type	c_1	c_2	R^2
European	$-0.49(\pm 0.12)$	$-1.94(\pm 0.05)$	0.992
Bermudan	$-0.59(\pm 0.08)$	$-1.84(\pm 0.03)$	0.997
American	$-0.83(\pm 0.08)$	$-1.65(\pm 0.03)$	0.995

Table 5.13: Regression coefficients for the error behaviour on low distortion grids (95% CI in parentheses).

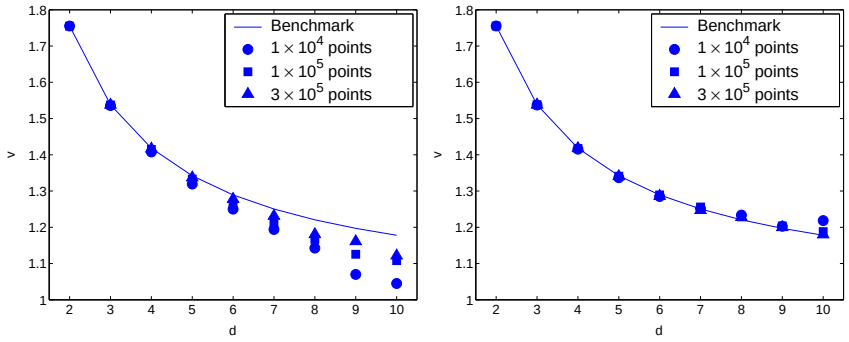


Figure 5.1: Bermudan pricing results for normal Sobol' grids presented raw (left) and using European price as control variate (right).

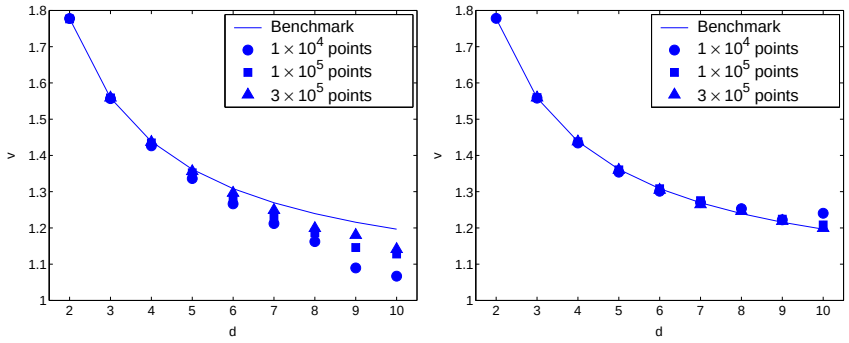


Figure 5.2: American pricing results for normal Sobol' grids presented raw (left) and using European price as control variate (right).

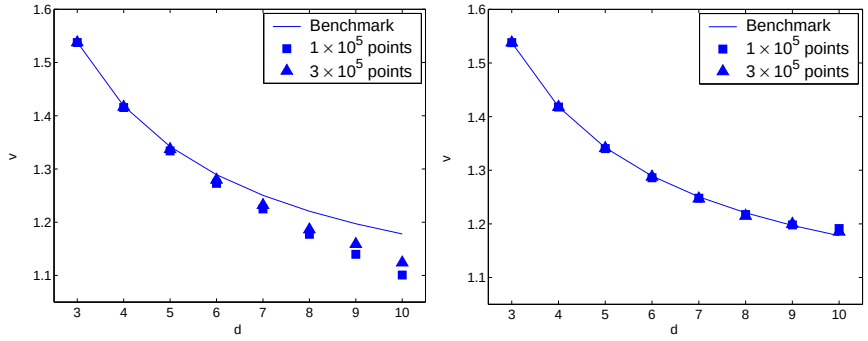


Figure 5.3: Bermudan pricing results for low distortion grids presented raw (left) and using European price as control variate (right).

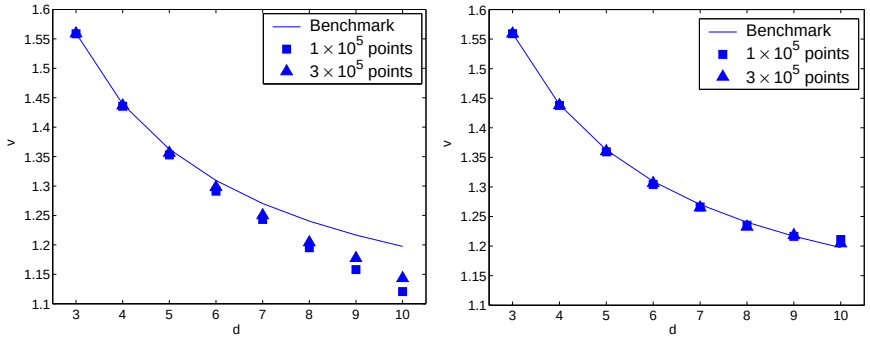
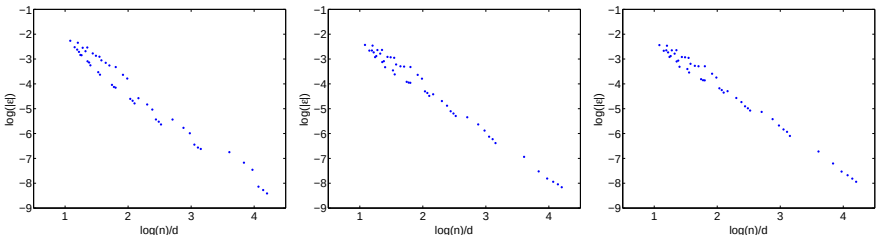


Figure 5.4: American pricing results for low distortion grids presented raw (left) and using European price as control variate (right).



(a) European (b) Bermudan (c) American

Figure 5.5: Log of absolute errors for European, Bermudan and American geometric average options plotted against $\log(n)/d$ for $d = 3, \ldots, 10$ for normal Sobol' grids. The points nearly lie in a straight line in all three cases, giving a clear indication of complexity. See Table 5.12 for regression results.

5.6 Timings

The irregular grid method presented in this paper can be divided into two computationally intensive stages: obtaining the generator matrix and performing the time stepping. The first is the most expensive, but once a matrix has been obtained it can be reused for a wide range of related problems. We do not consider computing transition matrices here; it suffices to say that the situation is very similar to generator matrices.

Here we provide indications of the timings involved; as usual this depends heavily on the hardware and software used. The software aspect is emphasised here since there is a huge difference in the performance of different algorithms for solving the linear programming problem and for solving linear systems of equations. The experiments are carried out in Matlab on a 866MHz Pentium III under Windows 2000.

5.6.1 Generator matrix

In dimension d we are interested in solving a large number of linear programming problems with $\eta_d = d(d+3)/2$ equality constraints and where all variables are nonnegative. The number of variables needed is not known a priori, but it has been found that $5\eta_d$ is sufficient for points close to the center of the grid, and an increased number of $20\eta_d$ is needed closer to the boundary. The strategy is thus to order the points according to their norm and try $5\eta_d$ neighbours until a certain failure rate is reached, then to switch to $20\eta_d$ neighbours on the remaining points.

In two dimensions a single problem takes about 0.06s and is not sensitive to the number of variables changing from $5\eta_d$ to $20\eta_d$. This is probably due to the relatively large overhead involved in the Matlab routines. In five dimensions we see an increase from 0.07s for $5\eta_d$ neighbours to 0.10s for $20\eta_d$. In ten dimensions we see a corresponding increase from 0.31s to 1.90s per problem. It is thus clear that parallelisation is desirable to keep the computation times reasonable, especially for higher dimensional problems.

5.6.2 Time stepping

In dimension d and with n grid points we use a generator matrix with n rows each with $\eta_d + 1$ nonzero entries. The complexity of implicit time stepping should thus be quadratic with dimension and linear with grid size.

For 300,000 points in five dimensions, explicit time steps take about 1.5s and implicit about 29s with CGS. For ten dimensions, explicit time steps take about 3.0s and implicit about 21s with CGS. The fact that implicit solutions can be faster

in a higher dimension is due to the conditioning of the matrix, making it more amenable to solution even though it is more dense.

One can thus perform about 10-20 times more explicit than implicit time steps for the same running time. However there is a tradeoff since the latter generally give much better precision.

5.7 Boundaries

We now compare the observed boundaries presented in Figures 5.7 and 5.8 to the naive predictions in Section 4.2 and Figure 4.2.

The proportion of boundary points goes up quickly with dimension, as predicted in Section 4.2. A simple calculation reveals that the proportion of boundary points for a regular grid with $n^{1/d}$ steps per dimension is $1 - (1 - 2n^{-1/d})^d$. For example, for $d = 10$ one requires a grid size of about 5×10^{14} to bring the proportion of boundary points down to 0.5. Using the irregular grid method one needs about 3×10^5, as seen in Figure 5.7.

We cannot compare our results directly to the predictions since we used a maximum of $20\eta_d$ neighbours when trying to satisfy local consistency. A direct comparison would require that we used all points in the grid. It is clear that the observed boundaries lie at a smaller radius r than the predicted ones. This may be partially due to the small number of neighbours considered, but may also be caused by the optimism inherent in the predictions, namely that only the minimum number of neighbours is required to satisfy the local consistency conditions.

We finally note that the boundaries are not monotone with grid size in Figures 5.7 and 5.8. In the case of normal Sobol' grids, this may be attributed to the fact that new points in the grid are infeasible with respect to the closest $20\eta_d$ neighbours.

6 Conclusions

We proposed a method for pricing options with several underlying assets and an arbitrary payoff structure. The method was tested for geometric average options, which can be easily benchmarked, in dimensions three to ten with very accurate results.

We saw a decay in precision for increasing dimension, a phenomenon which can be attributed to the increasing distance between points in the approximating Markov chain, and to the increasing size of the boundary region. An analysis of the error implies that the method has exponential complexity with dimension, but the use of control variates was shown to reduce the error substantially. The use of extrapolation is also expected to provide accurate approximations, although this

was not tested in the present work.

The computation of transition and generator matrices is expensive; however once generated these matrices can be reused for a large class of similar problems with time dependent parameters. Furthermore computations are cheap once the matrix is obtained.

Interestingly we found little difference in complexity between the cases where Sobol' and low distortion grids were employed. The complexity observed was exponential in dimension, of approximately the same order as regular grid discretisations.

Although the method extends naturally in principle to arbitrary Markov processes with parameters depending on state and time, further extensions to the numerical procedures are required to make the proposed method computationally attractive in such cases. For example, this is of interest when considering Bermudan swaptions where the drift is state dependent.

References

[1] Jérôme Barraquand and Didier Martineau. Numerical valuation of high dimensional multivariate American securities. *Journal of Financial and Quantitative Analysis*, 30:383–405, 1995.

[2] S.J. Berridge and J.M. Schumacher. An irregular grid approach for pricing high-dimensional American options. *CentER Discussion Paper*, 99, 2002.

[3] S.J. Berridge and J.M. Schumacher. An irregular grid method for solving high-dimensional problems in finance. In *International Conference on Computational Science (2)*, pages 510–519, 2002.

[4] S.J. Berridge and J.M. Schumacher. An irregular grid method for high-dimensional free-boundary problems in finance. Forthcoming, *Future Generation Computer Systems: Special Issue on Numerical Methods*, 2003.

[5] Phelim P. Boyle, Adam W. Kolkiewicz, and Ken Seng Tan. An improved simulation method for pricing high-dimensional American derivatives. *Mathematics and Computers in Simulation*, 62:315–322, 2003.

[6] Mark Broadie and Paul Glasserman. A stochastic mesh method for pricing high-dimensional American options. *Working paper*, 1997.

[7] Colin W. Cryer. The solution of a quadratic programming problem using systematic overrelaxation. *SIAM Journal of Control*, 9:385–392, 1971.

[8] M.A.H. Dempster and J.P. Hutton. Pricing American stock options by linear programming. *Mathematical Finance*, 9:229–254, 1999.

[9] Vladimir Druskin and Leonid Knizhnerman. Extended Krylov subspaces: approximation of the matrix square root and related functions. *SIAM Journal of Matrix Analysis and Applications*, 19:755–771, 1998.

[10] M. Evans and T. Swartz. *Approximating Integrals via Monte Carlo and Deterministic Methods*. Oxford University Press, Oxford, 2000.

[11] R. Glowinski, J.L. Lions, and R. Trémolières. *Numerical Analysis of Variational Inequalities*. North-Holland, Amsterdam, 1981.

[12] Martin B. Haugh and Leonid Kogan. Pricing American options: a duality approach. *Working Paper*, 2001.

[13] Marlis Hochbruck and Christian Lubich. On Krylov subspace approximations to the matrix exponential operator. *SIAM Journal of Numerical Analysis*, 34:1911–1925, 1997.

[14] Willem Hundsdorfer and Jan Verwer. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer-Verlag, Berlin, 2003.

[15] Patrick Jaillet, Damien Lamberton, and Bernard Lapeyre. Variational inequalities and the pricing of American options. *Acta Applicandae Mathematicae*, 21:263–289, 1990.

[16] Harold J. Kushner and Paul G. Dupuis. *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer-Verlag, New York, 1992.

[17] Francis A. Longstaff and Eduardo S. Schwartz. Valuing American options by simulation: a simple least squares approach. *Review of Financial Studies*, 14:113–147, 2001.

[18] Harald Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM, Philadelphia, 1992.

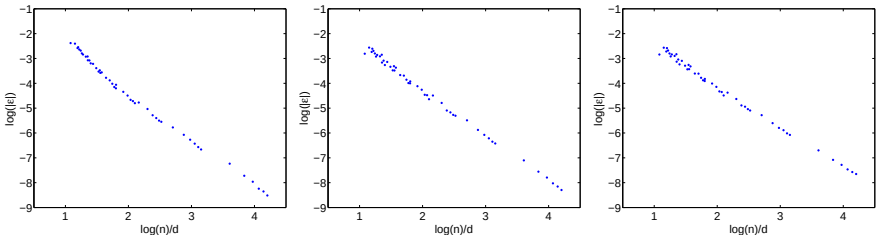
[19] Gilles Pagès. A space quantization method for numerical integration. *Journal of Computational and Applied Mathematics*, 89:1–38, 1997.

[20] L.C.G. Rogers. Monte Carlo valuation of American options. *Mathematical Finance*, 12:271–286, 2002.

[21] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Chichester, 1998.

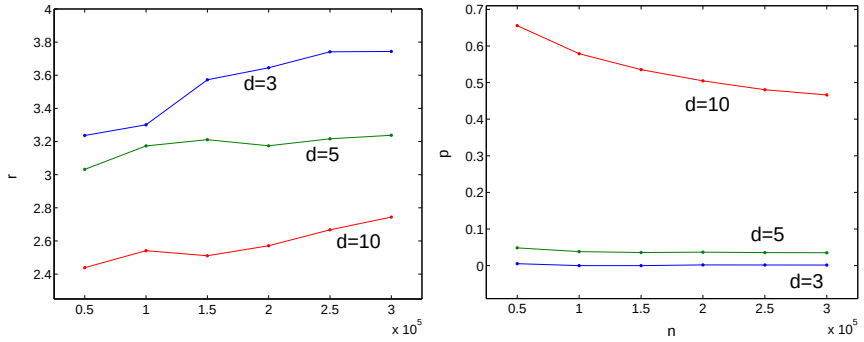
[22] James A. Tilley. Valuing American options in a path simulation model. *Transactions of the Society of Actuaries*, 45:499–520, 1993.

[23] John N. Tsitsiklis and Benjamin Van Roy. Regression methods for pricing complex American-style options. *IEEE Transactions on Neural Networks*, 12:694–703, 2000.



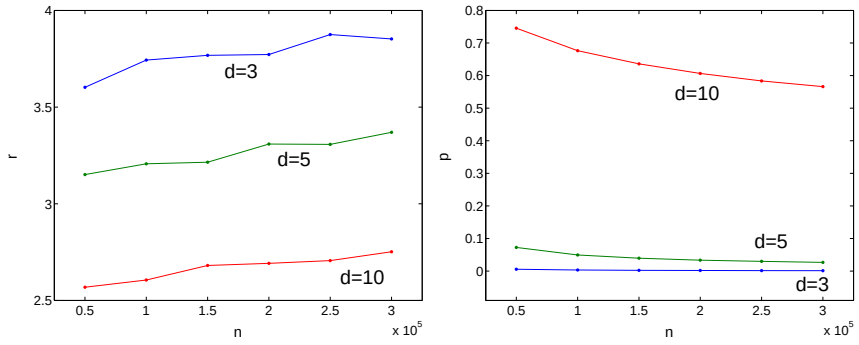
(a) European (b) Bermudan (c) American

Figure 5.6: Log of absolute errors for European, Bermudan and American geometric average options plotted against $\log(n)/d$ for $d = 3, \ldots, 10$ for low distortion grids. The points nearly lie in a straight line in all three cases, giving a clear indication of complexity. See Table 5.13 for regression results.



(a) Radius (b) Proportion

Figure 5.7: Smallest norms in normal Sobol' grids for which local consistency could not be satisfied and proportion of points in the boundary region with $20\eta_d$ nearest neighbours. Compare Figure 4.2.



(a) Radius (b) Proportion

Figure 5.8: Norms of the smallest points in low distortion grids for which local consistency could not be satisfied and proportion of points in the boundary region with $20\eta_d$ nearest neighbours. Compare Figure 4.2.