Valuing American options using fast recursive projections

COSMA, Antonio, GALLUCCIO, Stefano, SCAILLET, Olivier

Abstract
This paper introduces a new numerical option pricing method by fast recursive projections. The projection step consists in representing the payoff and the state price density with a fast discrete transform based on a simple grid sampling. The recursive step consists in transmitting coefficients of the representation from one date to the previous one by an explicit recursion formula. We characterize the convergence rate of the computed option price. Numerical illustrations with different American and Bermudan payoffs with discrete dividend paying stocks in the Black-Scholes and Heston models show that the method is fast, accurate, and general.

Reference

DOI : 10.2139/ssrn.2091236

Available at:
http://archive-ouverte.unige.ch/unige:41856

Disclaimer: layout of this document may differ from the published version.
Valuing American options using fast recursive projections

Antonio Cosma*, Stefano Galluccio**, and Olivier Scaillet***

*Université du Luxembourg

**BNPParibas

***Université de Genève and Swiss Finance Institute

June 2012

Abstract

This paper introduces a new numerical option pricing method by fast recursive projections. The projection step consists in representing the payoff and the state price density with a fast discrete transform based on a simple grid sampling. The recursive step consists in transmitting coefficients of the representation from one date to the previous one by an explicit recursion formula. We characterize the convergence rate of the computed option price. Numerical illustrations with different American and Bermudan payoffs with discrete dividend paying stocks in the Black-Scholes and Heston models show that the method is fast, accurate, and general.

Keywords: Option pricing, American option, Bermudan option, discrete transform, discrete dividend paying stock, numerical techniques.

JEL classification: G13, C63.
1 Introduction

The pricing of many among the most traded path dependent derivatives is a well understood theoretical exercise. Amidst these instruments, American options on dividend-paying stocks are arguably one of the most studied and of utmost interest for practitioners. Almost any departure from the plain vanilla European style options implies that closed form pricing formulas are no more available (see Detemple (2005) for an extensive review). A large array of numerical methods, including lattice (see for instance Cox, Ross and Rubinstein (1979)), finite differences (Brennan and Schwartz (1977), Clarke and Parrott (1999), Ikonen and Toivanen (2007)), Monte Carlo methods (Broadie and Glasserman (1997), Longstaff and Schwartz (2001), Rogers (2002), Haugh and Kogan (2004)), and solving of the associated integral equations (Kim (1990), Huang, Subrahmanyam and Yu (1996), Sullivan (2000), Detemple and Tian, (2002)) provides theoretically sound solutions to these pricing problems. However they are often cumbersome to implement in practice. Non-recombining trees for instance, the tool (Hull (2008)) of choice for pricing American options on dividend paying stocks, quickly become computationally unmanageable as soon as the number of dividends paid out in the time horizon of the option grows. As alternatives to these numerical methods, we may consider semi-analytical approximations as for instance Barone-Adesi and Whaley (1987), Broadie and Detemple (1996), Bunch and Johnson (2000), Ju (1998). Ibáñez (2003) provides a numerical algorithm to approximate the early exercise premium. Although fast and accurate, these methods cannot easily be extended beyond the Black-Scholes model. Roll (1977), Geske (1979), and Whaley (1981) provide a close form approximation to pricing an American option paying a single discrete dividend in the Black-Sholes framework. Medvedev and Scaillet (2010) developed an approximation to American option prices under stochastic volatility and stochastic interest rates using short-term asymptotics (see Lamberton and Villeneuve (2003) for the Black Scholes case). These approximations are accurate and flexible, but cannot accommodate discrete dividends. They are mainly
suited for index and exchange rate options.

Our contribution is a numerical method that is entirely novel and at a time intuitive, fast, accurate and widely applicable. By widely applicable, we mean that we can value a large number of path dependent contracts, American and Bermudan options being the most notable examples, for modeling choices beyond the Black-Scholes model. Our approach can handle an almost unrestricted variety of payoffs functions, and allows for a broad choice in the distributional assumption for the process of the underlying asset including stochastic volatility and jump processes.

We can view pricing of a derivative security essentially from two perspectives, the link between the two being given by the Feymann-Kac theorem. The first one is to solve the partial differential equation (hereafter PDE) yielding the price of the derivative assets. Numerically discretizing the differential operator leads to finite difference schemes. This is the most common approach when it comes to numerically finding solutions to complex pricing problems. The second perspective is to see the price of the derivative asset as the conditional expectation of the discounted future payoff. It exploits the knowledge of the probability distribution with respect to which the conditional expectation is taken. This viewpoint provides the basis for Monte Carlo simulation based integration methods. The binomial tree technology belongs to this second family of methods (although similar to finite differences in its implementation) as it provides a discrete approximation to conditional expectations.

This paper places itself in the second perspective. Contrary to simulation based methods, we do not use the probability function to generate trajectories. We directly approximate the state price density on a suitable set of functions within a discretization that makes computing the conditional expectation easy and fast. This permits the representation of the conditional expectation operator in the simplest form of standard linear operators, i.e. matrices, and the pricing of derivative contracts by means of linear algebra tools. We derive the projection of the operators and value functions on the space of finite dimensional discrete matrices by direct
sampling of the original functions. In this framework, function sampling corresponds to valuing a function on a given grid, and the computed price corresponds to successive matrix-times-vector multiplications. Hence, the effectiveness of the method is explained by two key features: a) we get matrices by a straightforward sampling, b) we get the final option price by a few matrix products. For instance, only two matrix-per-vector multiplications and a couple of seconds are needed to evaluate an American call option and its Greeks on a stock paying one dividend before the expiry date. The competing (non recombining-) tree method needs thousands of steps and thousands of seconds to yield the same accuracy.

To be more explicit, we design the numerical scheme so that we can fully take advantage of simple grid sampling representations of payoffs at exercise dates and transition probabilities (the so-called Green functions), and of the recursive nature of the transition probabilities to move backward between exercise dates. As recalled above, option valuation in arbitrage free economies amounts to using linear operators that assign prices today to payoffs on futures dates. In multiperiod economies, the temporal consistency in valuation insures that the family of such operators satisfies a semigroup property (Garman (1984)). Most valuation models correspond to Markov environments for the price of the underlying asset. Then the semigroup property is a recursive value property that connects pricing contingent claims written on the markov state indexed by the time horizons of exercise. The family of operators are conditional expectation operators which are related by the law of iterated expectations restricted to markov processes (Hansen and Scheinkman (2009)). From a computational point of view this means that pricing options requires to know (i) how payoffs depend on the price of the underlying assets at future exercise dates and (ii) how the price of the underlying assets transits from one state to another according to the elapsed time between exercise dates. Ideally, we would like to summarize and combine these two pieces of information in the most efficient way, and this is what our approach aims at.

Our method involves two steps at each pair of exercise dates: a projection step and a recursive
step. The payoff and state price densities are represented (projection step) in terms of suitable localized basis functions before being reprojected (recursive step) on the information available at the previous date. Simple and tractable projections of complex payoffs can already be found in the literature via representation on a set of basis functions (such as polynomials in Madan and Milne (1994), Lacoste (1996), Chiarella, El-Hassan, and Kucera (1999), Darolles and Laurent (1999)). Chiarella et al. (1999) have suggested a fast recursion projection method based on Hermite polynomials in the Black-Scholes model (see Chiarella and Ziogas (2005) for extension to the model of Merton (1976)). In a more recent application of Hermite polynomials to pricing, Xiu (2011) provides a general, closed-form approximation of European option prices by using a finite-term expansion of the transition density (see also Ait-Sahalia and Kimmel (2007, 2010) for related work on expansions). Kristensen and Mele (2010) approximate the option price by expanding the difference between the true model price and the Black-Scholes price.

Chiarella et al. (1999) also explain how their method provides a viable numerical method for the implementation of the path integral approach to option pricing as described in Linetsky (1998) for example. Our paper is an extension of the former in that we use sampling instead of projections to improve on speed and accuracy when facing strong discontinuities in the payoff function (as in the presence of a digital payoff). In the way that we approximate the integral operator, our method shares some features with the quadrature approach by Androcopoulos et al. (2003). We rely on the rectangle method, also called the midpoint or mid-ordinate rule, instead of a quadrature method. This allows us to sample the value functions on exactly the same grid at any point of time of interest. This is a mean to fully exploit the semigroup property in the discretization. We have no need to recompute the transition matrices throughout the entire recursion if the points in time are chosen equally spaced. In implementations, this is arguably a key advantage in terms of simplicity, practical applicability and speed, and would be unfeasible under time varying grids.

We extend Chiarella et al. (1999) and Androcopoulos et al. (2003) by being able to deal
with pricing based on the Fourier Transform, as in the stochastic volatility model of Heston (1993) for instance. Fourier transforms of transition probabilities often describe price evolution in affine models (Duffie, Pan and Singleton (2000)), quadratic models (Leippold and Wu (2002), Cheng and Scaillet (2007)), and variance gamma and Levy models (Madan, Carr and Chang (1998), Carr, Geman, Madan, Yor (2003)). The ability to handle pricing available by transform analysis, as in standard affine models for example, is a fundamental contribution of our work.

To summarize, our method is of general application in the pricing of path-dependent options where the path is monitored at discrete moments in time (see Androcopoulos et al. (2003, 2007) for several examples of such options), within a broad class of jump-diffusion models and Levy models, and for underlying assets paying discrete dividends.

As leading example, we check that our method is able to price quickly and accurately American options on a discrete dividend paying stock within the Black-Scholes model and Heston model. Our method is thus particularly suited for empirical work on individual equity options. NYSE Amex lists such options on hundreds of US companies, while NYSE Liffe lists such options on over 250 leading European companies via the Amsterdam, Brussels, London and Paris central order books. These available options prices on dividend paying stocks allow to analyze the optimal behavior of option exercise, in line with the empirical works of Poll, Stoll and Whaley (2008) and Barraclough and Whaley (2012). Ibáñez and Paraskevopoulos (2010) provide an estimate of the cost of suboptimal exercise of American options through a second order expansion of the derivative price with respect to the underlying security, the gamma of the option. Our method can deliver quick and precise estimates of the gamma of the option even in the case dividend paying stocks and stochastic volatility. As another example, we also check that our method is successful when applied to Bermudan options and Bermudan digital options, which are written over-the-counter by the financial industry.

The paper is organized as follows. In Section 2, we develop an introductory example based on the Black-Scholes model, and present some preliminary numerical results showing the advantages
of our technique. In Section 3, we study the general case of valuation by fast recursive projections in order to include standard affine models. We present numerical illustrations for the Heston model. We also provide the theoretical convergence of the computed option price in terms of the sampling frequency. Section 4 gathers some concluding remarks.

2 An introductory example

In this section, we show how a pricing problem for a European payoff in the Black-Scholes model translates into a functional projection. The derivation uses elementary concepts of calculus. Then, we explain how we exploit the projection to build fast recursive schemes to value path-dependent options such as Bermudan and American options when the stock pays discrete dividends. We design this introductory example to emphasize the intuition underpinning our approach. In the next section, we provide a general algorithm to solve pricing by transform analysis under a wider class of models, and characterize the convergence rate of the computed option price.

2.1 Description of the method: the European case

Let \( S_t \) be the price of the underlying asset at date \( t \) and assume that interest rates are constant to ease exposition. For \( S_t = x \), the value function \( V(x, t) \) for a European option is given by the following conditional expectation:

\[
(2.1) \quad V(x, t) = \mathbb{E}[e^{-r(T-t)}H(S_T, T)|S_t = x],
\]

where \( H(S_T, T) \) is the payoff function expressed as a function of time \( T \) and of the value of the underlying asset \( S_T \) at maturity date \( T \), and \( r \) is the constant risk-free interest rate. When the pricing operator in (2.1) admits a state price density \( G(x, t; y, T) \), the so-called Green function, which is the discounted value of the transition probability density from point \( x \) at time \( t \) to
point \( y \) at time \( T \), we obtain the familiar integral form:

\[
V(x, t) = \int H(y, T)G(x, t; y, T)dy.
\]

(2.2)

Now consider a regularly spaced grid of points \{\( y_1, y_2, \ldots, y_N \)\} that defines a finite interval \( D = [y_1 - \Delta y/2, y_N + \Delta y/2] \), with \( \Delta y = y_j - y_{j-1} \). We know that under appropriate regularity conditions, the integral (2.2) restricted to the interval \( D \) can be approximated by the Riemann sum:

\[
V(x, t) \sim \sum_{j=1}^{N-1} H(y_j, T)G(x, t; y_j, T)\Delta y, \quad t < T,
\]

(2.3)

where the \( \sim \) symbol means that the right hand term converges to the left hand term as \( \Delta y \to 0 \). The representation (2.3) is known as the ‘rectangle method’ in standard integral calculus.

If we are interested in computing the value of \( V(x, t) \) on a regularly spaced grid of points \{\( x_1, \ldots, x_M \)\}\(^1\), we can express (2.3) in a matrix form

\[
V(t) \sim v(t) = G(t; T)H(T),
\]

(2.4)

where \( V(t) = (V(x_1, t), \ldots, V(x_M, t))' \), \( v(t) \) is the approximation of \( V(t) \) obtained from the \( N \times 1 \) vector \( H(T) \) with entries \( H_j = H(y_j, T) \) and the \( M \times N \) matrix \( G \) with entries \( G_{ij} = \Delta y G(x_i, t; y_j, T) \). Equation (2.4) can be interpreted as a discretization of the economy.

The matrix \( G \) can be thought of as a matrix of Arrow-Debreu prices, where the rows represent discrete states \{\( x_i \}_{i=1, \ldots, M} \) at the current date \( t \), and the columns represent discrete states \{\( y_j \}_{j=1, \ldots, N} \) at the future date \( T \). The column vectors \( V(t) \) and \( H(T) \) may be thought of as vectors of state contingent values at dates \( t \) and \( T \). The matrix operator \( G \) discounts future payoffs at date \( T \) to current prices at date \( t \) (Garman (1984)).

\(^1\)For instance to plot the value function of the contract, or to compute the Greeks.
Let us look at the elements of $H(T)$ more closely. For readability, we define $y_i = y_i - \frac{\Delta y}{2}$ and $\bar{y}_i = y_i + \frac{\Delta y}{2}$. We can interpret the value $H(y_j, T)$ as an approximation of the quantity

\begin{equation}
(1/\Delta y) \int_{y_j}^{\bar{y}_j} H(y, T) dy = \frac{1}{\sqrt{\Delta y}} \int \mathbb{1}_{y_j, \bar{y}_j} H(y, T) dy \approx \frac{1}{\sqrt{\Delta y}} \int e_j(y) H(y, T) dy,
\end{equation}

where $e_j(y) = \frac{1}{\sqrt{\Delta y}} \mathbb{1}_{y_j, \bar{y}_j}$, and where $\mathbb{1}_{y_j, \bar{y}_j}$ is the indicator function of the interval $[y_j, \bar{y}_j)$. Since $\{e_j(y)\}_{j=1,...,N}$ is an orthonormal set given the standard $L_2$ inner product $\langle f, g \rangle = \int f(x)g(x) dx$, we can view the entries of the vector $H(T)$ as an approximation of $(1/\sqrt{\Delta y})$ times the coefficients of the decomposition of the payoff function $H(y, T)$ on the set of orthonormal indicator functions defined by the grid $\{y_1, \ldots, y_N\}$. A similar argument can be applied to the coefficients of the $G(t; T)$ matrix: every row of $G(t; T)$ is given by an approximation of $\sqrt{\Delta y}$ times the coefficients of the projection of the conditional density $G(x = x_j, t; y, T)$ on the same orthonormal set. The different choice in the normalization factors for the entries of $G$ and $H$ is justified by our wish to interpret all quantities appearing in (2.4) as prices. We have that $V(x, t)$ and $H(y, T)$ are value functions, while $G(x, t; y, T)$ is a state price density.

All in all, we can interpret the numerical approximation of the integral (2.2) as a projection of the functions on an orthonormal basis. Such an interpretation is key in the generalization of the recursive projection approach to more sophisticated models in the next section. Besides, we recover the well-known result that the integral of the product between two functions has the same value as the dot product of the vectors containing the coefficients with respect to the given basis. In our case, the functional projection boils down to sampling the given functions on a grid of $N$ points $\{y_j\}_{j=1,...,N}$ via a discrete transform. From a computational point of view, the entries of $H(T)$ summarize how payoffs depend on the price of the underlying assets at future exercise dates, and the entries of $G(t; T)$ summarize how the price of the underlying assets transits from one state to another according to the elapsed time between exercise dates. Figure 1 gives a graphical representation of the two computational steps, the projection step...
and of the recursive step, of our fast recursive projection approach. The value function at \( t \), in black on the left, and the state price density for a given value of \( S_s \), in red on the right, are sampled (projection step), the obtained arrays of values are multiplied element by element, and the products are summed to obtain the value of \( V(S_s, s) \) (the recursive step).

[Figure 1 about here]

2.2 Description of the method: the path-dependent case

Let us now address the valuation of path-dependent contracts. We start by considering a Bermudan option. The option can be exercised at a set of dates \( \{t_1 = t, \ldots, t_L = T\} \). At each \( t_l \), the holder of a Bermudan option may decide whether to exercise or not. He exercises if the intrinsic value \( H(S_{t_l}, t_l) = (S_{t_l} - K)_+ \equiv \max(S_{t_l} - K, 0) \) is higher than the value of keeping the option, i.e., the continuation value. Bermudan options are the ideal building block to study American call options on dividend-paying stocks. It is well known that it can be optimal to exercise American call options just before ex-dividend dates \( \{s_h, t < s_h < T\}_{h=1, \ldots, H} \), see for instance Poll, Stoll and Whaley (2008) for a discussion on early exercise strategies. This means that we have to monitor the option value function \( V(x, t) \) just before the ex-dividend dates, when the intrinsic value \( (S_{s_h} - \epsilon - K)_+ \) for a small \( \epsilon > 0 \) can be larger than the continuation value \( V(S_{s_h}, s_h) \). Then an American call option shares with a Bermudan option the feature that its value function has to be evaluated only at a finite number of dates.

The semigroup property of the pricing operator ensures that the value function \( V(x, t) \) of a Bermudan option can be computed recursively. The recursion consists in moving backwards in time and computing at each \( t_l, l = 1, \ldots, L - 1 \):

\[
V(x, t_l) = \max \{ H(x, t_l), \mathbb{E}[e^{-r(t_{l+1}-t_l)}V(S_{t_{l+1}}, t_{l+1})|S_{t_l} = x] \},
\]

with the boundary condition \( V(y, t_L) = H(y, t_L) \). To speed up the recursion, we impose that
the grid of values $\{y_j\}_{j=1,\ldots,N}$ at which the function $V(y, t_{l+1})$ is sampled and $\{x_i\}_{i=1,\ldots,M}$, the grid at which the function $V(x, t_l)$ is computed, coincide at each exercise date. Then, in the matrix notation of the approximation, we obtain:

$$(2.7) \quad V(t_l) \sim v(t_l) = \max\{H(t_l), G(t_l; t_{l+1})v(t_{l+1})\},$$

and the approximation $v(t_l)$ will be the input for the next step to compute an approximation $v(t_{l-1})$ of $V(t_{l-1})$. The convergence properties of $v(t_l)$ to $V(t_l)$ are formally established in Proposition 1 of Section 3. From (2.7), it is clear that computations only take place at the exercise dates defined in the Bermudan contract, and do not require any input at any other point in time. Furthermore, under the previous additional constraints on the grids, if the time interval $\tau = t_{l+1} - t_l, l = 1, \ldots, L - 1$, is constant and if the pricing operator enjoys a stationarity property (time translation invariance), the matrix $G(t_l; t_{l+1}) = G(\tau)$ has constant entries, and the algorithm only involves a single computation of the matrix.

The methodology extends easily in the presence of discrete dividends paid on potential exercise dates\(^2\). This means that the set of ex-dividend dates is a subset of the Bermudan exercise dates, and we have $\{s_h\}_{h=1,\ldots,H} \subset \{t_l\}_{l=1,\ldots,L}$. We only need to add the dividend to the continuation value in (2.6). Hence, in order to price an American option on a dividend paying stock, Equation (2.7) has to be modified by sampling the state price density $G(x, t_l; y, t_{l+1})$ at the grid $\{y_i - \delta\}_{i=1,\ldots,N}$ for the conditioning value $x$, whenever $t_l \in \{s_h\}_{h=1,\ldots,H}$. The entries of the matrix $G(t_l; t_{l+1})$ then become $G_{ij} = G(y_i - \delta, t_l; y_j, t_{l+1})\Delta y$. Given the freedom in choosing where to sample $G$, $\delta$ could be any function $\delta(x)$ of $x$. If $\delta(x) = r_d x$, we can accommodate for a proportional dividend. If $\delta(x) = d$, we can accommodate for a discrete dividend amount $d$. If $\delta(x) = 0$, we are back to the Bermudan option case. The value function $V(s_h)$ still gives the value of the contract at the grid points $\{y_1, \ldots, y_N\}$, so its approximation $v(s_h)$ can be used as

\(^2\)The method can be easily extended to the case when the ex-dividend date does not belong to the set of exercise dates. We do not explicit this case since we are mainly interested to Bermudan options as a building block to study American options.
the input for the following step of the algorithm, and the recursive property of the algorithm is
maintained. Figure 2 shows how the recursive scheme changes to accommodate for dividends.

The early exercise decision for American long put holders is more complicated. We consider
a Bermudan put with exercise dates \( \{t_1, \ldots, t_L\} \), and assume that the dates \( \{s_1, \ldots, s_H\} \) at
which dividends are paid form a subset of the exercise dates. Then, by taking \( L \) large, our
approach also provides a quick approximation to pricing American put options on single stocks
paying discrete dividends. Let us have a closer look at the Black-Scholes model. Since the state
price density is known in closed form:

\[
G(x, t; y, T) = \frac{1}{y \sqrt{2\pi \sigma^2(T-t)}} \exp\left(-\frac{\left(\ln y - \ln x - \frac{r - \sigma^2}{2}(T-t)\right)^2}{2\sigma^2(T-t)}\right),
\]

where \( \sigma \) is the volatility, and since the pricing operator satisfies the stationarity property, we
expect to get a fast, simple and accurate numerical algorithm. Before reporting the numerical
results, we stress that the setup of this introductory example has been kept as simple as possible
to emphasize intuition by limiting technical details. Although computational results will already
speak in favor of our construction, it will be in the next sections that the recursive projection
method will deploy its full potentiality with more complex dynamics.

### 2.3 Numerical illustrations in the Black-Scholes model

As a first numerical example in the Black-Scholes framework, we compare the convergence speed
of a binomial tree and of the recursive projection method in pricing an American call option on a
dividend paying stock. Two popular ways are a known cash amount \( d \) or a known dividend yield
\( r_d \). The latter is computationally friendly since it leads to a recombining tree. Empirical evidence
however shows that corporations tend to commit to paying out fixed amounts at regular dates
and to smooth their dividends rather than adjusting them downwards and signal a decrease in
cash flows\(^3\). These arguments justify our preference to model dividends as fixed known amounts

\(^3\)For a signaling based theory on dividend policy see, for instance, Miller and Rock (1985).
rather than given yields. The known dividend amount assumption does not lead to a recombining tree, and a new tree is originated at each node following an ex-dividend date. To give an example of the numerical complexity, consider an American option with maturity $T$ and ex-dividend date $\tau = T/2$. If $\nu$ is the number of time discretization steps, a recombining tree displays $\nu(\nu + 1)/2$ nodes, that is the number of nodes grows as $\nu^2$. In the case a dividend is paid at $\tau$, and assuming for simplicity that $\nu$ is even, a new tree with $\nu/2$ steps originates from each of the $\nu/2 + 1$ nodes at $\tau$, so that the resulting tree has $\nu/4(\nu/2 + 1) + (\nu/2 + 1)\nu/4(\nu/2 + 1) = \nu/4(\nu/2 + 1)(\nu/2 + 2)$, the number of nodes scales as the third power of $\nu$. In general, the number of nodes grows as $\nu^{H+2}$, where $H$ is the number of ex-dividend dates. On the contrary, the number of operations in the recursive projection method grows linearly with the number $H$ of dividends paid, since every dividend date adds only a single extra step in the recursive algorithm.

Table 1 compares the convergence speed of a binomial tree and of the recursive projection method in pricing an American call option on a discrete dividend paying stock. The option has a maturity of $T = 3$ years and a dividend $d = 2$ is paid out at the end of each year. Other parameters, namely interest rate, volatility and strike price, are set equal to $r = 0.05$, $\sigma = 0.2$, and $K = 100$. We compute 3 prices: at-the-money, in-the-money and out-of-the-money, corresponding to $S_t = 80, 100, 120$, respectively. The true value is obtained with 10,000 time steps in the binomial tree. The positive integer $J$ gives the resolution level of the sampling: the payoff function is sampled at $2^J = N = M$ points. The step $\Delta y$ of the grid at which the functions are sampled is related to $J$ through the equality $\Delta y = 2^{-J}a$, where $a$ is a positive constant giving the size of the support of the indicator functions when $J = 0$. Describing the convergence of the recursive projections in terms of the parameter $J$ emphasizes how the approximation error decreases each time the number of the grid points is doubled. In Section 3, we show that our approximation has a convergence rate of order $O((\Delta y)^2)$, i.e., the error between the true price and the computed price decreases at a rate $C(\Delta y)^2$, $C$ being a positive constant.
The total computation times for the three levels of $S_t$ are in seconds. Comparison of the speed for $J = 7$ and $\nu = 2000$ shows that the recursive projection enjoys a much faster speed with a factor $10^5$ for a comparable level of precision. The explanation of the difference in speed is twofold: 1) the numerical complexity outlined above of a non-recombining tree, and 2) the need for a new tree for each value of $S_t$. Instead, the recursive projection method delivers the entire value function $V(t)$ at once in a straightforward manner. This feature is particularly useful to compute Greeks through numerical differentiation.

As a side remark, if for $S_t = 100$, we approximate the known constant dividend $d = 2$ with a known continuous dividend yield $r_d = 0.013$, a binomial tree with 10000 steps delivers a value of 18.213 instead of 18.527, with a relative error of about 169bp. This error is way above observed bid-ask spreads. This simple example points to the importance of using in empirical analysis models that can explicitly deal with discrete dividends, instead of using approximations based on continuous dividend yields. To gain more insight on the error introduced by using a continuous dividend yield approximation, we remind that under this approximation only deep in-the-money options are exercised before maturity. This observation justifies the use of the European Black-Scholes formula as a crude and quick proxy for American options on stocks paying a continuous dividend. Then the price obtained is biased downwards, since the non-exercise policy of the European proxy yields a suboptimal exercise policy. In addition, the use of a European proxy to get the implied volatility also yields an overestimation of the volatility. As we have argued, the advantage of recursive projections over binomial trees is that the former method can deal with several dividend payment dates, while the latter can be reasonably used for, at most, two dividend payment dates.

Besides, we have chosen a sampling scheme equivalent to projecting the payoff function on a set of basis functions that are well localized. The localization is in the sense that their

---

4The yield is obtained by considering the dividends paid at $t = 1$ and $t = 2$ only, since the one paid at $t = 3$ has no impact on the price of the option. Considering a dividend yield of 2% would have given an option value of 16.857, a much larger error.
support is a closed interval. This means that local features of the payoff function, such as for
instance a discontinuity, are described by the coefficients relative to one or at most two basis
functions, those lying next to the discontinuity. This avoids a noisy approximation induced
by the unnatural artifacts like spurious oscillations when projecting discontinuities on basis
functions defined on the entire domain such as the Fourier sine-cosine basis or the Hermite
polynomial basis. From a computational point of view, this property translates into an accurate
approximation even for payoffs with strong discontinuities, such as for instance a digital payoff
\( H(S_{t_1}, t_1) = \mathbb{I}_{S_{t_1} < K} \) in a Bermudan digital put option. Table 2 (see the caption of the table
for the values of the parameters of the example) shows that the binomial tree has problems in
capturing the discontinuity in the payoff function. This yields an extremely slow convergence
of the tree method for at-the-money Bermudan digital put options. The recursive projection
method reaches convergence in less than 20 seconds, while the binomial tree still leaves a large
50bp error for at-the-money options after several minutes.

3 Valuation by fast recursive projections

In this section, we generalize the approach developed in the introductory example of the previous
section. We show how we can represent the pricing problem of options as a projection back in
time of a value function. This valuation by fast recursive projections is based on two steps: a
projection step and a recursive step. The first step consists in representing a payoff and a state
price density as linear combinations of suitable indicator functions. The second step consists in
reprojecting the coefficients of the representation on the information available at the previous
date.

In the following sections, we characterize the projection step and the recursive step in two
different frameworks. In Sections 3.1 and 3.2, we consider the case in which the state price density
is known in closed form, as in the Black-Scholes example developed in Section 2. In Sections
3.3 and 3.4, we characterize the projection step and recursion step when the characteristic
function of the state price density has a known analytic form, as in the Heston model. In our presentation, we generically refer to this second class of models as to the “stochastic volatility” case. The methodology developed in Sections 3.3 and 3.4 covers affine jump-diffusion models and Levy models, where we price by transform analysis. The key requirement here is to be able to compute numerically the characteristic function of the state price density.

3.1 Black-Scholes: the projection step

Following Section 2.1, the projection step is based on an approximation of the payoff function and of the state price density by the set of orthonormal functions \( \{ e_j(y) \}_{j \in \mathbb{Z}} \):

\[
\tilde{H}(y, T) = \sqrt{\Delta y} \sum_{j=-\infty}^{\infty} H(y_j, T) e_j(y),
\]

\[
\tilde{G}(x_i, t; y, T) = \sqrt{\Delta y} \sum_{j=-\infty}^{\infty} G(x_i, t; y_j, T) e_j(y).
\]

where \( \{ H(y_j, T) \}_{j \in \mathbb{Z}} \) and \( \{ G(x_i, t; y_j, T) \}_{j \in \mathbb{Z}} \) are the values of the functions \( H \) and \( G \), sampled on the grid \( \{ y_j \}_{j \in \mathbb{Z}} \). We have that \( \tilde{H}(y, T) \) and \( \tilde{G}(x_i, t; y, T) \) are piecewise constant approximations of \( H(y, T) \) and \( G(x_i, t; y, T) \). The values of \( \tilde{H}(y, T) \) and \( \tilde{G}(x_i, t; y, T) \) in each interval \( [y_j - \frac{\Delta y}{2}, y_j + \frac{\Delta y}{2}] \) are given by the values of the functions \( G \) and \( H \) sampled in the mid point \( y_j \) of each interval. As explained in Section 2.1, the motivation for the choice of sampling is twofold. First, it leads to a discretization of the economy in which we can interpret the quantities \( \{ H(y_j, T) \}_{j \in \mathbb{Z}} \) and \( \{ \Delta y G(x_i, t; y_j, T) \}_{j \in \mathbb{Z}} \) as prices. Second, we can view Equations (3.9) and (3.10) as approximations of the orthogonal projection of \( G \) and \( H \) on the orthonormal set \( \{ e_j(y) \}_{j \in \mathbb{Z}} \). In the algorithm, we only need the quantities \( \{ H(y_j, T) \}_{j \in \mathbb{Z}} \) and \( \{ G(x_i, t; y_j, T) \}_{j \in \mathbb{Z}} \), which are the approximations \( \tilde{H}(y, T) \) and \( \tilde{G}(x_i, t; y, T) \) evaluated at the grid values \( \{ y_j \}_{j \in \mathbb{Z}} \). We use the full representations (3.9) and (3.10) in Appendix A.1 to prove Proposition 1. There, we show that relying on the sampling approximations (3.9) and (3.10) instead of the orthogonal projections given by the inner products as in (2.5), leaves the
convergence properties of the pricing algorithm unaffected.

3.2 Black-Scholes: the recursive step

The following proposition gives a recursion formula relating the approximated values of the option at different points of the grid \( \{y_j\}_{j=-\infty}^{\infty} \) and at different points in time \( t_{l+1} \) and \( t_l \). It also states the rate of convergence of the algorithm.

**Proposition 1** Let \( v_i(t_l) \) be defined for a set of date \( \{t_l\}_{l=1,...,L} \), with \( t_L = T \), as follows:

\[
(3.11) \quad v_i(t_l) = \sum_{j \in \mathbb{N}} G(y_i; t_l; y_j, t_{l+1}) H(y_j, t_{l+1}) \Delta y, \quad \text{for } l = L - 1,
\]

\[
(3.12) \quad v_i(t_l) = \max \left\{ H(y_i, t_l), \sum_{j \in \mathbb{N}} G(y_i; t_l; y_j, t_{l+1}) v_j(t_{l+1}) \Delta y \right\}, \quad \text{for } l = 1, \ldots, L - 2.
\]

Then, for each \( t_l \) in \( \{t_1, \ldots, t_{L-1}\} \), the approximated values \( v_i(t_l) \) defined in (3.11) and (3.12) converge to the true value \( V(y_i, t_l) \) with an approximation error of the order \( O((\Delta y)^2) \).

**Proof.** See Appendix A.1. \( \square \)

Under a sampling scheme we have \( \Delta y = 2^{-J} a \). The convergence rate is driven by the resolution level \( J \), and is equal to \( C'2^{-2J} \), where \( C' \) is a positive constant.

The main difference between Equations (3.11) and (3.12) is the following. In the right hand side of (3.11), we find the exact values taken by the payoff function \( H(y, T) \) on the grid \( \{y_j\}_{j \in \mathbb{N}} \), and there is no approximation of the payoff. On the right hand side of (3.12), we find the values \( \{v_j(t_{l+1})\}_{j \in \mathbb{N}} \) obtained in the previous step of the algorithm, and these are approximations of the true values \( \{V(y_j, t_{l+1})\}_{j \in \mathbb{N}} \). Regardless of this fundamental difference, Proposition 1 states that the convergence rate is the same for both cases. In the case of a European option with maturity \( T \), by taking \( t = t_l \) in (3.11), we obtain the approximated price of the option at \( t \).

Equation (3.12) allows us to recursively compute the values of the option at different points in time, and thus to price Bermudan and American options.
In the implementation of Section 2.1, we project the payoff function on a finite number of indicator functions. The values \( \{H(y_j, T)\}_{j=1,...,N} \) and \( \{\Delta y G(y_i, t; y_j, t_{l+1})\}_{j=1,...,N} \) are then the entries of the finite dimensions matrices \( H(T) \) and \( G(t_l; t_{l+1}) \), respectively. Hence, we can express Equations (3.11) and (3.12) in exactly the same matrix forms as in (2.4) and (2.7). When the time step is a constant \( \tau = t_{l+1} - t_l \), we get the fast and easy-to-implement algorithm of the introductory example. By choosing \( N \) sufficiently large, we can make the error introduced by truncating the infinite summations in (3.9) and (3.10) arbitrarily small. The truncation only suppresses indicator functions localized in regions where the state price density vanishes, and the local contribution to the computed expectation can be neglected.

Furthermore, as in Section 2.2, whenever an exercise date \( t_l \) coincides with a dividend paying date \( s_h \), we just need to replace the entries \( \{\Delta y G(y_i, t_l; y_j, t_{l+1})\}_{j=1,...,N} \) of \( G(t_l; t_{l+1}) \) with the values \( \{\Delta y G(y_i - \delta, t_l; y_j, t_{l+1})\}_{j=1,...,N} \), whenever \( t_l \in \{s_h\}_{h=1,...,H} \), to accommodate for a discrete dividend \( \delta \).

### 3.3 Stochastic volatility: the projection step

In the class of stochastic volatility models, there are two state variables, the underlying asset \( S_t \) and the variance \( \sigma_t^2 \). The bivariate state price density \( G_2(x, \xi, t; y, w, T) \) describes the discounted transition probability density from the asset level \( x = S_t \) and variance level \( \xi = \sigma_t^2 \) at time \( t \) to the asset level \( y = S_T \) and variance level \( w = \sigma_T^2 \) at time \( T \). Denote by \( \hat{G}_2(x, \xi, t; \lambda, \kappa, T) \) its Fourier transform, so that \( G_2(x, \xi, t; y, w, T) = \frac{1}{4\pi^2} \iint d\lambda d\kappa e^{-\iota(\lambda y + \kappa w)} \hat{G}_2(x, \xi, t; \lambda, \kappa, T) \), where \( \iota \) is the imaginary unit.

For the underlying asset, let the grid \( \{y_j\}_{j \in \mathbb{N}} \) and orthonormal set \( \{e_j(y)\}_{j \in \mathbb{N}} \) be defined as in the Black-Scholes case. We denote by \( \{\hat{e}_j(y)\}_{j \in \mathbb{N}} \) the Fourier transforms of \( \{e_j(y)\}_{j \in \mathbb{N}} \), such that\(^5\): \( e_j(y) = \frac{1}{2\pi} \int d\lambda e^{-\iota \lambda y} \hat{e}_j(\lambda) \).

For the variance, we use an equally spaced grid \( \{w_q\}_{q \in \mathbb{N}} \) for the values taken by the variables.

\(^5\)See Appendix A.3 for the analytic form of \( \hat{e}_j(y) \).
\( \xi \) and \( w \), such that \( \Delta w = w_{q+1} - w_q \). Let \( \{ \varepsilon_q(w) \}_{q \in \mathbb{N}} \) be the normalized indicator functions centered on the grid \( \{ w_q \}_{q \in \mathbb{N}} \) and of width \( \Delta w \), and \( \{ \hat{\varepsilon}_q(\kappa) \}_{q \in \mathbb{N}} \) be their Fourier transforms. Furthermore, let \( \{ \lambda_m \}_{m \in \mathbb{Z}} \) and \( \{ \kappa_{m'} \}_{m' \in \mathbb{Z}} \) be two regularly spaced grids of values taken by the transformed variables \( \lambda \) and \( \kappa \), with constant widths \( \Delta \lambda \) and \( \Delta \kappa \), respectively.

The approximation of the payoff function \( H(y, T) \) is the same as in (3.9). The projection step for the bivariate state price density \( G_2(x, \xi, t; y, w, T) \) is based on the approximation:

\[
\hat{G}_2(x_i, \xi_p, t; y, w, T) = \sqrt{\Delta y \Delta w} \sum_{j,q=1}^{\infty} \left( \frac{1}{4\pi^2} \sum_{r,z=-\infty}^{\infty} \hat{G}_2(x_i, \xi_p, t; \lambda_r, \kappa_z, T) \hat{\varepsilon}_j(-\lambda_r) \hat{\varepsilon}_q(-\kappa_z) \Delta \lambda \Delta \kappa \right) e_j(y) \varepsilon_q(w)
\]

where the second equality in (3.13) defines the quantities \( \{ \Gamma_2(x_i, \xi_p, t; y_j, w_q, T) \}_{j,q \in \mathbb{N}} \). The function \( \hat{G}_2(x_i, \xi_p, t; y, w, T) \) is a piecewise constant approximation of the state price density \( G_2(x_i, \xi_p, t; y, w, T) \). The value of \( \hat{G}_2(x_i, \xi_p, t; y, w, T) \) in each rectangle \( [y_j - \frac{\Delta y}{2}, y_j + \frac{\Delta y}{2}] \otimes [w_q - \frac{\Delta w}{2}, w_q + \frac{\Delta w}{2}] \) is \( \Gamma_2(x_i, \xi_p, t; y_j, w_q, T) \). To parallel the discussion following (3.10) in the Black-Scholes case, we interpret the quantities \( \{ \Gamma_2(x_i, \xi_p, t; y_j, w_q, T) \}_{j,q \in \mathbb{N}} \) as approximations of the values \( \{ G_2(x_i, \xi_p, t; y_j, w_q, T) \}_{j,q \in \mathbb{N}} \) of the function \( G_2(x_i, \xi_p, t; y, w, T) \) sampled on the bivariate grid \( \{(y_j, w_q)\}_{j,q \in \mathbb{N}} \). We can motivate these approximations as follows. The orthogonal projection of the state price density \( G_2(x_i, \xi_p, t; y, w, T) \) on the two orthonormal sets \( \{ e_j(y) \}_{j \in \mathbb{N}} \) and \( \{ \varepsilon_q(w) \}_{q \in \mathbb{N}} \) is given by the inner products \( \int \int dydw G_2(x_i, \xi_p, t; y, w, T) e_j(y) \varepsilon_q(w) \). Since we only know the closed form\(^6\) of \( \hat{G}_2(x_i, \xi_p, t; \lambda, \kappa, T) \) and not \( G_2(x_i, \xi_p, t; y, w, T) \), we exploit the

\(^6\)For the closed form of \( G_2(x_i, \xi_p, t; \lambda, \kappa, T) \), see Griebsch (2012).
following key relationship:

\[
\int\int dydw G_2(x_i, \xi_p, t; y, w, T) e_j(y) \varepsilon_q(w) = \int\int dydw \frac{1}{4\pi^2} \int d\lambda d\kappa e^{i(\lambda y + \kappa w)} \hat{G}_2(x_i, \xi_p, t; \lambda, \kappa, T) e_j(y) \varepsilon_q(w)
\]

Then we can view each \( \Gamma_2(x_i, \xi_p, t; y_j, w_q, T) \) as an approximation of the integral \( \frac{1}{4\pi^2} \int d\lambda d\kappa \hat{G}_2(x_i, \xi_p, t; \lambda, \kappa, T) \hat{e}_j(-\lambda) \hat{\varepsilon}_q(-\kappa) \) obtained by a direct sampling of the Fourier transforms \( \hat{G}_2(x_i, \xi_p, t; \lambda, \kappa, T) \), \( \hat{e}_j(-\lambda) \) and \( \hat{\varepsilon}_q(-\kappa) \) on the bivariate grid \( \{(y_j, w_q)\}_{j,q \in \mathbb{N}} \).

As in Section 3.1, only the quantities \( \{\Gamma_2(x_i, \xi_p, t; y_j, w_q, T)\}_{j,q \in \mathbb{N}} \) are inputs for the pricing algorithm. The representation (3.13) is only used in Appendix A.2 to prove the convergence properties of the algorithm.

### 3.4 Stochastic volatility: the recursive step

In the stochastic volatility framework, the recursion for a Bermudan option consists in moving backwards as in (2.6) with:

\[
V(x, \xi, t_l) = \max\{H(x, t_l), \mathbb{E}[e^{-r(t_{l+1}-t_l)} V(S_{t_{l+1}}, \sigma^2_{t_{l+1}}, t_{l+1}) | S_t = x, \sigma^2_t = \xi]\}.
\]

(3.14)

The recursive step in the Heston model is then the sampling counterpart of (3.14). The following proposition gives a recursion formula relating the approximated values of the option at different points of the bivariate grid \( \{(y_j, w_q)\}_{j,q \in \mathbb{N}} \) and at different points in time \( t_{l+1} \) and \( t_l \). It also states the rate of convergence of the algorithm. Define

\[
\Gamma_1(y_i, w_p, t_l; y_j, t_{l+1}) = \sum_{q=1}^{\infty} \Gamma_2(y_i, w_p, t_l; y_j, w_q, t_{l+1}) \sqrt{\Delta w}.
\]
Proposition 2 Let $v_{ip}(t_l)$ be defined for a set of date $\{t_l\}_{l=1, \ldots, L}$, with $t_L = T$, as follows:

\begin{equation}
 v_{ip}(t_l) = \sum_{j=1}^{\infty} \Gamma_1(y_i, w_p, t_l; y_j, t_{l+1}) H(y_j, t_{l+1}) \sqrt{\Delta y}, \quad \text{for } l = L - 1,
\end{equation}

\begin{equation}
 v_{ip}(t_l) = \max \left( H(y_i, t_l), \sum_{j,q=1}^{\infty} \Gamma_2(y_i, w_p, t_l; y_j, w_q, t_{l+1}) v_{jq}(t_{l+1}) \sqrt{\Delta y \Delta w} \right), \quad \text{for } l = 1, \ldots, L - 2.
\end{equation}

Then, for each $t_l$ in $\{t_1, \ldots, t_{L-1}\}$, the approximated values $v_{ip}(t_l)$ defined in (3.15) and (3.16) converge to the true value $V(y_i, w_p, t_l)$ with an approximation error of the order $O(\Delta^2)$, with $\Delta = \max\{\Delta y, \Delta w\}$.

**Proof.** See Appendix A.2. \(\square\)

Equation (3.15) gives, for $t = t_{L-1}$, the price of a European option in the Heston model. Since the payoff function $H(y, T)$ only depends on the value $y$ taken by the underlying asset at $t_L = T$, the computed price $v_{ip}(t_{L-1})$ depends on the stochastic variance only through the conditioning value $\sigma^2_{t_{L-1}} = w_p$. This is why we can use $\{\Gamma_1(y_i, w_p, t; y_j, T)\}_{j \in \mathbb{N}}$ instead of $\{\Gamma_2(y_i, w_p, t; y_j, w_q, T)\}_{j,q \in \mathbb{N}}$. Using the values $\{\Gamma_1(y_i, w_p, t; y_j, T)\}_{j \in \mathbb{N}}$ in (3.15) is equivalent to applying the projection step on the Fourier transform $\hat{G}_1(y_i, w_p, t; \lambda, T) = \int d\kappa \hat{G}_2(y_i, w_p, t; \lambda, \kappa, T)$. It is the univariate function $\hat{G}_1$ and not the bivariate $\hat{G}_2$ that appears, for instance, in the original work by Heston (1993) for a European option.

Equation (3.15) for the Heston model is similar to Equation (3.11) for the Black-Scholes model. The main difference is that we can sample directly the true function $G(x, t_{L-1}; y, t_L)$ in (3.11), so that the quantities $\{G(y_i, t_{L-1}; y_j, t_L)\}_{j \in \mathbb{N}}$ are exact values. Instead, the quantities $\{\Gamma_1(y_i, w_p, t_{L-1}; y_j, t_L)\}_{j \in \mathbb{N}}$ are approximations of the true values $\{G_1(y_i, w_p, t_{L-1}; y_j, t_L)\}_{j \in \mathbb{N}}$, obtained by a direct sampling of the Fourier transforms $\hat{G}_1(y_i, w_p, t_{L-1}; \lambda, t_L)$ and $\hat{e}_j(-\lambda)$.

Figure 3 presents graphically the projection and recursion steps in the bivariate case.
In the implementation, we truncate the summations in (3.15) and (3.16), so that the grid
\{(y_j, w_q)\}_{j=1,\ldots,N; q=1,\ldots,W} has N \times W points. We denote by \( V_2(t_i) \) the N \times W matrix of computed prices at time \( t = t_i \), that is \( V_{2,jq}(t_i) = v_{jq}(t_i) \). Let \( \Gamma_2(y_i, w_p, t_i; t_{i+1}) \) be the N \times W matrix of the approximated transition probabilities from the initial point \((y_i, w_p)\) to the end points of the entire grid \{(y_j, w_q)\}_{j=1,\ldots,N; q=1,\ldots,W}. As in Section 2.1, we integrate the normalization parameter \( \sqrt{\Delta y \Delta w} \) in the definition of the transition matrix. We then have that \( \Gamma_{2,jq}(y_i, w_p, t_i; t_{i+1}) = \Gamma_2(y_i, w_p, t_i; y_j, w_q, t_{i+1}) \sqrt{\Delta y \Delta w} \). Let \( \phi_j = \{\hat{e}_j(-\lambda_r)\}_{r=1,\ldots,R} \) and \( \varphi_q = \{\hat{e}_q(-\kappa_z)\}_{z=1,\ldots,Z} \) be the values of the functions \( \hat{e}_j(-\lambda) \) and \( \hat{e}_q(-\kappa) \) sampled at the grids \( \{\lambda_r\}_{r=1,\ldots,R} \) and \( \{\kappa_z\}_{z=1,\ldots,Z} \), respectively. Furthermore, we define the \( R \times N \) matrix \( \phi = (\phi_1, \ldots, \phi_N) \), the \( Z \times W \) matrix \( \varphi = (\varphi_1, \ldots, \varphi_W) \), and the \( R \times Z \) matrix \( \hat{G}_2(y_i, w_p, t_i; t_{i+1}) \) with entries \( \hat{G}_{2,rz}(y_i, w_p, t_i; t_{i+1}) = \hat{G}_2(y_i, w_p, t_i; \lambda_r, \kappa_z, t_{i+1}) \). Then we can write the coefficients of the projection step (3.13) in matrix form as:

\[
(3.17) \quad \Gamma_2(y_i, w_p, t_i; t_{i+1}) = \phi' \hat{G}_2(y_i, w_p, t_i; t_{i+1}) \varphi \sqrt{\Delta y \Delta w}.
\]

The recursive step (3.16) becomes:

\[
(3.18) \quad v_{2,p}(t_i) = \sum_{j=1}^{N} \sum_{q=1}^{W} \Gamma_2(y_i, w_p, t_i; y_j, w_q, t_{i+1}) v_{jq}(t_{i+1}) \sqrt{\Delta y \Delta w} = \Gamma_2(y_i, w_p, t_i; t_{i+1}) : v_{2}(t_{i+1}),
\]

where the symbol “:\:” denotes the Frobenius, or entry-wise, product.

Equation (3.18) asks a new matrix \( \Gamma_2(y_i, w_p, t_i; t_{i+1}) \) being computed for each starting point \((y_i, w_p)\). The following two remarks greatly simplify the computation of the transition matrices. First, in (3.17), only \( \hat{G}_2(y_i, w_p, t_i; t_{i+1}) \) actually depends on \( y_i \) and \( w_p \), so that \( \phi \) and \( \varphi \) only need to be computed once. Second, the evolution of the logarithm of asset prices and of the stochastic variance in the stochastic volatility model has the property that increments are independent of
the price level. This entails that if $M_2(\log(x), w, t_i; \log(y), w, t_{i+1}) = G_2(x, w, t_i; y, w, t_{i+1})y$ is
the bivariate state price density as a function of $\log(y)$ and if $\hat{M}_2(\log(x), w, t_i; \lambda, \kappa, t_{i+1})$ is its
Fourier transform, we can rewrite (3.17) and define the $N \times W$ matrix $\Psi_2(\log(y_i), w_p, t_i; t_{i+1})$
as:

$$\Psi_2(\log(y_i), w_p, t_i; t_{i+1}) = \phi' \hat{M}_2(\log(y_i), w_p, t_i; t_{i+1}) \varphi \sqrt{\Delta y \Delta w},$$

where the $R \times Z$ matrix $\hat{M}_2(\log(y_i), w_p, t_i; t_{i+1})$ corresponds to the $R \times Z$ matrix $\hat{G}_2(y_i, w_p, t_i; t_{i+1})$. Equation (3.18) becomes:

$$v_{ip}(t_i) = \Psi_2(\log(y_i), w_p, t_i; t_{i+1}) : v_2(t_{i+1}).$$

Then we have that $\Psi_{2,jq}(\log(y_{i+\zeta}), w_p, t_i; t_{i+1}) = \Psi_{2,j-\zeta q}(\log(y_i), w_p, t_i; t_{i+1})$ for $\zeta \in \mathbb{Z}$, provided that $0 < i + \zeta < N$. In implementations, we compute $\Psi_2(\log(y_i), w_p, t_i; t_{i+1})$ only once
for at-the-money values of $(\log(y_i), w_p)$, and reconstruct the other transition matrices exploiting
the space-translation invariance property. Again, this exemplifies the computational advantage
of direct sampling based on equally-spaced grids.

### 3.5 Numerical illustrations in the Heston model

Let us investigate the performance of our method in a standard affine model like the Heston
(1993) model. We study an American option, written on an asset $S_t$ that pays discrete dividends
and that evolves according to the following stochastic volatility process:

$$dX_t = \left(r - \frac{1}{2} \sigma_t^2\right) dt + \sqrt{\sigma_t^2} \cdot dW_1t,$$

$$d\sigma_t^2 = \left(\alpha - \beta \sigma_t^2\right) dt + \omega \sqrt{\sigma_t^2} \cdot dW_2t, \quad E(dW_1t \cdot dW_2t) = \rho dt.$$  

(3.19)

In (3.19), $X_t = \ln S_t$ and $\sigma_t^2$ is the variance process. We carry out two simulation studies. In
the first, we assume that the American call has a time to maturity of one year, and that a
dividend $d = 10$ is paid out after six months. In the second simulation exercise, the time to
maturity is still one year, but 3 dividends are distributed at $t_t = 0.25, 0.5, 0.75$. The process parameters values are the following: $r = 0.05, \alpha = 0.08, \beta = 2$ and $\omega = 0.2$. Moreover we choose the parameter $\rho$ to be equal to zero. The price for an at-the-money option ($S_t = K = 100$) is computed. The benchmark method in this analysis is a finite-difference (hereafter FD) numerical solution of the partial derivatives equation (PDE) that describes the evolution of the price process $V_t$ of the American call. We implement an alternating direction implicit (ADI) variant of the finite-difference scheme. For a discussion of FD like schemes see, for instance, 't Hout and Foulon (2010). This implementation is equivalent to a Crank-Nicolson scheme, which in standard problems converges at a rate $O((\Delta t)^2)$, where $\Delta t$ is the temporal discretization interval. Assuming that the contemporaneous correlation $\rho = 0$ simplifies the implementation of the FD scheme, in the sense that neglecting the correlation between $X_t$ and $\sigma^2_t$ makes the FD scheme easier and faster. On the other side, the speed and complexity of the recursive projection method are unaffected by the value chosen for the parameter $\rho$. The correlation is taken care of in the Green function $G_2(x, \xi, t; y, w, T)$ and consequently in the coefficients of the matrix $G$. Since the speed of the method depends on the number of entries in the $G$ matrix, and not by the values taken by the entries, it is clear that the choice of $\rho$ has no consequences in the convergence rate of the recursive projections. This is the first advantage of the recursive projection over the finite-difference schemes. In both the FD scheme and in the recursive projections the evolution of the price $V_t$ is studied on a rectangular grid in the space $(X, \sigma^2)$, with $X \in [\log(K) - 12\sigma_0\sqrt{T}, \log(K) + 8\sigma_0\sqrt{T}]$ and $\sigma^2 \in [0, 0.2]$, $\sigma^2_0 = \alpha/\beta$ being the mean of the variance process. In the FD scheme the parameter $m_s$ gives the number of equally spaced grid points in the $X$ direction, and $m_v$ the number of equally spaced grid points in the $\sigma^2$ direction, so that the grid points are $\{(X_i, \sigma^2_p)\}_{i=1,\ldots,m_s; p=1,\ldots,m_v}$. Like in Section 2.3, $J$ gives the resolution level in the recursive projection scheme along the $X$ direction.

In order to price an American option on dividend paying stocks, we should implement the FD scheme-equivalent of the recombining tree. This is practically unfeasible since it would
mean computing at each ex-dividend date a new option price at each point of the grid. We opt instead to compare at each ex-dividend date $s_h$ and at each grid point $(X_i, \sigma^2_p)$ the intrinsic value $H(X_i, s_h)$ with the continuation value $V(\tilde{X}_i - d, \sigma^2_p, t_l)$, where $\tilde{X}_i - d$ is the value of the $X$ grid closest to $X_i - d$. This amounts to perturbing the FD scheme at each ex-dividend rate, which could translate in a slower convergence than the theoretical $O((\Delta t)^2)$. This is a second advantage of the recursive projection over the finite-difference schemes, since as we explained in Section 2.3 the recursive projections can easily adapt to discrete dividends without them affecting the convergence properties. The recursive projections achieve convergence quickly in the $\sigma^2$ direction. The method does not seem to improve by setting a resolution level greater than $J_w = 5$, so we keep this value fixed throughout our simulations. The FD scheme is also not very sensitive to the number of points used in the $\sigma^2$ direction. No improvement is found beyond $m_w = 31$. Figure 4 compares the convergence speed of the two methods in the 1 dividend case. The graph on the right displays the pricing error of the FD scheme as a function of the time discretization parameter $L_T$. Each line is relative to a different value of the spacial discretization parameter $m_s$. The FD scheme needs 464 seconds to reach a 4 basis points relative error, with parameters $m_s = 6400$ and $L_T = 1600^7$. The graph on the left plots the relative pricing error of the recursive projections against the resolution level $J$. The recursive projections deliver a relative error of less than 1 basis point in 8 seconds. For comparable relative errors, the recursive projections show a gain in computation time of the order $10^2$. The graphs are plotted on a log$_2$–log$_2$ scale. This choice allows to appreciate the decrease in the absolute value of the relative error each time we double the sampling points in the recursive projections, or the discretization parameter $m_s$ in the FD scheme. The regression line on the left graph shows that the estimated slope is almost exactly the slope of $-2$ predicted by the theoretical convergence results.

Figure 5 shows the results for the 3 dividend case. It is clear that the FD scheme cannot

---

As a comparison, pricing an American option on a non-dividend paying stock using the simulation-based approach of Longstaff and Schwartz (2001) would take dozens of minutes (see Medvedev and Scaillet (2010)).
handle 3 dividends, since it has not achieved convergence after more than 1000 seconds. On the other hand the recursive projection scheme delivers a relative error below 10 basis points in 13 seconds, and below 1 basis point in 200 seconds.

Another notable difference between the $FD$ and the recursive projection method is that the latter demands much less changes to be adapted to different pricing problems. We have already seen for example in the Black-Scholes case that a digital payoff, both in the put and in the call case, can be handled. In equation (2.4), the matrix $G(t;T)$ depends only on the dynamics of the underlying asset and not on the payoff. It could be computed once for all and used to price different options with different payoff styles, since the payoff functional form only impacts the vector $H(T)$. Such a design is particularly suited for object-oriented programming, often used in quant desks. In finite-difference schemes, the transition matrices are not independent of the payoff function, as they depend on the conditions imposed on the function $V_t$ on the boundaries of the rectangular grid on which it is computed. Options with different payoffs do not share the same boundary conditions, and hence do not share the same transition matrices. It is feasible, as we actually did, to build a toolbox that receives as input the parameters of the dynamics, the parameters of the option contract and the payoff, and that generates automatically the transition matrix and gives the prices as an output. A toolbox that upon receiving the same inputs automatically generates transition matrices in a finite-difference framework, and uses them to price an option, is simply unthinkable.

4 Concluding remarks

In this we paper we introduce a fast and accurate method for pricing a large class of path dependent options, those for which the path is being monitored at discrete moments of time. We start from the observation that by monitoring the value of an option at discrete times, and by sampling the value function of the contract on a finite grid of values of the underlying asset, the evolution of the price process can be described in terms of matrix operators. The interpretation
of the elements of such a matrix in terms of a functional projection allows to extend the matrix approach to the pricing of contracts written on assets following processes whose transition probabilities do not have a known analytical expression in the direct space. The stochastic volatility process and the variance gamma processes are example of such frameworks. The recursive projection method owes its speed to the simplicity of its algorithm, based on sampling. The speed and accuracy of the method allows to study problems that are usually difficult to address due to numerical complexity. Our recursive projection allows to price American options on assets receiving dividends at discrete points in time. Treating discrete dividends as if they were a continuous payment stream is likely to lead to a large bias in the estimation of the process parameters and to severe mispricing of the option. This is just one example of problems of more an applied nature that can be seen as natural applications of our methodology.

There are two possible directions of further investigation that origin from our work. On one side the natural application of the algorithm is an empirical study. The speed of the recursive projections allows to verify potential suboptimal exercise strategies due to the use of poor approximations. Another interesting line of research is the use of different function bases. Candidate basis systems are functions sharing with the indicator function the localization property, but that may display higher regularity and enhance the convergence speed. The faster convergence should not come at the cost of a more complicated numerical implementation, and so the research has to go in the direction of functional projections that result in linear transformations of the sampled values (Sweldens, 1996, 1997).

References


Journal of Economic Dynamics and Control, 24, 1721-1746.


A Convergence rates

A.1 Black-Scholes framework

We prove Proposition 1 in two steps. Lemma 1 proves the convergence of the algorithm for a European option, that is Equation (3.11). Lemma 2 extends the result to the multiperiod case, thus proving Equation (3.12).

To get a simple proof of Proposition 1, we only use basic Taylor expansion arguments, and do not rely on arguments based on projection on orthonormal sets. The latter are key in the extension to the stochastic volatility framework, and in the proof of Proposition 2. We underline nevertheless that splitting up the integral in intervals of equal length is equivalent to projecting the value function and the state price density on the indicator function of each interval. This interpretation underpins the concept of projection step introduced in Section 3.1, but is not needed to show (3.11) and (3.12).

Lemma 1 Let $v_i(t)$ be defined as in Equation (3.11), with $t = t_{L-1}$. Then $v_i(t)$ converges to the true price $V(y_i, t)$ at a rate of the order $O((\Delta y)^2)$.

Proof. We have to check the speed at which the pricing error:

$$
|V(y_i, t) - v_i(t)| = \left| \int_0^\infty dy G(y_i, t; y, T)H(y, T) - \sum_{j=1}^{\infty} G(y_i, t; y_j, T)H(y_j, T)\Delta y \right|
$$

goes to zero as a function of the parameter $\Delta y$. Unless differently stated, the summation on $j$ runs from 1 to $\infty$ hereafter. The approximation error is given by

$$
|V(y_i, t) - v_i(t)| = \left| \sum_j \left[ \int_{y_j}^{y_{j+1}} dy G(y_i, t; y, T)H(y, T) - G(y_i, t; y_j, T)H(y_j, T)\Delta y \right] \right|
$$

We consider each summand separately. The state price density function $G(y_i, t; y, T)$ is twice continuously differentiable with respect to the underlying asset value $y$. The value function
\( H(y, T) \) is twice continuously differentiable everywhere on the positive semi-axis except at the strike price \( y = K \). We denote by \([y_j, \bar{y}_j]\) the interval centered on the grid point \( y_j \), and to which \( y = K \) belongs. Whenever \( j \neq \bar{j} \), we have that, for \( \eta_j \in [y_j, \bar{y}_j] \):

\[
\int_{y_j}^{\bar{y}_j} dy G(y_i, t; y, T) H(y, T) - G(y_i, t; y_j, T) H(y_j, T) \Delta y = \\
\int_{y_j}^{\bar{y}_j} dy \left[ G(y_i, t; y_j, T) H(y_j, T) + \left( G(y_i, t; y, T) H(y, T) \right)'_{y=y_j} (y - y_j) \right. \\
+ \frac{(y - y_j)^2}{2} \left( G(y_i, t; y, T) H(y, T) \right)''_{y=y_j} \right] - G(y_i, t; y_j, T) H(y_j, T) \Delta y \\
= G(y_i, t; y_j, T) H(y_j, T) \Delta y + 0 + \frac{(\Delta y)^3}{24} \left( G(y_i, t; y, T) H(y, T) \right)''_{y=y_j} - G(y_i, t; y_j, T) H(y_j, T) \Delta y \\
= \frac{(\Delta y)^3}{24} \left( G(y_i, t; y, T) H(y, T) \right)''_{y=y_j}.
\]

If \( j = \bar{j} \), \( H(y, T) \) is only once weakly differentiable, that is there exists a function \( H'(y, T) \) defined almost everywhere in \([y_j, \bar{y}_j]\) such that

\[
H(y, T) = H(\vartheta, T) + \int_0^1 d\omega (H(\vartheta, T))'_{\vartheta=\vartheta + \omega(y-y)} (y - \vartheta), \quad \text{for } y, \vartheta \in [y_j, \bar{y}_j].
\]

We can then write:

\[
\int_{y_j}^{\bar{y}_j} dy G(y_i, t; y, T) H(y, T) - G(y_i, t; y_j, T) H(y_j, T) \Delta y = \\
\int_{y_j}^{\bar{y}_j} dy \left[ G(y_i, t; y_j, T) H(y_j, T) + (y - y_j) \int_0^1 d\omega (G(y_i, t; \bar{\omega}, T) H(\bar{\omega}, T))'_{\bar{\omega}=y_j + \omega(y-y_j)} \right] \\
- G(y_i, t; y_j, T) H(y_j, T) \Delta y.
\]
The pricing error is then bounded by:

\[ |V(y_i, t) - v_i(t)| \]
\[ \leq \sum_{j \neq i} \frac{\Delta y_j^3}{24} \left| \left( \frac{G(y_i, t; y, T)H(y, T)}{y = y_j} \right)^n \right| + \frac{(\Delta y_i)^2}{4} \sup_{-\Delta y_i < \delta_i < \Delta y_i} \left| \left( \frac{G(y_i, t; y, T)H(y, T)}{y = y_i + \delta_i} \right) \right| \]
\[ = O((\Delta y_i)^2). \]

The function \( G(y_i, t; y, T) \) is integrable on the positive real axis, and this ensures that the above summation on the index \( j \) is finite. \( \square \)

**Lemma 2** Let \( v_i(t_l) \) be defined as in Equation (3.12), with \( l = 1, \ldots, L-1 \). Then \( v_i(t_l) \) converges to the true price \( V(y_i, t_l) \) at a rate of the order \( O((\Delta y_j)^2) \).

**Proof.** We consider a payoff evaluated at two dates \{\( t_{L-2}, t_{L-1} \)} prior to maturity, \( t_L = T \), namely \( t_{L-2} < t_{L-1} < T \). Then:

\[ v_i(t_{L-2}) = \sum_j G(y_i, t_{L-2}; y_j, t_{L-1})v_j(t_{L-1})\Delta y \]
\[ = \sum_j G(y_i, t_{L-2}; y_j, t_{L-1})[V(y_j, t_{L-1}) - V(y_j, t_{L-1}) + v_j(t_{L-1})]\Delta y \]
\[ = \sum_j G(y_i, t_{L-2}; y_j, t_{L-1})v_j(t_{L-1})\Delta y + \sum_j G(y_i, t_{L-2}; y_j, t_{L-1})[v_j(t_{L-1}) - V(y_j, t_{L-1})]\Delta y. \]

The quantities \{\( V(y_j, t_{L-1}) \}\}_{j \in \mathbb{N}} are exact values, so it follows from Lemma 1 that

\[ \sum_j G(y_i, t_{L-2}; y_j, t_{L-1})V(y_j, t_{L-1}) = V(y_i, t_{L-2}) + O((\Delta y_j)^2). \] Since \( v_j(t_{L-1}) = \sum_j G(y_j, t_{L-1}; y_j', T)H(y_j', T)\Delta y \), we have \( v_j(t_{L-1}) = V(y_j, t_{L-1}) + O((\Delta y_j)^2). \) Then:

\[ \sum_j G(y_i, t_{L-2}; y_j, t_{L-1})\Delta y[v_j(t_{L-1}) - V(y_j, t_{L-1})] \leq e^{-r(t_{L-1} - t_{L-2})} \sup_j |v_j(t_{L-1}) - V(y_j, t_{L-1})| = O((\Delta y_j)^2). \]

In the last inequality, we take advantage of the fact that since \( G(x, t_{L-2}; y, t_{L-1}) \) is the deterministic discount factor times a density, \( \sum_j G(y_i, t_{L-2}; y_j, t_{L-1})\Delta y = e^{-r(t_{L-2} - t_{L-1})}. \) Indeed,
the approximation operators built on indicator functions are shape preserving, see Dechevsky and Penev (1997) and Cosma, Scaillet and von Sachs (2007), and the property of integration to one of a density is preserved. It readily follows that $v_i(t_{L-2}) = V(y_i,t_{L-2}) + O((\Delta y)^2)$. The extension to prior dates $t_l = t_{L-3}, t_l = t_{L-4}, \ldots$, immediately follows by recursively applying the same arguments used above. \qed
A.2 Heston framework

We start by providing some definitions useful for the proof of Proposition 2:

\[
V^\perp(y, w, T) = \sum_{j,q} \int d\theta_1 \int d\theta_2 V(\theta_1, \theta_2, T) e_j(\theta_1) \varepsilon_q(\theta_2) e_j(y) \varepsilon_q(w),
\]

\[
G^\perp_2(y, \xi_p, t; y, w, T) = \sum_{j,q} \int d\theta_1 \int d\theta_2 G_2(y_i, \xi_p, t; \theta_1, \theta_2, T) e_j(\theta_1) \varepsilon_q(\theta_2) e_j(y) \varepsilon_q(w).
\]

Equations (A.20) and (A.21) define the coefficients \( \{V^\perp_{jq}\}_{j,q \in \mathbb{N}} \) and \( \{G^\perp_{2,ipjq}\}_{j,q \in \mathbb{N}} \) of the orthogonal projections \( V^\perp(y, w, T) \) and \( G^\perp_2(y, \xi_p, t; y, w, T) \). By orthogonality of the orthonormal sets \( \{e_j(y)\}_{j \in \mathbb{N}} \) and \( \{\varepsilon_q(y)\}_{q \in \mathbb{N}} \), we get:

\[
v^\perp_{ip}(t) = \int dy \int dw G^\perp_2(y, \xi_p, t; y, w, T) V^\perp(\theta_1, \theta_2, T) = \sum_{jq} G^\perp_{2,ipjq} V^\perp_{jq}.
\]

Moreover, we denote by \( v^\star_{ip}(t) \) the following approximation:

\[
v^\star_{ip}(t) = \sqrt{\Delta y \Delta w} \sum_{jq} \Gamma_2(y_i, w_p, t; y_j, w_q, T) V(y_j, w_q, T).
\]

Equation (A.23) is the approximation of the price of a European option for the known payoff \( V(y, w, T) \). Equation (A.23) extends Equation (3.15) by allowing the payoff function to depend explicitly on the variable \( w \). Whenever \( V(y, w, T) = H(y, T) \), Equations (3.15) and (A.23) coincide.

Let \( \Delta = \max\{\Delta y, \Delta w\} \). Then the proof of Proposition 2 is organized in the following three steps: 1) In Lemma 3, we show that the computed price \( v^\star_{ip}(t) \) verifies \( v^\star_{ip}(t) = v^\perp_{ip}(t) + O(\Delta^2) \). 2) In Lemma 4, we prove that \( v^\perp_{ip}(t) = V(y_i, w_p, t) + O(\Delta^2) \). This entails that \( v^\star_{ip}(t) = V(y_i, w_p, t) + O(\Delta^2) \), which proves Proposition 2 in the European option case, i.e. Equation (3.15) by setting \( t = t_{L-1} \). 3) In Lemma 5, we conclude by proving the recursive formula of Equation (3.16).
Lemma 3. The approximation error between \( v_{i \parallel}^*(t) \) and \( v_{i \perp}^*(t) \) defined in (A.22) and (A.23) satisfies:

\[
(A.24) \quad v_{i \perp}^*(t) = v_{i \parallel}^*(t) + O(\Delta^2), \quad \text{as } \Delta \to 0.
\]

Proof. We need to bound the difference:

\[
\sum_{jq} \left| \sqrt{\Delta y \Delta w} \Gamma_2(y_i, w_p, t; y_j, w_q, T)V(y_j, w_q, T) - G_{i \parallel j q}^\perp V_{j q}^\perp \right|.
\]

By Fourier isometry, we have:

\[
\int \int d\theta_1 d\theta_2 G_2(y_i, w_p, t; \theta_1, \theta_2, T)e_j(\theta_1)\varepsilon_q(\theta_2) = \frac{1}{4\pi^2} \int \int d\lambda d\kappa \hat{G}_2(y_i, w_p, t; \lambda, \kappa, T)\hat{e}_j(-\lambda)\hat{\varepsilon}_q(-\kappa).
\]

Then, we deduce:

\[
\sum_{jq} \left| \sqrt{\Delta y \Delta w} \Gamma_2(y_i, w_p, t; y_j, w_q, T)V(y_j, w_q, T) - G_{i \parallel j q}^\perp V_{j q}^\perp \right|
\]

\[
= \frac{1}{4\pi^2} \sum_{jq} \left| \sqrt{\Delta y \Delta w} \left( \sum_{r, z = -\infty} \hat{G}_2(x_i, \xi_p, t; \lambda_r, \kappa_z, T)\hat{e}_j(-\lambda_r)\hat{\varepsilon}_q(-\kappa_z)\Delta \lambda \Delta \kappa \right)V(y_j, w_q, T)ight.
\]

\[
- \left( \int \int d\lambda d\kappa \hat{G}_2(y_i, w_p, t; \lambda, \kappa, T)\hat{e}_j(-\lambda)\hat{\varepsilon}_q(-\kappa) \right) \left( \int \int d\theta_1 d\theta_2 V(\theta_1, t; \theta_2)\varepsilon_j(\theta_1)\varepsilon_q(\theta_2) \right)\right|.
\]

The functions \( \hat{G}_2(y_i, w_p, t; \lambda, \kappa, T) \), \( \hat{e}_j(-\kappa) \) and \( \hat{\varepsilon}_j(-\lambda) \) are twice continuously differentiable. The value function \( V(y, w, T) \) is at least once weakly differentiable. Moreover, let \( \Delta = \max \{ \Delta \kappa, \Delta \lambda \} \).

Then, using the same Taylor expansion arguments as in the proof of Proposition 1, we have that:

\[
\sum_{m, m' = -\infty} \hat{G}_2(x_i, \xi_p, t; \lambda_r, \kappa_z, T)\hat{e}_j(-\lambda_z)\hat{\varepsilon}_q(-\kappa_z)\Delta \lambda \Delta \kappa = \int \int d\lambda d\kappa \hat{G}_2(y_i, w_p, t; \lambda, \kappa, T)\hat{e}_j(-\lambda)\hat{\varepsilon}_q(-\kappa) + O(\Delta^2), \quad \text{as } \Delta \to 0,
\]

36
and
\[ \sqrt{\Delta y \Delta w} V(y_j, w_q, T) = \int \int d\theta_1 d\theta_2 V(\theta_1, \theta_2, T)e_j(\theta_1)e_q(\theta_2) + O(\Delta^2), \quad \text{as } \Delta \to 0. \]

It suffices then to choose \( \Delta = O(\Delta) \) to prove the statement of the Lemma.

**Lemma 4** The following equality holds:

(A.25) \[ v_{i p}^\perp(t) = V(y_i, w_p, t) + O(\Delta^2), \quad \text{as } \Delta \to 0. \]

**Proof.** We study the difference:

(A.26) \[ V(y_i, w_p, t) - v_{i p}^\perp(t) = \int \int dy dw G_2(y_i, w_p, t; y, w, T) V(y, w, T) - \sum_{jq} G_{2,ijpq}^\perp V_{jq}^\perp \]

\[ = \sum_{jq} \int_{y_j}^{\bar{y}_j} dy \int_{w_q}^{\bar{w}_q} dw \frac{1}{\Delta y \Delta w} \frac{1}{\Delta \theta_{1,2}} \int \int_{y_j}^{\bar{y}_j} d\theta_1 \int \int_{w_q}^{\bar{w}_q} d\theta_2 G_2(y_i, w_p, t; \theta_1, \theta_2, T) V(y, w, T). \]

Performing a Taylor expansion on the generic term of the first summation gives:

\[ \int_{y_j}^{\bar{y}_j} dy \int_{w_q}^{\bar{w}_q} dw G_2(y_i, w_p, t; y, w, T) V(y, w, T) = \int_{y_j}^{\bar{y}_j} dy \int_{w_q}^{\bar{w}_q} dw \left\{ G_2(y_i, w_p, t; y_j, w_q, T) V(y_j, w_q, T) \right\} \]

\[ + \int_0^1 d\omega \left[ \frac{\partial}{\partial \omega_1} (G_2(y_i, w_p, t; \omega_1, \omega_2, T) V(\omega_1, \omega_2, T)) \bigg|_{\omega_1 = y_j + \omega(y - y_j), \omega_2 = w_q + \omega(w - w_q)} (y - y_j) \right. \]

\[ + \frac{\partial}{\partial \omega_2} (G_2(y_i, w_p, t; \omega_1, \omega_2, T) V(\omega_1, \omega_2, T)) \bigg|_{\omega_1 = y_j + \omega(y - y_j), \omega_2 = w_q + \omega(w - w_q)} (w - w_q) \left. \right\} \]

\[ = G_2(y_i, w_p, t; y_j, w_q, T) V(y_j, w_q, T) \Delta y \Delta w \]

\[ + \int_{y_j}^{\bar{y}_j} dy \int_{w_q}^{\bar{w}_q} dw \int_0^1 d\omega \left[ \frac{\partial}{\partial \omega_1} (G_2(y_i, w_p, t; \omega_1, \omega_2, T) V(\omega_1, \omega_2, T)) \bigg|_{\omega_1 = y_j + \omega(y - y_j), \omega_2 = w_q + \omega(w - w_q)} (y - y_j) \right. \]

\[ + \frac{\partial}{\partial \omega_2} (G_2(y_i, w_p, t; \omega_1, \omega_2, T) V(\omega_1, \omega_2, T)) \bigg|_{\omega_1 = y_j + \omega(y - y_j), \omega_2 = w_q + \omega(w - w_q)} (w - w_q) \left. \right]. \]
In a similar way, we can develop the generic term of the second summation as follows:

$$\frac{1}{\Delta y \Delta w} \int_{y_j}^{y_f} \int_{w_q}^{w_f} d\theta_1 \int_{w_q}^{w_f} d\theta_2 G_2(y_i, w_p, t; \theta_1, \theta_2, T) \int_{y_j}^{y_f} \int_{w_q}^{w_f} d\theta_1 \int_{w_q}^{w_f} d\theta_2 V(\theta_1, \theta_2, T)$$

$$= G_2(y_i, w_p, t; y_j, w_q, T)V(y_j, w_q, T)\Delta y \Delta w$$

$$+ G_2(y_i, w_p, t; y_j, w_q, T) \int_{y_j}^{y_f} \int_{w_q}^{w_f} d\theta_1 \int_{w_q}^{w_f} d\theta_2 \int_0^1 d\omega_2 \left[ \frac{\partial}{\partial \omega_1} V(\omega_1, \omega_2, T)(\theta_1 - y_j) + \frac{\partial}{\partial \omega_2} V(\omega_1, \omega_2, T)(\theta_2 - \omega_q) \right]_{\omega_1 = y_j + \omega_2(\theta_1 - y_j)}_{\omega_2 = w_q + \omega_2(\theta_2 - w_q)}$$

$$+ V(y_j, w_q, T) \int_{y_j}^{y_f} \int_{w_q}^{w_f} d\theta_1 \int_{w_q}^{w_f} d\theta_2 \int_0^1 d\omega_1 \left[ \frac{\partial}{\partial \omega_1} G(y_i, w_p, t; \omega_1, \omega_2, T)(\theta_1 - y_j) + \frac{\partial}{\partial \omega_2} G(y_i, w_p, t; \omega_1, \omega_2, T)(\theta_2 - \omega_q) \right]_{\omega_1 = y_j + \omega_2(\theta_1 - y_j)}_{\omega_2 = w_q + \omega_2(\theta_2 - w_q)}$$

$$+ \frac{1}{\Delta y \Delta w} \int_{y_j}^{y_f} \int_{w_q}^{w_f} d\theta_1 \int_{w_q}^{w_f} d\theta_2 \int_0^1 d\omega_2 \left[ \frac{\partial}{\partial \omega_1} V(\omega_1, \omega_2, T)(\theta_1 - y_j) + \frac{\partial}{\partial \omega_2} V(\omega_1, \omega_2, T)(\theta_2 - \omega_q) \right]_{\omega_1 = y_j + \omega_2(\theta_1 - y_j)}_{\omega_2 = w_q + \omega_2(\theta_2 - w_q)}$$

$$\int_{y_j}^{y_f} \int_{w_q}^{w_f} d\theta_1 \int_{w_q}^{w_f} d\theta_2 \int_0^1 d\omega_1 \left[ \frac{\partial}{\partial \omega_1} G(y_i, w_p, t; \omega_1, \omega_2, T)(\theta_1 - y_j) + \frac{\partial}{\partial \omega_2} G(y_i, w_p, t; \omega_1, \omega_2, T)(\theta_2 - \omega_q) \right]_{\omega_1 = y_j + \omega_2(\theta_1 - y_j)}_{\omega_2 = w_q + \omega_2(\theta_2 - w_q)}$$

Since all derivatives are uniformly bounded in each rectangle $[y_j, y_f] \times [w_q, w_f]$, we can bound each integral using the same techniques used in the univariate case. Each term in (A.26) contributes a quantity of the order $O((\Delta y)^2)$, $O((\Delta w)^2)$ or higher, times a term containing $G_2$ or a derivative of $G_2$ evaluated in the rectangle $[y_j, y_f] \times [w_q, w_f]$. As in the univariate case, the terms containing $G_2$, or one of its derivatives, ensure that the double infinite summation is bounded. Then, by bounding from above $\Delta y$ and $\Delta w$ by $\Delta$, we get:

$$|V(x_i, w_p, t) - v_{ip}(t)| \leq O(\Delta^2),$$

which proves the lemma. \[\square\]

**Lemma 5** Let $v_{ip}(t_l)$ be defined as in Equation (3.16), with $l = 1, \ldots, L - 2$. Then $v_{ip}(t_l)$
converges to the true price $V(y_i, w_p, t_l)$ at a rate of the order $O(\Delta^2)$.

**Proof.** The proof follows exactly the same steps as the proof of Lemma 2. □

The proofs of Proposition 1 and Proposition 2 can be carried out in a more general framework, and for basis sets other than indicator functions. The key requirement is that only a finite number of basis functions contribute to the approximation of a function at a given point $(y_i, w_p)$. Examples are orthonormal wavelets, non-orthogonal, bi-orthogonal wavelet bases and B-splines. The use of these function bases may be useful when we need a basis that better adapts to the specific geometry of more complicated pricing problems.

### A.3 Analytic form for $\hat{e}_j(y)$

Let $e_j(y)$ be the indicator function of the interval $[y_j, y_{j+1})$, normalized according to the $L_2$ norm. Then the Fourier transform $\hat{e}_j(\lambda)$ is given by:

$$
\hat{e}_j(\lambda) = \frac{2}{\sqrt{y_j - y_{j+1}}} \left[ \cos \left( \frac{y_j + y_{j+1}}{2} \lambda \right) + \iota \sin \left( \frac{y_j + y_{j+1}}{2} \lambda \right) \right] \sin \left( \frac{y_j - y_{j+1}}{2} \lambda \right) / \lambda 
$$

$$
= \frac{2}{\lambda \sqrt{\Delta y}} \left[ \cos(y_j \lambda) + \iota \sin(y_j \lambda) \right] \frac{\sin(\Delta y \lambda)}{\lambda}.
$$

39
<table>
<thead>
<tr>
<th>Prices</th>
<th>Tree</th>
<th>Recursive Projection</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$S_t$</td>
<td>80</td>
</tr>
<tr>
<td>$\nu$</td>
<td>200</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>424</td>
</tr>
<tr>
<td></td>
<td>2000</td>
<td>6296</td>
</tr>
<tr>
<td></td>
<td>4000</td>
<td>97099</td>
</tr>
<tr>
<td></td>
<td>True</td>
<td>7.181</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relative errors (bp)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\nu$</td>
<td>200</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>424</td>
</tr>
<tr>
<td></td>
<td>2000</td>
<td>6296</td>
</tr>
<tr>
<td></td>
<td>4000</td>
<td>97099</td>
</tr>
</tbody>
</table>

Table 1: Comparison between a binomial tree and the recursive projection method on an American call option written on a dividend-paying stock in the Black-Scholes case. The option has a maturity of 3 years and a dividend $d = 2$ is paid at the end of each year. Other parameters are set equal to $r = 0.05$, $\sigma = 0.2$, $K = 100$. 
<table>
<thead>
<tr>
<th>$S_0$</th>
<th>80</th>
<th>100</th>
<th>120</th>
<th>80</th>
<th>100</th>
<th>120</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prices</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\nu$</td>
<td>100000</td>
<td>24</td>
<td>0.9667456</td>
<td>0.6800893</td>
<td>0.2946314</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>150000</td>
<td>47</td>
<td>0.9670993</td>
<td>0.6888984</td>
<td>0.2957528</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>200000</td>
<td>94</td>
<td>0.9667495</td>
<td>0.6815248</td>
<td>0.2939859</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>400000</td>
<td>378</td>
<td>0.9669498</td>
<td>0.6873797</td>
<td>0.293138</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>800000</td>
<td>2092</td>
<td>0.9668951</td>
<td>0.6832498</td>
<td>0.2950593</td>
<td>13</td>
</tr>
<tr>
<td>True</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Relative errors (bp)

<table>
<thead>
<tr>
<th>$\nu$</th>
<th>100000</th>
<th>3</th>
<th>72</th>
<th>21</th>
<th>9</th>
<th>6</th>
<th>61</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>150000</td>
<td>1</td>
<td>57</td>
<td>60</td>
<td>10</td>
<td>1</td>
<td>17</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>200000</td>
<td>3</td>
<td>51</td>
<td>0</td>
<td>11</td>
<td>1</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>400000</td>
<td>1</td>
<td>35</td>
<td>29</td>
<td>12</td>
<td>1</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>800000</td>
<td>1</td>
<td>26</td>
<td>36</td>
<td>13</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 2: Comparison between a binomial tree and the recursive projection method on a Bermudan digital put option in the Black-Scholes case. The option has a maturity of 10 years and can be exercised 4 times per year. Other parameters are set equal to $r = .1$, $\sigma = 0.2$, and $K = 100$. 
Figure 1: Recursive Projection scheme in the Black Scholes case is composed of two steps. First the projection step: the value function $V(y, t_{l+1})$ at $t = t_{l+1}$ and state price density functions $G(x, t_l; y, t_{l+1})$ are sampled (thick arrows). Second the recursive step: the sampled values are multiplied by the transition weights to obtain the value function at $t = t_l$ (thin arrows), which in turn will be the input for the following step of the algorithm.
Figure 2: Recursive scheme without dividends and with discrete dividends. In the top graph, at date $t = t_{l+1}$, the intrinsic value $H(y_i, t_{l+1})$ is compared with the continuation value $V(y_i, t_{l+1})$ at the same point $x = y_i$ (black ball). In the bottom graph, at the ex-dividend date $t = t_{l+1}$, the intrinsic value $H(y_i, t_{l+1})$ at $x = y_i$ (black ball) is compared with the continuation value $V(y_i - \delta, t_{l+1})$ at $x = y_i - \delta$ (red ball).
Figure 3: Recursive Projection scheme in the stochastic volatility case is composed of two steps. First the projection step (thick arrows): the value function $V(y, w, t_i)$ at $t = t_{i+1}$ is sampled and the state price density function $G_2(x, \xi, t_i; y, w, t_{i+1})$ is approximated by at $(y_j, w_q)$ by $\Gamma_2(x, \xi, t_i; y_j, w_q, t_{i+1})$. Second the recursive step: the sampled values are multiplied by the transition weights to obtain the value function $V(x = y_i, \xi = w_p, t_i)$ at $t = t_i$ (thin arrows), which in turn will be the input for the following step of the algorithm.
Figure 4: Comparison between the finite difference scheme and recursive projection on an American call option written on a dividend-paying stock in the Heston case. The option has a maturity of 1 years and a dividend $d = 10$ is paid at $t_l = 0.5$. Other parameters are set equal to $S_t = 100$, $K = 100$, $r = 0.05$, $\alpha = 0.08$, $\beta = 2$, $\omega = 0.2$. The parameter $m_2$ gives the number of points in the $X$ grid for the FD scheme, while the $\sigma^2$ grid has $m_w = 31$ points. The sampling grid in the recursive projections has size $2^J$. The resolution level in the $\sigma^2$ direction is $J_w = 5$. The dashed line in the right panel corresponds to a fitted linear regression, and shows that the estimated slope is close to the slope of $-2$ predicted by the theoretical convergence results.
Figure 5: Comparison between the finite difference scheme and recursive projection on an American call option written on a dividend-paying stock in the Heston case. The option has a maturity of 1 years and a dividend $d = 2$ is paid at dates $t_t = 0.25, 0.5, 0.75$. Other parameters are set equal to $S_t = 100$, $K = 100$, $r = 0.05$, $\alpha = 0.08$, $\beta = 2$, $\omega = 0.2$. The parameter $m_2$ gives the number of points in the $X$ grid for the FD scheme, while the $\sigma^2$ grid has $m_{\sigma} = 31$ points. The sampling grid in the recursive projections has size $2^J$. The resolution level in the $\sigma^2$ direction is $J_w = 5$. The dashed line in the right panel corresponds to a fitted linear regression, and shows that the estimated slope is close to the slope of $-2$ predicted by the theoretical convergence results.