Derivative Pricing – Semi-Analytic Lattice Integrator Approaches

Abstract

The aim of the pages set out below is to summarise an approach to derivative pricing that involves approximating the payout function of the derivative by a sum of a set of domain-limited functions (i.e. functions that only take non-zero values for some specific range of inputs). The functions are chosen so that the price of each element of the overall payout function can be calculated analytically. The overall price of the derivative can then be calculated in a quasi-analytic manner, merely by adding together the value contributions arising from each individual function. This can considerably speed up calculation times and can reduce the numerical noise otherwise often introduced into hedging parameter computations.

Contents

1. Introduction
2. Carrying out the required integrations

References

1. Introduction

[SemiAnalyticLatticeIntegratorApproaches1]

1.1 The price of a (European-style) derivative can be calculated as the discounted expected value of its payout function, where the expectation is carried out using the risk-neutral probability distribution. If the payout function and the risk-neutral probability distribution are both simple functions then the price of the derivative may be derivable analytically. For example, the Black-Scholes formulae can be derived analytically (see e.g. Black-Scholes derivation as the limit of a binomial tree or Black-Scholes derivation using stochastic calculus). The hedging parameters, i.e. *greeks*, applicable to the derivative may also be analytically tractable (see e.g. hedging parameters applicable to vanilla and binary puts and calls in a Black-Scholes world). However, for more complicated pay-offs or more complicated risk neutral probability distributions it is usually not possible to derive equivalent analytical formulae.

1.2 The most common approach used to circumvent this problem is to use numerical approaches such as binomial or trinomial trees that converge to the correct price or other hedging parameter as the tree becomes more and more finely grained. Unfortunately, these algorithms are not usually very good at handling singularities in derivative payout functions. These singularities can arise directly in the payout function, e.g. payout functions applicable to digital options have discontinuities at the strike price. They can also arise via discontinuous first partial derivatives with respect to the underlying (price) process. These are more common. For example, the first derivative of the payout function of a vanilla call option is discontinuous at the strike price because the payoff function has a kink there.

1.3 Some of the problems these discontinuities create can be mitigated by judicious choice of where to position the nodes of the relevant tree. However, an arguably better approach, if it is practical to implement, is to approximate the payoff function as the sum of components that are analytically tractable. In particular, it is often possible to find payoff functions with specified domain limits (i.e. ranges over which they are non-zero) that are analytically tractable. This leads to the semi-analytic lattice integrator ('SALI') approach, see e.g. Hu, Kerkhof, McCloud and Wackertapp (2006.

1.4 As with other derivative pricing approaches, the SALI approach notes that the value, V_t, of a (non-dividend paying) derivative at time t, relative to the chosen numeraire asset, N_t, satisfies:

$$N_t^{-1}V_t = \mathbf{E}^N(N_T^{-1}V_T|I_t)$$

where I_t is the information set (or *filtration*) generated by the underlying processes and $\mathbf{E}^N$ is the relevant risk-neutral expectation operator. Usually, we will restrict ourselves to Markovian models, in which case the above formula can also be written as:

$$N_t^{-1}V_t = \mathbf{E}^N(N_T^{-1}V_T|z_t)$$

1.5 The observation underlying SALI is that if the payout can be written as a function of the underlying Markov process then it can be decomposed into the sum of a finite number of smooth subcomponents i.e. as:

$$N_t^{-1}V_t = \sum_i \mathbf{E}^N\left(a_i(z_T)\prod_j \mathbf{I}_{b_{ij}(z_t)}\middle|z_t\right)$$

where z_t denotes a (typically) low-dimensional underlying Markov process.

Here the $\mathbf{I}$ are indicator functions, i.e. $\mathbf{I}_{b_{ij}(z_t)} = 1$ if $b_{ij}(z_t) \geq 0, = 0$ otherwise. The decomposition might use one smooth function between consecutive discontinuities, or it might use several that are pasted together, e.g. cublic spline functions.

1.6 The pricing problem can thus be re-expressed as:

$$N_t^{-1}V_t = \sum_i \int_{D_i} a_i(z_T)p(z_T|z_t)dz_t$$

where $D_i = \{z_T: b_{ij}(z_T) \geq 0\ \forall j\} = \bigcap_j D_{ij}$ where $D_{ij} = \{z_T: b_{ij}(z_T) \geq 0\}$. D_i can in practice be truncated to be within some 'envelope of support' that includes essentially all of the probability density applicable to the pricing problem. For example, for a Weiner process, one might use an outer envelope spreading out to, say, four standard deviations, since virtually none of the probability density is outside this spread. However, care is needed in such a truncation if the payoff function becomes sufficiently large sufficiently rapidly at the edge of the distribution, see e.g. the Cost of Capital pricing model.

2. Carrying out the required integrations

[SemiAnalyticLatticeIntegratorApproaches2]

2.1 There are several possible choices for the 'basis' function elements of SALI, i.e. the $a_i(z_T)$. If we are focusing on a single factor model, then z_T is a scalar function rather than a vector function. Natural choices of basis functions are then:

(a) Low-order piece-wise smooth polynomials, such as cubic splines. Only a few node points are usually necessary to obtain a pretty accurate representation of a smooth function. Hu, Kerkhof, McCloud and Wackertapp (2006) focus on this approach.

(b) Higher order polynomial curve fits. There are many different ways of approximating arbitrarily accurately a function over a given range by using a polynomial series expansion, typically formulated using orthogonal polynomials, e.g. Legendre polynomials.

(c) Curve fits using other function series that can arbitrarily accurately approximate a function over a given range, where the functions in question are more easily or more accurately capable of being integrated against the probability density in question or can more succinctly match the payoff function in question.

2.2 One reason why (c) may be better than (b) can be seen by considering how SALI might be applied to the special case of European-style vanilla call and put options in a Black-Scholes world (for which there are already analytic formulae, see hedging parameters applicable to vanilla and binary puts and calls in a Black-Scholes world). The underlying process (for a non-dividend bearing underlying) in this case involves:

$$S_T = F_T e^{-\sigma\sqrt{T}z-\frac{1}{2}\sigma^2 T}$$

where $z \sim N(0,1)$.

Thus the natural curve fit to use in this instance is an exponential, since we then recover exactly the Black-Scholes formulae, see e.g. Black-Scholes derivation using stochastic calculus. This corresponds to polynomial curve fitting of $\log S$ rather than S itself.

2.3 Various analytical results that can be used in this context when the payoff function is approximated using basis elements that are either polynomials or exponentials of polynomials (if the underlying follows a Weiner process or some straightforward variants) are described in integration of piece-wise polynomials against a Gaussian PDF.

References

[SemiAnalyticLatticeIntegratorApproachesRefs]

Hu, Z., Kerkhof, J., McCloud, P. and Wackertapp, J. (2006). Cutting edges using domain integration. *Risk*, November 2006