Technical Paper:

Numerical Methods in UnRisk

MathConsult GmbH
Altenberger Straße 69
A-4040 Linz, Austria
15th October, 2009

1 Adaptive Integration

This section is devoted to the presentation of the basic ideas of Adaptive Integration. The reader is assumed to be familiar with the very basic theory of option pricing (for example: as presented in [Hull (1993)]). We want, at time t0, to price a European option on an underlying equity MyEquity, which expires at time T. In the classical Black-Scholes theory, the price of MyEquity at time t, S(t) is assumed to follow a geometric Brownian motion

$$dS_t = \mu S_t dt + \sigma S_t dW_t \tag{1}$$

where dW_t is the increment of a Wiener process W. These increments dW can be interpreted as realizations of a random variable which is normally distributed with mean 0 and variance dt. Arbitrage arguments then lead to the Black-Scholes partial differential equation for the fair value V of the option,

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

valid in the domain $[0, \infty)$ x $[t_0, T]$. Note that, by applying these arbitrage arguments, the growth speed μ disappeared and was replaced by the risk-free rate r. We will return to this point later. The Black-Scholes equation is a parabolic differential equation backwards in time, which requires a final condition at expiry T to be well-posed. This final condition is

$$V(S, T) = \text{payoff}(S)$$

where, in the case of a call $\text{payoff}(S) = \max(S - K, 0)$, in the case of a put $\text{payoff}(S) = \max(K - S, 0)$. A Green´s function for the Black-Scholes equation is available (see, e.g., [Wilmott (1998)])

$$\frac{e^{-r(T-t)}}{\sigma S_1 \sqrt{2\pi(T-t)}} e^{-\frac{\left((T-t)\left(r-\frac{\sigma^2}{2}\right)+\log\left(\frac{S}{S_1}\right)\right)^2}{2\sigma^2(T-t)}}$$

and therefore the solution of the Black-Scholes differential equation satisfying the final condition is given by

$$V(S, t) = \int_0^\infty G(S, S_1, t, T)\text{Payoff}(S_1)\, dS_1$$

Hence, if we know the payoff function, and if we can calculate this integral, we can calculate the value of the European option. If we have a vanilla call or put option, then this integral can be calculated by substitution and integration by parts, and one obtains the well-known Black-Scholes formulae.

More than one time step-the semigroup property

We can write the last formula also as

$$V(S, t) = \int_0^\infty G(S, S_1, t, T)\text{Payoff}(S_1, T)\, dS_1$$

This can be shown to remain true, when we introduce a sequence of points t_k with $t_0 < t_1 < .... < t_{n-1} < t_n = T$ and write the recursion

$$V(S, t_j) = \int_0^\infty G(S, S_1, t_j, t_{j+1})\text{Payoff}(S_1, t_{j+1})\, dS_1 \tag{2}$$

Numerical Integration and Interpolation

Let us assume that we know $V(S_1, t_{j+1})$ for all S1 and we want to calculate $V(S, t_j)$. In general, the integral cannot be calculated analytically but needs the application of some integration scheme. Hence we write

$$V(S,t_j) \approx \sum_{k=1}^{K} \omega_k G\left(S, S_k, t_j, t_{j+1}\right) V\left(S_k, t_{j+1}\right)$$

with ω_k being the weights and S_k the nodes of the integration rule. In our implementation, S_k depend on S, t_j and t_{j+1}.

Typically, the solutions of the Black-Scholes equation are quite smooth and therefore the application of a high-order scheme for the numerical integration makes sense. In the UnRisk PRICING ENGINE realization, a sum of four 6-point Gauss integration rules is applied.

Now assume that we know $V(S, t_{j+1})$ not for all S, but only for certain nodes $S_k(k = 0, .., k_{max})$. Then, in general, the Gauss integration points will be different from the nodes. We approximate the value of the option at these Gauss points by interpolation. Typically, cubic interpolation is used in the UnRisk PRICING ENGINE. There are some special cases (for example around the barrier in the case of a barrier option), in which cubic interpolation tends to lead to oscillating solutions. In these case, linear interpolation is used in order to obtain even more stable results.

Gridding

If not overruled by the user (by the function call option NumericalParameters), the number of intervals in direction of the underlying is 200, and the maximal time step is 30 days, but an actual time step may be shorter to guarantee that all key dates (like the ex-dividend day and the day before) are hit. The gridding in the direction of the underlying depends on the instrument to be valuated and is essentially equidistant (for interest rate derivatives) or equidistant in logarithm (equity derivatives). To obtain good accuracy and stability, the gridding is further adapted for some instruments: Thus, e.g., for barrier options the grid is refined near the barrier.

Adaptive Integration

Thus, Adaptive Integration has two main ingredients:

a) Use high-order integration schemes for the calculation of the integrals arising from the application of Green´s functions to the partial differential equations of mathematical finance.

b) Use adaptive gridding schemes for the underlying as well as for time to make sure that fine grids are introduced only where necessary.

Accuracy and Stability

We consider a plain vanilla European put option and valuate it using binomial trees and using the UnRisk PRICING ENGINE. The put option has a life time of one year, the spot price and the exercise price are 100, the interest rate is assumed to be flat 5% per year (continuous compounding) and the volatility is 35% per year. The analytic value for this option is 11.25137, obtained by the

Black-Scholes formula.
Adaptive Integration using 500 points in the direction of the underlying delivers

max. Timestep	Value
500	11.2432
200	11.2367
100	11.2522
50	11.2513
30	11.2513
10	11.2513
5	11.2513
2	11.2513
1	11.2513

If one uses binomial trees for numerically valuating options, it is well known that there is some oscillatory behavior of the obtained numerical values between even and odd numbers of discretization levels. For the European put option from above, the values obtained by binomial trees with various steps are as follows

Number of levels	Value	Number of levels	Value
2	10.0528	3	12.6379
4	10.6199	5	12.075
8	10.9292	9	11.7048
16	11.089	17	11.4899
32	11.1699	33	11.3738
64	11.2106	65	11.3134
128	11.231	129	11.2826
256	11.2412	257	11.267
512	11.2463	513	11.2592
1024	11.2488	1025	11.2553
2048	11.2501	2049	11.2533
4096	11.2507	4097	11.2524

The numerical schemes implemented in the UnRisk PRICING ENGINE have directed special attention also to a stable valuation of the Greeks. Let us consider a European up-and-out call option. Let the option expire in 2 days, let the interest rate be 5% per year, the volatility 25% per year and let the barrier be 120. It is well known that binomial trees do not deliver robust Greeks near the barrier. With Adaptive Integration the plots of Delta and Gamma (the first and second derivatives of the option value with respect to the spot price of the underlying) obtained by Adaptive Integration and obtained by the analytic solution, which is available for a flat interest rate and a flat volatility look absolutely identical.

The transition formula (2) can be interpreted (see, e.g.,Wilmott (1998)) not only as the application of a Green function for the Black-Scholes partial differential equation to the end condition but also as follows:
The fair value of an option is the present value of the expected payoff at expiry

under a risk-neutral random walk for the underlying. This risk-neutral random walk is the same as in (1) but with μ replaced by the risk-free rate r.
Under the last point of view, Adaptive Integration is easily transferred to the pricing of interest rate derivatives. According to the interest rate model under consideration, one has to take into account different risk-free random walks. However, there is one big difference. The rate used for discounting must reflect that it is itself the underlying (or corresponds to the underlying in some way) and therefore, in order to obtain good convergence rates in terms of the time step, for discounting not only the interest rate at time t_j but also the different interest rates in the Gauss points at time t_{j+1} should be taken into account. For discounting the contribution of the value of the derivative instrument at any Gauss point (at time level t_{j+1}), the average of the rate at t_j and of the rate in the Gauss point is used.

2 Streamline Diffusion

In this chapter the basic ideas for the numerical solution under two factor models are presented. Considering as example a generalized Hull & White two-factor interest rate model (see Hull & White (1994)) and the underlying two-dimensional partial differential equation we will explain briefly the method of finite elements, which forms the basis of our algorithm, and the idea of streamline diffusion.

Example: General Hull & White two factor short rate model

In this model the first factor, the spot rate, r_t and the second factor, some kind of long-term interest rate, u_t are assumed to fulfill the following stochastic differential equations:

$$dr(t) = (\theta(t) + u - a(t)r(t))dt + \sigma_1(t)dW_1(t)$$

$$du(t) = -b(t)u(t)dt + \sigma_2(t)dW_2(t)$$

where a is the mean reversion speed, θ+u is the reversion level of r, and u itself reverts to a level of zero at rate b. dW_1 and dW_2 are increments of Wiener processes with instantaneous correlation $\rho(t)$, i.e.,

$$E[dW_1, dW_2] = \rho(t)dt, \qquad -1 <= \rho(t) <= 1.$$

σ_1 and σ_2 are the volatilities.
No arbitrage arguments lead to the following two-dimensional partial differential equation for the price V[r,u,t] of a zero-coupon bond with maturity T:

$$\begin{aligned} 0 =& \frac{\partial V}{\partial t}(r,u,t) + \frac{1}{2}\sigma_1^2(t)\frac{\partial^2 V}{\partial r^2}(r,u,t) + \frac{1}{2}\sigma_2^2(t)\frac{\partial^2 V}{\partial u^2}(r,u,t) \\ &+ \rho(t)\sigma_1(t)\sigma_2(t)\frac{\partial^2 V}{\partial r \partial u}(r,u,t) + (\theta(t) + u - a(t)r)\frac{\partial V}{\partial r}(r,u,t) - rV(r,u,t) \end{aligned}$$

The appropriate final condition is V[r,u,T]=1. We want to price the zero-coupon bond at time t0 numerically in order to explain the used discretizations and methods.

General Formulation of the Partial Differential Equation

A general formulation of the partial differential equations above in conservative form is:

$$\frac{\partial V}{\partial t} + div(\begin{pmatrix} a_{rr} & a_{ru} \\ a_{ur} & a_{uu} \end{pmatrix} gradV) + (a_r, a_u)gradV + aV = f$$

(omitting the notation for dependency on r,u, and t furthermore, in order to keep the formulas short).
In continuum mechanics equations of this type occur quite frequently. They are called diffusion-convection-reaction equations. It is well known that standard discretization methods often fail to give stable solutions (see Morton (1996)). Especially the convective term, which is (a_r ,a_u) grad V can cause severe problems in the solution. If it is treated using standard discretization methods, high oscillations in the computed solution depending on r and u are the consequences. So-called upwind techniques have to be used in order to gain stability.

Time Discretization - Crank Nicolson

It was already stressed in the section about Adaptive Integration that the solution process is of course backwards in time. Under the assumption that we know the value of the bond V^{j+1} at time t(j+1) for each r and u we want to calculate the value V^j at time t(j). Doing the time discretization in a rather general way, we obtain (with $\Delta t^j := t(j+1) - t(j)$)

$$\begin{aligned} \frac{V^{j+1} - V^j}{\Delta t^j} &+ \alpha(div(\begin{pmatrix} a_{rr}^{j+1} & a_{ru}^{j+1} \\ a_{ur}^{j+1} & a_{uu}^{j+1} \end{pmatrix} gradV^{j+1}) \\ &+(a_r^{j+1}, a_u^{j+1})gradV^{j+1} + a^{j+1}V^{j+1}) \\ &+(1-\alpha)(div(\begin{pmatrix} a_{rr}^{j} & a_{ru}^{j} \\ a_{ur}^{j} & a_{uu}^{j} \end{pmatrix} gradV^{j}) \\ &+(a_r^{j}, a_u^{j})gradV^{j} + a^{j}V^{j}) \\ &= \alpha f^{j+1} + (1-\alpha)f^j \end{aligned} \quad (3)$$

If we choose the parameter α equal to 0.5, we obtain a scheme of Crank-Nicolson type, which leads to quadratic convergence rates for the time discretization.
In order to get a finite dimensional form of our problem which can be solved numerically, we have to further discretize the equation and the computational domain, which is spanned by the state variables r and u. The choice of the size of this domain we will discuss in one of the following sections.

Space Discretization - Finite Elements

The basic idea of the Finite Element method is to divide the domain into a large number of, example given, rectangles or triangles and to define on the resulting grid locally Ansatz- and Testfunctions. The numerical solution of the problem is then represented by a linear combination of these Ansatzfunctions. The coefficients of the linear combination are the unknowns of the discretized equation system, which has to be determined.

Multiplying (3) formally by a function U, integrating over the domain and then doing integration by parts in the second order terms leads to

$$\alpha(-(\begin{pmatrix} a_{rr}^{j+1} & a_{ru}^{j+1} \\ a_{ur}^{j+1} & a_{uu}^{j+1} \end{pmatrix} gradV^{j+1}, gradU) + ((a_r^{j+1}, a_u^{j+1})gradV^{j+1}, U)$$

$$+((a^{j+1} + \frac{1}{\alpha\Delta t^j})V^{j+1}, U)) + (1-\alpha)(-(\begin{pmatrix} a_{rr}^{j} & a_{ru}^{j} \\ a_{ur}^{j} & a_{uu}^{j} \end{pmatrix} gradV^{j}, gradU)$$

$$+((a_r^j, a_u^j)gradV^j, U) + ((a^j + \frac{1}{\alpha\Delta t^j})V^j, U))$$

$$= \alpha(f^{j+1}, U) + (1-\alpha)(f^j, U)$$

which is the so-call weak variational formulation of the problem (where (.,.) denotes the inner product on the 2-dimensional infinite domain. More details concerning this and the theory of finite elements including Sobolev spaces can be found in Ciarlet (1978).

Streamline Diffusion - Going with the flow!

Up to now the special type of the equation, including convection, was not taken into account in the numerical method. In order to obtain stability in the solution and to avoid unrealistic oscillations, we use so-called streamline diffusion, which is a special upwind technique. Its name is originated in computational fluid dynamics, where, very roughly speaking, streamline diffusion means adding artificial diffusion in the direction of the flow. So, each point in the computational domain gains information from where the information really comes from, from opposite streamline direction. A detailed mathematical description of the streamline diffusion method can be found in Roos et al. (1996). Since in our case the convection is mainly determined by the drift and by the mean reversion, the direction and the magnitude of the flow varies in the computational domain. The higher the magnitude of the flow, the more artificial streamline diffusion is necessary.

Gridding - Time and Space

The relevant time interval for the pricing process spans from the maturity date to the valuation date. It is clear that there exist several key dates which have to be met exactly by the time discretization in order to obtain accurate results, e.g. settlement date, coupon dates, dividend dates, call dates and so on. In our algorithm for two factor models these key dates are set at the beginning to do the time discretization between these dates in an equidistant way. The size of the computational domain, spanned by r und u, is set in a way that the information of the prescribed boundary condition does not penetrate to the center, during the considered time interval. The center of the domain is determined by the current short rates. So the choice of the boundary conditions, which have to be set for solving the partial differential equation, has no influence on the computed result. From the practical point of view this is reflected in the fact that the propability of very high or low, maybe even negative, interest rates is very small.
Therefore it is clear that the size of the computational domain depends on the lifetime of the considered instrument and on the parameters which form the co-

efficient functions of the partial differential equation: volatility, drift and mean reversion. The computational time to price an instrument therefore does not depend linearly on its lifetime. The longer the lifetime, the higher the distance of information transport and the bigger the computational domain. To hold the accuracy constant the refinement of the discretization at least has to be constant. As a consequence the number of unknowns has to be increased.
If not overruled by the user (by the function call option NumericalParameters2D), the maximal number of intervals in both directions of the factors is set to 50, and the maximal time step is 20 days. An actual time step may be shorter to guarantee that all key dates are hit in the way that we have already explained in the beginning of this section. The gridding in factor directions is done in a way so that the discretization is finer near the current factor settings, which determine the center of the grid and become coarser towards the boundaries.

Comparison Trinomial Trees - Streamline Diffusion

To price instruments by the use of trinomial trees accurately a reasonable depth of the tree is required. In order to guarantee that the weights in the trinomial tree can be interpreted as probability measures, they have to be ≥ 0. This can be done by allowing the branching to be non-standard at the edge of the tree (see Hull & White (1996)). Applying this special branching technique means to change the underlying model, which is commented in Leippold & Wiener (n.d.) with the statement: "Of course altering the geometry of the tree is an arbitrary manipulation of the pricing problem and thus subject to some criticism.".
In the following table you can find a comparison of the numerical results using trinomial trees and the method of streamline diffusion for zero coupon bonds. The underlying two-factor model is of type General Hull & White with constant parameters:

$$\theta = 0.012, \quad a = 0.2, \quad b = 0.1, \quad \sigma_1 = 0.01, \quad \sigma_2 = 0.001, \quad \rho = 0.3$$

The special branching starts in level 10. The timestep is 1/10 years.

Life Time	Trinomial Trees	Streamline Diffusion	Analytic Solution
1 years	0.950341	0.950353	0.950353
2 years	0.901665	0.901756	0.901756
4 years	0.808564	0.809131	0.809135
10 years	0.572741	0.57662	0.576645
20 years	0.315969	0.324654	0.324704
30 years	0.17375	0.183224	0.18328

As expected, the trinomial tree approach works quite well for short life times of the bond, when up- and down-branching does not play a too important role. However, when we consider the 30 years zero coupon bond, the mispricing of the trinomial tree delivers a price which differs from the analytical value by 6%, which is not acceptable. This mispricing does not result from a too large time step, but from a change of the model by up- and down-branching. On the other hand, streamline diffusion delivers prices within the range of a basis point. Since the numerical implementation of the method of trinomial trees was done in Mathematica, whereas Finite Elements with Streamline Diffusion has been implemented in C++ a comparison of the computation times of these examples would not be fair to the trinomial tree approach.

3 Monte Carlo Methods

In this chapter we give a short summary about Monte Carlo methods in financial engineering as described in Glasserman (2003).

Monte Carlo methods are based on the analogy between probability and volume. Consider, for example, the problem of estimating the integral of a function f over the unit interval. We may represent the integral

$$\alpha = \int_0^1 f(x)dx$$

as an expectation $\mathcal{E}[f(U)]$, with U uniformly distributed between 0 and 1. Evaluating the function f at n random points and averaging the results produces the Monte Carlo estimate

$$\hat{\alpha}_n = \frac{1}{n}\sum_{i=1}^{n} f(u_i)$$

The law of large numbers ensures that this estimate converges to the correct value as the number of draws increases.

What does this have to do with financial engineering? A fundamental implication of asset pricing theory is that under certain circumstances, the price of a derivative security can be usefully represented as an expected value. Valuing derivatives thus reduces to computing expectations. In many cases, if we were to write the relevant expectation as an integral, we would find that its dimension is large or even infinite. This is precisely the sort of setting in which Monte Carlo methods become attractive.

Konvergenzgeschwindigkeit

References

Ciarlet, P. G. (1978), *The Finite Element Method for Elliptic Problems, Series "Studies in Mathematics and its Applications"*, North-Holland, Amsterdam.

Glasserman, P. (2003), *Monte Carlo Methods in Financial Engineering (Stochastic Modelling and Applied Probability)*, Springer.

Hull, J. C. (1993), *Options, Futures, and Other Derivative Securities (2nd edition)*, Prentice-Hall.

Hull, J. & White, A. (1994), 'Numerical Procedures for Implementing Term Structure Models II: Two-Factor Models', *Journal of Derivatives* **2**(2), 37–48.

Hull, J. & White, A. (1996), 'Using Hull-White Interest-Rate Trees', *Journal of Derivatives* .

Leippold, M. & Wiener, Z. (n.d.), 'Algorithms Behind Term Structure Models of Interest RatesII: The Hull-White Trinomial Tree of Interest Rates.'.

Morton, K. W. (1996), *Numerical Solution of Convection-Diffusion Problems*, Chapman & Hall, London.

Roos, H.-G., Stynes, M. & Tobiska, L. (1996), *Numerical Methods for Singularly Perturbed Differential Equations, Convection-Diffusion and Flow Problems*, Springer Verlag.

Wilmott, P. (1998), *Derivatives: The Theory and Practice of Financial Engineering*, Wiley.