Numerical Algorithms Group

Title: **Derivative Pricing – Validation of Results**

Summary: In this article we describe some possible approaches to the question of validation. In particular we focus on methods that aim to verify known mathematical properties of the underlying model. The discussion uses the Black-Scholes equations as an example, but the general principles could be applied to a wide range of application areas.

Introduction

Developing software to value derivatives by solving the Black-Scholes partial differential equation is a relatively straightforward task. But if this software will potentially be used frequently by a large number of clients, or as part of a larger modeling problem, perhaps embedded in some other application software, then validation of the results is a vital task, and this can often be considerably more challenging than developing the software itself.

In this article we describe some possible approaches to the question of validation. In particular we focus on methods that aim to verify known mathematical properties of the underlying model. The discussion uses the Black-Scholes equations as an example, but the general principles could be applied to a wide range of application areas.

Theoretical Background

As an example consider the Black-Scholes partial differential equation

$$\frac{\partial f}{\partial t}+(r-q)S\frac{\partial f}{\partial S}+\frac{s^2S^2}{2}\frac{\partial^2 f}{\partial S^2}=rf, \qquad S_{\min}<S<S_{\max}, \quad t_{\min}<t<t_{\max} \tag{1}$$

for the value *S* of a European or American put or call stock option, with exercise price X. In equation (1) *t* is time, *S* is the stock price, *r* is the risk free interest rate, *q* is the continuous dividend, and σ is the stock volatility. The parameters *r*, *q* and σ may each be either constant, or functions of time. The quantities of interest are the option value *f* and the values of various Greeks, which are partial derivatives of *f*.

It is well known (Hull [3]) that for European options, and American call options with zero dividend *q,* an analytic solution of (1) is provided by the Black-Scholes formulae [2]. For example, for a European call option the solution is given by:

$$f = Se^{-\hat{q}(T-t)}N(d_1)-Xe^{-\hat{r}(T-t)}N(d_2) \tag{2}$$

where

$$d_1=\frac{\ln(S/X)+(\hat{r}-\hat{q}+\bar{s}^2/2)(T-t)}{\bar{s}\sqrt{(T-t)}}, \quad d_2=\frac{\ln(S/X)+(\hat{r}-\hat{q}-\bar{s}^2/2)(T-t)}{\bar{s}\sqrt{(T-t)}}=d_1-\bar{s}\sqrt{(T-t)}$$

N(x) is the cumulative Normal distribution function, and *N′(x)* is its derivative

$$N(x)=\frac{1}{\sqrt{2p}}\int_{-\infty}^{x}e^{-p^2/2}dp, \quad N'(x)=\frac{1}{\sqrt{2p}}e^{-x^2/2}$$

The functions $\hat{q}$, $\hat{r}$, $\hat{s}$ and $\bar{s}$ are average values of *q*, *r* and s over the time to maturity:

$$\hat{q} = \frac{1}{T-t}\int_t^T q(p)dp\,, \quad \hat{r} = \frac{1}{T-t}\int_t^T r(p)dp\,, \quad \hat{s} = \frac{1}{T-t}\int_t^T s\,(p)dp\,, \quad \bar{s} = \left(\frac{1}{T-t}\int_t^T s^{2}(p)dp\right)^{1/2} \qquad (3)$$

Note that $\bar{s}$ is a second-order average of s.

A similar analytic solution exists for European put options, or the option may simply be valued using the put-call parity relation. For American call options, and American put options with non-zero dividends there is no such analytic solution and the value must be approximated (for example by the lattice method, or a finite difference scheme).

Case Study

The discussion of validation techniques will use the following suite of software as a case study:

- FD - computes a finite-difference solution of the Black-Scholes equation (1), returning the value of the derivative, as well as values of various Greeks. The time-stepping scheme is a theta-scheme, which includes the forward and backward Euler methods, the Crank-Nicolson scheme, and a range of intermediate methods. The parameters *r*, *q* and σ may each be either constant, or functions of time described by values at a sequence of discrete times.
- ANALYTIC - computes an analytic solution of the Black-Scholes equation in the case of European options or American call options with zero dividend. The computation is based on equation (2) and the equivalent formula for European put options. The analytic values of various Greeks are also returned.
- AV - is a utility which computes time-averaged values $\hat{q}$, $\hat{r}$, $\hat{s}$, $\bar{s}$ of *q*, *r* and s as required by ANALYTIC. This is achieved by approximating the integrals in (3) by numerical integration. In cases where the time-dependent functions are known integrable functions of time the exact integrals could be supplied instead of the approximations generated by AV.

General Testing Strategies

The general testing strategy described here is based on the use of *stringent test programs*. These are developed at the same time as the software itself, and are designed to provide a test-bed which can be run on a wide range of different platforms and architectures, with different compilers and options. Detailed results from the stringent test programs can be written to file and compared with base results generated on a different machine, or on the same machine at a different version.

The stringent tests must do the following:

1. Trigger all possible error exits and exceptions generated by the function, and check that the behaviour and messages are correct.
2. Check that all input arguments are unchanged by the function in all cases.
3. Exercise as many lines of code as possible. Profiling tools can be used to check the code coverage.
4. Exercise as many potential paths through the code as possible.
5. Check that all special cases and trivial cases produce the correct results, for example a linear equation solver must handle the case of a diagonal matrix or a 1x1 matrix correctly.

6. Run a sequence of realistic test examples. In each case every result of the function must be compared against a result computed by some different method, or alternatively a known mathematical property of the result should be validated. For example, if a numerical integration method is known theoretically to have an error bounded by

$$\| e \| \leq \max_{z \in (0,1)} \left| \frac{h^4 u^{(4)}(z)}{24} \right|$$

for some function *u*, then the stringent test program should evaluate the fourth derivative of *u* and compute the upper bound in order to check that the bound is achieved.

In general the implementation of stages 1-5 are relatively straightforward. The testing of mathematical properties in 6 is the most challenging and interesting part of the process.
This is the part that we describe in the next section for the Black-Scholes software.

Validation of the Black-Scholes Software

In the case of the Black-Scholes software the existence of a known analytic solution in certain cases enables us to check the results of the finite-difference solver. However, we also need to validate the results produced by the function that computes this analytic solution. It would be possible simply to re-compute the formula (2) using a different piece of code, but this does not really test any mathematical properties. Instead we prefer to validate that the computed analytic solution satisfies the partial differential equation (1). To do this it is necessary to compute derivatives of the analytic solution returned. This can be achieved by returning to the fundamental definitions of derivatives from calculus

$$\left.\frac{\partial f}{\partial S}\right|_{(S,t)} = \lim_{dS \to 0} \frac{f(S+dS,t) - f(S,t)}{dS}, \qquad \left.\frac{\partial f}{\partial S}\right|_{(S,t)} = \lim_{dt \to 0} \frac{f(S,t+dt) - f(S,t)}{dt} \tag{4}$$

So for a sequence of random points within the *(S,t)* domain the following test is carried out. The analytic solution is computed using ANALYTIC at the chosen point, and also at a neighbouring point perturbed by a very small displacement in the *S* direction. Using these values in (4) the derivative with respect to *S* is calculated. This is then compared against the analytic expression returned from ANALYTIC for the Greek:

$$\Delta = \frac{\partial f}{\partial S} = e^{-\hat{q}(T-t)} N(d_1) + \frac{S e^{-\hat{q}(T-t)} N'(d_1) - X e^{-\hat{r}(T-t)} N'(d_2)}{\overline{s} S \sqrt{T-t}}$$

which can be obtained by differentiation of (2). By perturbing also in the *t* direction and applying equation (4) the stringent test program also validates the other Greeks calculated by ANALYTIC

$$\Gamma = \frac{\partial^2 f}{\partial S^2}, \ \Theta = \frac{\partial f}{\partial S}, \ \Lambda = \frac{\partial f}{\partial s}, \ r = \frac{\partial f}{\partial r}$$

Note that for Λ and r the process is slightly different. In these cases the perturbations are made to the parameters s and *r*, rather than *S* and *t*. However, the principle is the same. Having validated all the Greeks it remains to check that the analytic solution satisfies the partial differential equation (1). This is done by checking that *f* satisfies:

$$\Theta + (r - q)S\Delta + \frac{s^2 S^2}{2}\Gamma = rf \ ,$$

which is simply a restatement of (1) using the Greeks.

This validation of ANALYTIC is carried out for various types of options and other variations in the input arguments. In addition the stringent program checks that the boundary conditions are correctly satisfied.

In the case of time-varying values of *q*, *r* and S it is also necessary to validate the functioning of AV, which evaluates the average values defined in (3) and required in (2). Because of the numerical integration used internally by AV we know that the results for the first-order averages should be exact for all cubic polynomials. So the stringent test program for this routine tests the results for constant, linear, quadratic and cubic basis functions. The second-order average is only exact for linear functions, so this is tested for constant and linear basis functions.

Having validated the analytic solver the stringent test program proceeds to test the finite-difference method FD. Since this is an approximate method the results will not be exact, but should converge towards the analytic solution as the time-step and mesh size are reduced [5]. For all choices of time-stepping scheme the error in the option value and Greeks should decrease by a factor of at least half when the mesh and time-step are both halved. This shows that the error is at least first order in both *S* and *t*. To test this the finite-difference function is called repeatedly on a sequence of meshes of increasing refinement. On each mesh the error norm for *f* and all the Greeks is computed using the analytic solution. After completion of the sequence of refinements the convergence property is validated.

In cases where the analytic solution is not valid the results of FD are compared in a similar way against the analytic approximation of Macmillan/Barone-Adesi and Whaley [1].

Experience with this Approach

The validation procedures described here were applied by the Numerical Algorithms Group to a new suite of Black-Scholes routines developed for Mark 20 of its Numerical Library [4]. The stringent test program was developed in parallel with the software itself, and new tests were added to the stringent program as new functionality was added to the source code. The resulting early error detection helped to accelerate the development process. Despite careful evaluation of the mathematical expressions for the option value and Greeks, and careful programming of the source code, mistakes were still made, detected by the stringent test program, and corrected.

The code will now be implemented on a wide range of different platforms, and will form the basis of related functions for the NAG C Library. The stringent test program will provide a vital check of the correctness of all these implementations.

References

[1] Barone-Adesi G and Whaley R E, *Efficient analytic approximation of American option values*, Journal of Finance, 42, pp.301-320, 1987.

[2] Black F and Scholes M, *The pricing of options and corporate liabilities*, Journal of Political Economy, 81, pp 637-659, 1973.

[3] Hull J, *Options, Futures, and Other Derivative Securities* Prentice-Hall, 1989.

[4] NAG Ltd, *The Fortran 77 Library Mark 20*, NAG Ltd, Oxford, Forthcoming

[5] Wilmott P, Howison S and Dewynne J, *The Mathematics of Financial Derivatives*, Cambridge University Press, 1995.

By Gareth Shaw.
Gareth Shaw works for the Numerical Algorithms Group, which creates components and other software used by most of the major finance houses in the world. Questions can be forwarded to him via infodesk@nag.com.

Originally published by Financial Engineering News, August 2001 (www.fenews.com).

Numerical Algorithms Group
www.nag.com
infodesk@nag.com