Editor: Denis Donnelly, donnelly@siena.edu

Black–Scholes for Scientific Computing Students

By Desmond J. Higham

MATHEMATICAL FINANCE PROVIDES A MODERN, ATTRACTIVE SOURCE OF EXAMPLES AND CASE STUDIES FOR SCIENTIFIC COMPUTING CLASSES. IN THIS INSTALLMENT OF Education, I will show how educators can use the Nobel Prize-winning Black–Scholes option valuation theory to motivate exercises in Monte Carlo simulation, matrix computation, and numerical methods for partial differential equations (PDEs).

Option Valuation for Science Students

My colleague Xuerong Mao and I designed a class, "The Mathematics of Financial Derivatives," for final-year mathematics-based undergraduate students at the University of Strathclyde. The class pulls together ideas from mathematical modeling, stochastics, mathematical analysis, and computational methods, and has no prerequisites beyond freshman-level calculus and linear algebra. We have found that financial option valuation is an easily-motivated peg on which to hang a range of applied and theoretical tools. The class has proved popular, attracting approximately 80 percent of all possible takers, including many students aiming for joint degrees in mathematics and computer science; mathematics and physics; and mathematics, statistics, and economics. I'll focus here on the computational side of option valuation. Instructors can introduce the necessary finance background in just a few minutes of lecture time, and hence they may insert realistic option valuation problems into a range of numerical methods classes. This material is also eminently suitable as a source of open-ended, individual-study projects involving scientific computation. I'll detail three methods: Monte Carlo, binomial, and finite differences. In each case, I'll show how we can use the method to value a European call option. Short Matlab[1] codes make the ideas concrete; you can download these from www.maths.strath.ac.uk/~aas96106/algfiles.html. After illustrating each method, I provide a list of project suggestions. Lecturers can use these as the basis of homework problems or larger-scale independent study assignments.

Before discussing the three methods, let's review the concept of a European call option and the corresponding Black–Scholes formula.

Financial Options

Suppose I phone you today with the following offer: in three months' time, you have the option to purchase Microsoft shares from me for $25 per share.

The key point is that you have the *option* to buy the shares. Three months from now, you will check their market price and decide whether to exercise that option. (In practice, you would exercise the option if and only if the market price were greater than $25, in which case you could immediately resell for an instant profit.) This deal has no downside for you—three months from now you either make a profit or walk away unscathed. I, on the other hand, have no potential gain and an unlimited potential loss. To compensate, there will be a cost for you to enter into the option contract. You must pay me some money up front.

The option valuation problem is thus to compute a fair value for the option. More precisely, it is to compute a fair value at which the option may be bought and sold on an open market.

The option I just described is a *European call*. The Microsoft shares are an example of an *asset*—a financial quantity with a definite current value but an uncertain future value. Formalizing the idea and introducing some notation, we have: A European call option gives its holder the opportunity to purchase from the writer an asset at an agreed *expiry date* $t = T$ at an agreed *exercise price* E.

If we let $S(t)$ denote the asset value at time t, then the final time payoff for the European call is $\max(S(T) - E, 0)$ because

- If $S(T) > E$, the option will be exercised for a profit of $S(T) - E$.
- If $S(T) \leq E$, the option will not be exercised.

Options were first traded on the open market in 1973.

 Copublished by the IEEE CS and the AIP 1521-9615/04/$20.00 © 2004 IEEE

Since that time, demand for option contracts has undergone a remarkable growth; trading in options typically far outstrips that for the underlying assets. We can attribute popularity of options to three facts:

- They are attractive to *speculators* because they have an idea how the asset price will evolve and wish to gamble. Taking out an option generally gives a better upside (and correspondingly worse downside) than investing in the asset.
- They are attractive to individuals and institutions wishing to mitigate their *exposure to risk*. Options can be regarded as insurance policies against unfavorable movements in the market.
- There is a logical, systematic theory for working out how much an option should cost.

The first two points create a demand for option trading and the third makes it viable for "marketmakers" (who are obliged to buy and sell options whenever asked to do so) to find sensible prices.

A vast array of reference material on option valuation caters to all tastes and backgrounds. For a comprehensive treatment, ranging from the practicalities of how money is exchanged to theoretical and algorithmic aspects of option valuation, we recommend the classic finance/business-student oriented text.[2] My recent undergraduate book, *An Introduction to Financial Option Valuation*,[3] is aimed at mathematics-based students and gives equal weight to modeling, analysis, and computational methods. I cover all the material in this article in greater depth in my book.[3]

Real-world option data is freely available from several sources, including *The Wall Street Journal* (www.wsj.com), the *Financial Times* (www.ft.com), and the Yahoo! Finance site (http://finance.yahoo.com/). On a slightly lighter note, the fascinating story of how some of the academics behind option valuation theory tried—and eventually spectacularly failed—to put their ideas into practice with real money, is told in the highly readable book *The Predictors*.[4]

Asset Price Model

The Black–Scholes theory models the asset price as a *stochastic process*, a random variable that depends on t. From a computer simulation perspective, we need to know how to generate a typical discrete asset path. The model says that given $S_0 = S(0)$, prices $S(t_i)$ for the asset at times $t = t_i = i\Delta t$ can be generated from the recurrence

$$S(t_{i+1}) = S(t_i)e^{(\mu-\frac{1}{2}\sigma^2)\Delta t+\sigma\sqrt{\Delta t}\xi_i},$$

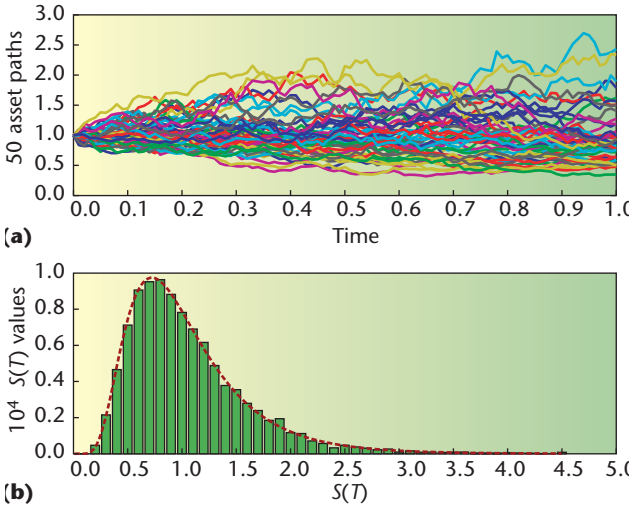

Figure 1. Asset price paths (a) and (b) a histogram at expiry time. Each asset path is determined by calls to a pseudorandom number generator. The histogram summarizes the values arising at time *t* = *1*.

where the parameter μ is the asset's *expected growth rate*, the parameter σ is the asset's *volatility*, and ξ_i is a sample from a normal(0,1) pseudorandom number generator.

In Matlab, we could compute and plot such a path as follows:

```
>> T = 1; N = 100; Dt = T/N; mu = 0.1; sigma
= 0.3; Szero = 1;
>>    Spath    =    Szero*cumprod(exp((mu-
sigma^2)*Dt+sigma*sqrt(Dt)*randn(N,1)));
>> plot(Spath)
```

Figure 1a shows 50 such paths—in each case, the discrete points $(t_i, S(t_i))$ are joined to give a piecewise linear curve. At the expiry date $t = T$, the asset price is a random variable with density given by

$$f(x) = \frac{\exp\left(\dfrac{-(\log(x/S_0)-(\mu-\sigma^2/2)T)^2}{2\sigma^2 T}\right)}{x\sigma\sqrt{2\pi T}}, \text{ for } x > 0,$$

with $f(x) = 0$ for $x \le 0$. To confirm this, Figure 1b gives a histogram where the final asset prices $S(T)$ for 10,000 paths have been binned. The density curve is superimposed as a dashed line.

Black–Scholes Formula

In addition to employing the simple asset price model, Fischer Black and Myron Scholes[5] imposed several simplifying assumptions about the options market. Then they made creative use of the no-arbitrage ("no free

```
function C = bsf(S,t,E,r,sigma,T)
% function C = bsf(S,t,E,r,sigma,T)
%
% Black-Scholes formula for a European call
%

tau = T-t;
if tau > 0
   d1 = (log(S/E) + (r + 0.5*sigma^2)*tau)/(sigma*sqrt(tau));
   d2 = d1 - sigma*sqrt(tau);
   N1 = 0.5*(1+erf(d1/sqrt(2)));
   N2 = 0.5*(1+erf(d2/sqrt(2)));
   C = S*N1-E*exp(-r*tau)*N2;
else
   C = max(S-E,0);
end
```

Figure 2. Listing of the function `bsf.m` for the Black-Scholes formula.

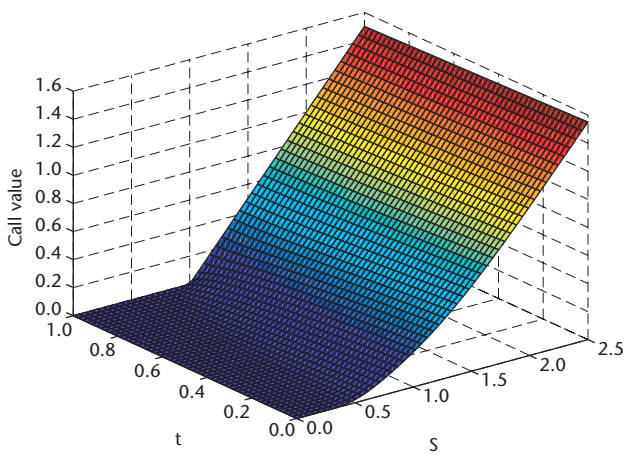

Figure 3. Black–Scholes surface for a European call. The height of the surface gives the option value for asset price *S* and time *t*.

lunch") principle to come up with the following formula for the value of a European call option at time t and asset price S:

$$C(S, t) = SN(d_1) - Ee^{-\rho(T-t)}N(d_2), \text{ where}$$

$$d_1 = \frac{\log(S/E) + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}},$$

$$d_2 = d_1 - \sigma\sqrt{T-t}$$

and $N(\bullet)$ is the Normal(0,1) distribution function,

$$N(x) := \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{s^2}{2}}\,ds.$$

The parameter r in the formula is the continuously compounded interest rate. If the asset price today (time zero) is S_0, then the Black–Scholes call option value is $C(S_0, 0)$. The Matlab function `bsf.m` in Figure 2 gives one way of implementing the formula, via the built-in error function `erf`. An example of the function in use is

```
>> S = 2; t = 0; E = 1; r = 0.05; sigma
     =      0.25; T = 3;
>> C = bsf(S,t,E,r,sigma,T)
C =
    1.1447
```

In Figure 3, we plot a Black–Scholes surface $C(S, t)$ as a function of S and t. Here, $T = 1$ and $E = 1$, and we see that at expiry the option value reduces to the "hockey stick" payoff $\max(S - E, 0)$. The time zero solution $C(S, 0)$ evaluated at the initial asset price $S = S_0$ solves the option valuation problem introduced earlier.

Although the basic European call option valuation problem has a simple analytical solution (under the Black–Scholes assumptions), many variations of the problem currently require use of numerical methods. In particular, a rich variety of *exotic* options are traded. They differ from the European call in that the payoff depends not only on the final time asset price, but also on its behavior during all or part of the time interval $[0, T)$. For example, the payoff might depend on the maximum, minimum, or average asset price, or might *knock-in* or *knock-out*, that is, switch on or off, depending on whether the price crosses a pre-

```
% MC Monte Carlo valuation for a European call
%

%%%%%%%%%% Problem and method parameters %%%%%%%%%%%%
S = 2; E = 1; r = 0.05; sigma = 0.25; T = 3; M = 1e6;
randn('state',100)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Svals = S*exp((r-0.5*sigma^2)*T + sigma*sqrt(T)*randn(M,1));
Pvals = exp(-r*T)*max(Svals-E,0);
Pmean = mean(Pvals)
width = 1.96*std(Pvals)/sqrt(M);
conf = [Pmean - width, Pmean + width]
```

Figure 4. Listing of the code `mc.m` for Monte Carlo valuation of an option.

determined barrier. As well, the option might have an *early exercise facility*, which gives its holder the freedom to exercise before the expiry date. The design and analysis of numerical methods for valuing exotic options is still a very active research topic. Let's look at three standard approaches to the simple European call case, with the motivation that such methods are needed in more exotic circumstances.

Monte Carlo Method

An extremely useful property of the Black–Scholes option value is that it can be regarded as the average payoff, suitably discounted for interest, under the risk neutrality condition $\mu = r$. In other words, we can reproduce the option value by setting $\mu = r$ in the asset model and computing the payoff average over all asset paths. In practice, we can do this through Monte Carlo simulation—averaging the payoff over a large number of asset paths. For a European call option, we only need to know about the asset price at expiry, so we can take $\Delta t = T$ in each path. A suitable pseudocode algorithm is

for i = 1 to M
  set $S_i = S_0 e^{(r-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}\xi_i}$
  set $P_i = e^{-rT} \max(S_i - E, 0)$
end
set $P_{mean} = \frac{1}{M}\sum_{i=1}^{M} P_i$

set $P_{var} = \frac{1}{M-1}\sum_{i=1}^{M}(P_1 - P_{mean})^2$

Here, P_i is the payoff from the ith asset path, discounted by the factor e^{-rT} to allow for payoff occuring at the future time $t = T$. The overall average P_{mean} is our Monte Carlo estimate of the option value. We can use the computed variance, P_{var}, to give an approximate 95-percent *confidence interval* $[P_{mean} - 1.96\sqrt{P_{var}/M}\ ,\ P_{mean} + 1.96\sqrt{P_{var}/M}\ ]$. Loosely, for sufficiently large M this interval will contain the true option value for 95 simulations out of every 100.

In Figure 4, we give a Matlab code that applies the Monte Carlo method. Here, we have made use of the vectorized mode of the random number generator and the high-level commands `mean` and `std`. The output is

```
>> mc
Pmean =
    1.1453
conf =
    1.1435   1.1471
```

and we recall that the Black–Scholes formula for these parameter values gave `C = 1.1447`. In Figure 5, we show how the Monte Carlo approximation varies with the number of samples, M. Here, we took $S = 10$, $E = 9$, $r = 0.06$, $\sigma = 0.1$ and $T = 1$. The xs in the figure give the Monte Carlo approximations and the horizontal lines show the extent of the confidence intervals. The Black–Scholes value is represented as a vertical dashed line.

Here are some project suggestions.

- Investigate using Monte Carlo for valuing a range of path-dependent options; that is, options with a payoff that depends on $S(t)$ for $0 \le t \le T$.
- Experiment with *variance reduction* methods for speeding up Monte Carlo computations.
- Investigate using *low-discrepancy sequences* and *Quasi Monte Carlo* methods in the context of option valuation.
- Test whether you can use the Monte Carlo method to compute approximations to *Greeks*—that is, partial derivatives of the Black–Scholes value $C(S, t)$ with respect to S, t or parameters such as σ and E.

If you are interested in learning more about Monte Carlo for option valuation, a good place to start is Phelim Boyle's

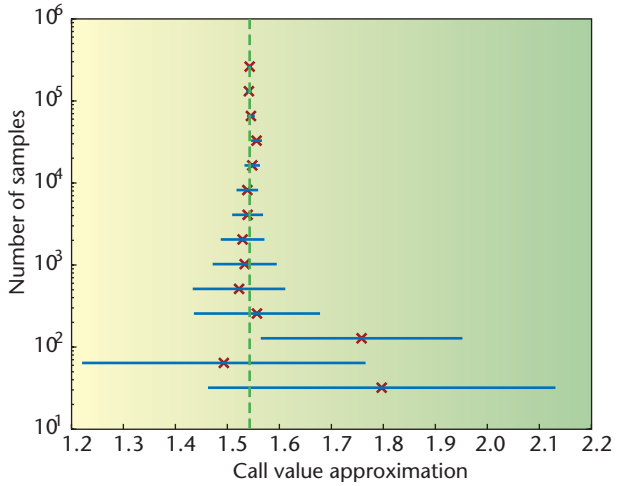

Figure 5. Monte Carlo approximations and confidence intervals for a European call option. The vertical axis represents the number of asset paths used. The *xs* gives the Monte Carlo approximations and the horizontal lines show 95 percent confidence intervals (valid for a large number of asset paths). The dashed vertical line shows the exact option value.

article,[6] which won the *Journal of Financial Economics*' All-Star paper Award 2002.

Binomial Method

The binomial method starts with a simplified asset price model. The time interval $[0, T]$ is discretized into equally spaced points $0 = t_0 < t_1 < \cdots < t_M = T$ with $t_i = i\Delta t$. Given asset price S_0 at time zero, we assume that the asset price at time t_1 arises from either a downward movement to dS_0 or an upward movement to uS_0, where $d < 1$ and $u > 1$. Then, at time t_2, the same restriction to down or up movements leads to three possible asset prices: d^2S_0, duS_0, and u^2S_0. Continuing this argument, there will be $i + 1$ possible asset prices at time $t_i = i\Delta t$, given by

$$S_n^i = d^{i-n}u^n S_0,\ 0 \le n \le i.$$

At expiry time $t_i = t_M = T$, there are $M + 1$ possible asset prices

$$\left\{S_n^M\right\}_{n=0}^M.$$

Letting

$$\left\{C_n^M\right\}_{n=0}^M$$

denote the corresponding expiry time payoffs from a European call option, we know that

$$C_n^M = \max(S_n^M - E, 0),\ 0 \le n \le M.$$

The binomial method proceeds by working backwards through time. An option value C_n^i corresponding to asset price S_n^i at time t_i is computed as a weighted average of the two asset prices C_n^{i+1} and C_{n+1}^{i+1} from time t_{i+1}. The formula is

$$C_n^i = e^{-r\Delta t}(pC_{n+1}^{i+1} + (1-p)C_n^{i+1}),\ 0 \le n \le i,\ 0 \le i \le M-1.$$

Here, we can regard parameter p as the probability of an upward movement in the asset price. The formula lets us go back to time zero and compute the required option value C_0^0. We must choose the method parameters Δt, u, d, and p so that the binomial asset model matches the Black–Scholes version in the $\Delta t \to 0$ limit. Once Δt is fixed, we get two equations for the three remaining parameters, and consequently many possible solutions. A popular choice is

$$d = A - \sqrt{A^2 - 1},\ u = A + \sqrt{A^2 - 1},\ p = \frac{e^{r\Delta t} - d}{u - d},$$

where $A = \frac{1}{2}(e^{-r\Delta t} + e^{(r+\sigma^2)\Delta t})$.

Figure 6 lists a Matlab code that implements the binomial method by using a matrix-vector product to work backwards through time. The approximate option value `W = 1.1448` agrees well with the Black–Scholes value `C = 1.1447` from `bsf.m` in Figure 2.

Figure 7 shows how the error in the binomial approximation behaves as a function of M, in the case $S = 5$, $E = 3$, $T = 1$, $r = 0.06$ and $\sigma = 0.3$. Figure 7a covers $100 \le M \le 300$, and Figure 7b covers $700 \le M \le 1{,}000$. Although the error generally decreases, the convergence is by no means monotonic with M. The dashed lines in the figure has the form "constant/M"—it may be shown that the error converges at this rate. The highly oscillatory nature of the convergence has been the subject of a number of research articles.

Here are some project suggestions.

- Can the execution time of `binom.m` be improved? (See the *SIAM Review* article[7] for a discussion of this issue.)
- A key advantage of the binomial method is its ability to incorporate an *early exercise* facility. Follow this up by implementing the method for *American*, *Bermudan*, or *shout* options.[2,3]
- Investigate the literature on curbing the oscillations in the binomial error.

```
%BINOM Binomial method for a European call
%

%%%%%%%%%% Problem and method parameters %%%%%%%%%%%%
S = 2; E = 1; r = 0.05; sigma = 0.25; T = 3; M = 256;
dt = T/M; A = 0.5*(exp(-r* dt)+exp((r+sigma^2)*dt));
d = A - sqrt(A^2-1); u = A + sqrt(A^2-1);
p = (exp(r*dt)-d)/(u-d);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Option values at time T
W = max(S*d.^([M:-1:0]').*u.^([0:M]')-E,0);

B = (1-p)*eye(M+1,M+1) + p*diag(ones(M,1),1);
B = sparse(B);
% Re-trace to get option value at time zero
for i = M:-1:1
       W = B(1:i,1:i+1)*W;
end
W = exp(-r*T)*W;
```

Figure 6. Listing of `binom.m` for the binomial method.

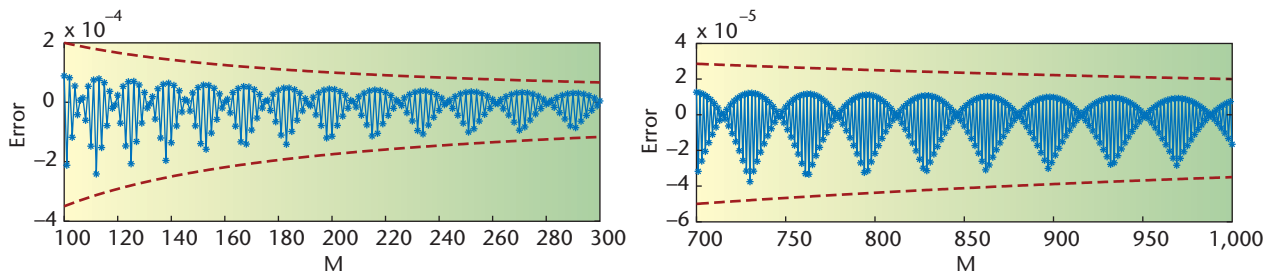

Figure 7. Error in binomial method as a function of *M*. As the disctretization is refined the error generally decreases, but the convergence is not monotonic.

- Examine whether you can use the binomial method to compute Greeks.

The paper that first introduced the binomial method remains highly accessible.[8]

Finite Differences for the Black–Scholes PDE

The Black–Scholes formula for the value of a European call option arises as the solution of a PDE. The PDE is of parabolic form, with Dirichlet boundary conditions. Unusually, a final time, rather than an initial time, condition completes the problem. Letting $\tau = T - t$ denote the *time to expiry*, we can convert to a more natural initial time specification. The PDE then has the form

$$\frac{\partial C}{\partial \tau} - \tfrac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} - rS\frac{\partial C}{\partial S} + rC = 0,$$

with initial data $C(S, 0) = \max(S(0) - E, 0)$ and boundary conditions $C(0, \tau) = 0$, $C(S, \tau) \approx S - Ee^{-r\tau}$ for large S, on the domain $S \geq 0$ and $0 \leq \tau \leq T$. Truncating the S range to $0 \leq S \leq L$ and using a finite difference grid $\{jh, ik\}$ with spacings $h = L/N_x$ and $k = T/N_t$, we can compute a discrete solution $V_j^i \approx C(jh, ik)$. Letting

```
%FORWARD Forward Time Central Space on Black-Scholes PDE
%        for European call
%

clf
%%%%%% Problem and method parameters %%%%%%
E = 4; sigma = 0.5; r = 0.03; T = 1;
Nx = 11; Nt = 29; L = 10; k = T/Nt; h = L/Nx;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

T1 = diag(ones(Nx-2,1),1) - diag(ones(Nx-2,1),-1);
T2 = -2*eye(Nx-1,Nx-1) + diag(ones(Nx-2,1),1) + diag(ones(Nx-2,1),-1);
mvec = [1:Nx-1]; D1 = diag(mvec); D2 = diag(mvec.^2);
Aftcs = (1-r*k)*eye(Nx-1,Nx-1) + 0.5*k*sigma^2*D2*T2 + 0.5*k*r*D1*T1;

U = zeros(Nx-1,Nt+1); Uzero = max([h:h:L-h]'-E,0);
U(:,1) = Uzero; p = zeros(Nx-1,1);

for i = 1:Nt
    tau = (i-1)*k;
    p(end) = 0.5*k*(Nx-1)*((sigma^2)*(Nx-1)+r)*(L-E*exp(-r*tau));
    U(:,i+1) = Aftcs*U(:,i) + p;
end

waterfall(U'), xlabel('j'), ylabel('i')
```

Figure 8. Listing of `forward.m` for finite difference solution of the Black-Scholes PDE.

$$\boldsymbol{V}^{\mathrm{i}} = \begin{bmatrix} V_1^i \\ V_2^i \\ \vdots \\ \vdots \\ V_{N_x-1}^i \end{bmatrix} \in \mathbb{R}^{N_x-1}$$

denote the numerical solution at time level i, we have $\mathbf{V}^0$ specified by the initial data, and the boundary values $\mathbf{V}_0^i$ and $\mathbf{V}_{Nx}^i$ for all $1 \le i \le N_t$ specified by the boundary conditions. Using a forward difference for the t-derivative and central differences for the S-derivatives gives the explicit method

$$\frac{V_j^{i+1} - V_j^i}{k} - \frac{1}{2}\sigma^2 (jh)^2 \frac{(V_{j+1}^i - 2V_{j-1}^i + V_{j-1}^i)}{h^2} - rjh\left(\frac{V_{j+1}^i - V_{j-1}^i}{2h}\right) + rV_j^i = 0.$$

The method can be expressed in matrix-vector form

$$\mathbf{V}^{i+1} = F\mathbf{V}^i + \mathbf{p}^i, \text{ for } 0 \le i \le \mathrm{N}_t - 1,$$

where $F \in \mathbb{R}^{(N_x-1)\times(N_x-1)}$ is tridiagional and the boundary conditions determine the vector $\mathbf{p}^i \in \mathbb{R}^{N_x-1}$.

The Matlab code `forward.m` in Figure 8 implements this method; in Figure 9, we show the resulting Black-Scholes surface. We used Matlab's `waterfall` plotting command to emphasize the time-stepping nature of the finite-difference iteration.

Here are some project suggestions.

- Speed up the code `forward.m` by replacing the matrix-vector multiplications with vector operations that access subarrays using Matlab's colon notation. (See Chapter 5 of *Matlab Guide* for details about colon notation.[1])
- Investigate the accuracy and stability of the explicit method in `forward.m`, and evaluate the improvements from (a) Crank-Nicolson and (b) upwinding for the $\partial C/\partial S$ term.
- Show that the binomial method may be regarded as a finite difference method and use this viewpoint to explain its convergence properties.
- Investigate the use of finite difference methods to value American options.

Having taught my class and written a textbook,[3] I see financial option valuation as an excellent and popular vehicle for motivating techniques in scientific computation.

Acknowledgments

A research fellowship from the Leverhulme Trust supported this work.

References

1. D.J. Higham and N.J. Higham, *Matlab Guide*, SIAM, 2000.
2. J.C. Hull, *Options, Futures, & Other Derivatives*, Prentice Hall, 2000.
3. D.J. Higham, *An Introduction to Financial Option Valuation*, Cambridge Univ. Press, 2004.
4. T.A. Bass, *The Predictors*, Penguin, 1999.
5. F. Black and M. Scholes, "The Pricing of Options and Corporate Liabilities," *J. Political Economy*, vol. 81, May/June, 1973, pp. 637–659.
6. P.P. Boyle, "Options: A Monte Carlo Approach," *J. Financial Economics*, vol. 43, no. 4, 1997, 323–338.
7. D.J. Higham, "Nine Ways to Implement the Binomial Method for Option Valuation in Matlab," *SIAM Rev.*, vol. 44, no. 4, 2002, pp. 661–677.
8. J.C. Cox, S.A. Ross and M. Rubenstein, "Option Pricing: A Simplified Approach," *J. Financial Economics*, vol. 7, no. 3, 1979, 229–263.

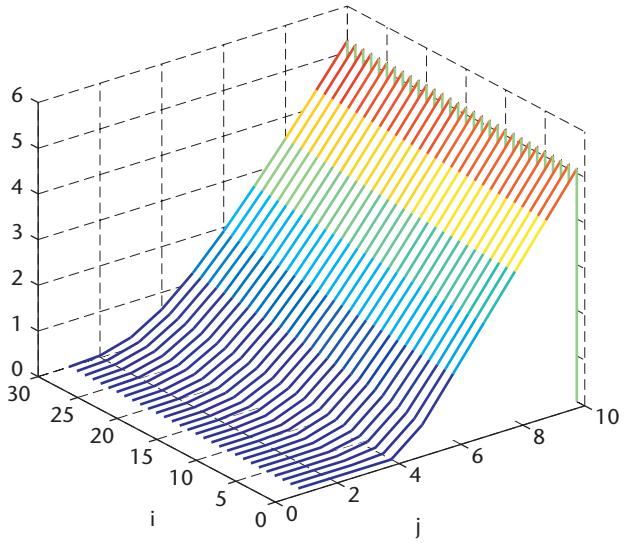

Figure 9. Finite difference Black–Scholes surface from `forward.m`. This is the output from `forward.m` in Figure 8.

Desmond J. Higham is a professor of mathematics at the University of Strathclyde, in Glasgow, Scotland. His research interests revolve around numerical analysis, especially stochastic computation, with applications in mathematical finance and bioinformatics. He serves on the editorial board of the *SIAM Journal on Scientific Computing*. Contact him at djh@maths.strath.ac.uk.

International Conference on Computational Science

Emory University, Atlanta Georgia

22–25 May 2005

Meeting Theme: Advancing Science through Computation

Education Session: High Performance Computing in Academia: Systems and Applications

Following on the success of the 2004 meeting in Krakow, preparations are underway for the fifth annual ICCS conference. There is still room for a few more papers in the education session. Please contact Dr. Denis Donnelly (donnelly@siena.edu) or Dr. Ulrich Ruede (ruede@immd10.informatik.uni-erlangen.de) for further information.